Energy Optimization Strategy for System-Operational Problems

ENERGY OPTIMIZATION STRATEGY FOR SYSTEM-OPERATIONAL PROBLEMS

BY

DHAFAR AL-ANI, B.Sc., M.Sc., M.A.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

© Copyright by Dhafar Al-Ani, 2012

All Rights Reserved

Doctor of Philosophy (2012) McMaster University

(Mechanical Engineering) Hamilton, Ontario, Canada

TITLE: Energy Optimization Strategy for System-Operational Problems

AUTHOR: Dhafar Al-Ani

B.Sc. (Aeronautical Engineering)

University of Baghdad, Iraq

M.Sc. (Applied Mechanical Engineering)

Jordan University of Science & Technology, Jordan

M.A.Sc. (Mechatronics Engineering)

American University of Sharjah, United Arab Emirates

SUPERVISOR: Dr. Saeid Habibi,

NUMBER OF PAGES: vii, 360

To My Beloved Mother,

To My Darling Wife,

To My Adorable Children,

To My Dearest Parents-in-Law,

&

To the Loving Memory of My Father

Abstract

The water supply industry is a very important element of a modern economy; it represents a key element of urban infrastructure and is an integral part of our modern civilization. Billions of dollars per annum are spent internationally in pumping operations in rural water distribution systems to treat and reliably transport water from source to consumers.

In this dissertation, a new multi-objective optimization approach referred to as energy optimization strategy is proposed for minimizing electrical energy consumption for pumping, the cost, pumps maintenance cost, and the cost of maximum power peak, while optimizing water quality and operational reliability in rural water distribution systems. Minimizing the energy cost problem considers the electrical energy consumed for regular operation and the cost of maximum power peak. Optimizing operational reliability is based on the ability of the network to provide service in case of abnormal events (e.g., network failure or fire) by considering and managing reservoir levels. Minimizing pumping costs also involves consideration of network and pump maintenance cost that is imputed by the number of pump switches. Water quality optimization is achieved through the consideration of chlorine residual during water transportation.

An Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) algorithm that combines the existing and new concept of Pareto-front, operating-mode specification, selecting-best-efficiency-point technique, searching-for-gaps method, and modified K-Means clustering has been proposed. APC-MOPSO is employed to optimize the above-mentioned set of multiple objectives in operating rural water distribution systems.

Saskatoon West is, a rural water distribution system, owned and operated by Sask-Water (i.e., is a statutory Crown Corporation providing water, wastewater and related services to municipal, industrial, government, and domestic customers in the province of Saskatchewan). It is used to provide water to the city of Saskatoon and surrounding communities. The system has six main components: (1) the pumping stations, namely Queen Elizabeth and Aurora; (2) The raw water pipeline from QE to Agrium area; (3) the treatment plant located within the Village of Vanscoy; (4) the raw water pipeline serving four major consumers, including PCS Cogen, PCS Cory, Corman Park, and Agrium; (5) the treated water pipeline serving a domestic community of Village of Vanscoy; and (6) the large Agrium community storage reservoir.

In this dissertation, the Saskatoon West WDS is chosen to implement the proposed energy optimization strategy. Given the data supplied by Sask-Warer, the scope of this application has resulted in savings of approximately 7 to 14% in energy costs without adversely affecting the infrastructure of the system as well as maintaining the same level of service provided to the Sask-Water's clients.

The implementation of the energy optimization strategy on the Saskatoon West WDS over 168 hour (i.e., one-week optimization period of time) resulted in savings of approximately 10% in electrical energy cost and 4% in the cost of maximum power peak. Moreover, the results showed that the pumping reliability is improved by 3.5% (i.e., improving its efficiency, head pressure, and flow rate). A case study is used to demonstrate the effectiveness of the multi-objective formulations and the solution methodologies, including the formulation of the system-operational optimization problem as five objective functions. Beside the reduction in the energy costs, water quality, network reliability, and pumping characterization are all concurrently enhanced as shown in the collected results. The benefits of using the proposed energy optimization strategy as replacement for many existing optimization methods are also demonstrated.

Acknowledgements

The completion of this dissertation has been an important accomplishment in my academic career. Accordingly, I am taking this opportunity to express my sincere gratitude to those who have helped me throughout my academic career. If it were not for my lovely wife (*Saba*) and dearest children (*Sama*, *Sala*, *Yousef* and *Abdullah*), who have supported me throughout my entire academic journey, I would not have finished.

During this dissertation, I have been accompanied and supported by many people, and thereby I offer my sincerest gratitude:

- First and foremost, to my supervisor Professor *Saeid Habibi* for providing me with an opportunity to undertake a research topic that is truly challenging, intriguing, and important in the field of energy optimization systems. I am deeply grateful to him for his spectacular advice and encouragement. Without his support, this work would never have been.
- To my committee members Drs. *Gary Bone*, *Stephen Veldhuis*, and *Christopher Anand* for providing me with their valuable feedback over many supervisory committee meetings and this dissertation.
- To Mr. *Jeff Mander* from Sask-Water and the operators of Saskatoon West Water Distribution System for their valuable help during my work.
- To Natural Sciences and Engineering Research Council of Canada (NSERC) for their support of this work.
- To Dr. *Carlos A.C. Coello* - Computational Science Department at National Polytechnic Institute, Dr. *Gregorio Toscano-Pulido* - Laboratory of Information Technologies Center at IPN, Dr. *Ponnuthurai N. Suganthan* - School of Electrical & Electronic Engineering at Nanyang Technological University, Dr. *Shi-Zheng Zhao* - Researcher at Nanyang Technological University, Dr. *Manuel Lopez-Ibanez* - Postdoctoral Researcher at University of Liber de Bruexelles, and Dr. *Philip Jonkergouw* - Department of Civil Engineering at University of Exeter for their help in providing manuscripts, codes, and experimental results.
- To *Florence Rozato*, *Vania Loyzer*, and *Lily Sazz-Fayter* for being supportive during my stay at Mechanical Engineering Department.
- Finally, to my family for love, faith, and unyielding support. *God bless you all.*

Contents

List of Figures

List of Tables

List of Symbols

Symbol	*Definition*
A_k	Cross-sectional area of tank (k) where k: Index for the tank
$\bar{A}$	Average area between section 1 and section 2 (L^2)
A_1	Cross-sectional area of section 1(L^2)
A_2	Cross-sectional area of section 2 (L^2)
bp	Billing period
C	Hazen-Williams C-factor (also known as Velocity adjusted C-factor or Velocity adjusted C-factor)
C_f	Unit conversion factor (4.66 English, 10.29 SI) (Manning), or (880 English, 1.22 SI) (Valve Coefficient), or (4.73 English, 10.7 SI) (Hazen-Williams)
$C_{n,i}$	Chlorine residual concentration at node (n) during time period (i)
C_o	Reference C-factor
C_v	Valve coefficient [gpm/$(psi)^{0.5}$, $(m^3/s)/(kPa)^{0.5}$]
$\bar{\bar{C}}_{max}$	Max. free chlorine concentration
$\underline{C}_{min}$	Min. free chlorine concentration
D	Diameter (L)
D	Diameter (ft, m)
$D(Pc_i)$	Discharge pumped at time interval (i) using pump combination Pc
D_1	Impeller diameter 1
D_2	Impeller diameter 2
$d_{i,k}$	Water enter the tank at period (i) for tank (k)
$\bar{d}$	Mean of all d_i
EC	Electricity consumed by pump combination Pc at the time interval (i) (kW/h)
E_{bp}^{j}	Max. energy consumption during the billing period (bp)
f	Darcy-Weisbach friction factor

$f(x)$	Objective functions represents the problems needs to be optimized
g	Gravitational acceleration constant (L/T^2)
g_{best}	Best position found in the swarm at that time
$g_i(x)$	Inequality constrained
H	Total head (L)
H_{bar}	Atmospheric pressure (at altitude of pumps) (ft, m)
H_{Ek}	Head of tank (k) at the end of the simulation run
H_{Emax}	Max. head tank
H_{Emin}	Min. head tank
$H_{E\,Tp,k}$	Head at tank (k) at optimization period (Tp)
$HP_{max,\,q}$	Max. pressure head at node (q)
$HP_{min,\,q}$	Min. pressure head at node (q)
HP_q	Pressure (energy head (m)) at node (q)
H_s	Static head (ft, m) (water elevation on suction side of pump – pump elevation)
H_{Sk}	Head of tank (k) at the start simulation time
H_{vap}	Water vapor pressure (corrected for temperature) (ft, m)
H_1	Pump head at speed 1 (L)
H_2	Pump head at speed 2 (L)
h_i	Tank water level (i)
$h_j(x)$	Equality constrained
h_L	Head loss due to friction (L)
h_l	Static lift (L)
h_{loss}	Sum of head and minor losses (from suction tank to pump) (ft, m)
h_m	Head loss due to minor losses (L)
h_{max}	Max. tank water level
h_{min}	Min. tank water level
i	Time interval period
j	Daily period index
K_L	Minor loss coefficient
K_M	Minor loss resistance coefficient (T^2/L^5)

K_P	Pipe resistance coefficient (T^z/L^{3z-1})
k	Index for the pipes in the network
L	Distance between section 1 and section 2 (L)
L	Distance between sections 1 and 2 (ft, m)
L_e	Equivalent length of pipe (L)
$Maxiter$	Total number of iterations
m	Index for the demand pattern
m_1	Power of the upper bound term usually equal to 2
m_2	Power of the lower bound term usually equal to 2
N	Perimeter of pipeline cross-section (L)
NBP	Number of the billing periods
ND	Total number of nodes in the network
$NPSH_a$	Net positive suction head available (ft, m)
$NPSW_{max}$	Max. total number of allowable pumping switches for the whole duration
NT	Total number of tanks in the network
n	Manning roughness coefficient
n_1	Pump speed 1 (1/T)
n_2	Pump speed 2 (1/T)
P	Pressure ($M/L/T^2$)
Pc	Pump combination at time interval (i)
Pu	Pump combination at interval (*i*)
P_1	Pressure at section 1($M/L/T^2$)
P_2	Pressure at section 2 ($M/L/T^2$)
p_{best}	Best position found by the particle by far
$\vec{p}_{ig}$	Swarm best vector
$\vec{p}_{il}$	Particles best vector
Q	Pipeline flow rate (L^3/T)
Q	Pipeline flow rate (cfs, m^3/s)
Q_1	Pump flow at speed 1 (L^3/T)
Q_2	Pump flow at speed 2 (L^3/T)
Q_i	Inflow to node in i-*th* pipe (L^3/T)

q	Index for the node
Re	Reynolds number
S_k	Surface area of tank (k)
T_{max}	Max. demand charge \$/max kW for the pump combination during billing period
Tp	Total optimization period (24 hrs x 30 days)
T_r	Tariff rate (Energy unit cost \$/KW.h)
U	Water used at node (L^3/T)
V	Average fluid velocity (L/T)
V_{max}	Maximum allowable flow speed in pipe (l/s)
V_o	Reference value of velocity at which C_o is determined (L/T)
$x = [x_1, \dots, x_n]^T$	Vector of decision variables that is mapped in the decision space
$x_{k\ lower}$	Lower limit for k element in this decision vector
$x_{k\ upper}$	Upper limit for k element in this decision vector
$\vec{\mathrm{v}}_{\mathrm{t+1}}$	Vector of the new particles velocities
w	Weighted parameter
$w_{\max}$	Maximum weighted parameter
$w_{\min}$	Minimum weighted parameter
$\vec{\mathrm{x}}_{\mathrm{t+1}}$	Vector of the new particles positions
Z	Elevation (L)
Z_1	Elevation of centroid of section 1 (L)
Z_2	Elevation of centroid of section 2 (L)
z	Exponent on flow term
z	2 (Darcy-Weisbach)
z	1.852 (Hazen-Williams)
z	2 (Manning)
$\frac{dS}{dt}$	Change in storage (L^3/T)
α	Angle of the pipe to horizontal
γ	Fluid specific weight ($M/L^2/T^2$)
ε	Index of internal pipe roughness (L)

μ	Absolute viscosity (M/L/T)
ρ	Fluid density (M/L^3)
τ_σ	Shear stress along pipe wall ($M/L/T^2$)
$\forall$	Universal quantification (for all, for any, for each). $\forall$ x: P(x) means P(x) is true for all x.
$\in$	Set membership (is an element of, is not an element of). A $\in$ S means A is an element of the set S; A $\notin$ S means A is not an element of S.
$\exists$	Existential quantification (there exists). $\exists$ x: P(x) means there is at least one x such that P(x) is true.
$\neg$	Logical negation (not). The statement $\neg$A is true if and only if A is false.
’	Derivative (prime) f′(x) is the derivative of the function f at the point x.
^{T}	Transpose: Swap rows for columns (matrix operation).
$\subseteq$	(subset) $A \subseteq B$ means every element of A is also an element of B
$\nexists$	$\nexists$ x: $P(x)$ means there is exactly one x such that $P(x)$ is true.
$\prec$	$L_1 \prec L_2$: means that the problem L_1 is dominated L_2
$\wedge$	Logical conjunction or meet in a lattice
$\varnothing$	Empty element in a vector
$\lvert . \rvert$	Represents first-norm of a vector
$\lVert . \rVert$	Represents second-norm of a vector

Chapter 1

Introduction

1.1 Preface

Globally, the overall water demand (i.e., residential, commercial, and industrial) is increasing while the sources (e.g., rivers, lakes, wells, etc.) are more limited. Water demand forecasting and management are therefore increasingly more important. In water distribution systems, water is routinely pumped from a source to supply nodes in a network in which water is stored in reservoirs and then consumed. In water distribution systems, pumping operations are expensive as they usually consume electrical energy.

Water quantity and quality vary depending on the physical characteristics of pipeline, reservoirs, and, pumps as well as the level of technology and expertise applied for their control. With today's high water production and supply costs, many researchers have been developing new strategies to optimize the operation of water distribution systems that may lead to reduce energy consumption.

During the last two decades, researchers have been developing various strategies for optimizing pump operations. The problem of pump scheduling (i.e., as an optimization problem) is difficult to solve because of the large search space, computational complexity, and the nonlinear and discontinuous nature of the problem.

Classical optimization techniques (e.g., gradient-based methods) are useful in finding the optimum solution of continuous and differentiable functions. Therefore, these methods are limited in scope as most practical applications involve objective functions that are not continuous and/or differentiable, [1]. Unlike classical optimization techniques, heuristic techniques (free derivative-based methods) have continuously shown competency and proficiency in solving multi-objective optimization problems (MOOPs). In particular, heuristic algorithms that mimic nature's evolutionary pattern have been proposed, including:

- Evolutionary Computation (EC): based on biological evolution, [2];

- Genetic Algorithms (GAs): based on imitating the natural evolution (e.g., mutation, selection, and crossover), [3,4];
- Deferential Evolution (DE): based on intelligent use of evolution strategy (e.g., differential mutation), [5];
- Ant Colony (AC): based on swarm intelligence theory, [6]; and
- Particle Swarm Optimizer (PSO): based on individual and social behavior using communication among agents, [7].

Multi-objective optimization relies on Pareto optimality and its aim is to find the Pareto-optimal front. A tentative solution (i.e., a vector of decision variables) is referred to as non-dominated or Pareto-front (PF), *if* it cannot be replaced by another solution which improves an objective without worsening another one. This concept usually does not give a single solution, but rather a set of quantitative solutions called the Pareto-front set that can be used for qualitative comparison. Finding such Pareto-front set is the goal when solving a multi-objective optimization problem.

There are two common conflicting forces in all multi-objective heuristics, particularly those that are population-based: *exploration* and *exploitation*. Hence, every good heuristic technique needs to maintain the right balance between the extent of exploration in the search space and the extent of exploitation of the information obtained up until the current searching iteration. Such a trade-off between the ability of maintaining diversity of the obtained solutions (i.e., globally searching the entire feasible region) and the ability of achieving convergence (i.e., locally searching promising sub-regions) will allow the heuristic techniques to have an adequate and effective search property, [8,9,10,11].

Furthermore, excessive exploitation will induce premature convergence and depress diversity. On the other hand, excessive exploration will result in slow convergence and arbitrary long computational time. Thus, in most advanced heuristic algorithms, these issues are controlled by suggesting an appropriate switching mechanism between exploration and exploitation during the search for better performing and flexible searching ability.

Despite the difference of heuristic techniques in their exploitation and exploration capability, they all share the common purpose of searching for an optimal, well-extended, and uniformly distributed Pareto-front for a given MOOP.

1.2 Problem Statement

Pumping cost and energy consumption are very large in water supply operations. Billions of dollars are yearly spent in pumping across water utilities worldwide. Hence, improving the operation of pumps can have large economic implications. Pumping optimization entails not only minimizing energy costs of Water Distribution Systems (WDS) without changing their basic elements, but also planning their operations within physical and operational constraints.

Saskatoon West WDS is a model for rural WDS that is chosen to implement the proposed Energy Optimization Strategy (EOS). This system supplies water to farmers, villages, residential areas, and large industrial users near the City of Saskatoon. This system is owned and operated by Sask-Water, which is interested in implementing the outcome of this research. In 2007, this network delivered 990 million gallons of water to five major customers across 36 kilometers of pipelines.

Pumping operation and energy optimization can be implemented without adversely affecting the supply capability and infrastructure of the WDS. The challenge ahead is to develop an energy optimization strategy, which is accurate, safe, and applicable to a wide range of WDS applications.

1.3 Dissertation Objectives

Pumping systems (e.g., water distribution systems, power plants, manufacturing factories, wastewater treatment facilities, etc.) consume nearly 25% of the energy consumed by electric motors, and on average account for approximately 30% of the total electrical energy usage in many industrial sectors, processes, and applications, [12]. Therefore, improving the design, retrofitting, and operating practices of pumping systems can have a large impact and would lead to significant opportunities to reduce operating costs.

In water distribution systems (WDS), pumps run continuously and require the heavy use of energy. Pumps are usually oversized and operated in a mode that is inefficient (i.e. they often operate away from their Best Efficiency Point (BEP)). Therefore, a considerable amount of energy is wasted. There are significant opportunities to operate pumps more effectively in a more, reliable, efficient, and economic manner through optimization.

This research considers the application of heuristics algorithms for minimizing the cost of electricity in pumping operations in rural water distribution systems while maintaining their safety and reliability. Most of the published works in optimizing the operations of pumps in WDS use empirical results combined with a bi-objective cost function including electricity (energy consumed) cost and pump scheduling (switching pumps ON or OFF). More recently, researchers have begun to consider other objectives, such as water quality and network reliability, [13,14,15].

1.3.1 Dissertation Hypothesis

The method proposed in this dissertation for energy optimization involves a combination of the proposed method of Particle Swarm Optimization (PSO) and the concept of Pareto-dominance, as well as new techniques such as:

- *Adaptive Parallel- Clustering-based Multi-objective PSO (APC-MOPSO)*: a multi-objective heuristic algorithm that is proposed to optimize the energy consumption and system operation of the Saskatoon West WDS.
- *Operating-Mode Pointer (OMP):* a technique that encodes the pumping operation into operating-modes that capture the pump scheduling, operational conditions, and system's settings and constraints. The operating modes are a set of feasible solutions (i.e., candidate solutions) that can be used by the optimization technique,
- *Searching-for-Gaps (SFG):* a technique that can enhance the performance of multi-objective heuristic techniques in terms of improving their diversity property,
- *Selecting-Best-Operating Point (SBOP):* a technique that utilizes the relationships between the pump's head, flow rate, speed, and efficiency to obtain an operating point that is closer to the Best Efficiency Point (BEP), and
- *Modified K-Means:* a technique that uses a modified version of K-Means as a clustering tool to improve the performance of the optimization technique.

The Energy Optimization Strategy (EOS) is applied to the Saskatoon West WDS. The implementation objective is to obtain savings of approximately 7 to 14% in electricity consumption without adversely affecting the service provided to Sask-Water's clients.

1.4 Novelty and Contributions

The main contribution of this dissertation is the development of a novel Energy Optimization Strategy (EOS) and its application to rural water distribution systems. The novel EOS generates new sets of operating-modes that enhance not only the network performance but also reduces the energy costs and in maintaining a high level of network reliability. The EOS method includes:

- A comprehensive literature review is presented on Multi-objective Particle Swarm Optimization (MOPSO) and their applications in the last two decades.
- A novel Multi-objective Particle Swarm Optimization algorithm, namely Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO).
- A novel Operating-Mode Pointer (OMP).
- A novel Searching-for-Gaps technique (SFG).
- A novel Selecting-Best-Operating Point technique (SBOP).

The energy optimization strategy (EOS) has unique properties:

- The APC-MOPSO is derivative free. *Its application to fifteen well-established benchmark problems indicates that the ACP-MOPSO provides the closest known Pareto-optimal front for all the problems considered*.

- Most remarkably, *its application to the real-world systems* (e.g., pumping systems that are complex, non-linear and have multiple peaks).

Finally, the proposed energy strategy has been applied to the Saskatoon West WDS. This application has entailed the following.

- A hydraulic model has been developed for the Saskatoon West WDS that considers both hydraulic operation as well as water quality.
- A new C++ interface has been developed for the hydraulic simulator (EPANET) and MATLAB.

1.5 Organization of the Dissertation

This dissertation is organized into seven chapters.

- Chapter 2 reviews the literature on Multi Objective Particle Swarm Optimization (MOPSO) algorithms. This comprehensive review is discussed under two broad headings. In the first part, the Particle Swarm Optimization's taxonomy, background, and state-of-the-art algorithms are comprehensively reviewed. These include modified, hybrid, parallel, cluster-based, and dynamic environment PSO versions that have been developed during the last two decades. The second part provides a survey of PSO applications, trends for future PSO applications, and ideas for new PSO algorithms.
- Chapter 3 investigates the physical structure (i.e., elements) of the Saskatoon West WDS as well as its dynamic behavior. A physical interpretation of the network's parameters is provided. A validated hydraulic model is developed to determine the flow of water in each pipe, the pressure at each node, the elevation of the water level in each tank, and the chlorine concentration throughout the network by simulation. A detailed discussion about the physical constraints (i.e., network limitations and operating restrictions) as well as the procedure used in modeling the network delivery profile is also given. The simulation results are compared with measured network data (provided by Sask-Water) for both steady-state and extend simulation period cases.
- Chapter 4 represents the new Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) algorithm. This algorithm is developed for constrained discrete-time nonlinear multi-modal multi-objective problems. It uses the following concepts: (i) Pareto-front, (ii) parallel computing (iii) modified K-Means clustering, (iv) external repository, and (v) dynamic model for updating the particle velocities and positions. The computational complexity of the APC-MOPSO is also considered. Its implementation on well-known benchmark test problems of APC-MOPSO is presented and comprehensively discussed. The APC-MOPSO is then compared with other well-known multi-objective optimization algorithms.

- Chapter 5 presents three new methods that augment the APC-MOPSO. These are the Searching-for-Gaps (SFG), Operating-Mode Pointer (OMP) and the Selecting-Best-Operating Point (SBOP). The new Searching-for-Gaps technique is used to enhance APC-MOPSO's exploration and exploitation abilities. The Operating-Mode Pointer and the Selecting-Best-Operating Point are designed to address the importance of obtaining a set of pumping operations that can be implemented on a real WDS in a safe and reliable manner.
- Chapter 6 amalgamates the new concepts and strategies considered in Chapters 4 and 5. The Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) algorithm is developed and used to optimize the Saskatoon West WDS. Hydraulic simulation for the network dynamics is directly integrated with the proposed APC-MOPSO algorithm to investigate both the steady-state and quasi steady-state (or called as extent period simulation) of the network behavior. Finally, a case study is used to evaluate the performance of the proposed energy optimization strategy when applied to the Saskatoon West WDS. The application considers energy consumption and cost, maintenance cost, water quality, network safety, and network reliability.
- Chapter 7 provides the concluding remarks.

Chapter 2

Literature Review

This chapter reviews a variety of methods for optimization problems that are often complex and involve functions with multiple peaks and multiple constraints. These methods often require a large computational power. In the past two decades, powerful heuristic algorithms have been developed in line with increasing power of computers. Promising new concepts that have emerged based on swarm intelligence theory, notably the Particle Swarm Optimization (PSO). A growing body of researchers in the optimization community is increasingly drawing on PSO to tackle hard optimization problems. The reasons behind the rising popularity of PSO is that it offers better searching efficiency (i.e. exploring and exploiting capabilities) for finding the global optimal solution, while providing fast convergence and reduced computational complexity.

This chapter provides the PSO's taxonomy, background, and state-of-the-art algorithms. These include the six categories that most of the PSO algorithms belong in modified, hybrid, parallel, clustering-based, and dynamic environment that have been developed during the last two decades. Furthermore, this chapter reviews different PSO applications and a variety of PSO numerical experiments.

2.1 Introduction

In 1858, Charles Darwin pieced together his evolutionary theory of natural selection. Evolution theory is originally defined as cumulative events occurring consequently and resulting in small genetic variations in the whole population. These variations usually result in organisms that are better adapted to their environment. Natural selection can be defined as variations in the genotype that increase an organism's likelihood of survival and reproduction. Organisms therefore prevail from generation to generation sometimes at the expense of others, [16]. Hence, evolutionary theory is still today one of the bedrocks of modern biology.

While Evolutionary Algorithms (EAs) are known as optimization tools that mimic the competitiveness of nature (i.e., reflect the nature and the theory of natural evolution or selection). Particle Swarm Optimization (PSO), originally proposed by Kennedy J. and Eberhart R.C. [7],

and Ant Colony Systems (ACS), proposed by Colorni et al. [6], both are classified as advanced forms of cooperative heuristic optimization that imitate the swarm intelligence theory. PSO is known as a search tool using a set of cooperating (interacting) particles flying through the search domain each with a fitness level or factor. These particles are naturally controlled by forces that direct them in a direction that would improve their "fitness", eventually leading to the global optimum.

PSO is a population-based heuristic technique that involves a swarm of particles that fly around the search space looking for "better" solutions. There is increasing evidence that Particle Swarm Optimization outperforms Genetic Algorithms (GA) and other Evolutionary Algorithms (EAs) proposed in the past decade in solving difficult Multi Objective Problems (MOPs), [17]. Accordingly, PSO is one of the more popular heuristic techniques.

2.2 Heuristic Optimization Techniques

Several heuristic optimization techniques have been developed in the last five decades that enable solving optimization problems that were considered as impossible or hard to solve. One of the most famous definitions of heuristic optimization is that of Osman et al. in 1996, [18]. The given definition introduces heuristic optimization as an iterative population-based process. It uses memory combined with different methods such as artificial intelligence, biological evolution, and natural/physical sciences for exploring and exploiting the search spaces to obtain better near-optimal solutions. According to this definition, heuristics are flexible techniques that can be reformed to fit any specific problem. The well-known heuristic techniques that are inspired from nature include:

- Simulated Annealing (SA) based on artificial intelligence [19],
- Tabu Search (TS) based on memory skill [20],
- Evolutionary Computation (EC) based on biological evolution [2],
- Genetic Algorithms (GAs) based on natural and physical sciences [3,4],
- Deferential Evolution (DE) based on evolution strategy [5], and
- Ant Colony (AC) and Particle Swarm Optimizer (PSO) based on individual and social behavior using communication among agents [7,6].

In recent years, researchers have focused on new/other heuristic techniques, such as Shuffled Frog-Leaping Algorithm (SF-LA) [21], Harmony Search (HS) [22], Cross Entropy (CE) [23], and Scatter Search (SS) [24]. These approaches interestingly showed great capabilities in solving Non-deterministic Polynomial (NP) problems, [25].

The basic techniques of heuristic are provided as follows, [26]:

- *Trajectory method:* current solution slightly modified (i.e., its search direction or state) by searching within the neighborhood in the search space. Notably examples are Simulated Annealing (SA) and Tabu-Searching (TS),

- *Discontinuous method:* exploit the full search space for new solutions. Discontinuity is induced by generating starting solutions which correspond to jumps in the search space. Examples include Genetic Algorithm (GA), Ant Colony (AC), and Particle Swarm Optimization (PSO),
- *Single agent method:* one single solution per iteration is processed. This is typically the case in Simulated Annealing (SA) and Tabu-Searching (TS),
- *Multi-agent or population based method:* a group of solutions (i.e., population) is used to explore in the search space. They constitute the algorithm's collective experience. Genetic Algorithms (GA), Ant Colony (AC), and Particle Swarm Optimization (PSO) are examples of the multi-agent method.
- *Guided search or search with memory usage method:* incorporates extra rules on where to explore in the search space. Examples are PSO whereas a swarm represents memory of the previously and recent flight experience, GA whereas a population represents memory of recent search experience, and AC whereas a pheromone matrix represents an adaptive type of memory of previously visited solutions in the search space, and
- *Unguided search or memory-less method:* relies completely on the search heuristic.

A comparison of heuristic techniques listed above and their features is provided in Table 2.1.

Table 2.1: Common Heuristic Techniques and their Features

The symbol √ *means that the feature is present,* (√) *means that the feature is partially present while* no *means not present.*

Features	**SA**	**TS**	**EC**	**GA**	**DE**	**AC**	**PSO**
Trajectory	√	√	√	(√)	no	(√)	(√)
Discontinuous	no	no	√	√	√	√	√
Single agent	√	√	no	no	no	no	no
Population based	no	no	√	√	√	√	√
Guided search	no	√	√	√	√	√	√
Unguided search	√	no	no	no	no	no	no

Heuristic techniques have the following advantages:

1. Most algorithms mimic nature,
2. Easy to implement,
3. Very flexible,
4. Readily handle constraints at low additional computational cost,
5. Robust to problem size, and
6. Solve hard non-deterministic polynomial (NP) problems.

On the other hand, heuristic techniques have the following disadvantages:

1. Require problem knowledge,
2. Convergence may not be,
3. Repeated searches may not guaranteed yielding the same solutions to the same problem, and
4. Computational complexity.

Classical optimization paradigms are preferable for problems that are simple, continuous, and differentiable. They fail to find a global optimum in problems that involve discrete-continuous or mixed variables, multiple competing objectives, discontinuity, and non-convex region. The disadvantages of heuristic optimization paradigms in terms of availability and efficiency are comparatively less pronounced.

In the early 1908s, "Meta-Heuristics" have developed and widely used to tackle several practical and difficult optimization problems. These families of approaches guide a subordinate heuristic by intelligently combining different concepts for exploring and exploiting the search space using a variety of learning strategies to effectively find near-optimal solutions, [18].

Recently, a new term in the optimization community is the "Hybrid Meta-Heuristic" technique; this refers to new approaches obtained by assembling different heuristics components in order to improve the robustness and performance of the search.

2.3 Multi-Objective Optimization Problems: Basic Concept

This section presents the basic concept and definitions used in Multi-objective Optimization Problems (MOOPs) (also known as multi-criteria, multi-dimensions, or vector optimization). MOOPs differ from the Single-objective Optimization Problems (SOOPs) by having not only one but a number of objective functions. These objective functions are usually in competition or in conflict with each other. Thereby, in solving such problems, many solutions can be obtained rather than having one optimum solution as in the SOOPs.

In real-world applications, optimization problems often need to consider many objectives and are thus better suited to a multi-objective optimization problem. The MOOPs can be utilized to provide multiple high-quality solutions. A decision maker can then select from a range of alternative solutions (e.g., Pareto-front), an option that would best fulfill the decision-makers preferences.

The general form of MOOP is as follows:

Minimize / Maximize $\boldsymbol{f}(x) = [f_1(x), f_2(x), \dots, f_m(x)]$

While the goal is to:

Find a solution $\boldsymbol{x}$ that optimizes $\boldsymbol{f}(x_m)$ $\quad m = 1,2, \dots, n$

Subject to

$g_i(x) \leq 0 \quad i = 1,2, \dots, q$

$h_j(x) = 0 \quad j = 1,2, \dots, p$

$x_{k\,lower} \leq x_k \leq x_{k\,upper} \qquad k = 1,2,\dots,l$

where:

$\boldsymbol{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables that is mapped in the decision space;

$f_m, \quad m = 1,2,\dots,n$ are the objective functions;

$g_i, \quad i = 1,2,\dots,q$ are the mathematical inequality constraint functions in the decision space;

$h_j, \quad j = 1,2,\dots,p$ are the mathematical equality constraint functions in the decision space;

$x_{k\,lower}, \; x_{k\,upper} \quad k = 1,2,\dots,l$ are called decision variable bounds that restrict each decision to be within the lower and upper bounds.

The aim of solving a multi-objective optimization problem is to trade-off between objectives, as seldom can all objectives be simultaneously satisfied. The concepts of dominance, Pareto optimality, Pareto-front set, and Pareto ranking are the powerful tools for qualitative and quantitative assessment of optimal solutions. These are presented as follows.

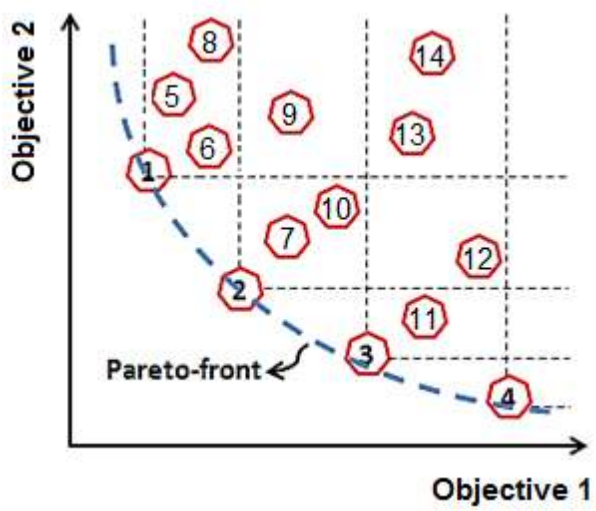
Definition 1: Pareto Dominance

In the context of multi-objective optimization, a solution candidate x_1 is said to dominate another candidate solution x_2 (denoted by $x_1 \preccurlyeq x_2$) *if and only if* $\boldsymbol{f}(x_1)$ is partially less than $\boldsymbol{f}(x_2)$ such as:

$\forall i \in \{1,2,\dots,k\} \qquad f_i(x_1) \leq f_i(x_2) \quad \wedge \quad \exists j \in \{1,2,\dots,k\} \quad f_j(x_1) < f_j(x_2)$

Based on the Dominance notion, it can be said that x_1 is preferable to x_2 when solution x_1 dominates solution x_2.

To illustrate the concept of *Pareto-dominance*, Figure 2.1 shows a set of candidate solutions for a minimizing bi-objective problem. In this figure, candidate solution 1 (x_1) dominates solutions {5, 6, 8, 9, 13, 14}, and four non-dominated solutions that are 1, 2, 3, and 4.



$1 \preccurlyeq 5, 6, 8, 9, 13, 14$
$1 \npreccurlyeq 7, 10, 11, 12$
$1 \npreccurlyeq 2, 3, 4$

$2 \npreccurlyeq 1, 3, 4$
$3 \npreccurlyeq 1, 2, 4$
$4 \npreccurlyeq 1, 2, 3$

$10 \preccurlyeq 13, 14$
$10 \npreccurlyeq 11, 12$

Figure 2.1: Example for the Dominance Relation in a Two-objective Minimization Function

Definition 2: Pareto Optimality

A feasible solution $x^* \in F$ (where F denotes the feasible region of the problem) is *Pareto-optimal*, *if and only if* there is no feasible solution $x_b \in F$ such that: $x_b \preccurlyeq x^*$.

According to this definition, x^* is referred to as *Pareto-optimal*, *if and only if* there does not exist any other feasible solution x_b that would improve some objectives without causing a simultaneous degradation in at least one other objective. Therefore, the solution to a multi-objective optimization problem (MOOP) considering Pareto optimality is a set of feasible, non-dominated solutions that is referred to as the *Pareto-optimal set*.

Definition 3: Pareto-optimal Set

For a given multi-objective problem, $\boldsymbol{f}(x)$, the *Pareto-optimal set* (P^*) is defined as:

$P^* \coloneqq \{x^* \in F |\ \neg\exists\, x \in F, \quad x \preccurlyeq x^*\}$

When the solutions in the *Pareto-optimal set* are plotted in the objective space (as illustrated in Figure 2.1), they are collectively called a *Pareto-front*.

Definition 4: Pareto-front

For a given multi-objective problem $\boldsymbol{f}(x)$ and its *Pareto-optimal set* (P^*), the *Pareto-front* (PF^*) is defined as:

$PF^* \coloneqq \{\boldsymbol{f}(x) = (f_1(x), f_2(x), \dots, f_n(x) |\ x \in P^*)\}$

Figure 2.2 depicts the Pareto-front of a bi-objective optimization problem.

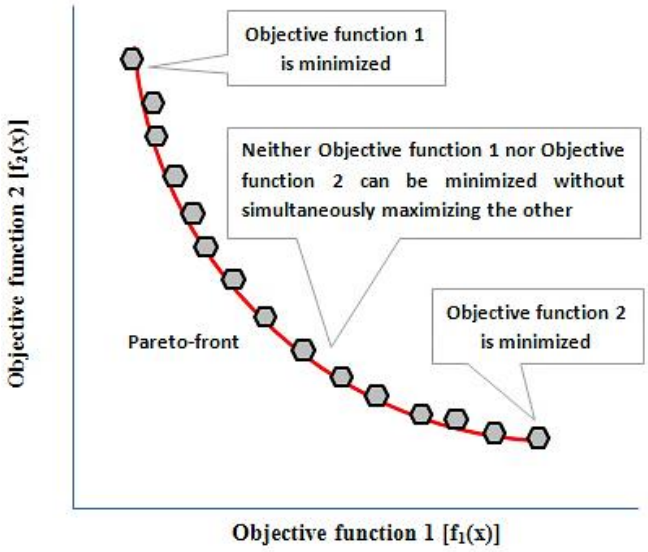


Figure 2.2: *Pareto-front* for Minimizing a Bi-objective Function

Definition 5: Pareto Ranking

An individual's rank corresponds to the number of individuals in the current population by which it is dominated. One of the simplest methods for computing fitness values is to let the

individuals directly reflect the Pareto-dominance relation. Figure 2.1 and Table 2.2 illustrate the Pareto-dominance relations in a population of 14 individuals for minimizing a bi-objective problem.

Table 2.2: The Pareto Domination Relation of the Individuals depicted in Figure 2.1

Solution	Is Dominated to	Is Dominated by	Rank
1	{5,6,8,9,13,14}	{∅}	0
2	{6,7,9,10,12,13,14}	{∅}	0
3	{11,12,13,14}	{∅}	0
4	{∅}	{∅}	0
5	{8,14}	{1}	1
6	{8,9,13,14}	{1,2}	2
7	{9,10,13,14}	{2}	1
8	{14}	{1,5,6}	3
9	{13,14}	{1,2,6,7}	4
10	{13,14}	{2,7}	2
11	{12,13,14}	{3}	1
12	{∅}	{2,3,11}	3
13	{14}	{1,2,3,6,7,9,10,11}	8
14	{∅}	{1,2,3,5,6,7,8,9,10,11,13}	11

Definition 6: Global and Local Pareto-optimal (see Figure 2.3)

Like global and local optimal solutions in the single-objective optimization, there will be global and local Pareto-optimal sets in multi-objective optimization (MOOP). Therefore, in the MOOP case, the concept of a globally optimal solution infers that an improvement in one objective results in deterioration in another. Similarly, local optima's may exist when a non-dominated set within a neighborhood is obtained.

Let $\acute{X} \in X$ be a set of decision vectors

i. The set $\acute{X}$ is denoted as a *local Pareto-optimal* set *if and only if*

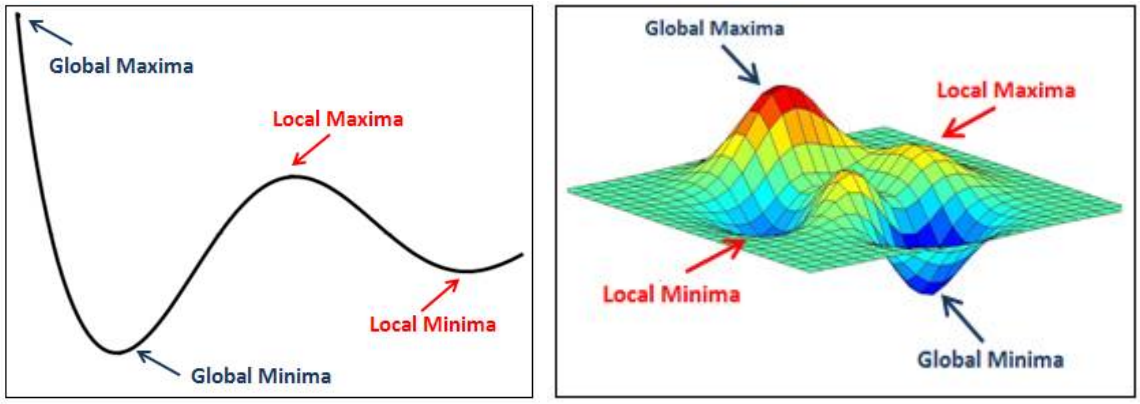
$\forall \acute{a} \in \acute{X} : \nexists a \in X : a \prec \acute{a} \wedge \|a - \acute{a}\| < \varepsilon \wedge \|f(a) - f(\acute{a})\| < \delta$

where

$\| . \|$ is the corresponding distance metric and $\varepsilon > 0,\ \delta > 0$.

ii. The set $\acute{X}$ is called a *global Pareto-optimal* set *if and only if*

$\nexists\ \acute{a} \in \acute{X} : \nexists\ a \in X : a \prec \acute{a}$



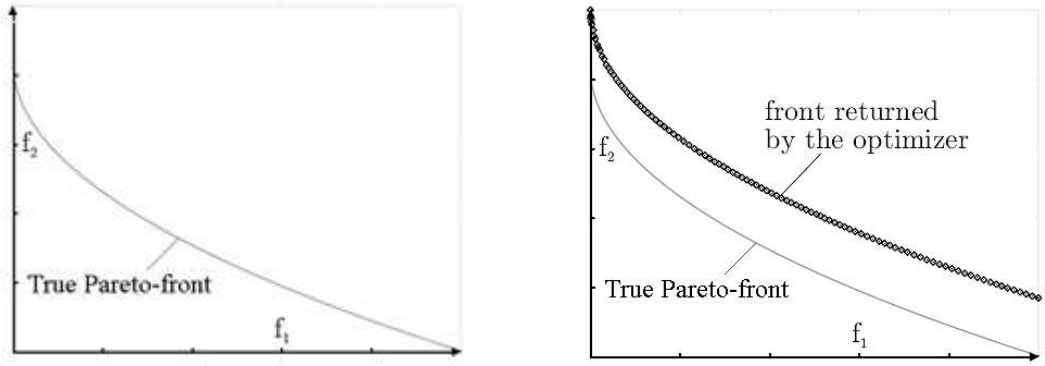
2- dimensional Representation 3-dimensional Representation

Figure 2.3: Local/Global Maxima/Minima are represented in the diagram

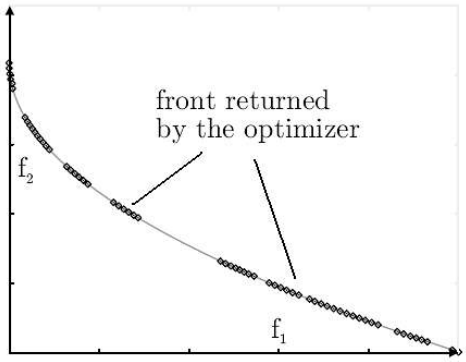
To illustrate the concept of Pareto-front set, Figure 2.4 shows many cases in which different set of non-dominated solutions are formed into different shapes based on their convergence and diversity. Figure 2.4.a shows the Pareto-optimal front (i.e., true Pareto-front) of a minimizing multi-objective optimization problem. Figure 2.4.b depicts a case in which the Pareto-front set has a good diversity but it lies far away from the Pareto-optimal front (i.e. poor quality). Figure 2.4.c shows another set of non-dominated solutions in which they are very close to the Pareto-optimal front (i.e. high-quality solutions) but have not covered the entire region in the objective space. Finally, Figure 2.4.d has a set of non-dominated solutions that are superior in both convergence and diversity.

According to the nature of the Multi Objective Problems (MOPs), Pareto-front sets can have different shapes. Figure 2.5 illustrates some examples of Pareto-front sets such as non-convex (concave), disconnected, linear, and non-uniformly distributed respectively.

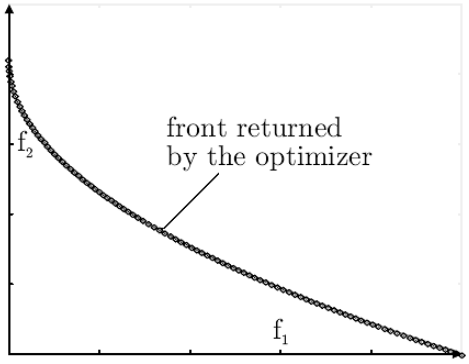
The last comment in this section is that not all global Pareto-optimal sets necessarily contain the Pareto-optimal solutions. A Pareto-optimal set refers to the entirety of the Pareto-optimal solutions. Hence, the set of objective vectors may be denoted as Pareto-optimal front.



a. Pareto-front Set for MOP b. Bad Convergence & Good Diversity

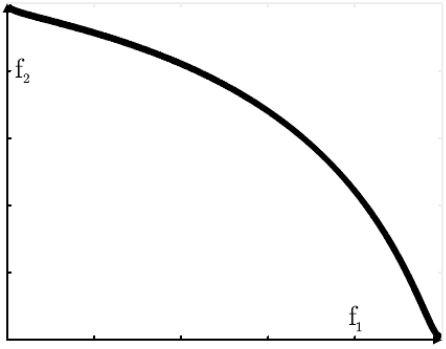


c. Good Convergence & Bad Diversity

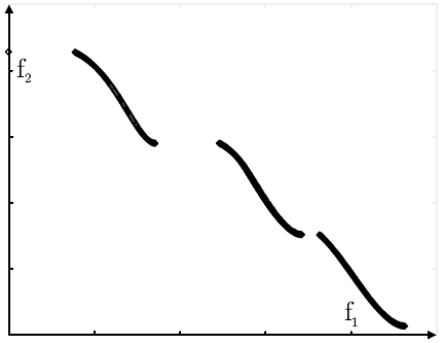


d. Good Convergence & Diversity

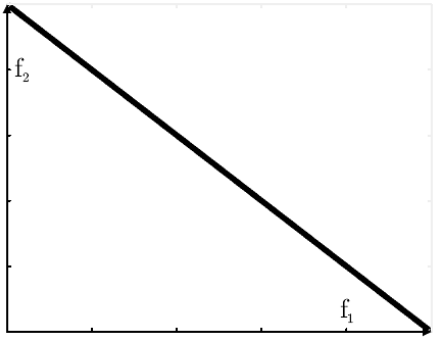
Figure 2.4: Examples of Approximate Pareto-front Set [27]



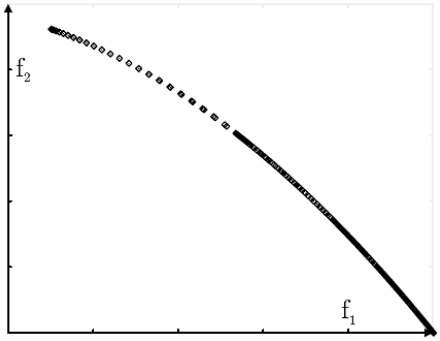
a. Non-Convex (Concave) Pareto-front



b. Disconnected Pareto-front



c. Linear Pareto-front



d. Non-Uniformly Distributed Pareto-front

Figure 2.5: Examples of Pareto-front Set

For more details on the concepts of global and local Pareto-optimal sets, Pareto dominance, Pareto-front and other theories, definitions, and lemmas in regards to multi-objective optimization, the readers are referred to [28].

2.4 Particle Swarm Optimization: Taxonomy

An Evolutionary Algorithm (EA) is an iterative stochastic process that is derived from the concept of natural evolution and ruled by the survival of the fittest principle. The standard Particle Swarm Optimization (PSO) is a global search technique that has its roots in two main methodologies; it originates from the notions of swarm intelligence and evolutionary programming, [7]. However, there is an apparent difference between standard PSO and other EA methods. Many EA techniques are drawn to the path of competition and commonly use some form of decimation (destruction) of the weak (worse) individuals and replace them by new (stronger) offsprings that can be generated from the genetic crossover between other individuals. In contrast, the standard PSO is modeled on the path of cooperation over competition in which the individuals are influenced by the best performance of their neighbors. They are never substituted or replaced during the run. These results in greater diversity and the PSO will tend to converge towards the Pareto-optimal front in a more stable manner compared with other EA approaches.

To understand the underlying differences between Swarm Intelligence (SI) and Evolutionary Algorithms (EAs), a comparative study that addresses some of their architectural differences is presented in Table 2.3 for which Particle Swarm Optimization (PSO) is used as a representative to the SI. Although Table 2.3 shows that PSOs and EAs share many common features, PSOs are obviously not considered a part of the EA family of algorithms in terms of their theoretical base, search strategy and evolution of their population. Preserving population diversity is essential for locating a global optimum, but is also crucial for coping with time-varying problems and tasks.

Table 2.3: Underlying Differences between PSO and EAs

The symbol √ means that the feature is present and × means that the feature is not present

Features	**PSO***	**EAs**
Heuristics in concept	√	√
Evolutionary computation style	×	√
Swarm intelligence style	√	×
Nature-inspired	√	√
Deterministic in solving	×	×
Stochastic in solving	√	√
Global search	√	√
Memory-based technique	√	√
Ignoring worst individuals	×	√
Population-based technique	√	√

Table 2.3: Underlying Differences between PSO and EAs (continue)

Features	PSO*	EAs
Leaders guide scheme	√	×
Parent representation scheme	×	√
Selection scheme	×	√
Mutation scheme	×**	√
Off-spring production	×	√
Single-objective problems	√	√
Multi-objective problems	√	√
Constraint optimization	√	√
Clustering scheme	√	√
Parallelism scheme	√	√
Pareto optimality concept	√	√
Premature convergence ***	√	√
Discrete decision variables	√	√
Continuous decision variables	√	√
Implicit-model representation (IR)	√	√

** This study is carried out considering standard and modified PSOs but not the hybrid ones.*

*** These features are incorporated in the hybrid versions of PSO and MOPSOs.*

**** Both methods cannot guarantee the convergence into global optimum.*

2.5 Particle Swarm Optimization: Basic Concepts

In the original PSO algorithm, each potential solution, called a particle, is compared to a bird flying across the objective space aiming to land on a global optimum region. In the context of PSO, a group of those particles is known as a swarm. Each particle in the swarm is assigned a fitness value, which is determined by the fitness function. Furthermore, each particle has the opportunity to adjust its trajectory, i.e. flying direction, towards the global region in the search space by adapting its velocity and position. Moreover, PSO is a memory-based technique, in which each particle is capable of recalling the instantaneous best position it ever visited as well as the swarm's best position. Each particle is biased towards its best previous position and towards the swarm best position. In general, there are two forms of PSO: continuous PSO and discrete PSO. The continuous version uses a real-valued multi-dimensional space to describe its trajectories. While the discrete uses binary values, i.e. either 0 or 1, with a certain probability scheme to describe its trajectories.

Problems can be formulated as either single objective or multiple objective functions. Accordingly, PSO can be formulated as Single Objective PSO (SOPSO), [7], or Multi Objective PSO (MOPSO), [29].

2.5.1 Basic PSO Terminologies

The term swarm used in PSO has its roots in the optimization literature; Millonas [30] used the "swarm" term when he developed his swarm intelligence model for applications in artificial life. Furthermore, he stated five main principles for swarm intelligence:

1. *The proximity principle:* the population (i.e., group of particles or individuals) should be able to perform searching in less computational time.
2. *The quality principle:* the population should be able to react to the quality factors in the environment.
3. *The diversity principle:* the population should not restrict its activities along extremely narrow channels.
4. *The stability principle:* the population should not alter its behavior every time the environment changes.
5. *The adaptability principle:* the population should be able to adjust its behavior whenever it needs in accordance with the expense of the computational complexity.

It is important to mention here that both principles 4 and 5 are in contradiction to one another. A brief explanation of the basic terminologies used in PSO algorithms is provided next.

- *Exploration:*

In any population-based optimization algorithm, exploration means the temptation of an individual (i.e. candidate solution or particle) in searching for or discovering new regions of the search space which has not been visited before. Further, the exploration process is applied with the aim of finding Pareto-optimal front or a better solution.

- *Exploitation:*

In any optimization algorithm, exploitation means the process of improving and combining the traits of the currently known particle(s) with the aim of gaining the utmost benefit from it.

- *Convergence:*

In any optimization algorithm, the search process is said to be converged *if and only if* there will be no further improvement to the candidate solution.

- *Premature Convergence:*

In any optimization algorithm, the search process is said to be prematurely converged into a local optimum *if and only if* there will be no further exploration of new areas of the decision space while there exists another region that contains a better solutions [31,32]. Figure 2.6 shows an example of premature convergence as the search process is trapped into a local minimum.

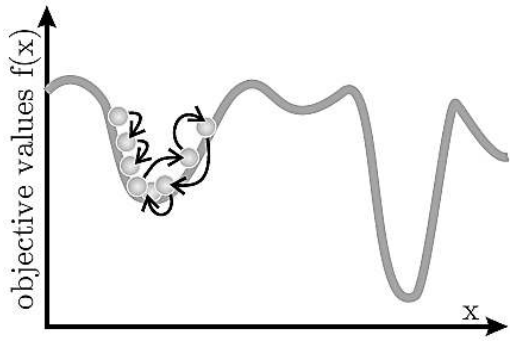


Figure 2.6: Example of Premature Convergence in Minimization Function [33]

- *Preserving Diversity:*

While solving a multi-objective problem with any population-based optimization algorithm, one of the main goals is diversity that is preserving a set of non-dominated solution candidates as widespread as possible. Furthermore, preserving diversity provides a balance between exploration and exploitation [34]. Interested readers can refer to the related publications on how diversity can be measured, including R.D. Routledge [35], A.E. Magurran [36], Morrison et al. [37], Paenke et al [34], and Wilke et al. [38].

As a measure of population (swarm) diversity, a parameter *D(t)* called the population average distance amongst particles is defined by Krink et al. [39]. The population average distance is expressed in Eq. (2.1):

$$D(t) = \frac{1}{M.L}\sum_{i=1}^{M}\sqrt{\sum_{d=1}^{N}(x_{id}^{t} - \bar{p}_d)^2} \qquad 2.1$$

where L denotes the length of the longest diagonal in the search space; M the size of the population; N the dimension of the solution space; x_{id}^{t} is the d-*th* dimension coordinate values of the i-*th* particle at *t-th* time step; $\bar{p}_d$ the average value of the d-*th* dimension coordinate values of all particles

The population average particle distance quantifies the degree of distribution between the particles in the population. The smaller *D(t)* is, the more focused the population will be. Note that

this diversity measure is dependent on the swarm size, the dimensionality of the problem, and the search range in each dimension.

2.5.2 Single PSO Algorithm

The original Particle Swarm Optimizer (PSO) algorithm was proposed by Kennedy and Eberhart, [7], and is mainly used to obtain optimal solutions for complex, non-linear, and non-convex problems. In PSO, individuals (so-called particles) that represent candidate or potential solutions for an optimization problem will be gathered in a set referred to as the swarm in which the behavior of each particle is governed by two adaptive physical properties that are velocity and spatial position. Hence, the particles fly throughout the search space towards possible regions of best solutions using an iterative technique. The equations that guide the particles' velocity and spatial position during their search are expressed as:

$$\vec{v}_{t+1} = v_t + c_1 \times r_1 \times (\vec{p}_{il} - \vec{x}_t) + c_2 \times r_2 \times \left(\vec{p}_{ig} - \vec{x}_t\right) \quad 2.2$$

$$\vec{x}_{t+1} = \vec{x}_t + \vec{v}_{t+1} \quad 2.3$$

Many modifications have been made to the original PSO algorithm, an example of that is the work done by Shi et al. [40]. In their work, a new additional design parameter, known as inertia weight factor (*w*) is introduced in order to adjust the balance between exploration (quantitative) and exploitation (qualitative) abilities within the searching technique (as expressed in Eqs. (2.4) and (2.5). The greater *w* is, the more iteration between particles is needed to attain convergence.

$$\vec{v}_{t+1} = w \times v_t + c_1 \times r_1 \times (\vec{p}_{il} - \vec{x}_t) + c_2 \times r_2 \times \left(\vec{p}_{ig} - \vec{x}_t\right) \quad 2.4$$

$$\vec{x}_{t+1} = \vec{x}_t + \vec{v}_{t+1} \quad 2.5$$

The second term of Eq. (2.4) called "cognitive" component, represents the exploration experience of each particle, i.e., the cognitive component that encourages the particles to move toward their own best positions found so far. The third term of Eq. (2.4) known as the "social" component that represents the collaborative influence of the individuals in a population in searching for global optimal solutions, i.e. social component draws the particles toward the swarm's best particle found so far.

It is obvious that in the case where the cognitive component is higher in value than the social component, there will be an excessive exploration of individuals through the search space. On the contrary, a relatively high social component's value may draw particles prematurely toward a local optimum. Figure 2.7 depicts the flight of a particle in the search space.

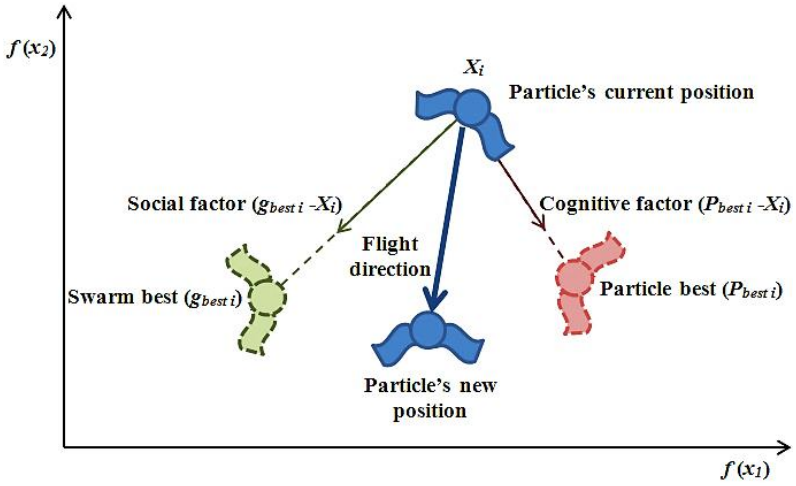


Figure 2.7: New Flying Direction for a Particle in the Search Space

2.5.3 Multi-Objective PSO Algorithm

When PSO was developed to solve multiple optimization problems, there were initially many concerns in the optimization community and even amongst the researchers that developed variant MOPSO algorithms. The concerns were based on the way that PSO could deal with the following:

- conflict objectives;
- quality of solutions;
- criteria of choosing the leader;
- memory strategy for non-dominant solutions;
- diversity along Pareto-front set; and
- premature convergence.

MOPSO algorithms have now evolved and have addressed the above concerns. They are capable of solving multi-objective optimization problems with high efficiency, accuracy, robustness, and stability. Pseudo-code 1 depicts the outline of a general MOPSO algorithm proposed by Reyes-Sierra et al. [41].

Pseudo-code 1: General MOPSO Algorithm

Initialize: swarm size (N), maximum iterations (*Maxiter*), *iter* = 0
Task: find Pareto-front set
for i =1:N
DO
Initialize randomly the position within the feasible region
Initialize randomly the velocity within the feasible region

end for
Initialize randomly the leaders in a repository
Initialize randomly the leaders for the swarm
Repeat
for i = 1:*N*
DO
Select leaders
Update Position (Flight)
Assess the fitness functions
Mutation
Update p_{best}
end for
Update leader in the repository
Update leaders for the swarm
iter = *iter* + 1
If (*iter* < *Maxiter*) or (stopping criterion is NOT satisfied)
return
end if
Obtain the results

2.5.4 Classical Social Structures in PSO

Although not all topologies used in computer networks can be directly implemented in the PSO's social structure. Nonetheless, this section highlights some well-known classical social topologies that have shown significant impact on the PSO's performance along the searching process that are star, ring, fully connected, and tree topologies.

For the ring topology, each particle updates its position according to the best solution found by its *n* closest neighbors. Figure 2.8.a shows the case in which each particle connects with two nearby neighbors ($n = 2$). The lower the value of *n*, the larger number of iterations required to achieve convergence of the swarm. The greater the value of *n*, the larger is the portion of the search space that is explored.

The fully connected social structure is another topology that can be used in a PSO algorithm and it is a special case of the ring topology in which *n* = swarm size - 1. Figure 2.8.b depicts the fully connected structure. This network presents the highest degree of connectivity among the particles in the swarm as all particles are connected to each other. In other words, all particles know each other's position and can adjust their own position according to the best position in the swarm. Therefore, all particles tend to follow a unique leader in the search for the global optimum. However, one of the major disadvantages of this topology is that the number of connections grows quadratically with the number of particles making it impractical for large

populations. Maderio et al. [42] concluded that the fully connected topology is more appropriate for single-objective optimization problems (SOOPs).

Star structure is one of the most common social topologies used in a PSO algorithm. It is a special case of the ring topology in which $n = 2$ except the central particle, in other words, each particle connects with two closest neighbors except the central particle that is connected to all other particles in the swarm as shown in Figure 2.8.c. In its simplest form, a star topology consists of one central particle that acts as means to broadcast the particles' best positions as well as the best position found in the swarm so far. When applied to a PSO, it enhances the chance of the swarm to search towards the global optimal regions (opportunity to obtain the best possible solutions). As in the fully connected type of social connectivity, a star topology is a preferable structure in the case of single-modal problems.

Kennedy et al. [43] presented the first comprehensive study on the performance of different sociomatric (i.e. this term is used by Kennedy to indicate the social communication channels between the particles in the swarm, it is also called as social network, topology). Von Neumann topology is defined as a two-dimensional grid (square lattice) in which the neighbors above, below, and on each side were connected. Figure 2.8.d depicts the shape of the social network.

A tree social structure in PSO is formed by dividing the whole swarm into several clusters (groups) in which each cluster can hold any number of particles, in other words, it is not necessary to have a uniform distribution of particles among the clusters. The connection of each cluster is chosen to be a fully connected topology while each cluster communicates with each other through a randomly chosen particle. This sole particle is acting as a messenger between clusters, thus, the information about better solutions found by a particular cluster conveys to the other clusters through that informant. To demonstrate how the information is transmitted and dealt with in a tree topology, Figure 2.8.e illustrates the tree topology in a case where there are four clusters. According to this figure, if the information, i.e., better global position found in a cluster by far, transmitted from one informant of cluster c_1 to the other informant of cluster c_2 is better than that in cluster c_2, the particles in c_2 would adapt their positions according to that new information.

Even though a Clan topology is proposed as being a new topology proposed by Carvalh et al, [44], it closely resembles the tree structure and should be classified as such [44]. The Clan topology is only differentiated from the tree topology by the mechanism that is used to select the informant particle in each cluster. In a tree structure, this occurs randomly whereas in the Clan topology the best solution amongst the cluster particles is chosen. Figure 2.8.f depicts the interconnections of the particles of four clan clusters. The idea of having a smaller group of leaders is to improve the PSO search performance wherein all particles in the swarm would adjust their positions according to the position of the best leader to be found in a sort of leaders group. Once those leaders update their positions, they convey the newly acquired information to their

corresponding clans. The results of the global search are then distributed locally to the other members of the clan.

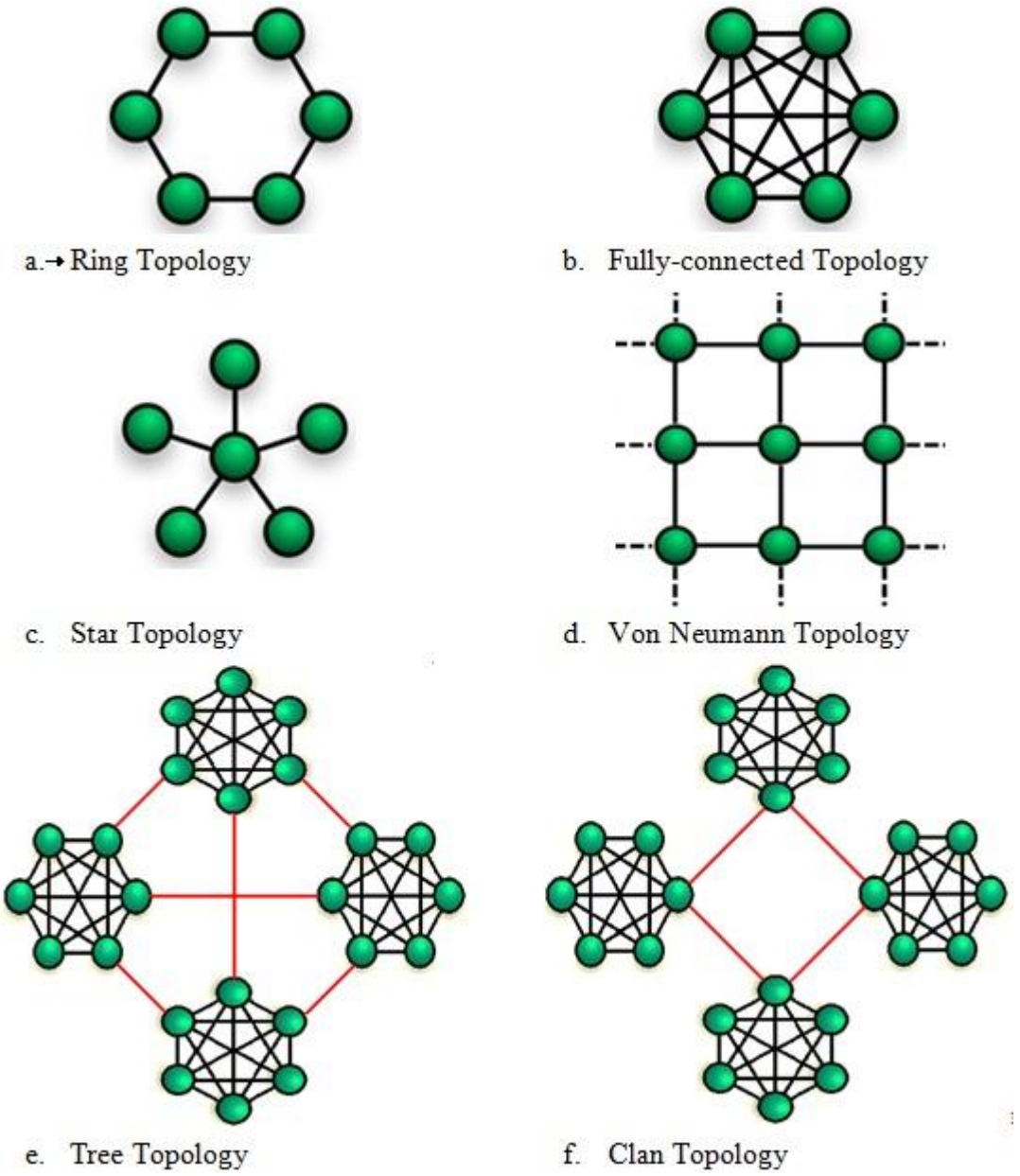


Figure 2.8: Classical Social Topologies

2.6 Particle Swarm Optimization: State-of-the-Art Algorithms

The following are the significant categories of the PSO methodologies proposed to date: Modified PSO approaches; Pareto-based PSO approaches; Clustering-based PSO approaches; Parallel-based PSO approaches; Hybrid-based PSO approaches; and Dynamic Environment PSO approaches. A literature review of PSO is presented based on the above categories.

2.6.1 Modified PSO

The standard PSO still plays a key role in the optimization process as it has strong global search ability and shows a fast convergence speed. The limitation of the original (standard) PSO is that it might produce inefficient (e.g., poor-quality) solutions with low probability of escaping from local optima at the end of the run, [45]. Numerous modified forms of PSO techniques have therefore been proposed since the original work of Kennedy and Eberhart in 1995, [7]. The modifications tackle problems involving many local optima and hard Non-deterministic Polynomial (NP) problems. A representative sample of the modified PSO approaches is presented next.

Zhang et al. [46], proposed a new modified MOPSO algorithm that employed a new selection mechanism in nominating the leaders that directs the swarm towards the Pareto-front set for multi-objective optimization problems. In this approach, the selecting mechanism involved determining the distance between the best pair of particles and the distance between the best pair of swarm-best for each of the objectives. These distances are then used in updating the particles' velocity and position for the next iteration. Many experiments are conducted to validate the proposed MOPSO algorithm using a set of benchmark problems commonly used in the optimization literature. The numerical results showed a comparable performance against NSGA, [47], and MOPSO, [40], algorithms. The outline of the proposed MOPSO algorithm and the procedure of a new selection mechanism are shown in Pseudo-code 2.

Pseudo-code 2: Multi-objective PSO Algorithm

Initialize: population size N, maximum iteration (*Maxiter*), *iter* = 1
Task: find set of non-dominated solutions that covered the entire Pareto-front
Repeat 1 (for each particle)
Randomly generate particle's speed v_i
Randomly generate particle's position x_i
Return
Repeat 1 (for each particle)
Repeat 2 (for each objective function)
Evaluate the fitness of each objective function *Fitness1[i] and Fitness2[i]*

Calculate the best individual solutions $p_{Best1}[\ i\ ]$ *and* $p_{Best2}[\ i\ ]$.
Return 2
Calculate the best global solutions $g_{Best}[1]$ and $g_{Best}[2]$.
Calculate the average of the two best global solutions
Evaluate g_{Best} from $g_{Best}[1]$ and $g_{Best}[2]$.
Evaluate the distance dg_{Best} between $g_{Best}[1]$ and $g_{Best}[2]$.
Calculate the distance $dp_{Best}[i]$ between $p_{Best}[1,i]$ and $p_{Best}[2,i]$
Calculate the best individual solution $p_{Best}[i]$,
if ($dp_{Best}[i] < dg_{Best}$)
Select e $p_{Best}[i]$ randomly between $p_{Best}[1,i]$ and $p_{Best}[2,i]$,
else
Evaluate $p_{Best}[i]$
end if
Update the velocity v_i using $g_{best}\ i$, $p_{best}\ i$

if (position x_i IS NOT in the quasi solution area)
Update the position x_i
end if
$iter = iter + 1$
if ($iter <= Maxiter$)
Return 1
end if
Obtain the Pareto-front set

Pulido et al. [48], presented a new Efficient MOPSO referred to as E-MOPSO. The novelty of the E-MOPSO algorithm is represented by the new mechanism that preserves the distribution of the solutions along all global regions of the search space. This is achieved by:

1. Adopting a new parameter called the turbulence operator to avoid premature convergence.
2. Employing a constraint-handling scheme to limit exploration within the feasible regions only.
3. Deploying a self-adaptation mechanism to automatically fine-tune the values of the design parameters for the MOPSO algorithm.
4. Introducing a new hyper-plane-distribution scheme to scatter the non-dominated solutions over the optimum regions.

Many well-known benchmark functions are used to validate the performance of the proposed algorithm. The numerical results showed competitive and promising approximated Pareto-optimal fronts compared to MOEA algorithms. The authors stated that the EMOPSO required the least number of fitness evaluations in solving multi-objective optimization problems.

Nebro et al. [49], presented a new MOPSO algorithm called Speed-constrained MOPSO (SMOPSO) that incorporated a new control scheme to narrow down the flying zone of the particles in the search space. The aim of the SMOPSO is to encourage the particles in the swarm to rely on their own-experiences (local best) in exploring the search space than the leaders-experiences (global best). This is important at the end of the searching process when the swarm comes close to the global regions. Furthermore, the proposed approach also employed the concept of polynomial mutation to enrich the swarm and used an external archive to store the leaders found during the search. Experiments on the SMOPSO are conducted against well-known multiple optimization test problems, and the results are comparable in terms of both convergence and quality.

Yang et al. [50], presented an improved form of PSO algorithm, namely an Improve Particle Swarm Optimization (IPSO), in order to solve problems with many local optima. In this work, a chaotic mutation mechanism is incorporated to PSO in order to dynamically adjust the step size of mutation. IPSO is validated using six well-known multi-criteria test problems, and the experimental results outperformed those obtained by the standard PSO, GA and Chaotic PSO algorithms.

Reyes-Sierra et al. [51], proposed and empirically compared the incorporation of three different adaptation schemes for MOPSO in order to obtain the highest accurate non-dominated solutions for multi-objective optimization problems. In this study, three main parameters are considered during adaptation: inertia weight factor (w), cognitive factor (c_1) and social factor (c_2). The aims of this work are first to investigate the impact of the three aforementioned parameters on the quality of the PSO search in terms of exploration and exploitation capabilities. Second, assigning the parameter that influenced the most on the MOPSO behavior, and third, finding the best finite set of values for these parameters that produced the highest quality approximated Pareto-front solutions. The three adaptive strategies adopted in this work are $\in$-Greedy, Soft Max, and Proportional strategies. Several set of numerical experiments are conducted to validate the proposed three strategies using a set of well-known benchmark test functions. The computational results showed that at least one of the proposed strategies is able to produce a high-quality set of solutions with respect to the other algorithms such as standard MOPSO, [52]. Finally, the authors concluded that achieving an on-line adaption mechanism to improve the quality of the produced solutions is affordable by considering the three proposed strategies.

Bartz-Beielstein et al. [53], proposed a new modified PSO algorithm that incorporated the Deletion and Selection functions to the MOPSO algorithm, called as D-PSO, to solve the multi-objective optimization problems. Besides the assessment of the fitness function, there are two more assessment processes applied in this approach; the first is evaluating the selection fitness function, by which a measure of influence of each particle on the diversity of the Pareto-front is determined. The second assessment process is evaluating the deletion fitness function, in which all non-dominated solutions in an external archive are ranked based on their deletion

fitness values. Accordingly, a swarm best particle is selected at each iteration based on its selection fitness value (i.e. the lower the fitness value, the closer the particle is to the Pareto-front). The members of the archive are chosen to be deleted based on their deletion fitness values (i.e., the lower the fitness values, the higher the chance for deletion). D-PSO algorithm had comparable solutions to other PSO and MOPSO algorithms. Higher computational complexity is a major drawback of the D-PSO algorithm.

Even more recently, many methods of this category (Modified PSO) are introduced, including Cagnina et al. [54], proposed a new Constrained PSO algorithm (CPSO) that incorporated a new constraints handling scheme to solve constrained optimization problems; Alvarez-Benitez et al. [55], proposed an extended version of the MOPSO (EMOPSO) based on a new dominance scheme for selecting leaders; Huang et al. [56], presented a new Comprehensive Learning PSO (CL-PSO) based on the Pareto-dominance to solve MOOPs; Reyes-Sierra et al. [57], proposed a new modified MOPSO referred to as oMOPSO that employed the concept of fitness-inheritance to minimize the total number of function evaluations; Gong et al. [58], presented a modified PSO algorithm based on the Minimal-particle Angle (MAPSO); Reyes-Sierra et al. [59], proposed a novel modified MOPSO algorithm that combined new Fitness Inheritance and approximation themes to the MOPSO, referred to as FI-MOPSO, to minimize the computational time; Hiao-Hua et al. [60], proposed a new Intelligent Particle Swarm Optimization (IPSO) based on underlying notations of agent-Environment-Rule model (AER); Zhen-Su et al. [61], presented a new Adaptive Mutation PSO (AMPSO) that employed an adaptive mutation to the PSO; Chatterjee et al. [62], proposed a new form of modified PSO that incorporated a Dynamic Inertia Weight scheme to the standard PSO (PSO-DIW); Liang et al. [63], presented a new Comprehensive Learning PSO algorithm (CL-PSO) to enhance the particles self-experiences; J. Kenned [64], developed a new Gaussian PSO (GPSO); Monson et al. [65], presented a new artificial PSO algorithm that combined the Kalman Filter to the standard PSO (KF-PSO); Mahfouf et al. [66], presented a new Adaptive Weighted PSO (AWPSO); V.D. Bergh et al. [67], proposed the Guaranteed Convergence PSO (GCPSO) that used a new form of dynamic equation to update the particle velocity; Ratnaweera et al. [68], proposed two new Time-Varying Acceleration-Coefficients models with the (MPSO-TVAC); Kennedy et al. [69], developed a new Discrete PSO (DPSO) that incorporated a binary scheme to code the particle velocity; Al-Kazemi et al. [70], proposed a new Multi-phase Discrete PSO (M-DiPSO) based on the hill climbing strategy; and Monson et al. [71], presented a new modified PSO algorithm that combined the tribes topology to the adaptive PSO (TRIBES-PSO).

2.6.2 Pareto-based PSO

An important consideration in MOPSO is the mechanism of selecting the leaders during the search process. The leaders need to be non-dominated and a member of the Pareto-front. Pareto dominance is a scheme that has been employed in several MOPSOs as follows.

Reyes et al. proposed [72], a novel MOPSO approach using crowding, mutation, and $\in$-dominance (Pareto-dominance). In this approach, the binary tournament scheme associated with an extra mechanism (known as the density estimator) is employed to select the leaders of the swarm. Furthermore, two external archives are suggested in this algorithm; one is used to store the leaders that led the flights at each iteration, while the other is used to store only the particles that outperform their p_{best} value and did not enter the leaders set (i.e., failed to store in the first external archive). Additionally, a clustering scheme is introduced to divide the swarm into a predefined number of sub-swarms (in their work, three sub-warms that are different in size are utilized) in which a different mutation probability operator is applied to each. Pseudo-code 3 illustrates the procedure of this approach. The proposed MOPSO algorithm is compared to well-known EA algorithms (such as NSGA-II [73] and SPEAII [74]) and other modified MOPSO algorithms. Simulation results showed that the performance of the novel MOPSO is highly competitive and is able to find approximate high-quality Pareto-fronts in problems for which other EAs or MOPSO algorithms had failed.

Pseudo-code 3: Modified MOPSO

Initialize: swarm size (*N*), maximum iteration (*Maxiter*), *iter* = 0,
Task: find Pareto-front set
for i = 1: *N*
Randomly initialize the velocity within the feasible regions
Randomly initialize the position within the feasible regions
end for
Initialize the leaders for the swarm
Send leaders to $\in$- archive
Calculate the crowding distance of leaders
Repeat
for i = 1:*N*
Select leaders
Perform flight towards its neighbor (closest) leader
Mutation
Assess fitness function
Update best particle position
end for

Update leaders
Send leaders to $\in$- archive
Calculate the crowding distance of leaders
iter = *iter* +1
if (iter <= Maxiter) or (stopping criterion is NOT satisfied)
Return
end if
Store the final solutions in $\in$-archive
Obtain the Pareto-front set

Coello et al. [52], proposed a new MOPSO algorithm that incorporated the concept of Pareto-dominance to enable the algorithm to solve multi-modal problems. In this approach, an external archive repository is deployed. Furthermore, a new mutation operator is introduced to enhance exploration. Many experiments are conducted to validate the proposed MOPSO algorithm using a set of benchmark functions derived from the optimization literature. Simulation results showed that the modified MOPSO approach is competitive compared to other MOEA algorithms (namely, NSGA [73], micro-GA [75], and PAES [76]). The outline of the proposed MOPSO algorithm is provided in Pseudo-code 4.

Pseudo-code 4: Modified MOPSO Algorithm

Initialize: Initialize: swarm size (*N*), maximum iterations (*Maxiter*), *iter* = 0,
Task: find the Pareto-optimal front set
for i = 1: *N*
DO
Assign zero velocity
Initialize randomly the particle position within the feasible region
end for i
Create an empty external archive for non-dominated solutions (*A*)
Repeat
for i = 1: *N*
DO
Evaluate the fitness function
if (the particle *i* is a non-dominated solution)
Store the particle *i* in the external archive
end if
Generate hyper-cubes of the search space explored so far, and locate the particles using these hyper-cubes as a coordinate system where each particle's coordinates are defined according to the values of its objective functions.

Store the particles best position found so far
Update the particle velocity
Update the particle position
Apply the problem constraints
Assess the fitness function
end for i
Update the external archive
Update the particles best positions
iter = *iter* + 1
if (*iter* <= *Maxiter*) or (stopping criterion is NOT satisfied)
Return
end if
Obtain the Pareto-front solutions

Cabrera et al. [77], presented a new Multi-objective Particle Swarm Optimization (MOPSO) technique, called micro-MOPSO, and it is differentiated by the small number of particles in the swarm (less than 10). Accordingly, this algorithm has required a very low number of function evaluations (3000 per run) in order to produce reasonably good approximations of the Pareto-front. The authors employed different approaches along the search process, such as selecting the leader and choosing the neighborhood for integrating the swarm. The leader selection technique is developed based on the Pareto-dominance concept. To preserve a well-distributed Pareto optimal set, the proposed approach performed a re-initialization process and used two external archives; the first archive, called auxiliary, is used for storing the solutions that the algorithm found along the search process, whilst the second archive, called final, is used for storing the final solutions. Additionally, in order to improve exploration, the proposed algorithm incorporated a mutation operator. The proposed approach tested on benchmark functions and the results outperformed those obtained by the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [73].

Fieldsend et al. [78], presented a new PSO algorithm that incorporated a novel Pareto-dominance called unconstrained elite external archive. Furthermore, a new data structure, referred to as dominate-tree, is introduced. The new selecting scheme started by locating the common point of the dominate-tree according to its dominance relations, then, the leader is selected based on its distance to that common point (i.e. leader is the particle whose distance is the smallest). The authors also suggested an original turbulence factor to improve the trajectory adjustment of each particle in the search space.

X. Li [79], proposed a new Non-dominated Sorting PSO algorithm (NSPSO) that combined the concept of dominated sorting, the concept of Pareto-dominance, and the external archive mechanism to the standard PSO. All particles in the swarm deposit their new position in a

pool of size double that of the swarm. In this pool, the previous best particles solutions and the current ones are stored in the archive. Only best solutions found at each iteration are kept in the swarm using the non-dominated sorting mechanism. Furthermore, the leaders for the swarm are randomly selected based on two mechanisms that are niche count and nearest neighbor density estimator.

Other PSO algorithms incorporating the concept of Pareto-dominance include: Moore et al. [80], proposed a new PSO algorithm, called MPSO, that adopted the concept of Pareto-dominance and three different dynamic models to update the particle's velocity during their search; Reyes-Sierra et al. [81], presented a new MOPSO that employed the concepts of Pareto-dominance and $\in$-dominance, the crowding distance technique, and the clustering scheme; Baumgartner et al. [82], proposed a new MOPSO algorithm that employed the concept of Pareto-dominance and the weighted sums scheme; and Coello et al. [29], proposed a new MOPSO algorithm that combined to the roulette-wheel selecting, external achieve, adaptive mutation techniques to solve a variety of multi-objective optimization problems.

2.6.3 Clustering-based PSO

The PSO computational complexity is mainly affected by communication between the particles of the swarm. In the PSO hypothesis, the information exchanged or broadcasted between the particles on the searching space can help to guide the particles towards the possible best solutions (i.e., global optimums if exist).

In a case when a particle finds out a possible good region in the search space, each network topology shows different ways to exchange the information among other particles in the swarm, in other words, the topology regulates how this information will be delivered out among its neighbor particles.

According to the connectivity-schema (topology) among the particles, the PSO performance is determined in which the higher the level of connectivity, the faster the convergence of the swarm towards regions nearby with possible good solutions. Nevertheless, this fast convergence may also lead the swarm to be trapped in local minima. On the contrary, slower convergence can allow the swarm to carefully explore the search space that may leap the swarm away from local minima. From the above connectivity-level abbreviation, it is vital to choose carefully the type of communication topology among particles in order to make a good balance between exploration and exploitation abilities out of the searching algorithm. A selection of clustering-based PSO approaches is briefly surveyed here.

Toscano-Pulido et al. [83], proposed an extension version of the MOPSO algorithm, referred to as AMOPSO, that employed the concept of Pareto-dominance to govern the flight direction of the swarm during the search process, and also adopted a clustering scheme where the entire swarm is divided into sub-swarms to improve the exploiting ability of the algorithm. In this

approach, each sub-swarm is allowed to execute the MOPSO algorithm using its own particles. Furthermore, after each iteration, the sub-swarms shared their best information amongst themselves. The AMOPSO algorithm is verified using benchmark problems, and the results showed very good performance with respect to the other MOEAs such as the NGSAII, [73], and the PSEAII, [74], algorithms. Finally, according to the vast improvement in the performance of the proposed AMOPSO algorithm, the future work of the authors is to further improve their algorithm by introducing new schemes to it such as self-adapted and efficient handling constraints schemes which enhance the quality of the obtained non-dominated solutions. The procedure of the proposed approach is provided in Pseudo-code 5.

Pseudo-code 5: AMOPSO Algorithm

Initialize: swarm size (*N*), *iter*= 1, maximum iterations (*Maxiter*), No. of sub-swarms (*M*)
Task: finds the multi global optimum points
for each sub-swarm
DO
Initialize its particles (i.e. initialize both particles' velocity and position)
Initialize g_{leader} set (i.e., the set of global leaders)
end for
Repeat
for each sub-swarm
DO
for each particle
DO
Select a leader
Perform the flight (i.e. update the particle's velocity and position)
Update values (i.e. assess the fitness function)
if (it is a leader)
add to the g_{leader} set
end if
end for
iter = *iter* + 1
While maximum number of iterations is not reached
Store leaders in g_{leader} set in *n*swarms groups
end for
Assign each leader group to a sub-swarm
if (*iter* <= *Maxiter*)
Return

end

Obtain the Pareto-optimal front solutions

Passaro et al. [84], proposed a new PSO algorithm known as *k*-PSO that incorporated some features such as the clustering scheme, parallel mechanism, and very promising neighborhood social network. The goal behind introducing the clustering scheme and neighborhood topology is improving the adjustment of a particle's flight direction in each sub-swarm. In this approach, a modified version of *k*-means clustering algorithm is adopted in which the notion of Bayesian information is deployed first to estimate the number of the cluster and then the standard *k*-means clustering executed afterward. Further, a promising Von Neumann topology is used (i.e. as the neighborhood topology) to properly enhance the sharing of information between particles via a powerful communication structure. The *k*-PSO algorithm is validated using many benchmark test problems and the results showed that performance is competitive and outperformed other PSO algorithms such as SPSO, [85], and ANPSO, [86].

Das et al. [87], presented a modified version of PSO that incorporated the Multi-Elitist scheme (ME) to the canonical PSO algorithm, known as MEPSO. The aim of this work is to deploy the MEPSO algorithm to optimize complex clustering and linearly non-separable datasets applications with no prior knowledge of the number of naturally occurring groups in the data. In this proposed algorithm, the Conventional Sum-of-Squares Distance scheme (CSSD) is replaced by a more efficient Kernel-Induced Similarity Measure scheme (KISM) allowing the proposed algorithm to classify data that is linearly non-separable in the original input space. Furthermore, a new selecting scheme is employed for the optimal number of clusters. Experiments are conducted to validate the MEPSO using standard low and high dimensional data sets. The computational results indicated that the performance of the proposed MEPSO is successful in obtaining accurate final classification and the mean number of classes over a test suit of several artificial and real life datasets.

Omarn et al. [88], proposed a new clustering PSO algorithm known as Dynamic Clustering PSO (DC-PSO) in which the Dynamic Clustering scheme is linked to the standard PSO in order to optimize an unsupervised image classification application. In this approach, no prior knowledge is required to specify the number of clusters in advance; rather the algorithm itself automatically estimated the optimal number of clusters. Further, the set of data is divided into a large number of clusters to reduce the impact of the initial conditions on the clustering process. Then, a binary PSO with the optimal number of clusters is applied. Moreover, the centers of these clusters are precisely located by employing the K-means clustering algorithm. The DCPSO algorithm is validated by testing it against variant set of natural images (i.e. MRI and satellite images), and then the simulation results are compared to those given by other unsupervised clustering algorithms. The authors concluded that DCPSO algorithm outperformed other approaches in terms of number of clusters, convergence rate and quality of final solutions.

Mostaghim et al. [89], presented a new version of the MOPSO algorithm, namely covering-MOPSO, with the aim of producing non-dominated solutions for multi-objective optimization problems that covered all Pareto-front regions. The proposed covering-MOPSO is applied in two stages; the goal of the first stage is obtaining an approximated high-quality Pareto-front set. In the second stage, several sub-swarms are introduced to individually perform explorations in the search space to cover the gaps between the non-dominated solutions produced by the first stage. The method is first validated using a set of benchmark test functions. It is also applied to a real-world problem to optimize an antenna design. The simulation results showed that the covering-MOPSO is able to produce non-dominated solutions that covered all Pareto-optimal fronts and then fulfilled one of the most important issues in optimizing multiple objectives that is preserving the diversity. The alternative antenna designs are compared to those given by Hybrid MOEA, [90], to find that covering-MOPSO outperformed in terms of less computational cost and fast convergence rate.

Sheng-Ta et al. [91], proposed a new version of Cluster-based Solution Exploration Strategy MOPSO algorithm referred to as CSES-MOPSO that incorporated the concept of the cluster algorithm to assign swarm leaders. In addition, the concept of external archive is adopted to store the non-dominated solutions produced by several sub-swarms. Further, a mutation operation is introduced to the CSES-MOPSO algorithm to improve the particles' exploitation capability in finding points very close to the global optimum. Several experiments are conducted to validate the CSES-MOPSO algorithm using well-known benchmark functions such as ZDT1, ZDT2, and ZDT3 developed by Zitzler et al. [92]. Different Metric schemes are used to compare the performance of the proposed algorithm with other MOEAs such as NSGAII, [73], and SPEA2, [74]. Accordingly, this comparison led to the conclusion that CSES-MOPSO algorithm is able to find higher quality and better-spread non-dominated solutions.

Cui et al. [93], proposed a new hybrid PSO-K-means algorithm that combined the standard PSO algorithm with the K-means clustering algorithm to avoid premature convergence. The aim behind the PSO-K-means approach is combining the global search ability of the PSO with the fast convergence rate of the K-means algorithm. Two phases are considered in this approach; the first phase used the PSO to obtain a set of non-dominated solutions, and the second phase refined these solutions to produce a Pareto-front set (i.e., final set of solutions) using a K-means clustering technique. The simulation results over a set of different text document datasets indicated that the proposed PSO-K-means algorithm is able to find good results with a reasonable computational complexity.

Passaro et al. [94], presented a new clustered version of PSO algorithm referred to as k-PSO that combined the K-means clustering scheme to the standard PSO algorithm in order to enhance the exploring capability of the algorithm in finding multiple global optima in problems with multi-objective domains. In this approach, the swarm is split into sub-swarms in which each cluster group performed its local search. The proposed approach is tested on benchmark problems

and the experimental results showed that the k-PSO performance is satisfactory compared to other clustered PSO algorithms.

2.6.4 Parallel-based PSO

Heuristic methods are parallel processes in nature. As multi-core processors become more readily available, they can play a vital role in offering valuable computing resources to the optimization community. The assessment process for each individual (i.e. candidate solution) in any purely iterative searching algorithm can often be done independently from competing solutions. Therefore, parallel mechanism in handling the evaluation process would be highly advantageous. In effect, the computational running time required for an application would be inversely proportional to the number of processors.

In spite of the fact that PSO is a population-based optimization technique that naturally lends itself to a parallel implementation, incorporating the parallel mechanism into a PSO algorithm provides two main advantages: increasing the convergence rate and enhancing the robustness of the algorithm.

Schutte et al. [95], presented a comprehensive study that explored the performance of the PSO search engine when hybridized with parallelism. The aim of this study is to investigate the effect of employing a parallel mechanism on the computational complexity and the global exploration ability of PSO algorithm. In this work, different Parallel PSO performances are analyzed using two categories of test problems with multiple local minima and with large scale analytical problems (either with high dimensions or large decision variables). In addition to that, the behaviors of Parallel PSO are investigated with respect to three different types of fitness functions that are cheap functions (i.e. such as simple linear functions), medium-scale functions (i.e. such as biomechanical system identification problems), and lastly with high computational cost (i.e. such as load-balanced design problems). The computational results confirmed the merits of the integration of parallelism into the PSO algorithm in all cases (e.g., in the load-balanced problems, the computational time is reduced by 95%). A parallel PSO algorithm is illustrated in Figure 2.9.

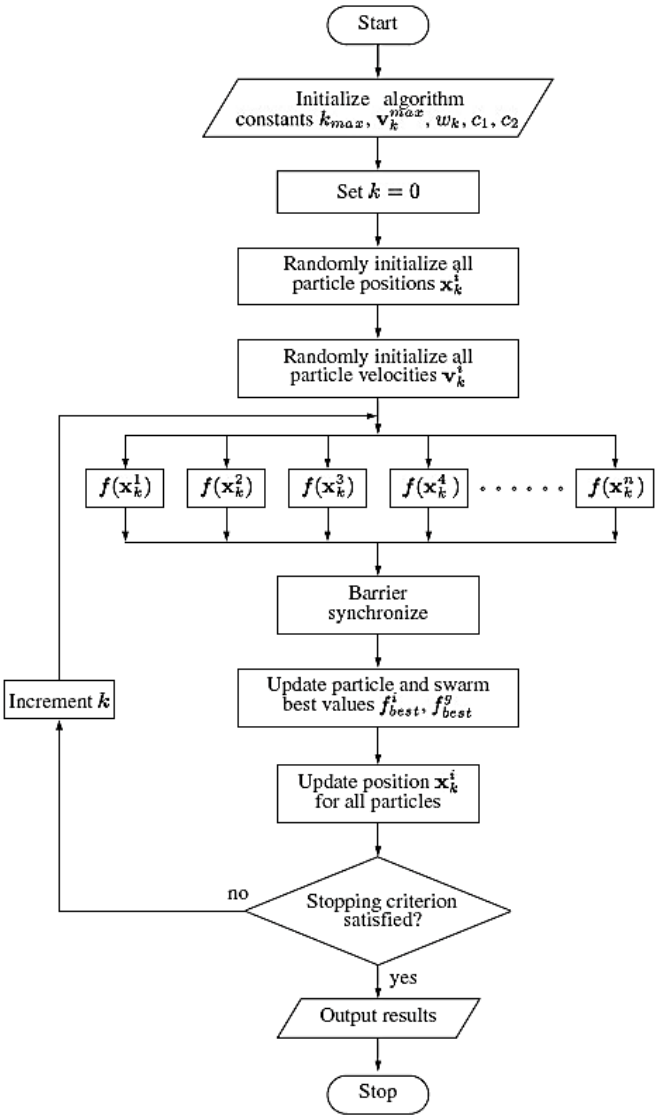


Figure 2.9: Parallel MOPSO Flowchart [95]

Brits et al. [96], presented a new modified PSO algorithm that employed the niching[1] scheme and the parallelism technique to improve the efficiency of the standard MOPSO algorithm, referred to as NPSO. In this approach, the concept of sub-swarms is used to obtain different regional optima. The authors claimed that their work is the first hybrid PSO algorithm that introduced the parallelism technique to the Niching PSO (NPSO). Many experiments are conducted using benchmark functions, and the results showed that the performance of the NPSO

[1] Niching is a term that refers to a technique of finding and preserving multiple stable *niches* (i.e., favorable in global regions of the search space) to prevent convergence to a local optimum [298].

algorithm is promising in terms of having a low computational cost and maintaining diversity along the Pareto-front domain. The NPSO algorithm is outlined in Pseudo-code 6.

Pseudo-code 6: NPSO Algorithm

Initialize: swarm size (*N*), maximum iterations (*Maxiter*), *iter* = 1, no. of sub-swarms (*M)*
Tasks: find the Pareto-optimal front set
for i = 1: *N*
DO
Initialize randomly the particle velocity within the feasible region
Initialize randomly the particle position within the feasible region
Assess the fitness function
end for i
Repeat
for i = 1: *N*
DO
Train main swarm particles using one iteration of the cognition *only model.*
Update fitness of each main swarm particle.
for j= 1: *M*
DO
Train sub-swarm particles using one iteration
Update each particle's fitness.
Update swarm radius
If possible, merge sub-swarms
Allow sub-swarms to absorb any particles from the main swarm that moved into it.
end for j
Search main swarm for any particle that meets the partitioning criteria – *if* any is found with this particle and its closest neighbor. create a new sub-swarm
end for i
iter = *iter* + 1
if (*iter* <= *Maxiter*) or (stopping criterion is NOT satisfied)
Return
end if
Obtain the Pareto-front solutions

Shu-Kai et al. [97], combined the Multi-objective Evolutionary Algorithms (MOEA), the Pareto-dominance concept, and an advanced parallel computing scheme to the standard PSO algorithm, all pieced together into a new hybrid PSO algorithm referred to as PPS-MOEA. The

aim of PPS-MOEA approach is to tackle real-world (i.e. high-dimensional) multi-objective optimization problems in an effective and efficient manner. In this approach, the clustering scheme is introduced to divide the swarm into several sub-swarms, and then the parallel computing strategy is applied by adopting the master-slave architecture to improve the social collaboration of the particles among the sub-swarms. Further, a novel selecting scheme (called "picking rule") is developed to preserve the size of the external repository. The proposed PPS-MOEA is verified against a set of well-known functions and the results are compared to other algorithms. Experimental results showed that the performance of the PPS-MOEA is robust, stable, competitive, and reliable in finding high-quality Pareto-front sets that are even very close to the Pareto-optimal fronts.

Parsopoulos et al. [98], presented a new version of the Parallel MOPSO algorithm that incorporated the notion of Vector Evaluated Genetic Algorithm (VEGA), developed by D. Schaffer [99], into the MOPSO yielding a new Parallel Vector Evaluated MOPSO algorithm (Parallel VEPSO). This work is considered as an extension to the VEPSO that is previously developed by two of the current authors, [100], with the aim of getting the advantages of the parallel implementation on the VEPSO search ability in terms of the computational cost and exploration. The new Parallel VEPSO is tested using four well-known benchmark functions and the numerical results showed that the Parallel VEPSO is competitive and superior in all cases compared to VEGA and VEPSO (series version).

Gies et al. [101], investigated the performance of the Parallel PSO with respect to the number sub-swarms. Their experimental results indicated that a set of ten sub-swarms is giving the fastest performance; i.e. comparatively eight time faster than the serial implementation. It is important to mention that specifying the number of independent paths or sub-swarms could vary based on the nature of the problem, the communication network, and the complexity of the fitness function. Therefore, using a multi-core CPU processor and a simple communication topology could improve the PSO efficiency.

Mostaghim et al. [102], proposed two new Parallel versions of a MOPSO algorithm for which both had different clustering schemes that divided the entire swarm into several sub-swarms and then each one independently performed its own MOPSO algorithm on different processors (CPUs). The aim of the work is extending the implementations of MOPSO algorithms to successfully cope with hard real-world problems that required several processors. Cluster-based sub-swarm MOPSO, known as C-MOPSO, is the first proposed algorithm that is designed to work on a fixed number of processors. In this approach, the main processor is responsible for:

1. Performing the clustering, in which several leaders are picked from the external repository,
2. Sending the leaders out to other processors,
3. Receiving new information from the leaders,
4. Updating the members in the external repository, and
5. Repeat the cycle.

The Hyper-volume-based sub-swarm MOPSO (H-MOPSO) is the second proposed algorithm that is designed to work on undetermined number of processors. In the H-MOPSO, the leaders are individually selected based on their domain in the hyper-volume (i.e. according to the size of the area dominated by the particle). H-MOPSO is compared to other methods using benchmark problems, and the results showed that the H-MOPSO algorithm outperformed the C-MOPSO algorithm. The authors concluded that the Hyper-volume-based sub-swarm is more efficient in choosing high-quality leaders than those given by the Clustering-based sub-swarms.

Nanbo et al. [103], suggested a novel evolutionary optimization technique for multi-band and wide-band patch of antenna design. In their work, the computational time is significantly reduced by implementing the optimization process on parallel clusters. Their new parallel PSO is applied to the design of rectangular patch antennas. Optimization results obtained are able to achieve optimal design. Both accuracy and robustness analysis on their parallel PSO are conducted. The authors concluded that parallelism in PSO is promising, but requires further examination of the communication protocol between processors and their synchronization.

Chang et al. [104], presented a new parallel PSO algorithm wherein a sub-swarm division technique is incorporated. In their algorithm, the evaluation of the fitness function is independently performed for each particle. The results showed that the efficiency of the algorithm is highly dependent on the network topology for the communication of the best position found by a swarm. Their conclusion is verified by investigating several types of topologies such as star, ring, and fully connected shapes.

2.6.5 Hybrid-based PSO

In general, a Hybrid system is the combination of two or more different schemes, aimed at achieving a particular objective or goal. Within the optimization context, Hybridization is the process of merging two or more optimization techniques into a single algorithm, usually combining problem knowledge with the algorithm concept. A review of the most common Hybrid PSO algorithms is presented as follows.

Parsopoulos et al. [100], presented a novel hybrid PSO algorithm referred to as VE-PSO that integrated the concept of a Vector Evaluation algorithm (VE) (i.e. derived from the notions of Vector Evaluation Genetic Algorithm (VEGA) developed by D. Schaffer [99]) to the standard PSO in order to simultaneously optimize multi-objective problems. The aim of this work is exploring more than one region in constructing the Pareto-front set. To achieve this, an aggregation scheme with fixed or adaptive weights is adopted. The first VE-PSO search engine used two swarms in which each is evaluated based on one objective (two objective functions are considered in this work). At the end of each iteration, both swarms shared their best information in a way that the best particle in the first swarm is used to guide the second swarm's particles. The effectiveness of the proposed VE-PSO algorithm is validated using non-trivial benchmark

test functions, and the computational results showed that the VE-PSO is able to find robust and accurate Pareto-front points, and come close to the Pareto-optimal front compared to the other MOEAs such as VEGA.

Esquiavel at al. [105], proposed two new hybrid MOPSO algorithms referred to as the global-best MOPSO (g-PSO) and the local-best MOPSO (l-PSO) algorithms. In these approaches, a uniform mutation operator and the concept of the neighborhood best leaders are combined with the standard constrained MOPSO algorithm in order to enhance the diversity of the non-dominated solutions and then avoid premature convergence. Six different experiments are conducted to test the effectiveness of the proposed algorithms using well-known benchmark problems, and the simulation results showed that the l-best PSO is superior and outperformed other MOEA algorithms. Further, it is found that the l-best algorithm performed better than the g-best algorithm. Finally, the authors concluded that incorporating the uniform and non-uniform mutation operators resulted in significant improvement in performance. The outlines of the two proposed algorithms are provided in Pseudo-codes 7 and 8.

Pseudo-code 7: Synchronous g-PSO Algorithm

Initialize: swarm size *N*, maximum iterations *Maxiter*, *iter* = 1, number of dimensions *D*
Task: find Pareto-optimal front set
for i = 1: *N*
DO
for j = 1: *D*
Initialize randomly particle position within feasible regions
Set particle velocity to zero
Copy the particle current position in particle best vector
Assess the fitness function
end for j
end for i
Search the swarm best and store them in the external archive
Swarm flight in the search space
Repeat
for i = 1: *N*
DO
for j = 1: *D*
Update the particle velocity (using particle vest vector and current particle position)
Apply velocity constraints
Update the particle position
Apply Mutation operation
end for j

```
    Assess the fitness function
    Update the particle best vector
end for i
iter = iter + 1
if (iter <= Maxiter) or (stopping creation IS NOT satisfied)
   Return
end if
Obtain the Pareto-front set
```

Pseudo-code 8: Asynchronous l-PSO Algorithm

```
Initialize: swarm size N, maximum iterations Maxiter, iter = 1, number of dimensions D
Task: find Pareto-optimal front set
for i = 1: N
    DO
    for j = 1: D
        Initialize randomly particle position within feasible regions
        Set particle velocity to zero
        Copy the particle current position in particle best vector
        Assess the fitness function
    end for j
end for i
Swarm flight in the search space
Repeat
for i = 1: N
    DO
    Search better in the k-neighborhood for particle x_i and record it in neighbors best vector l
    for j = 1: D
        Update the particle velocity (using particle vest vector and neighbors best)
        Apply velocity constraints
        Update the particle position
        Apply Mutation operation
    end for j
    Assess the fitness function
    Update the particle best vector
end for i
iter = iter + 1
if (iter <= Maxiter) or (stopping creation IS NOT satisfied)
    Return
```

end if

Obtain the Pareto-front set

Janson et al. [106], proposed a new hybrid MOPSO algorithm, referred to as Clust-MPSO, that combined a clustering mechanism with the MOPSO algorithm that is developed by Coello et al. [52]. With the clustering mechanism, the entire swarm is divided into several sub-swarms. This allowed each sub-swarm to direct its particles towards a different region of the search space. Accordingly, for each sub-swarm, different strategies for updating the personal best position, for selection of the neighborhood best, and for updating the total non-dominated particles are adopted. The proposed Clust-MPSO algorithm is applied to a well-suited biochemistry problem (the molecule docking problem). The task is predicting the three dimensional structure of a binding of a target receptor and a ligand. The numerical results showed that the Clust-MPSO algorithm is able to successfully outperform a well-known MOEA algorithm that is Lamarckian-GA, developed by Morries et al. [107].

Ho et al. [108], presented a novel Intelligent MOPSO (IMOPSO) with the aim of solving multi-objective optimization problems. In this approach, two main schemes are adopted in order to obtain an efficient and robust performance. These two schemes are a Generalized Pareto-based Scale-Independent Fitness function (GPSISF) and the Intelligent Move Mechanism (IMM). In GPSISF, each particle in the swarm is given a score on an assessment process. The Intelligent Move Mechanism (IMM) used the traditional flying method of PSO with the aim of improving the particles exploring ability during the search process. In IMM, a systematic reasoning method is employed to improve the quality of the particles over each iteration. This is achieved by sampling, analyzing, and then grouping the particles based on the swarm best (g_{best}). An evaluation of the IMOPSO performance is carried out using well-known test functions. Accordingly, the authors stated that the IMOPSO algorithm is competitive and is recommended for tackling real-world optimization applications.

Ching-Shih et al. [109], proposed a new hybrid MOPSO algorithm that combined the Local search and Clustering schemes to the MOPSO and is referred to as MOPSO-LC. The rationales behind the proposed algorithm are: 1) obtaining the advantages of employing the local search scheme by fine-tuning the particles search ability which then prevented the swarm from getting trapped in local minima; 2) gaining the benefits of incorporating the clustering scheme to enhance the convergence rate (i.e. by eliminating the extra non-dominated solutions from the external repository), hence, preserving the diversity along the Pareto-front set. The proposed MOPSO-LC algorithm is examined using a test function, and the simulation results are compared to the Strength Pareto Evolutionary Algorithm (SPEA) which is state-of-the-art in MOEAs [110,74]. The authors confirmed that the MOPSO-LC algorithm outperformed the SPEA in that test, and suggested more experiments on hard constrained multiple objective problems.

Salazar-Lechuga et al. [111], proposed a new hybrid MOPSO algorithm referred to as MOPSO-*fs* that combined the standard MOPSO algorithm with the Fitness-Sharing mechanism

(*fs*) for solving multi-model multi-objective optimization problems. In this approach, two consecutive operations are employed in order to select the global leaders (i.e., from the repository). The first operation is applying the concept of Pareto-dominance to distinguish the non-dominated solutions from other particles in the swarm, and then storing them in the repository. Next, a fitness-sharing mechanism is deployed to spread the solutions along the Pareto-front. In this mechanism, the fitness value is calculated for each of the solutions in the repository. A high value for fitness sharing means that the particle is not surrounded by other particles, or at least there are particles far away from this one, and vice versa. In the case of full repository, the fitness sharing is used to free some spots, in which the particle with worst fitness sharing is replaced by this new one. Having done this, only competitive and high-quality elements remain in the repository and then PSO is assured to converge with a fast rate and to preserve diversity among the best solutions. Furthermore, the stochastic universal sampling method is utilized to allow the particles in the swarm to choose their leaders. The MOPSO-*fs* is validated against several benchmark problems, and the results are superior to those obtained by MOPSO [52], NSGAII [73], and PAES [112]. Finally, the authors attributed the success of their MOPSO-*fs* algorithm to the combination of the Particle Swarm Optimization (PSO) and Fitness sharing. Figure 2.10 depicts the general structure of the MOPSO-*fs* algorithm.

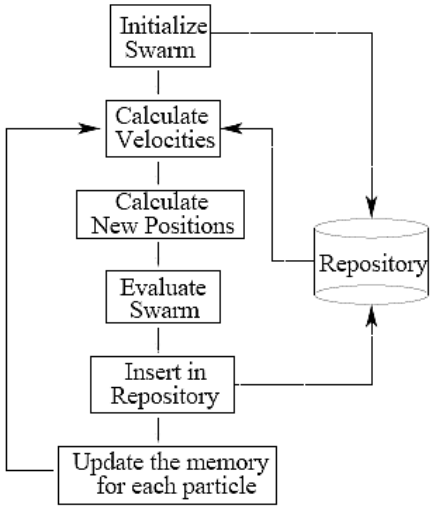


Figure 2.10: Flowchart of the MOPSO-*fs* Algorithm

There exists a vast body of research on Hybrid PSO category in which new and previously developed schemes are combined with the standard PSO or MOPSO, including Wickramasinghe et al. [113], proposed a novel hybrid PSO referred to as MDEPSO that combined the Multi Differential Evolution (MDE) with the standard PSO; Santana-Quintero at al. [114], proposed a novel hybrid MOPSO that incorporated the Constrained (C) and Scatter-Search (SS) schemes to the MOPSO, called C-MOPSO-SS; Ming-Rong et al. [115], proposed a new

form of hybrid PSO with Extremal Optimization (Hybrid PSO-EO); Liu et al. [116], presented a new Fuzzy Adaptive TPSO (FATPSO) technique that combined the turbulent PSO (TPSO) with the fuzzy logic controller; Poli et al. [117], developed a novel hybrid PSO that made use of Genetic Programming (GP) with the standard PSO in an algorithm known as GP-PSO; Jian et al. [118], presented a new hybrid PSO, called PSO-RDL, that combined the Recombination and Dynamic Linkage Discovery (RDL) with the standard PSO; P.J. Angeline [119], proposed a new hybrid PSO that combined the PSO algorithm with the tournament selection technique to replace the worst particles in the swarm (e.g., in minimizing problems, particles with the largest fitness values) with those of better particles (e.g., particles a low fitness value); Z. Geem [120], proposed a new Particle-Swarm Harmony Search (PSHS) that incorporated the concept of particle-swarm to the standard discrete-value harmony search algorithm; Meissner et al. [121], presented a new Optimized PSO algorithm (OPSO) that incorporated the new meta-optimized technique to the standard PSO; Srinivasan et al. [122], proposed a new hybrid Particle Swarm Inspired Evolutionary Algorithm (PS-EA) that combined the notion of EA to the particle swarm theory; Santana-Quintero et al. [123], proposed a new hybrid MOPSO algorithm referred to as PSOMORSA that combined the PSO to the new multi-objective constrained-handling technique, Rough Sets theory, and the $\in$-dominance; Krink et al. [124], developed a new hybrid PSO that coupled the new Lifecycle model with the standard PSO; Zhang et al. [125], presented a new hybrid PSO algorithm, known as DEPSO, that combined the Differential Evolution (DE) with the standard PSO; Robinson et al. [126], presented two new hybrid PSO algorithms; the first algorithm, known as GA-PSO, combined the Genetic Algorithms (GA) with the PSO, and the second algorithm, called PSO-GA, combined the PSO with the GA; Vesterstroom et al. [127], developed a new hybrid PSO algorithm referred to as PSO-DL that combined the standard PSO algorithm with the Divisions of Labor scheme (DL); T. Hendtlass [128], presented two new hybrid PSO algorithms referred to as PSO-ACO and PSO-DE. The first algorithm combined the PSO with the Ant Colony Optimization (ACO), and the second algorithm combined the PSO with the Differential Evolution; Lopes et al. [129], proposed a new hybrid PSO algorithm that incorporated the Fast Local Search (FLS) technique and the GA to the standard PSO; and Habibi et al. [130], proposed a new hybrid PSO algorithm that combined the Ant Colony Systems (ACS) and Simulated Annealing (SA) with the PSO.

2.6.6 Dynamic Environment-based PSO

Many real-world problems are in dynamic environments, meaning that the position, the environment, and the landscape of the global optimum can often change over time. Therefore,

tackling such problems would require the capability of dynamically altering (i.e. adapting) the fitness landscape with the direction of the exploration towards the global optimum [131,132,133,134]. In order words, dynamic fitness landscape means that the functions and the optimum solutions could change over time.

Standard PSO algorithms have not been validated against such problems when the speed of change is fast and the optimum keep moves across the decision space in steps faster than the swarm best position. Therefore, it is very important to develop new forms of PSO that are able to track this goal toward finding good approximate global optimum solutions. A summary of different PSO algorithms that tackled problems with dynamic traits is provided next.

Li et al. [135], developed a new version of Speciation-based PSO, known as S-PSO, in order to enhance the performance of the searching algorithm in dynamic fitness landscapes. The aims behind improving the S-PSO algorithm are first upgrading the S-PSO to handle a series of dynamic environments with varying number of peaks, and second, enhancing the S-PSO to preserve diversity within each species when it is needed. In the new S-PSO algorithm, the concept of Quantum- swarms is employed to enhance the adaptation ability of the algorithm for tracking a moving global optimum (or target). Further, a Particle-diversification scheme (PD) is adopted in this approach to promote the ability of S-PSO in arranging a good relationship between convergence and diversity within each species. The proposed approach is tested against many benchmark problems with different number of peaks, and the results showed good performances in tracking global optimum on most test problems. The outline of the proposed PSO algorithm is provided in Pseudo-code 9.

Pseudo-code 9: Species based PSO Algorithm.

Initialize: swarm size (*N*), maximum iterations (*Maxiter*), *iter* = 0
Task: find the Pareto-front solutions
for i = 1: *N*
DO
Randomly initialize the velocity within the feasible region
Randomly initialize the position within the feasible region
end for
Initialize the swarm best candidate solutions
Initialize the local best solution
Repeat
for i = 1: *N*
DO
Update the velocity
Update the position
Assess the fitness function

if (particle new solution ≼ particle best) or (particle new solution ≺≻ particle best)
best particle = particle new solution
end if
end for i
Update the particle best solution
for i = 1: *N*
DO
if (change is detected)
$\overrightarrow{p_i} \leftarrow \overrightarrow{x_i}$
end if
end for i
Sort all particles in an ascending order according to their fitness values
Call the speciation procedure algorithm to identify species seeds
Assign each identified species seed's $\overrightarrow{p_i}$ as the $\overrightarrow{p_g}$ to all individuals identified in the same species
Update the velocity
Update the position
for i = 1: *numberSeeds*
DO
if ($numParticles_i > P_{max}$)
Replace the excess particles with random particles into the search space
end if
end for i
iter = *iter* + 1
if (*iter* < = *Maxiter*) or (stopping criterion is NOT satisfied)
return
end if
Obtain the approximated Pareto-front solutions

Kiranyaz et al. [136], presented two new PSO algorithms in order to tackle Moving Peaks Problems (MPPs) and then find solutions that are stable, efficient, and robust. The first proposed approach is the Multi-Dimensional (MD) PSO, namely (MD-PSO). In this approach, an Inter-dimensional Navigation scheme (IDN) is incorporated to determine the optimum dimension wherever the non-dominated solutions are found. The importance of the MD-PSO algorithm is that it required no prior information about the global optimum dimension (i.e. regular PSO or even some modified PSO algorithms are often not able to keep track of a moving global optimum). In order to overcome premature convergence, the authors introduced their second approach that is the Fractional Global Best Formation technique (FGBF) with the Multi-swarm

PSO algorithm, called as FGBF-MS-PSO. This approach worked by gathering all the best dimensional components and then creating artificial Global-best solutions (aGBs). Next, the aGBs solutions are compared to each of the swarm's best solution (g_{best}) and the superiors amongst are selected to be the leaders. Finally, the two approaches are tested using dynamic benchmark functions, and the simulation results showed that both approaches are successful in achieving the tracking of the global optimum peak with the least error (using MD-PSO), and finding better solutions (using FGBF-MS-PSO).

Carlisle et al. [137], proposed a new Quantum-behaved PSO algorithm, called as Q-PSO, to solve problems with complex dynamic environments. In this work, the Q-PSO algorithm is applied to three different dynamic environments with different step size changes. The experimental results showed that the Q-PSO algorithm is able to precisely detect the changes that occurred in the environment, and effectively re- arrange the relationship between the global and the local searches. It is found that in cases for which problems had noisy environments with high change frequencies, Q-PSO performed well in terms of being fast, reliable, and accurate tracking.

Pan et al. [138], proposed two new hybrid PSO algorithms which are Swarm-core Evolutionary with PSO (SCEPSO) and PSO with Simulated Annealing (PSOwSA) for dynamic environments. In the SCEPSO algorithm, a clustering scheme is introduced to divide the swarm into three sub-swarms in which each is assigned a different task. Furthermore, the combination of the SCE and the PSO algorithms allowed the searching process to track continuously changing solutions with minimal error. In the PSOwSA algorithm, the concept of simulated annealing is incorporated to the standard PSO algorithm with the aim of enhancing the exploring ability. Both approaches are verified using problems with static and dynamic environments, and the computational results showed that the SCEPSO outperformed the standard PSO and PSOwSA algorithms. Moreover, it is found that the SCEPSO algorithm showed success in following the changing landscape with satisfactory but not high-quality Pareto-front solutions. The results also indicated that the PSOwSA algorithm suffered from premature convergence. Finally, the authors concluded that the SCEPSO performed well only when the dynamic problems had a big step size. They also concluded that the performance of the PSOwSA is acceptable only in problems with static environments.

Parrott et al. [139], proposed a new PSO approach that is used to track dynamic multi-modal peaks. This approach is inspired by the speciation technique (e.g., a technique that divides the swarm into groups of particles that share the same goal). In their work, a new speciation technique (i.e. a modified form of speciation technique developed by Li et al. [140]) and crowding mechanism are used to encourage simultaneous tracking of multiple peaks by preventing overcrowding at peaks. One of the limitations in implementing this technique is that the target velocity should be within velocity bounds of each particle. The results of this approach showed that this technique is less efficient than the original PSO algorithms due to its requirement to perform double fitness evaluation for each particle.

P.N. Suganthan [45], proposed a new modified PSO algorithm that employed a new dynamic network topology. In this approach, the adaptability in the communication topology is implemented. The algorithm started with the ring-topology, and then gradually increased in connectivity until it reached a fully connected topology. Furthermore, it is suggested to use the particle best position to direct the searching process for better exploration. The authors suggested some further improvements to their proposed algorithm as the simulation results did not present noticeable improvements.

Carlisle et al. [141], developed a new PSO algorithm that is able to solve dynamic optimization problems. A new dynamic technique, called Unimodal Environment (UE), is employed to reset (i.e., randomly initialize the velocity of each particle within its upper and lower limits) the record of each particle (i.e., its best position) as the environment changes, and then to avoid making direction and velocity decisions based on outdated flight information. The algorithm is tested using Kennedy's social models [142], and the tracking performance is improved over multi-tracking problems especially the problems when their targets are moving with different velocities.

Peram et al. [143], proposed a new modified PSO algorithm, called as PSO-WED, is incorporated the Weighted Euclidean Distance (WED) to identify the interaction neighbor for a particle. In the proposed WED algorithm, the particle with the highest Fitness-Distance Ratio[2] (FDR) is found for each vector element. Based on the FDR value, the algorithm directed each particle towards its good neighbors and then reduced the probability that the particle's interaction with a poor neighbor. Experiments are conducted to test the proposed algorithm using a set of benchmark functions. The results showed that the performance of the PSO-WED is good compared to the standard PSO algorithm.

Liang et al. [144], proposed a new modified PSO algorithm. In this approach, the entire swarm is randomly divided into sub-swarms and its connections randomly re-distributed after each set of iterations. The modified PSO is tested against benchmark problems, and it is found that when three sub-swarms and a set of five iterations for redistribution are used, good results are obtained.

Eberhart et al. [145], proposed a new PSO algorithm that combined the standard PSO with an adaptive technique to vary randomly the value of the inertia weight factor within a range of [0.5, 1]. The authors concluded that their approach worked satisfactorily on a ten dimensional parabolic test function.

Janson et al. [146], presented a new modified PSO algorithm that combined the standard PSO with the Dynamic Hierarchy scheme (DH), namely PSO-DH. In this approach, the particles are arranged in the DH based on their previous success (i.e. particle's best solution) and then all the particles are assorted accordingly. Particles with better performance are ranked up the

[2] FDR is defined as "the ratio of the difference between the target particle's fitness and the neighbor's fitness to the distance between them in the search space on that dimension" [27].

hierarchy. The proposed algorithm is verified using a test function, and the results showed an improved performance compared to the standard PSO and other modified PSO algorithms. The authors concluded that employing the DH scheme enhanced the exploration ability of the worst particles which in turn improved the performance of the entire swarm.

Yen et al. [147], developed two new hybrid approaches that are Dynamic PSO (DPSO) and Dynamic Particle Swarm Evolutionary Algorithm (DPSEA) to tackle MOOPs. DPSMO is derived from the concept of the Dynamic Multi Objective EA (DMOEA). In the DPSMO algorithm, a new sharing technique is adopted which is used to enhance the information flow amongst particles to improve the flying directions towards the global optimum. It is mentioned that DPSMO had difficulty in producing a high-quality Pareto-front. DPSEA is further combined with EA and PSO's information sharing scheme. Numerical experiments are conducted and the performances of both algorithms are compared. The results showed that DPSEA outperformed DMOEA and DPSMO in terms of diversity of the solutions along the Pareto-front, extending the Pareto-front into new areas, and producing approximated high-quality Pareto-fronts.

2.7 Trends for New PSO Applications

This section highlights the real-world applications of multi-objective optimization that conventional approaches cannot solve. PSO algorithms are becoming increasingly popular in the optimization community as they provide greater flexibility in problem formulation and can handle problems that classical approaches cannot easily tackle. The most promising future research directions of PSO algorithms are presented based on published case studies in Table 2.4.

Table 2.4: Recommended Case Studies for Future PSO Applications

Suggested Case Study	**No. of Objectives**	**Previously Tackled by**
Aerodynamic design	2 – to - 3	MOEA algorithm called as RM-MEDA developed by Zhang et al [148]
Industrial Neural Network design	2 – to - 3	GA based on Multi-objective Neural Net developed by Pettersson et al. [149]
Molecular structures for drugs	2 – to - 3	MOGA II developed by Poloni et al. [150]
Medical decision making	2 – to - 3	Non-dominated Points Multi-objective Linear Programming (NPMOLP) developed by Shao et al. [151]
Supply chain management	2 – to - 3	Multi-location Supply Network Planning (SNP) developed by J. Dickersbach [152]

Interactive airplane design	4 – to - 20	MOGA developed by Bandte et al. [153]
Land use planning	4 – to - 20	MOGA developed by Stewart et al. [154]
Cable-stayed bridge design	20 – to - 100	Multi-objective Genetic Programming (MOGP) developed by Nakayama et al. [155]

2.8 Conclusion

In this chapter, many forms of PSO are discussed, and hence, a summary of the reviewed PSO algorithms is shown in Table 2.5.

Table 2.5: List of Comparable Structure and Design Parameters Used in PSO Algorithms

Algorithm	**Topology**	***w***	***C_1***	***C_2***	**Mutation**	**Swarm best Guide**	**Neighborhood Best Guide**
l-PSO [105]	*Fully*	*C*	*C*	*C*	√	-	√
g-PSO [105]	*Fully*	*C*	*C*	*C*	√	√	-
Pswarm [17]	*Fully*	*L*	*C*	*C*	-	√	-
VEPSO [100]	*Fully*	*L*	*C*	*C*	-	√	-
PSOMORSA [123]	*Fully*	*C*	*C*	*C*	√	√	-
MOPSO [51]	*Fully*	*R*	*R*	*R*	√	√	-
OPSO [121]	*Fully*	*R*	*R*	*R*	-	√	-
Modified PSO [156]	*Fully*	-	*C*	*C*	√	√	-
IPSO [60]	*Fully*	*L*	*C*	*C*	√	√	√
PS-EA [122]	*Fully*	-	*C*	*C*	-	√	-
MOPSO [46]	*Fully*	*L*	*C*	*C*	-	√	-
MOPSO [157]	*Fully*	*C*	-	-	-	√	-
MOPSO [29]	*Fully*	*C*	-	-	-	√	-
MOPSO [112]	*Fully*	*C*	-	-	-	√	-
MDEPSO [113]	*Fully*	*C*	*R*	*R*	-	-	√
PAPSO [158]	*Fully*	*C*	*C*	*C*	√	√	-
Covering MOPSO [89]	*Fully*	*C*	*R*	*R*	-	√	-
Parallel MOPSO [95]	*Fully*	*D*	*C*	*C*	-	√	-
AMPSO [61]	*Fully*	*L*	*C*	*C*	√	√	-

w: Inertia weight, C_1: Cognitive factor, C_2: Social factor, *R*: Randomly drawn from range, *Fully*: Fully-connected, *L*: Linearly changes, *C*: Constant, *D*: Dynamic, —: N/A

Table 2.5: List of Comparable Structure and Design Parameters Used in PSO (continue)

Algorithm	**Topology**	***w***	***C_1***	***C_2***	**Mutation**	**Swarm best Guide**	**Neighborhood Best Guide**
VEPSO [98]	*Ring*	*C*	*C*	*C*	-	√	-
C-MOPSO [102]	*Fully*	*C*	*C*	*C*	√	√	-
H-MOPSO [102]	*Fully*	*C*	*C*	*C*	√	√	-
CPSO [159]	*Fully*	*R*	*R*	*R*	√	√	-
EMOPSO [48]	*Fully*	*D*	*D*	*D*	√	√	-
AMOPSO [160]	*Fully*				√	√	-
MOPSO [161]	*Fully*	*C*	*C*	*C*	-	√	-
MOPSOSS [162]	*Fully*	*C*	*C*	*C*	√	√	-
CPSO [54]	*Fully*	*R*	*R*	*R*	√	√	√
K-PSO [84]	*Fully*	*D*	*R*	*R*	-	-	√
EM-MOPSO [91]	*Fully*	*C*	*C*	*C*	√	√	-
DOPS [53]	*Fully*	*R*	*C*	*C*	-	√	-
MOPSO [55]	*Fully*	*C*	*C*	*C*	-	√	-
MOPSO [52]	*Fully*	*C*	-	-	√	√	-
CLPSO [56]	*Fully*	*L*	-	-	-	√	-
MOPSO [57]	*Fully*	*C*	*C*	*C*	√	√	-
MAPSO [58]	*Fully*	*C*	*C*	*C*	-	√	-
NichePSO [96]	*Fully*	*L*	*C*	*C*	-	√	-
PSO+KM [93]	*Fully*	*L*	*C*	*C*	-	√	-
OMPSO [81]	*Fully*	*R*	*R*	*R*	√	√	-
MEPSO [87]	*Fully*	*L*	*R*	*R*	-	√	-
k-PSO [94]	*Fully*	*C*	*C*	*C*	-	√	-
BinPSO [163]	*Fully*	*L*	*C*	*C*	-	√	-
Coop-PSO [164]	*Fully*	*L*	*C*	*C*	-	√	-
MDO/PSO [165]	*Fully*	*L*	*C*	*C*	-	√	-
$MOPSO_{LS}$ [166]	*Fully*	*C*	*C*	*C*	-	√	-
Modified MOPSO [167]	*Fully*	*R*	*R*	*R*	-	√	-
MOCPSO [168]	*Fully*	*D*	*C*	*C*	√	√	-
PSO [7]	*Fully*	-	*C*	*C*	-	√	-

w: Inertia weight, C_1: Cognitive factor, C_2: Social factor, *R*: Randomly drawn from range, *Fully*: Fully-connected, *L*: Linearly changes, *C*: Constant, *D*: Dynamic, ⁻ : N/A

Table 2.5: List of Comparable Structure and Design Parameters Used in PSO (continue)

Algorithm	**Topology**	***w***	***C_1***	***C_2***	**Mutation**	**Swarm best Guide**	**Neighborhood Best Guide**
SMPSO [49]	*Fully*	*D*	*R*	*R*	√	√	-
MOPSO [59]	*Fully*	*R*	*R*	*R*	√	√	-
PPS-MOEA [97]	*Fully*	*C*	*C*	*C*	-	√	-
PSO-EO [115]	*Fully*	*L*	*C*	*C*	√	√	-
Simplifying PSO [169]	*Fully*	*R*	*R*	*R*	-	√	-
MPSO [170]	*Fully*	*L*	*L*	*L*	-	√	√
TAPSO [171]	*Fully*	*L*	*D*	*D*	-	√	-
ACVPSO [42]	*Star*	*C*	*C*	*C*	-	√	√
ACPPSO [42]	*Star*	*C*	*C*	*C*	-	√	√
LDWPSO [172]	*Fully*	*L*	*C*	*C*	-	√	-
APSOM [173]	*Fully*	*D*	*C*	*C*	√	√	-
IPSO [50]	*Fully*	*C*	*C*	*C*	√	√	-
Micro-MOPSO [77]	*Fully*	*R*	*C*	*C*	√	-	√
ClustMPSO [106]	*Fully*	*C*	*C*	*C*	-	-	√

w: Inertia weight, C_1: Cognitive factor, C_2: Social factor, *R*: Randomly drawn from range, *Fully*: Fully-connected, *L*: Linearly changes, *C*: Constant, *D*: Dynamic, ⁻ : N/A

Chapter 3

Saskatoon West Water Distribution System

Hydraulic modeling of water distribution systems has had a lengthy history dating back to the work of the Babylonian era in the 5500 BC. The practice of exploring water resources (i.e., lakes, rivers, and wells) and transporting water through aqueducts has been around for over two millennia, [174].

In water distribution systems, *hydraulics*, *network*, and *simulation* are terminologies fundamentally used for designing and operating water distribution systems. H*ydraulics* refers to moving water through pipes. The concept of a *network* is fundamental to a water distribution model, since it contains all of the major components of the system such as pipes, nodes, pumps, valves, tanks, and reservoirs. Moreover, the network describes how those components are assembled and function as a whole system. The term *simulation* is the process of determining the hydraulic behavior of a network using its mathematical *model*.

A water distribution system is a hydraulic infrastructure consisting of junction nodes, pipelines, pumps, tanks, reservoirs, valves, and control equipment. Simulation is used to investigate and evaluate the behavior of a water distribution system over a period of time using numerical tools, referred to as hydraulic solvers. These are instrumented for designing, optimizing, controlling, and analyzing of water distribution systems, [174]. Hydraulic solvers are used to compute the flow in each pipeline and the pressure at junction nodes, [175]. They can also be utilized to evaluate the network response to events such as closing valves, changing pump settings, opening hydrants, and demand variations.

A commercial system referred to as the Saskatoon West water distribution systems (SW-WDS) is used as a case study in this dissertation. The system is a rural water distribution system operated by Sask-Water. It is well sized and manageable for conducting research experiments. It contains all the primary elements (e.g., pipelines, junction nodes, pumps, tanks, valves, etc.) as well as the problems associated with water supply management, including different pressure zones, treated and untreated water, and a large scope of water demands from minor to major users. The next section describes the layout of the Saskatoon West WDS and the derivation of the classical demand-head-driven analysis related to the pump operations. In section 3.5, the characterizing and modeling of the hydraulic behavior and the performance of a water

distribution system are developed. The SW-WDS's loading conditions and operating scenarios are simulated and compared to experimental data. The last section of the chapter presents the corollary of the distribution system upon the current operational scenario, and by comparing the simulation results to the collected field data; the hydraulic model for SW-WDS is demonstrated.

In this chapter, the operational considerations such as the electrical energy cost, maintenance cost, maximum power peak charges, network reliability, and chlorine residual are explained and mathematically represented for their use in optimization and the development of a cost function.

3.1. Water Distribution Systems: Models Representation

The model of a water distribution system contains all of the hydraulic elements of the system (i.e., pipes, junction nodes, valves, tanks, reservoirs, and pumps), and defines the constraints associated with the interconnection of these elements. In general, a network usually consists of nodes and pipelines, pumps, reservoirs, and valves (as shown in Figure 3.1).

There are many types of nodal elements in a water distribution model such as junction nodes, storage tank nodes, reservoir nodes, pump nodes, and valve nodes. Pipelines connect the nodes within the system. Table 3.1 lists common network elements, describes how they are differentiated in a numerical model, and explains their primary purpose.

Table 3.1: Common Elements for Model Representation

Element	Type	Main Modeling Function
Junction	Node	Inlet or outlet volume of water to/from the distribution system (i.e., demand, or inflow)
Tank	Node	Store and/or supply water
Reservoir	Node	Supplies water to the distribution system
Pipe	Link	Transfers water from one node to another in a system
Pump	Node or Link	Pumps water
Valve	Node or Link	Controls the flow and pressure at nodes

Table 3.2 lists data association to elements in a network model.

Table 3.2: Network Data for Model Representation [176]

Data	Detail
Nodes	▪ Number or name ▪ Coordinates, Elevation ▪ Type – Network junctions or end points, source of water

Pipelines	– Initial and end nodes – Diameter – nominal or internal – Length, Material, Construction year – Pipe roughness, Minor loss coefficients – Water Quality – Reaction rate coefficients: (bulk and wall)
Valves and Control Equipment	▪ Initial and end nodes ▪ Diameter, Length ▪ Roughness coefficients ▪ Type – Throttled, NRV, PRV, PSV, PCV, TCV, FCV
Pumping Stations	– Initial and end nodes – Diameter of suction and delivery pipe – Number or pump, name, pump type – Pump delivery rate, delivery head, power – Rotating speed, number of stage, efficiency – Pump characteristic "Q – H – P curve", protections – Type – Fixed or variable speed pumps
Reservoirs	▪ Number or reservoir name ▪ Shape and volume ▪ Inflow and outflow pipes arrangements ▪ Type – Storages, Water Towers

Figure 3.1 illustrates how these elements can be connected to one another to build a water distribution model.

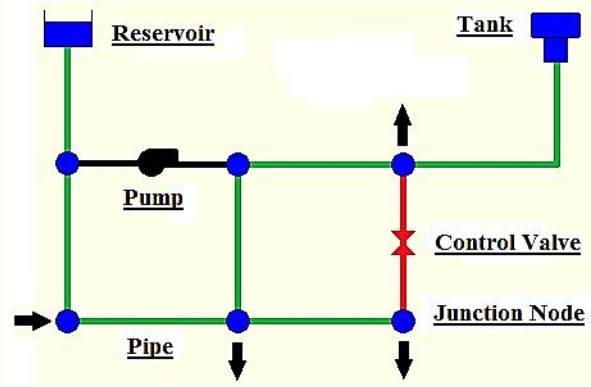


Figure 3.1: Physical Elements in a Water Distribution System

The representation of a water distribution model of a real system is developed by first gathering necessary information that describe the real system, followed by assembling the network model, and finally performing the simulations of the dynamics of the network.

3.1.1 Reservoirs

A reservoir represents a type of nodal element that supplies or accepts water. Reservoirs usually have a large capacity and a constant hydraulic grade lines. Reservoirs are considered as an infinite source of water and their Hydraulic Grade Lines (HGLs) are unaffected with respect to the flow of water. This means that reservoirs can theoretically handle any amount of inflow or outflow for any length of time. In reality, there are no such infinite sources; however, for modeling purposes, it is assumed that the inflow and outflow of water have no effect on the hydraulic grade at a reservoir.

In water distribution model, lakes, groundwater wells, treatment plants clearwells, and rivers are often represented as reservoirs. The primary input properties for a reservoir are the Hydraulic Grade Line (HGL) (water surface elevation in feet or meter units) and the initial quality of water in terms of chlorine residual as explained later in Section 3.5.1.5. Furthermore, to distinguish between reservoirs and tanks; no volumetric storage data is needed in reservoirs.

3.1.2 Tanks

In a network model representation, storage tanks are portrayed as nodal elements with a limited capacity for storing water. Tank characteristic capacity is affected by their form and geometry. Unlike reservoirs, the Hydraulic Grade Lines (HGLs) in tanks fluctuate according to the water level, and hence, by the in and out flow of water. They have a finite storage capacity, and can therefore be completely filled or partially exhausted.

For network security, water levels within tanks are maintained according a minimum and a maximum levels. HGLs in tanks are influenced by their geometry and water level as shown in Figure 3.2.

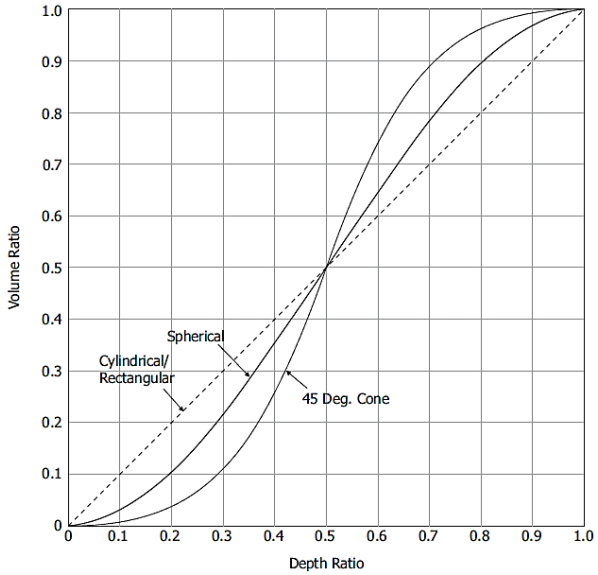
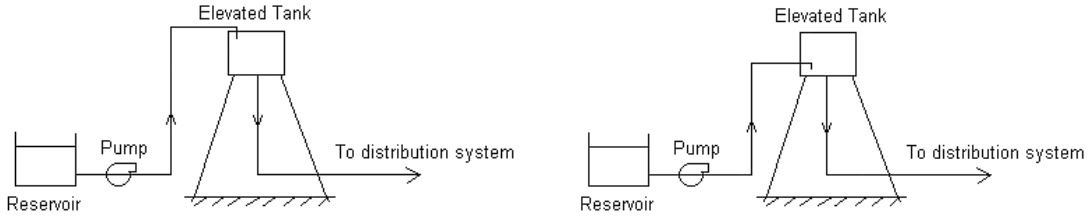


Figure 3.2: Various Elevation Conventions for Modeling Tanks [174]

Tanks in water distribution systems are fed by two methods: top-feed or bottom-feed gravity as shown in Figure 3.3. The first is most favored, since water level with the tank does not supply line pressure. The latter causes changes in the HGL of the system.



a. Top Feed Tank b. Bottom Feed Tank

Figure 3.3: Top Feed/Bottom Gravity Discharge Tank

3.1.3 Junction Nodes

A junction node is a location where two or more pipes are joined. Generally, a junction elevation is the only physical characteristic that is required to be defined by a modeler. This parameter is important in determining the node pressure. Figure 3.4 depicts four possible junction elevation options that can be used in a network model. It is often convenient to take one of the following options for the junction elevation: point A located at the centerline of the pipe; point B sited on the ground elevation above the pipe; point C that is the elevation of a hydrant; or point D that is the ground elevation of the user.

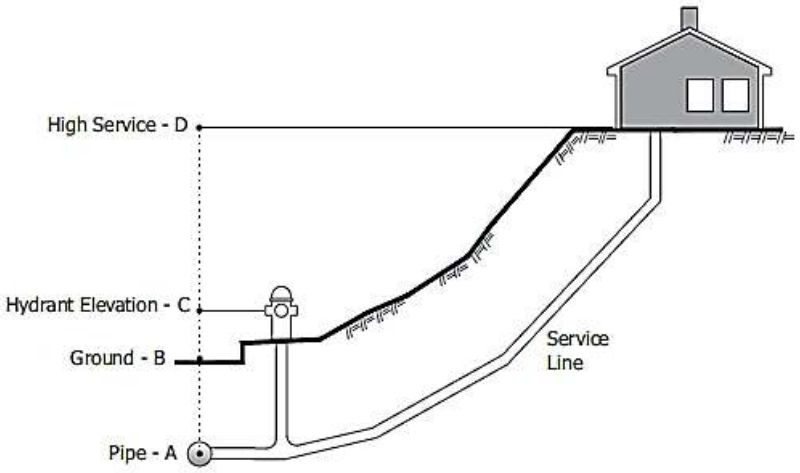


Figure 3.4: Elevation Options for a Junction Node [174]

3.1.4 Pipes

Pipes are used to transport water from one junction node to another. In a real distribution system, pipelines are joined together using various types of fittings, such as elbows (i.e., to sustain sudden changes in flow direction) or isolation valves (i.e., to block off flow through a particular section of pipelines, and usually found close to the reservoirs in water distribution systems).

To model pipelines in water distribution systems, properties such as bends, length, width, material type, and minor losses coefficients need to be considered.

3.1.5 Pumps

A pump is an element that imparts energy to a fluid, thereby raising its hydraulic grade. Pumps are generally classified into the two main categories of positive displacement and centrifugal. Both categories are related to the way pumps add energy to the moving water. Positive displacement pumps work by adding pressure to water that is essentially squeezed out using either piston strokes or shaft rotations. Centrifugal pumps work by adding kinetic energy to water using a spinning impeller. It should be noted that the most frequently used type of pumps in water distribution systems is the centrifugal pump. Pumps may be operated in an ON/OFF mode or in a variable frequency mode whereby their speed is regulated to maintain a fixed supply head pressure. An important consideration in pump modeling is the pump curves as shown in Figure 3.5. These curves demonstrate the relationship between the pressure head and flow as well as the efficiency of the pump at its various operating points.

3.1.5.1 Pump Operating Point

The pump operating point is only relevant to centrifugal pumps. The operating point is always where the pump curve and the system curve[3] intersected as shown in Figure 3.5. This point is very important as it determines the preferred operating point of the system, including pump flow rate, the pressure, and the energy consumption.

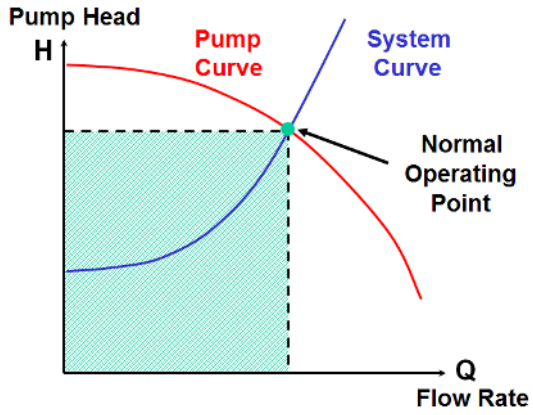


Figure 3.5: Pump Operating Point at Fixed Speed Pump

For a variable-speed pump, the characteristic curves are presented as a function of the pump speed. Hence, there will be a normal operating point for each pump speed as shown in Figure 3.6.

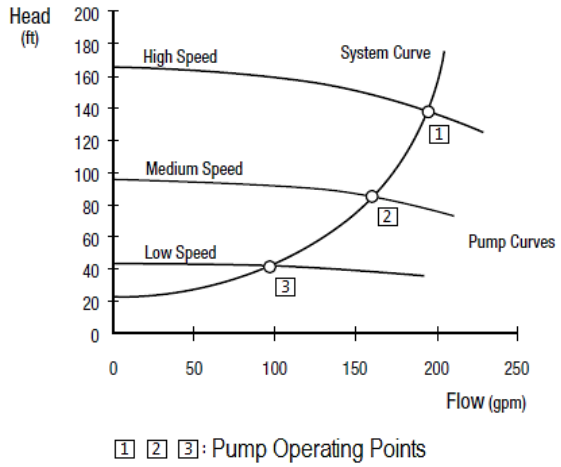


Figure 3.6: Pump Operating Points at Variable-Speed Pump

[3] The system curve is a graphical means to demonstrate the head required to transfer water through a system of pipes, fittings, and vessels. It is typically plotted as a 2-dimensional graph between the flow rate (x-axis) and the head (y-axis).

3.1.5.2 Best Efficiency Point (BEP)

Both the centrifugal pumps and the positive-displacement pumps have limitations that can affect their efficiency and reliability. These limitations can lead to fault conditions and subsequent reduction in the pump's working life. In terms of reliability, fault conditions could result from:

- Excessive load on the bearings,
- Excessive deflection of mechanical seals,
- Excessive radial thrust on the pump shaft, and
- Irregular wear on the shaft bearing,

The Best Efficiency Point: is the point where the normal operating point corresponds to the highest efficiency of the pump as shown in Figure 3.7. It is the point at which the pump produces the greatest amount of output for *the least amount of energy*.

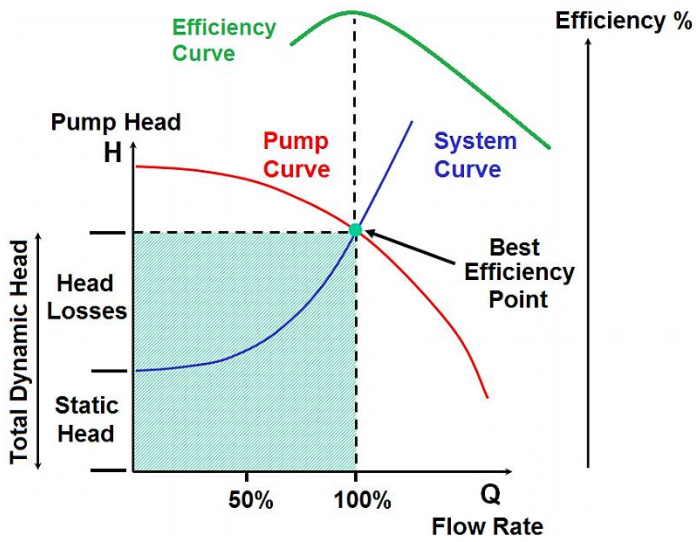


Figure 3.7: Pump Characteristic Curves illustrating the Concept of Best Efficiency Point

Benefits that ensue from operation at the Best Efficiency Point (BEP) include:

- Less energy consumption,
- Least pump maintenance cost,
- Efficient performance, and
- Prolonged motor and pump duty life.

In the case of variable-speed pumps, different pump speeds produce different head and flow characteristics. The rules of thumb suggest that pumps should operate within 80% to 110% of BEP. The centrifugal variable-speed pump curves as well as the range of best efficiency points are illustrated in Figure 3.8.

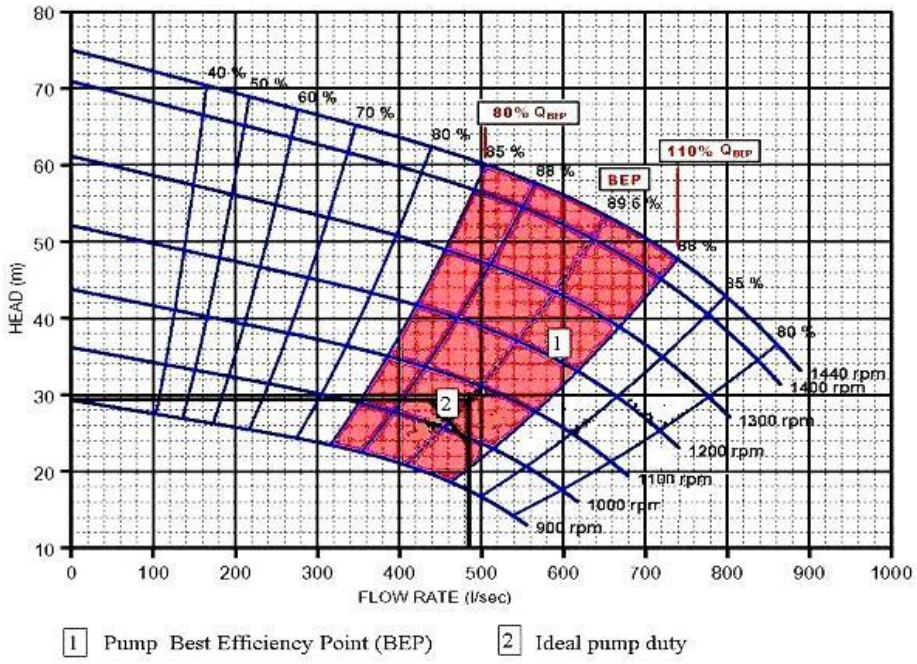


Figure 3.8: Performance Chart for Variable-Speed Pump

3.1.6 Valves

In General, valves are a common element in water distribution systems, and frequently modeled as links. The valve's role is to regulate the pressure or flow at a certain location in the network. They can be fully or partially opened or closed in order to change their resistances to flow. Valves in water distribution systems are broadly classified as follows:

- *Isolation Valves*

Isolation valves are commonly used to block off the flow of water through pipelines for maintenance (i.e., such as replacing broken pipes, leaky joints, regular maintenance, etc.) or emergencies (i.e., such as fire, major system failures, etc.). The most popular types of isolation valves that can be used are gate valves, butterfly valves, global valves, and plug valves.

- *Directional Valves*

Directional valves, known as check valves, are used to allow water to move in one direction through the pipeline. Check valves are usually found in pump stations, particularly at the discharge side of a pump, to prevent backflow by using hinged disk and flap mechanisms.

- *Altitude Valves*

This type of valve is widely used in water distribution systems, especially in pipelines that enter the tanks. These valves are designed to close when the tank level reaches its upper limit in order to prevent any extra flow from entering.

- *Control Valves*

Control valves, also known as regulating valves, are typically used to control flow and pressure. Flow control valves are operated by imposing a set of flow regulations, while throttle control valves are worked by adjusting its minor loss coefficient. Furthermore, pressure-based controls are functioned by either controlling the hydraulic grade or by applying a pressure setting. In network models (i.e., models that are driven by the changes in the HGL of the systems or its elements), both pressure settings and valve elevations are critically important to have in order to have a correct and efficient network operation.

Unlike pumps (i.e., which are either ON or OFF), all types of control valves can be in any one of the following operating status:

- Fully closed (i.e., automatically controlling flow, stated as active),
- Fully open (i.e., automatically controlling flow, stated as active),
- Throttle (i.e., automatically throttling pressure and flow, stated as active),
- Closed (i.e., manually controlling flow), and
- Inactive (i.e., ignored).

Most of the hydraulic solvers can deal with several types of control valves, including:

1. Pressure Reducing Valve (PRV) limits the pressure at specific location in the pipe network. PRV can be in three different states that are partially opened (i.e., active), fully open and closed (i.e., in cases where reversal flow is not allowed).
2. Pressure Sustaining Valve (PSV) maintains pressure settings at a specific location in the pipe network. Similar to PRV, PSV can also be in three different states that are partially opened (i.e., active), fully open and closed (i.e., in cases where reversal flow is not allowed).
3. Pressure Breaker Valve (PBV) is always used in numerical simulation. This valve forces a pressure loss to take place across the valve. PBV is virtually used to model situations where a certain pressure drop is known to exist (i.e., for the purposes of network modeling, as the PBV is not a true physical device).
4. Flow Control Valve (FCV) keeps the flow to a specified volume. FCV is operated without adding any additional head at the valve.
5. Throttle Control Valve (TCV) models one of the most common valves that can be partially closed.

3.1.7 Switches

Switches, known as operational controls, are commonly used in water distribution systems to automatically monitor and have a control action or to change the setting of an element.

For example, in a fixed-speed pump, the simplest control action that can be taken is to switch a pump ON or OFF. Whilst for a variable-speed pump, the control is obviously more sophisticated, and is applied by varying its speed. In pumping stations (i.e., where two or more

pumps are installed either in series or in parallel), pumps are often labeled as "*lead*" and "*lag"*. A pump is called *lead,* if it is the first pump that is turned ON, and called *lag*, if it is switched ON later.

3.1.8 Non-Physical Components

In addition to physical components that are equipped in real water distribution systems (likelihood aspects and elements that are previously mentioned), there are other components that are as important as the real ones called non-physical components. These non-physical components are essential in designing, modeling, and operating the distribution systems. Energy loss is an example of non-physical component, and is presented as follows:

3.1.8.1 Energy Losses

Energy loss is a non-physical consideration that has a large impact on the dynamics of water distribution systems. Energy loss can include friction losses or head losses (due to friction in pipes) and minor losses (due to turbulence within the bulk fluid).

3.1.8.1.1 Friction Losses

As depicted in Figure 3.9, a force balance on the water contained within a pipe section can be used to generally describe the formula of head loss due to friction. The free body diagram of the pipe segment illustrates the following forces:

- Pressure difference between sections 1 and 2,
- The weight of the water between sections 1 and 2, and
- The magnitude of shear stress developed at the pipe walls between sections 1 and 2.

Considering, for example, a constant flow velocity in the pipe, the system (i.e., forces exerted on the system) can be balanced according to the pressure difference, gravitational forces, and shear forces.

$$P_1 A_1 - P_2 A_2 - \bar{A} L \gamma \sin(\alpha) - \tau_\sigma N L = 0 \quad (3.1)$$

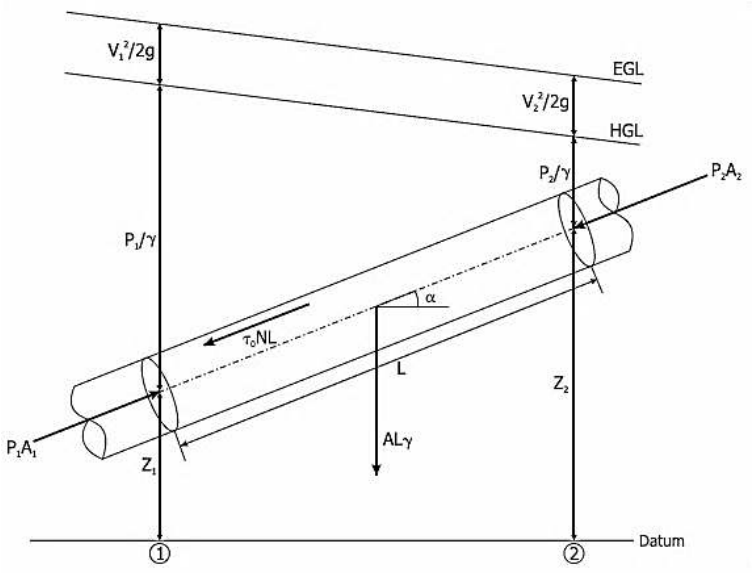


Figure 3.9: Free Body Diagram of Water Flow in an Inclined Pipe [174]

The friction losses along the pipe wall between the two sections are presented by the last term on the left side of Eq. (3.1). By equating $sin(\alpha) = (Z_2\text{-}Z_1)/L$, the equation for the associated head loss can be rewritten as:

$$h_L = \tau_\sigma \frac{N\,L}{\gamma\,A} = \left(\frac{P_1}{\gamma} + Z_1\right) - \left(\frac{P_2}{\gamma} + Z_2\right) \quad 3.2$$

For a laminar flow, the shear stress (τ_σ) can analytically be derived using Newton's law of viscosity, and is affected by the water viscosity (μ), the velocity gradient of water (V), the specific weight (or density) (γ), the pipe diameter (D), and the internal roughness of the pipe wall (ε).

The friction losses in water distribution systems can be calculated using one of the following formulas:

- **Darcy-Weisbach Formula**

The Darcy-Weisbach formula has been found to be the most accurate formula used to compute the friction losses and it applies over all flow regimes. The formula is written as follows:

$$h_L = f\,\frac{L\,V^2}{D\;2\;g} = \frac{8\,f\,L\,Q^2}{g\,D^5\,n^2} \quad 3.3$$

where f is The Darcy-Weisbach friction factor.

$$f = F\left(\frac{V\,D\,\rho}{\mu}, \frac{\varepsilon}{D}\right) = F\left(Re, \frac{\varepsilon}{D}\right) \quad 3.4$$

Recall that the Reynolds number correlates a fluid's velocity, density and viscosity as well as the pipe diameter. Also, note that the internal roughness is expressed in terms of a

variable referred to as relative roughness that correlates the internal roughness coefficient (ε) with the pipe diameter (D).

- **Colebrook-White Equation and the Moody Diagram**

There exist several formulas to correlate friction factor, Reynolds number, and relative roughness coefficient. The most common of these formulas is the Colebrook-White equation:

$$\frac{1}{\sqrt{f}} = -0.86 \ln\left(\frac{\varepsilon}{3.7\,D} + \frac{2.51}{Re\sqrt{f}}\right) \quad 3.5$$

Mathematically, Eq. (3.5) is considered hard to solve, since the friction factor (f) is found on both sides of the equation. Iterative methods are typically used for solving this equation. The Moody diagram (see Figure 3.10) is a graphical solution for the Darcy-Weisbach friction factor.

Considerable useful information can be attained from the Moody diagram, for instance, the friction factor is a linear function of the Reynolds number for laminar flows. Furthermore, the friction factor is only a function of the relative roughness coefficient for the turbulent flow.

- **Swamee-Jain Equation**

Swamee et al. proposed Eq. (3.6) in 1976, [177].

$$f = \frac{1.325}{\left[in\left(\frac{\varepsilon}{3.7\,D} + \frac{5.74}{Re^{0.9}}\right)\right]^2} \quad 3.6$$

This equation is much easier to solve than the Colebrook-White. Mathematically, the Swamee-Jain formula is an explicit function of the Reynolds number and the relative roughness. It is more accurate than the Colebrook-White equation by at least one percent over the following ranges:

$4 \times 10^3 \leq Re \leq 1 \times 10^8$ and $1 \times 10^{-6} \leq \varepsilon/D \leq 1 \times 10^{-2}$

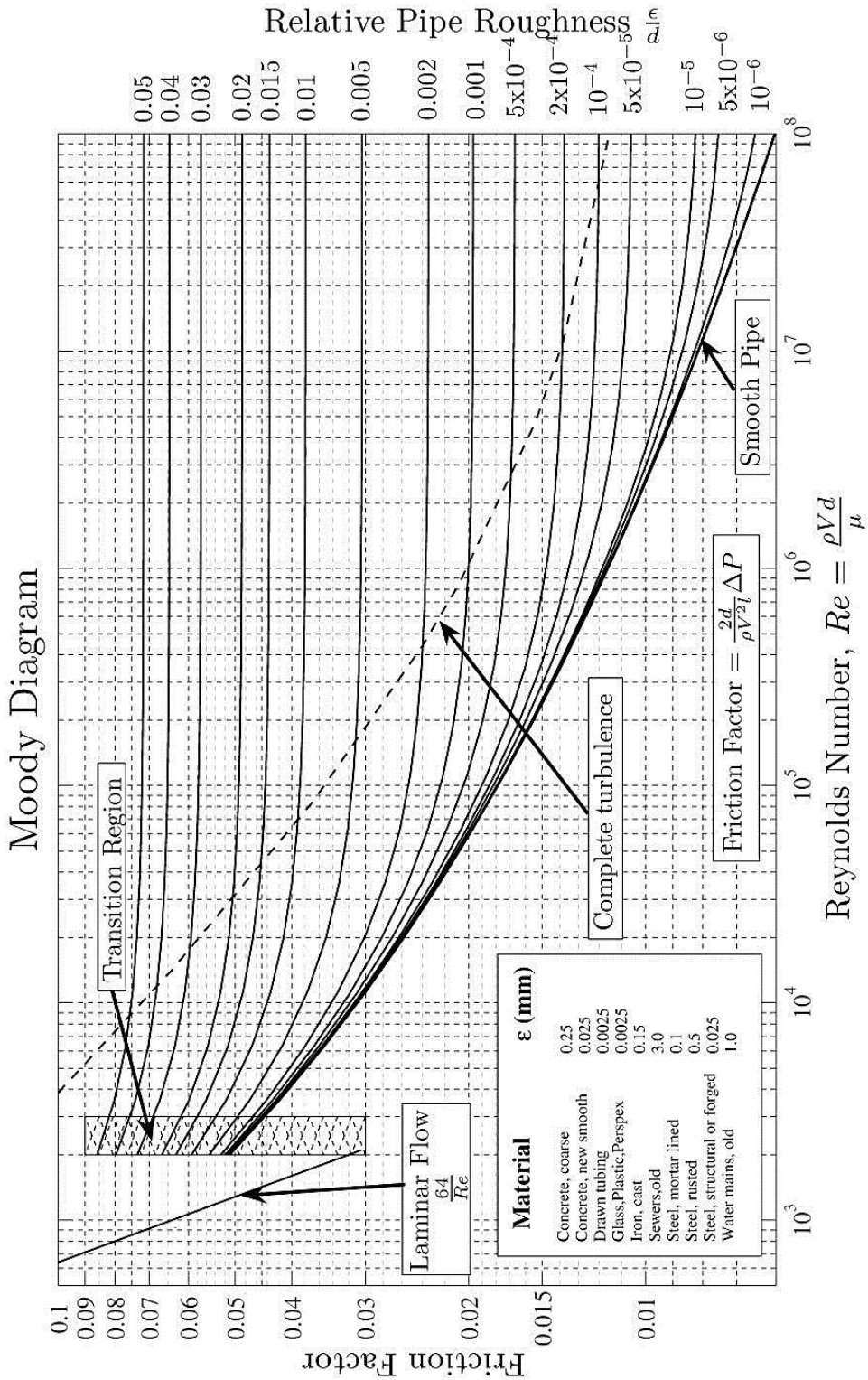


Figure 3.10: Moody Diagram [174]

- **Hazen-Williams Formula**

Because of its relative simplicity and reasonable accuracy, the Hazen-Williams formula has been considered as the most commonly used head loss formula in water distribution modeling. Eq. (3.7) expresses the Hazen-Williams friction loss formula, where *C* is a new variable called the carrying capacity factor.

$$h_L = \frac{C_f\, L}{C^{1.852}\, D^{4.87}}\, Q^{1.852} \tag{3.7}$$

Generally, the higher the *C*-factor the smoother the pipes. This increases their carrying capacity. Hazen-Williams formula is originally developed for turbulent flow only. Therefore, the *C*-factor of a pipe is varied in accordance with the flow velocity under turbulent conditions. Eq. (3.8) is used to adjust the *C*-factor for different velocities.

$$C = C_o \left(\frac{V_o}{V}\right)^{0.081} \tag{3.8}$$

- **Manning Formula**

The Chezy-Manning formula is typically applied for open channel flow. The head loss, in accordance with the Manning formula, is expressed as:

$$h_L = \frac{C_f\, L\, (nQ)^2}{D^{5.33}} \tag{3.9}$$

As can be seen in the Manning equation, the friction losses are dependent on the pipe size, the flow through the pipe, and the roughness coefficient.

- **Comparison of Friction Loss Methods**

There are three distinct formulas typically used to compute the friction losses (or head losses). Each formula uses a different category to define the pipe roughness coefficient that must be determined either mathematically or empirically.

In network modeling, most of the hydraulic solvers and simulators have the option of selecting the Darcy-Weisbach, the Hazen-Williams, or the Manning formulas for computing the friction losses. The selection is usually dependent on the nature of the problem and the modeler's preferences. The Darcy-Weisbach formula has a more theoretical basis than the other head loss formulas, as it is derived from Newton's Law. This formula can be used for any Newtonian fluid. The Hazen-Williams and the Manning formulas generally apply to water system with turbulent flow conditions. Table 3.3 lists expressions for the friction coefficient (S_f) in terms of friction loss per unit length of pipe.

Table 3.3: Friction Loss Equations in Different Units

Equation	$Q\ (m^3/s); D(m)$	$Q\ (cfs); D(mft)$	$Q\ (gpm); D(in)$
Darcy-Weisback	$S_f = \frac{0.083\ f\ Q^2}{D^5}$	$S_f = \frac{0.083\ f\ Q^2}{D^5}$	$S_f = \frac{0.083\ f\ Q^2}{D^5}$
Hazen-Williams	$S_f = \frac{10.7}{D^{4.87}} \left(\frac{Q}{C}\right)^{1.852}$	$S_f = \frac{4.73}{D^{4.87}} \left(\frac{Q}{C}\right)^{1.852}$	$S_f = \frac{10.5}{D^{4.87}} \left(\frac{Q}{C}\right)^{1.852}$
Manning	$S_f = \frac{10.3\ (nQ)^2}{D^{5.33}}$	$S_f = \frac{4.66\ (nQ)^2}{D^{5.33}}$	$S_f = \frac{13.2\ (nQ)^2}{D^{5.33}}$

3.1.8.1.2 Minor Losses

Minor Losses, occur at valves, fittings, bends, and other appurtenances associated with a real-world piping system. These losses are called "relatively" minor, as they do not substantially contribute to the overall energy losses (i.e., minor loss is much smaller in comparison to the energy loss due to friction) through the piping systems. Figure 3.11 illustrates the development of the turbulent vortices when water flows through a valve (i.e., the graph on the left) and a 90-degree bend (i.e., the graph on the right).

Although Minor losses are usually insignificant, in some cases, at high flow velocity and certain fitting configurations, minor losses can play a considerable role in the piping systems (i.e., at pump station).

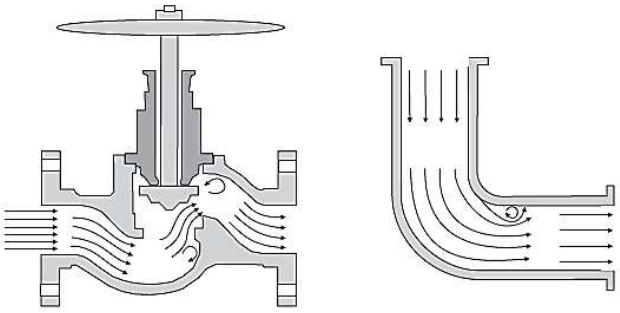


Figure 3.11: Example of Generating Minor Loss at Valve and Bending Cross-section [174]

The formula used to compute Minor losses is expressed as a function of the minor loss coefficient (K_L) and the velocity head (V), as shown in Eq. (3.10).

$$h_m = K_L \frac{V^2}{2\ g} = K_L \frac{Q^2}{2\ g\ A^2} \tag{3.10}$$

For network modeling, minor loss is treated as a pipe property, and therefore, in modeling water distribution systems, the minor loss coefficient is taken as a constant.

- **Valve Coefficient**

By using Eq. (3.11), the minor loss coefficient (K_L) is expressed as a function of the percent opening and valve coefficient (C_v) as shown in the equation below.

$$K_L = \frac{C_f D^4}{C_v^2} \quad 3.11$$

- **Equivalent Pipe Length**

In early system modeling, water distribution problems treaded the minor loss as a pipe property, and hence, the minor loss coefficient is computed indirectly by adding an equivalent length of pipe instead. The equivalent length of pipe to account for the same head loss is written as follows:

$$L_e = \frac{K_L D}{f} \quad 3.12$$

In modern network modeling, the surrogate minor loss coefficient by an equivalent length is no longer a valid practice. It is now directly incorporated into the network model.

3.1.8.2 Resistance Coefficients

For an ideal hydraulic model, a new variable, namely resistance coefficient, has been developed to mathematically express the head loss. Unlike the friction losses, the resistance coefficient is formatted to remain the same irrespective of which formula is utilized as presented in the following equations.

$$h_L = K_P Q^Z \quad 3.13$$

Equations for calculating K_P with the various head loss methods are expressed as follows:

- **Darcy-Weisbach**

$$K_P = f \frac{L}{2\ g\ A^2\ D} \quad 3.14$$

- **Hazen-Williams**

$$K_P = \frac{C_f\ L}{C^Z\ D^{4.872}} \quad 3.15$$

- **Manning**

$$K_P = \frac{C_f\ L\ n^Z}{D^{5.33}} \quad 3.16$$

- **Minor Loss**

Similar to the pipe resistance coefficient, the resistance coefficient for minor losses can be described in terms of the fitting or appurtenance and flow as shown in the following equation.

$$h_m = K_M \, Q^2 \tag{3.17}$$

Solving for the resistance coefficienct by subsituting from Eq. (3.10), yields:

$$K_M = \frac{\sum K_L}{2 \, g \, A^2} \tag{3.18}$$

3.1.8.3 Curves

3.1.8.3.1 Pump Characteristics Curve

The principal design parameters used to describe the pump performance are as follows:

1. Head: total dynamic head added by pump (i.e., energy supplied to fluid by a pump and usually measured by units of length),
2. Efficiency: the power that is imparted to the water divided by the power that came in over the electrical wires to the motors. It is also known as overall pump efficiency or wire-to-water efficiency,
3. Brake horsepower: is the real power going to the pump, not the power used by the motor (i.e., usually measured in units of kilowatt), and
4. Net Positive Suction Head (NPSH): the head that must be maintained in the pump to avoid cavitation.

Figure 3.12 shows the pump characteristics in an illustrated graph.

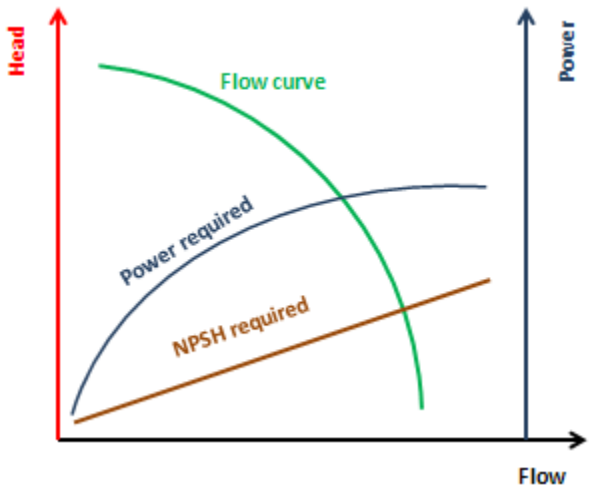


Figure 3.12: Pump Characteristics Curve

Since it is not practical to experimentally generate the pump characteristics curves at the site of the pumping systems, pump manufacturers usually provide pump curves to their customers. A pump curve represents the relationship between the head added to the water by the

pump and water discharge through the pump (i.e., flow rate), and is used to define reasonable settings for a wide operating range of the pump.

In water distribution systems, pump designers often select oversized pumps (i.e., more than the needs of a system) in order to allow for unexpected pressure increases. However, in systems with oversized pumps, there will be associated disadvantages, including:

- Operating with excessive flow,
- Increasing energy use,
- Higher pump maintenance costs, and
- Decrease in the operating-life of the pump.

3.1.8.3.2 System Curve

A System curve, known also as a system head curve or system resistance curve, is a prerogative non-physical element that defines what the pump is needed to perform (i.e., the required head to add) and under which conditions (or settings) the pump will operate. Furthermore, the system curve is continually sliding upward and downward along with the changes in the tank water levels and demands.

The system curve is generated by combining the static head[4] and variable frictional losses (i.e., head losses) based on the rate of discharge through the pump (as shown in Figure 3.7). Moreover, in a pumping system, the pump curve is often considered as a function of the pump and independent of the system, while the system curve is found to be pretty much dependent on the system and not the pump.

System curves can mathematically be described as a function of the water discharge, minor loss, pipe friction losses, and static head, as shown in the following case of a single pipeline between two points:

$$H = h_l + \sum K_P \, Q^Z + \sum K_M \, Q^2 \qquad 3.19$$

3.1.8.3.3 Efficiency Curve

An efficiency curve determines pump efficiency as a function of pump flow rate. It defines the overall wire-to-water efficiency of the pumping systems that count the mechanical losses in the pump as well as the electrical losses in the electric motor-driven centrifugal pumps.

[4] The static head, also called the static lift or fixed head, is defined as the elevation difference between the discharge tank water surface and the suction tank water surface.

Figure 3.13 shows the pump efficiency curves in the case of a variable-speed pump, as it is always preferable to operate the pump on a region where flow remains well controlled as close as to the highest efficiency curve as possible.

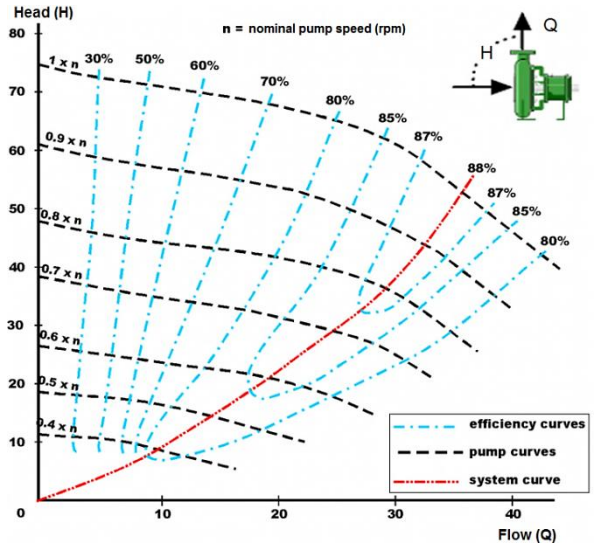


Figure 3.13: Example of System and Efficiency Curves in a Variable-Speed Pump [178]

3.1.8.3.4 Volume Curve

Another type of curve that is used in the hydraulic modeling of water distribution systems is the volume curve. A volume curve is very important in monitoring and controlling the operation of the storage tanks, especially for the non-fixed cross-sectional area, as it determines the volume of the storage tank as a function of water level. It is important to mention that the volume curve should contain the lower and upper levels between which the storage tank operates, as shown in Figure 3.14 whereas points 1 and 2 on the volume curve indicate the lower and upper water levels, respectively.

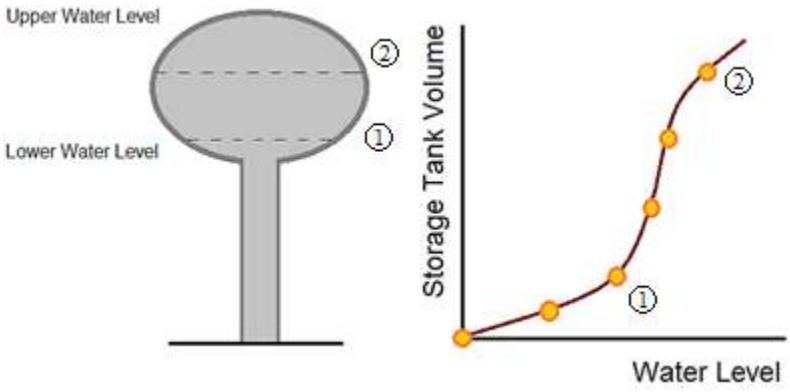


Figure 3.14: Storage Tank Volume Curve

3.1.8.3.5 Other Pump Curves

It is often convenient to mention that there are other pump curves than the aforementioned explained ones used to describe the pump's dynamic behavior, including power, water horsepower, head loss, and net positive suction head curves. An example is the Net Positive Suction Head (NPSH) curve that is involved in designing the pumping systems. NPSH is defined as a static head measured at the suction side of the pump (as shown in Figure 3.15). In designing a pumping system, the available NPSH must be higher than the required NPSH in order to avoid the occurrence of cavitation[5] due to the drop of local pressures within the pump below the vapor pressure of water. The NPSH available is commonly depended on the HGL of the water resource, the elevation of the pump and the head loss on the suction side of the pump. The NPSH required is given as a function of flow rate (as illustrated in Figure 3.16).

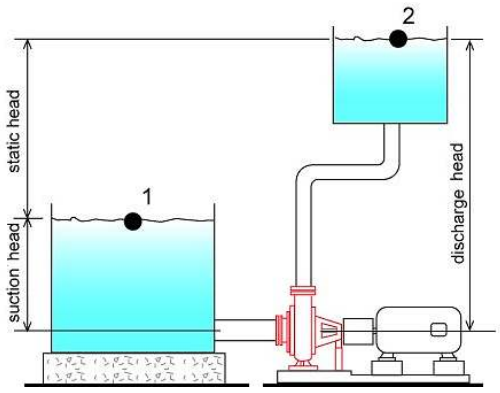


Figure 3.15: Suction and Discharge Sides of a Centrifugal Pump

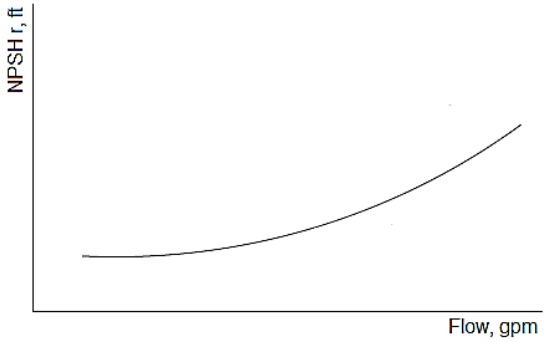


Figure 3.16: NPSH required Curve in a Centrifugal Pump

[5] Cavitation is a phenomenon that is usually observed in centrifugal type of pumps in which the system pressure (i.e., local pressures) is less than the vapor pressure of the fluid (i.e., water in distribution systems), causing the formation and violent collapse of tiny vapor bubbles that could severely damage the impeller of the pump which affect its overall performance.

An example of calculating the NPSAH available in a simple case where the pump takes suction directly from a tank can be expressed as follows:

$$NPSH_a = H_{bar} + H_s - H_{vap} - h_{loss} \quad 3.20$$

In network modeling, if the NPSH available is found to be less than the NPSH required, the cavitation will occur because of low local pressures, and in this case, the designer needs to choose one of the following corrective actions:

- Lower the height of the pump,
- Raise the suction tank water level,
- Replace the suction pipe with a bigger pipe diameter to reduce head loss, or
- Select a pump with a lower NPSH available or required.

3.1.8.4 Time Patterns

Within the hydraulic modeling of water distribution systems, there are many quantities that change over time, such as nodal demands, reservoir heads, pump schedules, and water quality inputs. To represent these changes, a time pattern is introduced to allow the modeler to accurately track the variations occurring within a fixed value of time interval. A time pattern consists of a set of multipliers that replicate themselves over 24 hours, and can be applied to any of the above quantities. At each time interval, a quantity remains constant, and it is calculated by multiplying the quantity nominal value by pattern's multiplier corresponding to that time interval.

For instance, assuming that the average demand of a node A is 5 GPM, and the time interval to be 4 hour over a 28 hour period (i.e., the duration of simulation), then the time pattern is expressed as follows:

Period	1	2	3	4	5	6
Multiplier	0.4	0.6	1.0	1.4	0.8	0.6

During the simulation, the actual demand applied at node A is found to be:

Hours	0-4	4-8	8-12	12-16	16-20	20-24	24-28
Demand	2	3	5	7	4	3	2

3.1.8.5 Affinity Laws for Variable-Speed Pumps

The Pump Affinity laws are commonly used in hydraulics, especially in water distribution systems, where the mathematical relationship between several pump characteristics (i.e., flow discharge, head discharge, and/or power consumption) directly influences its

performance. These laws can be applied to both types of centrifugal and axial flow pumps to predict the effects of changing the speed of a centrifugal pump or the diameter of the impeller on the pump performance. Being able to determine these effects would allow the design/maintenance engineer to assess the consequence of new measures before implementing changes.

3.1.8.5.1 Law 1: Affinity at Constant Impeller Diameter

In Centrifugal pumps, the first affinity law, with the impeller diameter held constant and the speed changed, is explained as follows:

1. Flow discharged is proportional to the speed of the motor, as illustrated in Eq. (3.21):

$$\frac{Q_1}{Q_2} = \left(\frac{n_1}{n_2}\right) \quad 3.21$$

2. Head produced is proportional to the square of speed of motor, as given in Eq. (3.22):

$$\frac{H_1}{H_2} = \left(\frac{n_1}{n_2}\right)^2 \quad 3.22$$

3. Power required is proportional to the cube of speed of motor, as yielded in Eq. (3.23):

$$\frac{P_1}{P_2} = \left(\frac{n_1}{n_2}\right)^3 \quad 3.23$$

The above equations are widely applied to variable-speed pumps in which a known pump curve at one speed is reasonable to predict the curve at another speed. Figure 3.17 shows the principles of applying affinity law with the impeller diameter held constant and the motor speed changed. Point 1 represents a known characteristic (i.e., flow, head or pressure, and power) measured at speed n_1, while point 2 represents the predicted characteristic at speed n_2.

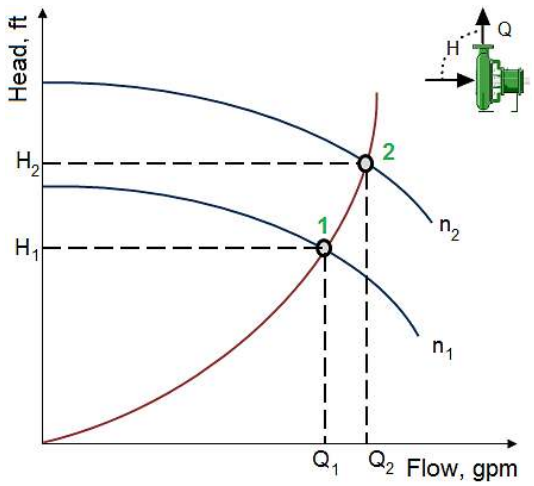


Figure 3.17: Principles of First Affinity Law in a Variable–Speed Pump

3.1.8.5.2 Law 2: Affinity at Constant Motor Speed

The second affinity law is useful to indicate the influence on the volume capacity (i.e., flow), head (i.e. pressure), and power (i.e., energy consumption) of a centrifugal pump due to being geometrically similar (i.e., change in impeller diameter). Applying this law to centrifugal pumps with the speed held constant and the impeller diameter changed is presented as follows:

1. Flow discharged is proportional to the impeller diameter, as shown in Eq. (3.24):

$$\frac{Q_1}{Q_2} = \left(\frac{D_1}{D_2}\right) \quad 3.24$$

2. Head produced is proportional to the square of impeller diameter, as stated in Eq. (3.25):

$$\frac{H_1}{H_2} = \left(\frac{D_1}{D_2}\right)^2 \quad 3.25$$

3. Power required is proportional to the cube of impeller diameter, as expressed in Eq. (3.26):

$$\frac{P_1}{P_2} = \left(\frac{D_1}{D_2}\right)^3 \quad 3.26$$

Using the above relationships, once the pump curve at any one diameter is known, and then the pump characteristic at another diameter can be approximately predicted.

3.2. Water Distribution Systems: Fundamentals of Hydraulics

In networks of interconnected hydraulic elements, every element is influenced by each of its neighbors; the entire system is interrelated in such a way that the condition of one element must be consistent with the condition of all other elements.

3.2.1 Network Principles of Hydraulics

In modern water distribution systems, utility customers are always expecting reliable and high quality water service. Therefore, operating such a water distribution system requires a good understanding of the fundamental flow relationships and the hydraulic concepts that govern the dynamic of flow in a complex water supply network.

Generally, there are two main concepts used to model the performance of the networks of interconnected hydraulic elements (i.e., water distribution systems) in terms of a set of mathematical non-linear equations. These concepts are *conservation of mass* and *conservation of energy*. The principle of *conservation of mass* states that the lumped (i.e., mathematical sum) of the fluid mass entering in and leaving from a node of a water distribution system must be equal to zero.

The mathematical representation of this concept is shown in the eq. (3.27):

$$\sum_{nodes} Q_i - U = 0 \qquad 3.27$$

In the case when the Extend-Period Simulation (EPS) is considered, a tank is used to store and withdraw water, thus, there is a need to add a new term to the above equation to describe the accumulation of water at certain nodes. Hence, eq. (3.27) can be re-written as follows:

$$\sum_{nodes} Q_i - U - \frac{dS}{dt} = 0 \qquad 3.28$$

Finally, It should be noted that the law of conservation of mass is applied to all nodes and tanks connected to the water distribution system (i.e., all nodes and tanks are represented in one equation).

In 1738, Bernoulli derived the principle of *conservation of energy*. He defined a way that regardless of the path taken between any two points, the difference in energy must be the same. The hydraulic interpretation of this law is illustrated as a relationship between the link flow and the link head loss. The mathematical formula of this law is expressed as shown in eq. (3.29):

$$Z_1 + \frac{P_1}{\gamma} + \frac{V_1^2}{2\,g} + \sum h_P = Z_2 + \frac{P_2}{\gamma} + \frac{V_2^2}{2\,g} + \sum h_L + \sum h_m \qquad 3.29$$

Eq. (3.29) shows that the difference in energy at any two points linked in a system is equal to the energy gains from pumps (i.e., adding head) and energy losses in pipes and fittings (i.e., due to friction in valves, pipes, etc.) that occur in the path between them. This equation can be applied to any open-path between any two points, or paths between reservoirs or tanks (i.e., with a known difference in head), or paths around a closed-path as the overall (i.e., resultant energy) changes in energy must equal to zero, as shown in Figure 3.18.

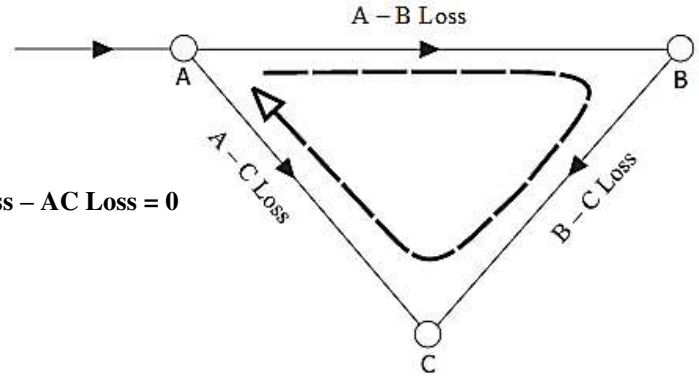


Figure 3.18: Principle of Conservation of Energy

3.2.2 Solving Hydraulic Models

A conceptual hydraulic model of a water distribution system can be presented in the form of an input/output system (as shown in Figure 3.19). Control Schedules (i.e., pumps and reservoir schedules), water demands (i.e., water that leave the system as per the demands of users), and initial conditions (i.e., tanks and reservoirs initial water levels) are considered the input to the hydraulic model (i.e., like a system in the context of control systems). The output comprises pressures at nodes, flows at elements (i.e., tank, pump, and pipe), and operating costs.

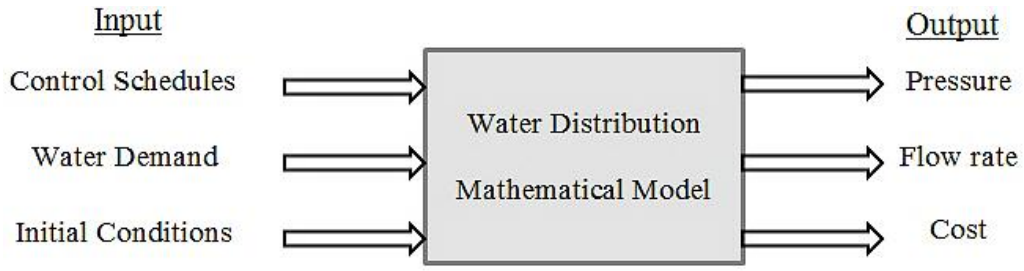


Figure 3.19: Concept of Hydraulic Model of Water Distribution System [179]

A comprehensive understanding of both hydraulics and solvers are needed to accurately simulate water distribution systems. From the principles of hydraulics, it is shown that network models of real water distribution systems consist of sets of governing continuity and energy equations. Hence, for each pipe in the system, one energy equation, and one continuity equation should be developed for each node.

Mathematically, due to the non-linear nature of energy equation systems expressed in terms of a flow-head relationship, the hydraulic models and water quality behavior of a water distribution system cannot directly be solved. Instead, the largely ongoing development in advanced digital computers and powerful numerical techniques allow the modelers to iteratively solve the system of equations with substantial computational accuracy.

Several approaches have been developed to numerically solve the hydraulic and water quality models in a water distribution system. These approaches can be classified as either Eulerian or Lagrangian. The Eulerian-based approach reformulates the governing flow equations into a series of sub-equations (i.e., as the case of a water quality mode in which each separate pipe is divided into equal length sub-links). The Lagrangian-based approach tracks changes in parcels of water as they travel through the pipe system. In both solution methods, a hydraulic model must determine the flow, flow direction, and velocity in each pipe over time during simulation. It should be noted that both approaches assume that a steady state hydraulic equilibrium is available.

3.2.2.1 The Eulerian Approach

The first viewpoint that is used to observe and analyze fluid flows by observing the fluid velocity at fixed positions is known as the Eulerian approach. In this approach, the hydraulic state of the system is updated at fixed positions as time progresses in uniform steps.

For modeling water quality in distribution systems, Grayman et al. proposed a new Eulerian solution technique in which each separate pipe is divided into a series of equal length sub-links. Then, the water quality concentration is adjusted during each water quality time as water moves from one sub-link to the next adjacent one. The sub-link lengths are varied from pipe to pipe and even within a pipe as flow changes, [180].

Figure 3.20 depicts the concept of Eulerian approach by observing the water (i.e., described as a boat) flow through the pipe of a water distribution system where the observer is standing on the sideline.

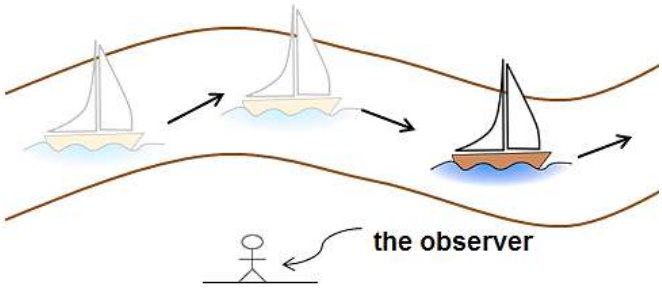


Figure 3.20: Concept of Eulerian Solution Method

3.2.2.2 The Lagrangian Approach

The second viewpoint of analyzing the fluid motion is to track the movement of a parcel of water throughout the pipe system at a fixed or variable time interval. In the Lagrangian approach, instead of observing the flow through a fixed grid (i.e., side point), the observer moves with the flow to fully track actual changes occurring over time.

The Lagrangian approach can be either time-driven or event-driven. In a time-driven method, the hydraulic state of the system is updated with a fixed time interval. In the event-driven case, the hydraulic state of the system is updated at the time when a change actually occurs (i.e., variable time interval). Rossman et al. compared the formulation and computational performance of the two methods, and found that the Lagrangian time-driven model is more efficient in obtaining accurate solutions than the event-driven for the hydraulic modeling of a water distribution system, [181].

Furthermore, the Lagrangian method is often found to be the most efficient approach used in simulating water distribution systems. It is also the most commonly used approach in

modeling water quality since it applies to moving parcels of water. Nevertheless, it is found that applying Lagrangian to a three-dimensional continuum problems is difficult, and thus, most of the theories in fluid mechanics are developed according to the Eulerian system of equations.

Figure 3.21 illustrates the concept of the Lagrangian approach by plotting the trajectory of parcels of water (i.e., described as boats) moving through the pipe of a water distribution system (i.e., described as a river).

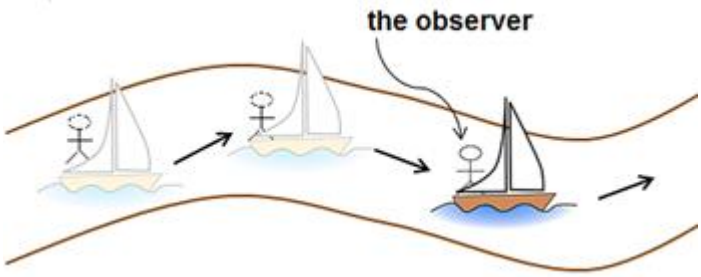


Figure 3.21: Concept of Lagrangian Solution Method

3.2.3 Calibrating Hydraulic Models

A concern associated with the use of hydraulic models is determining how well these models represent real physical systems, [174]. The main objective of any hydraulic model is to reproduce the behavior of a real system including its spatial and hydraulic dynamics in a manner that can be useful to analyze the system and then make decisions about system rehabilitation or major expansions.

The hydraulic simulators simply solve sets of system equations (i.e., continuity and energy equations) using field-data collected from the sites of a system. The accuracy of a hydraulic model essentially depends on the quality of the field-data. The process of comparing between the supplied field-data (i.e., also known as field observation data) and the results obtained from the hydraulic model is referred to as a calibration process or calibration analysis. Therefore, it is recommended that the calibration analysis should always be performed before a hydraulic model is used for decision-making purposes.

Within a calibration analysis, it is necessary that modelers do some adjusting to their hydraulic models (i.e., changing node demands, adjusting the roughness of pipes, altering pump operations, and fine-tuning other model characteristics) until a reasonable agreement between the predicted and measured system performance is obtained over a wide range of operating conditions.

3.2.3.1 Calibration Approach

A challenge associated with performing the calibration process is the adjustment required to bring the hydraulic model into agreement with the real system. Ormsbee et al., [182], suggested a guideline consists of seven steps to performing model calibration.

1. *Identify the intended use of the model:* identifying the application of the model, including system expansion (i.e., new pipes, new tanks, or new pumps), operational studies (i.e., valves settings, pumps settings, or water demands), design studies (i.e., pipe replacement or pipe sizing), and water quality studies (i.e., tank water levels). This would help the modelers to establish the level of information required in the model, the type of the measured data, and the level of tolerance for errors between prediction and actual.
2. *Determine estimates of model parameters:* there are some degrees of uncertainty associated with collecting field-data. Hence, parameters estimated from the hydraulic models are unlike those that are measured and may be changed with time. To compensate the difference between these two observations, special emphasis needs to be put on initial pipe roughness and node demand factors.
3. *Collect calibration data:* flow rate, pressure, and tank water level are the most common data measured for water distribution systems. According to the level of instrumentation and telemetry, most of those parameters are already collected as part of normal daily operations.
4. *Evaluate simulation results using an initial estimates of model parameters:* after properly obtaining the measured data (step 3), predicted results will be compared with them in order to evaluate the accuracy of the hydraulic models and for correcting the source of errors between the predicted and system results.
5. *Perform a rough-tuning or macro-calibration analysis:* when there are differences between the modeled and measured values, the acceptable approach is to review the data (i.e., input data) associated with the model and compare them with the field-data. During the macro-calibration, most of these errors and mistakes can be removed.
6. *Perform a sensitivity analysis:* at this stage, it is important to investigate the relationship between the parameter adjustments and their corresponding results.
7. *Perform a fine-tuning or micro-calibration analysis:* last procedure of the calibration process is that modelers attempt to fine-tune the model's parameters such as pipe roughness and nodal demands. Usually most of these attempts are performed using either an empirical approach (i.e., rules of thumb) or a trial-and-error approach.

In addition to these procedures, American Water Work Association – Engineering Computer Applications committee are frequently posting new standards and calibration guidelines on their websites in order to enhance the accuracy of the hydraulic models of water distribution systems.

3.3. Classes of Water Demands

Water demand, also called water consumption, is the driving force behind the hydraulic dynamics developing in a water distribution system. Therefore, it is important to precisely represent the consumption of water in hydraulic models. This would require the modelers to specify the locations for which water leave the system and categorize the demand patterns of water being delivered.

Spatial analysis is the most common method for describing water use at nodes by spatially distributing demands over time during simulation.

The node demands can be categorized as follows:

1. *Customer demand:* is the water required to satisfy the needs of users in the system, it represents the metered portion of the total water produced. There are many demand patterns used for this category, two of which represent the majority of water demands, namely residential and commercial patterns. Figures 3.22 and 3.23 show an example of these two patterns over the simulation time, respectively.

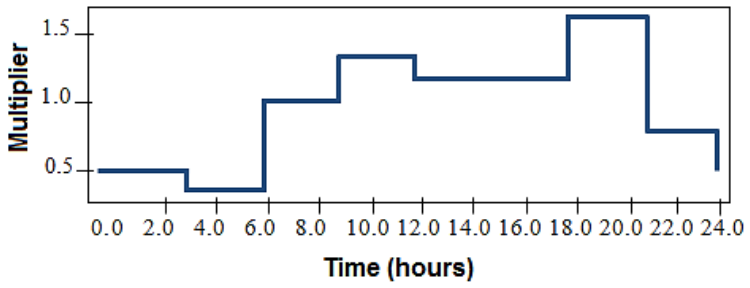


Figure 3.22: Residential Demand Pattern

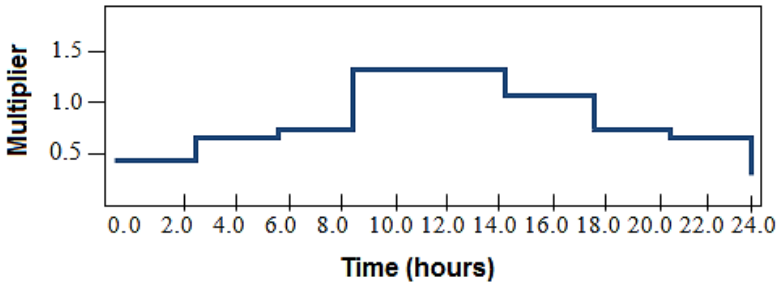


Figure 3.23: Commerial Demand Pattern

2. *Unaccounted-For Water (UFW):* is the portion of water that is produced and lost before it reaches the customers due to system leakage, theft, unmetered services, or other causes. Also, it represents the unmetered portion of total water produced in a system.

3. *Fire flow demand:* is a predetermined system capacity needed to control and extinguish fire, so that an adequate protection is supplied during fire emergencies. Figure 3.24 shows the fire demand pattern as an example of sudden event occurring in a water distribution system.

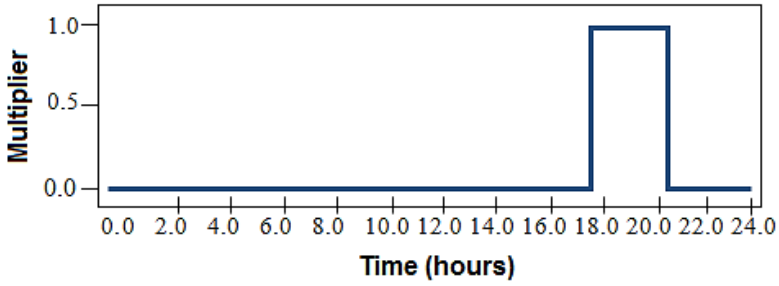


Figure 3.24: Fire Demand Pattern

An example of demand categories for the Olathe water distribution system in Kansas - USA is shown graphically in Figure 3.25. The total consumption data is obtained by tallying up the customers' bills between 2003 and 2005, [183]. From this graph, it can be seen that residential demand represents the majority of the water usage compared to other sectors. Two percent of the total water produced is considered water lost due to leakage.

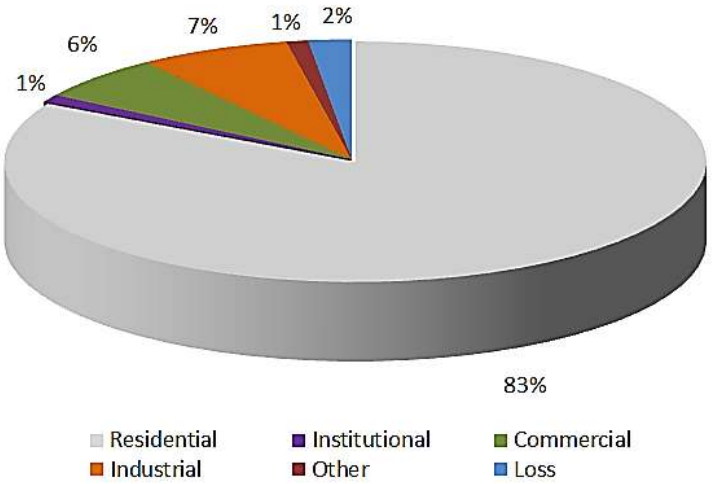


Figure 3.25: Water Demand by Customer Categories (2003-2005) [183]

3.4. Types of Hydraulic Simulations

There are several modes of operation of water distribution models, two of which are considered the most important ones: steady-state and Extended-Period Simulation (EPS). These two types of simulations are not only used to observe the hydraulic dynamics of a network, but

also to predict the hydraulic behavior that may occur due to changes in operating conditions, abrupt hydraulic events, or possible failure scenarios. Whereas steady-state simulation is mainly concerned with analyzing certain hydraulic behavior, including fire flow requirements, peak demand times, system elements failures, and master planning. An Extended-Period Simulation (EPS) tracks every hydraulic change that occurs over time by simulation.

To illustrate the fundamental concepts of these two types of simulations, basic definitions are provided:

- *Steady-State Simulation:* observes the states of the system (i.e., flows, pressures, pump operating status, valve position, etc.) under the assumption that hydraulic demands and boundary conditions are held constant during the simulation time.
- *Extended-Period Simulation (EPS):* performs a quasi-dynamic analysis for a system and determines the states of the system as a series of steady-state simulations. In this type of simulation, the assumption that hydraulic demands and boundary conditions are kept constant is not valid over period of time.

3.4.1 Steady-State Simulation

Steady-state mainly represents a state of a system[6] that does not vary over time, particularly the elements of a system that have achieved equilibrium during their long-term behavior. Tank water levels, reservoir hydraulic elevations, water demand, and pump operating status are all elements of a system that remain unchanged to define the boundary conditions of the simulation. Furthermore, hydraulic models are simulated using the steady-state mode to not only provide information about node pressures and pipe flow rate at equilibrium, but also to define other states of a system according to water demands and boundary conditions.

3.4.2 Extended-Period Simulation

It is essentially to mention that the mathematic foundation of Extend-period Simulation (EPS) can be represented as a sequence of steady-state simulations of the system whose water quality model, water demands, and boundary and operating conditions are updated over time.

One of the main applications of EPS is to perform water quality analysis. For example, water quality move at the same velocity as water through pipelines. Features at EPS enable the modelers to study many water quality issues, including water age (i.e., is the time taken by a

[6] State of a system is a set of numbers contains all the information required to estimate responses, with the use of characteristic equations describing the dynamics of the system, to present and future inputs without reference to the past inputs and outputs.

parcel of water in the network), residual chlorine (i.e., free chlorine concentration at nodes), and water source tracing (i.e., when water is drawn from two or more water resources).

3.5. Problem Formulation

One of the most researched areas in the hydraulics of water distribution systems is the operation optimization, [184,185]. Recently, population-based multi-objective optimization algorithms (e.g., Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony (AC)) become the preferred optimizers for operational excellence. Moreover, these algorithms offer opportunities for new operational strategies and insights into the requirements of a distribution system, [186].

Water distribution system operation usually involves multi conflicting objectives, including cost, reliability, and benefit. A Pareto concept in a multi-objective optimization is successfully applied to optimize the operation of a water distribution system. Savic et al. [187], proposed a bi-objective cost function to solve a pump scheduling problem. The multi-objective genetic algorithm used in their work is based on the concept of Pareto ranking from Goldberg [3]. They concluded that minimizing energy cost and the pump-switching criterion are achieved by producing a good set of solutions. Halhal et al. [188], developed a dual objective function, including capital costs and benefit (i.e., benefit in their work is referred to the network rehabilitation) using a structured messy genetic algorithm. The developed algorithm (SMGA) is able to produce a range of good solutions with varied costs. Doby et al. [189], investigated a genetic algorithm-based method for determining the least cost of looped networks while considering three objectives that are cost, redundancy, and water quality. The solutions produced by their approach are equally viable in response to the network problem. Farmani et al. [190], used a modified Non-dominated Sorting Genetic Algorithm method (NSGA-II) (i.e., developed by Deb et al. in 2000) to solve optimal design problems (i.e., in the context of water distribution systems). In their approach, the cost function is formulated using two objectives: minimizing the total cost of network expansion and rehabilitation, and maximize resilience for their “Anytown” network model. A third objective is added to their problem formulation that is minimum surplus head benefit that accounts for the network reliability in terms of flow provided through a system or head available at critical nodes under different load conditions. Their results are able to identify the trade-off characteristic between the cost of the network and its reliability.

In optimizing pump operation in a water distribution system, a simple model can be used because only relatively large mains between pumps, reservoirs, and junction nodes are important in calculations [174]. Therefore, in this work, a simplified hydraulic model based on a real-world water distribution system in Saskatoon is considered. Figure 3.26 shows the schematic model using five pumps installed in two pumping stations and has many grounded and elevated reservoirs. In pumping stations, pumping capacities are assumed during the time interval, and

therefore, each pump combination has a fixed maximum flow rate, electrical energy consumption, and maximum power. Table 3.4 presents the characteristics of a pump combination for the Saskatoon West WDS model.

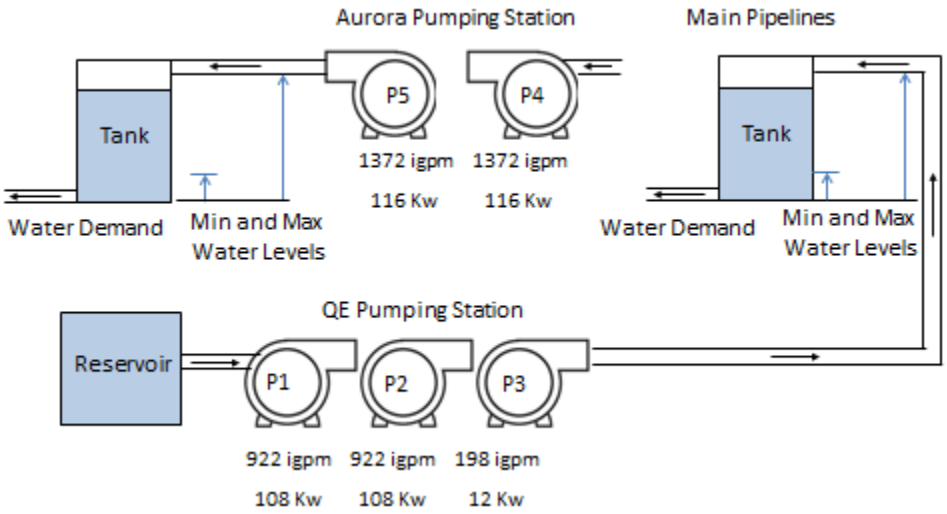


Figure 3.26: Schematic of Simplified Water Distribution Model

Table 3.4: Technical Characteristics of Pump Combinations

Pump Combination	Code P1...P5	Max. Flow (IGPM)	Power (KW)	Pump Combination	Code P1...P5	Max. Flow (IGPM)	Power (KW)
0	00000	0	0	16	10000	922	108
1	00001	1372	116	17	10001	2294	224
2	00010	1372	116	18	10010	2294	224
3	00011	N/A*	N/A	19	10011	N/A	N/A
4	00100	198	12	20	10100	1120	120
5	00101	1570	128	21	10101	2492	336
6	00110	1570	128	22	10110	2492	336
7	00111	N/A*	N/A	23	10111	N/A	N/A
8	01000	922	108	24	11000	1844	216
9	01001	2294	224	25	11001	3116	332
10	01010	2294	224	26	11010	3116	332
11	01011	N/A	N/A	27	11011	N/A	N/A
12	01100	1120	120	28	11100	N/A	N/A
13	01101	2492	336	29	11101	N/A	N/A
14	01110	2492	336	30	11110	N/A	N/A
15	01111	N/A	N/A	31	11111	N/A	N/A

* *N/A stands for not applicable due to system's regulation*

The mathematical expressions of the multi-objective function problem and its constraints are presented as follows:

3.5.1 Mathematical Definition of the Problem: Objectives

In this dissertation, a multi-objective analysis is adopted to mathematically represent the system-operational problems in WDS. The main objective of any energy optimization strategy, for a water distribution system, is to reduce costs while ensuring an acceptable level of service to water users, [191]. According to the multi-objective nature of the problem, some goals are in conflict with each other such that improving one objective will cause deterioration in another. Minimizing the operational cost commonly places the system under more risk of not being able to handle unusual events such as pipe ruptures, [192]. Energy optimization can be realized by many means, from the field-testing, standard maintenance of equipment, and proper rehabilitation of the network, to the use of optimal operating strategies. Energy optimization can be achieved by:

- Reducing energy costs in pumps (e.g., making efficient use of storage tanks, by filling them during off-peak (i.e., lowest electricity tariff) period and draining them during the peak period), and
- Increasing pump efficiency (e.g., operating the pumps near their best efficiency points).

The cost function can be considered as an evaluation criterion that is used to determine the quality of a solution. A multi-objective cost function for water distribution systems can include the following elements: electricity cost, pump maintenance, network reliability, maximum power peak, and water quality. A detailed explanation of the cost function and its mathematical formulation is provided as follows:

3.5.1.1 Electrical Energy Cost (f_1)

This is the cost of electrical energy consumed by all pumps of the pumping stations during the optimization period, [193,194]. This objective has a remarkable influence on the pump scheduling by substantially reducing the electricity energy cost when using the smallest possible number of pump combination during the operating time interval, [195,187]. It can take the advantage of storing water in the reservoirs when pumps are turned ON, and then using it to satisfy the community's demands during the period when pumps are turned OFF. In this work, a fixed-charge structure is considered when calculating the electricity cost. The mathematical representation of calculating the electrical energy cost is given by Eq. (3.30), [195]:

$$f_1 = \sum_{i=1}^{Tp} \underbrace{\left[T_r \; EC[P_C(i)]\right]}_{Electrical\ energy\ cost} \tag{3.30}$$

where:
i : time interval index,
Tp: whole optimization period,
Tr: rate of the electricity charge,
$Pc(i)$: pump combination at interval i using Np to denote the total number of pumps in the system, $Pc(i)$ is coded as a binary string in $\{0, 1\}^{Np}$. The code of pumps scheduling is provided in Table 3.4 (for $Np = 5$), and
$EC(Pc(i))$: is the electric energy consumed by pump combination $Pc(i)$ at time interval i, see *power column* in Table 3.4.

3.5.1.2 Pump Maintenance Cost (f_2)

The number of pump switches is the number of 0 1 sequences. A Pump switch is only counted if the pump was not working (i.e., OFF) in the preceding time interval and then turned ON. Therefore, a pump that was already ON in the preceding interval and then switched OFF does not count for a pump switch for the present interval. Figure 3.27 illustrates the concept of counting the pump switches for the two consecutive time intervals.

Pumps Status				
OFF (0)	OFF (0)	ON (1)	ON (1)	At time interval t
OFF (0)	ON (1)	OFF (0)	ON (1)	At time interval t+1
Not counted	Counted	Not counted	Not counted	

Figure 3.27: The Mechanism of Counting Pump Switches

Maintenance cost can be as important as electrical energy cost, and is characterized by the total number of times pumps are switched ON. This number is simply calculated by counting the number of pump switches at every time interval. A pump switch is counted only if the pump was OFF in the preceding time interval and then has been turned ON, [192]. Hence, the total number of pump switches is commonly used as a measure of maintenance cost. Therefore, minimizing pump switches is considered as an objective in the cost function, [196,191,197]. The mathematical representation of this objective is shown in Eq. (3.31):

$$f_2 = \sum_{i=1}^{Tp} \underbrace{\left|max\left(0; Pc(i) - Pc(i-1)\right)\right|}_{Total\ number\ of\ pump\ switches} \qquad 3.31$$

where $|\ \cdot\ |$ represents the 1-norm of a vector.

3.5.1.3 Network reliability (f_3)

Reliability in water distribution systems is defined as the ability of the system to meet the forecast demands that are placed on it according to the criteria of water flow to be supplied and pressure to be provided at the junction nodes, [198]. The above definition may be referred to as "the ability of the network to provide service at an acceptable level in spite of abnormal conditions". Accordingly, reservoir capacity and operation have a large impact on network reliability, [199]. Hence, at the end of each time interval, reservoir water level must be in a position between the maximum level (h_{max}) and the minimum level (h_{min}). These two levels are defined as follows:

- *Minimum level*: that guarantees enough pressure in the pipeline. This level should always be maintained for security reasons, so that the system can supply a large amount of water in a short time when such unexpected events could happen, and
- *Maximum level*: that is compatible with the reservoir's capacity.

Since network reliability cannot be mapped into a cost ($ dollar), it is stated that the reservoir level variation (Δh) between the beginning and the end of the optimization period is considered as a distinct objective to be minimized. The equation that describes the network reliability is given in Eq. (3.32):

$$f_3 = \Delta h = \underbrace{\sum_{i=1}^{Tp}\sum_{k=1}^{NT}\frac{\left[D\left(Pc_{i,k}\right) - d_{i,k}\right]}{S_k}}_{Reservoir\ level\ varaition} \tag{3.32}$$

where:

NT: total number of reservoirs in the system,

S_k: reservoir's surface, assumed constant,

D(Pc(i)): discharge pumped at time interval *i* using pump combination *Pc(i)*, and

$d_{i,k}$: water demand at time interval *i*.

3.5.1.4 Maximum power peak (f_4)

Some electrical suppliers charge their clients based on a limited power peak. An additional charge (an expensive charge) is added when the clients exceed their maximum allowable electric energy. It is computed, for billing purpose, according to the maximum peak power reached during the period. Therefore, in this work, a daily power peak is chosen as a separate objective in the cost function rather than considering it as a penalty added to the electricity energy cost, [200]. The maximum power peak is computed using Eq. (3.33):

$$f_4 = \sum_{i=1}^{Tp} \underbrace{max\left[P\left(Pc(i)\right)\right]}_{Cost\ of\ max.\ power\ peak} \tag{3.33}$$

where *P(Pc(i)):* power at interval *i* using pump combination *Pc(i)*, see Table 3.4.

3.5.1.5 Water quality (f_5)

Due to the complexity of most water distribution systems, water quality can undergo significant changes as water travels through the pipelines of the system from the treatment plant to the point of delivery, [201]. For example, chlorine degrades with time due to reaction with pipelines walls. During regular operations, the quality of water is distorted due to the acts of both mixing water from various resources and frequently switching the pumps ON and OFF which allows an excessive biofilm to be formed[7], [202]. Water quality is treated here as a distinct element of the objective function. This objective is mathematically described as shown in Eq. (3.34)

$$f_5 = \sum_{i=1}^{Tp}\left(\sum_{n=1}^{ND} \underbrace{\left(min\left[0, C_{n,max} - C_{n,i}\right]\right)^{m1}}_{Max\ free\ chlorine\ concentration} - \underbrace{\left(min\left[0, C_{n,i} - C_{n,min}\right]\right)^{m2}}_{Min\ free\ chlorine\ concentration}\right) \quad 3.34$$

where:

ND: total number of junction nodes (i.e., demand nodes),

$C_{n,max}$ and $C_{n,min}$: maximum and minimum bounds of free chlorine concentration at junction node *n*, and

$C_{n,i}$: chlorine concentration at junction node *n* during time interval *i*.

m_1 and m_2 are the power of the upper and lower bounds (usually equal to 2).

3.5.2 System Operational Problem: Multi-objective Cost Function

Energy saving optimization in water distribution systems are not concerned with cost reduction only, but also must consider other operational objectives that cannot easily be expressed in economic terms. Hence, effective network operation is a trade-off between multiple objectives. The goals of implementing a new energy optimization strategy for a water distribution system can successfully be achieved by:

- Minimizing energy costs,
- Reducing the costs of maintenance,
- Preserving network reliability, and
- Improving water quality.

[7] A biofilm is a complex aggregation of microorganisms. Biofilms are often characterized by surface attachment, structural heterogeneity, genetic diversity, complex community interactions, and an extracellular matrix of polymeric, [297].

In this research, a multi-objective approach is adopted with the above-defined objectives; the multi-objective system operational problem can be stated as follows:

$$Minimize\ y = \left(f_1(x), f_2(x), f_3(x), f_4(x), f_5(x)\right) \quad 3.35$$

where:

f_1: electrical energy cost; see Eq. (3.30),

f_2: number of pump switches; see Eq. (3.31),

f_3: network reliability; see Eq. (3.32),

f_4: maximum peak power; see Eq. (3.33),

f_5: water quality; see Eq. (3.34),

$x \in X \subset B^{168.Np}$ is the decision vector, $B = \{0,1\}$, and

$y = (y_1, y_2, y_3, y_4, y_5) \in Y \subset \mathbb{R}^5$ is the objective vector.

3.5.3 System Operational Problem: Non-linear Inequality Constraints

Mathematically, constraints on the optimization problem define the criteria that classify the solutions, obtained by the optimization algorithm, whether they are feasible[8] or not feasible in order to define the system performance. Constraints may be defined as operational and boundary conditions on junction node pressure, pipe velocity, tank water level, and pumps switches for a given water distribution optimization problem. In this work, the multi-objective cost function (as shown in Eq. (3.35)) is subject to a set of non-linear inequality constraints that is defined as follows:

- *Pressure Constraints at a Junction Node:*

For reliable service, the water should be supplied to consumers at adequate head pressure. Therefore, for each operational interval *i*, the pressure at any junction node *j* must always be maintained between the maximum and minimum values. This can be expressed as:

$$Pmin_j \leq P_j(i) \leq Pmax_j \quad , \quad \forall j\ \forall i \quad 3.36$$

where $P_j(i)$ denotes the pressure at node *j* at time interval *i;* $Pmin_j$ is the minimum pressure required at node *j*; and $Pmax_j$ is the maximum pressure allowed at node *j*.

- *Water Level Constraints in a Tank:*

A storage tank in a water distribution system operates between the minimum and maximum allowable water levels to prevent the tanks from being empty or overflowing. The limits on the tank water levels can be described as:

[8] A solution is called feasible if it satisfies all problem constraints. Likewise, an infeasible solution is defined as a solution to an optimization problem that does not satisfy all or potentially any of the constraints.

$$h_j(i) = h_j(i-1) + \frac{D\left(Pc(i)\right)_j - d_j(i)}{S_j}$$

$$h_j(i) \geq hmin_j \quad , \quad \forall j \; \forall i \qquad 3.37$$

$$h_j(i) \leq hmax_j \quad , \quad \forall j \; \forall i \qquad 3.38$$

where $hmin_j$ represents the minimum water level required at tank *j* to maintain an adequate emergency service*;* $h_j(i)$ is the estimated water level of tank *j* at time *i*; and $hmax_j$ represents the maximum allowable water level at tank *j*.

- *Pump Switching Constraints:*

In a water distribution system, the total energy cost can effectively be reduced by frequently turning pumps ON and OFF during an optimization period. However, the pump maintenance cost is relatively increased as pump switches increase due to excessive wear on the pumps. In order to limit the pump-wear-out cost, the total number of pump switches must be less than a maximum allowable value, given as:

$$SW_k \leq SWmax_k \quad , \quad \forall k \qquad 3.39$$

where SW_k represents the number of pump switching for pump *k* and $SWmax_k$ designates the maximum number of pump switches for pump *k*.

- *Chlorine Constraints at Junction Node:*

Constraints corresponding to water quality reflect the properties of chlorine concentration in the water at junction node. For each time interval *i*, the chlorine concentration at any junction node *j* must always be maintained within a maximum value and a minimum value. This constraint can be expressed as:

$$\underline{C}min_j \leq C_j(i) \leq \bar{C}max_j \quad , \quad \forall j \; \forall i \qquad 3.40$$

where $C_j(i)$ represents the chlorine concentration at node *j* at time interval *i;* $\underline{C}min_j$ is the minimum chlorine concentration required at node *j*; and $\bar{C}miax_j$ is the maximum allowable chlorine concentration at node *j*.

In summary, the defined multi-objective system operational problem and its nonlinear inequality constraints considers five conflicting objectives, including electrical energy cost, maintenance cost, network reliability, maximum power peak, and water quality are minimized. At the same time, the optimization algorithm evaluates the solutions according to the operational and boundary constraints associated with the optimization problem such as pressure and chlorine concentration at junction nodes, water level at tanks, and the total number of pump switches.

3.6. Hydraulic Simulations

The basics of hydraulic models as it applies to water distribution systems are presented in this section. It summarizes fundamentals of hydraulics and other related topics, including water

demand and allocation, model calibration, steady-state and extended period simulation, water quality analysis, and problem formulation.

The hydraulic simulation of water distribution systems is mainly used to solve the relationship between pressure and flow of the system over the course of the simulation time period. The simulation allows an accurate prediction of the network operation and potential events such as failures in network elements (i.e., burst pipes), level of water leakage, pressure fluctuations, energy use, and so on.

Hydraulic network simulation software packages use a graphical user interface (GUI) that makes it easier for modelers to build models and then visualize the dynamics of a network over a wide range of design and operating conditions. The most commonly used software packages are:

- WaterCAD: is a popular and efficient hydraulic simulator that also considers water quality modeling solution for water distribution systems. WaterCAD can analyze the water networks in both steady state and extended period simulation analyses, [203].
- EPANET: is software developed by EPA's Water Supply and Water Resources Division, and is used to model water distribution systems. It is able to simulate both hydraulic and water quality behaviors within pressurized water networks, [204].

3.6.1 Augmenting Hydraulic Solver in an Optimization Process

This research has developed an optimization cycle referred to as optimization cycle for water distribution systems. It integrates the optimization algorithm with the hydraulic simulation software (i.e., EPANET) as shown schematically in Figure 3.28. The arrows in this figure indicate the normal process of the optimization cycle. For a real-world water distribution problem, the optimization process typically begins with identifying full details of the problem. Moving from the *real-world problem* to the *optimization formulation* is known as analysis. It is here that the hydraulic model of the water distribution system is built using the important elements extracted from the previous step. Moving from the *optimization formulation* to the *algorithm or solution technique* is often known as a cost function model. In this step and based on the problem complexity, the cost function is mathematically expressed by two or more objectives; each has its own space dimension and usually are in conflict with other objectives. Laying bare the essential information of a problem has often led to insights into how well an algorithm or a solution technique can approach the problem. Moving from the *algorithm or solution technique* to the *computer implementation* is referred to as a numerical model. This covers issues related to analyzing, solving, and simulating the hydraulic model of the WDS using the hydraulic solver. Digital computers, MATLAB package, and the concept of multi-islands (i.e., parallel algorithm) are very helpful in obtaining an accurate performance and efficient implementation of the optimization algorithm.

On the return path of the process, moving from the *computer implementation* back to the *algorithm or solution technique* is called verification. The main idea is that the hydraulic solver verifies that the proposed operating scenario meets the network's boundaries and operational constraints. Performance metrics and sensitivity analysis are used for validation and for measuring the quality of the obtained solutions. Validation is defined as the process of confirming that the hydraulic model, solutions technique, and obtained solutions are all appropriate for the real-world problem, while sensitivity analysis studies the effect of changing specific algorithm parameters on the results. These two processes occur when moving from the *algorithm or solution technique* and the *optimization formulation*. Moving from the *optimization formulation* to the *real-world problem* is known as a decision maker. It is here that the loop of the optimization cycle is completed and finally the obtained results are compared to choose a solution amongst a set of Pareto-front solutions that is the most reliable, appropriate, and suitable to the problem.

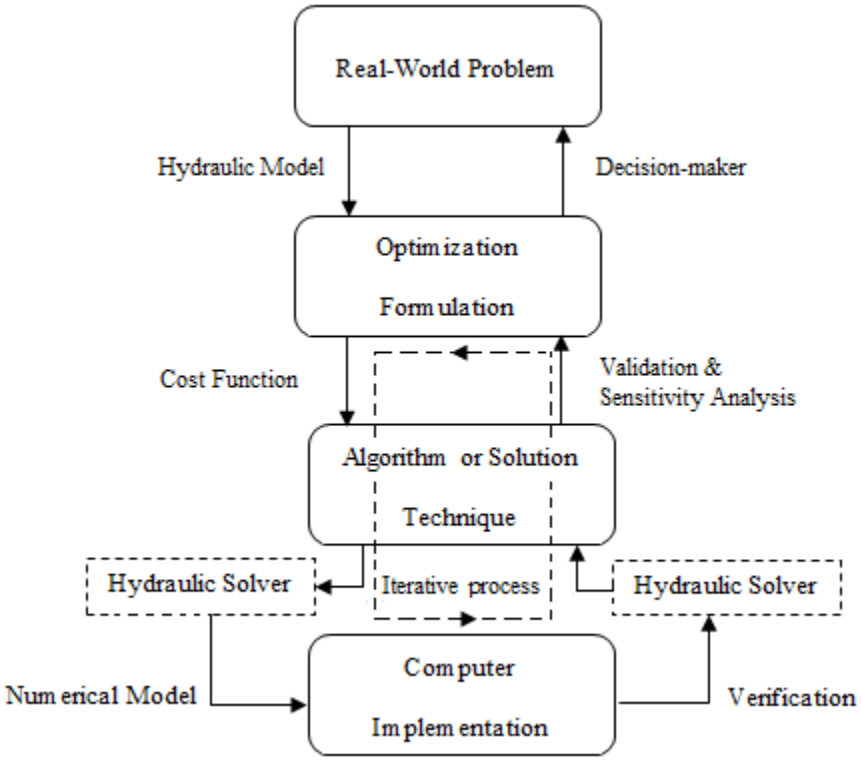


Figure 3.28: Flowchart of the Optimization Process

3.6.2 EPANET Software

In this research EPANET, hydraulic simulation software, is chosen to solve and simulate the response of network models. EPANET is developed by the water supply and water resources

division (formerly the drinking water research division) of the U.S. Environmental Protection Agency's National Risk Management Research Laboratory. It is a Graphical User Interface (GUI)-based computer program that runs both steady-state and extended-period simulation of hydraulic and water quality behavior for water distribution systems. A water distribution system is represented by a network model that consists of pipes, nodes, pumps, valves and storage tanks, and reservoirs. EPANET tracks the flow of water in pipes, the pressure at nodes, the water levels in tanks, and the concentration of a chemical species through the network model over the course of the simulation. Furthermore, EPANET can also simulate water age and source tracing.

Moreover, EPANET can also be used to assess alternative management strategies for enhancing water quality throughout a system. Finally, EPANET provides an integrated environment with other computer programs for editing network input data, performing hydraulic and water quality simulations, and reporting/viewing the results in different types of formats.

3.6.2.1 EPANET Modeling Capabilities

EPANET has been designed to have full-featured hydraulic solver that accurately simulates network models and effectively analyzes water quality. For the hydraulic modeling, EPANET includes the following capabilities:

- handles systems of any size (i.e., there is no limit on the size of the network to be analyzed);
- computes friction headloss using most common methods such as Hazen-Williams, Darcy-Weisbach, or Chezy-Manning;
- includes minor head losses for bends, and fittings when calculating energy losses;
- models both constant and variable speed pumps;
- computes pump energy consumption and costs;
- models most common types of valves such as shutoff, pressure reducing, pressure sustaining, pressure breaker, and flow control valves;
- considers uniform and non-uniform storage tank shapes; and
- allows multiple demand patterns at junction nodes.

3.6.2.2 EPANET Applications

Over the past two decades, EPANET has been exclusively used for research on optimizing the design, operation, or energy costs of water distribution systems. Samples of this work reported by many researchers to design and operate water distribution systems are presented as follows:

Manuel et al. [205], considered a disaggregated (dual-level) methodology that linked a Strength Pareto Evolutionary Algorithms II (SPEA II) with EPANET in order to satisfy the explicit system constraints and thereby enrich the Pareto-optimal set with more feasible solutions.

Their methodology in the pump scheduling problem is able to consider complex network instances. Moreover, a feasibility handling technique based on the dominance criteria is used instead of penalty functions. Their proposed method is tested and an assessment of the results is carried out by means of empirical attainment surfaces.

Fred et al. [206], used a simulated annealing algorithm linked with EPANET for obtaining an optimal operation for two networks that are the northwest pressure zone in the Austin, Texas and the North Marin Water District, Novato, California system, in terms of water quality and hydraulic performance. From their work, a total of 36 pumps operations are considered and it is found that the water quality violations as well as energy costs are reduced.

Elad et al. [207], used a new model for optimal operation of water distribution systems under unsteady water quality conditions referred to as Optimization using Genetic Algorithms with EPANET (optiGA). The new model has been derived using the concept of multi-quality water distribution system (MWDS) (i.e., initially proposed by T. Liang et al. [208]). Their optimal operation strategy included treatment facilities, time dependent electrical tariffs, control valves, and hydrological constraints. The simulation in their work is carried out for a period of 24 h. The model is explored through two example applications, example 1 based on Ostfeld et al. [209], and example 2 based on EPANET Network Example 3, [210]. The feasible solutions obtained by optiGA for the optimal operation problem are summarized by the total costs of the electricity as well as water treatment, and purchase.

Ostfeld [211], used EPANET to perform an extend time simulation for Three water distribution systems in Austin - Texas, North Marin Water District - Novato, and California system. EPANET is used to track the violation of the system and bounding constraints.

Boulos et al. [212], presented a new management model, referred to as H2ONET Scheduler, for determining the cost of a pump operation in a water distribution system. The developed optimal model makes use of the advantages of genetic algorithm optimization with a quasi-dynamic hydraulic network solver to produce the resulting operational policies over a 24-hour simulation period. The proposed operational model is tested and verified on different large-scale water distribution networks. The results of the proposed model showed a significant reduction in the cost of energy consumed for a pump operation.

3.7. Saskatoon West Water Distribution System

Saskatoon West water distribution system (WDS) is owned and operated by Sask-Water, which is Saskatchewan's Crown water utility. The operation of this network includes:

1. Intake of water from rivers, and
2. Supply and convey water to farmers, villages, residential areas, and large industrial users near the City of Saskatoon.

Sask-Water provided operational data on the Saskatoon West WDS. In 2006, this network delivered 909 million gallons of water to five major customers across a 36-kilometer pipeline. The network is perfectly sized and manageable for conducting research experiments. It contains all the features and problems associated with water supply management in terms of having: five pumps in two pumping stations (one is an auxiliary pumping station for emergency supply). Moreover, it has four reservoirs (i.e., elevated and grounded), zones with treated and untreated water at steady state pressures ranging from 20 to 220 psi, pipelines that operate at a 2% water loss in the rural and 17% in the urban sections, and a range of users on the network that include heavy industries.

3.7.1 Network Layout

Saskatoon West WDS infrastructure, like any other rural distribution system, consists of arterial pipes, junction nodes, control valves, reservoirs, tanks, primary lifting pumping stations (Queen Elizabeth (QE) and Aurora stations), hydrants, main-line meters, service connections, and backflow preventers at QE. A simple schematic diagram of this system is shown below.

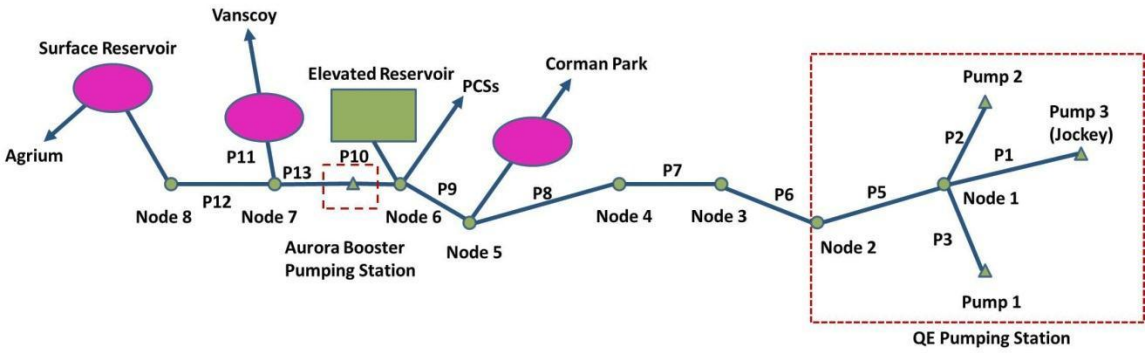


Figure 3.29: Simple Schematic Diagram of Saskatoon West WDS

Saskatoon West WDS supplies water to five major clients as follows: (see Table 3.5)

Table 3.5: Major Clients in Saskatoon West WDS

Client Name	Customer Type	Supplied Water
R. M. of Corman Park - Cedar Villa	Residential	Raw water
PCS - Cory (Mining Company)	Industrial	Raw water
PCS - Cory R & D	Industrial	Raw water
PCS Cogen (SPI / ATCO)	Industrial	Raw water
Village of Vanscoy	Residential	Treated water
Agrium	Industrial	Raw water

The Saskatoon West WDS contains 169 pipes, 153 junction nodes, 6 pressure reducing valve, one treatment plant, four elevated and grounded storage reservoirs, 5 variable-speed pumps located at QE and Aurora pumping stations. Table 3.6 provides a summary of the network's elements.

Table 3.6: Types and Characteristics of Elements in Saskatoon West WDS

Elements	**Characteristics**				
Pipe	**Number**	**Diameter (mm)**	**H-W**	**Material**	**Length (m)**
	169	81-1000	100-140	Steel, PVC	5-1951
Junction	**Number**	**Demand (IGPM)**		**Elevation (m)**	
	153	2.8-1000		472-521.4	
Valve	**Number**	**Type**		**Setting**	
	6	PSV		0	
Tank	**Number**	**Volume (m^3)**		**Elevation (m)**	
	5	143-15300		0-6	
Pumps	**Number**	**Efficiency %**	**Curve**	**Speed**	**Max. Flow (IGPM)**
	5	78 - 81.5	1,2,3	VFD	465-930

Saskatoon West WDS is served by the QE and the Aurora pumping stations. In the QE pumping station, two duty pumps (pump 1 and 2 are identical) connected in parallel as shown in Figure 3.30. The third one, called the jockey pump, is turned ON only under emergency conditions.

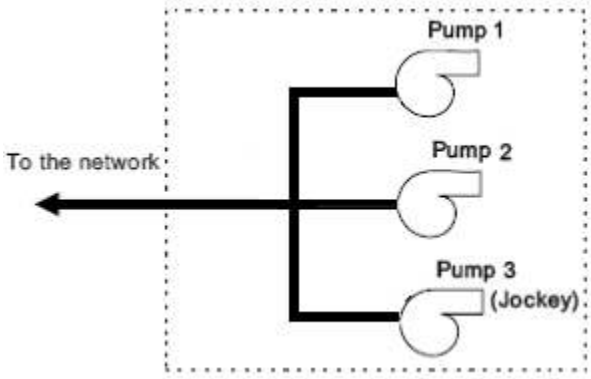


Figure 3.30: Simple Schematic Diagram of Pumps Connections in QE Station

Either one, two, or three pumps may be operated at any time interval. The pumps are connected in parallel, and their resulting performance curve can be obtained by adding their flow rates at the same head (i.e., constant pump head). Figure 3.31 illustrates the pump duty curves.

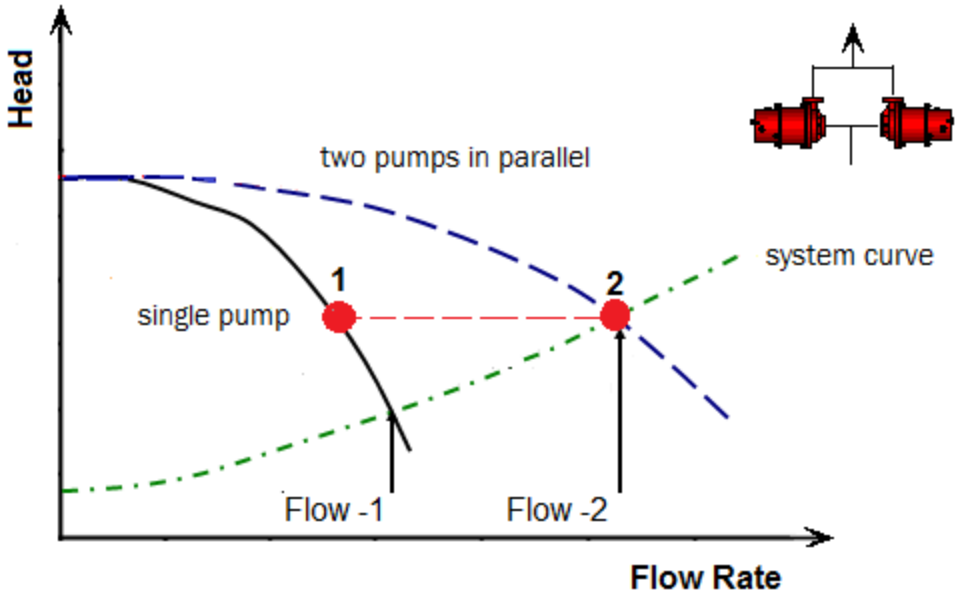


Figure 3.31: Performance Curve for Two Pumps in Parallel

Figure 3.32 shows the two pumps at QE pumping station in cases where single or two pumps are on duty.

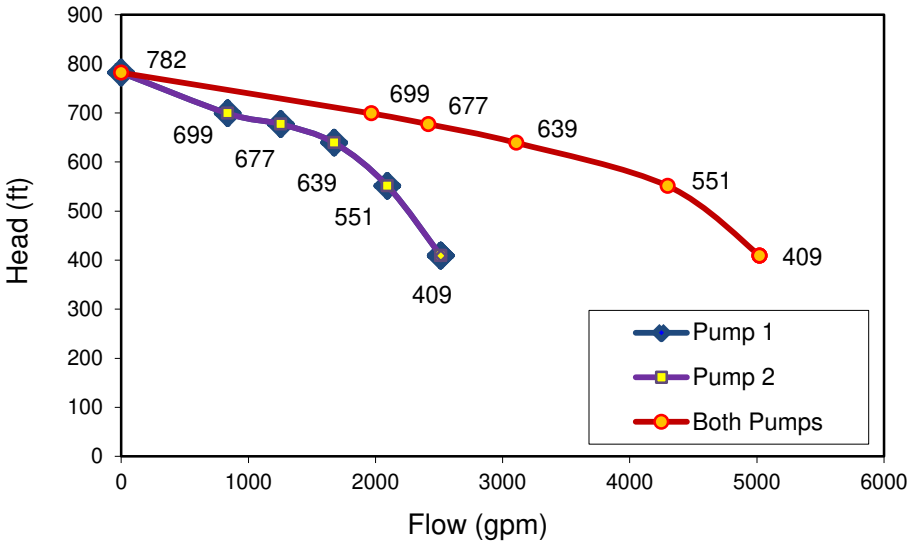


Figure 3.32: Queen Elizabeth – Parallel Operation of Pumps 1 and 2

The specification, data sheet for the pumps in the QE pumping station are given in Tables 3.7 and 3.8. The characteristic curves for the duty pumps and jockey are shown in Figures 3.33 and 3.34.

Table 3.7: Specification of Main Duty Pump in QE Pumping Station

Category	Pump Information
Model	Could Model 8x 12 R JHC
Type	Vertical Turbine
No. of Stages	8
Pump Capacity	1107.8 USGPM @ 525 ft
Efficiency	82%
Maximum head pressure	632 ft (273 psi)

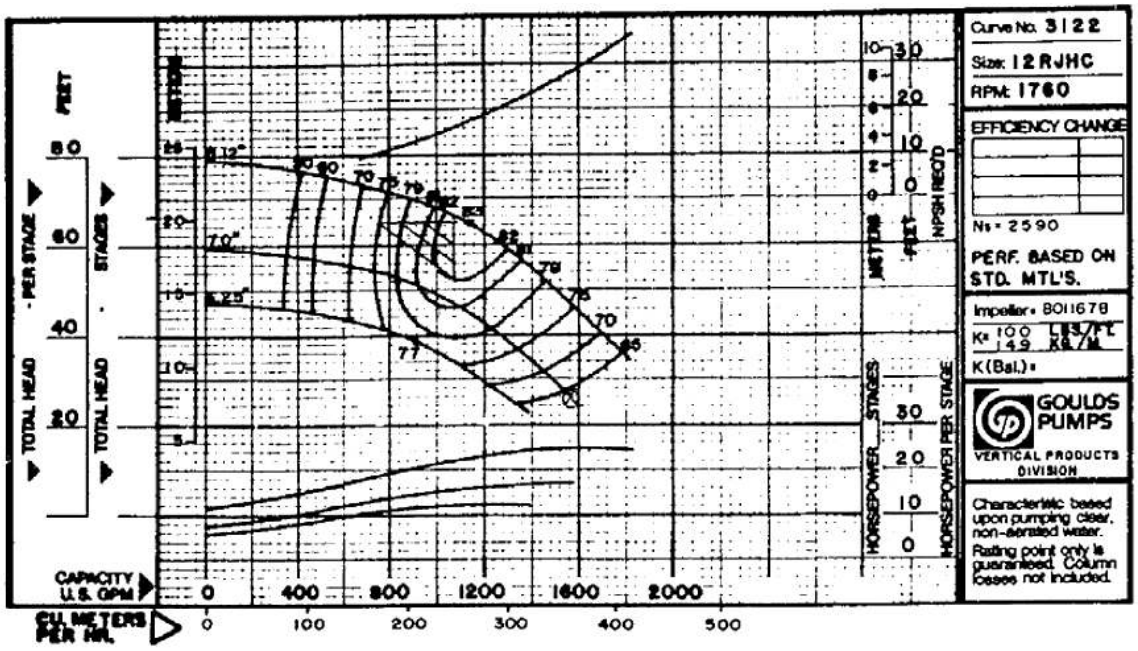


Figure 3.33: Characteristic Curves of the Main Duty Pumps in QE Pumping Station

Table 3.8: Specification of Jockey Pump in QE Pumping Station

Category	Pump Information
Model	Could Model 6x 10 R AHC
Type	Vertical Turbine
No. of Stages	6
Pump Capacity	238 USGPM @ 300 ft
Efficiency	87%
Maximum head pressure	311 ft (130 Psi)

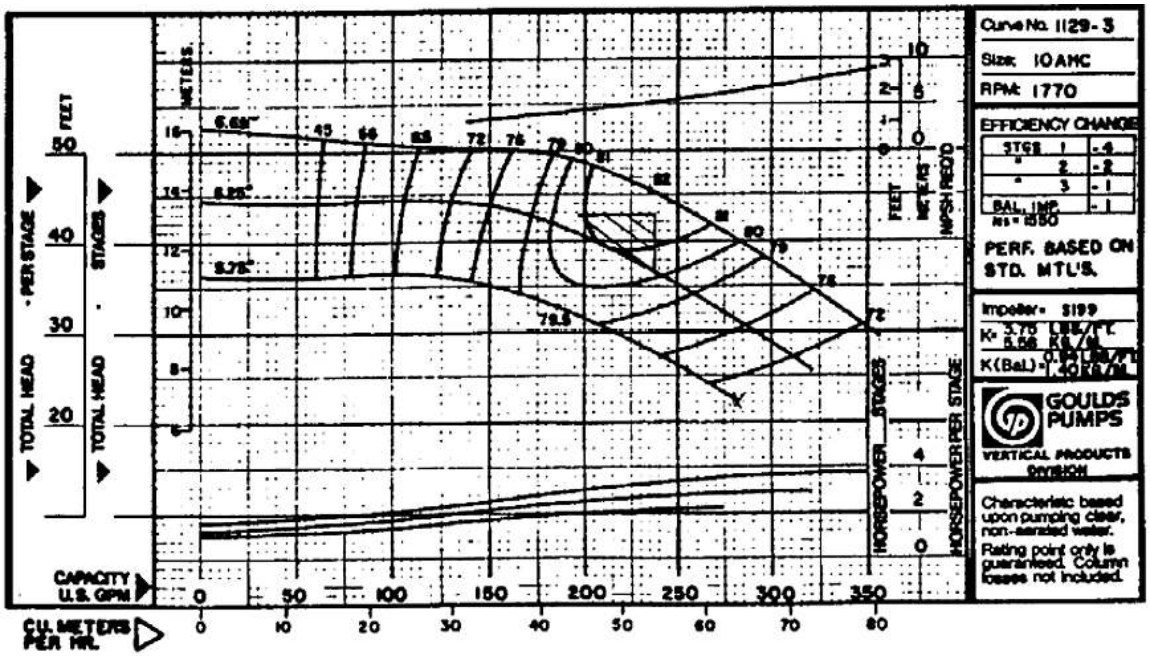
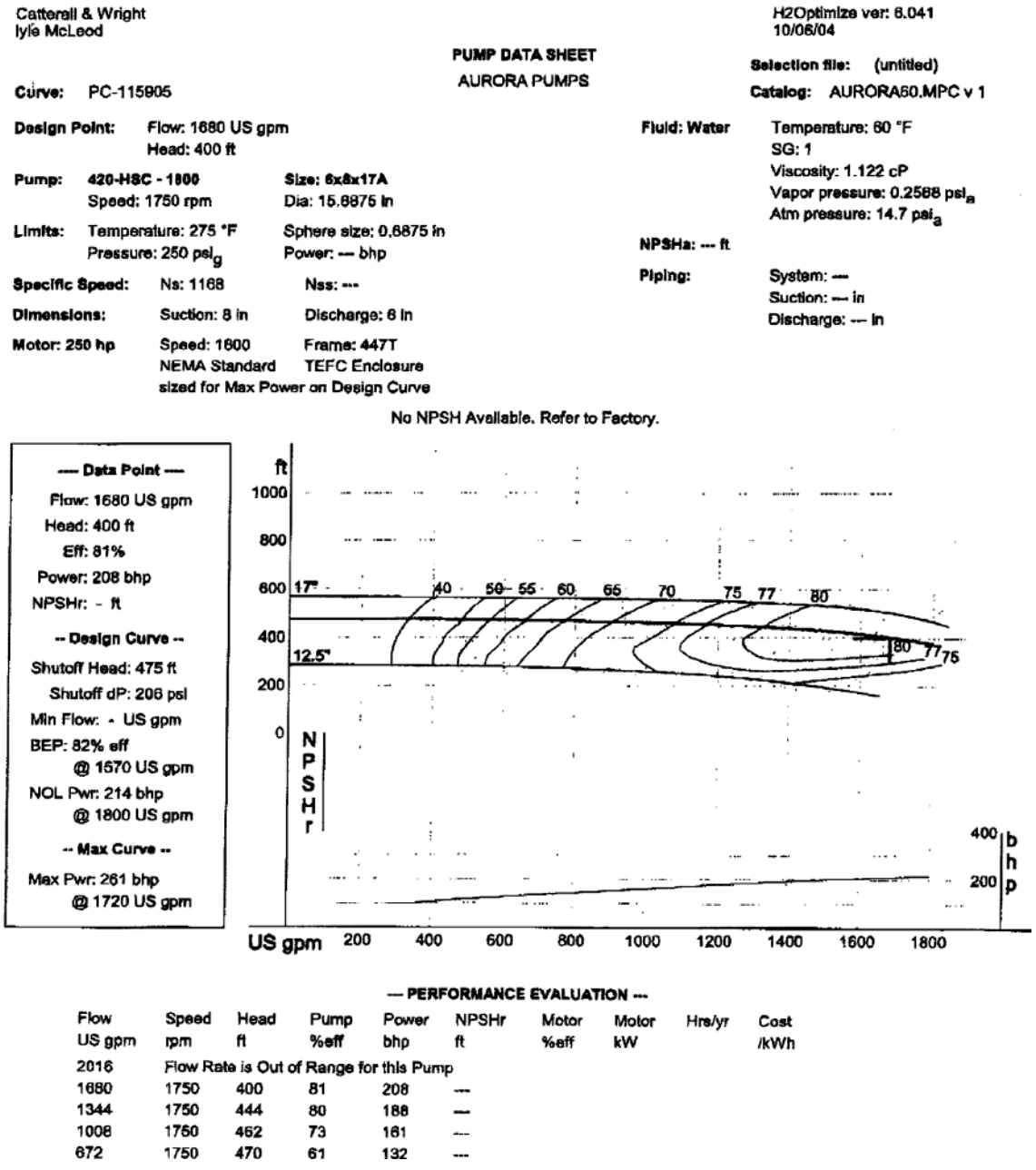


Figure 3.34: Characteristic Curves of the Jockey Pump in QE Pumping Station

At the Aurora boosting station, two pumps are installed; one is a duty pump and one is standby. This is used to add more head pressure to the water to serve clients that are either distant or at higher elevation. The data sheet and the characteristics curves of the Aurora's duty pump are given in Table 3.9 and Figure 3.35.

Table 3.9: Specification of Main Duty Pump in the Aurora Pumping Station

Category	**Pump Information**
Model	Could Model 6x 10 R AHC
Type	Vertical Turbine
No. of Stages	6
Pump Capacity	1680 USGPM @ 400 ft
Efficiency	82%
Maximum head pressure	475 ft (204 Psi)



--- PERFORMANCE EVALUATION ---

Flow US gpm	Speed rpm	Head ft	Pump %eff	Power bhp	NPSHr ft	Motor %eff	Motor kW	Hrs/yr	Cost /kWh
2016	Flow Rate is Out of Range for this Pump								
1680	1750	400	81	208	---				
1344	1750	444	80	188	---				
1008	1750	462	73	161	---				
672	1750	470	61	132	---				

Figure 3.35: Characteristic Curves of the Main Duty Pumps in Aurora Pumping Station

The estimated pipeline parameters for this simplified network are derived from the data provided by Sask-Water and presented in Table 3.10.

Table 3.10: Estimated Pipeline Parameters

Pipe	Length (m)	Diameter m (in)	Roughness Coefficient	Minor Loss Coefficient
1	3.5	0.295(12.75")	120	1.2
2	3.5	0.295(12.75")	120	1.2
3	2	0.295(12.75")	120	0.8
4	2	0.295(12.75")	120	0.8
5	39	0.295(12.75")	120	3.6
6	735	0.295(12.75")	120	0
7	30	0.387(16")	120	2.98
8	6951	0.387(16")	120	0
9	4147	0.387(16")	120	0
10	7	0.387(16")	120	0
11	13902	0.295(12.75")	120	0
12	10874	0.295(12.75")	120	0

The elevation of the junction nodes for the simplified network is given as follows:

Table 3.11: Junction Node Elevation

Junction Node	Length (m)
1	479
2	479
3	477
4	483
5	503
6	505
7	521
8	504

Moreover, there are four reservoirs in this network; one is elevated and three reservoirs are grounded (i.e., surface), and the water from these reservoirs only served their owners. The estimated capacity of the four reservoirs is given in Table 3.12.

Table 3.12: Estimated Reservoirs' Capacities in Saskatoon West WDS

Reservoir	Owner	Volume (m^3)	Elevation (m)
1	Corman Park	143	Surface
2	PCS Cory	1135	6
3	Village of Vanscoy	416	Surface
4	Agrium	15300	Surface

3.7.2 Network Demand Profile

In 2006, Saskatoon West WDS supplied a total of 909 million gallons of water at an annual electricity cost of approximately $ 300,000. The total water demand in 2006 per customers is tabulated in Table 3.13, while the variation of total water demand per month is illustrated graphically in Figure 3.36.

Table 3.13: Annual Customers Total Demand (2006)

Customer	2006 Demand in Millions of Gallons	
Agrium	264.1410	(29.05%)
PCS - Cory Division	291.3929	(32.04%)
PCS - Cory R&D	19.5794	(2.15%)
R.M. of Corman Park - Cedar Villa	4.3244	(0.48%)
Vanscoy, Village	11.3561	(1.25%)
SPI/ATCO	318.4337	(35.03%)
Total	**909.3844**	**100%**

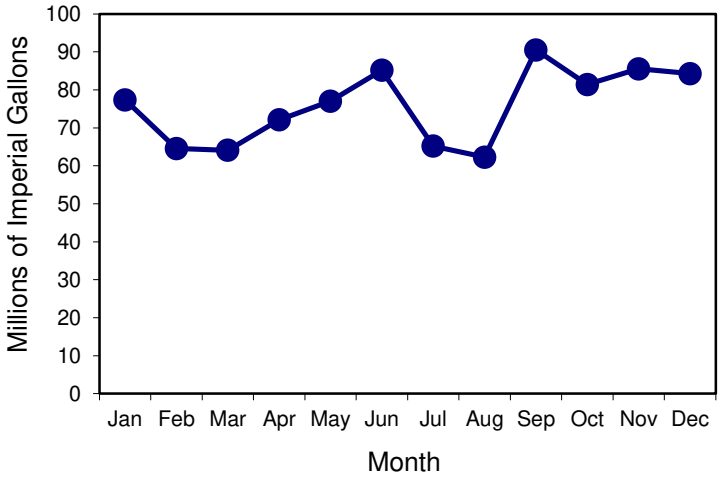


Figure 3.36: Monthly Customers Total Demand (2006)

For this research, Sask-Water provided set field-data over the period 1997 to 2006. Accordingly, the monthly-metered water supplied from the QE pumping station and the daily-metered water for the major users (i.e., water measured on their sites) are used for the implementation of a new energy optimization strategy. Summaries of the field-data (i.e., measured in 2006) are shown in the following figures, including Figure 3.37 that depicts the total discharge flow from the QE pumping station, Figure 3.38 that illustrates the total demand profile for the major clients, Figure 3.39 that provides the tanks profiles and the water losses based on the total supplied water, and Figure 3.40 that shows the total water demand over the period 1997 - 2006.

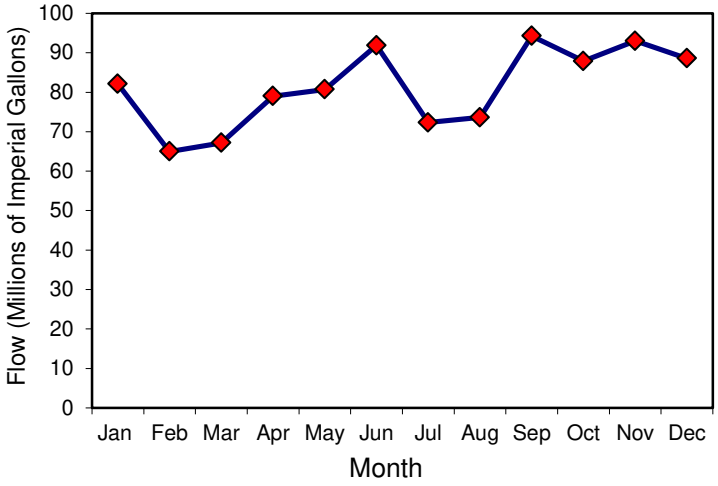


Figure 3.37: Queen Elizabeth - Total Supplied Profile (2006)

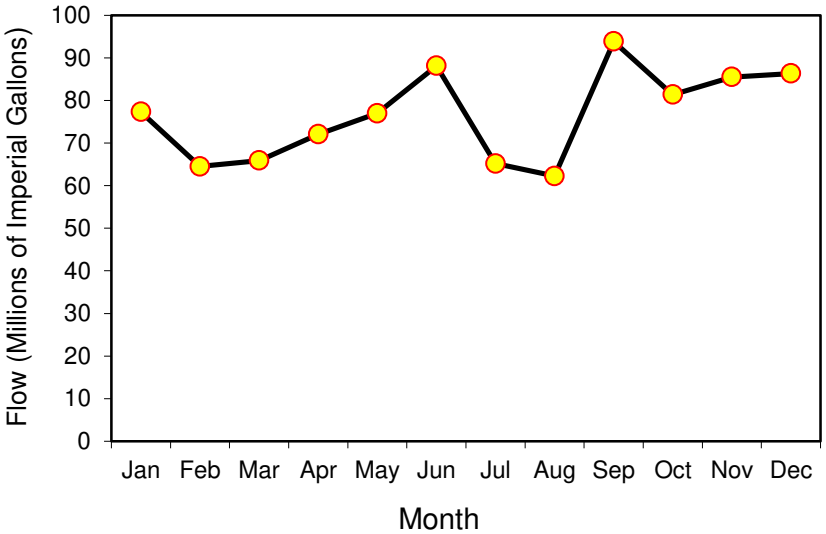


Figure 3.38: Total Consumption Profile (2006)

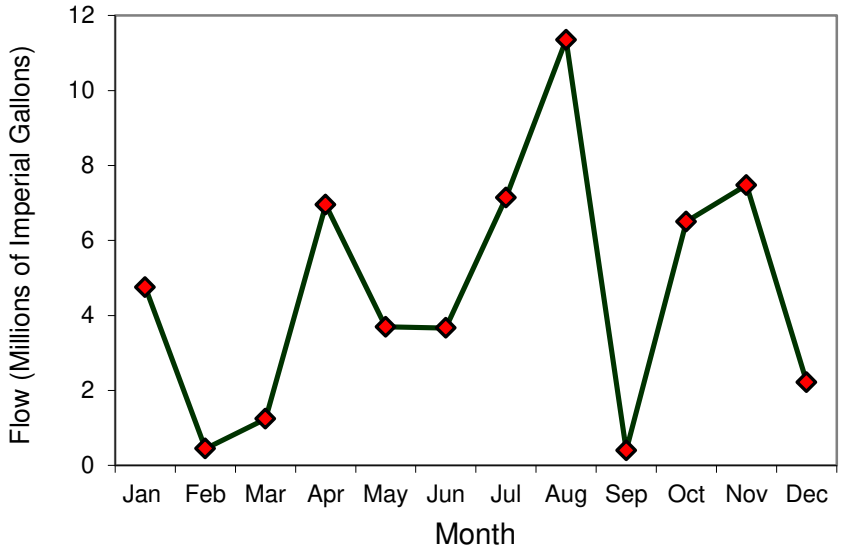


Figure 3.39: Tanks Profiles + Losses (2.8% of the Total Supplied Profile in 2006)

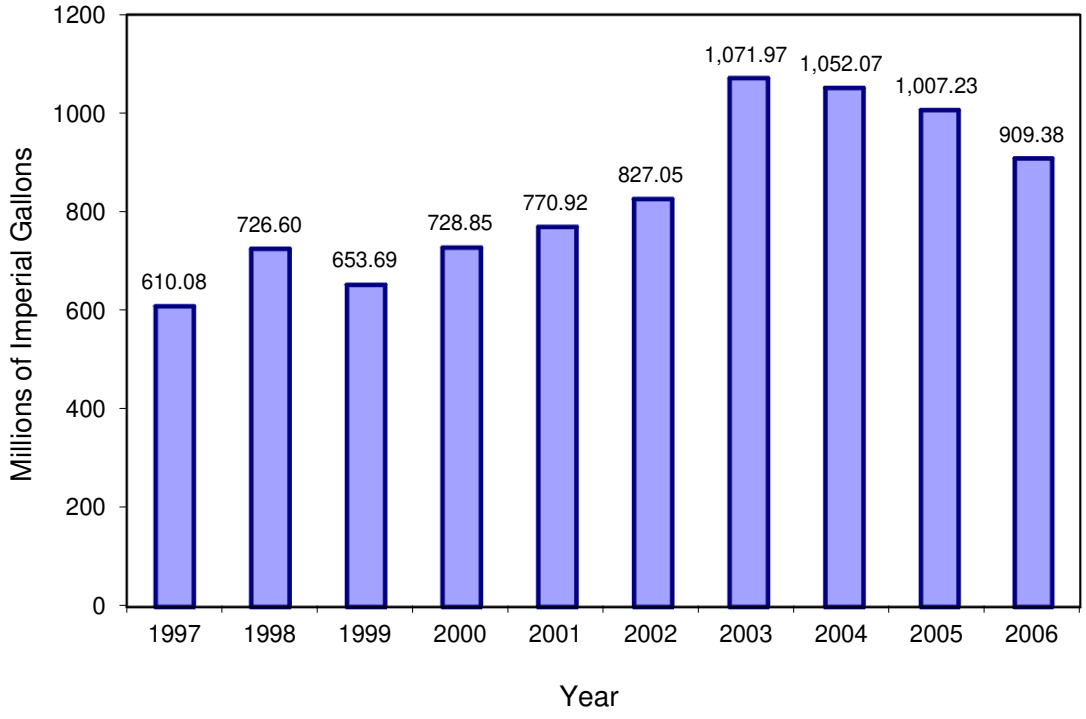


Figure 3.40: Saskatoon West WDS –Total Water Demands (1997 - 2006)

3.7.3 Network Settings

The current operating conditions at the Queen Elizabeth (QE) pumping station and the overall systems are:

- QE station is programmed to maintain a pressure within the range of 180-220 psi,
- The pumps come ON based on pressure and turn OFF based on flow,
- When no pumps are ON and the pressure drops below 180 psi, pump 1 comes ON, when pump 1 is ON and the pressure drops below 180 psi, pump 2 comes ON and the two pumps balance the load. Once the flow drops to 1500 IGPM, pump 2 turns OFF and when the flow drops to zero, pump 1 turns OFF,
- Duty pumps are classified as:
  1. "*lead*" pump, pump 1 is the first to come ON, and
  2. "*lag*" pump, pump 2 is the second to come ON.

3.7.4 Network Simulation Results

The Saskatoon West WDS is simulated using two well-known hydraulic simulations software: WaterCAD[9] and EPANET. Steady-State (SS) simulation and Extend-Period Simulation (EPS) are performed. The goal of developing simulation model (i.e., hydraulic model) of the system is to potentially obtain an accurate hydraulic analysis and accurate evaluation for operating scenarios, energy consumption, and water quality. Furthermore, by varying the parameter values, the system modeler can obtain valuable insight into system performance, from which one can study the behavior of the system under a wide range of diverse circumstances (i.e., system and environment conditions).

3.7.4.1 Case Example

The features of the hydraulic model are given in Figure 3.41, and the simulation results of this model are obtained using the following time setup:

[9] In Saskatoon West WDS, all of the reservoirs are designed to be fed from the top. This part of the hydraulic modeled is only modeled using WaterCAD, and then augmented with the rest of the model using EPANET.

Table 3.14: Time Setup – Case Example 1

Times Options	
Property	**Hrs:Min**
Total Duration	168
Hydraulic Time Step	0.01
Quality Time Step	3:01
Pattern Time Step	0
Reporting Time Step	3:00
Report Start Time	0
Clock Start Time	7 am

Figures 3.42, 3.43, and 3.44 show comparisons of the nodal pressure, nodal flow, and the system flow balance for the network. The results corresponding to this case example are obtained using EPANET and measured field-data. It can be observed that the hydraulic model is able to precisely estimate the flow in links and slightly underestimate the pressures since it assumes that the nodal demands are always entirely satisfied within a range of the nodal heads. The assumption of complete satisfaction of nodal demands implies that the hydraulic model overestimates the head losses in the links and thereby underestimates the nodal pressures. The error in the nodal pressures as calculated from the hydraulic model varies from as low as 0.7% at node 1 to as high as 5% at node 8. Furthermore, it can also be observed that there are very slight errors in the nodal flow values that are obtained from EPANET, which varies from as low as 0.05% (at link 9) up to 0.32% (at links 5-8).

On the whole, the validity of the hydraulic model for Saskatoon West WDS is verified. The results obtained (i.e., computed data) using EPANET are compared with those measured from the field as illustrated in Figures 3.45 and 3.46. Moreover, these results are tabulated and presented in Tables 3.15 and 3.16.

It is proven that the use of hydraulic model for the analysis of the Saskatoon West WDS can result in very accurate estimations of both nodal heads and nodal flows. Hence, the hydraulic model is used to compute the energy costs of the system for a one-month period of time (see Table 3.17). The results are compared with those obtained from the system facility for May 2006 (as shown in Table 3.18).

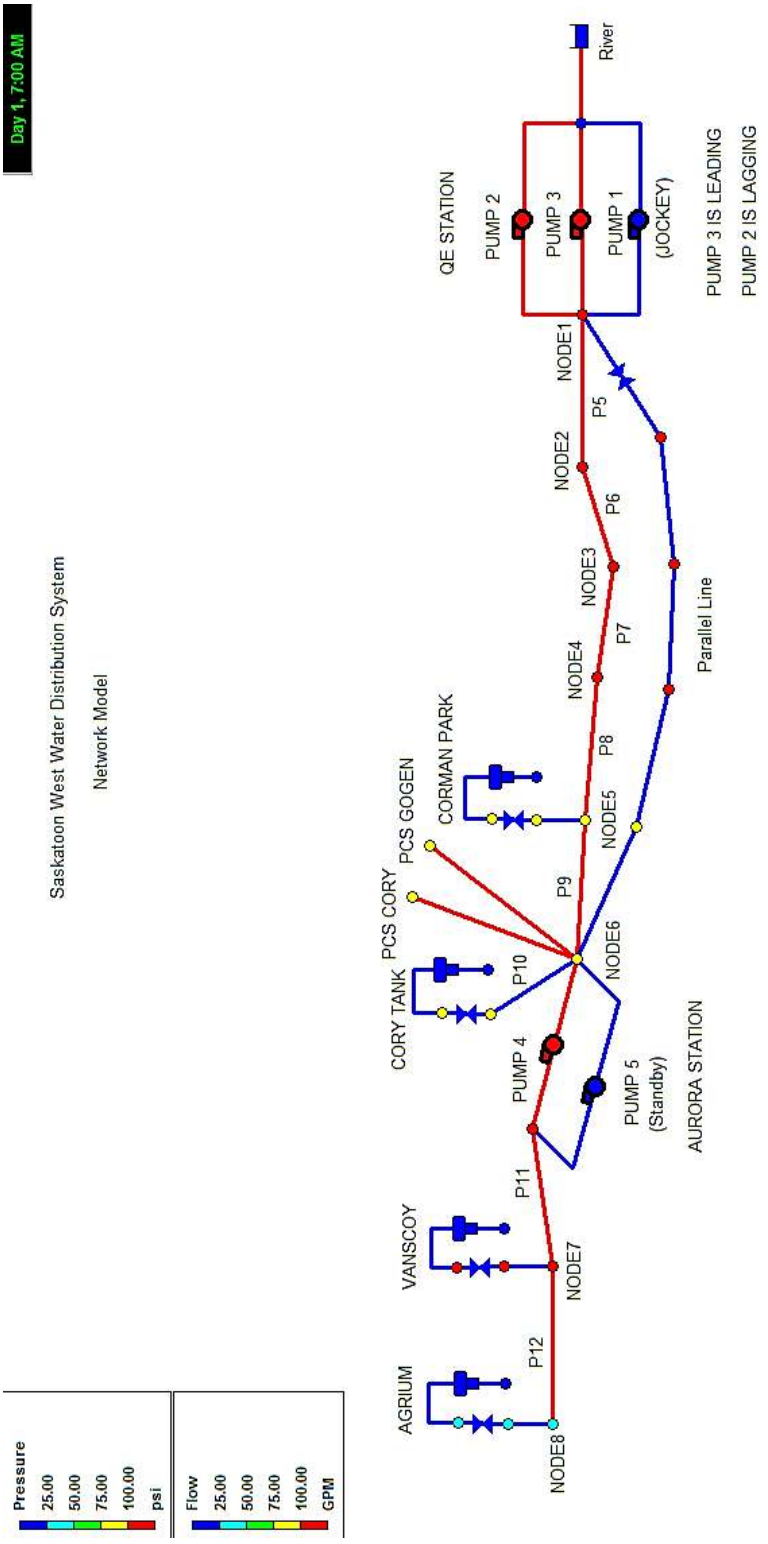


Figure 3.41: Hydraulic Model for the Saskatoon West WDS - Case Example

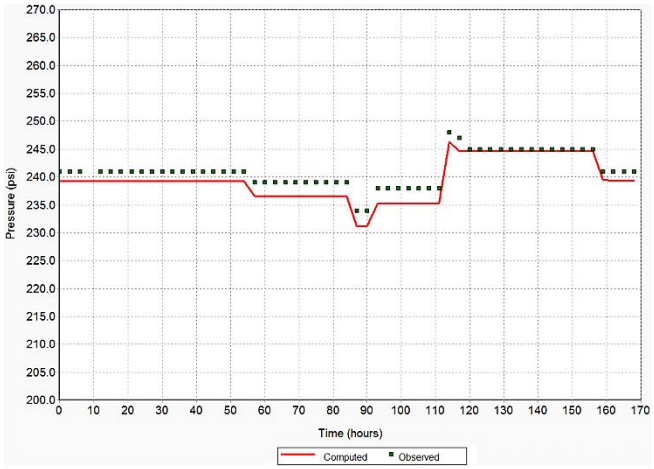


Figure 3.42: Pressure at Node 1 – Case Example

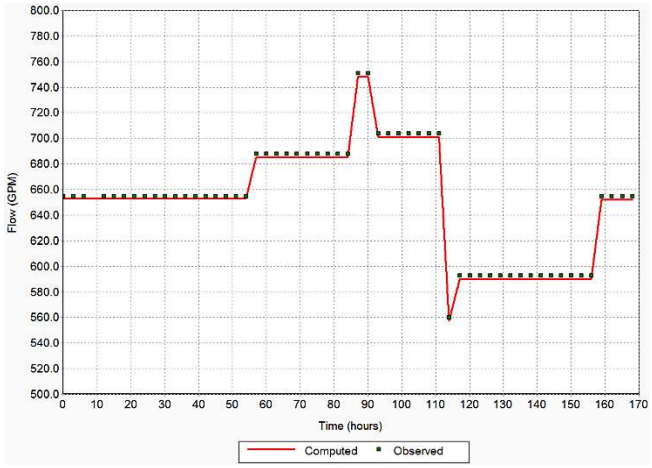


Figure 3.43: Flow at Link 11 – Case Example

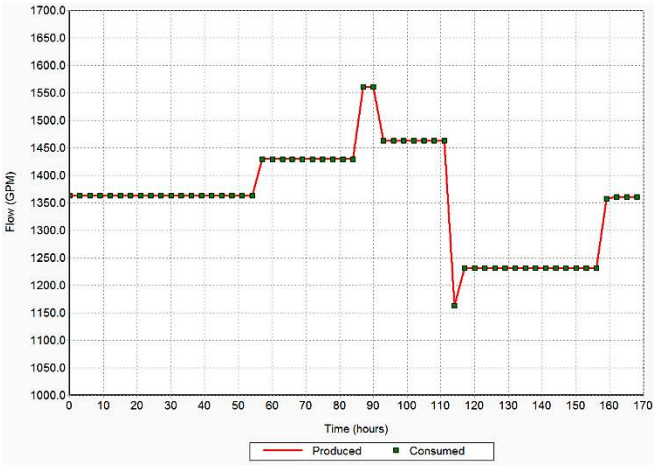


Figure 3.44: System Flow Balance – Case Example

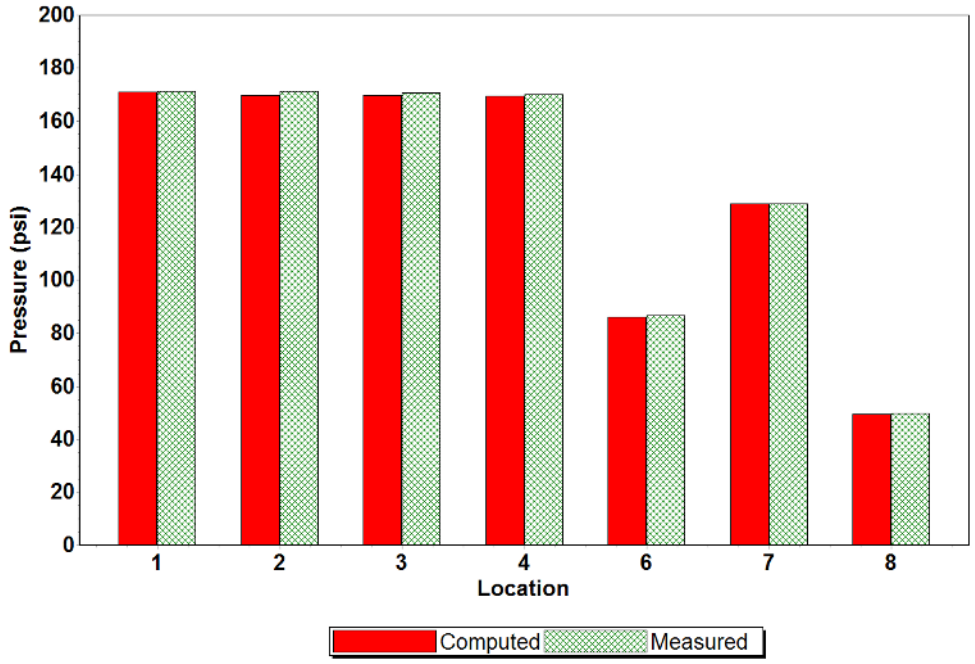


Figure 3.45: Comparison of Mean Pressure Values – Case Example

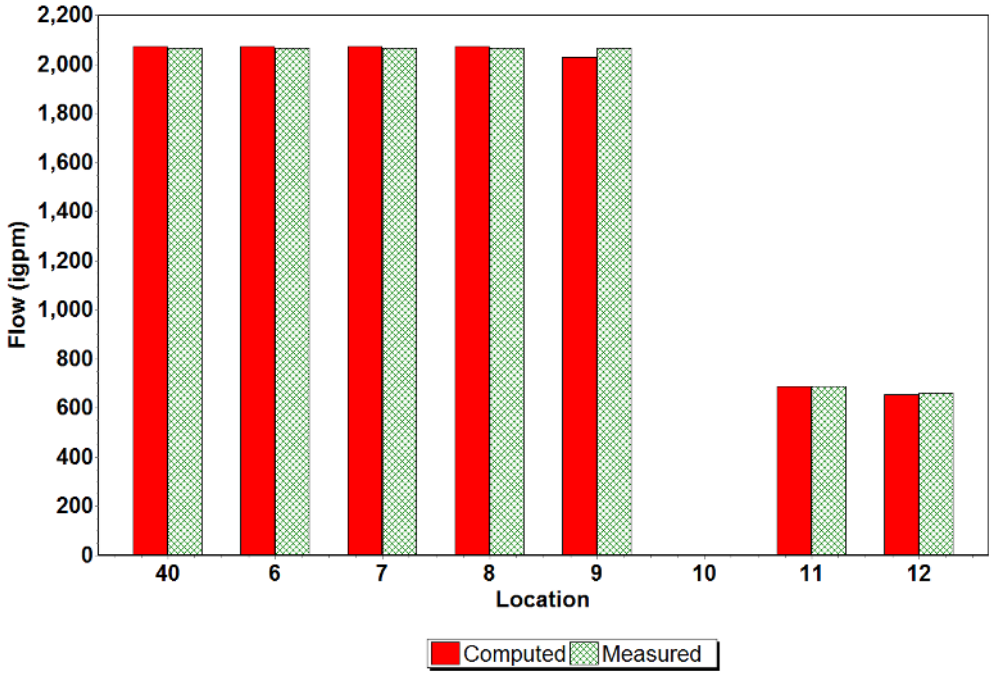


Figure 3.46: Comparison of Mean Flow Values – Case Example

Table 3.15: Calibration Statistics for Pressure – Case Example

Location	Observed	Observed Mean	Computed Mean	Mean Error	RMS Error	% Error
1	55	241.18	240.71	1.704	1.904	0.7
2	55	239.82	240.21	0.881	0.982	0.37
3	55	239.82	241.04	0.826	1.116	0.34
4	55	231.75	231.89	0.685	0.888	0.3
5	55	180.91	180.58	0.679	0.89	0.38
6	55	165.20	166.84	1.791	2.172	1.08
7	55	99.25	100.93	3.454	4.134	3.48
8	55	92.13	92.2	4.663	6.054	5.01
Network	440	186.26	186.57	1.835	2.903	0.99

* *Correlation between Means: 1.0*

Table 3.16: Calibration Statistics for Flow – Case Example

Location	Observed	Observed Mean	Computed Mean	Mean Error	RMS Error	% Error
2	55	678.96	679.59	1.57	1.87	0.23
3	55	678.96	679.59	1.57	1.87	0.23
5	55	1357.76	1359.17	4.271	5.361	0.32
6	55	1357.76	1359.17	4.271	5.361	0.32
7	55	1357.76	1359.17	4.271	5.361	0.32
8	55	1357.76	1359.17	4.271	5.361	0.32
9	55	1357.76	1359.17	0.727	0.895	0.05
10	55	0	0.01	0.012	0.012	0.0
11	55	653.22	650.64	2.577	2.613	0.4
12	55	627.98	626.28	1.703	4,591	0.27
Network	440	942.79	943.37	16.763	22.961	1.66

* *Correlation between Means: 1.0*

Table 3.17: Daily Computed Energy Costs – Case Example

Pumps	Percent Utilization	Average Efficiency	KW-hr/Mgal	Average KW	Peak KW	Cost/day CAD $
2	100	82	2429.8	98.93	105.31	193.75
3	100	82	2429.8	98.93	105.31	193.75
4	100	82	243.14	9.47	11.69	18.55
					Total cost	**406.05**
					Demand charge	**143.72**

Table 3.18: Energy Costs Comparison for Saskatoon West WDS in May 2006

Category of Cost	Simulation Results	Actual Results
Electrical cost	$ 12,181.5	$ 12,176
Demand charge	$ 4,311.6	$ 4,308.36
Basic monthly charge	$ 57.92	$ 57.92
Surcharge (10%)	$ 1,655.1	$ 1,654.23
Taxes (7%)	$ 1,274.43	$ 1,273.76
Total	**$ 19,480.55**	**$ 19,470.27**

3.7.4.2 Network Simulation: Summary of the Results

The proposed operational scenario for modeling and solving the Saskatoon West WDS have been found to accurately determine the nodal head pressure and nodal flow under different operating and system conditions. This has been verified using the popular hydraulic solver EPANET. The outcome obtained from the case example is summarized as follows:

- The hydraulic model for the Saskatoon West WDS is able to successfully demonstrate results that are very similar to the real data measured from the field, which then prove the high accuracy of it.
- The model shows very good matches between the observed profile and the computed ones in terms of energy costs, pressure, and flow profiles.
- Flow simulation demonstrates a strong correlation between both simulation and real data.
- Pressure profile shows slight differences due to ignoring valves (or fittings) connected through the network and small minor nodes that are not included in the simplified model.
- The key to successfully obtaining an accurate hydraulic model for the Saskatoon West WDS is to verify the hydraulics of the system and the rules of controlling the tank operation. This is significantly observed and studied in this research.

3.8. Conclusion

In this dissertation, a new multi-objective formulation for the Saskatoon West WDS operational problem (could be "any rural" system) is provided. The cost function is posed as minimizing the electricity costs, total number of pump switches, the cost of maximum power peak, reservoir water level variation, and free chlorine residual as five objectives associated with a set of operating constraints that specifies the lower and upper limits of the system. In the next chapter, an Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO is explained to be used as an *algorithm or solution technique* to solve the constrained multi-objective *real-world problem* in order to obtain a set of trade-off solutions that

satisfy the objectives and constraints being considered. The new formulation is brought out to improve the ability of the Saskatoon West WDS to handle any future demands and hydraulic uncertainties with higher safety and reliability.

Chapter 4

Multi-objective Optimization Algorithm

Real-world problems are usually better-formulated using multi-objective models; this means that more than one objective needs to be considered for their optimization. In this chapter, a new form of hybrid multi-objective PSO algorithm is presented that used a novel Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization algorithm, referred to as APC-MOPSO. This algorithm is then analyzed and tested by using well-known benchmark test problems. The performance of the proposed algorithm is then compared to other methods that represent the state-of-the-art in Multi-objective Evolutionary Algorithms (MOEAs) and Multi-objective Particle Swarm Optimization algorithms (MOPSOs). Computational results showed that the performance of the APC-MOPSO is robust, accurate, and computationally efficient for the benchmark test problems considered in this research. The results demonstrated that the APC-MOPOS algorithm is well suited to high-dimensional NP-hard real-world problems.

4.1 Introduction

Real-world problems are often very complex and may need to deal with constrained multi-modal multi-objective optimization problems. This has led to a growing interest in multi-objective optimization techniques that involve more than one objective function to be simultaneously optimized. Accordingly, at the end of the multi-objective optimization process, there will be more than one solution to be considered. This enables a *trade-off* of *high-quality* solutions and provides options to the decision-maker to choose a solution based on qualitative preferences.

Generally, the population-based heuristic algorithms can be divided into competitive search and cooperative search models. The competitive search model involves individuals that compete against each other for dominance in the population. As a result, individuals with high fitness will be retained, while those with low fitness will be eliminated from the population and replaced by new individuals through crossover and mutation operations. In other words, this type of search is modeled on Charles Darwin's survival of the fittest or a natural selection concept.

Competitive search models have been implemented to solve Multi-objective Optimization Problems (MOOPs). These include Evolutionary Algorithms (EA) [16,213,214]. Other competitive-based algorithms are Genetic Algorithms (GA) [4], Differential Evolutionary (DE) [215], Evolutionary Strategy (ES) [216], Evolutionary computation (EC) [217,218], Evolutionary Programing (EP) [219], and Genetic Programming (GP) [220].

On the other hand, cooperative search models involve a number of individuals working together to solve the problem. This idea is the basis of the Swarm Intelligence theory (SI) [221,222]. In this model, instead of eliminating the weak individuals (i.e., with lowest fitness) from the population, the best individuals in the population (i.e., called leaders) guide the rest of the population. Hence, heuristic algorithms based on this type are referred to as social optimizing techniques. Cooperative search algorithms have been extensively studied in the past decade and deployed to find feasible solutions for real-world problems. Common forms include a Particle Swarm Optimization (PSO) algorithm, [7], and an Ant Colony System (ACS), [223].

Since 1995 when Kennedy and Eberhart first proposed PSO [7], several modifications have been undergone on the standard PSO. Therefore, PSO algorithms have been categorized into many classes including Pareto-based PSO, Modified PSO, Clustering-based PSO, Parallel PSO, Dynamic PSO, and recently Hybrid PSO (as reviewed earlier in Chapter 2).

In this chapter, a novel Hybrid PSO algorithm referred to as Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) is proposed. This approach combines the following:

- The concept of Pareto-dominance,
- A new adaptive technique[10] for updating PSO's design parameters,
- A modified K-Means clustering,
- A new form of parallel implementation,
- A new adaptive mutation operator, and
- A new mechanism for a variable external repository size[11] (i.e., unlike the previous work where which a fixed size of external repository is usually used).

A brief introduction to the original PSO algorithm is presented in Sections 4.1. In Section 4.2, the various forms of PSO algorithms are reviewed. Section 4.3 provides a detailed description of the proposed APC-MOPSO algorithm. Selected test problems, performance metrics, and numerical results and their analysis are given in Section 4.4.

[10] In optimization methods, adaptive techniques are commonly used to dynamically adjust (adapt) the required values of the design parameters for their better performances.

[11] The size of the external repository means the number of candidate solutions that can be stored in the repository after each iteration during the search process.

4.2 Particle Swarm Optimization

4.2.1. Basic Concept and Algorithms

J. Kennedy and R.C. Eberhart [7], first developed the Particle Swarm Optimization (PSO) algorithm. PSO is a heuristic, population-based and stochastic optimization algorithm. In its original form, it is used for solving unconstrained single-objective optimization problems. The PSO notion is inspired by the concept of the Swarm Intelligence (SI) that is naturally observed in a swarm of insects, a flock of birds, or a school of fish when searching to find food[12].

This natural journey in search of food demonstrates two forms of intellectual behavior. The first is the individual behavior where each member utilizes its own best experience to draw its next flight direction. The second is the collective behavior that demonstrates the social behavior of the swarm and the influence of the leaders in guiding the rest of the members towards a common target.

PSO implements search on both an individual as well as a communal level to find an optimal set of solutions in the objective space. In PSO, the search process is started by initially considering the position of individuals, referred to as particles, in the population, referred to as a swarm, as potential solutions to the given problem. Then, the particles fly or travel through the search space, looking for new and/or better solutions to the problem. To illustrate the concept of the searching process, Figure 4.1 depicts the concept of updating a particle's position in the search space, and Figure 4.2 shows the underlying notion of the searching process in which each particle alters its current position using its personal as well as the swarm's best information.

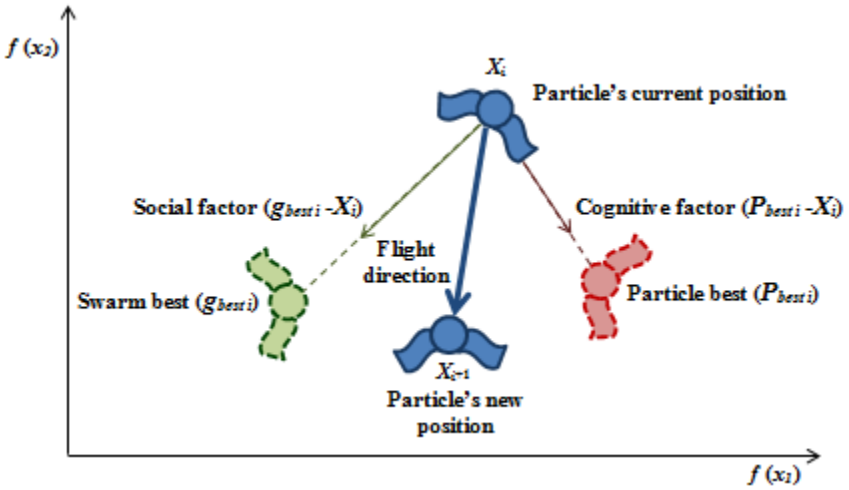


Figure 4.1: Updating Particle's Position in the Search Space - PSO Algorithm

[12] In SI theory, a flock, a swarm and a school are defined as groups under the guidance of leaders.

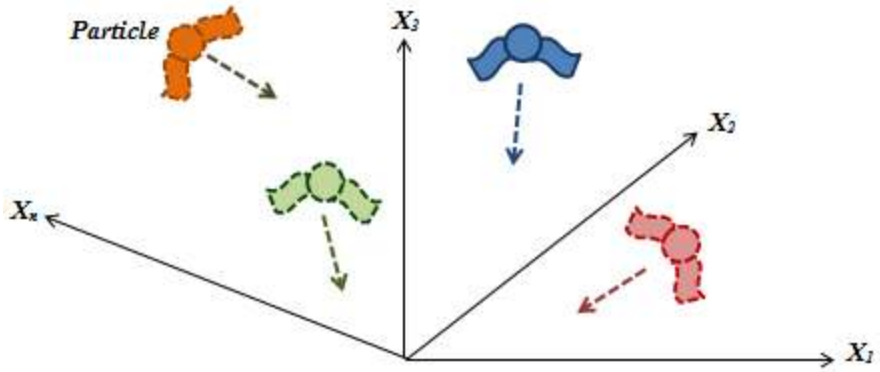


Figure 4.2: The Concept of Exploring for the Particles in the Search Space - PSO Algorithm

4.2.1.1 Basic PSO terminologies

The definitions commonly used in the PSO literature are as follows:

- *Particle:* is a candidate solution or individual or a member of a set of solutions to the problem being optimized (i.e. often initialized randomly).
- *Swarm:* is a group of particles that represents the population of the algorithm.
- *Leader:* a particle that is chosen according to a certain criterion and responsible to lead other particles in the swarm towards the global regions.
- p_{best} *(particle best):* the best position (i.e. best position means the closest point to the global optimum) ever visited by a particle during its flight in the search space.
- l_{best} *(local best or neighborhood best):* the best position of a particle found in the neighborhood around a given particle so far.
- g_{best} *(global best or swarm best):* the best position amongst particles in the swarm so far.
- *Velocity (v):* is a particle's flight velocity (i.e. rate of the positional change of the particles). The velocity is updated at every iteration along the optimization process.
- *Position (x):* is the location of a particle during the optimization process.
- *Inertia weight factor (w):* a design parameter used to balance between the impacts of the particle best position (p_{best}), the swarm best position (g_{best}), and the neighborhood best (l_{best}) in determining the velocity of a new flight trajectory.
- *Cognitive factor* (C_1)*:* represents the successful impact of a particle's self-experience on adjusting its new flight direction towards its best previous position.
- *Social factor* (C_2)*:* represents the impact of the swarm best position (g_{best}) on the flight directions of other particles.
- *Social topology:* represents the structure of the communication network that allows the particles in the swarm to exchange their best information.

- *Exploration:* in any population-based optimization algorithm, exploration means the temptation of an individual (particle) in searching for or discovering new regions of the search space that have not been visited before.
- *Exploitation:* means the process of improving and combining the traits of the currently known individual(s) with the aim of gaining the utmost benefit from them.
- *Convergence:* in any optimization algorithm, the search process is said to have converged *if and only if* there will be no further improvements in that particle's position.
- *Premature Convergence:* in any optimization algorithm, the search process is said to have prematurely converged into a local optimum *if and only if* there will be no further exploration of new areas of the decision space while there exists another region that contains better solutions [31,32]. Figure 4.3 shows an example of premature convergence as the search process is trapped into a local minimum.

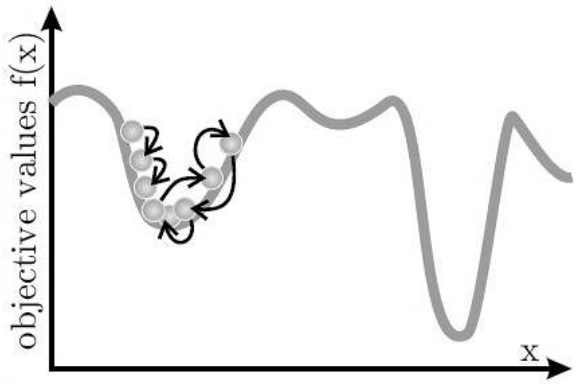


Figure 4.3: Example of Premature Convergence in Minimization Function [33]

- *Preserving Diversity:* while solving a multi-objective problem with any population-based optimization algorithm, one of the main goals that should be taken under consideration is that of preserving a set of non-dominated solutions that are widely dispersed or diversified. Preserving diversity can be directly related to maintaining a good balance between exploration and exploitation, [34]. Related selected publications on how diversity can be measured, can be found in [35,36,37,34,38].

4.2.1.2 Original (Single-objective) Particle Swarm Optimization

The original Particle Swarm Optimization (PSO) is a single-objective and stochastic search engine that exhibits a powerful ability in finding optimal (best) solutions for complex, non-linear and non-convex functions.

PSO algorithms have greatly evolved with modifications that have improved their convergence rate, complexity, exploration ability, and search quality. The most famous and recognized version of the Single-objective PSO, referred to as SOPSO, is proposed by Shi et al. [40]. In their work, the inertia weight factor (w) is introduced in order to adjust the balance between exploration (quantitative solutions) and exploitation (qualitative solutions). In other words, w plays the role of balancing between the global and local searches in the objective space. It is found that a greater w resulted in more iteration before convergence. Shi et al. used two equations that govern a particle's movement or travel through the search space, namely its velocity and position at each time step, as given in Eqs. (4.1) and (4.2):

$$\vec{v}_{t+1} = w \times \vec{v}_t + c_1 \times r_1 \times (\vec{p}_{ibest} - \vec{x}_t) + c_2 \times r_2 \times (\vec{p}_{ig} - \vec{x}_t) \quad 4.1$$

$$\vec{x}_{t+1} = \vec{x}_t + \vec{v}_{t+1} \quad 4.2$$

Eq. (4.1) is made of three terms. Its first term represents the particle's velocity in the previous iteration. Its second term is called the "cognitive" component, and represents the self-exploring experience of each particle by which the particle is encouraged to move towards its own best position found so far. The third term is known as the "social" component, and it represents the collective influence of the swarm in the particle's flight direction. The detailed outline of the SOPSO algorithm is provided in the following Pseudo-code 10, [40].

Pseudo-code 10: Original PSO Algorithm for Single-objective Optimization Problems

Initialize: swam size (N), maximum iterations (*Maxiter*), iteration counter (*iter* = 1), inertia weight factor (w)
Task: find an optimal solution
for i = 1: N
DO
Initialize randomly the particle velocity v_i
Initialize randomly the particle position x_i
end for i
Set the particle best (p_{best}) vector according to the current positions
Set the swarm best (g_{best}) vector according to the current positions
Repeat
for i = 1: N
DO
Assess the fitness function for particle i
if (particle i current position $\preccurlyeq$ $p_{best,\,i}$)
$p_{best,\,i}$ = particle i current position
end if
if ($p_{best,\,i} \preccurlyeq g_{best,\,i}$)
$g_{best,\,i} = p_{best,\,i}$

end if
end for i
for i = 1: *N*
DO
Update particle *i* velocity using Eq. (4.1)
Update particle *i* position using Eq. (4.2)
end for i
iter = *iter* + 1
if (*iter* ≤ *Maxiter*) or (stopping criteria in NOT satisfied)
Return
end if
Obtain the optimal global solution

4.2.1.3 Standard Multi-objective Particle Swarm Optimization

Many modifications and improvements have been proposed since the introduction of the Particle Swarm Optimization (PSO) algorithm in 1995. Most of these modifications are essential to the application of the PSO to constrained, complex, and multi-objective sets of problems. Multi-objective versions of the PSO (MOPSO) have been deployed in many research and application areas.

The most well-known version of Multi-objective PSO (MOPSO) developed by Reyes-Sierra et al. [81]. In their work, the concept of Pareto-dominance complemented by a new ranking scheme, an efficient mechanism for handling constraints, a mutation operator, and finally the notions of the external repository are incorporated. These proved very effective for solving Constrained Multi-objective Optimization Problems (C-MOOPs). At every iteration, Pareto-dominance is used for classifying the solutions as dominated or non-dominated. Next, a copy of all non-dominated solutions is stored in the external repository to enhance the quality of the algorithm's exploration and to preserve diversity (i.e. have solutions along most of the global areas). The MOPSO process is outlined in Pseudo-code 11, [81].

Pseudo-code 11: Standard MOPSO Algorithm

Initialize: swarm size (*N*), maximum iterations (*Maxiter*), iteration counter (*iter* = 0), inertia weight factor (w), cognitive coefficient (C_1), social coefficient (C_2), mutation probability factor (p_m), repository size (*Rep*), upper and lower bounds of each dimension (U_b, L_b)
Task: find Pareto-front set
for i =1:*N*
DO

Randomly initialize the position within the feasible region
Randomly initialize the velocity within the feasible region (i.e. in the range of [L_b, U_b])
end for
Randomly initialize the leaders in the repository with size (*Rep*)
Randomly initialize the leaders for the swarm
Repeat
for i = 1:N
DO
Select leaders
Update particle i velocity (Flight) using Eq.(4.1)
Update particle i position using Eq.(4.2)
Evaluate the fitness functions
Apply Mutation
Update p_{best}
end for
Update leader in the repository
Update leaders for the swarm
$iter = iter + 1$
if ($iter$ < *Maxiter*) or (stopping criterion is NOT satisfied)
Return
end if
Obtain the results

4.3 Adaptive Parallel Clustering-based MOPSO Algorithm

Three major concerns in Multi-objective Optimization Problems (MOOPs) that also apply to MOPSO algorithms are:

1. Premature convergence (i.e. trapped into local optima),
2. Diversity (i.e. well-distributed non-dominated solutions), and
3. Quality guarantees (i.e., how far the non-dominated solutions are from a global optimum (Pareto-optimal front)).

The goal of this work is to propose an Adaptive Parallel Clustering-based MOPSO (APC-MOPSO) algorithm that can effectively overcome the above difficulties that exist in MOPSO algorithms. The result is a powerful optimization tool that can tackle NP-hard and complex real-world problems. In the APC-MOPSO approach, the following elements are incorporated:

1. A new dynamic model to update particles' velocity and position in the search space in order to avoid premature convergence,
2. A new external repository,

3. A new technique that dynamically adjusts the values of the design parameters[13] and the size of the external repository, and adaptively changes the values of the inertia weight factor, mutation operator, and the boundaries of the search space,
4. A modified K-Means clustering algorithm mainly to increase the quality of the selected leaders for the swarm and then to enhance the distribution of the non-dominated solutions along the Pareto-front regions,
5. A mutation operator to improve the performance of the MOPSO algorithm by providing a set of approximately optimal solutions that is very close to the Pareto-optimal front, and
6. A parallel islands model.

The APC-MOPSO algorithm is described in the following subsections.

4.3.1 New Dynamic Model for Updating a Particle Position

In most single and multi-objective PSO algorithms, particles only "remember" (i.e. keep record of) their own best position (p_{best}) and the swarm best position (or global best, g_{best}). Particles flight directions are determined and guided by the particles' self-experience and the best position found so far by the swarm during the search process. Usually, the p_{best} and g_{best} positions are far apart from one another in the search space. This adds more pressure to the particles to rapidly move towards the swarm best, often resulting in premature convergence and stagnation in a local optimum.

One of the main reasons for premature convergence is the amount of information shared in the swarm. In particular, there exists no sharing of information amongst particles in the swarm, except that with the particle that has the latest best position. This could lead the PSO algorithm to be trapped into a local optimum.

To overcome this undesirable stagnation behavior, a new dynamic model (i.e., velocity and position updating equations) is proposed. This model involves utilizing a new information from all members of a particle's neighbor referred to as the local best (l_{best}) that falls somewhere between the p_{best} and g_{best} (based on the work proposed by Xiao-Hua et al. [60] and Yong et al. [170]). Accordingly, the third leader, (l_{best}), is defined as the best position of the closest neighbor of each particle and is introduced in the equation for updating the velocity of the particles (as shown in Eq. (4.3)).

Based on this new model, each particle adjusts its position based on the following:

1. the current particle position;
2. the current particle velocity;
3. the distance between the current particle position and its best position (p_{best});

[13] Design parameters in the proposed APC- MOPSO algorithm include the cognitive (C_1), social (C_2), contiguous (C_3), position (C_p), and velocity (C_v) coefficients.

4. the distance between the current particle position and the swarm best position (g_{best}); and
5. the distance between the current particle position and the closest best position in neighbor (l_{best}).

This new model is found to increase the communication amongst particles in the swarm, and also raise the probability of breaking away from a local optimum.

Figure 4.4 shows a modified concept of updating the particle position in the search space by the PSO algorithm (i.e. a different model than is shown in Figure 4.1).

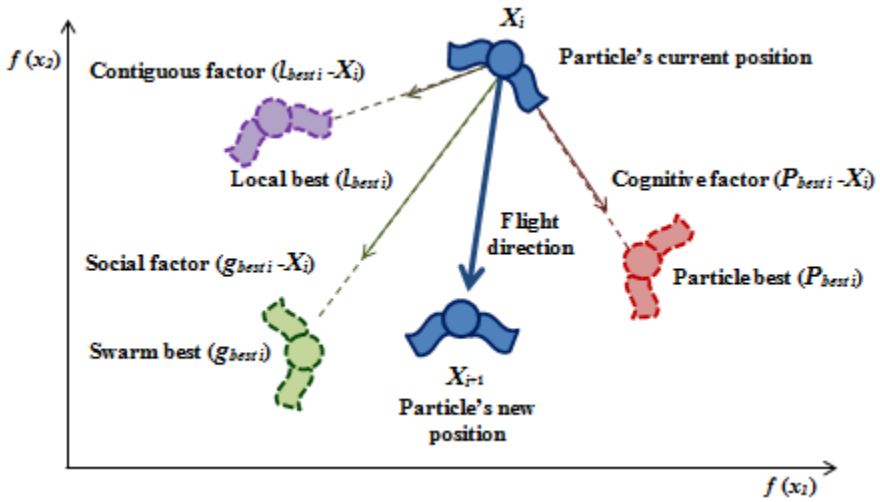


Figure 4.4: Updating Particle Position in the Search Space –APC-MOPSO Algorithm

The two leaders l_{best} and g_{best} are considered as collaborative behaviors for the swarm. Accordingly, it is pointed out that g_{best} is responsible for propelling the particles towards the global optima (i.e. speed the convergence rate). Whilst closest best neighbor position, l_{best}, is a mechanism that enables escape from a local optimum.

In this work, Eqs. (4.1) and (4.2) are modified with a new dynamic model (as mathematically expressed in Eqs. (4.3) and (4.4)) by incorporating the combination of three guides p_{best}, l_{best} and g_{best} in updating the particles velocity and position in order to steer their flight directions towards the Pareto-optimal regions. The proposed model involves two new position and velocity coefficients, known as C_p and C_v, respectively. These enhance the quality of the MOPSO algorithm in exploring new global optimum areas.

$$\vec{v}_{t+1} = w.[\vec{v}_t + c_1 \times rand(\ ) \times (\vec{p}_{best,i} - \vec{x}_t) + c_2 \times rand(\ ) \times (\vec{g}_{best,i} - \vec{x}_t) + c_3 \times rand(\ ) \times (\vec{l}_{best,i} - \vec{x}_t)] \quad 4.3$$

$$\vec{x}_{t+1} = C_p \times \vec{x}_t + C_v \times (\vec{x}_t + \vec{v}_{t+1}) \quad 4.4$$

where

$$C_{p_i} + C_{v_i} \leq 1 \qquad \forall i \in [1,2,\dots Maxiter] \quad 4.5$$

where three new coefficients are used in the above equations: C_3 is referred to as a contiguous coefficient that uses the Xiao-Hua expression for the closest best neighbor position, C_p and C_v correspond to the position and velocity coefficients, respectively.

The first term of Eq. (4.3) represents the particle's previous velocity; the second term, called the *cognitive* component represents the influence of the particle's self-experience in drawing its flight direction; the third component known as the *social* component represents the influence of the swarm leaders in guiding the other particles towards their best positions. In addition, this term embodies the collaborative behavior of the particles in the swarm by which a leader shares its best position with other particles. The last term of the Eq. (4.3) is called the *contiguous* component and represented the influence of the particle's closest neighbor in choosing its flight path. It should be noted here that the contiguous component shows another communal activity in the swarm by establishing a new communication structure amongst particles in addition to that of the swarm's leaders.

In Eq. (4.4), the new model makes an energy balance between the social activities in directing particles towards new areas (i.e. enhance search exploring) and the personal activities of the particles in moving around their previous positions and fine-tuning their search (i.e. enhance search exploitation). Accordingly, the condition of Eq. (4.5) must be satisfied at each iteration in order to maximize performance for updating the particle position (a proof of Eq. (4.5) is provided in Appendix A). In this work, a new adaptive scheme is used to automatically adjust the APC-MOPSO design parameters, including the inertia weight (w), cognitive (C_1), social (C_2), contiguous (C_3), position (C_p), and velocity (C_v) coefficients used in the above equations as shown next.

4.3.2 New Adaptive Techniques

Selection of design parameters, known as control parameters, is an important issue in PSO and MOEA algorithms as often considered as their main drawback. Therefore, research efforts have been directed on investigating the influence of the control parameters on the performance of PSO. Accordingly, these specifications can be made adaptively as reported by Shi et al. [224,225], Carlisle et al. [226], V.D.F. Bergh [227], Clerc et al. [228], I.C. Trelea [229], Bratton et al. [230], and Zhi-Hui et al. [231]. The underlying principle for the adaptation of the control parameters is that the dynamic behavior of the particles must change from exploration (i.e. discovering new areas) at the beginning of the search process into exploitation (i.e. fine-tuning their landed regions) at the end. The control parameter settings can place different emphasis between exploration and exploitation activities. This means that the parameter settings should vary over time. An adaptive technique is therefore increasingly used in new modified or hybrid PSO algorithms.

In this work, a new self-directed adaptive technique is proposed. In this technique, the adaptivity and refinement are applied to most of the control parameters, namely:

- the inertia weight, cognitive, social, and contiguous factors;
- position and velocity coefficients;
- mutation operator;
- the size of the external repository; and
- the boundaries of the search space.

As a result, the proposed APC-MOPSO algorithm is capable of efficiently exploring the search space and effectively exploiting high-quality solutions.

4.3.2.1 Adaptive Inertia Weight Factor

The choice of the inertia weight factor is a key for MOPSO algorithms to successfully solve complex, nonlinear, multi-peaks, and multi-constraint problems. In this new adaptive technique, the inertia weight factor (w) adaptively changes its value to balance global exploration (i.e. required at the beginning of the search process) with local exploitation (i.e. required at the end of the search process). This ability would lead to significant improvements in the performance of MOPSO algorithms.

In this work, two measured indicators are introduced to control the adjustment of the inertia weight factor during the iterative searching process. The two indicators are the evolution speed and the crowding distance. A detailed explanation of these indicators is provided in the following definitions.

Definition 1: Evolution Speed of Swarm (ES) [232]

Evolution Speed (ES) is an important indicator that measures the performance of the MOPSO algorithm in terms of not only the dynamic behavior of a particle's evolution (i.e. when a particle changes its position in the search space) but also the evolution speed of the swarm. Accordingly, the smaller ES is, the faster the evolution speed of the swarm. Hence, when ES is equal to 1.0 over several iterations, this indicates that the algorithm has converged to the global optimum. The mathematical representation of the ES is given in Eq. (4.6).

$$ES_t = \frac{min\{f(g_{best,\ t-1}), f(g_{best,\ t})\}}{max\{f(g_{best,\ t-1}), f(g_{best,\ t})\}}\ , \qquad 0 < ES \leq 1 \tag{4.6}$$

Here, it should be noticed that in a search process, the current fitness value of the global optimal solution $f(g_{best}(t))$ should be either better or equal to the value of the last iteration $f(g_{best}(t-1))$. For the minimization problem, $f(g_{best}(t)) \leq f(g_{best}(t-1))$, while for the maximization problem, $f(g_{best}(t)) \geq f(g_{best}(t-1))$.

Definition 2: Crowding Distance (CD) [233]

The Crowding Distance (CD) is another indicator that is used to measure the performance of the search engine (i.e. APC-MOPSO algorithm in this work). It indicates the density of solutions surrounding a particular candidate optimal solution (i.e. any non-dominated solution stored in the external repository) in the swarm for each objective function. For each objective function, solutions with smallest and largest function values (i.e., referred to as boundary solutions) are assigned an infinite distance value. Other intermediate solutions (i.e., solutions that fall within the boundary solutions) are assigned a distance value calculated using Eq. (4.7) as shown below:

$$CD = \sum_{i=2}^{num\_sol-1} \sum_{j=1}^{num\_obj} \frac{f_j(i+1) - f_j(i-1)}{max\ f_j - min\ f_j} \quad 4.7$$

where $j = 1,2,..,num\_obj$ $(number\ of\ objectives)$

$i = 2,3,..,num\_sol - 1$ $(number\ of\ solutions)$

This calculation is continued for all objective functions. Hence, the overall crowding distance value is then calculated as the sum of the individual distance values corresponding to each objective. Accordingly, after all solutions are assigned a crowding distance value, any two solutions (in the population) can be compared for their extent of proximity with other solutions. A solution with a smaller distance value (i.e., an overall crowding distance value for that solutions) is, more crowded by other solutions. The outline procedure for computing the crowding distance is given in Pseudo-code 12.

Pseudo-code 12: Procedure for Computing the Crowding Distance

Initialize: number of the non-dominated solutions in the external repository (*num_sol*), number of objective functions (*num_obj*)
for i = 1 : *num_sol* (initializing distance)
$CD(i) = 0$
end for i
for j = 1: *num_obj*
CD= sort(*CD*, *j*) (sorting using each objective value)
$CD(1) = CD(num\_sol) = \infty$
for i = 2 to (*num_sol* - 1)
Calculate CD using Eq. (4.7) as shown below
$CD(i) = CD(i) + \frac{f_j(i+1) - f_j(i-1)}{max\ f_j - min\ f_j}$
end for i
end for j
Normalize *CD* $0 < CD \leq 1$

In the first loop, initializing with zero the distance of each element i of the size (*num_sol*). In the second loop, at each objective j, the whole set of solutions (i.e., of size *num_sol*) is sorted in a ranking according to the value for j. Then, the boundary solutions for each objective j are defined by assigning the distance value for the first and last position as infinity (∞) in order to preserve solutions with extreme values.

The inner loop presents the calculation of a distance value for each remaining solution i (i.e., from position 2 to (*num_sol* – 1)). First, it requires finding the difference between the objective values for the neighbors of i. Then, finding the difference between the boundary solutions (i.e., between the highest and lowest value). Finally, the crowding distance value for solution i is updated by the sum of its previous value with the new normalized value. Figure 4.5 depicts the calculation of crowding distance for a given solution i. Within this context, the distance value for i will be $r + s$ where:

$$s = \frac{f_1(i-1) - f_1(i+1)}{f_1(a) - f_1(z)} \qquad r = \frac{f_2(i+1) - f_2(i-1)}{f_2(z) - f_2(a)}$$

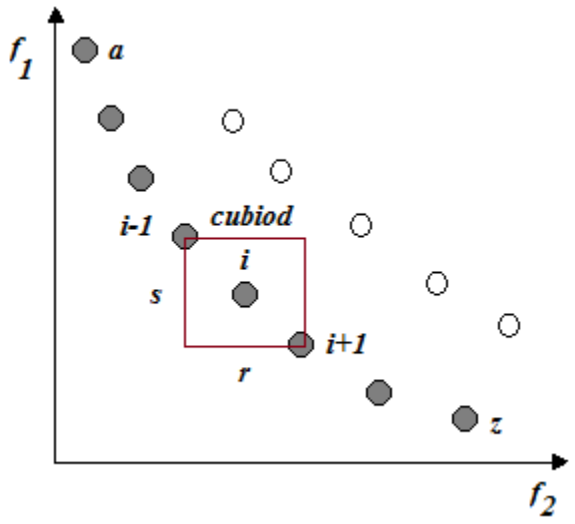


Figure 4.5: The Crowding Distance Calculation

In this new adaptive technique, the inertia weight factor (w) adjusts its value according to the change of evolution speed and crowding distance indicators. For the former (ES indicator), the inertia weight factor should be increased when the ES of the swarm is fast, which enhances the global exploration ability of the MOPSO in finding a new optimal region in the search space. On the contrary, the inertia weight factor should be decreased when the ES of the swarm slows down to zoom into a region of the search space for fast convergence. For the latter (CD indicator), the inertia weight factor should be decreased when the CD is low in order to prevent the swarm from falling into a local optimum by improving the global exploration ability of the MOPSO algorithm. On the other hand, if the CD is high, this means that the particles are scattered and the

swarm has no tendency to converge into local optima. As a result, the inertia weight factor (w) is designed to be increased as ES becomes fast. While it should be decreased as CD gets slow. The formula that adapts the inertia weight as a function of both ES and CD is given in Eq. (4.8).

$$w_t = w_{start} - rand(0,1) \times ES_t - rand(0,1) \times CD_t \quad (4.8)$$

where w_{start} is the initial value of the inertia weight factor.

4.3.2.2 Adaptive Cognitive, Social, and Contiguous Factors

Based on the new dynamic model (as explained in subsection 4.3.1), the cognitive, social, and contiguous coefficients, referred to as C_1, C_2 , and C_3 respectively, are very important for the exploration, exploitation, convergence, and escape abilities of our MOPSO algorithm. According to the effect of these three parameters, the proposed APC-MOPSO dynamically adjusts their values in relation to four conditions as provided in Table 4.1. Therefore, the self-adjustments of those coefficients can lead the APC-MOPSO algorithm to produce high-quality Pareto-front solutions.

Table 4.1: The Strategies for Tuning the Cognitive, Social, and Contiguous Factors

Cases	Durations	States	C_1	C_2	C_3
Case 1	At the beginning	Exploration	Larger	Smaller	Larger
Case 2	At the end	Exploitation	Smaller	Larger	Smaller
Case 3	When landed-in	Convergence	Decreased	Increased	Decreased
Case 4	When stagnating	Jumping-out	Increased	Decreased	Increased

The four strategies (i.e., the cases outlined in the above table) are listed according to the evolutionary states of the search process. The values of C_1 and C_3 are initialized with a reasonably large value and gradually changes in the exploration state, while a larger C_2 is required in the convergence state. Along with the search algorithm, a balance between global and local search ability is adaptively attained. Moreover, in the case of getting trapped into a local optimum (i.e., in the jumping-out state), larger C_1 and C_3 with a smaller C_2 are required to make the swarm separate from the local region and fly to the new and better region.

The cognitive, social, and contiguous coefficients are dynamically changing their values according to Eqs. (4.9), (4.10), and (4.11):

$$C_1 = C_{1,\ max} - \left(C_{1,\ max} - C_{1,\ min}\right) \times e^{-\left(5 \times \frac{iter}{Maxiter}\right)^2} \quad (4.9)$$

$$C_2 = C_{2,\ min} + \left(C_{2,\ max} - C_{2,\ min}\right) \times e^{-\left(5 \times \frac{iter}{Maxiter}\right)^2} \quad (4.10)$$

$$C_3 = C_{3,\ max} - \left(C_{3,\ max} - C_{3,\ min}\right) \times e^{-\left(5 \times \frac{iter}{Maxiter}\right)^2} \quad (4.11)$$

* *The exponential term value varies within the range of [1, 0], 1 at the first iteration, and 0 at the last iteration.*

4.3.2.3 Adaptive Position and Velocity Coefficients

According to the new dynamic model (previously explained in subsection 4.3.1), a particle's position is updated based on its previous position and velocity. Thus, there is an urgent need to balance the influence of those two components in steering the particle towards a global optimum. Hence, in this subsection, a new dynamic technique is proposed to linearly vary (i.e. increase or decrease) the values of the position and velocity coefficients along with the iterative search process.

In the early stages of the search process, the particles are encouraged to widely explore and visit new areas in the search space. Therefore, a larger influence of the velocity component is preferred in updating the current particle's position. Accordingly, the velocity coefficient is chosen to be initially high and to gradually decrease with the increasing number of iterations. At late stages of the search process, the particles converge and then exploit a region of the search space. At this stage, a larger influence of a particle's previous position is preferred in moving the particle. Accordingly, the initial value of the position coefficient is set low and then is linearly increased along with the number of iterations. The position and velocity coefficients are adapted by using Eqs. (4.12) and (4.13), as follows:

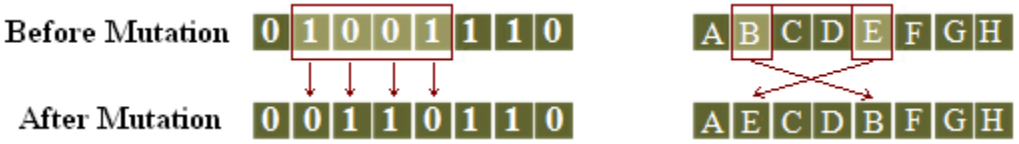
$$C_p = C_{p,\ lower} + \left(C_{p,\ higher} - C_{p,\ lower}\right) \times (iter/Maxiter) \quad 4.12$$

$$C_v = C_{v,\ higher} - \left(C_{v,\ higher} - C_{v,\ lower}\right) \times (iter/Maxiter) \quad 4.13$$

It is important to note that both C_p and C_v are limited within the range (0.1, 1).

4.3.2.4 Adaptive Mutation Operation

Generally, mutation operation is known as a process that occurs very infrequently (i.e. occasional operation) to introduce random modifications to elements within the population in order to improve the population diversity and then to prevent the population from stagnating at any local optima. Usually, mutation operation is executed according to the mutation probability (p_m), and when the population diversity is deteriorated along with the iterative searching process. Examples of mutation in binary numbers are illustrated in Figure 4.6.



a. Multi-bit Flip Type of Mutation b. Multi-bit Swapped Type of Mutation

Figure 4.6: Examples of Different Types of Mutation Operation

The main advantages of incorporating mutation in multi-objective heuristic algorithms can be described as follows:

- randomness: introduces new particles to the swarm that can induce random walk through the search space;
- preserving diversity: new particles explore new areas by which premature convergence is prevented; and
- enhancing solution quality: helps the search algorithm to arrive at better solutions than were previously possible.

Attempts have been made to improve the performance of the MOPSO to handle hard real-world problems. Examples of the adaptive mutation operation have been reported by Zhi-Hui et al. [231], Tang et al. [234], Higashi et al. [235], Stacey et al. [236], Pant et al. [166], Chen et al. [237], Esquivel et al. [105], Li et al. [238], Li et al. [239], Li et al. [240], Ratnaweera et al. [68], and Wang et al. [241].

In this work, a new adaptive mutation technique is proposed that takes into account the particles' position and velocity. To illustrate the procedure of the proposed mutation technique, two definitions need to be introduced as follows.

Definition 1: *Space Position* (S_p) [172]

Let S_p be the space-position aggregation degree in the *t-th* iteration such that:

$$S_p = \max_{1\le i\le n}\left\{\left\|x_i^t - g_{best}^t\right\|_2\right\} / \max_t\left\{\max_{1\le i\le n}\left\{\left\|x_i^t - g_{best}^t\right\|_2\right\}\right\} \quad 4.14$$

This parameter is used to describe the position of the particles in the decision space. A higher value of S_p indicates that the particles are widely spread in this decision space, and a smaller value of S_p means that the particles are crowded in a certain region in the decision space.

Definition 2: Space Fitness (S_f)

Let S_f be the space-fitness aggregation degree in the *t-th* iteration. Then, let:

$$S_f = \max_{1\le i\le n}\left\{\left\|f(x_i^t) - f(g_{best}^t)\right\|_2\right\} / \max_t\left\{\max_{1\le i\le n}\left\{\left\|f(x_i^t) - f(g_{best}^t)\right\|_2\right\}\right\} \quad 4.15$$

where fg_{best}^t is the fitness of the swarm best particle at the *t-th* iteration.

Space Fitness (S_f) is used as a measure of how far a set of particles is spread out in the search space. The higher the space-fitness (S_f), the farther the particles are located in the search space, and vice versa.

In this adaptive mutation technique, with increasing iteration number, the space position and the space fitness of particles become closer to zero, [172]. This indicates that the algorithm is likely to stagnate into a local optimum. Hence, the mutation operation needs to be deployed to improve the search process (i.e., requires higher mutation probability). On the contrary, when space position (S_p) or space fitness (S_f) is larger, particles are dispersible (i.e., flight in all

directions); and the need for the mutation operation is small (i.e., requires lower mutation probability).

The comparison between the values of the two indicators (S_f and S_p) is used to determine the mutation probability (p_m). The equations used to calculate the mutation probability (p_m) are given as follows.

$$p_m = \begin{cases} 1/S_p & if\ (S_p > S_f) \quad (4.16) \\ 1/S_f & if\ (S_p < S_f) \quad (4.17) \end{cases}$$

Then, the mutation operation is implemented according to Eq. (4.18).

$$x_i^t = r \times {l_{best}}_i^t + (1 - r) \times x_i^t \qquad if\ r < p_m, \quad (4.18)$$

where *r* is a random number in the range of [0,1] and x_i is the position of the *i-th* particle at iteration *t.*

The procedure of the adaptive mutation is outlined in Pseudo-code 13.

Pseudo-code 13: Adaptive Mutation Technique

At *t-th* iteration, calculate: • S_p using Eq. (4.14) • S_f using Eq. (4.15)
if $(S_p > S_f)$ (space position indicator is selected)
Comparing between the space position and space fitness values to decide which indicator is used to determine the mutation probability (p_m)
Use Eq. (4.16)
if $r < p_m$,
Apply the mutation operation according to Eq. (4.18)
end if
else (space fitness indicator is selected)
Obtain the mutation probability as given in Eq. (4.17)
if $r < p_m$,
Implement mutation operation according to Eq. (4.18)
end if
end if

4.3.2.5 Adaptive Search Space Boundaries

Adjusting the boundaries of the search space has a vital role in landing particles into a Pareto-optimal front, [242]. Additionally, to accelerate the convergence speed, a new adaptive search space strategy is devised based on the distance between the best position of the swarm and the boundaries of the search space. In this technique, the size of the search space is adjusted to

ensure that particles progressively find a Pareto-optimal front. Moreover, the search-space reduction strategy is applied whenever needed to search for a Pareto-optimal front.

In other words, when the performance of the algorithm is not improved during a pre-specified number of iterations, the adaptive mechanism dynamically adjusts the search space (i.e., reduces) according to the distance between the swarm best position (g_{best}^{t}) and the minimum and maximum particle's best position at the same iteration ($p_{best,min}^{t}$ and $p_{best,max}^{t}$, respectively). To determine the adjusted minimum/maximum size of the search space at iteration (*t*), a new variable referred to as $DS_{reduction}$ is introduced (as given in Eqs. (4.19) and (4.20)).

$$if\ \left(p_{best,j,max}^{t} - g_{best,j}^{t}\right) > \left(g_{best,j}^{t} - p_{best,j,min}^{t}\right)$$

$$DS_{reduction} = \frac{\left(g_{best,j}^{t} - p_{best,j,min}^{t}\right)}{3} \quad 4.19$$

else

$$DS_{reduction} = \left(p_{best,j,max}^{t} - g_{best,j}^{t}\right)/3 \quad 4.20$$

where $DS_{reduction}$ is calculated according to the smaller distance.

This variable represents the distance that is subtracted (added) from the maximum (minimum) size of the search space at iteration (*t+1*) as described by Eqs. (4.21) and (4.22).

$$p_{best,j,max}^{t+1} = p_{best,j,max}^{t} - DS_{reduction} \quad 4.21$$

$$p_{best,j,min}^{t+1} = p_{best,j,min}^{t} + DS_{reduction} \quad 4.22$$

Figure 4.7 depicts how the search space of each iteration is dynamically reduced when activated. The procedure of determining the size of the search space is outlined as follows.

Pseudo-code 14: Adaptive Searching-Space Technique

if ($p_{best,j}$ is NOT changed) (at iteration *t*)
if $\left(p_{best,j,max}^{t} - g_{best,j}^{t}\right) > \left(g_{best,j}^{t} - p_{best,j,min}^{t}\right)$
$DS_{reduction}$ is calculated using Eq. (4.19)
else
$DS_{reduction}$ is calculated using Eq. (4.20)
end if
$p_{best,j,max}^{t+1}$ is determined using Eq. (4.21)
$p_{best,j,min}^{t+1}$ is determined using Eq. (4.22)
end if

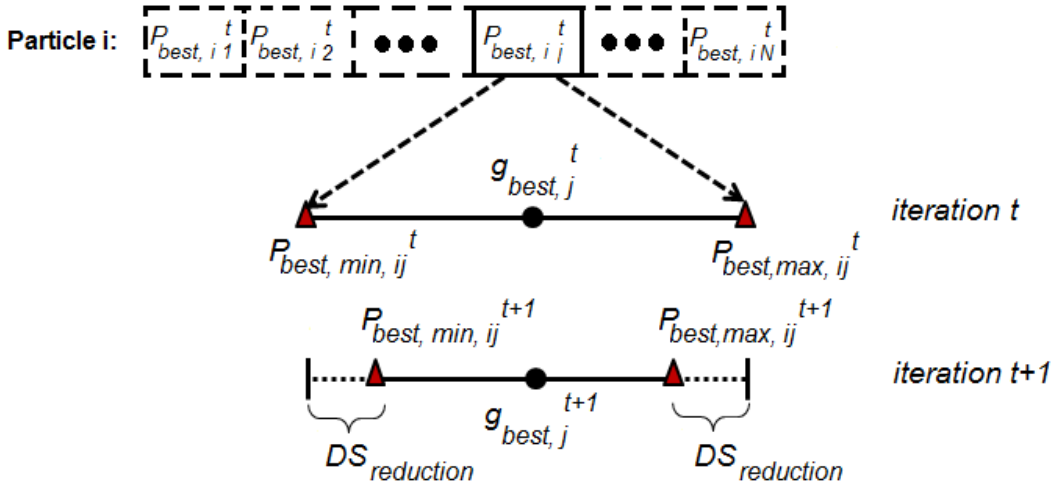


Figure 4.7: Schematic of the Adaptive Search-Space Boundaries

4.3.2.6 Adaptive External Repository Size

The external repository is a mechanism that has a large impact on the performance of any heuristic search engine, including Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). Within the context of PSO, the external repository keeps a historical record of the non-dominated solutions obtained at each iteration and provides the search algorithm with solutions that may be considered as leaders to guide the particles towards the Pareto-optimal front. It can also enhance the overall efficiency of algorithms by implementing a broad distribution of the non-dominated solutions along the Pareto-front.

The following three cases illustrate the concept, the mechanism, and the implementation of the external repository when trying to store a non-dominated solution. It should be noted that all (or any one) of these cases can occur during the iterative searching process:

- Case (1): when the external repository is empty (i.e. at the beginning of the search process),
- Case (2): when the external repository is neither empty or full, and
- Case (3): when the external repository is full.

Figure 4.8 depicts (Case 1) when the external repository is initially empty, and explains the mechanism for storing the new selected "better" solutions (i.e. the non-dominated solutions found at each iteration). In this case, all new non-dominated solutions are archived and deposited into the external repository.

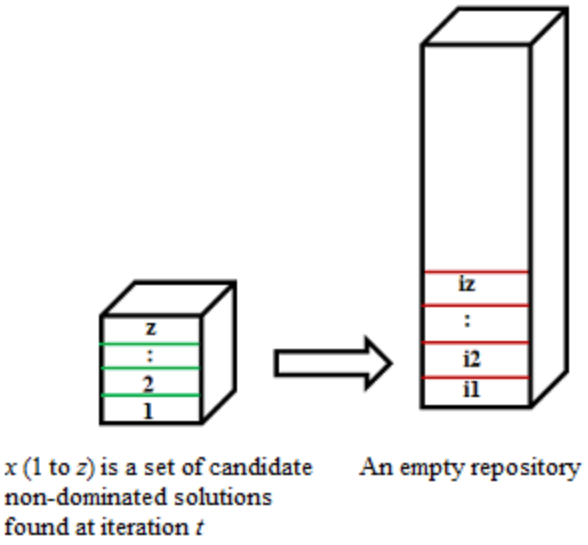
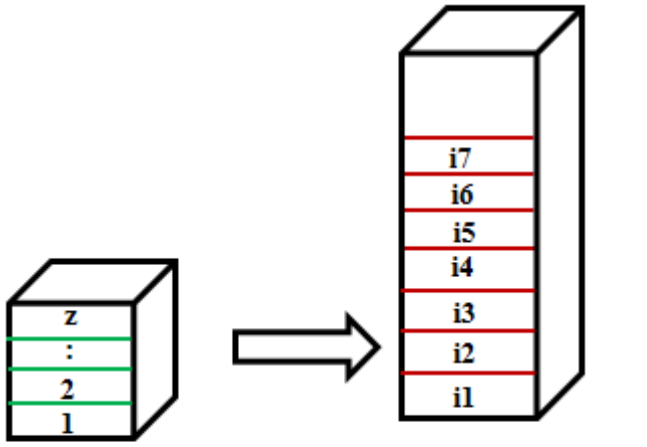


Figure 4.8: Archiving New Solutions into an Empty External Repository (Case 1)

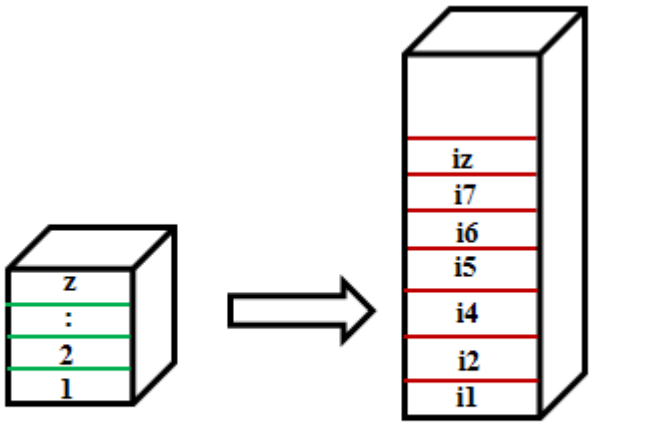
Figure 4.9 illustrates the mechanism of storing new non-dominated solutions in the external repository when it is not empty or full (Case 2). Three different scenarios are considered in this technique. In each of these scenarios, there will be a comparison between the new batch of solutions and the solutions stored in the external repository using the concept of Pareto-dominance. For the first scenario of (Case 2), the new Pareto-front solution (e.g. solutions z) is rejected when it is dominated by at least one of the archived solutions in the external repository (as shown in Figure 4.9.a). The second scenario occurs when the new Pareto-front solution (e.g. solutions z) dominates one or more archived solutions in the external repository. This result in all eliminating all archived solutions in the external repository and then substituting by the new Pareto-front solution (as depicts in Figure 4.9.b). The last scenario of (Case 2) occurs if the new candidate solution(s) (e.g. solution *z*) is not dominated by the archived solutions in the external repository, then the new solution is added to the external repository (as illustrates in Figure 4.9.c).



x (1 to *z*) is a set of candidate non-dominated solutions found at iteration *t*+1

External repository is NOT full

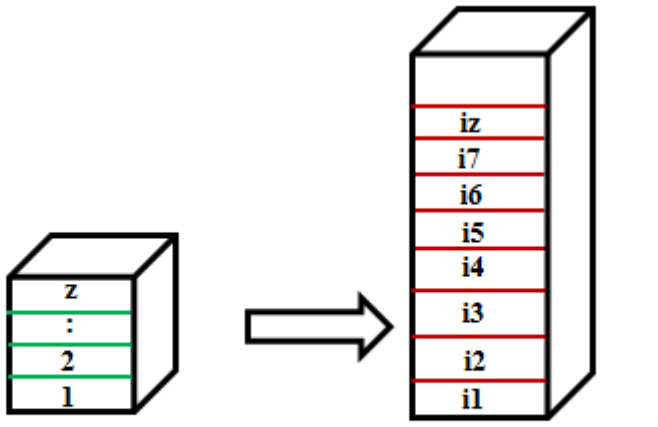
a. *if* candidate solution (z) is dominated by at least one solution in the repository, *then* solution (z) is discarded.



x (1 to *z*) is a set of candidate non-dominated solutions found at iteration *t*+1

External repository is NOT full

b. *if* one (or more) stored solution in the repository (e.g., i3) is dominated by candidate solution (z), *then* the dominated archived solution(s) are eliminated, and candidate solution (z) is stored instead.



c. *if* one (or more) candidate solution (e.g., solution *z*) is NOT dominated by any archived solution in the repository, *then* candidate solution (*z*) is stored.

Figure 4.9: Storing New Non-dominated Solutions in the External Repository[14] (Case 2)

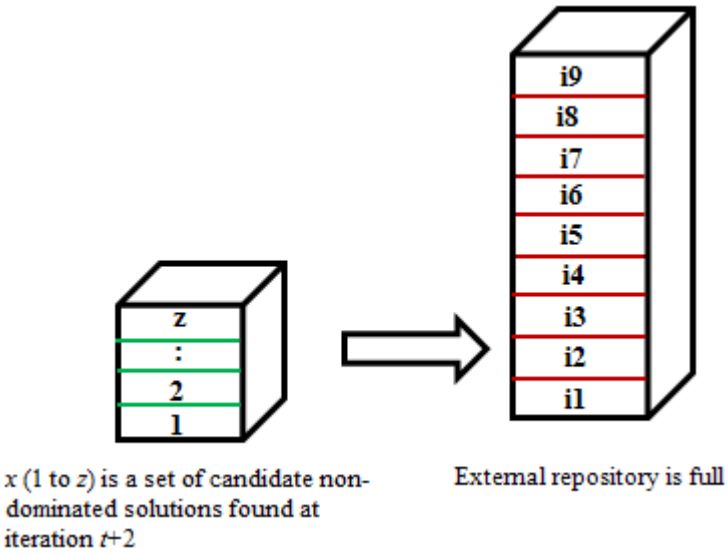
Just like in the previous case, three scenarios are considered for (Case 3) when the external repository is full. In the first scenario, the new candidate solution (e.g. solution *z*) is ignored if it is dominated by at least one solution in the repository (see Figure 4.10.a). The second scenario represents one of the most important situation in which the external repository is full and the new Pareto-front solution (e.g. solutions *z*) dominates one or more solutions in the repository. In this scenario, the repository will free at least one place for the new competitive solution (see Figure 4.10.b).

An additional scenario that needs to be considered for (Case 3) is when the external repository is full and there is a new non-dominated solution(s) (e.g. solution *z*) that is non-dominated by any solution in the repository. This scenario often occurs at the end of the iterative process when there is no free space in the external repository. This problem can be resolved by using clustering techniques on the external repository to reduce the number of the archived solutions in the repository and free spots for new solutions (as shown in Figure 4.10.c). In this dissertation, a modified K-Means clustering technique is proposed for resolving this scenario as described in the next section.

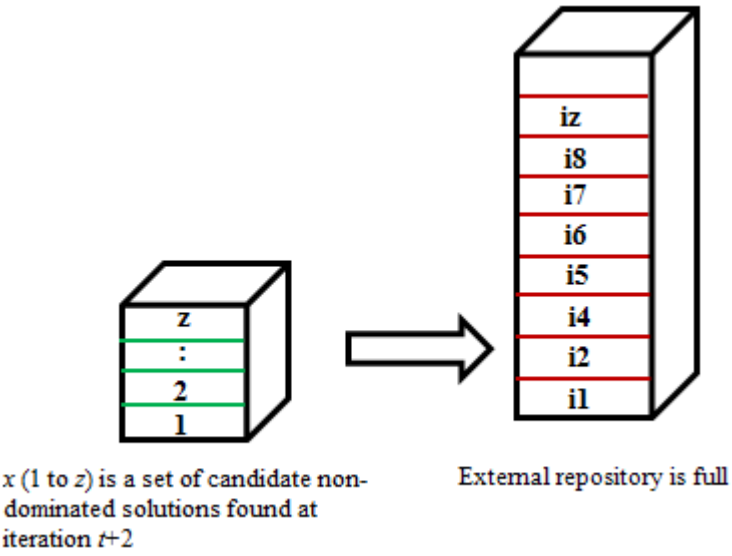
A fixed size of the external repository has been used in most optimization algorithms that incorporate archiving [81,77,97,243]. In this work, an adaptive repository size is proposed to

[14] When there is at least one solution already in the repository, it must be checked for dominance against those solutions.

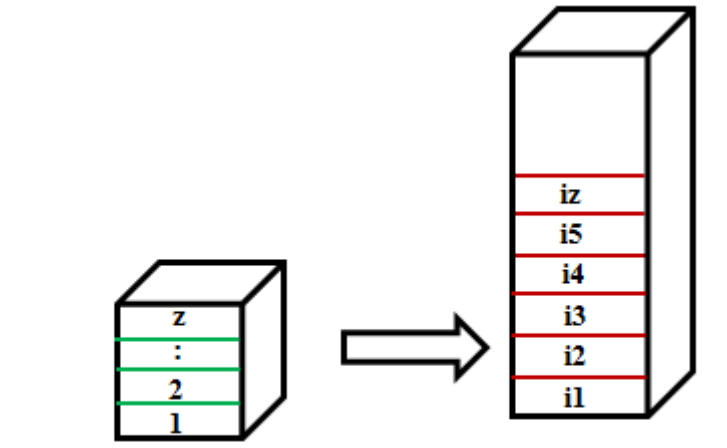
allow greater flexibility in preserving non-dominated candidate solutions along the Pareto-front that are well distributed in the search space.



a. *if* the candidate solution (z) is dominated by at least one solution in the repository, *then* candidate solution (z) is rejected.



b. *if* one or more archived solutions in the repository (e.g., i3 and i9) are dominated by the candidate solution (*z*), *then* those archived solutions are eliminated and solution (*z*) is stored instead.



c. *if* candidate solution (z) is NOT dominated by any solution in the repository, *then* K-Means takes place to eliminate some of the stored solutions and solution (z) is stored.

Figure 4.10: Storing New Non-dominated Solution in the Repository[15] (Case 3)

The variable repository size (S_{rep}) is calculated according to the percentage ratio, known as $N\%$, of the total number of non-dominated solutions found at each iteration to the population size. The S_{rep} is then calculated for 2, 3, and 5 objectives and above using Eqs. (4.23), (4.24), and (4.25), respectively.

- Two-objectives Problem:

$$S_{rep} = \begin{cases} pop_{size} & if\ N\% < S_{min} \\ ceil\left(pop_{size} \times \left(\frac{N\% + 3 \times S_{min}}{S_{max}+S_{min}}\right)\right) & if\ S_{min} \le N\% \le S_{max} \\ pop_{size} \times 2 & if\ N\% > S_{max} \end{cases} \quad 4.23$$

- Three-objectives Problem:

$$S_{rep} = \begin{cases} pop_{size} \times 3 & if\ N\% < S_{min} \\ ceil\left(pop_{size} \times \left(\frac{N\% + 12 \times S_{min}}{S_{max}+S_{min}}\right)\right) & if\ S_{min} \le N\% \le S_{max} \\ pop_{size} \times 5 & if\ N\% > S_{max} \end{cases} \quad 4.24$$

- Five-objectives Problem (or more):

[15] When the repository is full, it is required to check for non-dominance and apply K-Means clustering.

$$S_{rep} = \begin{cases} pop_{size} \times 8 & if\ N\% < S_{min} \\ ceil\left(pop_{size} \times \left(\frac{N\% + 40 \times S_{min}}{S_{max}+S_{min}}\right)\right) & if\ S_{min} \le N\% \le S_{max} \\ pop_{size} \times 10 & if\ N\% > S_{max} \end{cases} \quad 4.25$$

where S_{min} and S_{max} are the minimum and maximum bounds of the percentage ratio (*N%*). In this work, the S_{min} and S_{max} are chosen to be 25% and 75%, respectively (Constants 3, 12, and 40 are chosen according to the proposed S_{min} and S_{max} values, and can be re-calculated if new boundary values are used).

The procedure of updating the size of the external repository is given in following pseudo-code.

Pseudo-code 15: Adaptive External Repository Size Algorithm

if (No. of the objective function = = 2)
S_{rep} is calculated using Eq. (4.23)
elseif (No. of the objective function = = 3)
S_{rep} is calculated using Eq. (4.24)
elseif (No. of the objective function > 3)
S_{rep} is calculated using Eq. (4.25)
end if

4.3.3 K-Means Clustering Technique

In contrary to the Single-objective Optimization Problems (SOOPs), there is more than one solution to the Multi-objective Optimization Problems (MOOPs). Moreover, there are a number of non-dominated solutions located on the Pareto-optimal front. Within the context of MOPSO algorithms, each non-dominated solution can be a leader, g_{best}, and guide other particles towards its position. Thus, for any heuristic algorithm, the problem is "*how to deal with a large number of non-dominated solutions*?" and then "*how to select leaders among these solutions*?". Together, these two questions have resulted in one of the main difficulties in solving MOOPs that is preservation of diversity and preventing the poor distribution of non-dominated solutions along the Pareto-optimal front.

In the previous adaptive external repository size subsection, (Case 3), the need for clustering is discussed. Clustering can be used as a mechanism for archiving new non-dominated solutions when there is no free space in the external repository. Incorporating clustering techniques has been found to be very effective in enhancing the selection of better solutions and improving the distribution of non-dominated solutions along the Pareto-optimal front [81,77,97,243]. Use of a clustering technique greatly improves diversity as it effectively preserves

the size of the repository, while enhancing the ability of the algorithm to explore the search space more effectively and efficiently (i.e., coverage of the search space).

Before describing the proposed K-Means clustering and its implementation on the external repository, it is important to first provide the concept of the standard K-Means algorithm, [244]. K-Means is an algorithm to cluster n objects based on attributes into k partitions, $k < n$. K-Means method is considered an easy, simple, and fast cluster algorithm.

The K-Means algorithm comprises of a simple re-estimation four steps as follows. Step 1, initially the data points are randomly selected from the k clusters. For step 2, the centroid is calculated for each set. In step 3, each point is assigned to the cluster whose centroid is nearest to that point. Step 4 is the re-estimation of steps 2 and 3 until there is no further change in the assignment of the data points. Pseudo-code 16 illustrates the procedure of the standard K-Means.

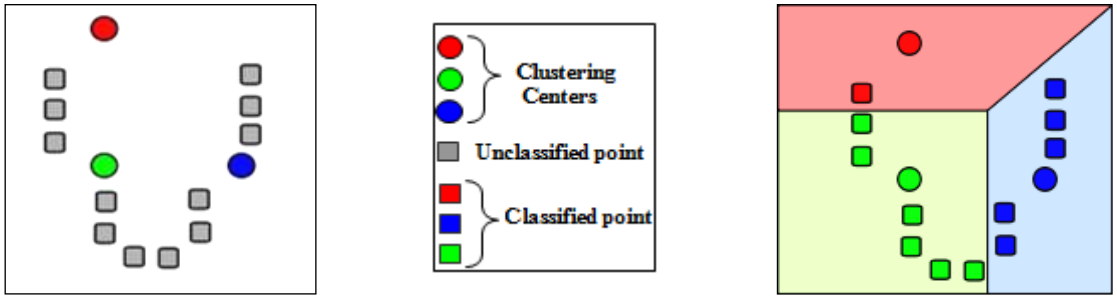
Pseudo-code 16: Standard K-Means Clustering

1. Choose randomly the k number of the cluster centers $C = \{c_1, c_2, \dots, c_k\}$
2. *for* each $i \in \{1,2, \dots, k\}$ do:
   - Set the cluster C_i to be the set of points in X that are closer to c_i than they are to c_j for all $j \neq i$
3. *for* each $i \in \{1,2, \dots, k\}$ do:
   - Set the cluster ci to be center of mass of all points in C_i:

$$c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

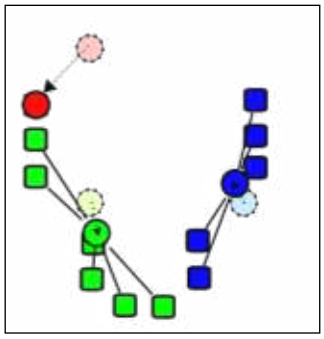
4. Repeat steps 2 and 3 for no longer changes in C

Figure 4.11 shows an example for the standard K-Means in a 2-dimensional case.

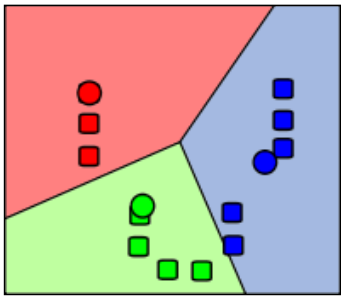


a) k initial are randomly selected from the data set ($k = 3$) (as shown in color)

b) k clusters are created by associating every observation with the nearest mean



c) The centroid of each of *k* clusters becomes the new means.



d) Steps 2 & 3 are repeated until convergence has been reached.

Figure 4.11: An Example of the Standard K-Means Algorithm [245]

David et al. [246], proposed a new K-Means^{++} algorithm to minimize the average squared distance between the points in the same cluster. The algorithm adopts a very simple randomized seeding to improve the speed and the accuracy of K-Means. The algorithm starts by randomly selecting only one cluster center, and then, the rest of the cluster centers are chosen based on a new probability criterion calculating by using Eq. (4.26). Next, the algorithm proceeds with the standard K-Means steps (2-4).

$$\frac{D(\acute{x})^2}{\sum_{x\in X} D(x)^2} \qquad 4.26$$

where $\acute{x}$ represents a member of the data set $(\chi \subset \mathbb{R}^d)$.

In this section, a modified version of the K-Means^{++} clustering algorithm is proposed to enhance the mechanism of storing new solutions in the external repository. In the modified K-Means^{++} algorithm, a new radius *r* is introduced to the K-Means^{++} algorithm (step 5), and is determined by using Eq. (4.27). Then, circles with radius *r* are drawn (i.e., in the case of the 2-dimenional case) from each cluster center (i.e., each cluster center represents a center of one of the circles), so that the data points that are located inside them are eliminated.

$$r = \frac{\sum_{i=1}^{n-1} d\{(f_i^{min}, f_{i+1}^{max}), (f_i^{max}, f_{i+1}^{min})\}}{2\, S_N}$$

$$OR$$

$$r = \frac{\sum_{i=1}^{n-1} sqrt\left(\left(f_i^{min} - f_i^{max}\right)^2 - \left(f_{i+1}^{max} - f_{i+1}^{min}\right)^2\right)}{2\, S_N} \qquad 4.27$$

where, S_N denotes the number of non-dominated solutions that exist in the repository found so far, *d* is the Euclidean distance, and f_i^{min}, f_i^{max} are the maximum and minimum of the *i-th* objective in the non-dominated solutions found so far, respectively.

The procedure of applying the modified K-Means^{++} clustering algorithm to the external repository is provided in Pseudo-code 17:

Pseudo-code 17: Modified K-Means^{++} Clustering Algorithm

Let D(x) denotes the shortest measured distance between a data point *x* and the closest center that has already chosen. Then:

1. Choose randomly *k* number of the cluster centers $C = (c_1, c_2, \dots, c_k)$
   a. Choose an initial center c_1 uniformly at random from *X*.
      where X represents a set of non-dominated solutions stored in the repository
   b. *for i* = 1: *k*
      DO
      - Choose the next center c_i
      - Selecting $c_i = \acute{x} \in X$ with a probability determined by Eq. (4.26):

      where $\acute{x}$ *represents a member of the non-dominated solutions stored in the repository (X)*

      end for i
2. *for i* = 1: *k*
   DO
   - Set the cluster C_i to be the set of points in *X* that are closer to c_i than they are to c_j for all $j \neq i$

   end for i
3. *for i* = 1: *k*
   DO
   - Reset the cluster center c_i to be center of mass of all points in C_i:

     $$c_i \frac{1}{|C_i|} \sum_{x \in C_i} x$$

   end for i
4. Repeat steps 2 and 3 until there are no more changes in *C*.
5. Computing radius *r* as expressed in Eq. (4.27)
6. Assign each clustering center obtained from step 4 with a radius *r*.
7. Eliminate all the particles lay inside the K-Means circles.

Figure 4.12 illustrates the mechanism of eliminating the solutions for minimizing a bi-objective problem. When the external repository is full, the modified K-Means^{++} is applied to eliminate some of the stored solutions and free spots in the external repository.

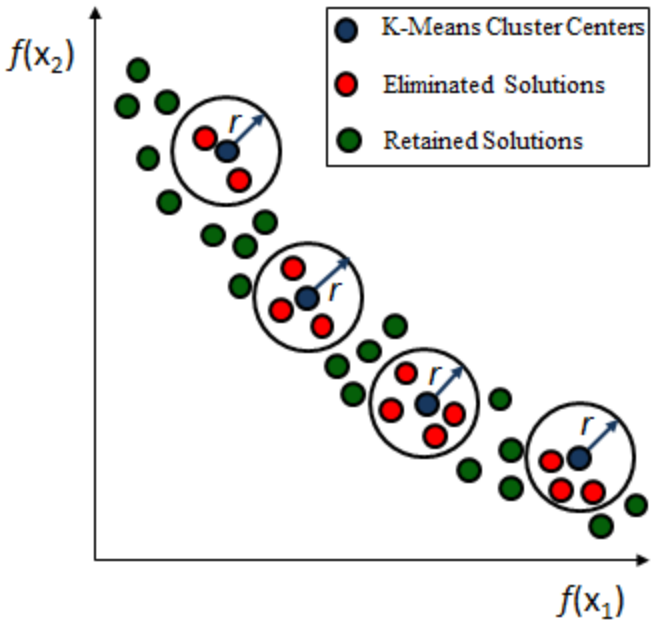


Figure 4.12: Modified K-Means^{++} Clustering Technique

4.3.4 Parallel Islands Model

When applied to large hard problems, MOPSO algorithms may become too slow. One way to overcome time and size constraints is to parallelize them (i.e., the computational engines, CPU or multi-core processor, simultaneously perform the MOPSO algorithm). Recent MOPSO algorithms lend themselves well to parallel computing to improve computational complexity. This is done by sharing the workload, in which a *N*-processor system will do the computation nearly *N* times faster than a uniprocessor system, [247]. The main characteristic of this mechanism is that the full swarm exists in distributed form. A common use of the decomposition approach is referred to as the Parallel Islands. In this approach, the swarm is divided into several sub-swarms. Each sub-swarm is assigned to a different processor (*island*). Each processor runs a sequential MOPSO on its swarm. Parallel islands models allow migration (i.e. periodic exchange) of good candidate solutions from one island to another after every fitness evaluation, [248]. Semi-isolated sub-swarms help maintain MOPSO diversity (as shown in Figure 4.13). Therefore, the swarm of each island can explore a different part of the search space. Thereby parallel islands models allowing MOPSO algorithms to efficiently solve the Multi-objective Optimization Problems (MOOPs), [249,250,251,252]. Furthermore, parallel islands models can also be seen as a key factor for the success of the external repository technique by providing a sufficient number of good solutions found by the sub-swarms, [249].

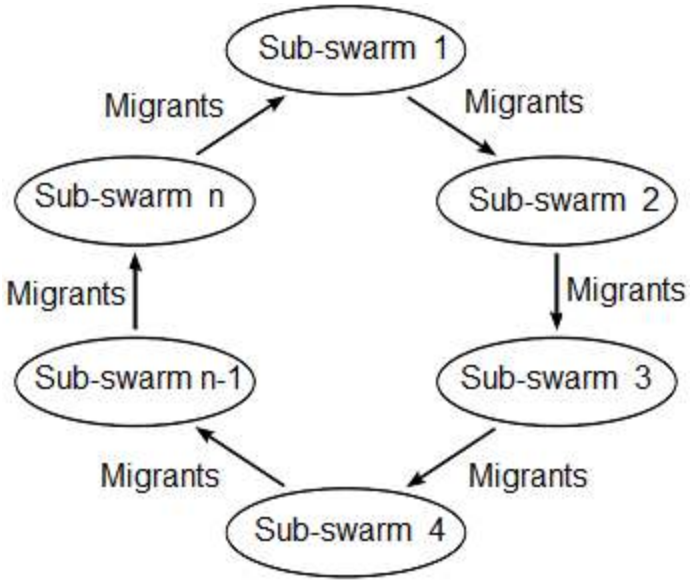


Figure 4.13: The Islands Model of *n* Semi-isolated Sub-swarms

In MOPSO algorithms, parallel islands models have been found to be better in search performance than single swarm models in terms of the quality of the solutions and the reduced computational time, [249]. In this work, the parallel islands model is implemented using the Parallel Computing toolbox and MATLAB Distributed Computing Server. The toolbox allows solving computationally problems using MATLAB® on multi-core and multiprocessor computers. The parallel computing toolbox consists such that it implements task-parallel and data-parallel algorithms without programming for a specific hardware architecture. The toolbox also performs the execution of multi-task job by evaluating each of its tasks and returning the results. Moreover, the parallel computing toolbox allows up to eight MATLAB workers to run (i.e., the server's individual MATLAB sessions are called workers) on a local machine. Scheduling or job manager is the part of the server software that coordinates the execution of jobs and the evaluation of the task. Figure 4.14 shows the basic parallel computing configuration.

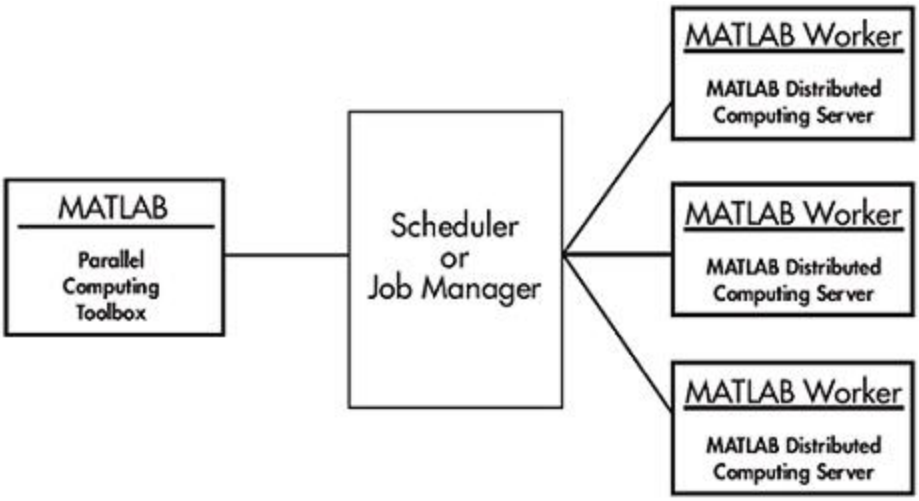


Figure 4.14: Basic Parallel Computing Architecture in a Multi-core Processor

4.3.5 Overall Adaptive Parallel Clustering-based MOPSO Algorithm

The proposed Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) not only has a technique to guide the search for better solutions, but also aims to have well-distributed solutions along the Pareto-optimal front. To accomplish this, the previously described concept of Pareto-dominance, a new dynamic model, an adaptive weight, cognitive, social, contiguous, position, and velocity coefficients, an adaptive mutation, an adaptive external repository size, an adaptive search-space boundaries, a modified K-Means clustering algorithm, a parallel islands model, and a parallel computing architecture are incorporated into it.

The flowchart of the proposed Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization algorithm (APC-MOPSO) is depicted in Figure 4.15.

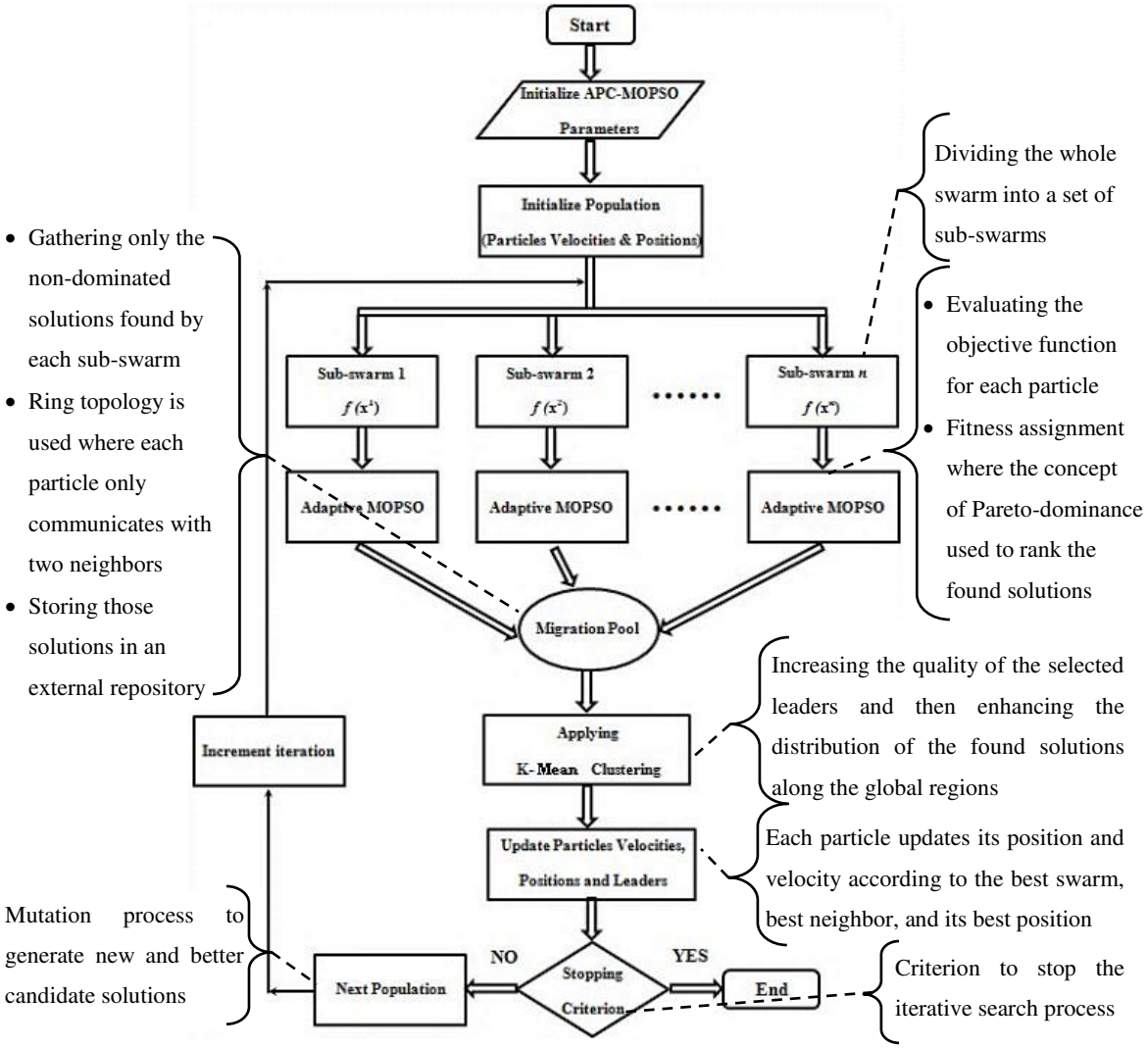


Figure 4.15: Flowchart of the Proposed APC-MOPSO

It is important to describe some of the mechanisms associated with the flow of the proposed APC-MOPSO algorithm shown in the above figure:

- As discussed in Section 4.3.1, a new dynamic model is proposed to increase the exploration ability of the particles in a way that enables particles to be guided not by a single leader, but a set of leaders during their flight. Accordingly, three leaders that are: (1) the best position (i.e., solution) in the swarm (g_{best}), (2) the particles' previously best position (p_{best}), and (3) the best closest position in the neighbor (l_{best}) are proposed.
- To automatically adjust the values of the APC-MOPSO's parameter, including inertia weight (w), cognitive (C_1), social (C_2), contiguous (C_3), velocity (C_v), position (C_p), mutation probability (p_m), boundaries of the search space ($DS_{reduction}$), and the size of the external

repository, new adaptive techniques are proposed (as illustrated in Section 4.3.2). These techniques enable the APC-MOPSO to further promote the search under various conditions.

- One way to preserve a reasonable number of non-dominated solutions during the search process is to use an external repository mechanism (as described in Section 4.3.2.6). Eventually, this mechanism archives only the good solutions found after each iteration.
- To maintain this repository, an update process is performed during each iteration of the algorithm. Accordingly, the good solutions (i.e., that are not dominated by any archived solutions) will be stored and those which are no longer non-dominated will be deleted from the repository.
- A modified K-Means^{++} clustering is proposed to eliminate some of the archived solutions from the repository. When the repository is full and there is no space to insert new non-dominated solutions, K-Means deletes some of the stored solutions to give a chance for good solutions to archive in the repository (as explained in Section 4.3.3). In this way, the external repository with a modified K-Means^{++} clustering helps in maintaining a set of non-dominated solutions until the end of the run.
- Having assigned leaders to guide the search, particles in each sub-swarm starts to perform its own search using the concept of parallel islands. In this approach, each sub-swarm explores different Pareto-front regions. For more efficient and fast function evaluations, parallel computing is proposed to shorten the computation time as well as to enhance the performance of the search process (as illustrated in Section 4.3.4).

As shown in the previous figure, the main steps of the proposed APC-MOPSO algorithm are described. For the sake of completeness, another diagram of the proposed APC-MOPSO algorithm is shown in Figure 4.16 to explain in detail its cycle run similar to that of MOPSO paradigms mentioned in Chapter 2. The underlining notions of the proposed APC-MOPSO are described for each step given in the diagram:

1. In the first step, all algorithm design parameters (i.e., mentioned above) are initialized. Particles (swarm) are randomly initialized in the search space (i.e., set the position and velocity for each particle), and the three leaders (g_{best}, l_{best}, and p_{best}) are selected based on the particles current locations (i.e., with respect to each objective in the problem). The external repository is filled with all initialized swarm best position (g_{best}).
2. Leaders are chosen from the repository to guide the search of the particles. The criterion is based on the fitness values of the particles such that particles with higher fitness are selected over the less fitness ones. Those leaders then guide the particles in each sub-swarm to explore regions that are less explored in the search space. The velocity for the particles in each sub-swarm is updated as shown in Section 4.3.2.3.
3. The position of the particles is updated according to their new velocity calculated in the previous step (as expressed in Section 4.3.2.3).
4. The fitness value for each particle in the sub-swarms is evaluated.

5. Having ranked all solutions in each sub-swarm, non-dominated solutions from each sub-swarm are migrated to the solutions pool using the concept of Pareto-dominance and ring topology. Then, the repository is updated with the current solutions found by the particles.
6. The particles' best position is updated using the Pareto-dominance criterion; *if* the current location of the particle dominates the stored one, *then* the current position is considered instead.
7. Finally, a new generation of particles is produced using an adaptive mutation technique as explained in Section 4.3.2.4. The new swarm (i.e., a combination of both new and old particles) is then distributed into sub-swarms to perform a new execution of the search process.

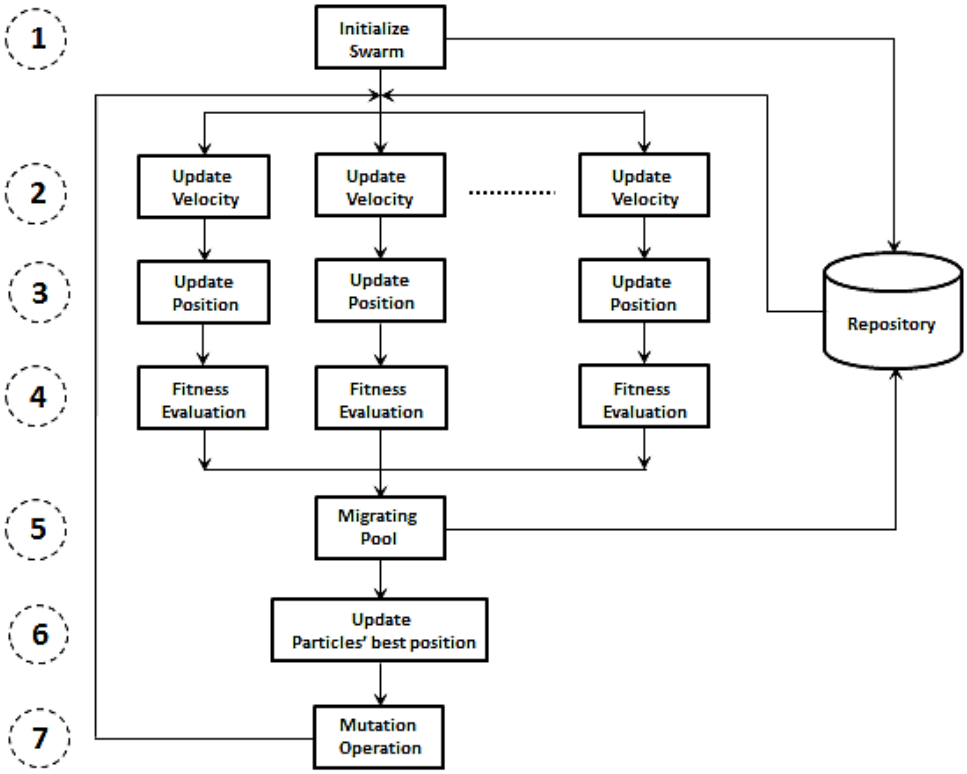


Figure 4.16: The Simplified Diagram of the APC-MOPSO Algorithm

The features and advantages of the proposed APC-MOPSO algorithm are outlined as follows:

- the ability of automatically adjusting the design parameters of the APC-MOPSO algorithm (i.e., using new adaptive schemes);
- the ability of flexibly selecting leaders (i.e., using the new dynamic model);

- the ability of guiding the particles towards the global regions (i.e., using the external repository);
- the ability of maintaining a diverse Pareto-front set of solutions in the search space (i.e., using a mutation operator technique);
- the ability of fast converging (i.e., using the concept of parallel islands),
- the ability of shortening the computation time (i.e. using a parallel computing technique); and
- the ability of producing high quality Pareto-front solutions (i.e., using the concept of Pareto-dominance).

4.4 Experiments and Comparison of Results

4.4.1 Competitive Multi-objective Optimization Algorithms

To compare the performance of the proposed APC-MOPSO, five other well-known optimization algorithms in the multi-objective literature are used. Those algorithms are all developed based on the notion of Pareto optimum, and they are: Strength Pareto Evolutionary Algorithm (SPEA2) [110,74], Non-dominated Sorting Genetic Algorithm (NSGA-II) [233], Multi-objective PSO (MOPSO) [52], Two-local Best MOPSO (2LB-MOPSO) [243], and Parallel Elite Genetic Algorithm (PEGA) [253]. Brief descriptions of the algorithms are provided in this section.

1. Parallel Elite Genetic Algorithm (PEGA)

A new Parallel Elite Genetic Algorithm (PEGA) for a global optimization problem is proposed by Hsu-Chih et al. [253]. The core operators of the proposed Elite GA (EGA) are: selecting, crossover, mutation, fitness function, reproduction strategy with elite policy, and diversity pool. A brief explanation of these operators is provided as follows:

- *Selection*: the task of this process is to select individuals from the current population so that they can be sent to crossover and mutation. This operator is considered one of the key operators to ensure survival of the fitness.
- *Crossover*: it is implemented by the exchange of strings between two parent chromosomes. This process is important to find the optimal solution.
- *Mutation*: is the process of making random alteration to the bit(s) of the chromosomes. This process is necessary for maintaining diversity in the population.
- *Fitness Function*: is the process of evaluating the fitness of new chromosomes generated by operators such as crossover and mutation.

- *Reproduction Strategy with Elite Policy*: it is used to avoid destroying the best solution in the next generation (i.e., offsprings have fitness worse than parents do). This is done by taking elite preservation policy into account, in which 20% of the old population is preserved in the next generation.
- *Diversity Pool*: this process is used to increase the diversity in the population. This is implemented by adding a random population of 20% to the new generation. This operator increases the ability of EGA to find global solutions.

The PEGA is composed of two parallel EGA algorithms which are concurrently executed on multi-core processor. The PEGA is proposed to take advantage of preserving diversity, avoiding premature convergence, and faster computing than conventional or serial GAs do. Figure 4.17 depicts the architecture of the PEGA.

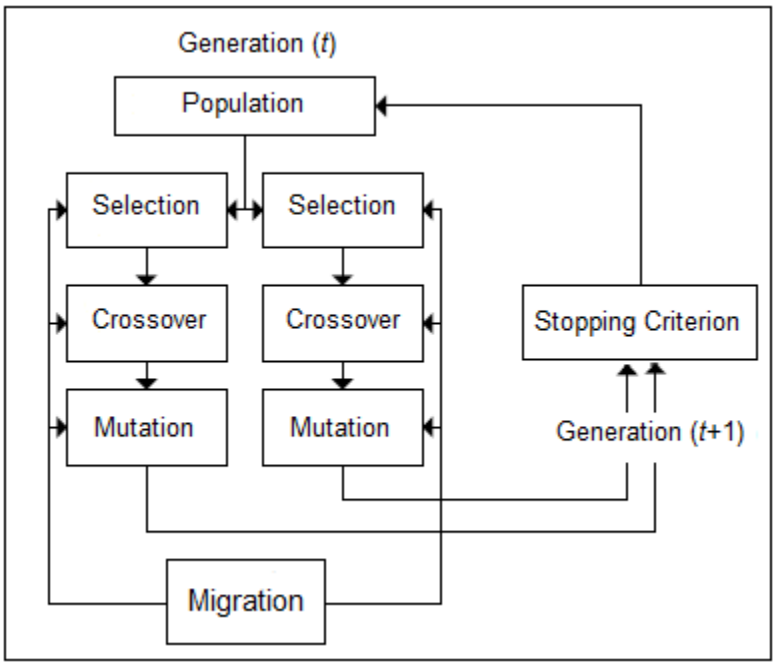


Figure 4.17: Schematic of Parallel Elite Genetic Algorithm

The PEGA is outlined by the following pseudo-code.

Pseudo-code 18: Parallel Elite Genetic Algorithm (PEGA)

for each individual
DO
- Step1. Randomly generate the population.
end for
Step2. Set the two parents from the population-based tournament selection.

Step3. Execute the procedure of crossover
while (until maximum number of function evaluations is NOT reached)
for each chromosome
Do
while
if the new chromosome is acceptable
return
else
- Repeated until acceptable chromosomes are obtained.
end if
end while
end for
Step4. Perform the mutation process
Step5. Repeat the above steps until the convergence is reach.
end while

2. Strength Pareto Evolutionary Algorithm (SPEA2)

A modified version of Strength Pareto Evolutionary Algorithm, called as SPEA2, has been proposed by Zitzler et al. [110,74]. The algorithm introduced elitism by an external repository on non-dominated solutions at all time. It starts by randomly initializing the population (P) and an external repository of size (P'). At every iteration, the non-dominated solutions from (P) are ranked based on their fitness and best individuals are copied from the population (P) to the external repository (P'). The external repository is updated by eliminating archived members that are no longer non-dominated and store the new individuals instead. This mechanism allows the external repository to keep only the best solutions found over time. When the external repository is full, a new archive truncation method (proposed by Zitzler et al. [254]) is adopted to determine which solutions from the external repository (P') will be retained. Figure 4.18 shows the flow of the SPEA2 algorithm.

To generate offsprings, SPEA2 evaluates the fitness of all individuals by using a nearest neighbor density estimation technique (proposed by Terrell et al. [255]) and uses genetic operators (i.e., mutation and crossover, [3]), and tournament selection, [256], to create a new population.

The equation used to determine the fitness (i.e., called strength) of an individual i in the external repository is given as follows.

$$S_i = \frac{n_i}{N+1}$$

where n_i is the number of solutions that a solutions i dominates in the population, and N is the size of the population.

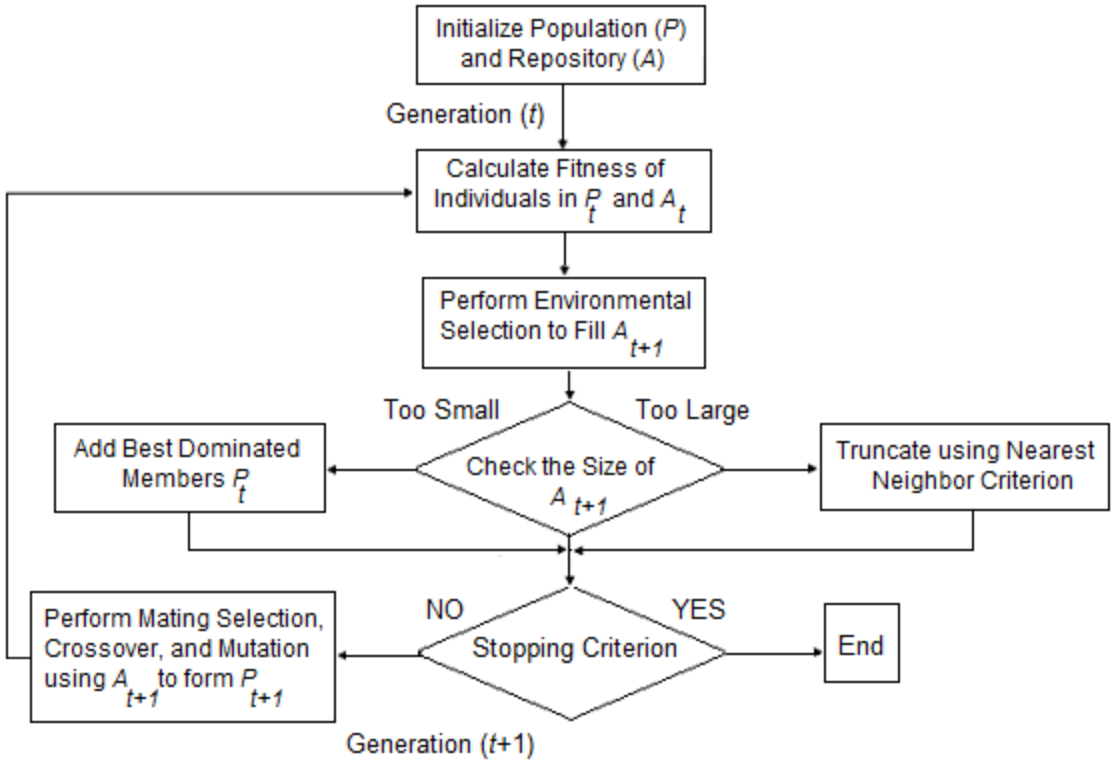


Figure 4.18: Flowchart of SPEA2 Algorithm

The outline of SPEA2 algorithm is given in Pseudo-code 19.

Pseudo-code 19: Strength Pareto Evolutionary Algorithm (SPEA2)

Initialize population (P) with randomly generated solutions
Evaluate the fitness value of each solution
Create empty external repository (P')
while (until stopping creation is NOT satisfied)
Store (i.e., copy) non-dominated individuals of (P) to (P')
Delete elements from (P) which are dominated by any other solutions of (P')
if (P') is FULL (exceeds its maximum capacity)
Decide which elements to be removed by means of the truncation operator
end if
Evaluate the fitness value of individuals from (P) and (P')
Apply binary tournament selection to select individuals from ($P + P'$)
Apply crossover (i.e., single point operation)
Apply mutation
end while

3. Non-dominated Sorting Genetic Algorithm (NSGA-II)

In 2001, Deb et al. proposed the second version of their elitist Non-dominated Sorting Genetic Algorithm that is referred to as NSGA-II, [233]. NSGA-II is developed based on several layers of classifications of the individuals (i.e., solutions) as suggested by Goldberg, [3]. The algorithm begins by initializing a random population of size (P), and then, generates an offspring population of size (Q) using the crowded tournament selection, crossover, and mutation operators (as shown in Figure 4.19). Both populations are combined (size of $2N$) in order to select solutions for a new population in the next generation as depicted in Figure 4.20.

The main mechanism of NSGA-II involves a non-dominated crowding sort process; a process in which the individuals in the population are ranked based on the concept of non-domination: all non-dominated individuals are classified into one category. Accordingly, a crowding distance is calculated for the individuals in each front (see Figure 4.20). In an NSGA-II algorithm, a sorting mechanism is used to select the solutions of the new population in the next iteration. Individuals in the fronts closer to the Pareto-optimal front always have a higher priority rather than those in the far fronts. In this way, the crowding distance assignment approach keeps diversity without specifying any additional parameters.

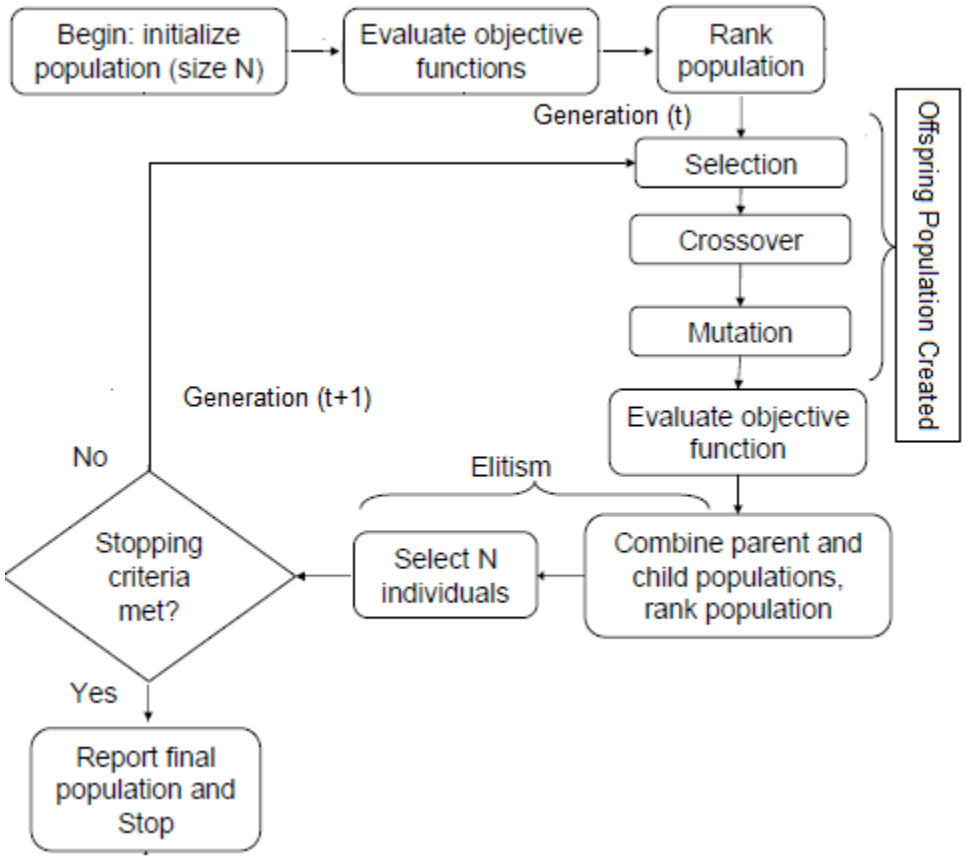


Figure 4.19: Flowchart of NSGA-II Algorithm

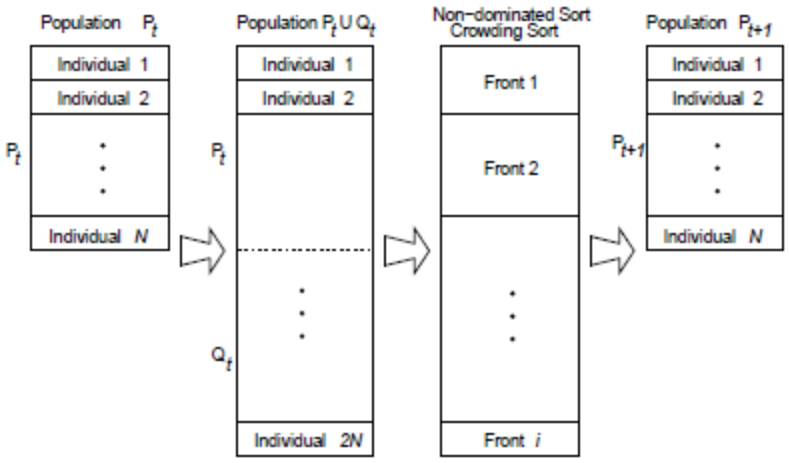


Figure 4.20: Schematic of Sorting and Selecting the Solutions – NSGA-II Algorithm [73]

The main steps of NSGA-II algorithm are given in the following pseudo-code.

Pseudo-code 20: Non-dominated Sorting Genetic Algorithm (NSGA-II)

Initialize population with randomly generated solutions (P)
Evaluate the fitness value of each solution
- Apply Selection operator
- Apply Crossover operator
- Apply Mutation operator
- Create the offspring population (Q)
while
- Apply rank based on Pareto-dominance ($R \leftarrow P \cup Q$)
- Compute niche count (R)
- Apply Crowding Sort (R)
- Apply selection via stochastic universal sampling (Roulette Wheel)
- Apply single point crossover
- Apply mutation
- Evaluate new candidates ($P \leftarrow P\ and\ R$)
end while (until stopping creation is satisfied)

4. Multi-objective Particle Swarm Optimization Algorithm (MOPSO)

Coello et al. (2004) proposed a modified version of the standard PSO that can solve Multi-objective Optimization Problems (MOOPs) referred to as Multi-objective PSO algorithm (MOPSO), [52]. In this technique, an external repository to store non-dominated solutions is employed. Then, the external repository is used to select the best solution (g_{best}) as a leader to guide the search of the particles towards the Pareto-front and to provide a mechanism that

maintains diversity along the Pareto-front. The fitness of the archived members is evaluated for every iteration. A geographically based technique is incorporated to keep only solutions that have the highest fitness values in the external repository. In order to improve diversity, MOPSO approach uses a mutation operator that acts on the particles of the swarm. The flow of the MOPSO algorithm is shown in Figure 4.21.

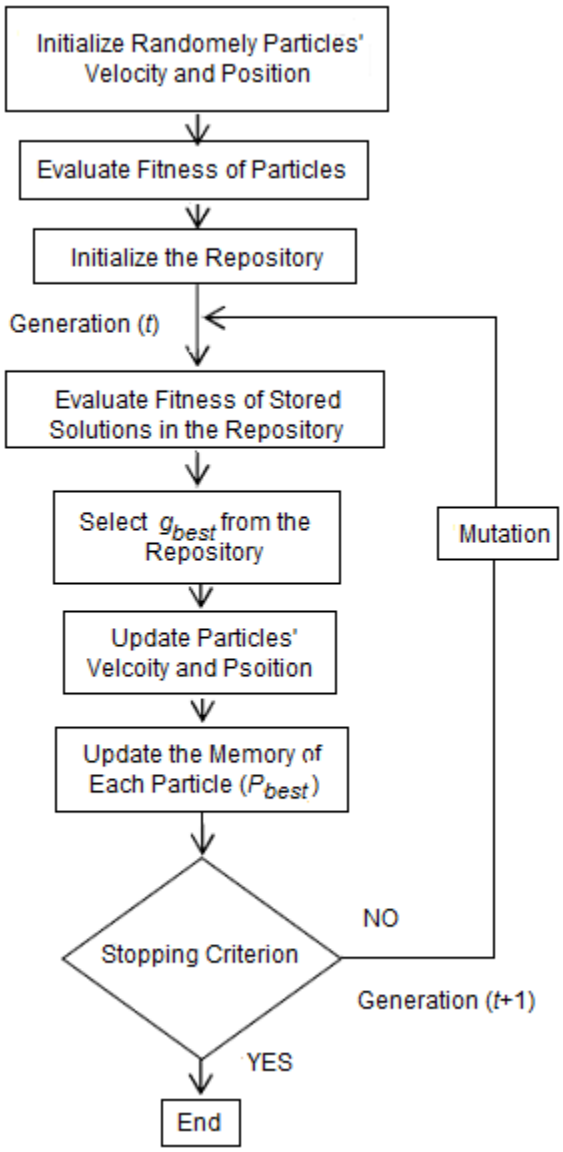


Figure 4.21: Flowchart of the MOPSO Algorithm

The outline of the MOPSO is provided in Pseudo-code 21.

Pseudo-code 21: Multi-objective Particle Swarm Optimization Algorithm (MOPSO)

for each particle
Do
- Randomly initialize the position $\vec{x}$ using Eq. (4.1)
- Randomly initialize the velocity using Eq. (4.2)
- Evaluate fitness function($\vec{x}$)
- Initialize $p_{best} = \vec{x}$
end for
while (until maximum number of iterations is NOT reached)
for each particle
Do
- Compute the new velocity on each dimension (along each objective)
- Compute the new position on each dimension (along each objective)
- Evaluate the fitness using the new position
end for
- Update the solutions in the repository *REP*
if $\vec{x}$ dominates to P_{best}
$P_{best} = \vec{x}$ (update leaders to guide the rest of the swarm during their search)
end if
end while

5. Two-*lbests* based Multi-objective Particle Swarm Optimizer (2LB-MOPSO)

Two-lbests MOPSO algorithm (2LB-MOPSO) is proposed by Zhao et al. in 2010 using a new elitism mechanism that maintains diversity even when there are few non-dominated solutions in the external repository, [243]. This approach suggested a new form of leaders that is referred to as the local best (l_{best}) instead of the swarm best solution (g_{best}) commonly used in the standard MOPSO algorithms. Two local best leaders are chosen from the non-dominated solutions in the external repository. The concept of non-domination sorting is applied to obtain the solutions generated in every iteration. During this process, the new solutions are ranked (i.e., from the lowest to the highest fronts (*front1* to *front 4*) as shown in Figure 4.22) and retained two indicators, namely the front rank and the crowding distance value, [73]. Only the solutions with the lowest front rank will be stored in the external repository. When the size of the external archive reaches its maximum capacity, the crowding distance is applied to select the required number of solutions to be stored in the external repository for the next iteration (i.e., from the lowest front that still has unselected solutions in the current iteration). As each particle is guided by two neighborhoods (i.e., in decision and objective spaces) with the lowest front (e.g., *front 1*), the flight of each particle will be in the direction between the positions of the two l_{best} *leaders*. Unlike the search behavior of the MOPSOs, 2LB-MOPSO is proposed to focus the search around

small regions in the decision space in the vicinity of the best fronts. The main procedure of the 2LB-MOPSO algorithm is illustrated in Pseudo-code 22.

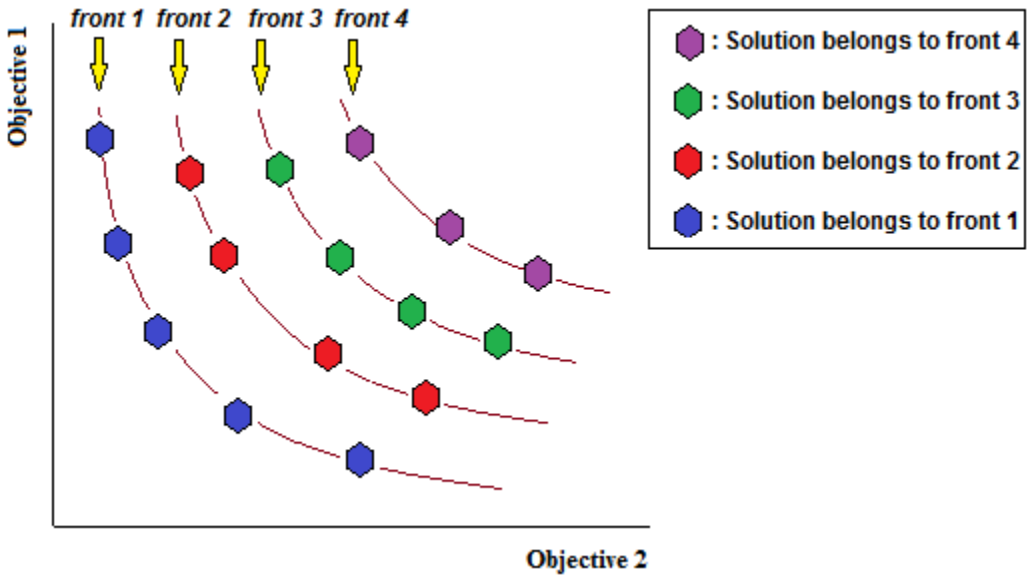


Figure 4.22: An Example of Ranking the Solutions in the External Repository – 2LB-MOPSO

Pseudo-code 22: Two-*lbests* Multi-objective PSO Algorithm (2LB-MOPSO)

for each particle
Do
- Set initial position $\vec{x}$ (0)
- Randomly initialize the velocity
- Evaluate fitness value
- Initialize $l_{best} = \vec{x}(0)$
end for
Iteration-count = 1
while (until maximum number of function evaluations is NOT reached)
for each particle
Do
if iteration-count > 5
- Randomly choosing an objective
- Randomly choosing a bin of the selected objective
- Choosing the solutions that have the lowest front and the largest crowding distance in the chosen bin
- Select the second l_{best} from the neighborhood of the first l_{best} in the decision space

else

if $\vec{x}$ dominates to l_{best}

$l_{best} = \vec{x}$

- Update the two leaders l_{best} in the same bins used in the last iteration-count

end if

end if

- Update the velocity on each dimension (along each objective)

$$V(t+1) = w \times [V(t) + C_1 \times rand(\ ) \times \big(lbest1(t) - pbest(t)\big) + C_2 \times rand(\ ) \times \big(lbest2(t) - pbest(t)\big)]$$

- Limit the velocity for each dimension (d)

$$V_i(d) = (\min\big(V_{max}(d), V_i(d)\big) \,\&\, max(\min\big(-V_{max}(d), V_i(d)\big)$$

- Update the position on each dimension (d) (along each objective)

$$x_i(d) = x_i(d) + V_i(d)$$

if position exceeds the search space

Set a new search space boundary

end if

Evaluate the fitness using the new position (Q)

end for

- Sort non-dominated solutions on combined (P+ Q)

- Update the solutions in the external repository for the next iteration

end while

4.4.2 Test Problems

To test the performance of multi-objective optimization algorithms, there are several benchmark test problems whose Pareto-optimal front sets are known. In this work, three groups of benchmark test problems are considered: the first group contains bi-objective problems, and is used to assess the performance of the optimization algorithm in terms of its global convergence ability to different Pareto-optimal front geometry shapes, including convex, non-convex, linear, non-linear, continuous, disconnect, unimodal, and multimodal. Zitzler et al. [92], developed a set of bi-objective benchmark test problems, namely ZDT1[16], ZDT2, ZDT3, ZDT4, and ZDT6. Moreover, Deb et al. [257], identified another set of benchmark test problems, namely DTLZ1[17], DTLZ2, DTLZ3, DTZL4, DTLZ5, and DTZL6 for MOEA and MOPSO algorithms. Lastly, T. Okabe [258], F. Kursawe [259], and C. Fonseca [202] proposed bi-objective benchmark test

[16] ZDT stands for the initials of the authors Zitzler, Deb and Thiele.

[17] DTLZ stands for the initials of the authors Deb, Thiele, Zitzler, and Laumanns.

problems that are very challenging and are difficult to solve for most multi-objective optimization technique, [260].

The second group contains three-objective problems, namely DTLZ2, DTLZ5, and DTLZ7. These problems are used to evaluate the performance of the optimization algorithm in terms of its ability to finding solutions that cover the entire Pareto-optimal front. The third group contains five-objective problems (e.g., DTLZ2 and DTYLZ5) that are used to measure the performance of the optimization algorithm in terms of its computational speed and solution quality.

All these benchmark problems are used to validate and compare the performance of the APC-MOPSO algorithm. A Summary of these test problems is provided as follows.

- *ZDT1 two-objective Test Problem*

  Has a convex Pareto-optimal front with 30 decision variables, Zitzler et al. [92].

Geometry	*m*	Constraints	Formulations
Convex	30	$0 \le x_i \le 1$	$f_1(\vec{x}) = x_1,$ $f_2(\vec{x}) = g(\vec{x})\left[1 - \sqrt{x_1/g(\vec{x})}\right]$ $g(\vec{x}) = g1 + 9\left(\sum_{i=2}^{m} x_i\right)/(m-1)$

- *ZDT2 two-objective Test Problem*

  Has a non- convex Pareto-optimal front with 30 decision variables, Zitzler et al. [92].

Geometry	*m*	Constraints	Formulations
Non-convex	30	$0 \le x_i \le 1$	$f_1(\vec{x}) = x_1,$ $f_2(\vec{x}) = g(\vec{x})\left[1 - \left(x_1/g(\vec{x})\right)^2\right]$ $g(\vec{x}) = g1 + 9\left(\sum_{i=2}^{m} x_i\right)/(m-1)$

- *ZDT3 two-objective Test Problem*

  Has a disconnected, non-contiguous, and convex Pareto-optimal front with 30 decision variables, Zitzler et al. [92].

Geometry	*m*	Constraints	Formulations
Convex and dis-continuous	30	$0 \le x_i \le 1$	$f_1(\vec{x}) = x_1,$ $f_2(\vec{x}) = g(\vec{x})\left[1 - \sqrt{\frac{x_1}{g(\vec{x})}} - \frac{x_1}{g(\vec{x})}\sin(10\pi x_1)\right]$

	$g(\vec{x}) = 1 + 9\left(\sum_{i=2}^{m} x_i\right)/(m-1)$

- *ZDT4 two-objective Test Problem*

Has a convex and multimodal Pareto-optimal front with 10 decision variables, Zitzler et al. [92]. It contains 21^9 local Pareto-optimal fronts, and therefore, used to test the optimization algorithm's ability to deal with multimodality.

Geometry	*m*	Constraints	Formulations
Convex and multi-modal	10	$0 \le x_1 \le 1$ $-5 \le x_i \le 5$	$f_1(\vec{x}) = x_1,$ $f_2(\vec{x}) = g(\vec{x})\left[1 - \left(x_1/g(\vec{x})\right)^2\right]$ $g(\vec{x}) = 1 + 10(m-1) + \sum_{i=2}^{m} x_i\left(x_i^2 - 10\cos(4\pi x_i)\right)$

- *ZDT6 two-objective Test Problem*

Has a non-convex and multimodal Pareto-optimal front with 10 decision variables, Zitzler et al. [92]. Due to the non-uniformity of the search space, solving this test problem includes a difficulty that the density of the solutions are lowest close to the Pareto-optimal front and highest away from it.

Geometry	*m*	Constraints	Formulations
Non-convex	10	$0 \le x_i \le 1$	$f_1(\vec{x}) = 1 - exp(-4x_1).sin^6(6\pi x_1),$ $f_2(\vec{x}) = g(\vec{x}).\left[1 - \left(\frac{f_1}{g(\vec{x})}\right)^2\right]$ $g(\vec{x}) = 1 + 9\left[\left(\sum_{i=2}^{m} x_i\right)/(m-1)\right]^{0.25}$

- *FONSECA two-objective Test Problem*

Has a non-convex and Pareto-optimal front with 3 decision variables, C. Fonseca [202].

Geometry	*m*	Constraints	Formulations
Concave	3	$-4 \le x_i \le 4$	$f_1(\vec{x}) = 1 - exp\left(-\sum_{i=1}^{m}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right)$ $f_2(\vec{x}) = 1 - exp\left(-\sum_{i=1}^{m}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right)$

- *OKA2 two-objective Test Problem*

Has a convex and Pareto-optimal front with 3 decision variables, T. Okabe [258]. This test problem is very difficult for any MOEA and MOPSO to solve, since the closer the individuals (or particles) get to the Pareto-front, the sparser the probability density becomes.

Geometry	*m*	Constraints	Formulations
Convex	3	$-\pi \le x_1 \le \pi$ $-5 \le x_i \le 5$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = 1 - \frac{1}{4\pi^2}(x_1 + \pi) + \lvert x_2 - 5\,cos(x_1)\rvert^{1/3} + \lvert x_3 - 5\,sin(x_1)\rvert^{1/3}$

- *KURSAWE two-objective Test Problem*

Has several disconnected and unsymmetrical global area in the search space. The Pareto-optimal front of this test problem consists of three disconnected Pareto curves with 3 decision variables, F. Kursawe [259].

Geometry	*m*	Constraints	Formulations
Non-convex and non-connected	3	$-5 \le x_i \le 5$	$f_1(x) = \sum_{i=1}^{m-1}\left(-10e^{(0.2)*\sqrt{x_i^2 + x_{i+1}^2}}\right)$ $f_2(x) = \sum_{i=1}^{m}(\lvert x_i\rvert^{0.8} + 5\,sin(x_i)^3)$

- *DTLZ1 M-objective Test Problem*

It is a scalable test problem that has a linear and multimodal Pareto-optimal front with 12 decision variables, Deb et al. [257]. The difficulty in this problem is to converge to the hyper-plane, as the search space contains $(11^5 - 1)$ local Pareto-optimal fronts.

Geometry	*m*	Constraints	Formulations
Linear and Multi-modal	12	$0 \le x_i \le 1$	$f_1(\vec{x}) = \frac{1}{2}x_1x_2\big(1 + g(\vec{x})\big)$ $f_2(\vec{x}) = \frac{1}{2}x_1(1 - x_2)\big(1 + g(\vec{x})\big)$ $f_3(\vec{x}) = \frac{1}{2}(1 - x_1)\big(1 + g(\vec{x})\big)$ $f_{M-1}(\vec{x}) = \frac{1}{2}x_1(1 - x_2)\big(1 + g(\vec{x})\big)$ $f_M(\vec{x}) = \frac{1}{2}(1 - x_2)\big(1 + g(\vec{x})\big)$

$$g_m(\vec{x}) = 100\left[|X_m| + \sum_{i=3}^{m}(x_i - 0.5)^2 - cos\big(20\pi(x_i - 0.5)\big)\right]$$

- *DTLZ2 M-objective Test Problem*

It is a scalable test problem that has a non-convex Pareto-optimal front with 12 decision variables, Deb et al. [257]. It is used to investigate the algorithm's ability to scale up its performance with a large number of objectives. In this problem, the summation of all objective function values must satisfy the condition given in Eq. (4.28).

$$\sum_{i=1}^{M}(f_\mathrm{i})^2 = 1 \qquad 4.28$$

Geometry	*m*	Constraints	Formulations
Non-convex	12	$0 \le x_i \le 1$	$f_1(\vec{x}) = \big(1 + g(\vec{x})\big)cos(x_1\pi/2)cos(x_2\pi/2)$ $f_2(\vec{x}) = \big(1 + g(\vec{x})\big)cos(x_1\pi/2)sin(x_2\pi/2)$ $f_3(\vec{x}) = \big(1 + g(\vec{x})\big)sin(x_1\pi/2)$ $f_3(\vec{x}) = \big(1 + g(\vec{x})\big)sin(x_1\pi/2)$ $f_{\mathrm{M}-1}(\vec{x}) = \big(1 + g(\vec{x})\big)cos\left(\frac{x_1\pi}{2}\right)sin\left(\frac{x_2\pi}{2}\right)$ $f_{\mathrm{M}}(\vec{x}) = \big(1 + g(\vec{x})\big)sin(x_1\pi/2)$ $g_m(x) = \sum_{i=3}^{m}(x_i - 0.5)^2$

- *DTLZ3 M-objective Test Problem*

It is a scalable test problem that has a concave and multimodal Pareto-optimal front with 12 decision variables, Deb et al. [257]. Moreover, the problem has $(3^{10} - 1)$ local Pareto-optimal fronts.

Geometry	*m*	Constraints	Formulations
Concave	12	$0 \le x_i \le 1$	$f_1(\vec{x}) = \big(1 + g(\vec{x})\big)cos(x_1\pi/2)cos(x_2\pi/2)$ $f_2(\vec{x}) = \big(1 + g(\vec{x})\big)cos(x_1\pi/2)sin(x_2\pi/2)$ $f_3(\vec{x}) = \big(1 + g(\vec{x})\big)sin(x_1\pi/2)$ $f_{\mathrm{M}-1}(\vec{x}) = \big(1 + g(\vec{x})\big)cos\left(\frac{x_1\pi}{2}\right)sin\left(\frac{x_2\pi}{2}\right)$ $f_M(\vec{x}) = \big(1 + g(\vec{x})\big)sin(x_1\pi/2)$ $g_m(x) = 100\left[|X_m| + \sum_{i=3}^{m}(x_i - 0.5)^2 - cos\big(20\pi(x_i - 0.5)\big)\right]$

- *DTLZ4 M-objective Test Problem*

It is a scalable test problem that has a concave and unimodal Pareto-optimal front with 12 decision variables, Deb et al. [257].

Geometry	*m*	Constraints	Formulations
Concave	12	$0 \le x_i \le 1$ $\alpha = 100$	$f_1(\vec{x}) = \big(1 + g(\vec{x})\big)cos\left(\frac{x_1^{\propto}\pi}{2}\right)cos\left(\frac{x_2^{\propto}\pi}{2}\right)$ $f_2(\vec{x}) = \big(1 + g(\vec{x})\big)cos\left(\frac{x_1^{\propto}\pi}{2}\right)sin\left(\frac{x_2^{\propto}\pi}{2}\right)$ $f_3(\vec{x}) = \big(1 + g(\vec{x})\big)sin(x_1\pi/2)$ $f_{M-1}(\vec{x}) = \big(1 + g(\vec{x})\big)cos\left(\frac{x_1^{\propto}\pi}{2}\right)sin\left(\frac{x_2^{\propto}\pi}{2}\right)$ $f_M(\vec{x}) = \big(1 + g(\vec{x})\big)sin(x_1\pi/2)$ $g(\vec{x}) = \sum_{i=3}^{m}(x_i - 0.5)^2$

- *DTLZ5 M-objective Test Problem*

It is a scalable test problem that has a concave and unimodal Pareto-optimal front with 12 decision variables, Deb et al. [257]. Same condition given in Eq. (4.28) is applied, in which the summation of the square objective values must equal to 1.

Geometry	*m*	Constraints	Formulations
Non-convex and uni-modal	12	$0 \le x_i \le 1$	$f_1(\vec{x}) = \big(1 + g(\vec{x})\big)cos(\theta_1\pi/2)cos(\theta_2\pi/2)$ $f_2(\vec{x}) = \big(1 + g(\vec{x})\big)cos(\theta_1\pi/2)sin(\theta_2\pi/2)$ $f_3(\vec{x}) = \big(1 + g(\vec{x})\big)sin(\theta_1\pi/2)$ $f_{M-1}(\vec{x}) = \big(1 + g(\vec{x})\big)cos(\theta_1\pi/2)sin(\theta_2\pi/2)$ $f_M(\vec{x}) = \big(1 + g(\vec{x})\big)sin(\theta_1\pi/2)$ $g(\vec{x}) = \sum_{i=3}^{m}(x_i - 0.5)^2$ $\theta_i = \frac{\pi}{4\big(1 + g(\vec{x})\big)}(1 + 2g(\vec{x})x_i)$ *for i=2, 3,…, (M-1)*

- *DTLZ6 M-objective Test Problem*

It is a scalable test problem that has a concave and unimodal Pareto-optimal front with 22 decision variables, Deb et al. [257].

Geometry	*m*	Constraints	Formulations
Concave and uni-modal	22	$0 \le x_i \le 1$	$f_1(\vec{x}) = \big(1 + g(\vec{x})\big)cos(\theta_1\pi/2)cos(\theta_2\pi/2)$ $f_2(\vec{x}) = \big(1 + g(\vec{x})\big)cos(\theta_1\pi/2)sin(\theta_2\pi/2)$ $f_3(\vec{x}) = \big(1 + g(\vec{x})\big)sin(\theta_1\pi/2)$ $f_{M-1}(\vec{x}) = \big(1 + g(\vec{x})\big)cos\left(\frac{\theta_1\pi}{2}\right)sin\left(\frac{\theta_2\pi}{2}\right)$ $f_M(\vec{x}) = \big(1 + g(\vec{x})\big)sin(\theta_1\pi/2)$ $\theta_i = \frac{\pi}{4\big(1 + g(\vec{x})\big)}(1 + 2g(\vec{x})x_i)$ *for i=2, 3,…, (M-1)* $g(\vec{x}) = \sum_{i=3}^{m}(x_i)^{0.1}$

- *DTLZ7 M-objective Test Problem*

It is a scalable test problem that has four disconnected Pareto-optimal fronts with 20 decision variables, Deb et al. [257]. This problem is used to test the ability of the optimization algorithm to maintain and distribute sub-populations in different Pareto-optimal regions.

Geometry	*m*	Constraints	Formulations
non-connected	20	$0 \le x_i \le 1$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = x_2$ $f_3(\vec{x}) = \big(1 + g(\vec{x})\big)h(f_1, f_2, g)$ $f_{M-1}(\vec{x}) = x_{M-1}$ $f_M(\vec{x}) = \big(1 + g(\vec{x})\big)h\big(f_1, f_2, \dots, f_{M-1}, g\big)$ $g(\vec{x}) = 1 + 9/\lvert X_m\rvert\sum_{i=3}^{m} x_i$ $h(f_1, f_2, g) = M - \sum_{i=1}^{M}\left[\frac{f_i}{1 + g(\vec{x})}\right].\big(1 + sin(3\pi f_i)\big)$

4.4.3 Numerical Settings

This subsection investigates several key features of the proposed APC-MOPSO algorithm, as well as the effect of different parameter (i.e. numerical) settings on the quality of the solutions and the rate of convergence to determine their influence. The performance of the proposed APC-MOPSO algorithm is compared to other state-of-the-art MOEAs and MOPSOs. Accordingly, each experiment for every benchmark test problem is repeated 25-times (i.e., to

ensure consistency) and then the average is taken for validating and measuring the performance of each method. The configurations and the numerical settings of the proposed APC-MOPSO, MOPSO, PEGA, NSGA-II, SPEA2, and 2LB-MOPSO are provided in Tables 4.2 to 4.5.

Table 4.2: Configurations of the Design Parameters for the APC-MOPSO

Design Parameters	Lower Limit	Upper Limit	Remark
Inertia weight (w)	0.4	2.5	Dynamically changes
Social factor (C_1)	1.5	2.75	Exponentially changes
Cognitive factor (C_2)	1.5	2.75	Exponentially changes
Contiguous factor (C_3)	1.5	2.75	Exponentially changes
Position coefficient (C_p)	0.1	1	Linearly changes
Velocity coefficient (C_v)	0.1	1	Linearly changes

Table 4.3: Configurations of the Algorithm Parameters for the APC-MOPSO

Design Parameters	Numerical Value	Remarks
Maxiter	100	Maximum iterations
Maxeval	10,000	Maximum number of function evaluations
Swarm_size	100	The size of the swarm
Repository_size	variable	The size of external repository, it is computed using Eqs. (4.21)-to-(4.25)
Random	[0, 1]	Uniform distribution
Clus_num	variable	No. of clustering group = No. of objectives
MigrationInterval	1500	Executing migration every 1500 FEs
MigrationFraction	0.2	The percentage of the migrated particles
MigrationDirection	'forward'	Particles migrate towards ahead sub-swarms

Table 4.4: Architectural Specifications for the APC-MOPSO Algorithm

Features	Numerical Value	Remarks
Parallelism	variable	No. of parallel paths = No. of objectives
Cooperative topology	Ring	The communication among particles
K-Means	Probability-based	No. of centers = No. of the objectives
Subswarm_size	*Swarm_size/Clsu_num*	Number of particles in each sub-swarm

Table 4.5: Parameter Settings for MOPSO, SPEA2, NSGA-II, PEGA, and 2LB-MOPSO

Parameterization used in MOPSO	
Population Size	100 particles
Repository Size	100 particles
Maximum Iteration	100
phi1	2.05
phi2	2.05
Grid inflation parameter	0.1
Parameterization used in MOPSO	
Number of grids per each dimension	10
Leader selection pressure parameter	4
Repository member selection pressure	2
Parameterization used in SPEA2	
Maximum number of generation	100
Population size	100
Number of individuals for tournament	2
Individual mutation probability	1
Variable recombination probability	1
Individual recombination probability	1
Variable swap probability	0.5
Eta mutation	15
Eta recombination	5
Parameterization used in NSGA-II	
Population Size	100
Maximum iteration	100
Crossover ratio	0.8
Mutation ratio	0.3
Number of parents	80
Number of mutation	30
Parameterization used in PEGA	
Population Size	100 individuals (50 - 50)
Pareto Fraction	1
Maximum generation	100
Crossover Fraction	0.8
Migration Direction	'forward'
Migration Interval	20
Migration Fraction	0.2
Parameterization used in 2LB-MOPSO	
Number of count	5
Number of bins	10
Population size	100 particles
Maximum function evaluations	10,000
Archive Size	100
Weight factor	0.729
Social factor	2.05
Cognitive factor	2.05

4.4.4 Performance Metrics

Comparing various heuristic optimization methods experimentally involves the notion of performance. In a multi-objective optimization, the quality of the solutions is substantially more complex than for single-objective optimization. This is because the optimization itself is multiple (i.e., multi-objective), including minimizing the distance between the non-dominated solutions and the Pareto-optimal front, a distribution (in most cases uniform) of the non-dominated solutions, and maximizing the extent of the non-dominated solutions for each objective.

A summary of the performance metrics used to assess the performance of the APC-MOPSO and other MOEA and MOPSO algorithms is provided in Table 4.6.

Table 4.6: Performance Metrics for Multi-objective Optimization Techniques

Spacing (S)	
Mathematical Formulation	**Remark**
$S = \sqrt{\frac{1}{n-1}\sum_{i+1}^{n}(\bar{d} - d_i)^2}$ where $\bar{d}$ is the mean of all d_i $d_i = min_j \left(\left\|f_1^i(\boldsymbol{x}) - f_1^j(\boldsymbol{x})\right\| + \left\|f_2^i(\boldsymbol{x}) - f_2^j(\boldsymbol{x})\right\|\right),$ $i, j = 1,2,\ldots,n$	• A value of zero is ideal for this metric, since it indicates all members of PF_{known} are equidistantly spaced (uniformly distributed).
Generational Distance (GD)	
Mathematical Formulation	**Remark**
$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}$ where d_i is the Euclidean distance between each member, i, of PF_{known} and the closest member in PF_{true} to that member, i, in the objective space	• If the value of GD indicates zero, this means that the algorithm is efficiently converged to the Pareto-optimal front (100% coincided). Conversely, any other value will indicate the degree of matching with the Pareto-optimal front (PF_{true}).
Error Ratio (ER)	
Mathematical Formulation	**Remark**
$ER = \frac{\sum_{i=1}^{n} e_i}{n}$	• If $e_i = \begin{cases} 1 & if \text{ solution } i \text{ is member of the true PF} \\ 0 & if \text{ solution } i \text{ is not member of the true PF} \end{cases}$ • Hence, if $ER = \begin{cases} 0 & every \text{ solution } i \text{ in } PF_{known} \text{ is in } PF_{true} \\ 1 & none \text{ solution } i \text{ in } PF_{known} \text{ is in } PF_{true} \end{cases}$

Table 4.6: Performance Metrics for Multi-objective Optimization Techniques (continue)

Two Set Coverage (C)	
Mathematical Formulation	**Remark**
$C(PF_{known}, PF_{true}) = \frac{\lvert\{a \in PF_{known}; \exists\, b \in PF_{true} : a \succcurlyeq b\}\rvert}{\lvert PF_{known}\rvert}$	• When C = 1, means that $PF_{known} = PF_{true}$. In other words, all elements in PF_{known} dominate or equal to all elements in PF_{true}. Contrarily, when C = 0, means that $PF_{known} \neq PF_{true}$.

4.4.5 APC-MOPAO Comparative Results

The purpose of this section is to compare the performance of the proposed APC-MOPSO with other methods using the above-mentioned benchmark test problems. This comparison is made against well-known multi-objective optimization algorithms, such as Two-local Best MOPSO (2LB-MOPSO), original MOPSO, Strength Pareto Evolutionary Algorithm (SPEA2), Non-dominated Sorting Genetic algorithm (NSGA-II), and Parallel Elite Genetic Algorithm (PEGA). Two-set Covering (C), Error Ratio (ER), Generational Distance (GD), and Spacing (S) are the four performance metrics (i.e., as explained above) used to evaluate the effectiveness of the proposed APC-MOPSO.

4.4.5.1 Results for Two-Objectives Test Problems

This section presents comparative results for the set of two-objective benchmark test problems, including ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, FONSECA, OKA2, KURSAWE, DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, and DTLZ7. For each test problem, 25 independent runs are executed for statistics collection. Furthermore, the results are displayed in four performance metrics that are Spacing (S), Two-set Coverage (C), Error Ratio (ER), and Generational Distance (GD). Samples of the experimental results obtained by the proposed APC-MOPSO, the 2LB-MOPSO, MOPSO, NSGA-II, SPEA2, and PEGA against the above test problems are discussed as follows.

- **ZDT2 Test Problem**

Figure 4.23 shows the graphical results obtained by the APC-MOPSO, the 2LB-MOPSO, the PEGA, the NSGA-II, and the SPEA2 against the ZDT2 benchmark test problem. This problem has a non-convex Pareto-optimal front, and Tables 4.7, 4.10, 4.13, and 4.16 present the numerical comparison of the Pareto-optimal fronts considering the Two-set Covering (C), Error Ratio (ER), Generational Distance (GD), and Spacing (S) metrics. Neither NSGA-II nor PEGA are converged to the optimal region and are unable to find non-dominated solutions after 100

iterations. This is also demonstrated from their high average values of C, ER, and S metrics. It can be seen that APC-MOPSO performed well, obtained non-dominated solutions spread over the entire optimal regions, and returned the best average values with respect to C, ER, and GD, while returned values that marginally below the 2LB-MOPSO with respect to S. Note however, that only APC-MOPSO found solutions that covered the entire Pareto-front of the problem.

- **ZDT4 Test Problem**

ZDT4 is a very challenging problem for any multi-objective optimization algorithm because of its modality (i.e., contains 21^9 local Pareto-optimal fronts). However, amongst the algorithms considered, the proposed APC-MOPSO is the only algorithm that converged onto the Pareto-optimal front, while 2LB-MOPSO, MOPSO, NSGA-II, PEGA, and SPEA2 did not. The solutions produced by all algorithms on this benchmark test problem are compared in terms of C, ER, GD, and S performance metrics as provided in Tables 4.7, 4.10, 4.13, and 4.16. Figure 4.24 graphically presents the solutions for the six algorithms and compares them with respect to the Pareto-optimal front. Figure 4.33 presents ZDT4's results in terms of the GD metric for all algorithms. The results demonstrated that the adaptation strategies broadly improved performance of the APC-MOPSO.

- **OKA2**

OKA2 benchmark test problem is a very difficult for any multi-objective optimization algorithm, since the algorithm converges closer to the Pareto-front, the probability density becomes sparser [260]. Figure 4.25 shows the graphical results obtained after 10,000 function evaluations by the APC-MOPSO, the 2LB-MOPSO, the PEGA, the NSGA-II, and the SPEA2 for this test problem. Tables 4.8, 4.11, 4.14, and 4.17 show the comparison of results for the six algorithms according to four commonly used performance metrics. Four algorithms that are MOPSO, NSGA-II, SPEA2, and PEGA are failed to converge and are poorly performed in terms of finding non-dominated solutions. Moreover, it can be seen that only the proposed APC-MOPSO and the 2LB-MOPSO converged to the Pareto-optimal front with 10,000 function evaluations. The results showed that the APC-MOPSO outperformed the other five algorithms. With respect to C, ER, GD, and S, APC-MOPSO returned the best average values compared to 2LB-MOPSO. Note that the results of this benchmark test problem demonstrated the effectiveness and stability of the proposed APC-MOPSO in performing the search process.

- **KURSAWE Test Problem**

The computational results produced by the proposed APC-MOPSO, the 2LB-MOPSO, the PEGA, the NSGA-II, and the SPEA2 in the KURSAWE benchmark test problem are shown in Figure 4.26. The non-dominated solutions obtained by the six algorithms are compared to the Pareto-optimal front, which has several disconnected and unsymmetrical areas in the search space (i.e., which may cause difficulty in finding non-dominated solutions in all optimal regions).

Tables 4.8, 4.11, 4.14, and 4.17 show the results for the six algorithms. From these tables, it can be noticed that the APC-MOPSO performed very well (i.e., in finding the entire Pareto-front) in terms of C and ER metrics. Furthermore, APC-MOPSO returned better average GD and S values compared to the NSGA-II and 2LB-MOPSO, respectively. With respect to SPEA2 and PEGA, they failed to find a full set of optimal solutions and to cover the entire Pareto-front of the problem.

- **DTLZ1 Test Problem**

Although DTLZ1 is considered a simple benchmark test problem (i.e., has a linear Pareto-optimal front), nonetheless it seems hard for many multi-objective optimization algorithm. Besides the high number of local Pareto-fronts this problem has (i.e., contains $(11^k - 1)$ local Pareto-fronts) [18], the total number of variables that represent the solutions in the decision space can further make it difficult for most optimization algorithms to converge to the linear Pareto-front. Figure 4.27 shows the Pareto-front sets for this problem obtained by the proposed APC-MOPSO, the 2LB-MOPSO, the MOPSO, the NSGA-II, the PEGA, and the SPEA2. Four performance metrics are adopted in this test in order to track and assess the convergence and efficiency of all algorithms. The results in Tables 4.8, 4.11, 4.14, and 4.17 showed that the APC-MOPSO and the 2LB-MOPSO algorithms are successful in solving the problem, while PEGA and MOPSO are partially successful in obtaining parts of the Pareto-fronts. SPEA2 and NSGA-II failed to converge and then to produce a set of non-dominated solutions. With respect to the considered metrics, it is clear that the proposed APC-MOPSO obtained the desired Pareto-front better than the other algorithms, especially in C, ER, and GD average values, except for the S metric whereas APC-MOPSO is slightly below the 2LB-MOPSO. Accordingly, it can be concluded that APC-MOPSO outperformed the others in solving DTLZ1problem.

- **DTLZ3 Test Problem**

Six multi-objective optimizers that are the APC-MOPSO, 2LB-MOPSO, MOPSO, NSGA-II, SPEA2, and PEGA are applied to this test with respect to the C, ER, GD, and S metrics. The DTLZ3 benchmark test problem has a concave, multimodal, and continuous Pareto-optimal front). For the this problem, only the APC-MOPSO and 2LB-MOPSO returned vectors of Pareto-front with average values of Error Ratio (ER) metric ranging from 1% to 9% and from 2% to 27%, respectively (as illustrated in Table 4.12). With regards to Spacing (S) metric, Table 4.18 shows that the APC-MOPSO consistently produced solutions that are well distributed in the range of 0.00086 to 0.0111, while the 2LB-MOPSO's results are way beyond that (i.e., ranging from 0.005 to 0.25). A similar trend can be observed when considering Generational Distance (GD) and Two-set Coverage (C) metrics (see Tables 4.12 and 4.15, respectively, for their ranges).

[18] k represents a design parameter for all DTLZ benchmark problems, and it is calculated by using the following formula: $k = number\ of\ variables - number\ of\ objectives + 1$

NSGA-II, SPEA2, PEGA, and MOPSO algorithms returned vectors farther away from the Pareto-optimal front, and they failed to converge to the global regions in all 25 independent runs. Figure 4.28 illustrates the results of the six algorithms for the DTLZ3, and also compares the solutions with respect to the Pareto-optimal front.

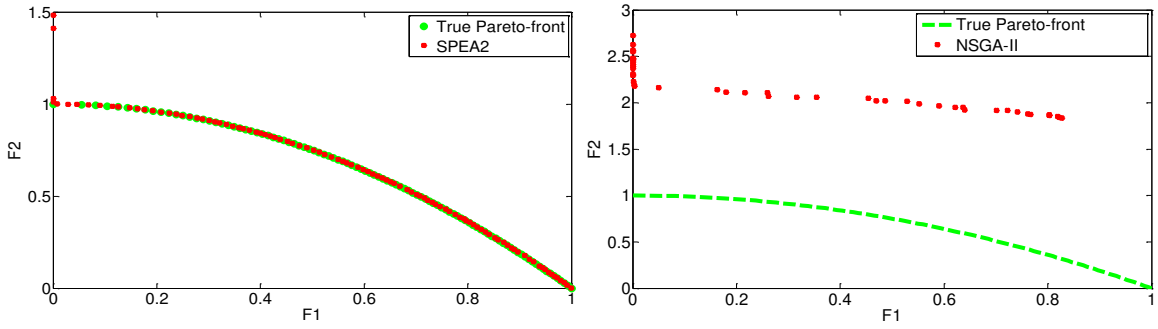
- **DTLZ6 Test Problem**

Another hard problem is chosen to measure the quality of the overall performance of the APC-MOPSO and five other well-known algorithms that are 2LB-MOPSO, MOPSO, NSGA-II, SPEA2, and PEGA. The DTLZ6 benchmark test problem has a Pareto-optimal front that is unimodal, continuous, and non-convex. Figure 4.29 shows graphically the set of non-dominated solutions obtained by each of the above algorithms compared to the Pareto-optimal front of the problem. It can be seen that the APC-MOPSO evidently converged faster to the global front and has a wider converge length than others have. This can easily be confirmed by looking at the average values of C, ER, GD, and S for the APC-MOPSO compared to the other algorithms (see Tables 4.9, 4.12, 4.15, and 4.18).

- **DTLZ7 Test Problem**

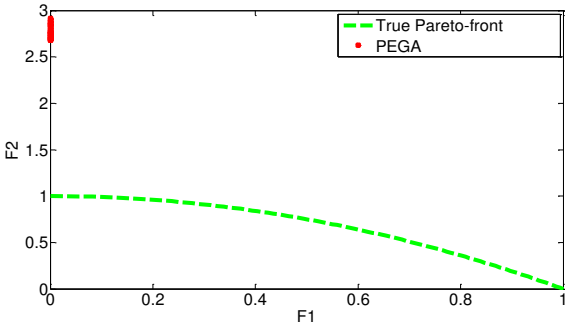
The last benchmark test problem in the two-objectives set is DTLZ7, which has disconnected Pareto-optimal regions. The results produced by the APC-MOPSO, 2LB-MOPSO, MOPSO, NSGA-II, SPEA2, and PEGA with respect to C, ER, GD, and S metrics are graphically illustrated in Figure 4.30, while their statistical values (i.e., best, worst, average, median, and standard deviations for each metric) are tabulated in Tables 4.9, 4.12, 4.15, and 4.18. From this figure, it is clear that the APC-MOPSO, the 2LB-MOPSO, and the SPEA2 produced solutions that covered the entire disjointed front, while PEGA and MOPSO did converge to a segment of that front. With regards to NSGA-II, it returned the worst non-dominated solutions. Furthermore, by looking at the four tables, the average values of the four metrics showed that the APC-MOPSO outperformed all other algorithms. An example of that is the average values of ER returned by the APC-MOPSO that ranged from 1% to 6%, while for 2LB-MOPSO and SPEA2, the ranged from 2% to 7% and from 5% to 14%, respectively.

Finally, it is important to provide examples of the four performance metrics and their average values that are returned by the six algorithms. Figures 4.31 to 4.34 illustrate the average values of the four metrics against ZDT1, ZDT3, ZDT4, and ZDT6 benchmark test problems. From these figures, it is clearly shown that the proposed APC-MOPSO returned the better (i.e., smallest) average values for all metrics compared with other algorithms.

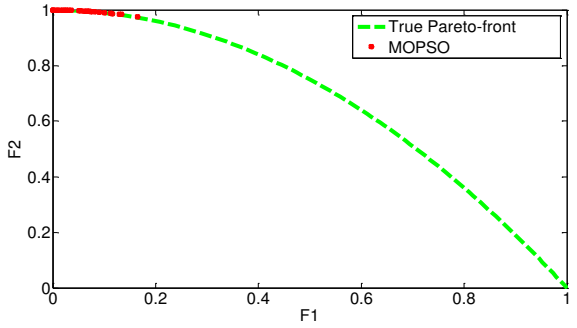


a. Pareto-front Produced by SPEA2

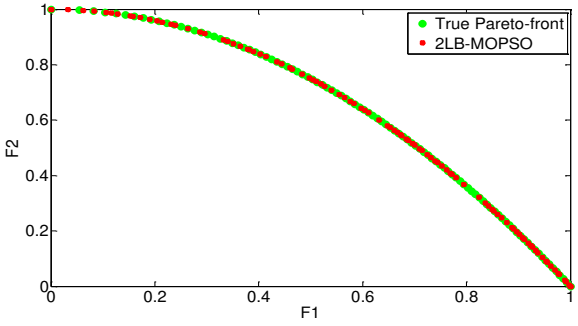
b. Pareto-front Produced by NSGA-II



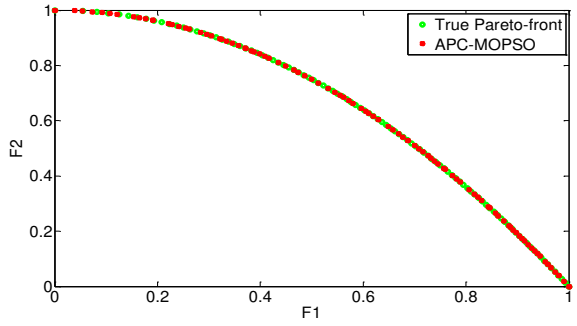
c. Pareto-front Produced by PEGA



d. Pareto-front Produced by MOPSO

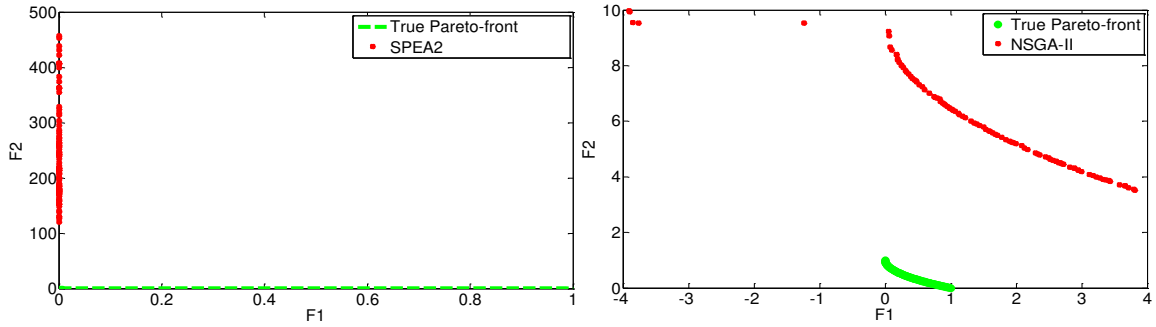


e. Pareto-front Produced by 2LB-MOPSO



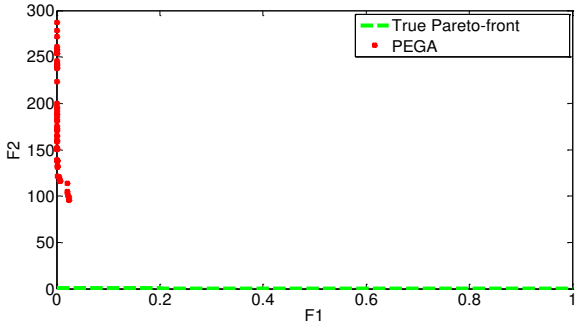
f. Pareto-front Produced by APC-MOPSO

Figure 4.23: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the ZDT2 Benchmark Test Problem (2-D)

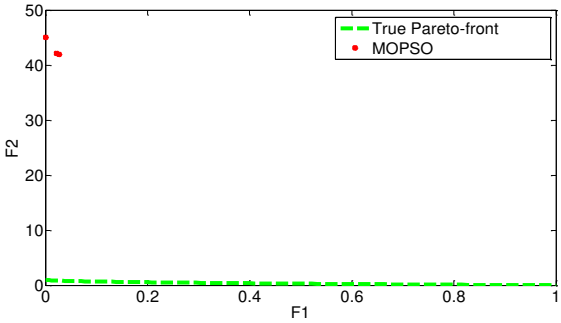


a. Pareto-front Produced by SPEA2

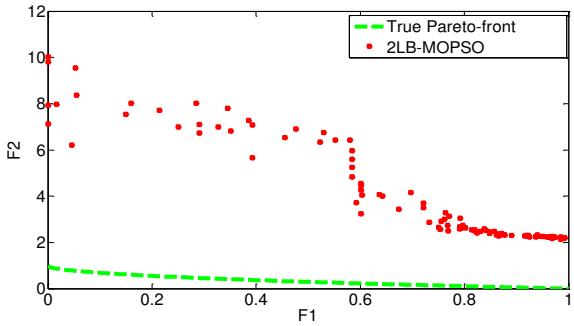
b. Pareto-front Produced by NSGA-II



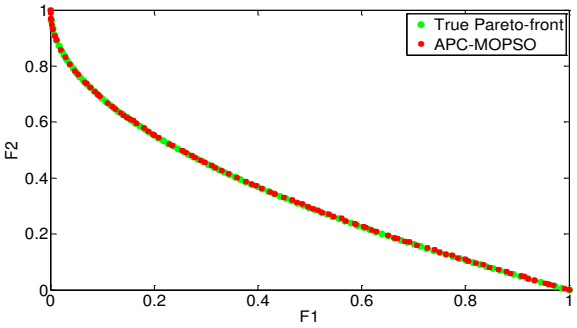
c. Pareto-front Produced by PEGA



d. Pareto-front Produced by MOPSO

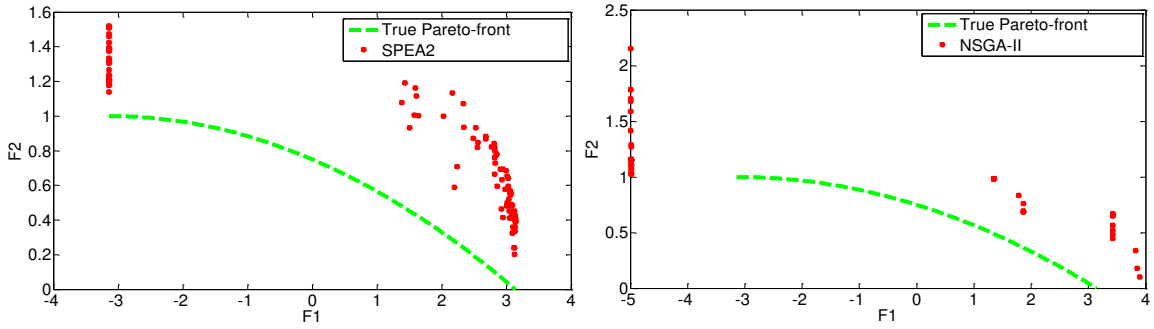


e. Pareto-front Produced by 2LB-MOPSO



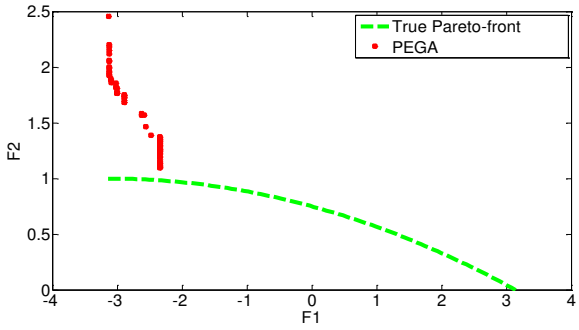
f. Pareto-front Produced by APC-MOPSO

Figure 4.24: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the ZDT4 Benchmark Test Problem (2-D)

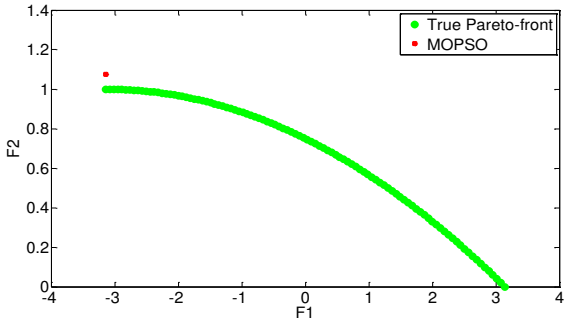


a. Pareto-front Produced by SPEA2

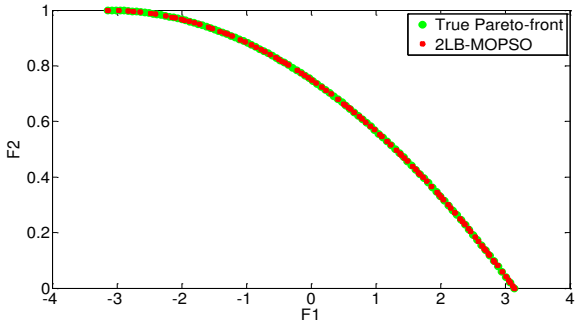
b. Pareto-front Produced by NSGA-II



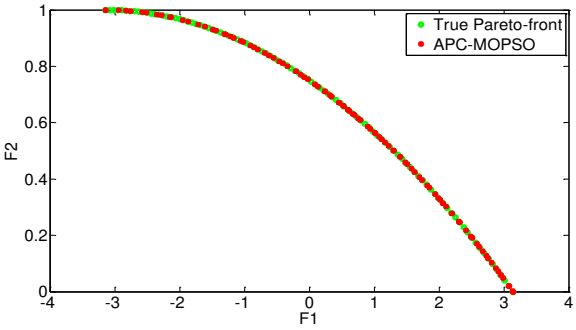
c. Pareto-front Produced by PEGA



d. Pareto-front Produced by MOPSO

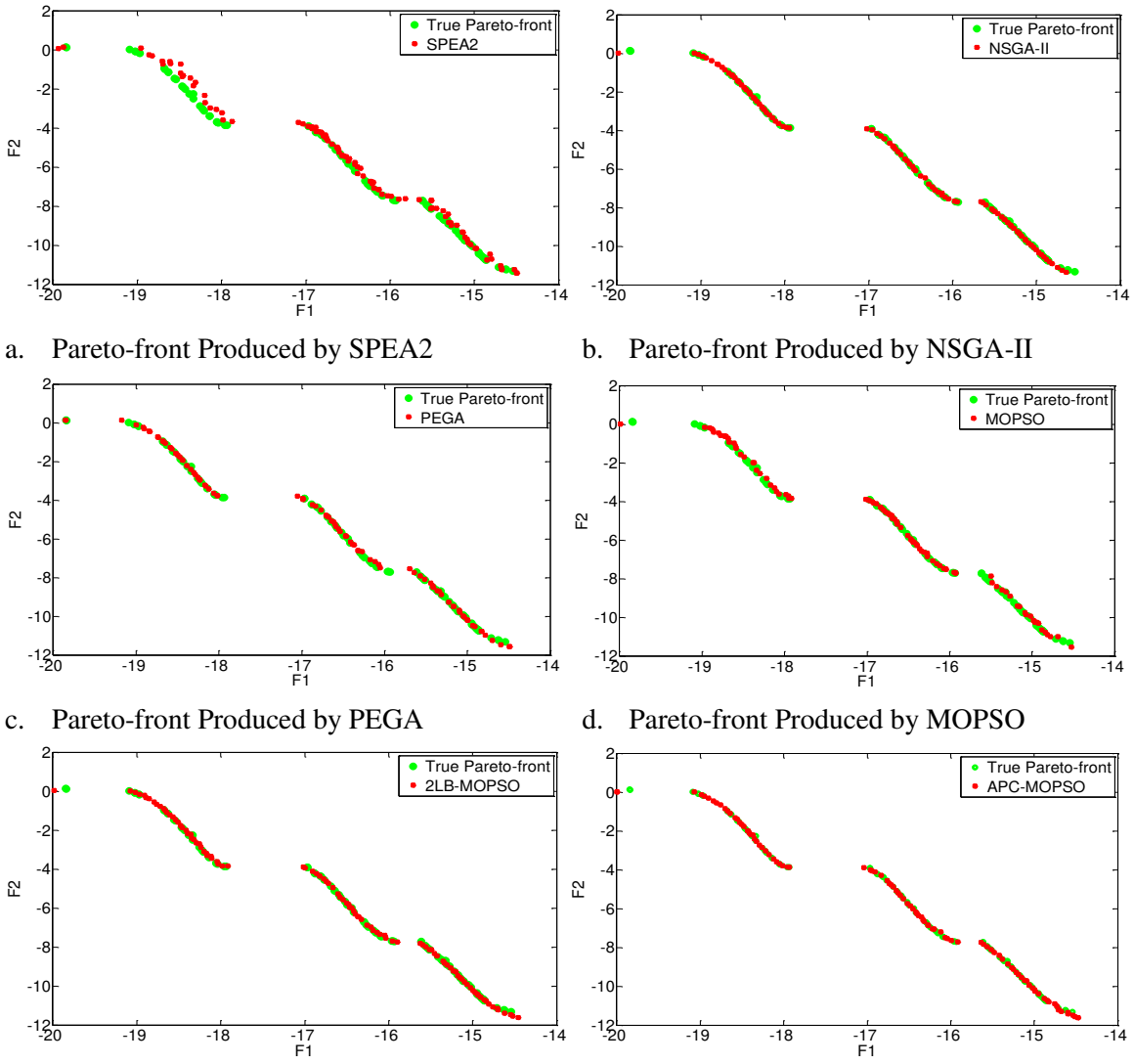


e. Pareto-front Produced by 2LB-MOPSO



f. Pareto-front Produced by APC-MOPSO

Figure 4.25: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the OKA2 Benchmark Test Problem (2-D)



a. Pareto-front Produced by SPEA2

b. Pareto-front Produced by NSGA-II

c. Pareto-front Produced by PEGA

d. Pareto-front Produced by MOPSO

e. Pareto-front Produced by 2LB-MOPSO

f. Pareto-front Produced by APC-MOPSO

Figure 4.26: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the KURSAWE Benchmark Test Problem (2-D)

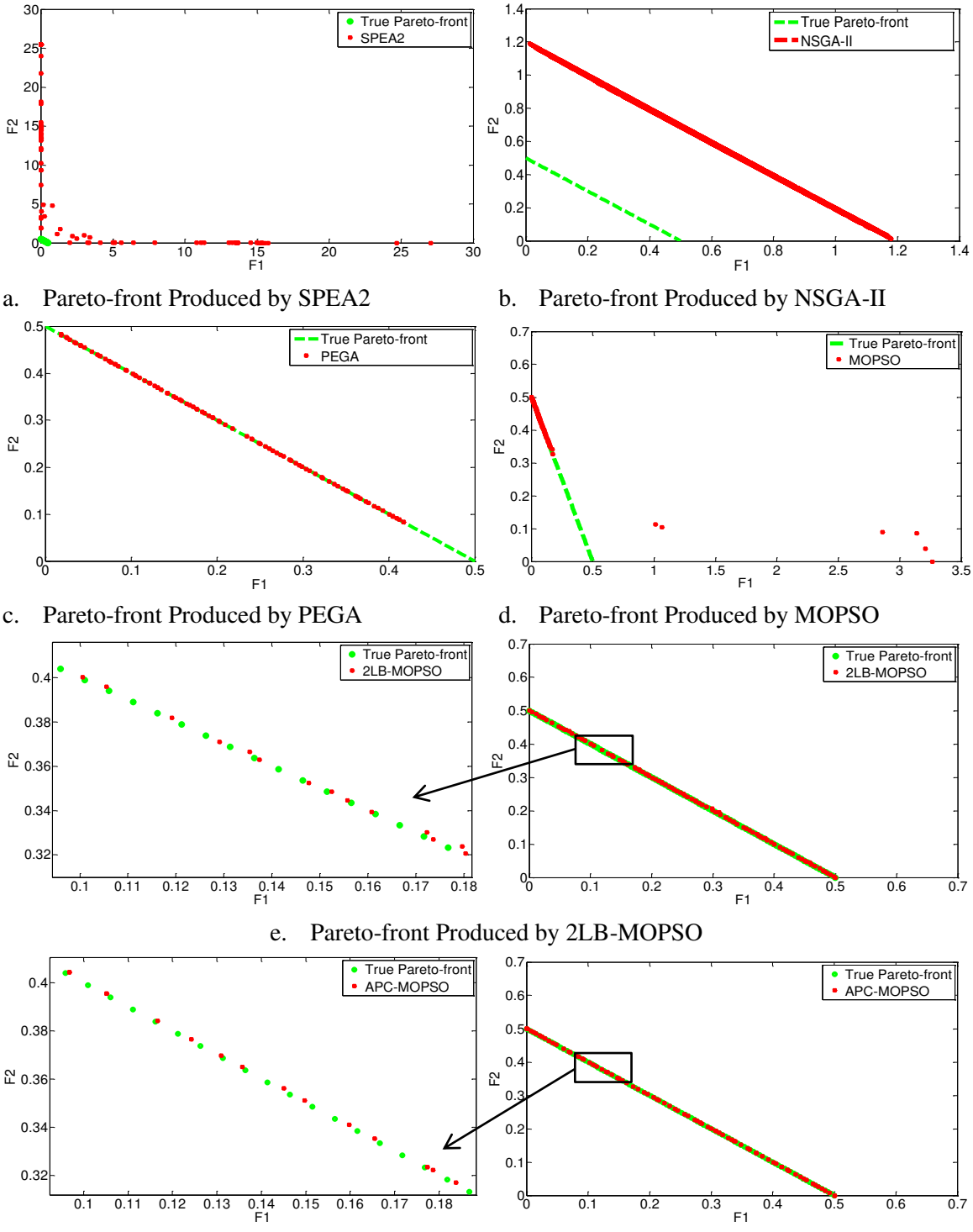
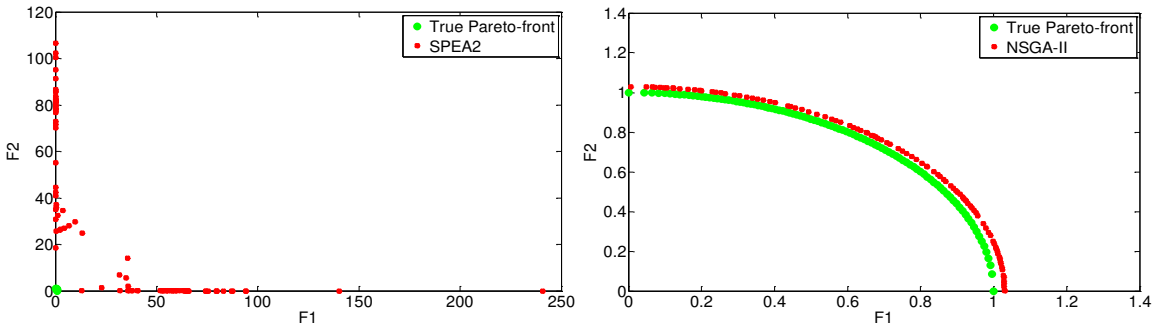
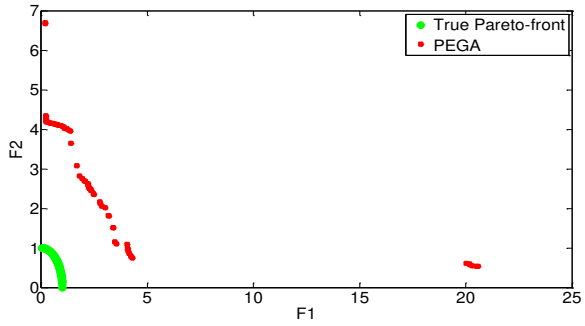


Figure 4.27: Pareto-front Produced by the MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ1 Test Problem (2-D)

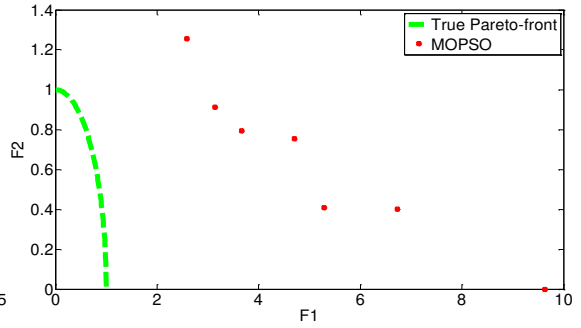


a. Pareto-front Produced by SPEA2

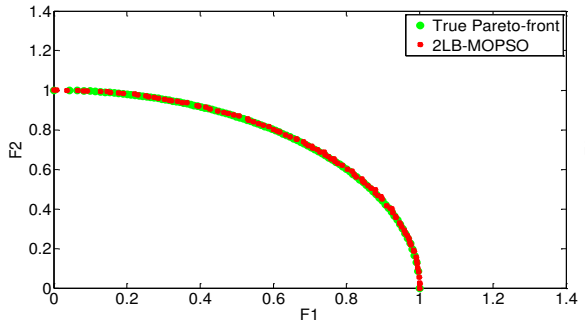
b. Pareto-front Produced by NSGA-II



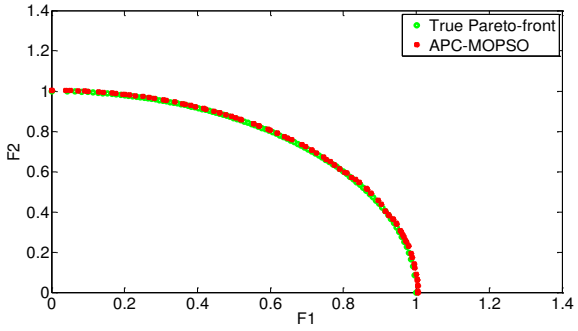
c. Pareto-front Produced by PEGA



d. Pareto-front Produced by MOPSO

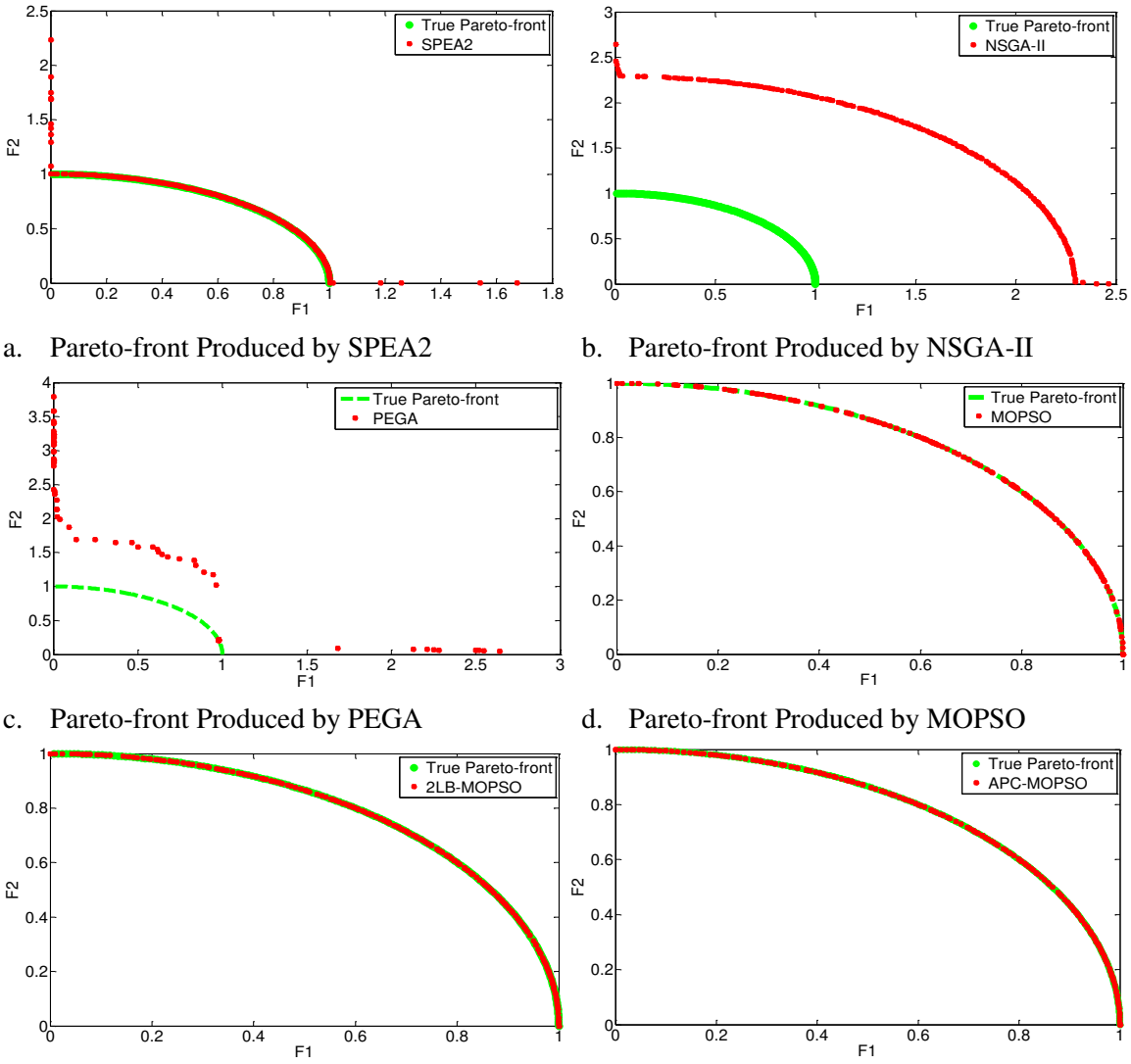


e. Pareto-front Produced by 2LB-MOPSO



f. Pareto-front Produced by APC-MOPSO

Figure 4.28: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ3 Benchmark Test Problem (2-D)



a. Pareto-front Produced by SPEA2

b. Pareto-front Produced by NSGA-II

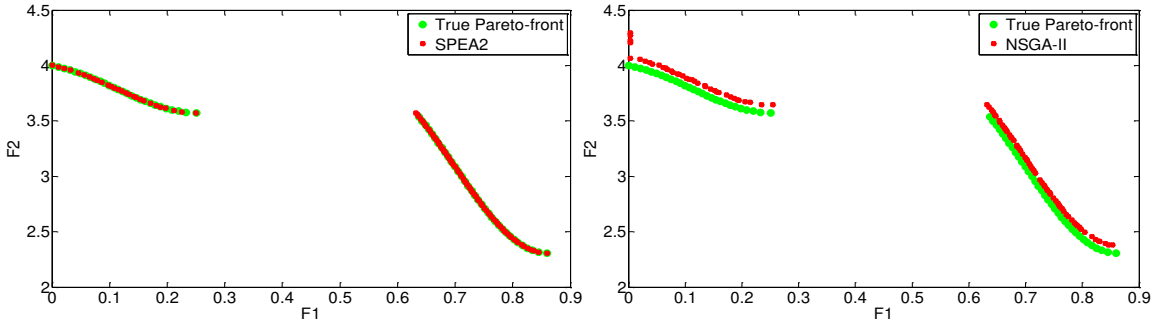
c. Pareto-front Produced by PEGA

d. Pareto-front Produced by MOPSO

e. Pareto-front Produced by 2LB-MOPSO

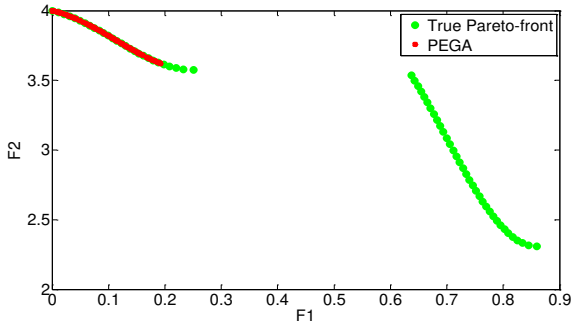
f. Pareto-front Produced by APC-MOPSO

Figure 4.29: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ6 Benchmark Test Problem (2-D)

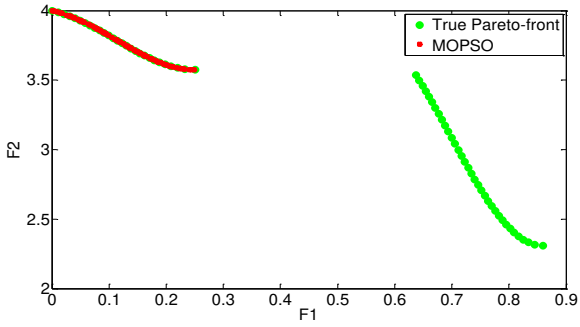


a. Pareto-front Produced by SPEA2

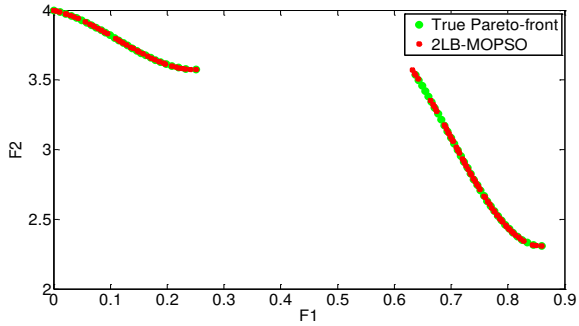
b. Pareto-front Produced by NSGA-II



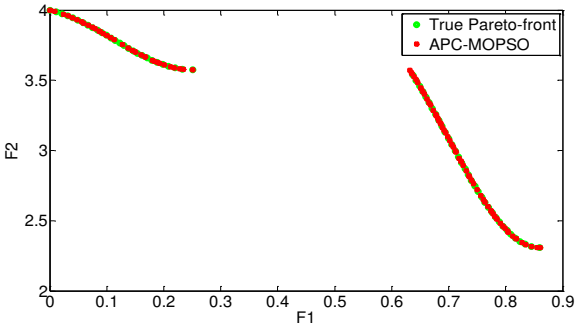
c. Pareto-front Produced by PEGA



d. Pareto-front Produced by MOPSO



e. Pareto-front Produced by 2LB-MOPSO



f. Pareto-front Produced by APC-MOPSO

Figure 4.30: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ7 Benchmark Test Problem (2-D)

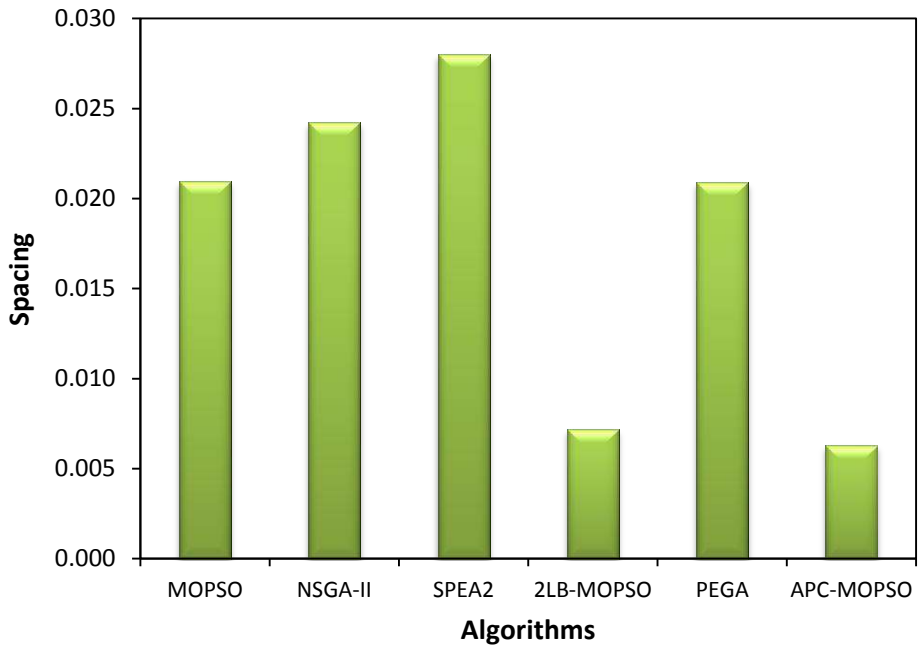


Figure 4.31: Overall Results for ZDT1 Benchmark Test Problem Using Spacing Metric

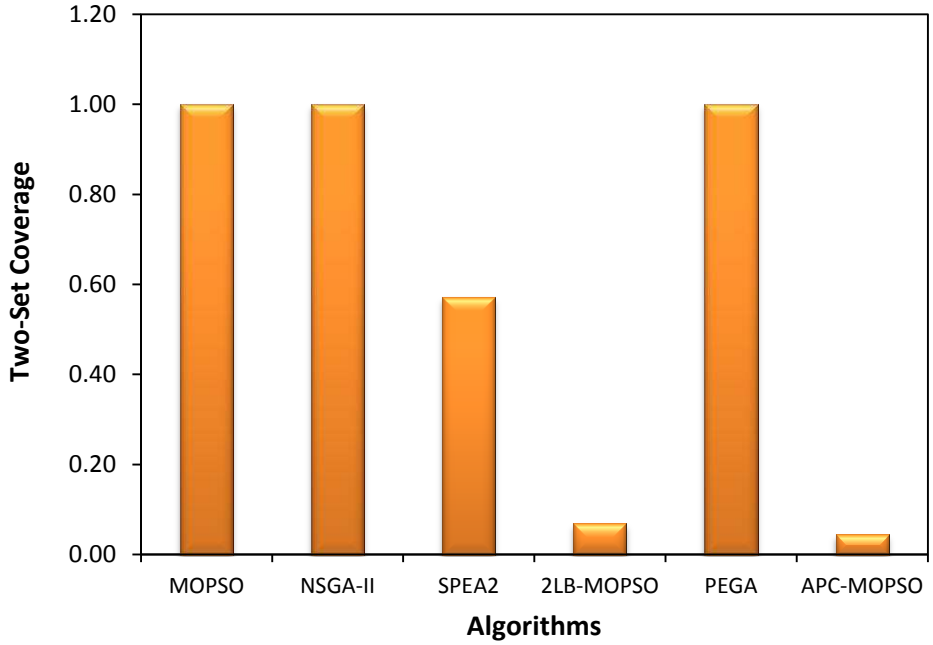


Figure 4.32: Overall Results for ZDT3 Benchmark Test Problem Using Coverage Metric

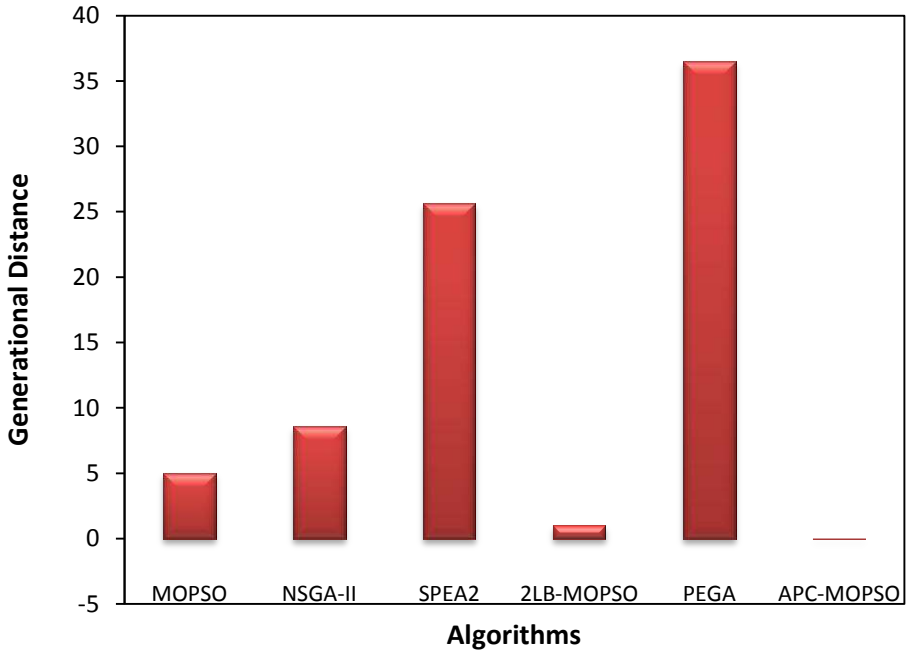


Figure 4.33: Overall Results for ZDT4 Benchmark Test Problem Generational Distance Metric

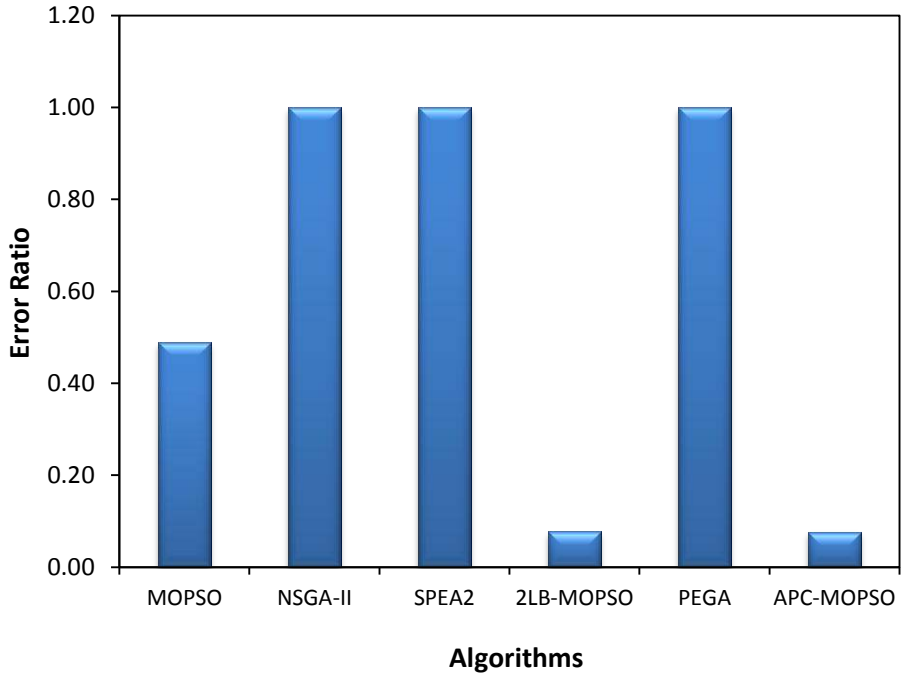


Figure 4.34: Overall Results for ZDT6 Benchmark Test Problem Using Error Ration Metric

Table 4.7: The Results for Two-set Coverage Metric on Benchmark Test Problems 1-5 (2-D)

Method	Statistics	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
APC-MOPSO	Best	0.01000	0.04000	0.02000	0.12000	0.02000
	Worst	0.04000	0.04000	0.07000	0.22000	0.02000
	Mean	0.02600	0.04000	0.04400	0.17400	0.02000
	Median	0.02000	0.04000	0.04000	0.20000	0.02000
	Std	0.01242	0.00000	0.01502	0.04561	0.00000
2LB-MOPSO	Best	0.02000	0.04000	0.02000	0.28000	0.02000
	Worst	0.08000	0.08000	0.12000	0.96000	0.02000
	Mean	0.04520	0.04800	0.06880	0.83680	0.02000
	Median	0.04000	0.04000	0.07000	0.89000	0.02000
	Std	0.01782	0.01190	0.02403	0.15842	0.00000
NSGAII	Best	1.00000	1.00000	1.00000	1.00000	0.12000
	Worst	1.00000	1.00000	1.00000	1.00000	0.60000
	Mean	1.00000	1.00000	1.00000	1.00000	0.29810
	Median	1.00000	1.00000	1.00000	1.00000	0.25930
	Std	0.00000	0.00000	0.00000	0.00000	0.12591
SPEA2	Best	0.09000	0.21000	0.43000	0.02000	0.74000
	Worst	0.28000	1.00000	0.76000	0.38000	0.85000
	Mean	0.16760	0.57160	0.57120	0.20960	0.78700
	Median	0.16000	0.34000	0.57000	0.20000	0.78000
	Std	0.04763	0.36047	0.08343	0.10106	0.04596
MOPSO	Best	0.96880	0.07000	1.00000	0.75000	0.00000
	Worst	0.98920	1.00000	1.00000	0.98940	0.68570
	Mean	0.98218	0.86797	1.00000	0.91917	0.11140
	Median	0.98410	1.00000	1.00000	0.94740	0.02000
	Std	0.00565	0.30079	0.00000	0.06812	0.18346
PEGA	Best	0.92210	1.00000	1.00000	0.45280	0.49120
	Worst	1.00000	1.00000	1.00000	1.00000	0.78080
	Mean	0.97584	1.00000	1.00000	0.90778	0.68047
	Median	0.98210	1.00000	1.00000	1.00000	0.68250
	Std	0.02485	0.00000	0.00000	0.19861	0.07616

Table 4.8: The Results for Two-set Coverage Metric on Benchmark Test Problems 6-10 (2-D)

Method	Statistics	FONSECA	KURSAWE	OKA2	DTLZ1	DTLZ2
APC-MOPSO	Best	0.14830	0.08000	0.04000	0.08000	0.09000
	Worst	0.18510	0.20000	0.04000	0.28000	0.33000
	Mean	0.17503	0.13400	0.04000	0.19000	0.13400
	Median	0.17895	0.13000	0.04000	0.17000	0.11000
	Std	0.01214	0.03562	0.00000	0.08206	0.07074
2LB-MOPSO	Best	0.25290	0.08000	0.04000	0.19000	0.24000
	Worst	0.31720	0.25000	0.04000	0.39000	0.47000
	Mean	0.28535	0.16680	0.04000	0.28600	0.33200
	Median	0.28510	0.16000	0.04000	0.28000	0.32000
	Std	0.01815	0.04375	0.00000	0.05317	0.06215
NSGAII	Best	0.18390	0.07000	0.33330	1.00000	0.05000
	Worst	0.26210	0.34000	0.70000	1.00000	0.09000
	Mean	0.21580	0.13400	0.54573	1.00000	0.07333
	Median	0.20920	0.13000	0.58245	1.00000	0.08000
	Std	0.02200	0.05635	0.14724	0.00000	0.02082
SPEA2	Best	0.71720	0.57000	1.00000	1.00000	0.31000
	Worst	0.81380	0.84000	1.00000	1.00000	0.51000
	Mean	0.76450	0.68920	1.00000	1.00000	0.41800
	Median	0.76550	0.69000	1.00000	1.00000	0.43000
	Std	0.02694	0.05802	0.00000	0.00000	0.06303
MOPSO	Best	0.13560	0.29000	1.00000	1.00000	0.10000
	Worst	0.20920	0.52000	1.00000	1.00000	0.40000
	Mean	0.17501	0.41051	1.00000	1.00000	0.24700
	Median	0.18390	0.42000	1.00000	1.00000	0.26000
	Std	0.02515	0.06312	0.00000	0.00000	0.09117
PEGA	Best	0.27060	0.15560	1.00000	0.04120	0.13830
	Worst	0.37170	0.51160	1.00000	1.00000	0.81010
	Mean	0.33182	0.30559	1.00000	0.76383	0.52001
	Median	0.33420	0.31460	1.00000	1.00000	0.57040
	Std	0.02859	0.12415	0.00000	0.37554	0.22640

Table 4.9: The Results for Two-set Coverage Metric on Benchmark Test Problems 11-15 (2-D)

Method	Statistics	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
APC-MOPSO	Best	0.14000	0.04000	0.07500	0.01000	0.02000
	Worst	0.86000	0.09000	0.15000	0.01000	0.02000
	Mean	0.55333	0.06700	0.10350	0.01000	0.02000
	Median	0.55000	0.07000	0.10500	0.01000	0.02000
	Std	0.21633	0.01829	0.02199	0.00000	0.00000
2LB-MOPSO	Best	0.34000	0.22000	0.27000	0.01000	0.02000
	Worst	1.00000	0.43000	0.38000	0.01000	0.03000
	Mean	0.57900	0.33800	0.32150	0.01000	0.02100
	Median	0.45000	0.35500	0.31000	0.01000	0.02000
	Std	0.27562	0.06795	0.04184	0.00000	0.00316
NSGAII	Best	1.00000	0.02000	0.00500	0.94000	1.00000
	Worst	1.00000	0.11000	0.07500	0.99500	1.00000
	Mean	1.00000	0.07000	0.02800	0.97506	1.00000
	Median	1.00000	0.07500	0.02250	0.97750	1.00000
	Std	0.00000	0.03916	0.01889	0.01799	0.00000
SPEA2	Best	1.00000	0.33000	0.40500	0.87000	0.02000
	Worst	1.00000	1.00000	0.51000	0.96000	0.11000
	Mean	1.00000	0.45300	0.44650	0.92550	0.05133
	Median	1.00000	0.38000	0.44250	0.93750	0.05000
	Std	0.00000	0.19528	0.03598	0.02733	0.02416
MOPSO	Best	1.00000	0.06000	0.11000	0.01000	0.03000
	Worst	1.00000	1.00000	0.21500	0.43140	0.20000
	Mean	1.00000	0.69889	0.14700	0.05714	0.10300
	Median	1.00000	1.00000	0.14500	0.01500	0.09500
	Std	0.00000	0.45195	0.03199	0.13160	0.04715
PEGA	Best	1.00000	0.24420	0.11890	0.27680	0.07140
	Worst	1.00000	1.00000	0.30820	0.99490	1.00000
	Mean	1.00000	0.64985	0.23250	0.88727	0.88311
	Median	1.00000	0.55280	0.23235	0.96080	1.00000
	Std	0.00000	0.26215	0.05414	0.21875	0.30909

Table 4.10: The Results for Error Ratio Metric on Benchmark Test Problems 1-5 (2-D)

Method	Statistics	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
APC-MOPSO	Best	0.00000	0.01000	0.07000	0.00000	0.04000
	Worst	0.04000	0.05000	0.14000	0.02000	0.12000
	Mean	0.01667	0.02600	0.10267	0.01200	0.07667
	Median	0.02000	0.03000	0.10000	0.01000	0.08000
	Std	0.01047	0.01056	0.02017	0.00837	0.02717
2LB-MOPSO	Best	0.00000	0.01000	0.07000	1.00000	0.04000
	Worst	0.03000	0.05000	0.16000	1.00000	0.12000
	Mean	0.01720	0.02760	0.11200	1.00000	0.07720
	Median	0.02000	0.03000	0.11000	1.00000	0.08000
	Std	0.00843	0.00926	0.02432	0.00000	0.02031
NSGAII	Best	1.00000	1.00000	1.00000	1.00000	1.00000
	Worst	1.00000	1.00000	1.00000	1.00000	1.00000
	Mean	1.00000	1.00000	1.00000	1.00000	1.00000
	Median	1.00000	1.00000	1.00000	1.00000	1.00000
	Std	0.00000	0.00000	0.00000	0.00000	0.00000
SPEA2	Best	0.04000	0.06000	0.11000	1.00000	1.00000
	Worst	0.17000	1.00000	0.30000	1.00000	1.00000
	Mean	0.08760	0.46800	0.18960	1.00000	1.00000
	Median	0.08000	0.14000	0.18000	1.00000	1.00000
	Std	0.03689	0.44392	0.04860	0.00000	0.00000
MOPSO	Best	1.00000	0.84620	0.42310	1.00000	0.23000
	Worst	1.00000	1.00000	1.00000	1.00000	1.00000
	Mean	1.00000	0.98188	0.97457	1.00000	0.48833
	Median	1.00000	1.00000	1.00000	1.00000	0.40000
	Std	0.00000	0.04513	0.11549	0.00000	0.27407
PEGA	Best	1.00000	1.00000	1.00000	1.00000	1.00000
	Worst	1.00000	1.00000	1.00000	1.00000	1.00000
	Mean	1.00000	1.00000	1.00000	1.00000	1.00000
	Median	1.00000	1.00000	1.00000	1.00000	1.00000
	Std	0.00000	0.00000	0.00000	0.00000	0.00000

Table 4.11: The Results for Error Ratio Metric on Benchmark Test Problems 6-10 (2-D)

Method	Statistics	FONSECA	KURSAWE	OKA2	DTLZ1	DTLZ2
APC-MOPSO	Best	0.06670	0.09000	0.04000	0.01000	0.04000
	Worst	0.10110	0.15000	0.11000	0.17000	0.09000
	Mean	0.07963	0.12400	0.06333	0.08714	0.06600
	Median	0.07590	0.12000	0.06000	0.07000	0.07000
	Std	0.01170	0.01724	0.01799	0.05345	0.01430
2LB-MOPSO	Best	0.08970	0.12000	0.02000	0.04000	0.05000
	Worst	0.14940	0.23000	0.11000	0.15000	0.08000
	Mean	0.12033	0.16080	0.06467	0.10000	0.06900
	Median	0.11950	0.16000	0.07000	0.10000	0.07000
	Std	0.01543	0.02644	0.02416	0.03091	0.00994
NSGAII	Best	0.07820	0.09000	1.00000	1.00000	0.05000
	Worst	0.14250	0.19000	1.00000	1.00000	0.09000
	Mean	0.10773	0.12800	1.00000	1.00000	0.07333
	Median	0.10800	0.12000	1.00000	1.00000	0.08000
	Std	0.01733	0.02533	0.00000	0.00000	0.02082
SPEA2	Best	0.46210	0.22000	1.00000	1.00000	0.04000
	Worst	0.59540	0.46000	1.00000	1.00000	0.11000
	Mean	0.52542	0.31040	1.00000	1.00000	0.08300
	Median	0.52640	0.31000	1.00000	1.00000	0.08500
	Std	0.03627	0.05863	0.00000	0.00000	0.02058
MOPSO	Best	0.06210	0.10000	1.00000	1.00000	0.01000
	Worst	0.10800	0.35000	1.00000	1.00000	0.05000
	Mean	0.08845	0.18225	1.00000	1.00000	0.02700
	Median	0.08740	0.17000	1.00000	1.00000	0.03000
	Std	0.01241	0.05073	0.00000	0.00000	0.01252
PEGA	Best	0.13260	0.06820	1.00000	0.00000	0.07610
	Worst	0.19050	0.31870	1.00000	1.00000	0.41770
	Mean	0.15552	0.19145	1.00000	0.74306	0.25499
	Median	0.15260	0.17980	1.00000	1.00000	0.25465
	Std	0.01795	0.06001	0.00000	0.39471	0.12360

Table 4.12: The Results for Error Ratio Metric on Benchmark Test Problems 11-15 (2-D)

Method	Statistics	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
APC-MOPSO	Best	0.01000	0.03000	0.00000	0.01000	0.01000
	Worst	0.09000	0.09000	0.01000	0.01000	0.06000
	Mean	0.05000	0.04500	0.00700	0.01000	0.03417
	Median	0.04000	0.04000	0.01000	0.01000	0.03000
	Std	0.02398	0.01780	0.00483	0.00000	0.02109
2LB-MOPSO	Best	0.02000	0.03000	0.00500	0.01000	0.02000
	Worst	1.00000	0.06000	0.00500	0.01500	0.07000
	Mean	0.27600	0.04700	0.00500	0.01050	0.04500
	Median	0.05000	0.05000	0.00500	0.01000	0.04500
	Std	0.38831	0.01059	0.00000	0.00158	0.01581
NSGAII	Best	1.00000	0.03000	0.00500	1.00000	0.62000
	Worst	1.00000	0.07000	0.01000	1.00000	0.87000
	Mean	1.00000	0.05500	0.00700	1.00000	0.72000
	Median	1.00000	0.06000	0.00500	1.00000	0.67000
	Std	0.00000	0.01732	0.00258	0.00000	0.13229
SPEA2	Best	1.00000	0.03000	0.00500	0.28500	0.05000
	Worst	1.00000	1.00000	0.03000	1.00000	0.14000
	Mean	1.00000	0.15000	0.01400	0.87150	0.07667
	Median	1.00000	0.06000	0.01250	0.99000	0.07000
	Std	0.00000	0.29900	0.00699	0.24686	0.02320
MOPSO	Best	1.00000	0.06000	0.00500	0.00500	0.00000
	Worst	1.00000	1.00000	0.04000	0.43140	0.41000
	Mean	1.00000	0.69111	0.02450	0.05764	0.06900
	Median	1.00000	1.00000	0.02500	0.01750	0.00500
	Std	0.00000	0.46337	0.01039	0.13151	0.14091
PEGA	Best	1.00000	0.00000	0.00520	0.55360	0.00000
	Worst	1.00000	0.23600	0.23900	1.00000	1.00000
	Mean	1.00000	0.08667	0.11249	0.95536	0.86196
	Median	1.00000	0.07895	0.10925	1.00000	1.00000
	Std	0.00000	0.07039	0.07633	0.14116	0.33930

Table 4.13: The Results for Generational Distance Metric on Test Problems 1-5 (2-D)

Method	Statistics	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
APC-MOPSO	Best	0.000311	0.000427	0.000610	0.000418	0.00031
	Worst	0.000505	0.000531	0.000731	0.000506	0.00037
	Mean	0.000447	0.000492	0.000664	0.000467	0.00034
	Median	0.000459	0.000490	0.000660	0.000469	0.00034
	Std	0.000053	0.000034	0.000039	0.000032	0.00002
2LB-MOPSO	Best	0.000389	0.000424	0.000656	0.27470	0.000310
	Worst	0.000492	0.000550	0.000776	1.83460	0.000377
	Mean	0.000454	0.000500	0.000715	1.03966	0.000343
	Median	0.000452	0.000502	0.000714	0.97330	0.000340
	Std	0.000026	0.000030	0.000035	0.42173	0.000018
NSGAII	Best	0.05270	0.18190	0.06510	0.56080	0.56600
	Worst	0.12840	0.29830	0.11220	44.78510	0.87590
	Mean	0.08696	0.23014	0.09424	8.58564	0.71754
	Median	0.08400	0.22710	0.09380	4.90910	0.71590
	Std	0.01758	0.03117	0.01173	9.68899	0.09186
SPEA2	Best	0.00220	0.00480	0.00270	20.94510	0.02140
	Worst	0.05140	1.00000	0.04840	35.56060	0.18290
	Mean	0.01306	0.40664	0.01748	25.61957	0.10725
	Median	0.00770	0.01470	0.01320	24.73090	0.11185
	Std	0.01290	0.49448	0.01298	3.85661	0.05240
MOPSO	Best	0.07280	0.00110	0.02940	0.72040	0.02850
	Worst	0.16290	1.00000	0.22720	21.29110	1.21310
	Mean	0.11388	0.70236	0.14167	5.01252	0.37875
	Median	0.11490	1.00000	0.13960	3.40710	0.22530
	Std	0.02527	0.46038	0.04799	4.59958	0.38673
PEGA	Best	0.11500	0.24380	0.13810	7.81260	0.55790
	Worst	0.27950	0.51100	0.28210	133.41250	0.76580
	Mean	0.20087	0.37037	0.19969	36.50758	0.67525
	Median	0.19520	0.37020	0.19280	17.94635	0.69330
	Std	0.04582	0.07124	0.04462	42.06396	0.07244

Table 4.14: The Results for Generational Distance Metric on Test Problems 6-10 (2-D)

Method	Statistics	FONSECA	KURSAWE	OKA2	DTLZ1	DTLZ2
APC-MOPSO	Best	0.000061	0.00930	0.00170	0.000207	0.000803
	Worst	0.000065	0.01170	0.00190	0.000282	0.000945
	Mean	0.000063	0.01068	0.00183	0.000227	0.000853
	Median	0.000063	0.01080	0.00180	0.000216	0.000838
	Std	0.000001	0.00056	0.00006	0.000026	0.000043
2LB-MOPSO	Best	0.000062	0.00970	0.00170	0.000211	0.000800
	Worst	0.000067	0.01180	0.00190	0.000251	0.000959
	Mean	0.000064	0.01078	0.00181	0.000233	0.000891
	Median	0.000064	0.01080	0.00180	0.000235	0.000912
	Std	0.000001	0.00050	0.00008	0.000014	0.000056
NSGAII	Best	0.000058	0.00910	0.11260	0.05460	0.000817
	Worst	0.000065	0.01080	0.26130	0.27490	0.000922
	Mean	0.000062	0.00998	0.18658	0.12858	0.000886
	Median	0.000063	0.01000	0.18920	0.11015	0.000918
	Std	0.000002	0.00045	0.05816	0.07484	0.000059
SPEA2	Best	0.000115	0.01190	0.04890	1.67060	0.00080
	Worst	0.000143	0.01900	0.05820	4.65500	0.00390
	Mean	0.000126	0.01422	0.05322	3.03853	0.00144
	Median	0.000124	0.01380	0.05365	2.89600	0.00105
	Std	0.000007	0.00164	0.00281	1.09419	0.00093
MOPSO	Best	0.0000550	0.00920	1.00000	0.48620	0.00052
	Worst	0.0000591	0.01390	1.00000	6.09610	0.00078
	Mean	0.0000577	0.01154	1.00000	1.82460	0.00066
	Median	0.0000581	0.01150	1.00000	1.36680	0.00067
	Std	0.0000012	0.00114	0.00000	1.64930	0.00008
PEGA	Best	0.000070	0.00890	0.08160	0.00020	0.00100
	Worst	0.000080	0.06520	0.18450	4.04260	0.02480
	Mean	0.000074	0.01908	0.13577	0.92276	0.01243
	Median	0.000074	0.01230	0.13410	0.28360	0.01470
	Std	0.000003	0.01792	0.03219	1.15026	0.00959

Table 4.15: The Results for Generational Distance Metric on Test Problems 11-15 (2-D)

Method	Statistics	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
APC-MOPSO	Best	0.00060	0.00060	0.000160	0.000160	0.00090
	Worst	0.00340	0.00110	0.000170	0.000169	0.00100
	Mean	0.00108	0.00078	0.000164	0.000165	0.00099
	Median	0.00080	0.00080	0.000164	0.000165	0.00100
	Std	0.00088	0.00013	0.000003	0.000003	0.00003
2LB-MOPSO	Best	0.00070	0.000656	0.00280	0.000159	0.00090
	Worst	0.21080	0.000869	0.00320	0.000175	0.00100
	Mean	0.02255	0.000785	0.00295	0.000165	0.00099
	Median	0.00080	0.000809	0.00290	0.000163	0.00100
	Std	0.06617	0.000064	0.00014	0.000005	0.00003
NSGAII	Best	0.00310	0.000716	0.00290	0.09190	0.00660
	Worst	0.09180	0.000916	0.00370	0.16240	0.02110
	Mean	0.03283	0.000816	0.00328	0.10861	0.01337
	Median	0.00360	0.000816	0.00325	0.10160	0.01240
	Std	0.05107	0.000082	0.00025	0.02040	0.00730
SPEA2	Best	6.07670	0.00070	0.00190	0.00550	0.00100
	Worst	12.19850	1.00000	0.00480	0.01860	0.05060
	Mean	8.73841	0.10123	0.00271	0.01296	0.00736
	Median	8.53120	0.00095	0.00215	0.01355	0.00130
	Std	2.04655	0.31580	0.00110	0.00468	0.01392
MOPSO	Best	1.78570	0.00089	0.00450	0.00020	0.00050
	Worst	34.41650	1.00000	0.00580	0.07960	0.00110
	Mean	11.09382	0.66697	0.00518	0.00814	0.00065
	Median	5.74310	1.00000	0.00525	0.00020	0.00060
	Std	11.72776	0.49954	0.00041	0.02511	0.00020
PEGA	Best	0.81210	0.00000	0.00440	0.09280	0.21000
	Worst	53.33880	0.01800	0.01410	0.16610	0.53000
	Mean	25.30813	0.00396	0.00931	0.13289	0.30500
	Median	28.27340	0.00130	0.00945	0.13620	0.27000
	Std	15.06526	0.00578	0.00357	0.02381	0.10384

Table 4.16: The Results for Spacing Metric on Benchmark Test Problems 1-5 (2-D)

Method	Statistics	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
APC-MOPSO	Best	0.00520	0.00760	0.00760	0.00620	0.00630
	Worst	0.00770	0.01110	0.01100	0.00740	0.00850
	Mean	0.00627	0.00964	0.00905	0.00680	0.00732
	Median	0.00620	0.00980	0.00900	0.00680	0.00720
	Std	0.00067	0.00104	0.00085	0.00045	0.00068
2LB-MOPSO	Best	0.00610	0.00660	0.00660	0.03450	0.00500
	Worst	0.00810	0.00910	0.01020	0.77350	0.00760
	Mean	0.00715	0.00750	0.00811	0.15811	0.00590
	Median	0.00710	0.00750	0.00800	0.11790	0.00570
	Std	0.00052	0.00066	0.00099	0.15271	0.00066
NSGAII	Best	0.01100	0.01790	0.01250	0.38050	0.01910
	Worst	0.06860	0.04860	0.06400	42.11720	0.13080
	Mean	0.02422	0.02916	0.02875	4.28973	0.05025
	Median	0.02280	0.02860	0.02400	1.77030	0.04460
	Std	0.01242	0.00756	0.01384	8.40363	0.02555
SPEA2	Best	0.00410	0.00640	0.00550	1.34190	0.03660
	Worst	0.16680	1.00000	0.14310	77.63700	0.37370
	Mean	0.02799	0.42528	0.03803	10.53770	0.20668
	Median	0.01930	0.08080	0.02300	3.47700	0.19985
	Std	0.03136	0.48027	0.03851	16.67469	0.11874
MOPSO	Best	0.01310	0.00300	0.00820	0.01520	0.03160
	Worst	0.02910	1.00000	0.04100	1.39870	0.32560
	Mean	0.02094	0.69903	0.02069	0.26096	0.11661
	Median	0.02170	1.00000	0.02050	0.06500	0.10690
	Std	0.00454	0.46527	0.00758	0.39013	0.07643
PEGA	Best	0.00970	0.00080	0.01130	0.80970	0.01420
	Worst	0.04720	0.09460	0.03460	23.08210	0.04110
	Mean	0.02089	0.01406	0.02247	5.68275	0.02189
	Median	0.01730	0.00970	0.02240	2.04200	0.02090
	Std	0.01053	0.01882	0.00687	7.34578	0.00683

Table 4.17: The Results for Spacing Metric on Benchmark Test Problems 6-10 (2-D)

Method	Statistics	FONSECA	KURSAWE	OKA2	DTLZ1	DTLZ2
APC-MOPSO	Best	0.00140	0.05120	0.02260	0.00490	0.00605
	Worst	0.00160	0.10710	0.03300	0.00560	0.00670
	Mean	0.00155	0.07681	0.02967	0.00533	0.00624
	Median	0.00160	0.06330	0.03010	0.00550	0.00620
	Std	0.00008	0.02319	0.00285	0.00026	0.00019
2LB-MOPSO	Best	0.00110	0.04290	0.02590	0.00230	0.00590
	Worst	0.00140	0.10480	0.03250	0.00350	0.00670
	Mean	0.00129	0.06457	0.02801	0.00296	0.00628
	Median	0.00130	0.05420	0.02730	0.00315	0.00630
	Std	0.00006	0.02116	0.00207	0.00046	0.00027
NSGAII	Best	0.00120	0.04690	0.02330	0.00990	0.00630
	Worst	0.00150	0.10690	0.27720	0.04110	0.00700
	Mean	0.00131	0.07892	0.07315	0.02079	0.00663
	Median	0.00130	0.08870	0.03330	0.01520	0.00660
	Std	0.00009	0.01945	0.10034	0.01289	0.00035
SPEA2	Best	0.00180	0.06700	0.03140	0.49960	0.00340
	Worst	0.00220	0.16880	0.14020	7.01610	0.00970
	Mean	0.00203	0.08922	0.05157	3.48157	0.00497
	Median	0.00200	0.08630	0.04010	3.26990	0.00405
	Std	0.00010	0.01868	0.03219	2.03365	0.00198
MOPSO	Best	0.00210	0.06510	1.00000	0.04330	0.00800
	Worst	0.00280	0.16010	1.00000	4.10340	0.01740
	Mean	0.00236	0.08868	1.00000	1.45905	0.01167
	Median	0.00240	0.08330	1.00000	0.83090	0.01040
	Std	0.00017	0.02180	0.00000	1.43332	0.00346
PEGA	Best	0.00200	0.06140	0.01260	0.00330	0.00860
	Worst	0.00230	0.11770	0.18470	3.53970	0.02160
	Mean	0.00218	0.09030	0.05089	0.52290	0.01472
	Median	0.00220	0.08670	0.03830	0.05910	0.01470
	Std	0.00008	0.01961	0.04744	0.97500	0.00513

Table 4.18: The Results for Spacing Metric on Benchmark Test Problems 11-15 (2-D)

Method	Statistics	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
APC-MOPSO	Best	0.00860	0.00410	0.00300	0.00390	0.00910
	Worst	0.01110	0.00760	0.00340	0.00440	0.01070
	Mean	0.00986	0.00632	0.00322	0.00415	0.00993
	Median	0.01000	0.00640	0.00320	0.00410	0.00975
	Std	0.00081	0.00094	0.00017	0.00019	0.00056
2LB-MOPSO	Best	0.00500	0.00490	0.000169	0.00310	0.00680
	Worst	0.25450	0.00690	0.000191	0.00400	0.00830
	Mean	0.03287	0.00615	0.000183	0.00347	0.00771
	Median	0.00590	0.00620	0.000183	0.00345	0.00795
	Std	0.07800	0.00069	0.000007	0.00029	0.00060
NSGAII	Best	0.00680	0.00570	0.000155	0.00750	0.00670
	Worst	0.05370	0.00870	0.000170	0.04900	0.01120
	Mean	0.02260	0.00665	0.000163	0.02097	0.00957
	Median	0.00730	0.00610	0.000163	0.01405	0.01080
	Std	0.02693	0.00138	0.000005	0.01537	0.00249
SPEA2	Best	2.07480	0.00370	0.00020	0.01150	0.00350
	Worst	13.44780	1.00000	0.00270	0.06600	0.23400
	Mean	6.41544	0.10395	0.00053	0.02988	0.02611
	Median	5.55065	0.00460	0.00030	0.02880	0.00490
	Std	3.80201	0.31484	0.00077	0.01631	0.05953
MOPSO	Best	0.39870	0.00790	0.00015	0.00450	0.21000
	Worst	16.17870	1.00000	0.00017	0.04080	0.53000
	Mean	4.94942	0.66999	0.00016	0.00925	0.30500
	Median	2.65135	1.00000	0.00016	0.00580	0.27000
	Std	5.35356	0.49502	0.00001	0.01112	0.10384
PEGA	Best	0.06970	0.00000	0.00040	0.01190	0.00260
	Worst	45.89020	0.02490	0.01430	0.05860	0.09700
	Mean	12.38026	0.00946	0.00672	0.02288	0.02891
	Median	9.32280	0.00955	0.00615	0.01825	0.02250
	Std	11.56206	0.00672	0.00478	0.01377	0.02464

4.4.5.2 Results for Three-Objective Test Problems

This section presents the results derived from applying the proposed APC-MOPSO and 2LB-MOPSO to the benchmark test problems DTLZ2, DTLZ5, and DTLZ7. Two cases are considered in this section; the first case is conducted using 25 independent runs, a maximum of 10,000 function evaluations, and a population size of 100 individuals. In the second case, 15 independent runs, a maximum of 10,000 function evaluations, and a population size of 1000 individuals are used.

The reason behind selecting these three benchmark test problems is that their Pareto-optimal front correspond to $x_i = 0.5$ for all $x_i \in x_M$, (where x_i represents a solution, and M is the number of objectives) and all objective function values (i.e., fitness values) must satisfy the condition (i.e., referred to as a Pareto-front condition) given in Eq. (4.28).

By using the Pareto-front condition, the solutions produced by the APC-MOPSO, the 2LB-MOPSO, the NSGAII, the MOPSO, the SPEA2, and the PEGA can be compared based on the average value returned by each algorithm. Thus, it can be seen that the closer the average value is to 1.0, the better the performance in favor of converging and diversity. The results obtained from the two cases and their statistical analyses against the selected benchmark test problems are discussed as follows.

- **DTLZ2 Test Problem**

The obtained solutions are shown in Figures 4.35 and 4.36. The proposed APC-MOPSO obtained a better Pareto-front (i.e., returned an average value ranging from [1.00045 to 1.0465]), compared to 2LB-MOPSO, NSGA-II, MOPSO, and PEGA (i.e., returned average values within the ranges of [1.0022, 1.19899], [1.00334, 1.1192], [1.00768, 1.3455], and [1.01312, 1.52378], respectively, as shown in Table 4.19). For SPEA2, Figure 4.36 and Table 4.19 clearly show that the SPEA2 failed to reach the Pareto-optimal front (i.e., returned an average value of 1.01088 to 2.82756)[19]. Figures 4.35 and 4.36 show that the proposed APC-MOPSO outperformed all other algorithms and successfully returned an average value that is closer to the reference point of 1.0 (i.e., the value given by the Pareto-front condition shown in Eq. (4.28)). Table 4.19 supports the conclusion made by observing the graphical representation of the results in Figures 4.35 and 4.36. In this table, statistical analysis such as best, worst, mean, median, and standard deviation values based on a Pareto-front condition are provided for each algorithm. Figures 4.37 and 4.38 show the solutions obtained for the second case (i.e., with a population size of 1000 individuals) on a DTLZ2 test problem in two different views. These figures imply that both APC-MOPSO and 2LB-MOPSO algorithms converged to the global region after 10,000 function evaluations.

[19] In this section, and for the sake of the comparison among the results of the six algorithms on DTLZ2 and DTLZ5 benchmark test problems, it is decided to consider the algorithm that returns an average value within a range bigger than 2.0 to be unsuccessful.

However, to evaluate the performance of both algorithms against DTLZ2, 25 independent runs are conducted, and the Pareto-front condition is applied to sum up the function values for each run. The sum of the objectives of each solution for DTLZ2 is determined, and presented in Table 4.20, whereas the average for the minimum and maximum function values of each run is displayed. Here, it is found that the range of the objective values for APC-MOPSO is [1.0, 1.055], and is [1.0, 1.1577] for the 2LB-MOPSO. Accordingly, the proposed APC-MOPSO has the ability to produce more accurate non-dominated solutions than the 2LB-MOPSO.This test again demonstrated that the proposed APC-MOPSO substantively performed better than all other algorithms.

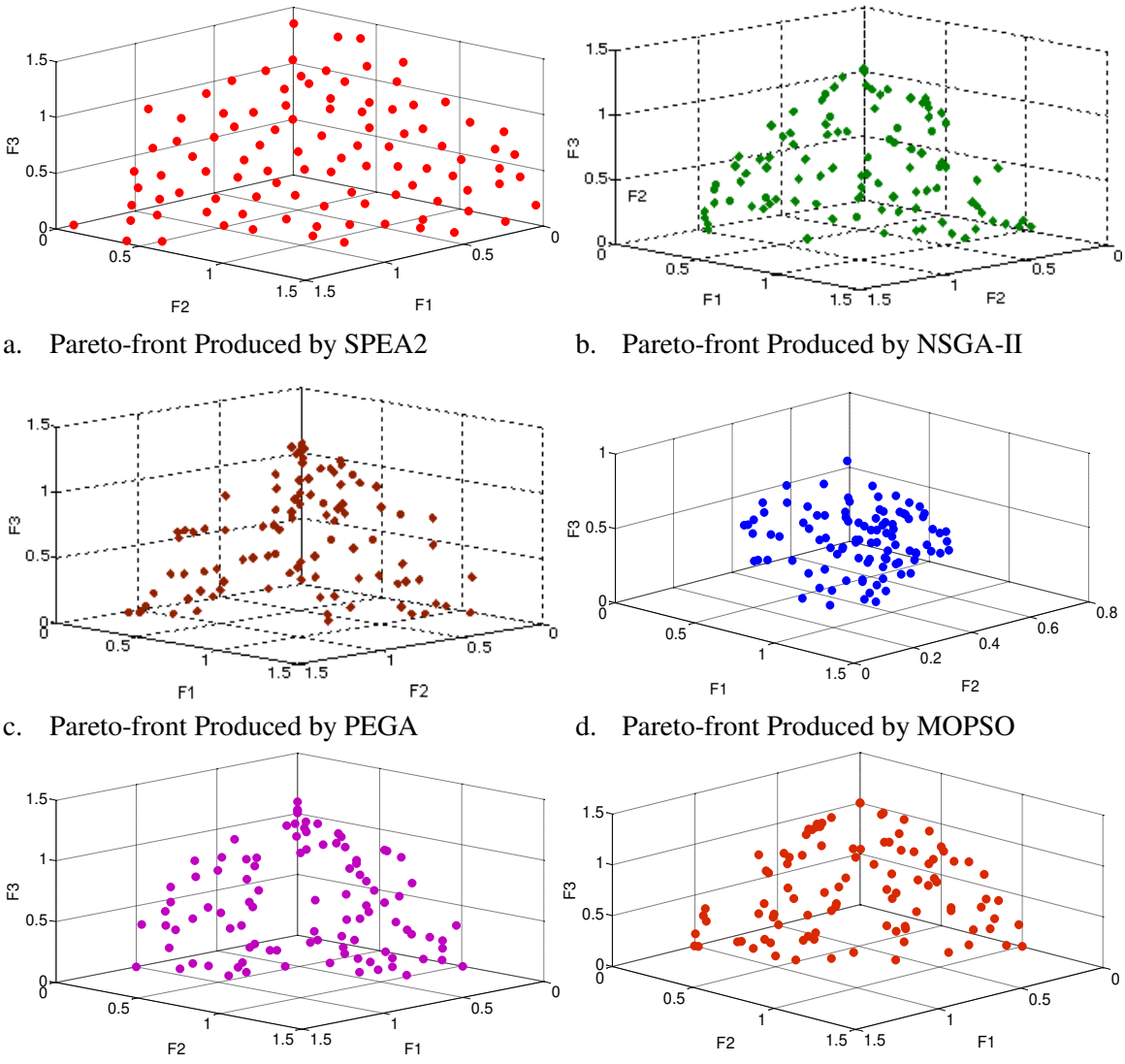
- **DTLZ5 Test Problem**

For the case of using a population size of 100 individuals, the solutions produced by the APC-MOPSO, the 2LB-MOPSO, the NSGA-II, the MOPSO, the SPEA2, and the PEGA for the DTLZ5 benchmark test problem showed that for each non-dominated solution (x_i), the sum of its squared objective values is equal to 1.0 (as shown in Eq. (4.28)). Table 4.21 shows the statistical results for every algorithm. The APC-MOPSO algorithm returned an average value within the range of [1.00005, 1.01126], while 2LB-MOPSO returned an average value ranging from [1.00006, 1.02214]. MOPSO, NSGA-II, SPEA2, and PEGA failed to converge to the global fronts (i.e., these four algorithms returned average values that are greater than 2.0). Figure 4.40 visually demonstrated the solution fronts obtained by the two algorithms for three-objective DTLZ5 benchmark test problems using a population size of 1000 individuals. This figure implies that both algorithms converged to Pareto-optimal curve, but still APC-MOPSO is able to locate solutions on the region of the Pareto front closer than 2LB-MOPSO algorithm. For this case, Table 4.22 shows that APC-MOPSO returned an average value within the range of [1.0, 1.00358] and 2LB-MOSO produced solutions with average values ranging from 1.0 to 1.00386. As can be seen here, increasing the size of the population could enhance the performance of the algorithm. It should be pointed out that this inference mainly depends on the complexity of the test problems.

- **DTLZ7 Test Problem**

This problem is chosen to test the ability of APC-MOPSO, 2LB-MOPSO, NSGA-II, MOPSO, SPEA2, and PEGA in producing non-dominated solutions that cover four Pareto-optimal regions as shown in Figure 4.41. It is important to note that APC-MOPSO is the only algorithm that converged to the optimal fronts and produced solutions that are spread into all four global regions (i.e., global regions 1, 2, 3, and 4). 2LB-MOPSO converged to the optimal fronts and produced a limited number of solutions that covered the global regions 1 and 4, while NSGA-II failed to converge. Both SPEA2 and PEGA converged to a Pareto-optimal front and partially covered the four global regions. It can also be seen from Figure 4.41, that SPEA2 did not explore either global region 1 or global region 3. MOPSO converged only to one global region that is

region 1. In the case where 1000 individuals are used, Figures 4.42 and 4.43 illustrate the Pareto-front sets produced by the APC-MOPSO and the 2LB-MOPSO. It can be seen that the APC-MOPSO is more successful in locating more solutions and covering wider front areas in the four global regions than the 2LB-MOPSO. In general, this test also indicated that the proposed APC-MOPSO is superior in giving solutions that are close and spread well over disconnected Pareto-optimal fronts.



a. Pareto-front Produced by SPEA2

b. Pareto-front Produced by NSGA-II

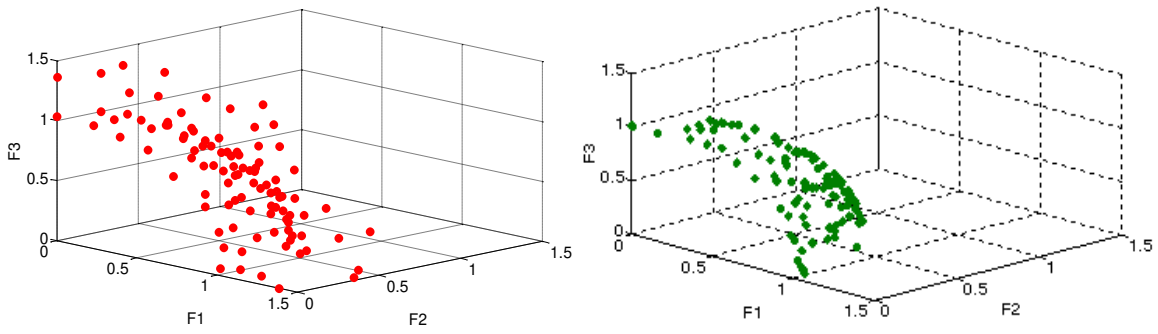
c. Pareto-front Produced by PEGA

d. Pareto-front Produced by MOPSO

e. Pareto-front Produced by 2L-MOPSO

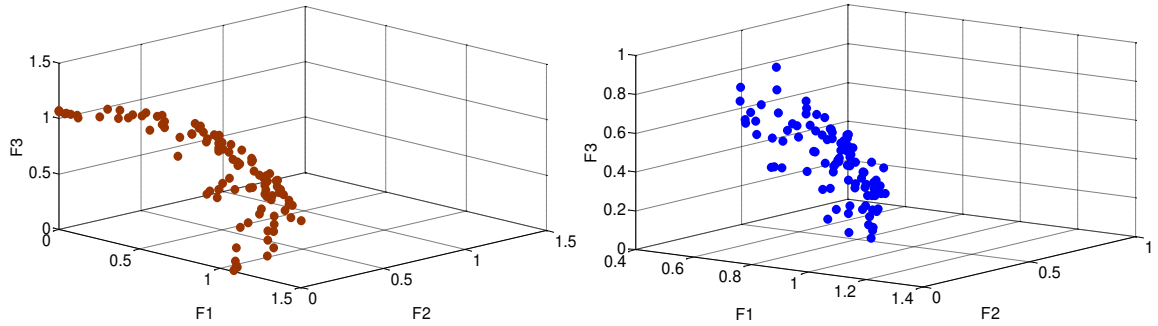
f. Pareto-front Produced by APC-MOPSO

Figure 4.35: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ2 Benchmark Test Problem (3-D) 1st View (100 Particles)



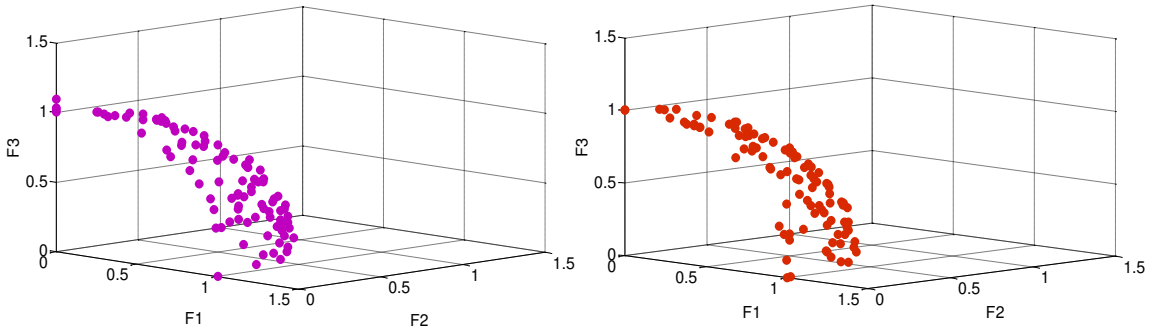
a. Pareto-front Produced by SPEA2

b. Pareto-front Produced by NSGA-II



c. Pareto-front Produced by PEGA

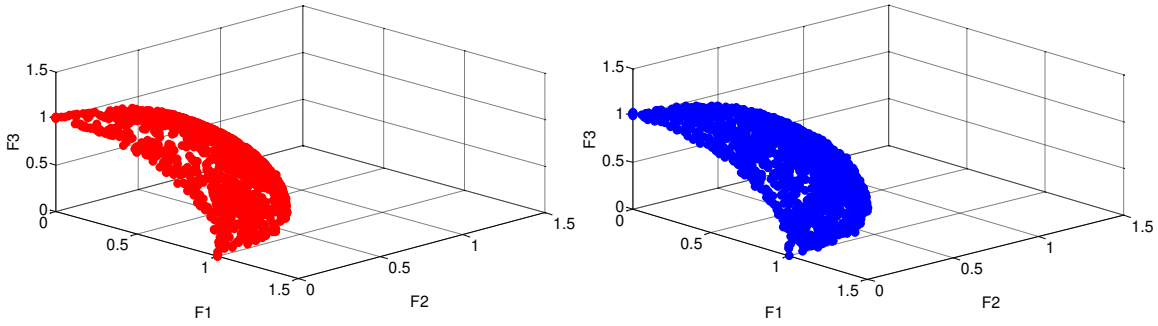
d. Pareto-front Produced by MOPSO



e. Pareto-front Produced by 2L-MOPSO

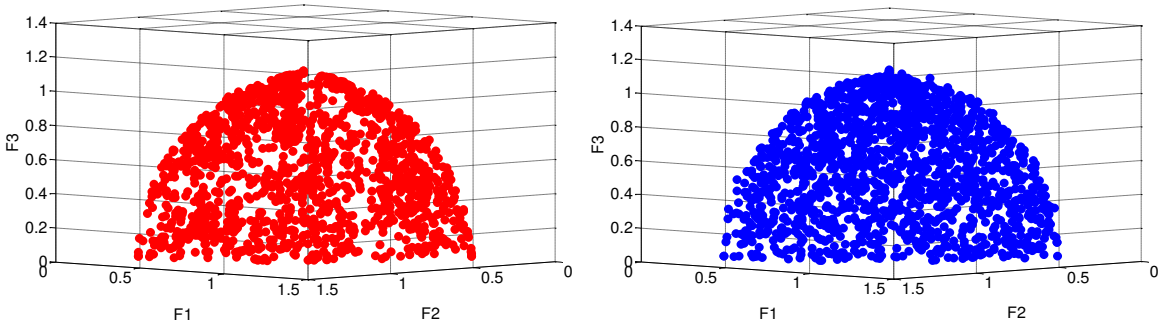
f. Pareto-front Produced by APC-MOPSO

Figure 4.36: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ2 Benchmark Test Problem (3-D) 2nd View (100 Particles)



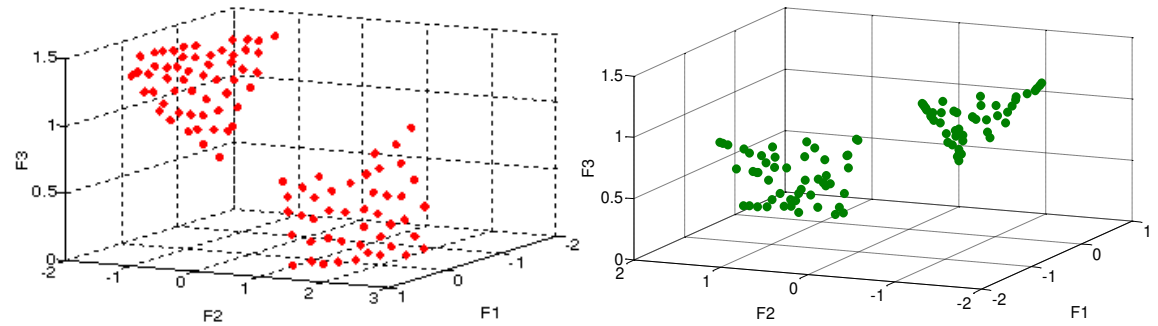
a. Pareto-front Produced by APC-MOPSO b. Pareto-front Produced by 2LB-MOPSO

Figure 4.37: Pareto-fronts Produced by the APC-MOPSO and 2LB-MOPSO for the DTLZ2 Benchmark Test Function (3-D) 1st View (1000 Particles)

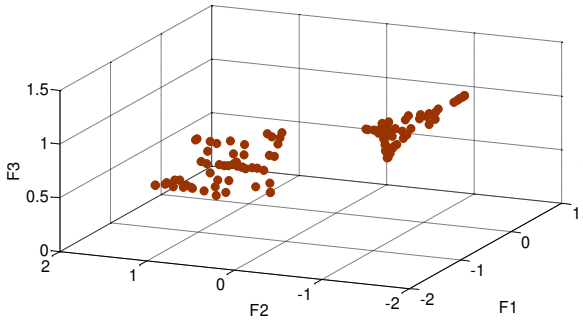


a. Pareto-front Produced by APC-MOPSO b. Pareto-front Produced by 2LB-MOPSO

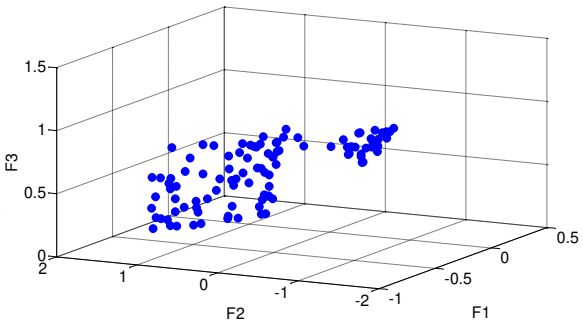
Figure 4.38: Pareto-fronts Produced by the APC-MOPSO and 2LB-MOPSO for the DTLZ2 Benchmark Test Function (3-D) 1st View (1000 Particles)



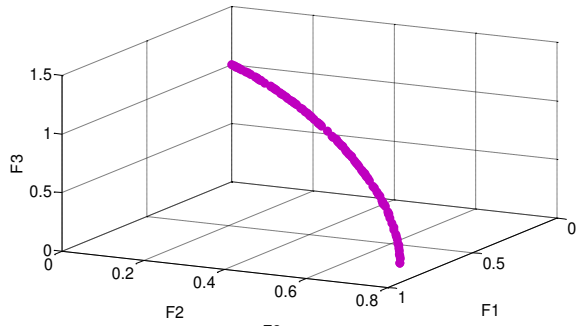
a. Pareto-front Produced by SPEA2 b. Pareto-front Produced by NSGA-II



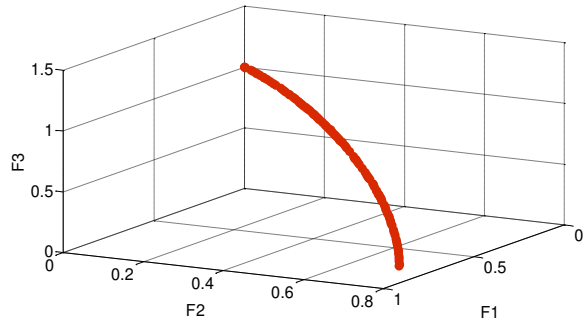
c. Pareto-front Produced by PEGA



d. Pareto-front Produced by MOPSO

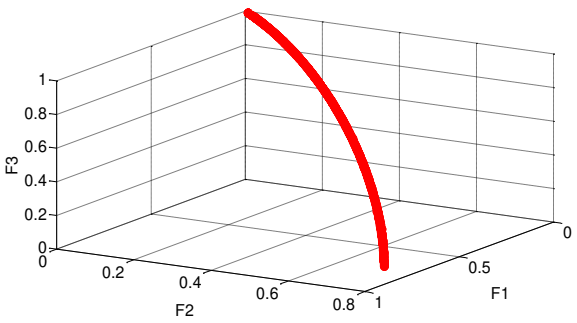


e. Pareto-front Produced by 2LB-MOPSO

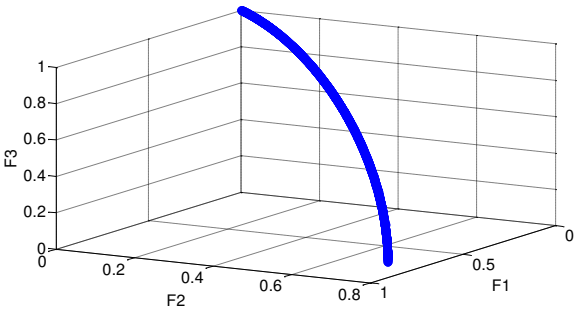


f. Pareto-front Produced by APC-MOPSO

Figure 4.39: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ5 Benchmark Test Problem (3-D) (100 Particles)

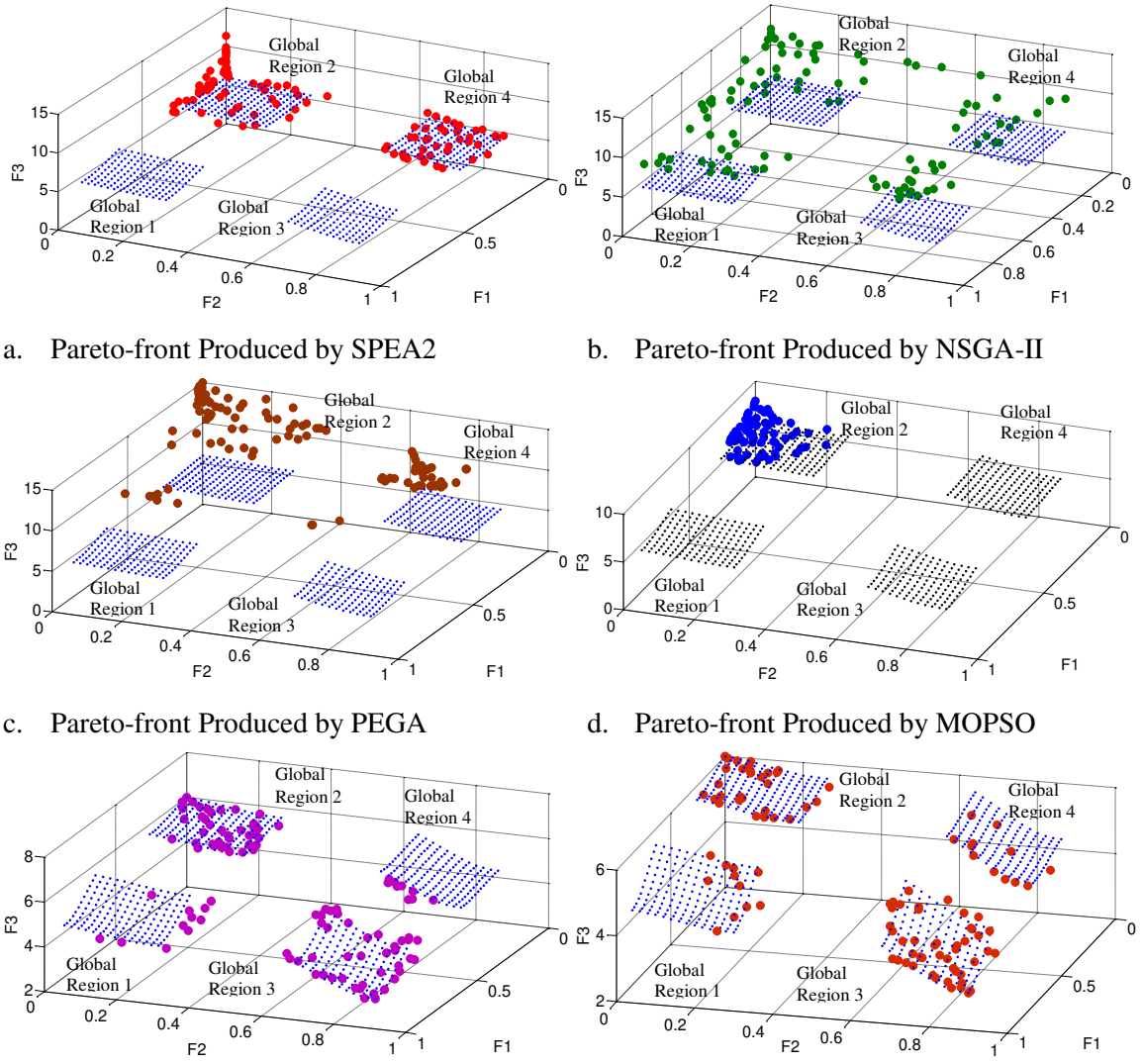


a. Pareto-front Produced by APC-MOPSO



b. Pareto-front Produced by 2LB-MOPSO

Figure 4.40: Pareto-Fronts Produced by the APC-MOPSO and 2LB-MOPSO for the DTLZ5 Benchmark Test Function (3-D) (1000 Particles)



a. Pareto-front Produced by SPEA2

b. Pareto-front Produced by NSGA-II

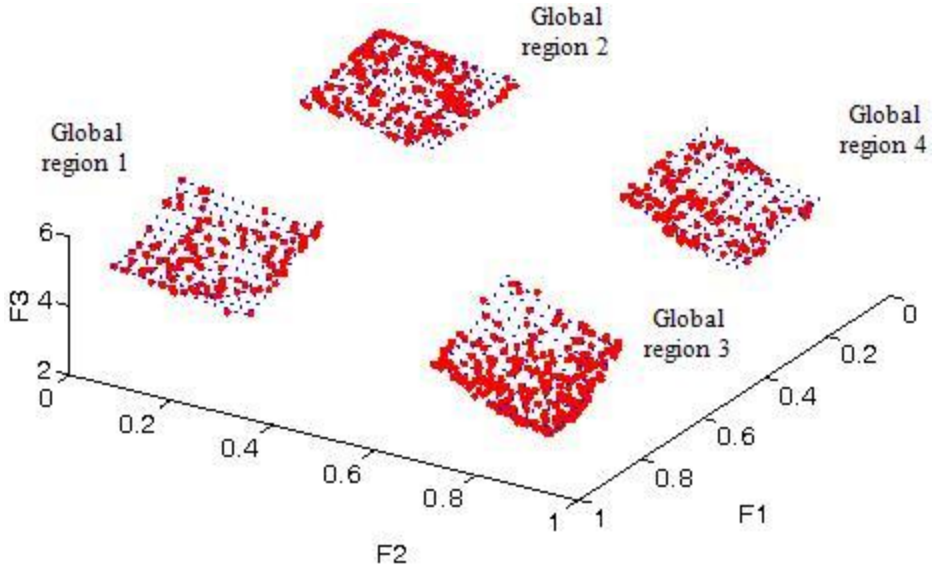
c. Pareto-front Produced by PEGA

d. Pareto-front Produced by MOPSO

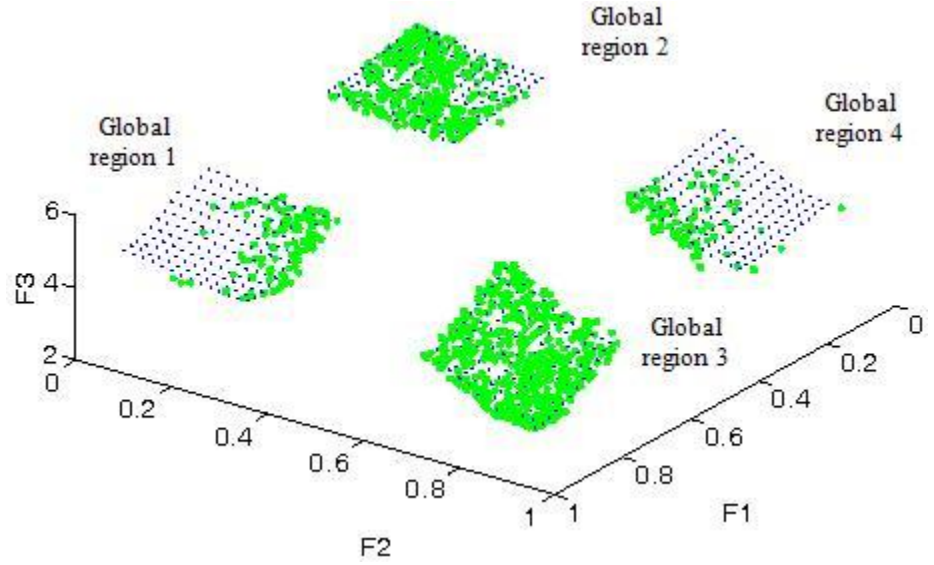
e. Pareto-front Produced by 2LB-MOPSO

f. Pareto-front Produced by APC-MOPSO

Figure 4.41: Pareto-front Produced by SPEA2, NSGA-II, PEGA, MOPSO, 2LB-MOPSO, and APC-MOPSO for the DTLZ7 Benchmark Test Problem (3-D) (100 Particles)

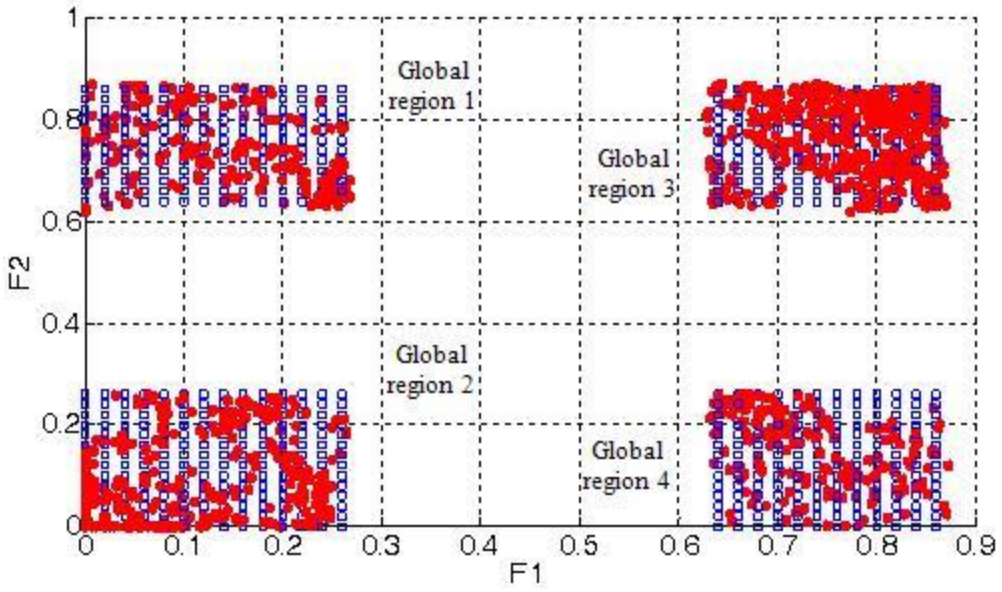


a. Pareto-front Produced by APC-MOPSO

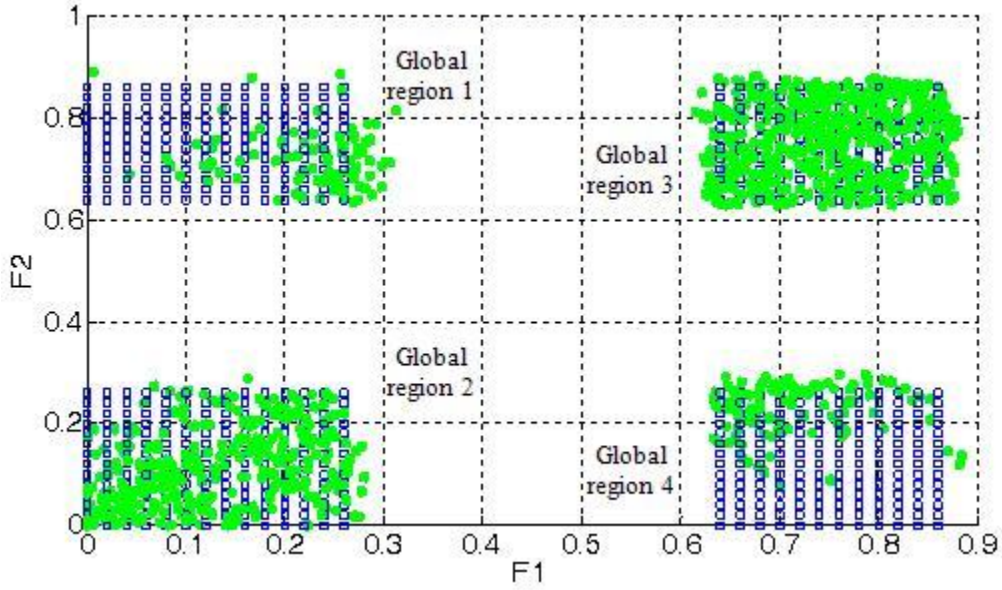


b. Pareto-front Produced by 2LB-MOPSO

Figure 4.42: Pareto-Fronts Produced by the APC-MOPSO and 2LB-MOPSO for the DTLZ7 Benchmark Test Function (3-D) 1st View (1000 Particles)



a. Pareto-front Produced by APC-MOPSO



b. Pareto-front Produced by 2LB-MOPSO

Figure 4.43: Pareto-Fronts Produced by the APC-MOPSO and 2LB-MOPSO for the DTLZ7 Benchmark Test Function (3-D) 2nd View (1000 Particles)

Table 4.19: The Range of Results for DTLZ2 Benchmark Test Problem (3-D) (100 Particles)

DTLZ2			
100 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.00030	1.02640
	Worst	1.00070	1.06280
	Mean	1.00045	1.04650
	Median	1.00044	1.04500
	Std	0.00016	0.01437
2LB-MOPSO	Best	1.00040	1.153200
	Worst	1.00480	1.285200
	Mean	1.00220	1.198980
	Median	1.00190	1.180700
	Std	0.00162	0.051859
NSGA-II	Best	1.00190	1.084400
	Worst	1.00470	1.194500
	Mean	1.00334	1.119200
	Median	1.00360	1.102700
	Std	0.00107	0.045347
SPEA2	Best	1.00630	2.252300
	Worst	1.01890	3.947000
	Mean	1.01098	2.827560
	Median	1.00970	2.547500
	Std	0.00504	0.714232
MOPSO	Best	1.00570	1.208400
	Worst	1.00990	1.628900
	Mean	1.00766	1.343580
	Median	1.00750	1.291000
	Std	0.00155	0.164210
PEGA	Best	1.00270	1.195100
	Worst	1.02190	1.981800
	Mean	1.01312	1.523780
	Median	1.01610	1.465200
	Std	0.00933	0.298204

Table 4.20: The Range of Results for DTLZ2 Benchmark Test Problem (3-D) (1000 Particles)

DTLZ2			
100 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.00000	1.019800
	Worst	1.00000	1.095400
	Mean	1.00000	1.055043
	Median	1.00000	1.046700
	Std	1.00000	0.027973
2LB-MOPSO	Best	1.00000	1.108300
	Worst	1.00000	1.140400
	Mean	1.00000	1.157700
	Median	1.00000	1.076400
	Std	1.00000	1.092100

Table 4.21: The Range of Results for DTLZ5 Benchmark Test Problem (3-D) (100 Particles)

DTLZ5			
100 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.00000	1.010000
	Worst	1.00010	1.013100
	Mean	1.00005	1.011260
	Median	1.00000	1.011200
	Std	0.00005	0.001150
2LB-MOPSO	Best	1.00000	1.014900
	Worst	1.00020	1.027900
	Mean	1.00006	1.022140
	Median	1.00000	1.023400
	Std	0.00009	0.005301
NSGA-II	Best	1.00850	4.267300
	Worst	1.01320	4.747900
	Mean	1.01046	4.513960
	Median	1.00950	4.617700
	Std	0.00200	0.214307

SPEA2	Best	1.01000	5.010600
	Worst	1.04760	5.046600
	Mean	1.03220	5.029120
	Median	1.03030	5.027400
	Std	0.01526	0.014589
MOPSO	Best	1.04330	3.220900
	Worst	1.12390	5.042500
	Mean	1.07992	4.037620
	Median	1.07030	3.857000
	Std	0.03803	0.720195
PEGA	Best	1.00850	4.002800
	Worst	1.04520	4.479900
	Mean	1.02496	4.353260
	Median	1.02120	4.430900
	Std	0.01404	0.197796

Table 4.22: The Range of Results for DTLZ5 Benchmark Test Problem (3-D) (1000 Particles)

DTLZ5			
1000 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.00000	1.003200
	Worst	1.00000	1.003900
	Mean	1.00000	1.003580
	Median	1.00000	1.003600
	Std	0.00000	0.000249
2LB-MOPSO	Best	1.00000	1.003200
	Worst	1.00000	1.004700
	Mean	1.00000	1.003867
	Median	1.00000	1.003700
	Std	0.00000	0.000764

4.4.5.3 Results for Five-Objective Test Problems

Another set of experiments are performed on five-objective DTLZ2 and DTLZ5 benchmark test problems to provide both qualitative and quantitative performance comparisons of the APC-MOPSO with other algorithms that are 2LB-MOPSO, NSGA-II, MOPSO, SPEA2, and PEGA. Two population sizes of 100 and 1000 individuals are considered to study the effect of increasing the population size on the performance in terms of the exploring ability of the algorithm in finding global solutions to problems that have a multimodal and a disconnected Pareto-front.

- **DTLZ2 Test Problem**

In the case of 100 individuals, the mean values of 15 runs achieved by the APC-MOPSO and the other algorithms that are 2LB-MOPSO, NSGA-II, MOPSO, SPEA2, and PEGA at 10,000 function evaluations are compared on the DTLZ2 test problem (see Section 4.4.2). As shown in Table 4.23, all solutions are compared with respect to the reference value of 1.0 (i.e., this value represents the sum of the square of all objectives in each solution, as shown in Eq. (4.28)). From this table, the average values of the solutions produced by the proposed APC-MOPSO are in the range of [1.00959, 1.15650], the NSGA-II's are in the range of 1.02336 to 1.67188, and the MOPSO's ranged from 1.0118 to 1.255. The 2LB-MOPSO, SPEA2, and PEGA failed to converge to the Pareto-optimal front. It is clear that the proposed APC-MOPSO outperformed all other algorithms by returning the closest average value to 1.0. Finally, for the case of 1000 particles, the results obtained by the APC-MOPSO and the 2LB-MOPSO are in the ranges of [1.00031, 1.6594] and [1.00034, 2.08047], respectively. Table 4.24 indicates that the APC-MOPSO performed better than the 2LB-MOPSO.

- **DTLZ5 Test Problem**

The last test in this section is used to compare the results produced by the proposed APC-MOPSO, the 2LB-MOPSO, the NSGA-II, the MOPSO, the SPEA2, and the PEGA algorithms on the five-objective DTLZ5 problem (with 10 decision variables). The performance of the six algorithms is compared with respect to the Pareto-optimal front (see Eq. (4.28)). For the case of 100 individuals, APC-MOPSO converged to the global region of the problem, and found solutions with an average value in the range of [1.0753, 1.78806] as shown in Table 4.25. All other algorithms are failed to converge to the true Pareto-front. Table 4.26 shows the results obtained for the case where a size of 1000 individuals is considered. In this case, both APC-MOPSO and 2LB-MOPSO are tested. A range of [1.00069 to 1.3846] is returned by the APC-MOPSO, while 2LB-MOPSO is failed to converge. The conclusion drawn from this test is that the APC-MOPSO has a huge potential to consistently solve problems with high dimensions in an effective and stable manner.

Table 4.23: The Range of Results for DTLZ2 Benchmark Test Problem (5-D) (100 Particles)

DTLZ2			
100 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.00842	1.02120
	Worst	1.01080	1.17430
	Mean	1.00959	1.12400
	Median	1.00931	1.15650
	Std	0.00092	0.06232
2LB-MOPSO	Best	1.01150	2.408600
	Worst	1.05380	2.707800
	Mean	1.03354	2.556900
	Median	1.03080	2.541700
	Std	0.01561	0.130275
NSGA-II	Best	1.00980	1.565400
	Worst	1.04740	1.788600
	Mean	1.02336	1.671880
	Median	1.02320	1.663200
	Std	0.01511	0.097096
SPEA2	Best	1.07150	5.553800
	Worst	1.31140	5.982900
	Mean	1.16848	5.737720
	Median	1.16840	5.722700
	Std	0.09165	0.155800
MOPSO	Best	1.00650	1.190300
	Worst	1.01570	1.327700
	Mean	1.01180	1.255440
	Median	1.01290	1.254600
	Std	0.00364	0.056564
PEGA	Best	1.12420	2.700600
	Worst	2.06180	4.018400
	Mean	1.36242	3.243520
	Median	1.16440	3.149600
	Std	0.39736	0.502471

Table 4.24: The Range of Results for DTLZ2 Benchmark Test Problem (5-D) (1000 Particles)

DTLZ2			
1000 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.00028	1.611400
	Worst	1.00034	1.685500
	Mean	1.00031	1.659433
	Median	1.00031	1.681400
	Std	0.00003	0.041649
2LB-MOPSO	Best	1.00010	2.07930
	Worst	1.00050	2.08150
	Mean	1.00034	2.08047
	Median	1.00042	2.08060
	Std	0.00021	0.00111

Table 4.25: The Range of Results for DTLZ5 Benchmark Test Problem (5-D) (100 Particles)

DTLZ5			
100 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.02140	1.50560
	Worst	1.16380	1.97600
	Mean	1.07520	1.78806
	Median	1.06530	1.76730
	Std	0.05666	0.19306
2LB-MOPSO	Best	1.00140	5.124600
	Worst	1.00830	6.250000
	Mean	1.00402	5.743800
	Median	1.00400	5.701000
	Std	0.00284	0.505098
NSGA-II	Best	1.01670	5.024400
	Worst	1.06800	5.690800
	Mean	1.03534	5.426880
	Median	1.03450	5.521000
	Std	0.02087	0.259993

SPEA2	Best	1.03970	6.093000
	Worst	2.39860	6.227200
	Mean	1.41426	6.174860
	Median	1.16180	6.182700
	Std	0.55946	0.049387
MOPSO	Best	1.09370	3.285100
	Worst	2.34990	5.335700
	Mean	1.41238	3.915980
	Median	1.22460	3.611600
	Std	0.52957	0.825956
PEGA	Best	1.00970	4.915800
	Worst	1.42270	5.634400
	Mean	1.15088	5.304120
	Median	1.07740	5.344300
	Std	0.16330	0.274983

Table 4.26: The Range of Results for DTLZ5 Benchmark Test Problem (5-D) (1000 Particles)

DTLZ5			
1000 Particles with 10,000 FEs			
Method	**Statistics**	**Min**	**Max**
APC-MOPSO	Best	1.00010	1.35500
	Worst	1.00120	1.41260
	Mean	1.00069	1.38463
	Median	1.00078	1.38630
	Std	0.00056	0.02884
2LB-MOPSO	Best	1.01530	2.21390
	Worst	1.11350	2.22320
	Mean	1.06427	2.21870
	Median	1.06400	2.21900
	Std	0.04910	0.00466

4.4.5.4 Experimental Timing Analysis

A timing analysis of the experiments conducted in this chapter revealed an important consideration regarding the computational speed of the six algorithms. The platform of this analysis is designed to use a set of two, three, and five-objective benchmark test problems, a

population size of 100 individuals, a maximum of 10,000 function evaluations, and 25 independent runs for each problem to determine the average CPU[20] times for all algorithms. Figures 4.44 and 4.45 depict the average CPU time obtained for each algorithm in solving the benchmark test problems. For the two-objective tests, Figure 4.44 shows that the PEGA and the SPEA2 are executed faster than other algorithms with the same average CPU time of 26.9 seconds. It is important to note that for the PEGA algorithm, adopting a parallel mechanism enhanced its speed of computation. Moreover, Figure 4.44 shows that the NSGA-II has the longest average CPU time to converge compared to other algorithms (with an average CPU time of 551 seconds). Also, Figure 4.44 shows that the APC-MOPSO algorithm performed faster than the other MOPSO algorithms, and this is the main reason for employing both parallel islands model and parallel computing MATLAB toolbox to the proposed APC-MOPSO algorithm. The APC-MOPSO has an average CPU time of 65.7 seconds, while 2LB-MOPSO and MOPSO have average CPU times of 102.2 and 166.2 seconds, respectively.

Table 4.27 presents the average CPU time for each two-objective test problem. Furthermore, Table 4.28 shows the mean of CPU times needed for each algorithm to solve the whole set of the two-objective benchmark test problems. The last column of this table provides the number of the benchmark test problems that each algorithm did solve. Accordingly, NSGA-II and PEGA solved only seven problems out of a total of fifteen. The proposed APC-MOPSO is the only algorithm that successfully solved the whole suite of the benchmark test problems. The NSGA-II is the slowest algorithm in performing the search process.

For the three-objective experiments, 25 independent runs, a maximum of 10,000 function evaluations, three selected benchmark problems, and a population size of 100 individuals are used to calculate the average executing CPU time for each of the six algorithms. Tables 4.29 and 4.30 report the CPU times required to solve DTLZ2, DTLZ5, and DTLZ7 by the APC-MOPSO, the 2LB-MOPSO, the NSGA-II, the MOPSO, the SPEA2, the PEGA. As shown in the last column of Table 4.30, PEGA, MOPSO, and SPEA2 solved only one test problem out of the selected three. NSGA-II failed to solve DTLZ5, while 2LB-MOPSO did not perform well on DTLZ7. Evidently, the proposed APC-MOPSO is the best as it is able to successfully solve all three benchmark test problems. Figure 4.45 shows that APC-MOPSO is faster in executing its algorithm than 2LB-MOPSO and MOPSO. In the case of 1000 individuals, two algorithms are only considered, in which the proposed APC-MOPSO needed an average CPU time of 3282 seconds (i.e. equivalent to ≈ 54 minutes) to solve the three problems compared to 4624 seconds (i.e., equivalent to ≈ 77.minutes) for 2LB-MOPSO (as shown in Table 4.31).

For the five-objective set of experiments, the timing analysis is conducted by using the benchmark problems DTLZ2 and DTLZ5, a maximum of 10,000 function evaluations, 15

[20] The PC used to conduct the timing analysis has the following specification: a Genuine Intel(R) CPU T2500 @ 2.00GHz and 2.00GB of RAM with WINDOWS 7.

independent runs, and a population size of 100 individuals. Table 4.32 reports the CPU time needed for each algorithm to solve the selected test problems. As per this table, APC-MOPSO required an average CPU time of 153.523 to successfully solve the test problems. Table 4.33 shows that the proposed APC-MOPSO successfully solved the test problems with a competitive average CPU time compared to the CPU times for the others. The 2LB-MOPSO, the SPEA2, and the PEGA failed to converge to the global front. NSGA-II and MOPSO solved one problem, namely DLTZ2. Even though these two algorithms obtained solutions for the DTLZ2, they did not outperform the APC-MOPOS. Hence, it is worth concluding that the APC-MOPSO is the only algorithm that successfully solved the hardest set of experiments with better results. In the case of using a population size of 1000 individuals, Table 4.34 shows the average CPU times needed for APC-MOPSO and 2LB-MOPSO to solve five-objective DTLZ2 and DTLZ5 benchmark test problems; 31370 seconds (i.e., equivalent to ≈ 9 hrs.) for the APC-MOPSO and 40217 seconds (i.e., equivalent to ≈ 11 hrs.) taken by the 2LB-MOPSO.

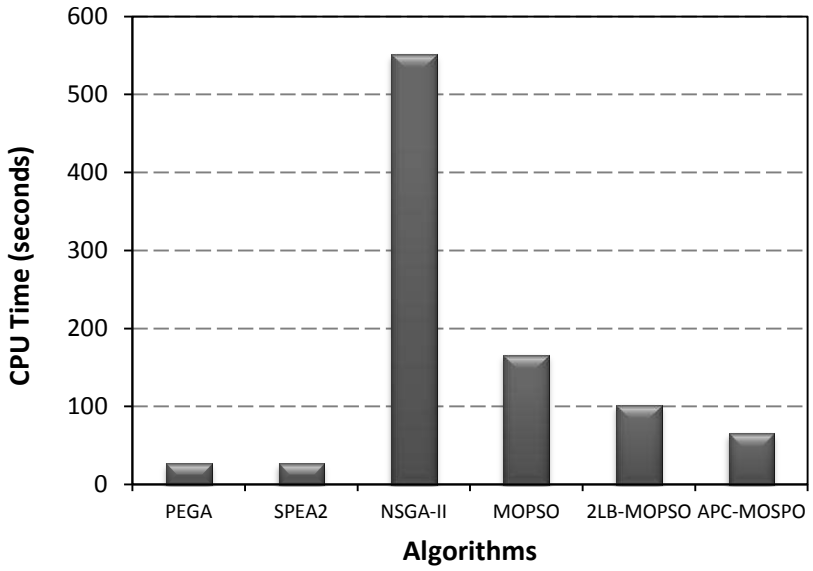


Figure 4.44: Timing Analysis for the Two-Objective Benchmark Test Problems

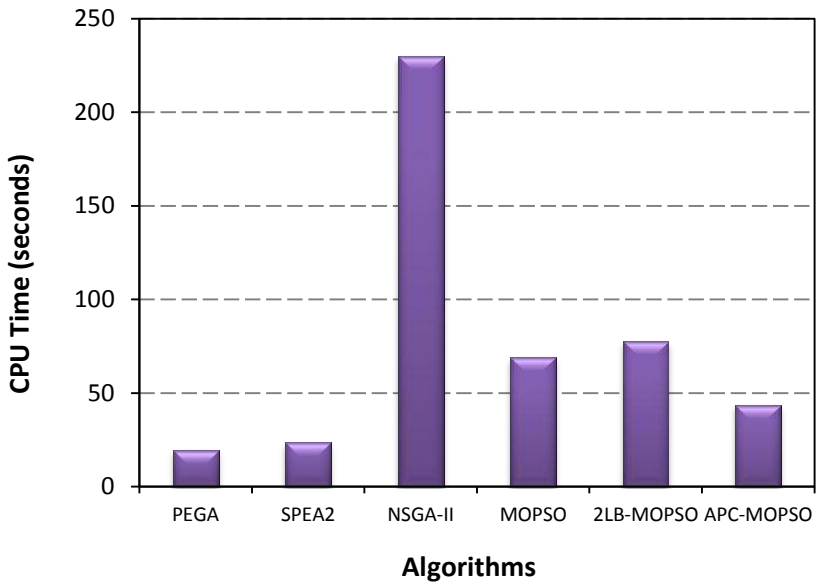


Figure 4.45: Timing Analysis for the Three-Objective Benchmark Test Problems

Table 4.27: Average CPU Times for the Two-Objective Test Problems (100 Particles)

Problems	**Average CPU Time* (in seconds)**					
2-Obective	**PEGA**	**SPEA2**	**NSGA-II**	**MOPSO**	**2LB-MOPSO**	**APC-MOSPO**
ZDT1	11.87	23.594	208.25	18.8594	44.5156	42.3419
ZDT2	12.0938	23.281	210.0781	17.500	46.5156	43.0971
ZDT3	11.375	22.969	208.9531	22.046	44.5312	42.1532
ZDT4	11.9688	23.906	215.968	16.5313	31.3594	39.4878
ZDT6	11.375	22.188	211.0938	19.125	48.9844	41.7793
FONSECA	170.9531	59.718	3997.012	1329.1	685.8594	285.224
KURSAWE	11.14	25.156	211.281	32.0156	49.2813	42.5446
OKA2	19.282	32.662	240.391	85.050	53.0625	44.794
DTLZ1	10.625	19.844	211.7344	36.6563	49.9219	42.5446
DTLZ2	10.9735	21.094	207.625	87.6406	48.3281	44.083
DTLZ3	10.9219	19.8531	213.9531	36.219	32.7656	42.2785
DTLZ4	12.6094	21.094	211.8906	51.875	46.000	41.2239
DTLZ5	43.3125	31.313	830.3906	354.2031	147.453	93.7009
DTLZ6	44.752	32.250	876.251	361.145	154.0313	98.625
DTLZ7	10.8750	25.140	210.2656	24.156	48.750	41.528

* *Average CPU Time is calculated by taking the mean time of 25 independent runs for each test problem and algorithm*

Table 4.28: Summary of the Algorithms for Two-Objective Test Problems (100 Particles)

Algorithm	Execution Speed (CPU Time in seconds)*	Language (Platform)	Problems Solved
APC-MOPSO	65.6937	MATLAB and C	**15 / 15**
2LB-MOPSO	102.0906	MATLAB and C	14 / 15
NSGA-II	551.0092	MATLAB	7 / 15
SPEA2	26.9425	MATLAB and C	10 / 15
MOPSO	166.1415	MATLAB	8 / 15
PEGA	26.9418	MATLAB	7 / 15

* *Execution CPU time is calculated for each algorithm as a mean time of all 2-objective test problems*

Table 4.29: Average CPU Times for the Three-Objectives Test Problems (100 Particles)

Problems	Average CPU Time* (in seconds)					
3-Obective	**PEGA**	**SPEA2**	**NSGA-II**	**MOPSO**	**2LB-MOPSO**	**APC-MOSPO**
DTLZ2	19.640	23.9531	229.3125	116.14	81.796	43.480
DTLZ5	19.328	23.468	230.2813	57.590	75.760	43.391
DTLZ7	19.676	23.5938	230.0781	33.2656	75.234	44.192

* *Average CPU Time is calculated by taking the mean time of 25 independent runs for each test problem and algorithm*

Table 4.30: Summary of Algorithms for Three-Objective Test Problems (100 Particles)

Algorithm	Execution Speed (CPU Time in seconds)*	Language (Platform)	Problems Solved
APC-MOPSO	43.687	MATLAB and C	**3 / 3**
2LB-MOPSO	77.597	MATLAB and C	2 / 3
NSGA-II	229.8906	MATLAB	2 / 3
SPEA2	23.6716	MATLAB and C	1 / 3
MOPSO	68.998	MATLAB	1 / 3
PEGA	19.548	MATLAB	1 / 3

* *Execution CPU time is calculated for each algorithm as a mean time of all 2-objective test problems*

Table 4.31: Average CPU Times for the Three-Objective Test Problems (1000 Particles)

Problems	**Average CPU Time* (in seconds)**	
3-Objective	**2LB-MOPSO**	**APC-MOSPO**
DTLZ2	4560	3210
DTLZ5	4472	3157
DTLZ7	4840	3480
Average	4624	3282

* *Average CPU Time is calculated by taking the average time of 15 independent runs for each test problem and algorithm*

Table 4.32: Average CPU Times for the Five-Objectives Test Problems (100 Particles)

Problems	**Average CPU Time* (in seconds)**					
5-Obective	**PEGA**	**SPEA2**	**NSGA-II**	**MOPSO**	**2LB-MOPSO**	**APC-MOSPO**
DTLZ2	20.4219	44.688	230.1406	159.328	315.043	152.764
DTLZ5	19.7188	45.000	230.9036	163.875	279.468	154.282

* *Average CPU Time is calculated by taking the mean time of 25 independent runs for each test problem and algorithm*

Table 4.33: Summary of the Algorithms for Five-Objective Test Problems (100 Particles)

Algorithm	**Execution Speed (CPU Time in seconds)***	**Language (Platform)**	**Problems Solved**
APC-MOPSO	153.523	MATLAB and C	**2 / 2**
2LB-MOPSO	297.2555	MATLAB and C	0 / 2
NSGA-II	230.5221	MATLAB	1 / 2
SPEA2	44.844	MATLAB and C	0 / 2
MOPSO	161.6015	MATLAB	1 / 2
PEGA	20.070	MATLAB	0 / 2

* *Execution CPU time is calculated for each algorithm as a mean time of all 2-objective test problems*

Table 4.34: Average CPU Times for the Five-Objective Test Problems (1000 Particles)

Problems	**Average CPU Time* (in seconds)**	
5-Objective	**2LB-MOPSO**	**APC-MOSPO**
DTLZ2	41142	31516
DTLZ5	39291	31223
Average	40217	36370

* *Average CPU Time is calculated by taking the average time of 15 independent runs for each test problem and algorithm*

4.5 Summary of Results

Generally, the simulation results demonstrated that the proposed APC-MOPSO performed better than the MOEAs represented by NSGA-II, SPEA2, and PEGA, and the well-known MOPSOs such as the original MOPSO and the 2LB-MOPSO. It can be observed that APC-MOPSO outperformed the others in terms of the distance of its solutions to the Pareto-optimal front (i.e., also known as true Pareto-front) and the distribution of the non-dominated solutions along the Pareto-optimal front. In this chapter, different benchmark test problems are used. These vary in terms of convexity, concavity, continuity, and linearity of the Pareto-optimal fronts. Convex problems (i.e., ZDT1) present the least amount of difficulty for the multi-objective optimizers.

On the other hand, the non-convex problems (i.e., like ZDT4 and OKA2) are more challenging. Only APC-MOPSO and 2LB-MOPSO can produce evenly distributed Pareto-front sets that are close to the Pareto-optimal fronts. Furthermore, it is clear that the APC-MOPSO outperformed the 2LB-MOPSO in terms of the high quality and the number of non-dominated solutions. In the case of discontinuous Pareto-front (i.e., ZDT3 and KURSAWE, and DTLZ7), APC-MOPSO is superior to all others. When considering performance metrics with respect to the benchmark test problems, a clear hierarchy of algorithms can and summarized as follows:

1. APC-MOPAO
2. 2LB-MOPSO
3. NSGA-II
4. MOPSO
5. PEGA
6. SPEA2

Furthermore, the APC-MOPSO algorithm is superior in its global search capability, finding high-quality non-dominated solutions, maintaining diversity (i.e., evenly distributed Pareto-front set), and enhanced ability to leap away from local optima (i.e., avoiding premature convergence).

For the two-objective experiments, Table 4.35 shows the results obtained by using the APC-MOPSO, the 2LB-MOPSO, the MOPSO, the NSGA-II, the SPEA2, and the PEGA. For the C metric, the APC-MOPSO produced the smallest mean and standard deviations for C metric than the other five methods in all of the benchmark problems, except in DTLZ2 and DTLZ5 where APC-MOPSO is ranked the second best.

The quality of all non-dominated solutions obtained by the APC-MOPSO, the2LB-MOPSO, the MOPSO, the NSGA-II, the SPEA2, and the PEGA after 10,000 function evaluation (i.e., equivalent to 100 iterations in the case of 100 individuals in the population) on fifteen two-objective benchmark test problems are measured by using the Generational Distance (GD) metric. The results provided in Table 4.35 stated that the. APC-MOPSO is considerably outperformed the

other algorithms in terms of returning the lowest mean and standard deviations of GD in most of the problems (except for DTLZ6 and DTLZ7, APC-MOPSO produced the second best values).

Furthermore, the results of the ER metric are also presented in Table 4.35 (i.e., ER is considered as one of the most common performance measures for multi-objective optimization algorithms). Accordingly, the APC-MOPSO and 2LB-MOPSO algorithms performed almost equally well for the whole two-objective problems. NSGA-II performed somewhat well on ZDT4, DTLZ4, and DTLZ5, while MOPSO, SPEA2, and PEGA returned high mean values of ER. Not surprisingly, the APC-MOPSO outperformed the other algorithms by returning the lowest mean value of ER for almost each test problem (except DTLZ2 test problem which obtained the second best), a fact that may largely be attributed to its outstanding performance on both C and GD metrics.

Lastly, the Spacing (S) metric is used to measure the spread and the distribution of the obtained Pareto-front solutions over a non-dominated region for each algorithm, and its results are provided in Table 4.35. The results indicated that the APC-MOPSO produced well-spread solutions and returned the best average values for six test problems. Moreover, the results showed that the APC-MOPSO obtained the second best in the rest of the fifteen test problems. In addition, the 2LB-MOPSO takes the lead in ZDT2, ZDT3, ZDT6, DTLZ1, DTLZ4, DTLZ7, and FONSECA; nevertheless, it is important to keep in mind its partial lack of performance in terms of other metrics (e.g., C, ER, and GD).

Table 4.35: Ranking of the Algorithms: Two-Objectives Test Problems

ZDT1						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	5	**1**	3	2	4	**1**
ER	5	5	3	2	5	**1**
GD	4	3	2	**1**	5	**1**
S	4	5	6	2	3	**1**
ZDT2						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	4	5	3	2	5	**1**
ER	5	5	3	2	5	**1**
GD	6	3	5	2	4	**1**
S	6	4	5	**1**	3	2
ZDT3						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	4	4	3	2	4	**1**
ER	4	5	3	2	5	**1**
GD	5	4	3	2	6	**1**
S	3	5	6	**1**	4	2

Table 4.35: Ranking of the Algorithms: Two-Objectives Test Problems (continue)

ZDT4						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	5	6	2	3	4	**1**
ER	2	2	2	2	2	**1**
GD	3	4	5	2	6	**1**
S	3	4	6	2	5	**1**
ZDT6						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	2	3	5	**1**	4	**1**
ER	3	4	4	2	4	**1**
GD	3	5	2	**1**	4	**1**
S	5	4	6	**1**	3	2
DTLZ1						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	4	4	4	2	3	**1**
ER	4	4	4	2	3	**1**
GD	4	2	5	**1**	3	**1**
S	5	3	6	**1**	4	2
DTLZ2						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	3	**1**	5	4	6	2
ER	**1**	4	5	3	6	2
GD	**1**	3	4	3	5	2
S	5	4	**1**	3	6	2
DTLZ3						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	3	3	3	2	3	**1**
ER	3	3	3	2	3	**1**
GD	5	3	4	2	6	**1**
S	4	2	5	3	6	**1**
DTLZ4						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	6	2	4	3	5	**1**
ER	6	3	5	2	4	**1**
GD	5	2	4	**1**	3	**1**
S	6	3	5	**1**	4	2

Table 4.35: Ranking of the Algorithms: Two-Objectives Test Problems (continue)

DTLZ5						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	3	**1**	6	5	4	2
ER	5	2	4	**1**	6	**1**
GD	**1**	**1**	4	3	5	**1**
S	5	4	**1**	2	6	3
DTLZ6						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	2	5	4	**1**	3	**1**
ER	3	6	4	2	5	**1**
GD	3	5	4	**1**	6	2
S	3	4	6	**1**	5	**1**
DTLZ7						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	4	6	3	2	5	**1**
ER	3	5	4	2	6	**1**
GD	**1**	4	3	2	5	2
S	6	3	4	**1**	5	2
KURSAWE						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	5	**1**	4	2	3	**1**
ER	4	2	6	3	5	**1**
GD	5	**1**	4	3	6	2
S	4	3	5	**1**	6	2
OKA2						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	3	2	3	**1**	3	**1**
ER	3	3	3	2	3	**1**
GD	5	4	2	**1**	3	**1**
S	6	5	4	2	3	**1**
FONSECA						
Metrics	**MOPSO**	**NSGAII**	**SPEA2**	**2LBMOPSO**	**PEGA**	**APC-MOPSO**
C	**1**	3	6	4	5	**1**
ER	2	3	6	4	5	**1**
GD	**1**	**1**	3	**1**	2	**1**
S	6	2	4	**1**	5	3

For the three-objective experiments, the rank of the six algorithms is provided in Table 4.36. This rank is obtained based on how far the solutions produced by each algorithm are from the Pareto-optimal front. For the DTLZ7 test problem, the visualization of the graphical representation of the produced solutions is used for determining and ranking performance. Accordingly, for all three problems, APC-MOPSO performed better than the other selected algorithms.

Table 4.36: Ranking of the Algorithms: Three-Objectives Test Problems (100 Particles)

Test Problem	2LBMOPSO	APC-MOPSO	MOPSO	NSGAII	SPEA2	PEGA
DTLZ2	3	**1**	4	2	Fail	5
DTLZ5	2	**1**	Fail	Fail	Fail	Fail
DTLZ7	2	**1**	Fail	Fail	3	Fail

For the five-objective experiments, the performance ranking is provided in Table 4.37. This ranking considered the ability of the APC-MOPSO, the 2LB-MOPSO, the NSGA-II, the MOPSO, the SPEA2, and the PEGA algorithms for solving the DTLZ2 and the DTLZ5 problems. This ranking is determined based on the values returned by summing of all objectives of each solution. It can be seen that the APCMOPSO clearly outperformed the others. It is also important to note that the MOPSO well performed in solving DTLZ2 compared to its performance in solving 2- and 3-objective benchmark test problems. In addition, it is worth noting that although 2LB-MOPSO showed competitive results in two- and three-objective test problems, but it failed in producing solutions that are close to the Pareto-optimal front in both five-objective problems.

Table 4.37: Ranking of the Algorithms: Five-Objectives Test Problems (100 Particles)

Test Problem	2LBMOPSO	APC-MOPSO	MOPSO	NSGAII	SPEA2	PEGA
DTLZ2	Fail	**1**	2	3	Fail	Fail
DTLZ5	Fail	**1**	Fail	Fail	Fail	Fail

4.6 Conclusion

PSO algorithms are increasingly being used to solve NP-hard and complex real-world applications that involve multi-objective mathematical formulation. PSO algorithms provide a powerful search capability in finding accurate and robust solutions. Furthermore, various PSO algorithms have performed well in many applications as described in [237,75,101,80,232,261,242,262,263,27]. To improve the overall performance of the MOPSO, an Adaptive Parallel Clustering-based MOPSO (APC-MOPSO) algorithm has been proposed. This algorithm introduced a new adaptive technique that automatically (i.e. dynamically, linearly, and exponentially) adjusted the settings of the algorithm's parameters, such as inertia weight (w), as

well as its cognitive (C_1), social (C_2), contiguous (C_3), position and velocity coefficients (C_p and C_v, respectively), and mutation operator (p_m). Furthermore, the proposed APC-MOPSO employed a modified K-Means^{++} clustering technique for managing the number of the non-dominated solutions in its external repository. Lastly, the APC-MOPSO is implemented in a parallel platform so that its diversity, convergence, and computational time are enhanced.

As a conclusion, the proposed APC-MOPSO algorithm has been demonstrated, through an extensive experimental analysis, a strong potential to efficiently solve multi-objective real world applications such as energy optimization in water distribution systems as shown next in Chapter 6.

Chapter 5

Auxiliary Techniques for Energy Optimization Strategy

This chapter begins with a brief overview of three new mechanisms that are searching-for-gaps, operating-mode pointer, and selecting-best-operating point. Each is explained in relation to the proposed energy optimization strategy. Searching-for-gaps algorithm is formally defined, and some experiments are conducted using the benchmark test problems previously explained in Chapter 4. Next, a novel operating-mode pointer is introduced. Some key concepts and notations on encoding and capturing pumping operations and settings are presented. Operating-modes are settings that have been tested and successfully implemented in pumping stations of water distribution systems. Common approaches for controlling variable-speed pumps are reviewed, and a strategy for selecting-best-operating point is described. Finally, the above three mechanisms are analyzed, tested, and assessed to be integrated to the new energy optimization strategy for optimizing pumping operations in rural water distribution systems.

5.1 Introduction

Water utilities are amongst the biggest energy consumers and use one fifth of the world's total power. One of the largest operating costs for water utilities is for electrical energy for pumping. This cost can be reduced by developing energy optimization strategies that would overcome the problem of pumps operate far from their most efficient operating point.

To successfully implement a new optimization technique on pumping stations for energy saving, novel techniques that are searching-for-gaps, operating-mode pointer, and selecting-best-operating point are proposed in this dissertation. The main reason of developing the three techniques and then combining them to the proposed Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization algorithm (APC-MOPSO) is to have a new energy optimization strategy that can effectively deal with real-world applications and efficiently produce approximately optimal solutions. The aims of the new energy optimization strategy are

not only significantly reducing the electric bill, but also improving the pump life cycle cost, the reliability of the pump operation, and the water quality in a distribution system. In this context, many research efforts on energy management in water distribution systems have been reported, but very few of them have in practice resulted in reducing the energy cost and promoting other good operating conditions [264,265,266,267,268,269,270,271,272].

5.2 Searching-for-Gaps

Although many researchers have reported the use of archiving strategies in multi-objective optimization [273,274], not many have proposed new techniques to archive non-dominated solutions of a multi-objective optimizer. What follows, is a review of literature in this specialized domain.

Hiroyasu et al. [275], proposed the Divided Range Multi-Objective Genetic Algorithm (DRMOGA). In this algorithm, the whole population is sorted according to an objective function with randomly selected elements that are changed after a number of iterations. The population is divided into sub-populations as shown in Figure 5.1. After a certain number of iterations, the non-dominated solutions produced in each sub-population are gathered. The process is then repeated with another randomly rearranged objective function. This approach is mainly used to improve the search ability and diversity of MOGA.

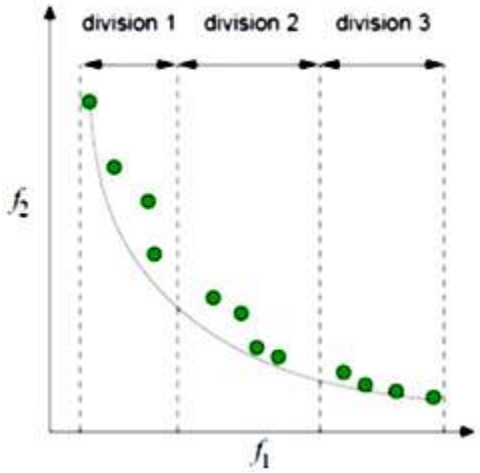


Figure 5.1: Division of Population in Divided Range MOGA [276]

J.D. Knowles [277], proposed a new Adaptive Grid Algorithm (AGA) to address some of the problems associated storing and selecting the non-dominated solutions obtained during the search process for multi-objective optimization problems. The AGA is developed based on the archiving method used in Pareto Archived Evolutionary Strategy (PAES) (J.D. Knowles et al. [76]). Figure 5.2 illustrates the mechanism of AGA, in which an adaptive grid in the hyper-

dimensional objective space changes its location and size as the solutions in the repository change with iterations $(t_i < t_j < t_k)$. In this mechanism, the grid adjusts its boundaries to envelop all the solutions. Furthermore, the grid is also used for selecting which solutions to eliminate from the archive when it reaches its capacity limit. The principle of crowding distance is also adopted in the archiving process, where only a single solution from the most crowded region(s) is selected to be stored. The author concluded that proposing AGA would allow the multi-objective optimizers to not only have solutions that cover the entire front, but also are evenly distributed.

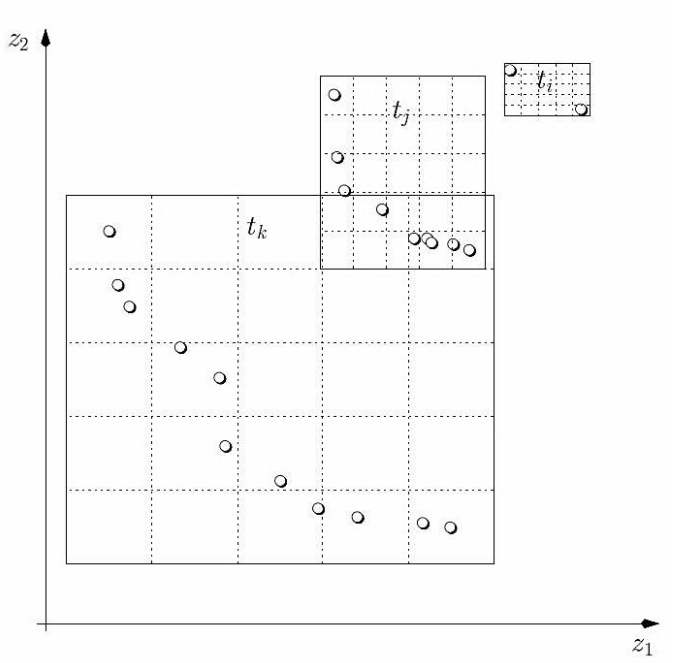


Figure 5.2: The Mechanism of Changing the Shape in Adaptive Grid Algorithm [277]

S. Fan et al. [97], developed a new picking rule for the external repository to obtain an evenly-distributed Pareto-front. In their work, a new technique, known as a picking rule, to select the non-dominated set of solutions in the external repository is used to maintain a 'constant' number of non-dominated solutions at every iteration. This technique is designed to be executed only when the number of currently non-dominated solutions exceeded the size of the external repository. Figure 5.3 depicts the mechanism of applying a picking rule on the external repository. For example, assuming that 13 non-dominated solutions are produced at *t*-iteration, and the external repository is restricted to a maximum size of 10 solutions. The first step is to select the solutions at the extremities; solutions S_1 and S_{13} shown in Figure 5.3 (a). Next, the distance from S_1 to S_{13}, is approximated as the sum of piecewise line segments between two adjacent Pareto solutions. Then, 10 temporary index points (i.e., $I_{1,2,\ldots,10}$) are equidistantly assigned along the Pareto-front (see Figure 5.3 (b)). The last step of this technique is to pick a non-dominated solution that is closest to the index points (see Figure 5.3 (c)). The objective of the

picking rule is to enhance the diversity of the solutions in the external repository. However, inspection of the plots provided in ZDT4 test problem show that the picking rule experienced problems producing solutions over the full length of the front. Thus, this technique cannot guarantee a fair basis of selecting solutions that preserves the Pareto-front.

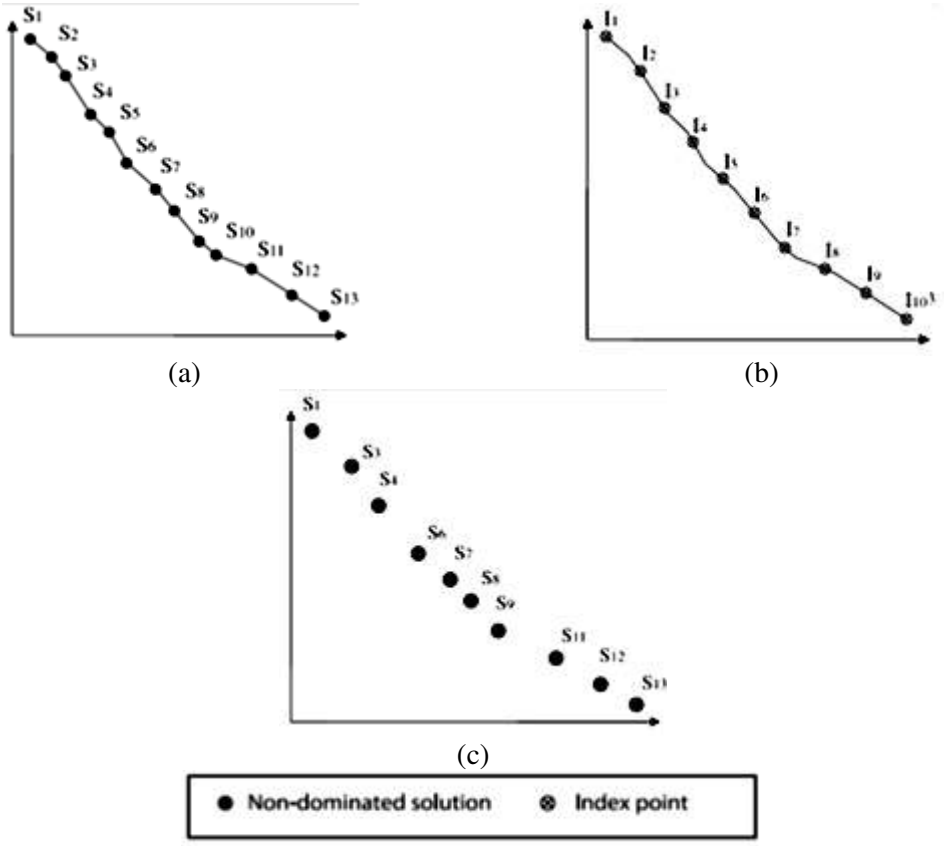


Figure 5.3: Graphical Representation of the Pick Rule for External Archive Scheme [97]

C.A.C. Coello and M.S. Lechunga [29], proposed two fixed-size repositories with a modified version of MOPSO. The first repository is used to archive the global best particles found so far by the search process; while the second stores the local best for each particle of the swarm. The authors used the truncate method from [76] to archive the global particles. In this method, the objective space is divided into a number of hyper-cubes (an adaptive grid), and the densely populated hyper-cubes are truncated if the repository exceeded its limit. A fitness value is assigned to each hypercube that contained archived members, equal to dividing 10 by the number of resident particles. Hence, a lower score will be assigned to a densely populated hypercube and vice versa. An illustration of this mechanism is given in Figure 5.4. For example, hypercube (4, 1) has only one solution, and thus, assigned a fitness value of 10.0, while hypercube (1, 4) contains 5 solutions, and therefore, a fitness value of 2.0 is given to it. Accordingly, the global best for the swarm is selected based on a roulette wheel of a hypercube first, and then randomly

choosing a member of that hypercube. Unlike the original method proposed in [76], the adopted archive method in this work biases selection toward under-represented areas of the Pareto-front. For the two-objective test problems, the proposed MOPSO showed better results than those obtained by Pareto Archived Evolutionary Strategy (PAES), [76]. Again, for higher number of objectives, this method failed to produce a significantly better solution when compared to competing algorithms.

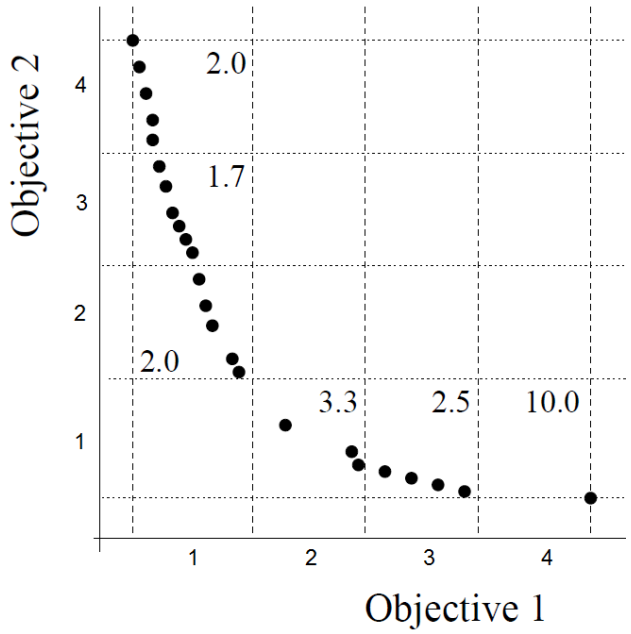


Figure 5.4: Graphical Representation of Grid Based Selection Scheme [29]

O. Schutze, et al. [278], proposed a technique called "gaps-free" or "tight Pareto-front" for multi-objective stochastic search algorithms to obtain a finite size of approximately optimal solutions for continuous multi-objective optimization problems. With the concept of ε-dominance [273], a gaps-free scheme is proposed to obtain a finite non-dominated set of solutions that cover the entire front. Furthermore, to determine the bounds of the repository size, two memetic strategies[21] are included into the search process. One uses the ε-dominance to store only the solutions that are non-dominated by any other members in the repository. This strategy allows limiting the number of entries of the repository and then guaranteeing the convergence of the archived members to the Pareto set. The second repository is used to obtain a uniform distribution and a better approximation quality by changing the criterion of accepting candidate solutions (i.e.,

[21] Combining global and local search is a *strategy* used by a hybrid optimization approach, namely Memetics Algorithm (MA). MA is an Evolutionary Algorithms (EA) that applies some sort of a local search to further improve the fitness of individuals in the population-based approach. "Memetics Algorithm" is first introduced by Richard Dawkins in 1976, [299].

used in the first repository) and replacing it by ε-(approximate) Pareto and uniformity level strategies. The numerical results summarized the behavior of the two archiving strategies against selected benchmark functions. Finally, it should be noted that the proposed strategy did not guarantee a gap-free approximation of the Pareto-front. More selection pressure in the first archiver prevents the members of the repositories from converging toward the global front, while for the second respository, it allows for such a convergent behavior, but at the expense of lossing its uniformity level.

5.2.1 The Proposed Searching-for-Gaps Technique (SFG)

In most MOPSOs, particles draw their flight directions based on two leaders that are particle best and swarm best positions, p_{best} and g_{best}, respectively. In reality, these two positions could misguide the particles since they are far apart from each other in the parameter space (i.e., decision space). Therefore, in the proposed Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) (explained in Chapter 4), the flight directions of the particles in the search space is guided by three leaders that are the particle best, the swarm best, and the local best (p_{best}, g_{best}, l_{best}, respectively). Hence, APC-MOPSO is said to have an effective guide to accelerate the particles in multiple directions when solving hard problems (see the results section of Chapter 4).

For the proposed APC-MOPSO, a new Searching-for-Gaps (SFG) technique is developed to guarantee: (i) storing and maintaining solutions at the extremities of the front, (ii) storing and maintaining solutions in all the critical Pareto-occupied regions, (iii) preserving diversity by converging towards global fronts, and (iv) distributing uniformly and evenly the solutions along Pareto-optimal front. In this variant, a second external repository, an adaptive grade algorithm, and leader selection technique are adopted as follows.

The proposed APC-MOPSO with SFG used two external repositories: one, referred to as the primary, for archiving the non-dominated solutions produced throughout the iterative search, and another, referred to as auxiliary or secondary repository, which is used for archiving neighbor best positions (l_{best}) that APC-MOPSO found along the search process.

For the SFG to cover the entire front as well as the extremes for each objective, an Adaptive Grid Archiving (AGA) algorithm (proposed by J.D. Knowles [277]) is adopted. In this variant, adding the AGA to the divided d-dimensional search space has an advantage over the crowding methods used in some MOEAs and MOPSOs. The AGA is used to remove the solutions from the auxiliary repository when it reached its maximum capacity bound. In this algorithm, there are two boundaries to the adaptive grid (i.e., upper and lower bounds, ub_{kt} and lb_k, respectively) for all objectives in the auxiliary repository. The boundaries are updated to enclose all the solutions found at the extremes of the search space.

The AGA works by calculating the range for the current solutions stored in the auxiliary repository, and accordingly, adjusting the grid boundaries to cover the entire range. It should be noted that in order to avoid recalculating the ranges too frequently, a threshold parameter is introduced so that the grid boundaries are only recalculated when the change in the range of the archived solutions is bigger than the threshold. The number of grid regions (known as bins), which sets the number of the subdivisions (i.e., hyper-cubes) of the search space are in each dimension is chosen priori by the user. This number, referred to as *div* $(div \geq 2)$, remains constant over time and is independent on the number of the non-dominated solutions in the auxiliary repository at each iteration. In other words, the space occupied by the stored solutions is frequently updated according to the locations of the solutions in the search space. It should be noted that the modified K-Means^{++} explained in Chapter 4, is also used in the auxiliary repository to truncate the extra solutions.

To illustrate the mechanism of the adaptive grid archiving, assume the following: when a new solution *z* extends the range of the grid in an objective *k* (i.e., the change in the range is larger than the threshold value along that objective), a new $range_k$ is calculated, and the grid boundaries are updated as shown in Eqs. (5.2) and (5.3).

$$range_k \leftarrow max_{z \in N}(z_k) - min_{z \in N}(z_k) \quad 5.1$$

$$ub_k \leftarrow max_{z \in N}(z_k) + \big(1/(2.div)\big)(range_k) \quad 5.2$$

$$lb_k \leftarrow min_{z \in N}(z_k) - \big(1/(2.div)\big)(range_k) \quad 5.3$$

where *N* is the set of non-dominated solutions archived in the auxiliary repository.

The two external vectors $Ext\ lb_k$ and $Ext\ ub_k$ in objective *k* will be located at the center of the outer grid regions as depicted in Figure 5.5.

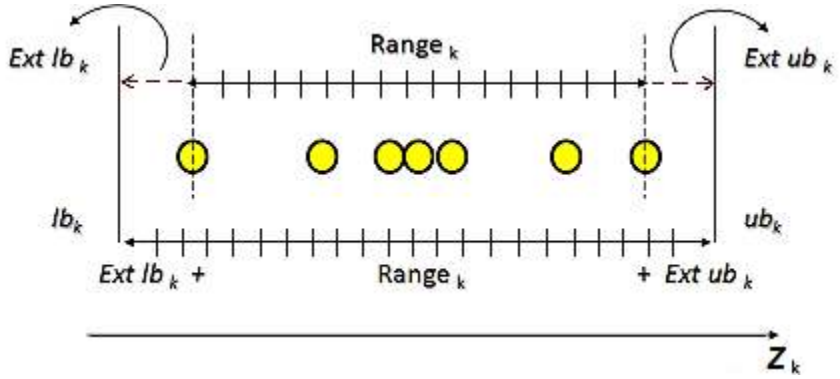


Figure 5.5: Example of Adaptive Grid Boundaries Technique in the Auxiliary Repository

At each iteration and after updating the particles' velocity and position, the range of each objective function is divided into a number of bins (by using AGA). The l_{best} is then selected from the archived members located in the neighboring bin. The l_{best} selection technique used in the auxiliary repository is adopted from the Two Local Best Multi-objective Particle Swarm Optimization (2LB-MOPSO), [243]. In order to assign l_{best} for a particle, an objective and a non-

empty bin is randomly selected. Within the bin, the archived member is selected if it has the lowest objective value and is amongst those with the highest crowding distance. As each particle is guided by a third leader (l_{best}), the archiving technique allowed the particles to update their velocity in the direction between the swarm best (g_{best}), their personal best (p_{best}), and their local best (p_{best}). Hence, the archiving strategy adopted for the auxiliary repository is oriented to improve the convergence and distribution of the resulting Pareto-front as shown in Figure 5.6.

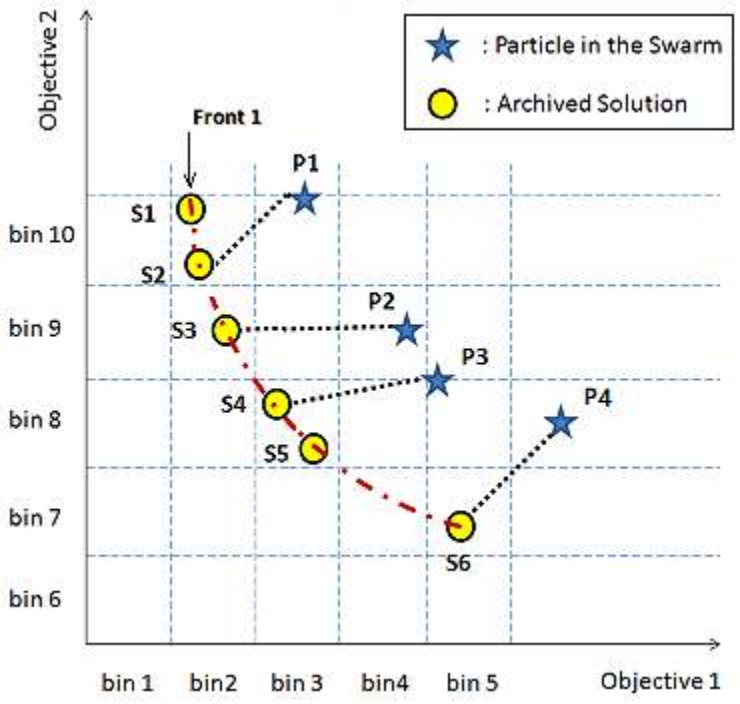


Figure 5.6: Graphical Representation of Selecting l_{best} from the Auxiliary Repository

In Figure 5.6, a two-objective minimization example is shown. Each objective range in the auxiliary repository is equally divided to create bins. For example, the auxiliary repository has 6 archived solutions (**S1** to **S6**), each of them is a non-dominated front 1 solution. The four particles (**P1** to **P4**) fly in directions guided by their corresponding p_{best}, g_{best}, and l_{best}. For particle **P1**, it is best to randomly select one objective and bin for its l_{best}. Here, objective 2 and bin 10 are assumed for **P1**. Between the two archived solutions **S1** and **S2** in bin 10, the **S2** is chosen as the l_{best} since it has the larger crowding distance. Assuming the local best (l_{best}) of **P3** is assigned to objective 1 and bin 3 in the current iteration. **S4** is selected because it has crowding distance larger than **S5**. Following the same rules, **S3** and **S6** can be chosen as the local best (l_{best}) for **P2** and **P4**, respectively.

Finally, Searching-for-Gaps (SFG) with an auxiliary repository, an Adaptive Grid Archiving (AGA), and a selection leader technique ensure obtaining Pareto-optimal solutions that cover a wide extent in the search space and are "well-distributed".

Next, the dissuasion of the results obtained by implementing the APC-MOPSO with SFG on the benchmark test problems (previously explained in Chapter 4) is provided.

5.2.2 Results for Two-Objective Test Problems

To assess the performance of the proposed APC-MOPSO with SFG, a set of two-objective and three-objective benchmark problems (previously described in Chapter 4) are used. Twenty-five independent runs are performed for each test problem, and the results are compared with respect to the proposed APC-MOPSO without SFG. The total number of function evaluations is set to 10,000 in all tests, and the results obtained by the proposed APC-MOPSO with SFG are displayed in the four metrics: two-set Coverage (C), Error Ratio (ER), Generational Distance (GD), and Spacing (S). The best, worst, mean, median, and standard deviations of the metrics are reported for each test problem. The results are presented as follows:

- **ZDT1 Benchmark Test Problem**

The first test compared the APC-MOPSO with SFG with APC-MOPSO without SFG for the ZDT1 test problem. This test is performed using the same parameters used in the standard tests performed in Chapter 4. Table 5.1 tabulates the experimental results, and Figure 5.7 a and b illustrate the Pareto-front obtained by the APC-MOPSO with and without SFG versus the Pareto-optimal front of the problem. It is important to note that adopting SFG with APC-MOPSO significantly improved the distribution of the Pareto-front set as shown in Figure 5.7 c. In this figure, the solutions are evenly and uniformly distributed along the Pareto-optimal front.

- **ZDT2 Benchmark Test Problem**

This test is performed to illustrate the improvement in the performance of the proposed APC-MOPSO when SFG is used for the ZDT2 benchmark problem. In this test, the APC-MOPSO returned average values of 0.04, 0.02, 0.000467, and 0.00665 for the C, ER, GD, and S metrics that are less than those produced in Chapter 4. Furthermore, Figure 5.8 showed that APC-MOPSO with SFG produced a Pareto-front set that is better in filling the entire front as well as precisely converging to the Pareto-optimal front.

- **ZDT3 Benchmark Test Problem**

Figure 5.9 showed the Pareto-front produced by APC-MOPSO with and without SFG versus the Pareto-optimal front for the ZDT3 test problem. Table 5.1 presented the results of C, ER, GD, and S averaged over 25 independent runs. In this table, a significant improvement in performance is observed by considering the average values of C and S metrics, in which APC-MOPSO with SFG returned 0.035 and 0.00765, while APC-MOPSO without SFG returned 0.044 and 0.00964, respectively.

- **ZDT4 Benchmark Test Problem**

Although ZDT4 is considered a hard test problem, the proposed APC-MOPSO with SFG performed well and repeatedly converged to the Pareto-optimal front. In this test, a significant reduction in the average values of metrics C and S are reported in Table 5.1. Figure 5.10 depicts the Pareto-front obtained by APC-MOPSO with and without SFG technique.

- **ZDT6 Benchmark Test Problem**

The uniformly distributed solutions obtained by APC-MOPSO with SFG are better than those produced by APC-MOPSO without SFG. In this test, a full set of optimal Pareto-front (i.e., tight Pareto-front) is produced, in which the algorithm returned average values of 0.02, 0.0204, 0.003207, and 0.00563 for the four metrics of C, ER, GD, and S, respectively. Table 5.1 shows that the improvements of ER and S metrics are superior in comparison to the corresponding values of 0.07667 and 0.00732 for ER and S by APC-MOPSO without SFG. Figure 5.11 shows that vectors produced by both APC-MOPSO with/without SFG are clearly different.

- **DTLZ1 to DTLZ7 Benchmark Test Problems**

Table 5.1 presents the results of C, ER, GD, and S obtained by using the proposed APC-MOPSO with SFG. In all test problems, APC-MOPSO with SFG returned smaller average and standard deviations for C, ER, GD, and S values than those obtained by APC-MOPSO without SFG. For the proposed APC-MOPSO with SFG, the mean values of the four metrics are significantly reduced. The consistency of producing high quality vectors for DTLZ1 to DTLZ7 test problems are shown in Figures 5.12 and 5.18. It can also be seen from these figures that APC-MOPSO with SFG is not only converged to the Pareto-optimal front but also remarkably filled all gaps present in the solutions.

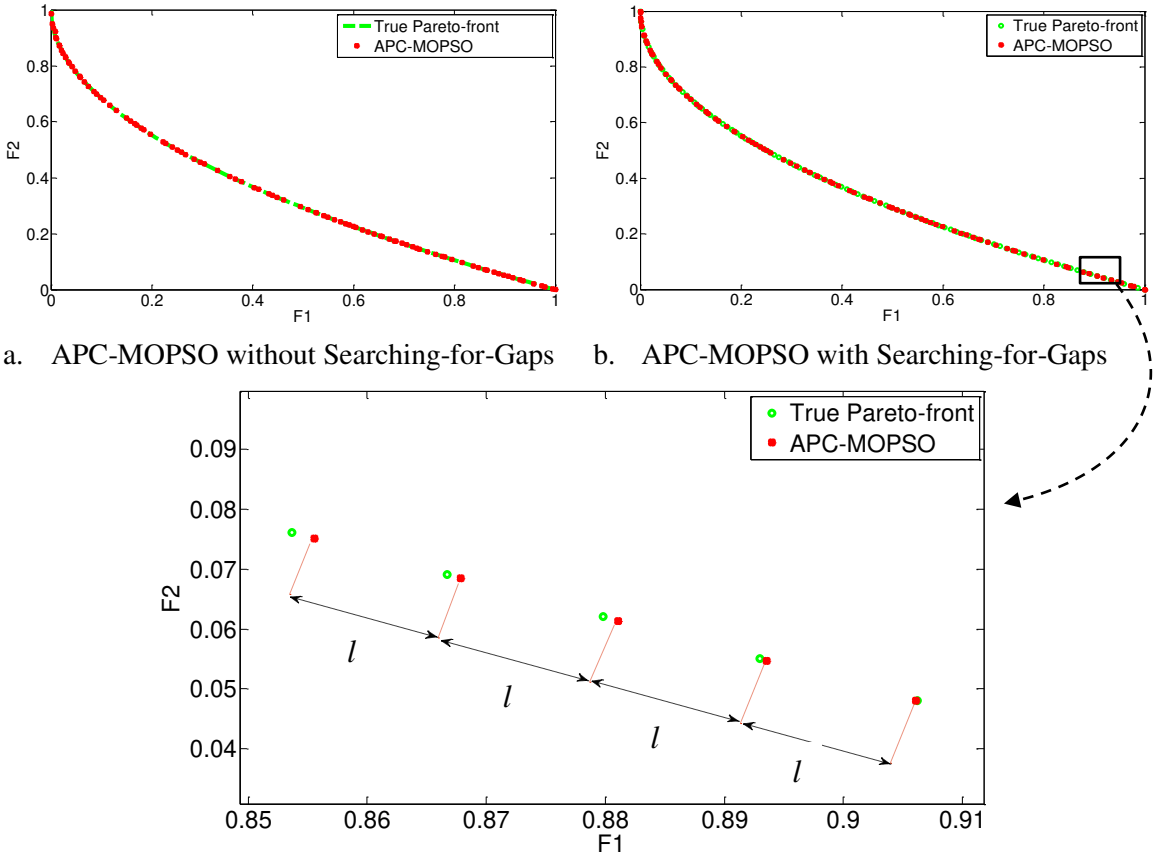
- **OKA2 Benchmark Test Problem**

Figure 5.19 shows the graphical representation of the results obtained by the proposed APC-MOPSO with SFG on OKA2 test problem. In this figure, it can be seen that the produced Pareto-front is more accurate in converging to the Pareto-optimal front and well spread and distributed along the Pareto-optimal front of the problem. The results of APC-MOPSO with SFG are provided in Table 5.1.

- **KURSAWE Benchmark Test Problem**

In the last test for the two-objective benchmark test problems, the KURSAWE problem is chosen to illustrate the importance of the proposed APC-MOPSO with SFG. The results of APC-MOPSO with SFG indicated considerable improvement compared to the case without SFG in the average values of C, ER, GD, and S metrics. The average value of C is reduced from 0.134 to 0.075; the ER's is reduced from 0.124 to 0.115; the GD's is reduced from 0.0106 to 0.0095; and the S's is reduced from 0.07681 to 0.0631 (as provided in Table 5.1 and shown in Figure 5.21).

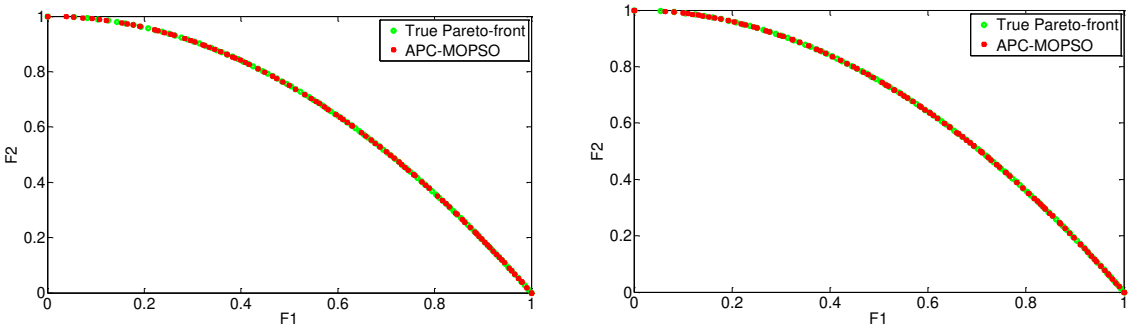
The obtained Pareto-fronts produced by the APC-MOPSO with and without SFG are illustrated in Figure 5.20. Furthermore, this figure shows that the solutions obtained by APC-MOPSO with SFG are diverse and well-distributed.



a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

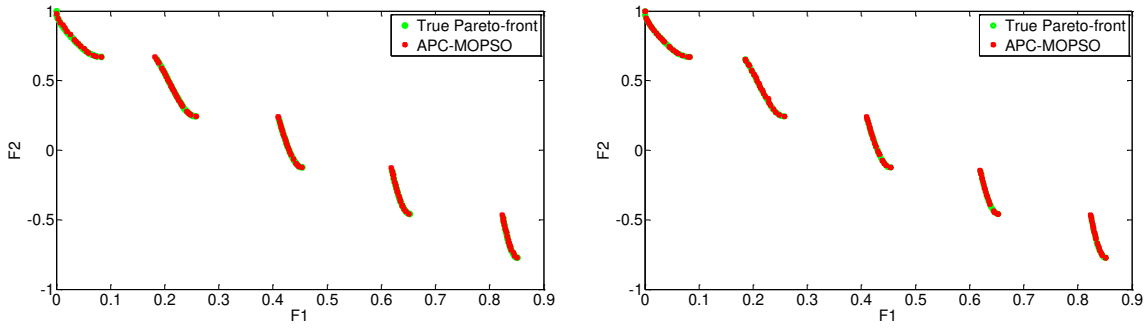
c. APC-MOPSO with Searching-for-Gaps - Zoom View

Figure 5.7: Pareto-front Produced by the APC-MOPSO for the ZDT1 Benchmark Problem



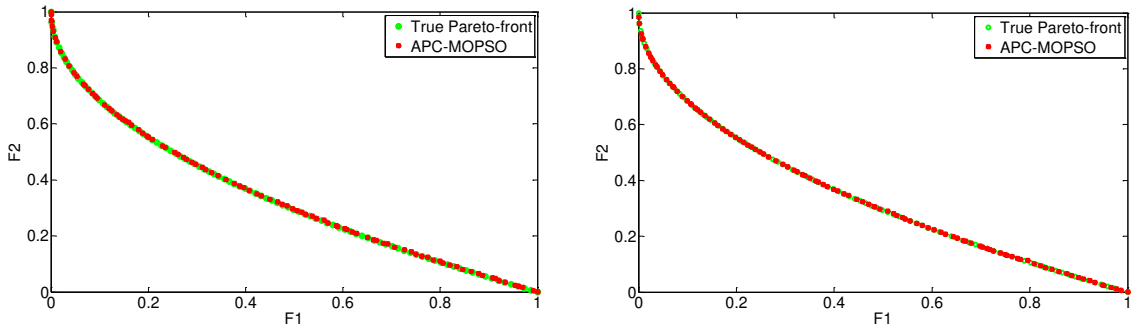
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.8: Pareto-front Produced by the APC-MOPSO for the ZDT2 Benchmark Problem



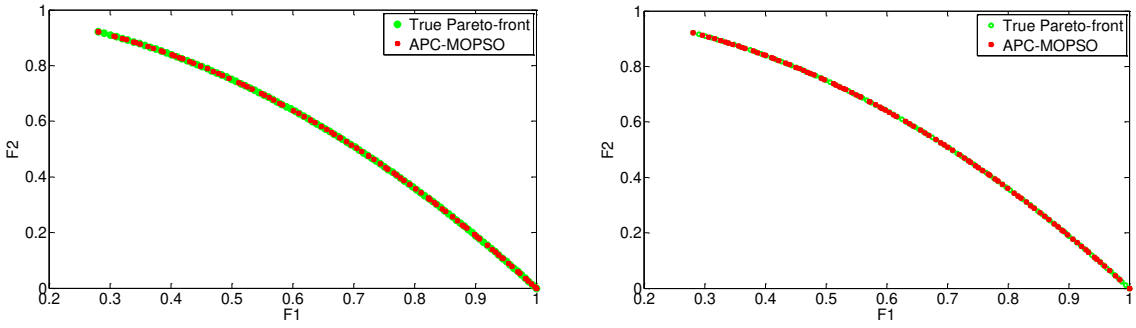
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.9: Pareto-front Produced by the APC-MOPSO for the ZDT3 Benchmark Problem



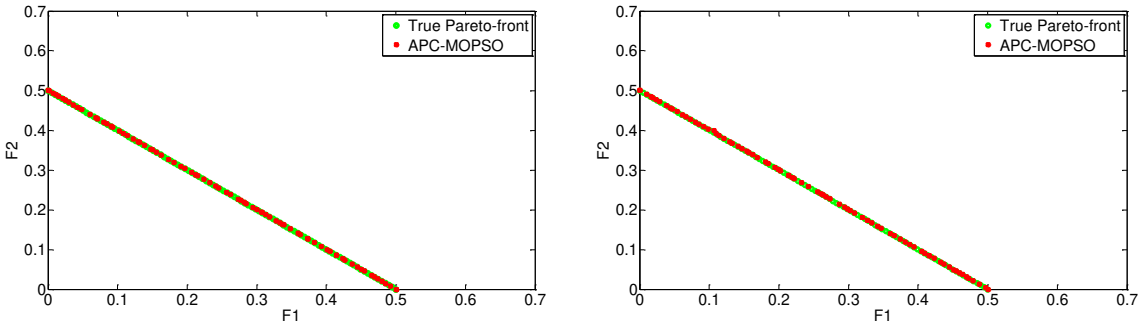
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.10: Pareto-front Produced by the APC-MOPSO for the ZDT4 Benchmark Problem



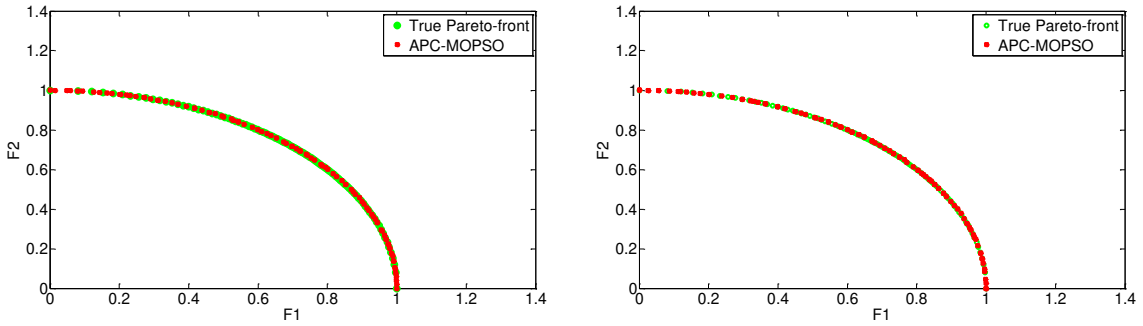
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.11: Pareto-front Produced by the APC-MOPSO for the ZDT6 Benchmark Problem



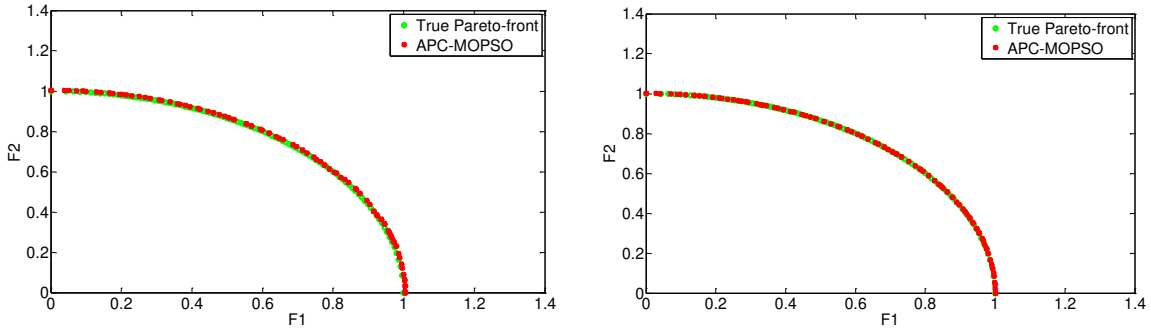
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.12: Pareto-front Produced by the APC-MOPSO for the DTLZ1 Benchmark Problem



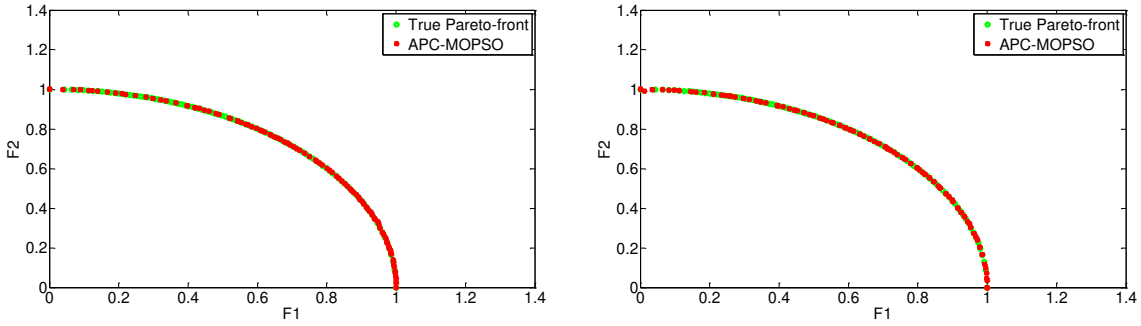
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.13: Pareto-front Produced by APC-MOPSO for the DTLZ2 Benchmark Problem (2-D)



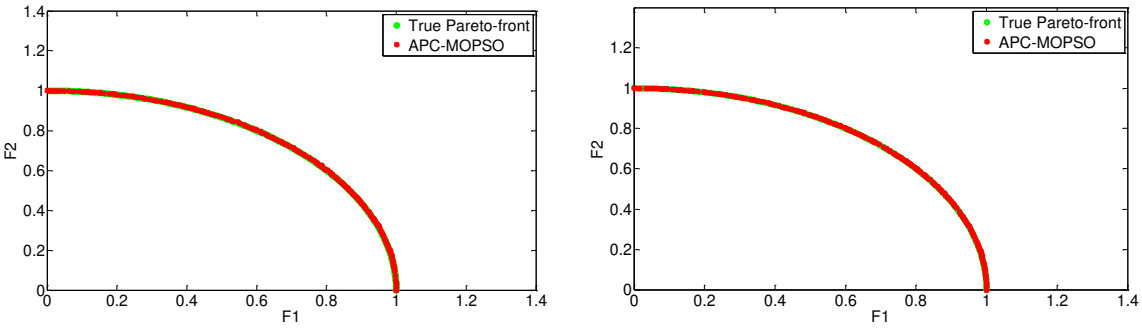
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.14: Pareto-front Produced by the APC-MOPSO for the DTLZ3 Problem



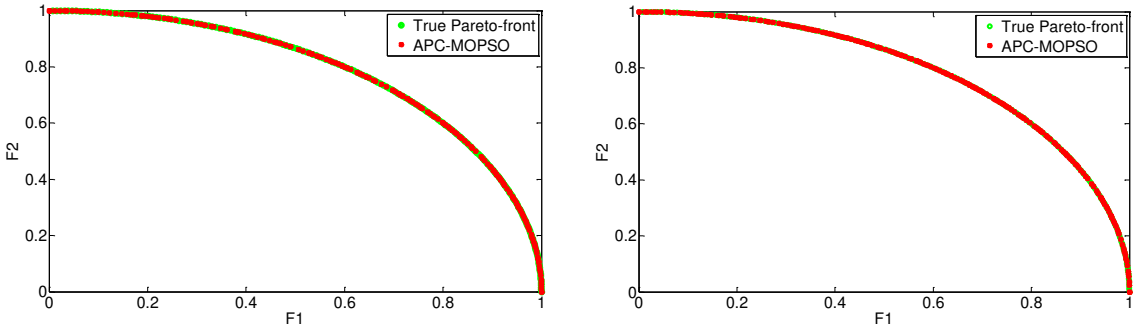
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.15: Pareto-front Produced by the APC-MOPSO for the DTLZ4 Benchmark Problem



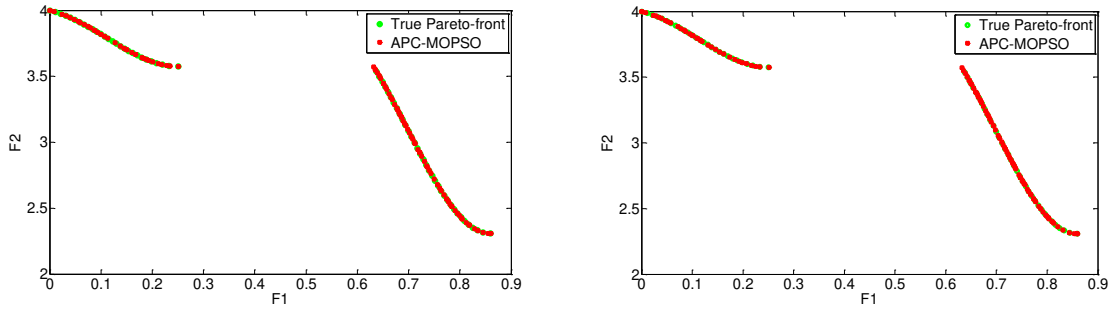
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.16: Pareto-front Produced by the APC-MOPSO for the DTLZ5 Benchmark Problem



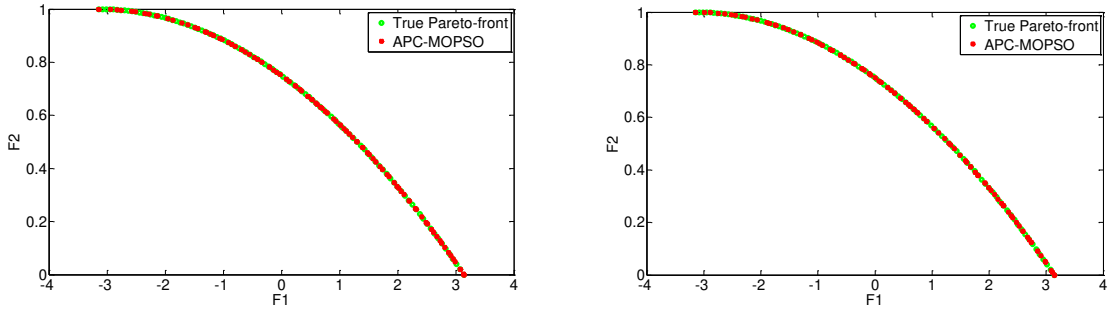
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.17: Pareto-front Produced by the APC-MOPSO for the DTLZ6 Benchmark Problem



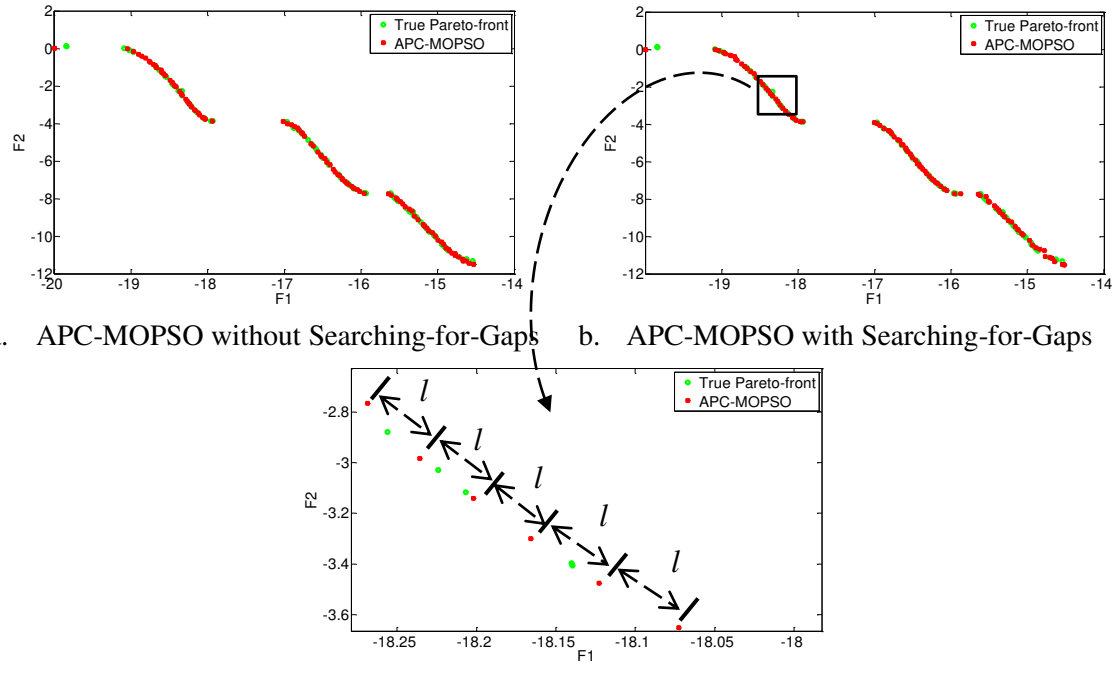
a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.18: Pareto-front Produced by the APC-MOPSO for the DTLZ7 Benchmark Problem



a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.19: Pareto-front Produced by the APC-MOPSO for the OKA2 Benchmark Problem



a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

c. APC-MOPSO with Searching-for-Gaps - Zoom View

Figure 5.20: Pareto-front Produced by APC-MOPSO for the KURSAWE Benchmark Problem

Table 5.1: The Results for APC-MOPSO with SFG on Benchmark Test Problems (2-D)

Problem	Statistics	C	ER	GD	S
ZDT1	Best	0.02000	0.00000	0.000382	0.00460
	Worst	0.02000	0.01000	0.000437	0.00710
	Mean	0.02000	0.00800	0.000406	0.00572
	Median	0.02000	0.01000	0.000411	0.00500
	Std	0.00000	0.00447	0.000024	0.00122
ZDT2	Best	0.04000	0.01000	0.000418	0.00610
	Worst	0.04000	0.02000	0.000505	0.00680
	Mean	0.04000	0.01500	0.000467	0.00652
	Median	0.04000	0.01500	0.000474	0.00665
	Std	0.00000	0.00548	0.000031	0.00029
ZDT3	Best	0.02000	0.07000	0.006027	0.00710
	Worst	0.06000	0.10000	0.006475	0.00830
	Mean	0.03500	0.09167	0.006175	0.00765
	Median	0.03500	0.10000	0.006135	0.00755
	Std	0.01517	0.01329	0.000178	0.00055
ZDT4	Best	0.05000	0.01000	0.004440	0.00600
	Worst	0.13000	0.01000	0.004710	0.00630
	Mean	0.08500	0.01000	0.004567	0.00613
	Median	0.08000	0.01000	0.004560	0.00610
	Std	0.03416	0.00000	0.000118	0.00015
ZDT6	Best	0.02000	0.01000	0.003155	0.00550
	Worst	0.02000	0.04120	0.003280	0.00590
	Mean	0.02000	0.02040	0.003207	0.00563
	Median	0.02000	0.01000	0.003187	0.00550
	Std	0.00000	0.01801	0.000065	0.00023
DTLZ1	Best	0.07000	0.01000	0.000180	0.00380
	Worst	0.15000	0.07000	0.000234	0.00460
	Mean	0.10800	0.03400	0.000208	0.00437
	Median	0.10000	0.02000	0.000208	0.00451
	Std	0.03564	0.02510	0.000021	0.00030
DTLZ2	Best	0.02000	0.01000	0.000829	0.00711
	Worst	0.02000	0.06000	0.000854	0.00780
	Mean	0.02000	0.03500	0.000842	0.00746
	Median	0.02000	0.03500	0.000842	0.00746
	Std	0.00000	0.03536	0.000018	0.00049
DTLZ3	Best	0.28000	0.01000	0.000780	0.00850
	Worst	0.53000	0.04000	0.001020	0.00930
	Mean	0.42250	0.02250	0.000892	0.00895
	Median	0.44000	0.02000	0.000883	0.00900
	Std	0.11500	0.01258	0.000101	0.00033

Table 5.1: The Results for APC-MOPSO with SFG on Benchmark Test Problems (2-D) (continue)

Problem	Statistics	C	ER	GD	S
DTLZ4	Best	0.04000	0.02000	0.000600	0.00590
	Worst	0.06000	0.03000	0.000736	0.00630
	Mean	0.05000	0.02250	0.000639	0.00611
	Median	0.05000	0.02000	0.000611	0.00613
	Std	0.00816	0.00500	0.000065	0.00018
DTLZ5	Best	0.10000	0.00400	0.000158	0.00300
	Worst	0.10000	0.00500	0.000160	0.00320
	Mean	0.10000	0.00433	0.000159	0.00310
	Median	0.10000	0.00400	0.000159	0.00310
	Std	0.00000	0.00058	0.000001	0.00010
DTLZ6	Best	0.01000	0.01000	0.000152	0.00305
	Worst	0.01000	0.01000	0.000161	0.00360
	Mean	0.01000	0.01000	0.000157	0.00329
	Median	0.01000	0.01000	0.000158	0.00325
	Std	0.00000	0.00000	0.000004	0.00026
DTLZ7	Best	0.02000	0.03000	0.00082	0.00720
	Worst	0.02000	0.04000	0.00095	0.00820
	Mean	0.02000	0.03200	0.00088	0.00750
	Median	0.02000	0.03000	0.00087	0.00740
	Std	0.00000	0.00447	0.00006	0.00040
KURSAWE	Best	0.07000	0.11000	0.009300	0.05890
	Worst	0.08000	0.12000	0.009700	0.06730
	Mean	0.07500	0.11500	0.009500	0.06310
	Median	0.07500	0.11500	0.009500	0.06310
	Std	0.00707	0.00707	0.000283	0.00594
OKA2	Best	0.04000	0.01000	0.001700	0.02200
	Worst	0.04000	0.06000	0.001800	0.03000
	Mean	0.04000	0.04200	0.001740	0.02721
	Median	0.04000	0.04000	0.001700	0.02917
	Std	0.00000	0.02049	0.000055	0.00357

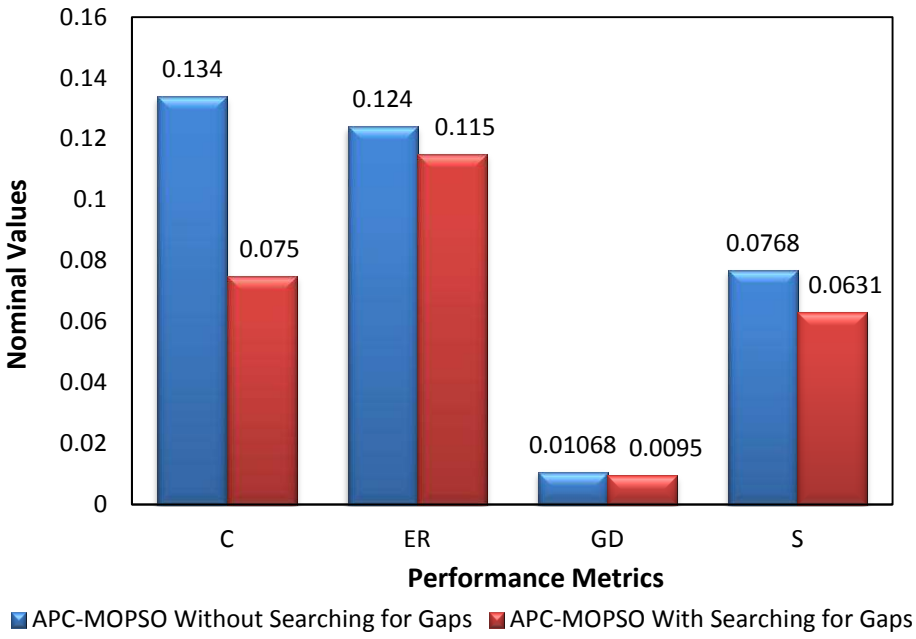


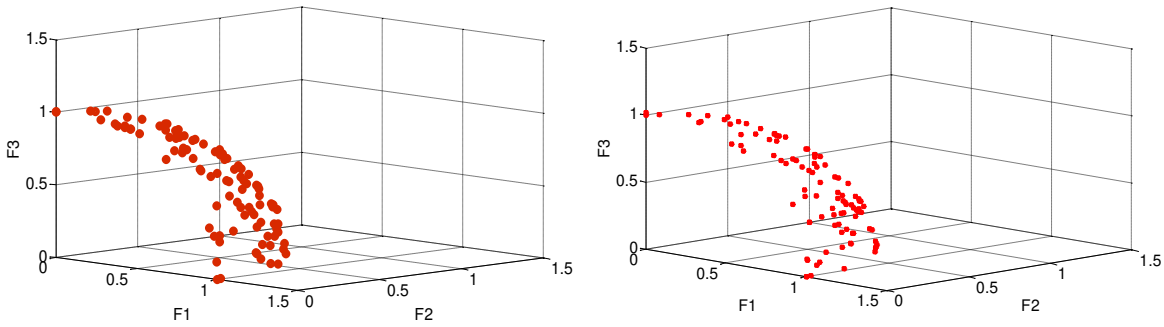
Figure 5.21: Performance Comparison between APC-MOPSO with and without SFG on KURSAWE Benchmark Problem

5.2.3 Results for Three-Objective Benchmark Test Problems

In the second set of tests, the proposed APC-MOPSO with SFG is compared with APC-MOPSO without SFG on the DTLZ2 and DTL5 benchmark test problems. The total number of function evaluations is set to 10,000 for the suite of DTLZ test problems. A swarm of 100 particles is used as the size of the population, and 25 independent runs are performed for each test problem. Note that in these problems the summation of all square objectives is equal to 1.0.

- **DTLZ2 Test Benchmark Problem**

Although both algorithms yield satisfactory results, the proposed APC-MOPSO with SFG converges closer to the Pareto- optimal front than the proposed APC-MOPSO without SFG. The results are provided in Table 5.2. Figure 5.22 shows that the proposed APC-MOPSO with SFG achieved a comparatively better solution spread and a better convergence to the Pareto-optimal front.



a. APC-MOPSO without Searching-for-Gaps b. APC-MOPSO with Searching-for-Gaps

Figure 5.22: Pareto-front Produced by APC-MOPSO on DTLZ2 Benchmark Problem (3-D)

- **DTLZ5 Benchmark Test Problem**

The proposed APC-MOPSO with SFG is implemented to solve the DTLZ5 test problem. It can be observed from the results shown in Table 5.2 that the proposed APC-MOPSO with SFG produced clearly better solutions with respect to the reference point of 1.0, as it returned a range of [1.0005 to 1.00951], while APC-MOPSO without SFG returned a range of [1.005, 1.0112]. Evidently, from this comparison, the proposed SFG is an improvement over the APC-MOPSO without SFG in most performance metrics.

Table 5.2: The Results for APC-MOPSO with SFG on Benchmark Problems (3-D) (100 Particles)

Problem	Statistics	Min	Max
DTLZ2	Best	1.00000	1.02070
	Worst	1.00020	1.03490
	Mean	1.00007	1.02620
	Median	1.00005	1.02535
	Std	0.00008	0.00496
DTLZ5	Best	1.00000	1.00850
	Worst	1.00010	1.01060
	Mean	1.00005	1.00951
	Median	1.00005	1.00950
	Std	0.00005	0.00069

5.2.4 Summary of Results for APC-MOPSO with Searching-for-Gaps

In the previous adaptive grid and auxiliary archiving methods, researchers reported that adopting these methods is limited by the shape of the Pareto-front, and thus, gaps-free or niching techniques (used in many MOEAs and MOPSOs) would not guarantee the convergence to the entire Pareto-optimal front [277,279,280,278,281]. For the two-objective experiments, Figures 5.7 to 5.22 show that the proposed APC-MOPSO with SFG demonstrates convergence and spread

of solutions regardless of the shape of the Pareto-optimal front. This includes problems that have multi-local Pareto-optimal fronts (ZDT4 and ZDT6), disconnected set of Pareto-optimal fronts (ZDT3, DTLZ7, and KURSAWE), convex Pareto-front (ZDT1 and OKA2), non-convex Pareto-front (ZDT2 and DTLZ5), and concave Pareto-front (DTLZ3, DTLZ4, and DTLZ6).

From Figures 5.23 and 5.24, the proposed APC-MOPSO with SFG is significantly better compared to APC-MOPSO without SFG. In particular, for the ER and C metrics, the use of SFG reduced these measures down by 55% and 33%, respectively. With respect to GD and S, again the average values are reduced by 10% and 15%, respectively.

The ranking comparison for the six algorithms as presented in Chapter 4 is updated and presented in Tables 5.3 and 5.4. By combining APC-MOPSO with SFG, a better spread of solutions and convergence is obtained compared to the other five algorithms: 2LB-MOPSO, NSGA-II, MOPSO, SPEA2, and PEGA.

For the three-objective experiments, and based on the computational results discussed in Section 5.2.3, it can be concluded that APC-MOPSO with SFG's results are more accurate, proximate, and well-spread for general multi-objective optimization problems. Table 5.5 and Figure 5.25 show that the proposed APC-MOPSO with SFG returned the best average values of performance metrics for both DTLZ2 and DTLZ5 benchmark problems, a fact that may be attributed to its excellent trade-off between exploration and exploitation abilities (i.e., balance the selection pressure).

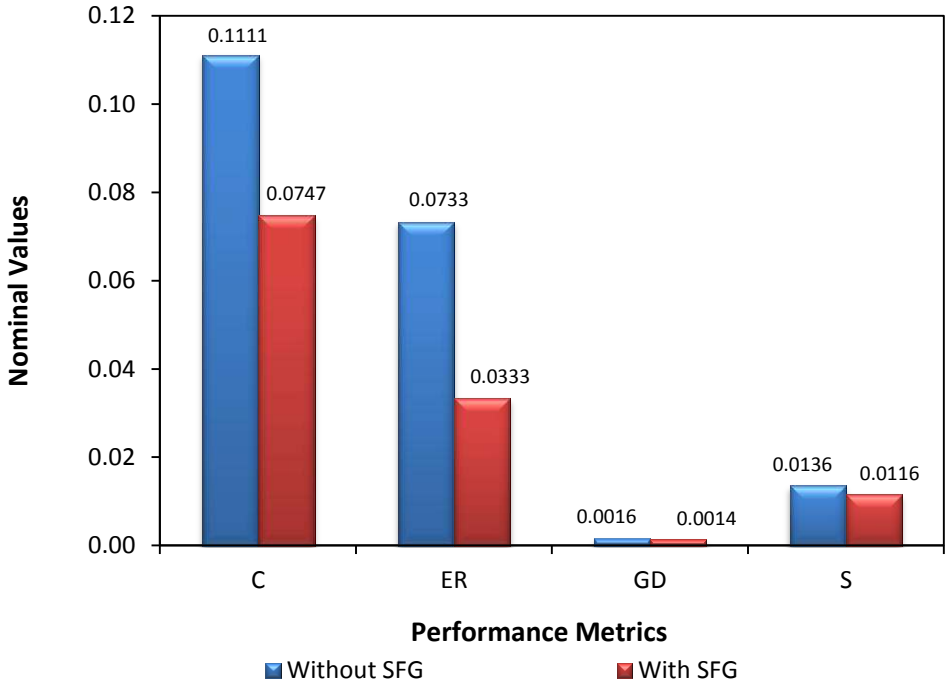


Figure 5.23: Overall Average Metrics for APC-MOPSO with and without Searching-for-Gaps on Two-Objective Benchmark Test Problems

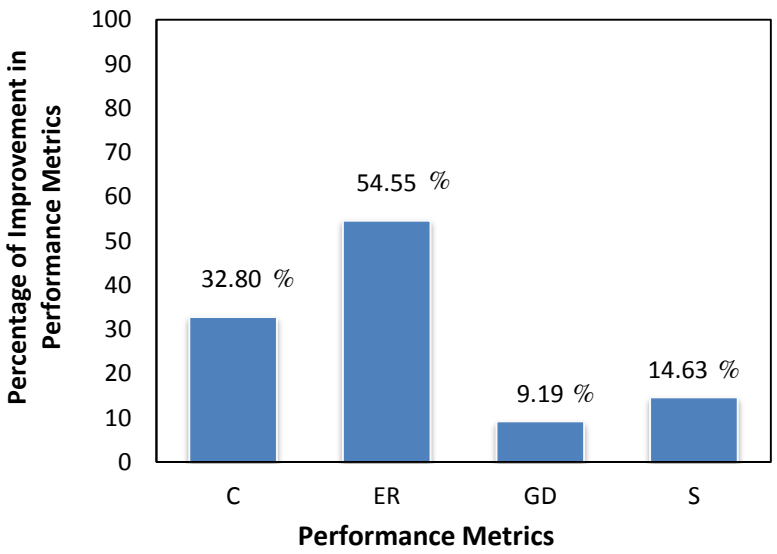


Figure 5.24: Overall Percentage Improvements in APC-MOPSO's Performance on Two-Objective Benchmark Problems with Searching-for-Gaps

Table 5.3: Update Ranking of the APC-MOPSO with SFG on Two-Objective Problems

Test Problems	Performance Metrics			
	C	ER	GD	S
ZDT1	1	1	1	1
ZDT2	1	1	1	**1 (2)***
ZDT3	1	1	1	**1 (2)**
ZDT4	1	1	1	1
ZDT6	1	1	1	**1 (2)**
DTLZ1	1	1	1	2
DTLZ2	**1 (2)**	2	2	2
DTLZ3	1	1	1	1
DTLZ4	1	1	1	1
DTLZ5	2	1	1	**1 (3)**
DTLZ6	1	1	**1 (2)**	1
DTLZ7	1	1	2	**1 (2)**
OKA2	1	1	1	1
KURSAWE	1	1	**1 (2)**	**1 (2)**

* (#) *means the rank of APC-MOPSO without searching for gaps technique with respect to five other algorithms against two-objective test problems*

Table 5.4: Average Performance Metrics Using APC-MOPSO with and without Searching-for-Gaps on Two-Objective Benchmark Test Problems

Problem	Performance Metrics							
	C		ER		GD		S	
	Without SFG	With SFG	Without SFG	With SFG	Without SFG	With SFG	Without SFG	With SFG
ZDT1	0.02600	0.02000	0.01667	0.00800	0.00045	0.000406	0.00627	0.00572
ZDT2	0.04000	0.04000	0.02600	0.01600	0.00049	0.000467	0.00964	0.00652
ZDT3	0.04400	0.03500	0.10267	0.09167	0.00066	0.000618	0.00905	0.00765
ZDT4	0.17400	0.08500	0.01200	0.01000	0.00047	0.000457	0.00680	0.00613
ZDT6	0.02000	0.02000	0.07667	0.02040	0.00340	0.003207	0.00732	0.00563
OKA2	0.04000	0.04000	0.06330	0.04200	0.00180	0.00174	0.02970	0.02721
KUR	0.13400	0.07500	0.12400	0.11500	0.01068	0.00950	0.07680	0.06310
DTLZ1	0.19000	0.10800	0.08714	0.03400	0.00023	0.000208	0.00533	0.00437
DTLZ2	0.13400	0.02000	0.06600	0.03500	0.00085	0.00084	0.00624	0.00746
DTLZ3	0.55333	0.42250	0.05000	0.02250	0.00108	0.000892	0.00986	0.00895
DTLZ4	0.06700	0.05000	0.04500	0.02550	0.00078	0.00639	0.00632	0.00611
DTLZ5	0.10350	0.10000	0.00500	0.00430	0.00016	0.000159	0.00322	0.00310
DTLZ6	0.01000	0.01000	0.01000	0.01000	0.00016	0.000157	0.00415	0.00329
DTLZ7	0.02000	0.02000	0.34170	0.03200	0.00099	0.00087	0.00993	0.00750
Average	0.11113	0.07467	0.07329	0.03331	0.001586	0.001851	0.013616	0.011624

Table 5.5: Comparison between APC-MOPSO with and without Searching-for-Gaps on Three-Objective Benchmark Test Problems (100 Particles)

Problem	Pareto-front Condition				
	Without SFG		With SFG		% Improvement
	Min	Max	Min	Max	
DTLZ2	1.00005	1.04650	1.00007	1.02620	43.66
DTLZ5	1.00050	1.01120	1.00005	1.00951	15.09
average	1.000275	1.02885	1.00006	1.017855	38.11

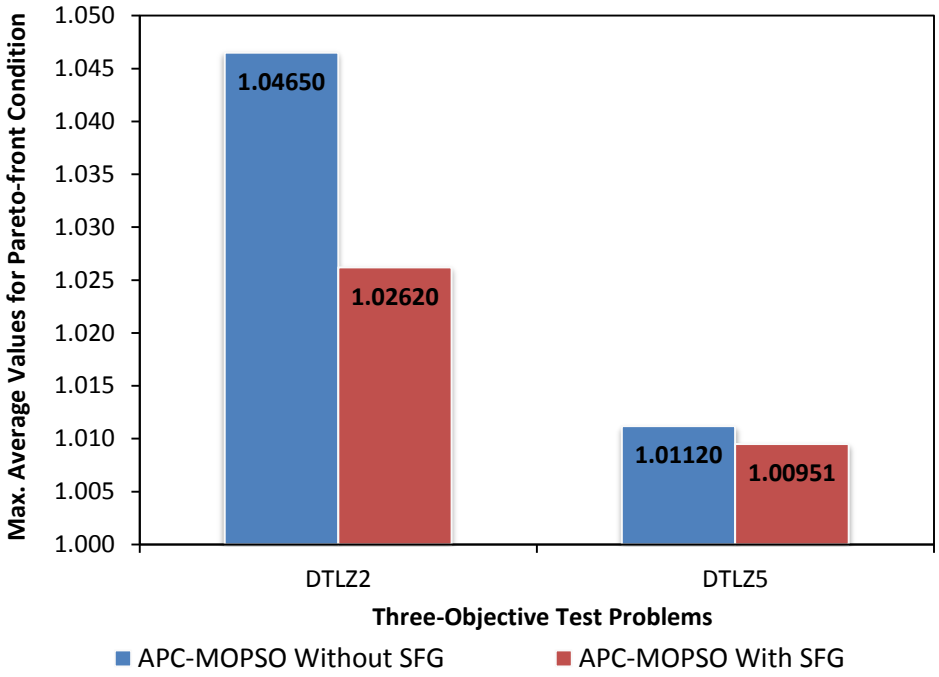


Figure 5.25: Maximum Average Values for APC-MOPSO with and without Searching-for-Gaps on Three-Objective Benchmark Test Problems

5.3 Operating-Mode Pointer

In the applications of multi-objective optimization, the choice of which encoding technique to use as a gene-code for capturing information about solutions is considerably important, [249]. Maximizing the correlation between the encoding technique and the problem landscape has many advantages such as speed of the convergence and resolution of the deceptive problem, [282].

There have been numerous encoding strategies that have been proposed [283,284,285], with very few and limited experimental implementations. None of the reported implementations address the safety, complexity (i.e., computational time), and reliability requirements of implementation by using the operating-mode concept as proposed here.

The Operating-Mode Pointer (OMP) is novel and can be categorized as an encoding technique. In essence, the method captures the existing operating scheme of the system as a set of "operating-modes" pertaining to safe and tested operating conditions. New operating modes are then gradually introduced, resulting in a slight departure from the existing normal operation. This novel technique can successfully be applied to energy optimization, design trade-off analysis, and other industry applications to allow a way of practically implementing a more efficient operation in a safe and controlled manner.

In this technique, the operating-modes are stored in a library, which is designed with the purpose of providing the optimization algorithm with initial candidate solutions. The optimization results pertain in a new (optimal) sequence of operating-modes that are originally derived from the library. This OMP technique can integrate to most optimization algorithms to develop an accurate knowledge of the problem and then can use that in generating new optimal solutions.

This section briefly discusses some encoding schemes that have been already used in Evolutionary Algorithms (EAs). Early work (1970s) on Genetic Algorithms (GA) encoded the solutions as strings (chromosomes) of bits from a binary alphabet. Traditional operations of GA such as selection, crossover, and mutation are specifically designed for only binary encodings. In the 1980s, with the development of GA and Evolutionary Algorithms (EA), new forms of encoding are introduced such as permutation encoding[22]. In permutation encoding, every chromosome is a string of integer numbers, [283]. Moreover, to better represent real-life optimization problems, a direct value encoding is used, where every chromosome is represented as a string of floating point number. In the 1990s, Genetic Programming (GP) is developed and employed tree encoding, where every gene represented as an object, such as a function, mathematical operation, or a command in a programming language, [283]. A review of the history of real coding schemes in relation to the multiple evolutionary paradigms is given by D. Goldberg [286].

In this dissertation, a novel gene-coding technique, referred to as the Operating-Mode Pointer (OMP) is proposed. The OMP is an updated form of tree encoding which refers to the use of a variable-length code table for encoding a source symbol. This means that it is possible to use OMP to capture, encode, and manipulate a consecutive chain of events as a discrete, a continuous, or a mixed form for real-world applications.

5.3.1 The Proposed Operating-Mode Pointer Technique (OMP)

The Operating-Mode Pointer (OMP) uses a different type of gene representation than other common encoding schemes such as binary alphabet, permutation, floating-point number, direct value, and tree. The OMP approach separates time into segments of varying length bordered by certain events. Such segments, referred to as the operating-modes, are denoted by indices at which the functions describing the system's behavior (i.e., the functions which specify the ongoing process). A high-level semantics for encoding about systems dynamic behavior is developed in this work. In doing this, the encoding technique is combined with numerical scheme offering qualitative and quantitative assessment of optimal solutions.

[22] In permutation encoding, every chromosome is encoded by a string of numbers, which represents number in a sequence.

In this dissertation, the Operating-mode Pointer (OMP) is employed to represent the pump operations in the Saskatoon West WDS. The extracted operating-modes contain the status of all of the network's elements, including the operational status of pumps (e.g., whether they are ON or OFF as well as their head pressure) and the percentages of water delivered to the demand nodes within its chromosome.

In this technique, the constant-demand periods are extracted by using piecewise linear-approximation over the original demand profile (i.e., the total discharge flow at the QE pumping station). The network operation is captured and then separated over the piecewise linear regions as operating-modes. Each operating-mode contained information on discharge flow and head pressure, and is represented by a string of 11-elements. This new representation is then applied to the Saskatoon West WDS (as illustrated in Figure 5.26). Here, the first two elements indicate the status of the two duty pumps at the QE station (1 *if* the pump is ON or 0 *if* it is OFF), followed by two elements indicating the two pumps' head pressures. Similarly, elements 5 and 6 indicate the operating status of the booster pump at the Aurora pumping station. The rest of the string, elements 7 to 11, contains the percentage (corresponds to valve settings) of flow delivered to the demand nodes (i.e., Corman Park, PCS Cogen, PCS Cory, Village of Vanscoy, and Agrium). Table 5.6 summarizes the description of the 11-elements operating-mode string.

Next, to represent the pumping operations over a period of one month, a set of 168 operating-modes[23] are combined (in series) to construct what is known as a chromosome (i.e., each chromosome represents an initial candidate solution used by the optimization algorithm). Herein, each operating-mode string is referred to as a gene. Each gene in the chromosome is represented by an index (i.e. pointer) that refers to a certain operating mode extracted and stored in the library from the true network operation (a complete description of the library is presented in the following subsection). Finally, a set of 100 chromosomes[24] are constituted the population (i.e. swarm) in the proposed APC-MOPSO (as illustrated in Figure 5.27).

1	**1**	**241**	**241**	**1**	**100**	**6.8**	**452.5**	**524.9**	**17.7**	**410.3**
P1	**P2**	**$Head_1$**	**$Head_2$**	**P3**	**$Head_3$**	**Corman**	**Cory**	**Cogen**	**Vanscoy**	**Agrium**

Figure 5.26: An Example of the Operating-Mode String

[23] For an hourly basis and for a one week period of pumping operation, each chromosome should have 24 hrs x 7 days = 168 genes is the length of each chromosome.

[24] According to the sensitivity analysis performed for the APC-MOPSO, the population size of 100 individuals performs well for the most multi-objective optimization problems.

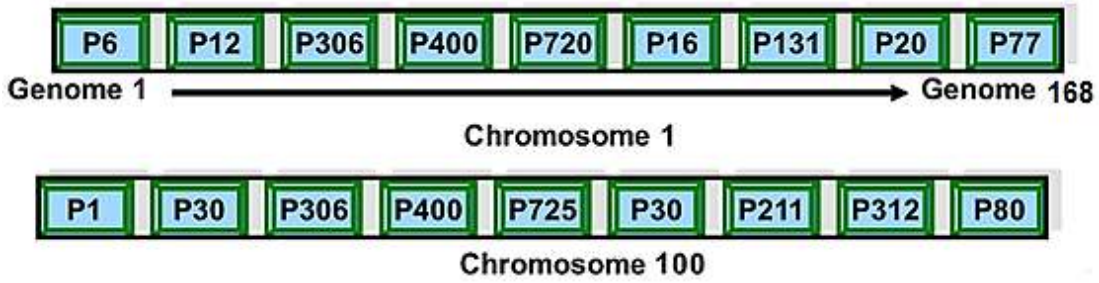


Figure 5.27: Examples of the Chromosomes Constructed Using OMP Technique

Table 5.6: Description of Operating-Mode String

Location	Remark
1	Status of duty pump 1 at QE pumping station (1 = On, 0 = OFF)
2	Status of duty pump 2 at QE pumping station (1 = On, 0 = OFF)
3	Head pressure at duty pump 1 at QE pumping station (psi)
4	Head pressure at duty pump 2 at QE pumping station (psi)
5	Status of duty pump 3 at Aurora pumping station (1 = On, 0 = OFF)
6	Head pressure at duty pump 3 at Aurora pumping station (psi)
7	Percentage of demand flow delivered to Corman Park (L/min)
8	Percentage of demand flow delivered to PCS Cory (L/min)
9	Percentage of demand flow delivered to PCS Cogen (L/min)
10	Percentage of demand flow delivered to Village of Vanscoy (L/min)
11	Percentage of demand flow delivered to Agrium (L/min)

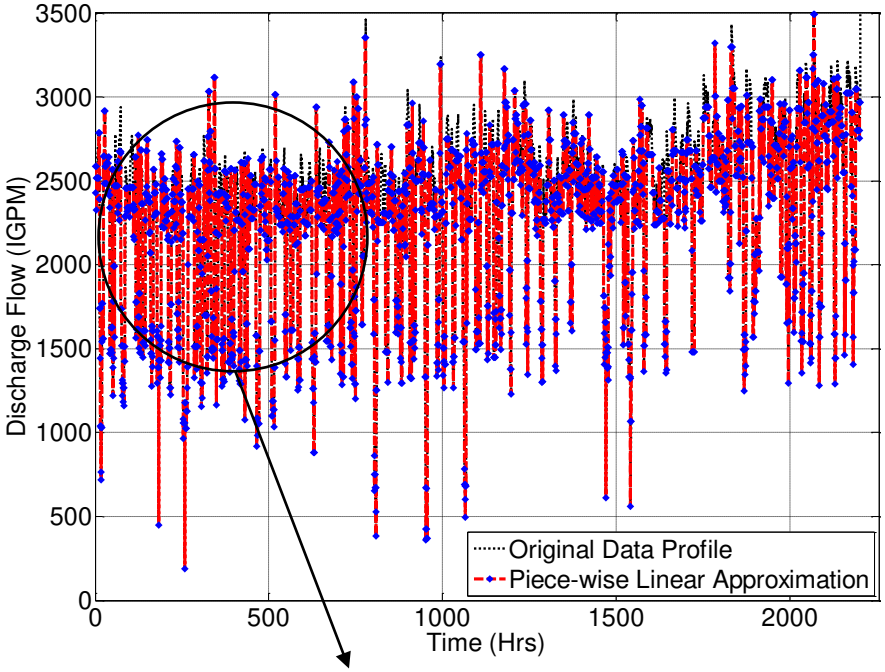
5.3.2 Operating-Mode Pointer: Library

In the new Operating-Mode Pointer (OMP), two libraries are proposed. The first library is designed to store the proven operating-modes that are originally extracted from the network profile, and the second library is created to archive new operating-modes produced by the proposed APC-MOPSO.

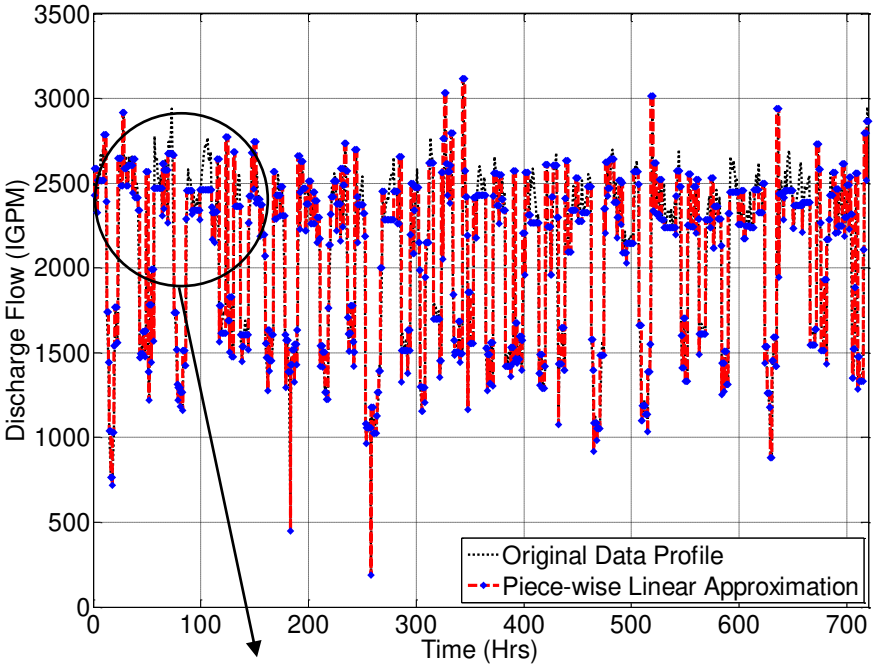
For the new energy optimization strategy, the discharge flow, pressure, and tank water level are recorded over 90 days by using SCADA systems. Figure 5.28 (a) shows the implementation of the piece-wise linear approximation on the original demand profile from March 1st to June 1st 2011 (≈2256 hrs). Figure 5.28 (b) and (c) depict the operating-modes obtained for one month (720 hrs) and one week (168 hrs), respectively. The operating-modes for the first operational day are shown in Figure 5.28 (d). Examples of the information (e.g., the discharge flow for the pumps and the water volume supplied for each demand nodes) obtained from the operating-modes are contained in Tables 5.7 and 5.8 (i.e., for the first 24 hour of the

recorded data). The corresponding pump discharge, pump scheduling, pump combination, and demands of the customers are tabulated and provided in Appendix B.

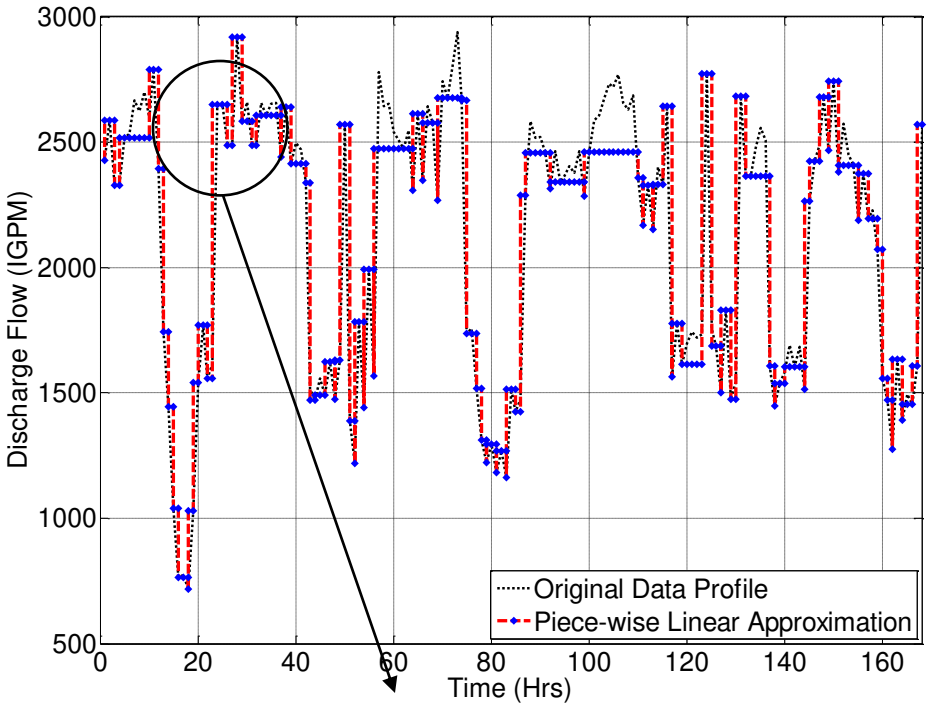
a. 2256 hrs



b. 720 hrs



c. 168 hrs



d. 24 hrs

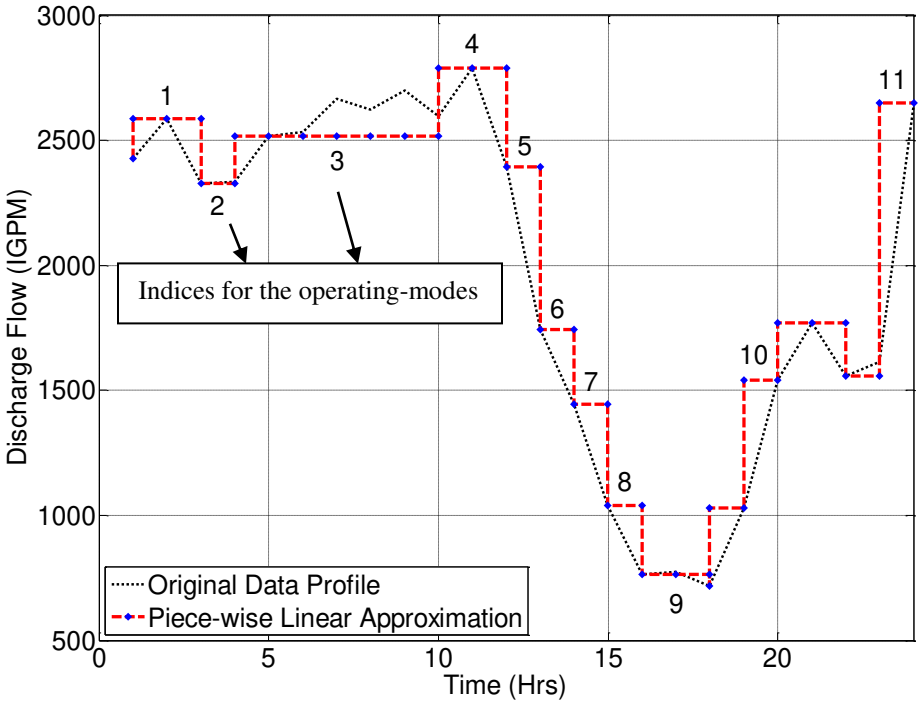


Figure 5.28: Extract the Operating Modes from the QE Total Supplied Profile Using Piece-wise Linear Approximation (March – May 2011)

Table 5.7: Operating-Modes for the Duty Pumps at QE Pumping Station for 24 Hours

Operating-Mode Pointer	Time Period		Discharge Rate(L/min)	
	From	To	1	2
1	Day 1 11:00 AM	Day 1 1:00 PM	5516	5516
2	Day 1 1:00 PM	Day 1 2:00 PM	5291	5291
3	Day 1 2:00 PM	Day 1 8:00 PM	5308	5308
4	Day 1 8:00 PM	Day 1 10:00 PM	5542.5	5542.5
5	Day 1 10:00 PM	Day 1 11:00 PM	5442	5442
6	Day 1 11:00 PM Day 1 6:00 AM	Day 1 12:00 AM Day 1 8:00 AM	3940	3940
7	Day 1 12:00 AM	Day 1 1:00 AM	3326	3326
8	Day 1 1:00 AM Day 1 4:00 AM	Day 1 2:00 AM Day 1 5:00 AM	2798	2798
9	Day 1 2:00 AM	Day 1 4:00 AM	1683	1683
10	Day 1 5:00 AM Day 1 8:00 AM	Day 1 6:00 AM Day 1 9:00 AM	1623	1623
11	Day 1 9:00 AM	Day 1 11:00 AM	3556	3556

Table 5.8: Operating-Modes for the Demands Nodes at First 24 Hours

Operating-Mode Pointer	Demand Flow (L/min)				
	Corman Park	PCS Cory	PCS Cogen	Village of Vanscoy	Agrium
1	53.0	3534.6	4101.7	137.9	3204.8
2	50.8	3390.5	3934.4	132.3	3074.1
3	51.0	3401.4	3947.0	132.7	3084.0
4	56.6	3782.3	4389.1	147.6	3429.4
5	52.2	3487.2	4046.7	136.1	3161.8
6	37.8	2524.8	2929.8	98.5	2289.1
7	31.9	2131.3	2473.2	83.2	1932.4
8	26.9	1793.0	2080.6	70.0	1625.6
9	16.15	1078.5	1251.5	42.1	977.8
10	15.6	1040.0	1206.9	40.6	943.0
11	34.1	2278.6	2644.2	88.9	2066.0

5.4 Selecting-Best-Operating Point Technique (SBOP)

In water distribution systems, pumps operate on a continuous basis, 24 hours a day and 365 days per year. Their operation entails heavy use of energy, which relates to their total

operational costs. The costs of pumping usually account for 60 to 80% of the total operational cost of water distribution systems, and the average pump efficiency is estimated to be below 40%, [287]. Furthermore, many pumps in water distribution systems are run at operating points that are varied over the course of the day. A simple and useful kind of analysis for a pump can be done by calculating the pump production versus time of day and compare it with the discharge rate at the pump's BEP. Figure 5.29 illustrates an example of a pump discharge over a two-day period. *If* the pump's BEP is approximately 400 gpm, *then* the pump is running efficiently. *If*, however, the pump's BEP is 600 gpm, *then* the pump is not running efficiently and is wasting energy. Therefore, correcting pump operation (i.e. make the pump operates closer to its best efficiency point (BEP)) can reduce costs in terms of energy usage and maintenance requirements.

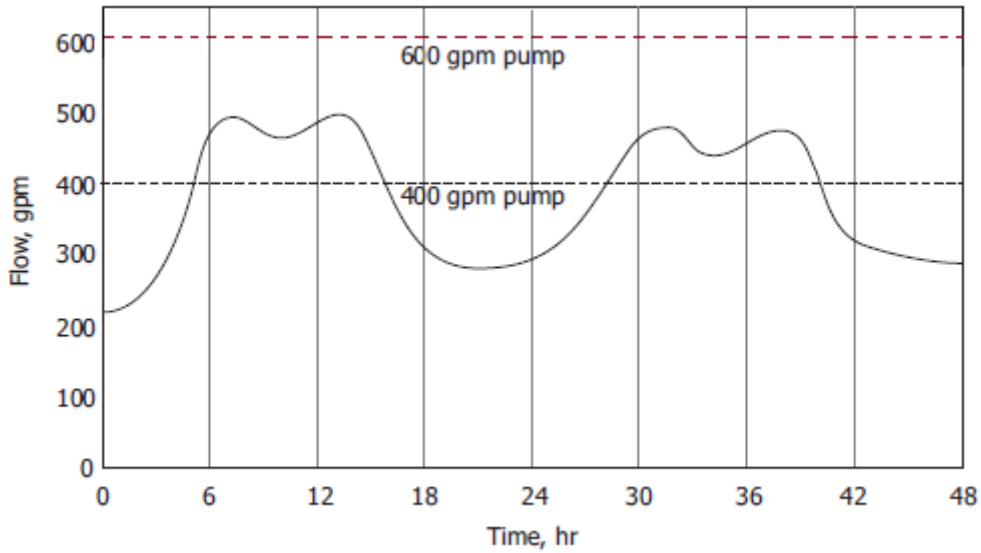


Figure 5.29: An Example of Pump Operates Away from its BEP

In this work, a new post-processing technique, referred to as Selecting-Best-Operating Point (SBOP) is proposed to provide solutions that are:

- more efficient in terms of energy costs; and
- not only satisfying the hydraulic constraints of the system, but also has operating points close to the best efficiency point (BEP).

The proposed SBEP technique is described in the following sections.

5.4.1 Reliability Curve

In a water distribution system, the definition of reliability is given as the probability of a pump to perform its prescribed duty (i.e., pump-schedule) without failure over a specified period of time and under pre-specified conditions, [288].

Figure 5.30 illustrates the concept of the proposed Selecting-Best-Operating Point (SBOP) by introducing a new curve referred to as the Reliability Curve that is superimposed on the pump-system characteristic curve. In this figure, the area under the reliability curve is divided into three regions that are the *target* region (-10% to 5% of the BEP), *re-consider* region (-20% to 10% of the BEP), and *avoid* region (-30% to 15% of the BEP). The preferred pump operations are obtained when they are in the *target* region and close to the BEP.

When a complete set of optimized solutions[25] is obtained, the proposed Selecting-Best-Operating Point (SBOP) is applied to move their operating points to the *target* region. For each solution, the proposed SBOP started by evaluating the operating point of each operating-mode (i.e., finding out how far the operating point is from the BEP). Then, SBOP classified each operating-mode as to whether it belonged to *avoid*, *re-consider*, or *target* regions. For each solution and after evaluating and classifying all operating-modes, three actions are taken: (1) *if* the operating point of an operating-mode is located in the *target* region, *then*, the operating-mode is accepted, (2) *if* the operating point is found to be in the *avoid* region, *then*, it is eliminated, and (3) *if* the operating point belonged to the *re-consider* region, *then* the SBOP uses one of the two new schemes referred to as the Speed-up and the Speed-down. These two schemes change the pump speed (RPM) in order to move the pump operating point close to the BEP. The pseudo-code for the proposed SBOP technique is given in Table 5.9.

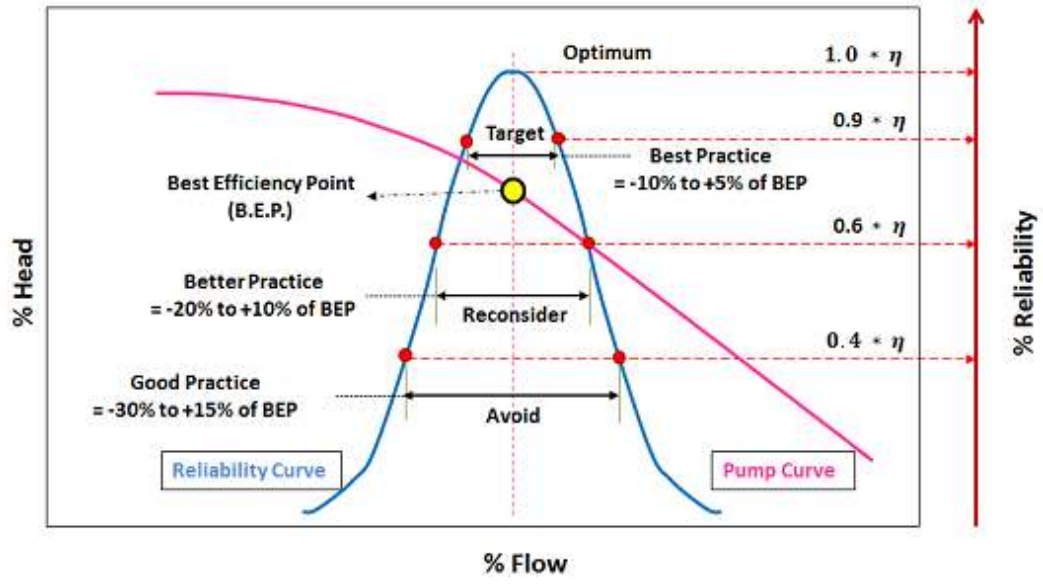


Figure 5.30: The Characteristics of Reliability Curve

[25] A set of optimized solutions (i.e., pump(s) schedules) is mainly the solutions produced by the APC-MOPSO with SFG and OMP techniques. Each solution consists of a number (i.e., varies for each solution) of different operating-modes over the duration of 168 hrs.

Table 5.9: Pseudo-code of the Proposed Selecting-Best-Operating Point Technique

From the optimization algorithm, • A set of pumping operations are produced over certain durations.
From the Selecting-Best-Operating Point (SBOP),
for each pumping operation DO
• Evaluate how far its operating point from the BEP
• Check on which region the operating point belongs
if the operating point is located in the *avoid* region, *then*, rejects that operation
else if the operating point is located in the *target* region, *then*, accepts that operation
else if the operating point is located in the *re-consider* region, *then* apply Speed-up or Speed-down approaches
end for

For the Saskatoon West WDS, the regulations provided by Sask-Water including the range of the head pressure at the QE pump station [180-220 psi] and the nominal pump speed of 1770 rpm, provides an opportunity to implement the proposed Speed-up and Speed-down approaches. Figure 5.31 illustrates the relationship between the pump's head and discharge (with respect to its speed. As shown in this figure, the range of required pump speed can vary between 1200 to 2200 rpm. This speed range is wide enough for the Speed-up and Speed-down to shift the operating points towards the BEP. Similarly, Figures 5.32 and 5.33 show head-discharge-speed relationship for the jockey pump at the QE pump station and for the pump at the Aurora pump station. The corresponding values of head, discharge, and speed for the pumps are provided in Table 5.10. The Speed-up and Speed-down schemes are explained next.

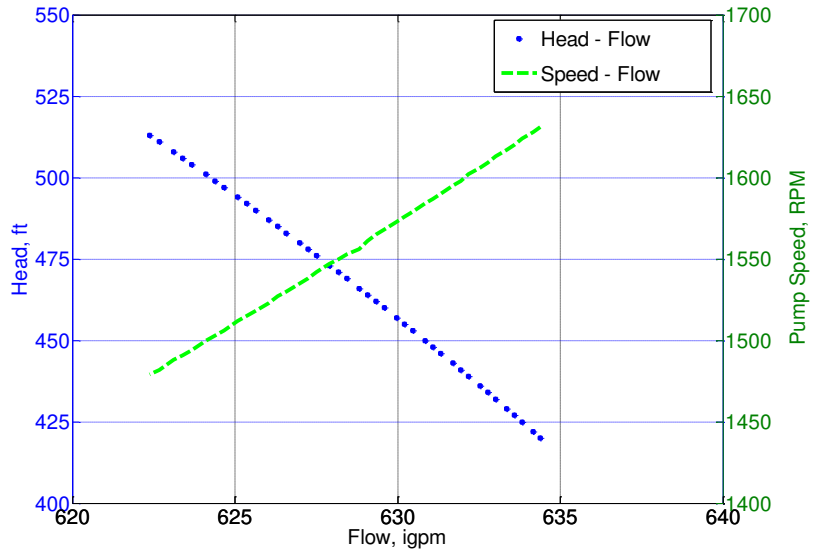


Figure 5.31: Characteristics Curves for the Pumps at the QE Pumping Station [180–220 psi]

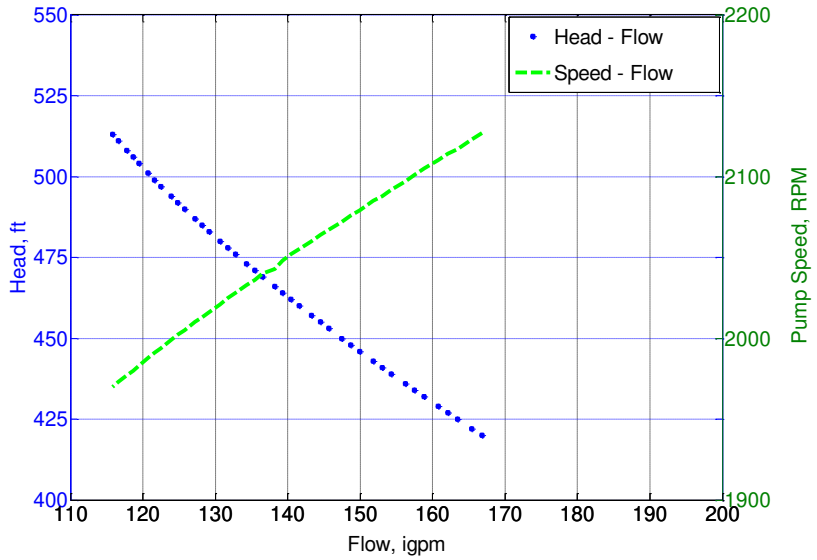


Figure 5.32: Head-Flow-Speed Relationship for the Jockey Pump at the QE Pumping Station [180–220 psi]

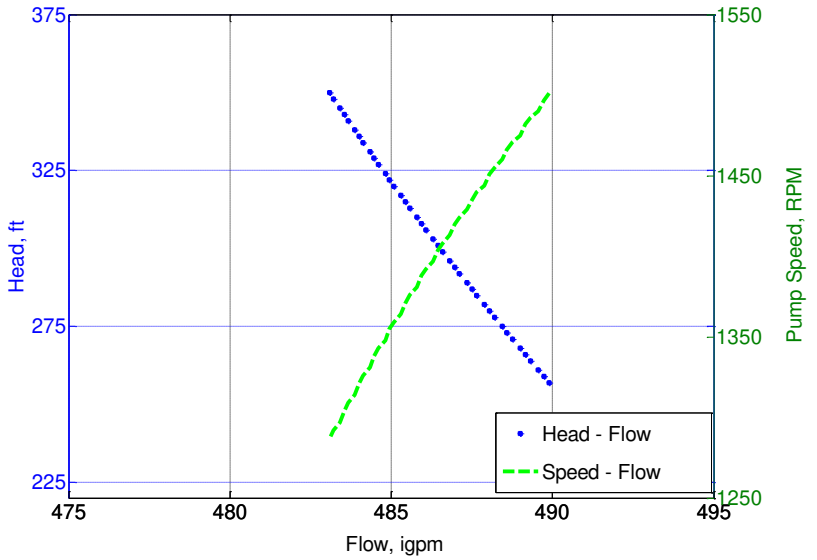


Figure 5.33: Head-Flow-Speed Relationship at Aurora Pumping Station [110–150 psi]

Table 5.10: Estimated Pumps Characteristics at the QE and Aurora Pump Stations

Station	Head (ft)		Discharge (igpm)		Speed (RPM)	
	From	To	From	To	From	To
QE Duty Pumps	420	513	622	634	1479	1631
QE Jockey Pump	420	513	116	167	1973	2127
Aurora Duty Pump	257	350	483	490	1287	1501
Aurora Jockey Pump	257	350	483	490	1287	1501

5.4.2 Speed-up and Speed-down Schemes

The idea behind the post-processing Speed-up and Speed-down schemes is to move the operating point of a pump towards its best efficiency point (BEP) by either increasing or decreasing its speed (RPM).

The proposed Speed-down scheme is applied when the operating point produced by the optimization algorithm is in the *re-consider* region of the reliability and has a flow rate bigger than that of the BEP's. Hence, the action of lowering the pump speed (RPM) is required to bring that operating point closer to the BEP. In contrast, the speed-up scheme is triggered when the operating point belongs to the *re-consider* region of the reliability curve and has a flow rate that is less than that of the BEP's. Then, an action of increasing the pump speed (RPM) is required to shift that point towards the BEP. The mathematical formulations for the two approaches are explained as follows.

Pump head can be approximated as a function of speed (RPM) by a quadratic equation:

$$H = a\,Q^2 + b\,Q + c \tag{5.4}$$

At the QE pumping station, two identical duty pumps are used in parallel, hence, the pump equation can be specified as:

$$H = a\left(\frac{Q}{2}\right)^2 + b\left(\frac{Q}{2}\right) + c \tag{5.5}$$

At the BEP, the above equation can be written as:

$$H_{BEP} = a\left(\frac{Q_{BEP}}{2}\right)^2 + b\left(\frac{Q_{BEP}}{2}\right) + c \tag{5.6}$$

If the pump operates at a discharge flow different from Q_{BEP}, then by applying the Affinity Laws[26], [289]:

$$\frac{Q}{Q_{BEP}} = \frac{RPM}{RPM_{BEP}} = Sr \quad and \quad \frac{H}{H_{BEP}} = \left(\frac{RPM}{RPM_{BEP}}\right)^2 = Sr^2 \tag{5.7}$$

where Sr is tha Affinity ratio

Then, Eq. (5.7) can be re-written as:

$$\frac{H}{Sr^2} = a\left(\frac{Q_{BEP}}{2 \times Sr}\right)^2 + b\left(\frac{Q_{BEP}}{2 \times Sr2}\right) + c$$

$$H = a\left(\frac{Q_{BEP}}{2}\right)^2 + b\left(\frac{Q_{BEP}}{2}\right) \times Sr + c \times Sr^2 \tag{5.8}$$

Write the above equation for *Sr*:

[26] The affinity laws are used to describe the relationship between variables involved in pumps such as head, flow rate, efficiency, speed, and power. They are used to predict the head-discharge characteristic of a pump from a known characteristic measured at a different speed or impeller diameter. The only requirement is that the two pumps must be dynamically similar.

$$\hat{a}\, Sr^2 + \hat{b}\, Sr + \hat{c} = 0 \quad (5.9)$$

where: $\hat{a} = c \;\; , \;\; \hat{b} = b\frac{Q}{2} \;\; , \;\; \hat{c} = a\left(\frac{Q}{2}\right)^2 - H$ (5.10)

Sovling the quatratic equation, Eq. (5.9), yields:

$$Sr = \frac{-\hat{b} \pm \sqrt{\hat{b}^2 - 4\,\hat{a}\,\hat{c}}}{2\hat{a}} \quad (5.11)$$

Sr is re-calculated whenever the operating point, produced by the optimization algorithm, belongs to the *re-consider* region of the reliability curve. Figure 5.34 shows the flowchart of the proposed Speed-up and Speed-down schemes.

To apply the Speed-up and Speed-down schemes to the pumps at the Saskatoon West WDS, some of the pump congfiguration parameters need to be determined, including pump coefficiencts and pump head, discharge, and speed at its BEP. Table 5.11 provides the values of the parameters for the pumps at the system.

Table 5.11: Configuration Parameters for the Pumps at the Saskatoon West WDS

Pump	**Coefficients**			**Characteristics at BEP**		
	a	**b**	**b**	**Head (ft)**	**Discharge (igpm)**	**Speed (RPM)**
QE Duty Pumps	0.2267	-1.179E-04	587.53	550	922	1770
QE Jockey Pump	0.0864	-9.808E-04	290.29	300	199	1170
Aurora Duty Pump	0.1388	-8.967E-05	416.199	400	1400	1750
Aurora Jockey Pump	0.1388	-8.967E-05	416.199	400	1400	1750

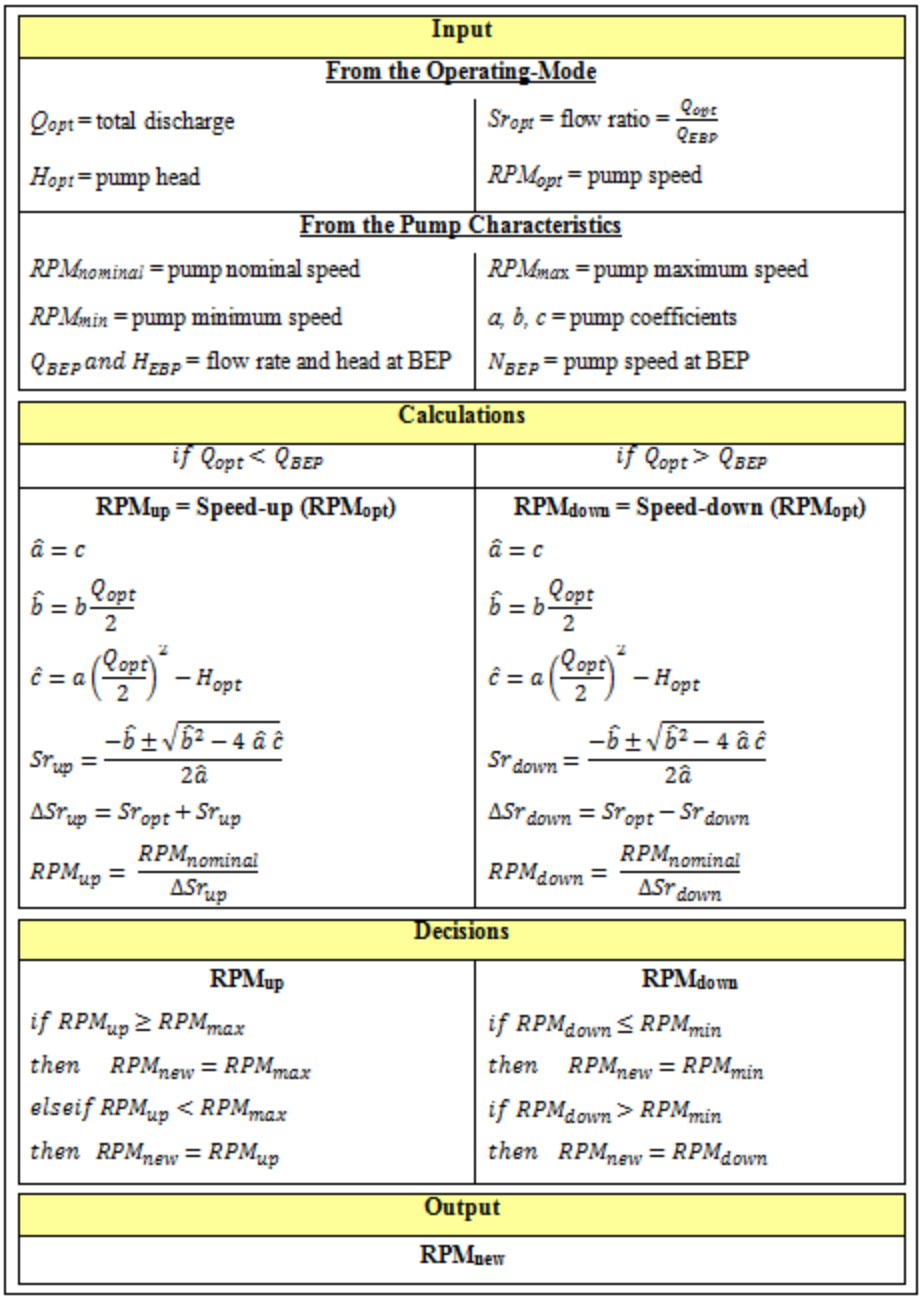


Figure 5.34: The Mechanism of Using Speed-up and Speed-down Schemes

5.5 Conclusion

The convergence proof of the current heuristic search algorithms toward the Pareto-optimal front of multi-objective optimization problems is given in Chapter 4. Therefore, the focus of the research is directed to obtain a finite set of non-dominated solutions that capture the entire

front region by using what is referred to as the Gaps Free (or 'tight') techniques. Although several archiving strategies are proposed for obtaining a finite set of Pareto-optimal front, the strategies do not guarantee a gap free of their front sets. That is, in this work, a new Searching-for-Gaps (SFG) is proposed to use an auxiliary external repository with an adaptive grid archiving for obtaining Pareto-front solutions that cover the entire global regions. The results demonstrated the effectiveness of the APC-MOPSO with SFG, and showed well how the diversity and the distribution capability are improved.

In water distribution systems, it is very difficult and not practical to achieve increases of the operational efficiency and/or reductions of the operational costs by means of physical modifications in the facilities. Operating-Mode Pointer (OMP) is proposed to be a technique that practically guarantees obtaining of efficient pumping operations in terms of effective-cost, safety and reliability.

In pumping stations, several problems can result when pumps operate away from their BEP. These problems are attributed to maintenance cost, operational costs, and the costs associated with wasted energy. The Selecting-Best-Operating Point (SBOP) is proposed to enhance the reliability of the optimized operating-modes in terms of flow rate, head pressure, and speed.

Chapter 6

Energy Optimization Strategy

As time goes on, more and more operating-modes based on changing demand profiles will be compiled to enrich the range of feasible solutions for a water distribution system. This implies the conservation of energy consumed by a water pumping station and improves the ability for energy optimization. Another important goal is improving safety, reliability, and maintenance cost. In this work, three important goals are addressed: cost-effectives, safety, and self-sustainability operations of water distribution systems.

In this dissertation, the objectives to optimize are: electrical energy consumption; total number of pump switches; cost of maximum power peak; reservoir water level variation; and water quality while preserving the service provided to water users.

To accomplish these goals, the new energy optimization strategy of Adaptive Parallel Clustering-based Multi-objective Particle Swarm optimization (APC-MOPSO) with Searching-for-Gaps (SFG), Operating-Mode Pointer (OMP), Selecting-Best-Operating-Point (SBOP), and Modified EPANET is used. The strategy resulted in a Pareto-front with solutions that are quantitatively equally good. This provides the decision maker at the water distribution system with the opportunity to qualitatively compare the solutions before their implementation into practice.

In this chapter, the energy optimization strategy for WDS and its main components are first presented. Section 6.2 describes the characteristics of the Saskatoon West WDS components: consumption profile, demand profiles, pump discharge, pump efficiency, the derivation of the classical demand-head-driven analysis related to the pump operations, and reservoir level over the week of Tuesday March 1st 2011. This is followed by investigating, discussing, and comparing the optimization results obtained for the case study considered in this work with experimental data.

6.1 Approach Proposed in this Thesis

The multi-objective optimization approach developed in this dissertation follows modern applications that combine an optimization algorithm with a network simulation model by using full hydraulic simulations and distributed demand models, [290]. Figure 6.1 shows the architecture of the proposed optimization system for the system-operating problem[27]. The input to the system includes both a complete model of the water distribution system as well as its demand profiles over the period of optimization. The Operating-mode Pointer (OMP) and the optimization algorithm, the Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization algorithm (APC-MOPSO) with Searching-for-Gaps (SFG), then generate candidate solutions (pump schedules). These are in turn evaluated by the hydraulic simulation, EPANET, to satisfy the system and operation constraints (e.g., node pressure and pump discharge) as well as the constraint violations.

In this architecture, the optimization algorithm is independent from the hydraulic simulator (i.e., modifications to either of them do not imply changes to the other). Selecting-Best-Operating Point (SBOP) is applied to modify the optimized operating-modes in terms of pump characteristics such as head pressure, discharge, speed, and efficiency.

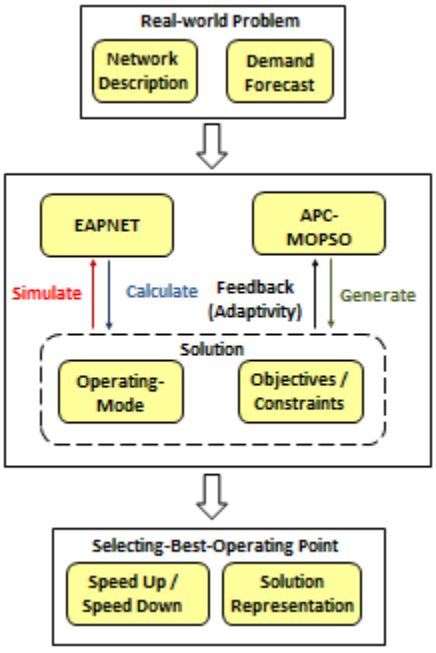


Figure 6.1: Schematic Diagram of Energy Optimization System

[27] The term "system-operating problem" is used to denote the combination of pump-scheduling and reservoir-operating problem considered in this work.

6.2 Saskatoon West WDS: Current Operating Scenario

In Chapter 3, a mathematical model of the Saskatoon West WDS has been outlined and verified based on field data provided by Sask-Water. The model is used to simulate the network's hydraulic behavior and to analyze its pumping operations over the three-month period starting from March to May 2011. The current operating scenario (i.e., within a three month period) is analyzed and used to extract the operating-modes as previously explained in Chapter 5. A total of 297 operating-modes are extracted, each containing the status of all system components at that time.

In this work, the energy optimization strategy is applied over a period of one week from 1st to 7th March 2011. The total supplied profile for this week is measured as illustrated in Figure 6.2.

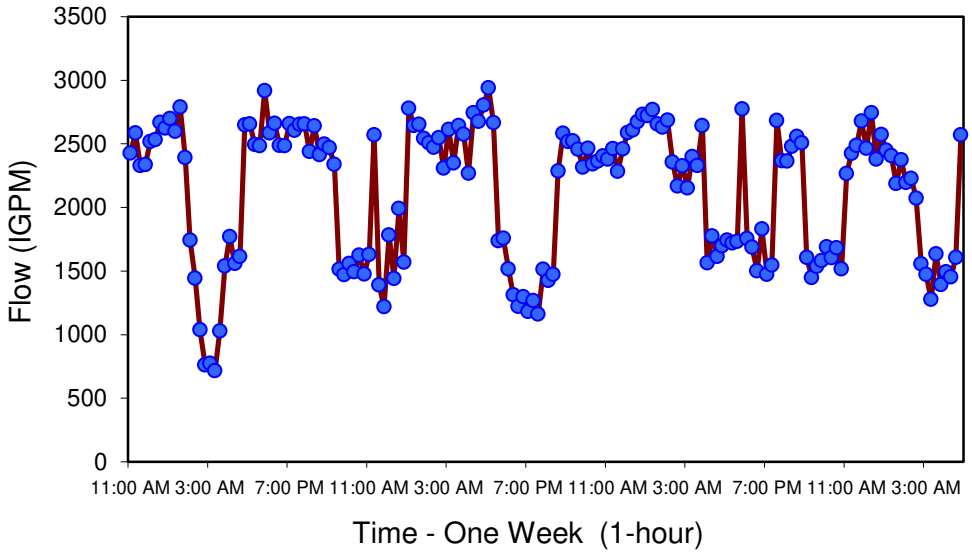


Figure 6.2: Total Volume of Water Supplied at QE Pumping Station (1st-7th March 2011)

Hourly-based customer consumption profiles are derived using the total water supply profile (i.e., shown above). The percentage of monthly consumption of each of the main clients is listed in Chapter 3 (see Table 3.17). The weekly demand profiles for the customers are summarized in Figures 6.4, 6.5, and 6.6.

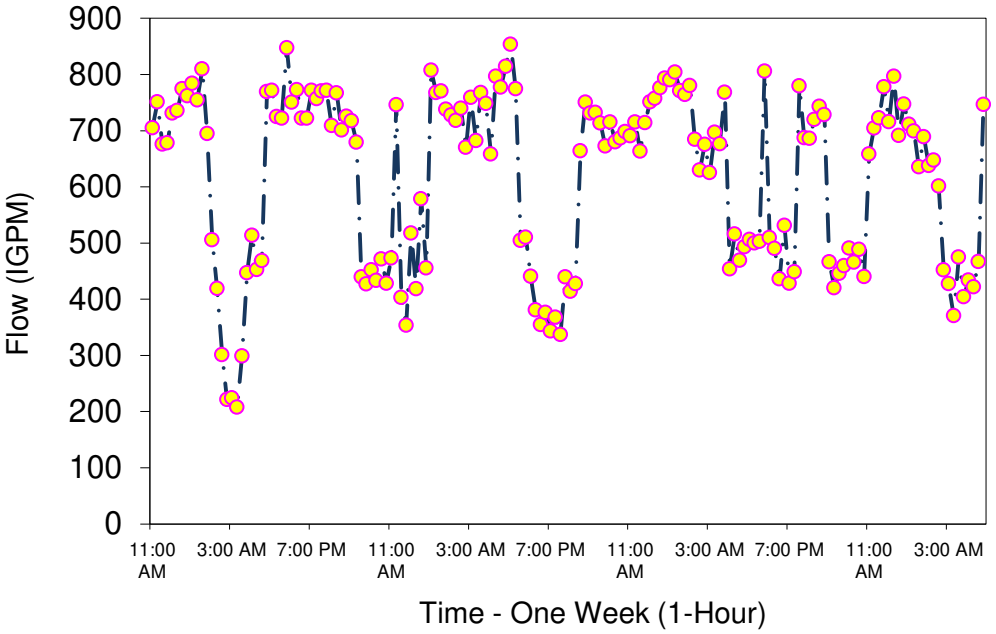


Figure 6.3: Demand Profile of Agrium (1st to 7th March 2011)

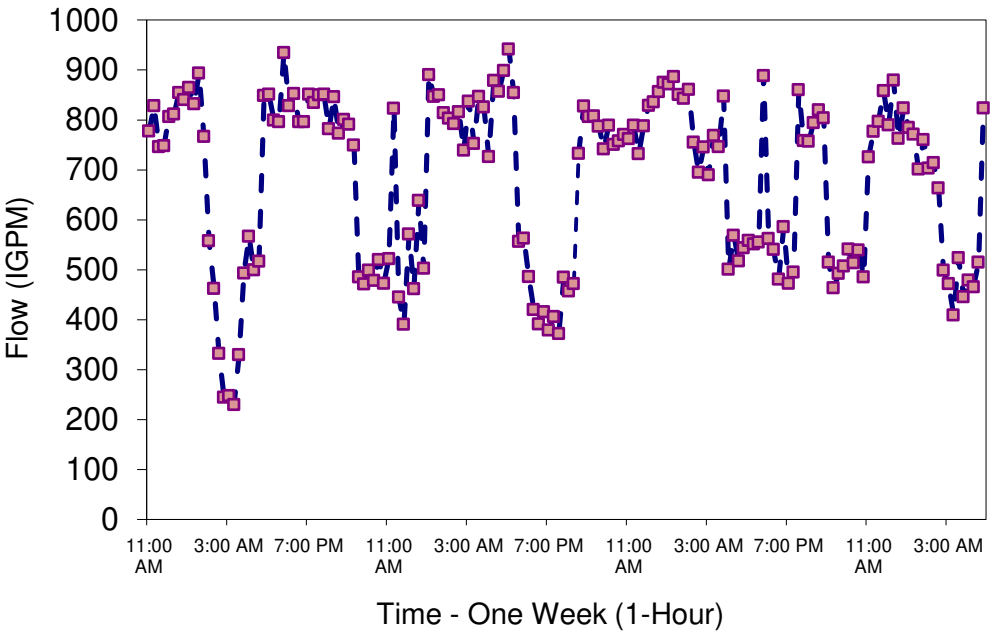


Figure 6.4: Demand Profile of PCS Cory (1st to 7th March 2011)

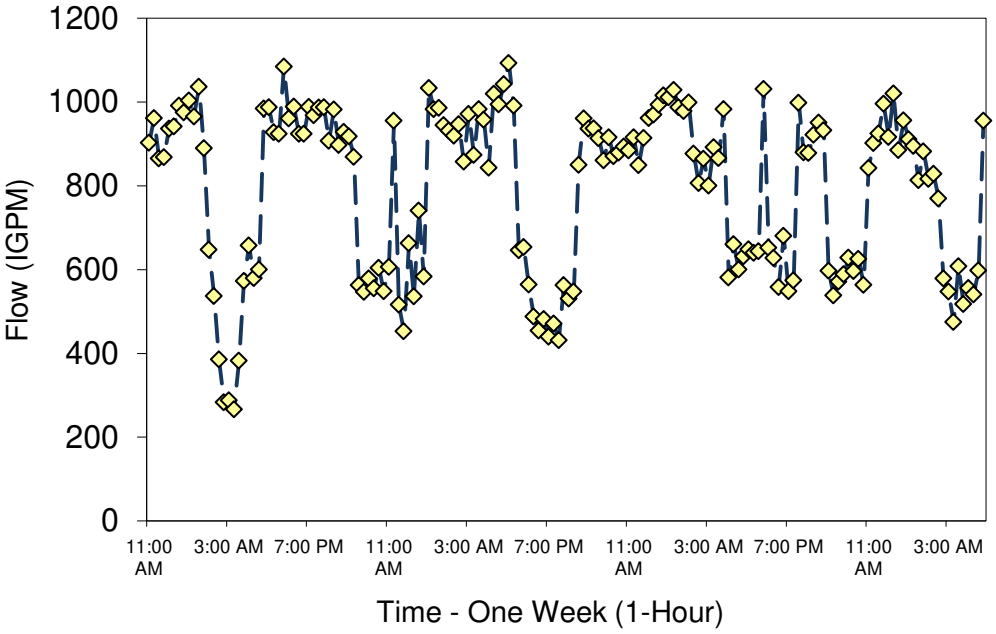


Figure 6.5: Demand Profile of PCS Cogen (1st to 7th March 2011)

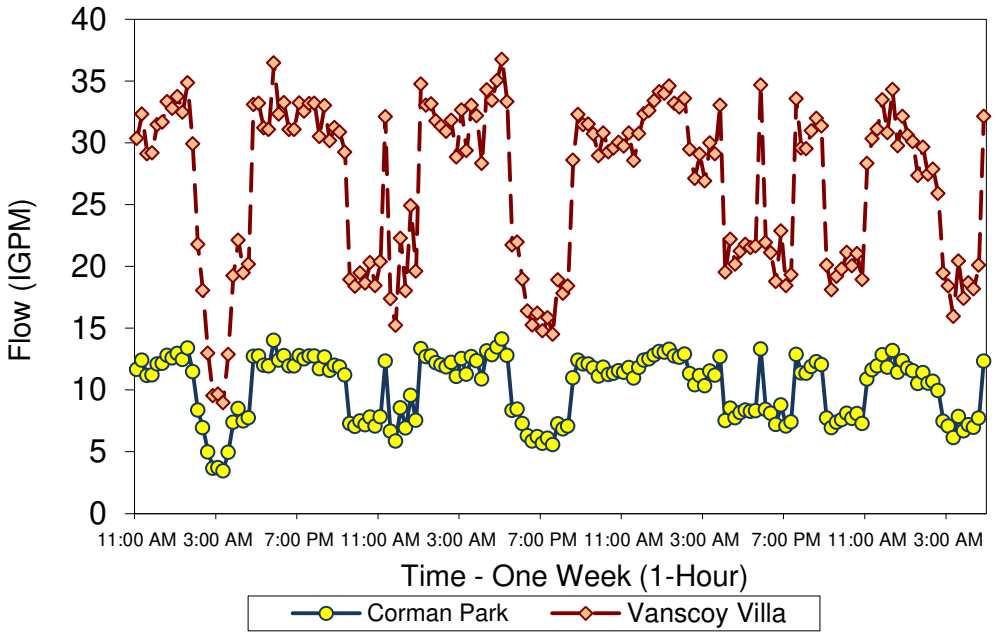


Figure 6.6: Demand Profiles of Corman Park and Vanscoy Villa (1st to 7th March 2011)

The discharge flow and the efficiency variation of the two duty pumps at the QE pumping station for the study period of one week are illustrated in Figures 6.7 and 6.8.

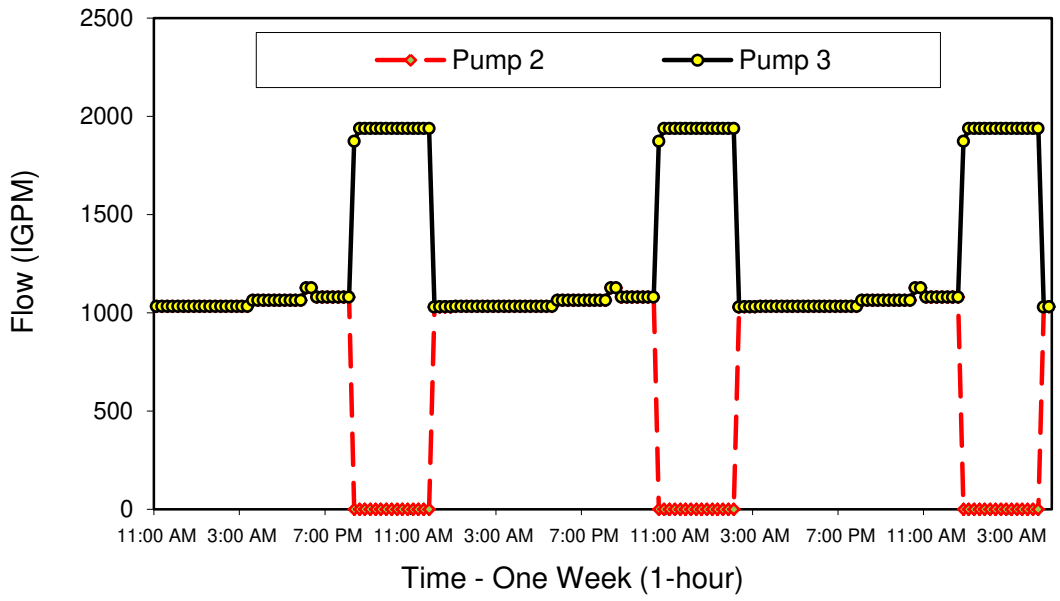


Figure 6.7: Pumps Flow Rates at QE Pumping Station (1st to 7th March 2011)

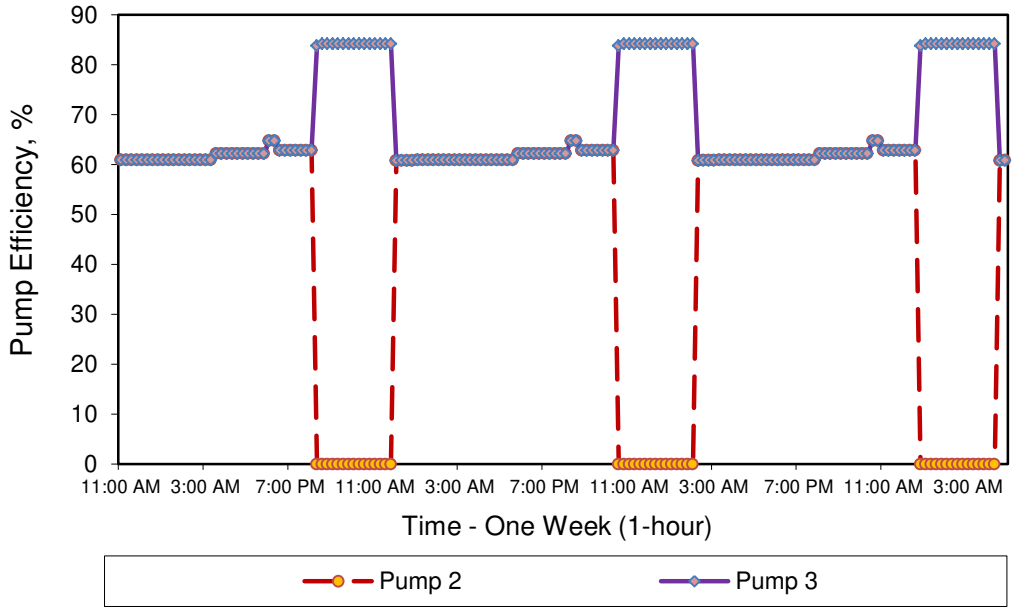


Figure 6.8: Pumps Efficiency for the Duty Pumps (1st to 7th March 2011)

For Selecting-Best-Operating Point (SBOP) (proposed in Chapter 5), the existing modes of operation need to be changed with respect to the head and discharge values of the pump best efficiency point (BEP). Therefore, second and third order plynomial functions are used to invesitgate the relationships between the efficiency, flow, and speed. It should be noted that these relationships are derived according to the range of the head presure between 180-220 psi for the

pumps at the QE station and between 110-150 psi for the pumps at the Aurora station (as shown in Figures 5.31 and 5.32).

Figures 6.9 to 6.12 show the fluctuation in the water level of the main four tanks at Agrium, the Villa of Vanscoy, PSC Cory, and Corman Park. For each tank profile, a dashed line is used to indicate its maximum allowable capacity.

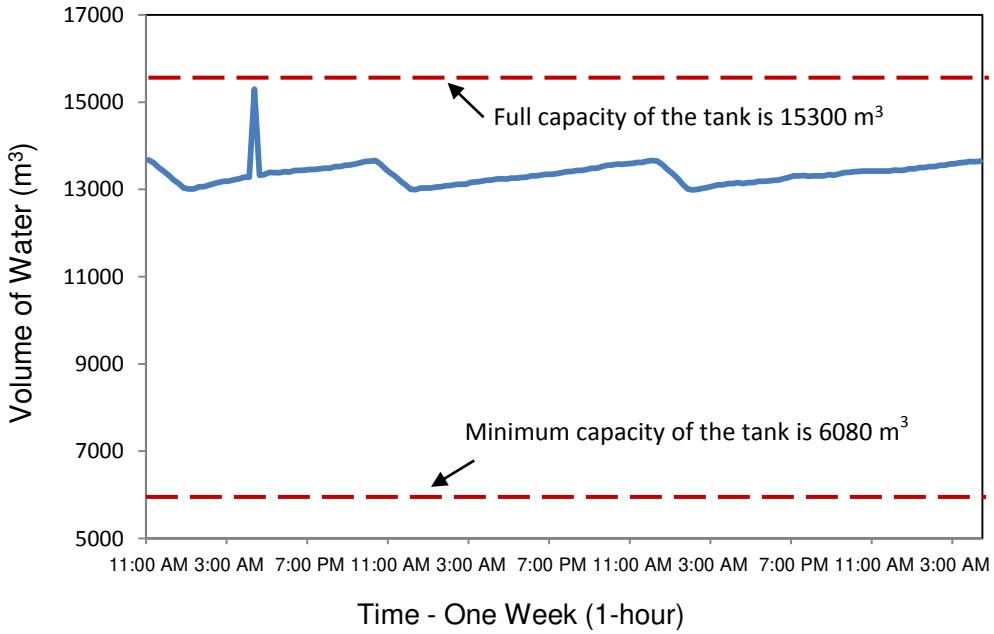


Figure 6.9: Reservoir Level at Agrium (1st to 7th March 2011)

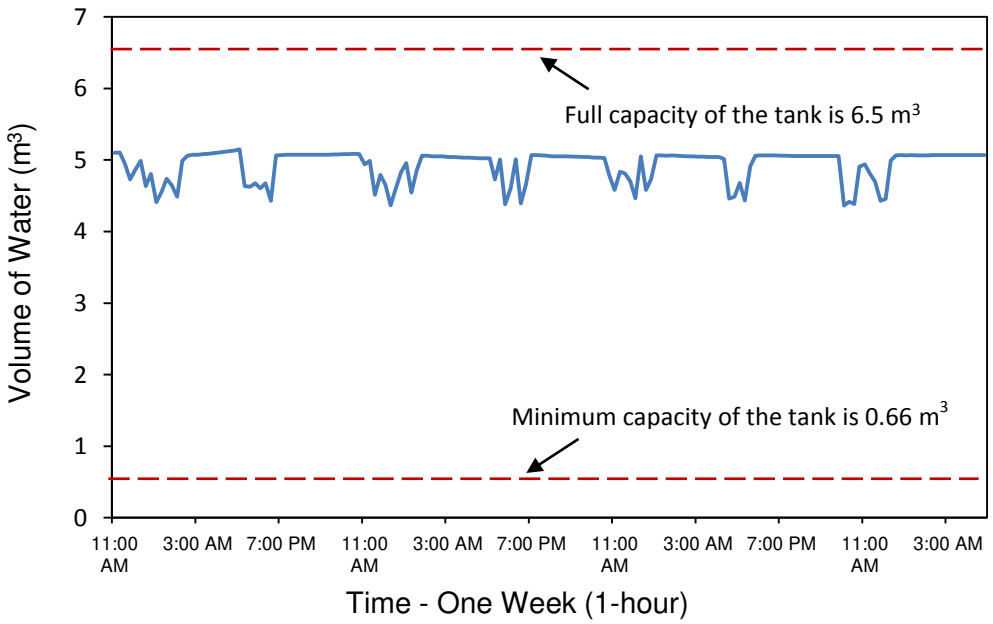


Figure 6.10: Reservoir Level at Village of Vanscoy (1st to 7th March 2011)

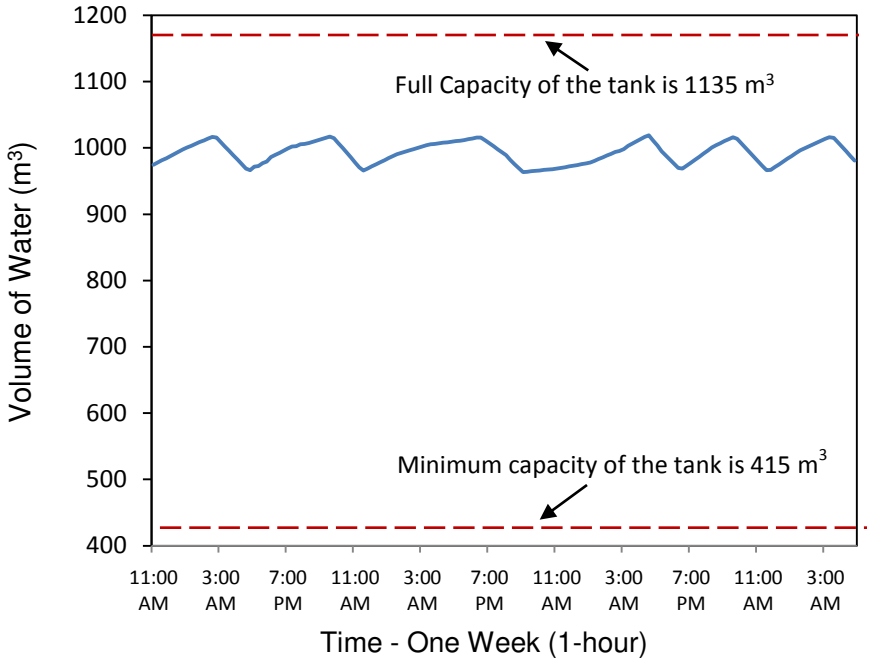


Figure 6.11: Reservoir Level at PCS Cory (1st to 7th March 2011)

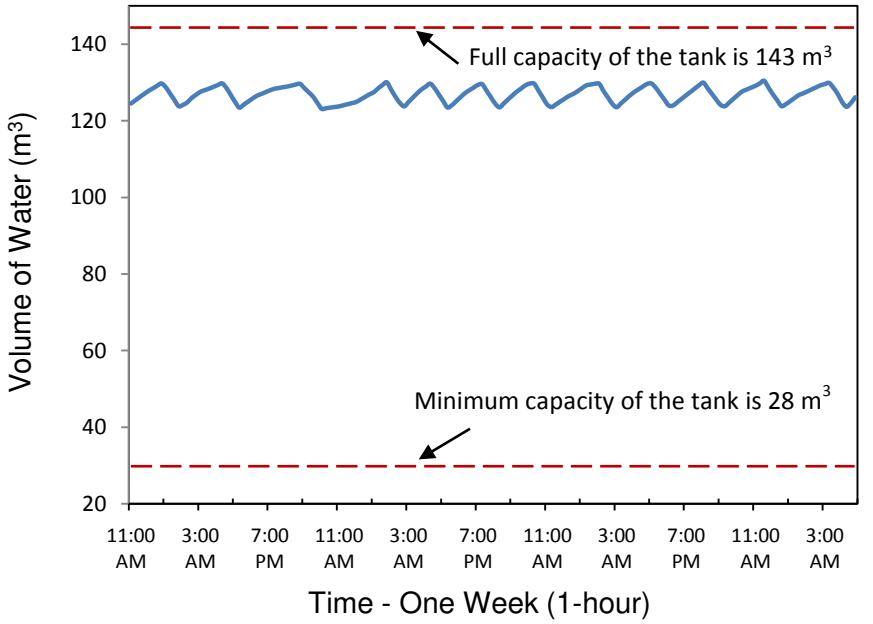


Figure 6.12: Reservoir Level at Corman Park (1st to 7th March 2011)

6.3 Energy Optimization Strategy: Results and Analysis

The energy optimization strategy is applied to the Saskatoon West WDS to determine the optimal pumping-operation schedule and analyze some important factors in pump maintenance and reliability.

The proposed APC-MOPSO algorithm combined with EPANET, Operating-Mode Pointer (OMP), and Selecting-Best-Operating Point (SBOP) has to be tested on the hydraulic model of the Saskatoon West WDS.

A reduction of approximately 7 to 14% in total operation cost is targeted after applying the proposed energy optimization strategy. Accordingly, changes in the amount and the profile of the consumers' demand are considered without affecting their consumption profiles. This change is possible because of the availability of reservoirs that allow the customers to store water and consume it later.

In this work, a case study is investigated as explained in the following sections.

6.3.1 Setup

To overcome the problem of defining network boundary and operational conditions, it is important to plan pumping operations for more than one day, [192]. Planning pumping operations for multiple days is useful for systems with tanks that are large enough to hold significant storage volumes (e.g., in Saskatoon West WDS). In such situations, water can be stored when pumps are turned ON, and the tanks can be emptied when pumps are turned OFF without affecting the consumption profile of the customers. Hence, for the Saskatoon system, results are obtained for 168 fixed time intervals of 1 hour[28].

For the case scenario, each run is conducted using the values of the APC-MOPSO parameters described in Chapter 4 (see Tables 4.2, 4.3, and 4.4) and a different randomly selected initial population of 100 candidate solutions is chosen. Moreover, each run using APC-MOPSO and EPANET is evaluated for 100 generations and the subsequent results are used for comparison purposes. The parameters of the system-operational problem considered in this work are based on technical characteristics of the main pumping stations of the Saskatoon West WDS, and described in Table 6.1. All experiments are run on a Pentium D (2.8 GHz) with 2 GB Ram using GNU/WINDOWS 7.

[28] For 5 pumps and 168 time intervals, the number of possible potential solutions (i.e., possible ways of pump combination) is found to be $2^{168\times5} > 10^{252}$ which defines the search space of the problem.

Table 6.1: Technical Characteristics and Parameters of Saskatoon West WDS

Chlorine Concentration (mg/L)	
Junction Node	**Range***
Corman Park	[0.27 – 51]
PCS Cory	[0.3 – 0.6]
Village of Vanscoy	[0.2 - 0.41]
Agrium	[0.23 – 0.25]
Reservoir Water Level (ft)	
Reservoir	**Initial Level**
Corman Park	4.9
PCS Cory	8.2
Village of Vanscoy	4.5
Agrium	20
QE Pumping Station	
Parameter	**Status**
Discharge Rate	• Flow rate is evenly divided when two duty pumps are working at the same time interval
Number of Pump switches	$\frac{168}{2} \times 3 = 252$
Pumping-operation	• Two pumps are working for each time interval
Aurora Pumping Station	
Parameter	**Status**
Pumping-operation	• One pump is working for each time interval
Number of Pump switches	$\frac{168}{2} \times 2 = 168$
Hydraulic Model (EPANET)	
Component	**Description**
Pumps	• Each pump is assigned to a H-Q curve
Junction Nodes	• Each node is assigned to demand pattern
Electricity Tariff Period	• 168 hours starting at 11:00 am and ending at 10:00 am

* *The range is determined based on the flow rate range of [1000 gpm – to- 2000 gpm]*

The details of the case study and the analysis of its optimized results are thoroughly presented as follows.

6.3.2 Case Study

In this case study, the system-operational problem is set as a multi-objective optimization formulation that tries to determine the most economically effective pumping operation for the Saskatoon West WDS. These objectives include:

- minimizing the electrical energy cost;
- minimizing the total number of pump switches (i.e., reducing maintenance cost);
- minimizing the cost of maximum power peak (i.e., reducing demand charge)
- minimizing the reservoir water level variation (i.e., enhancing network reliability); and
- minimizing the free chlorine residual (i.e., improving water quality).

Given multiple competing objectives, it is not possible to improve one objective (e.g., reduce the risk of not meeting demands, network reliability) without making at least one of the other objectives worse (e.g., increasing electrical energy cost and/or the cost of maximum power peak).

The mathematical formulation for the five objectives is provided in Chapter 3. An optimization period of one week is considered starting from Tuesday 11:00 am.

The trade-off between the electricity cost, cost of maximum power peak, maintenance cost, network reliability, and water quality is analyzed and presented in more detail for the Saskatoon West WDS as follows.

6.3.2.1 Optimization Results

To compare the solutions for the Saskatoon West WDS obtained by the energy optimization strategy, the range of corresponding objectives values of electrical energy cost ($), number of pump switching, cost of maximum power peak ($), reservoir level (ft), and chlorine concentration (mgL) need to be identified for all solutions. Accordingly, Table 6.2 provides the range of each objective.

Amongst the solutions[29] provided, fifteen solutions are then selected to evaluate and analyze the performance of the proposed strategy. Therefore, there is a need for a criterion that identifies and labels the solutions based on their values with respect to each objective. And thus, a statistical criterion is applied to select three solutions for each objective based on their highest, median, and lowest corresponding values. A set of fifteen solutions is then used to make the comparison and the discussion of the experimental results for this case study.

For the first set of solutions (solutions 1, 2, and 3), the whole solutions are sorted based on their electrical cost. Three solutions are then selected: one that has the highest electrical cost,

[29] There are 100 solutions obtained by the energy optimization strategy when it is applied to the Saskatoon West WDS.

one that produced the lowest cost, and one from the median of the cost range. Table 6.3 provides not only the electrical costs of these solutions, but also their values with respect to the other objectives.

Similarly, for the second set of solutions, the solutions are sorted according to their number of pump switches. Another three solutions (solutions 4, 5, and 6) are selected using the statistical criterion mentioned above. Table 6.4 reports the values of these solutions in regards to all objectives. The same steps are repeated for the reservoir level (solutions 7, 8, and 9), cost of maximum power peak (solution 10, 11, and 12), and chlorine concentration (solutions 13, 14, and 15). Solutions with the highest (best), the median, and the lowest (worse) values for the three objectives are selected. The corresponding values of the solutions (6 to15) are given in Tables 6.5, 6.6, and 6.7.

Table 6.2: Range of the Experimental Results for the Saskatoon West WDS

Objective	**Range of the Objective**	
	From	**To**
Electrical Energy Cost ($)	2926.6	3138.9
Number of Pump Switching	73	143
Reservoir Level Variation (ft)	-2.9	3.4
Maximum Power Peak Cost ($)	1323.5	1405.2
Chlorine Concentration (mgL)	1.1	2.4

Table 6.3: Experimental Results for the Saskatoon West WDS – Objective 1

Based on the Range of Electrical Energy Cost							
		Cost of Electrical Energy ($)			**Cost of Maximum Power Peak ($)**		
Solution	**Label**	**Actual Cost**	**Optimized Cost**	**Cost Saving (%)**	**Actual Cost**	**Optimized Cost**	**Cost Saving (%)**
Highest	1	3315.5	**3138.9**	5.31	1460.4	1384.6	5.19
Median	2	3315.5	**3020.2**	8.91	1460.4	1389.4	4.86
Lowest	3	3315.5	**2926.6.4**	11.73	1460.4	1415.5	3.07
			Average	**8.65 %**		**Average**	**4.37 %**

Solution	**Label**	**Number of Pump Switching**	**Reservoir Level (ft)**	**Chlorine Residual (mgL)**
Highest	1	81	3.1	1.3
Median	2	95	-1.9	1.8
Lowest	3	131	-2.7	1.4

Table 6.4: Experimental Results for the Saskatoon West WDS – Objective 2

Based on the Range of the Total Number of Pump Switching							
		Cost of Electrical Energy ($)			Cost of Maximum Power Peak ($)		
Solution	Label	Actual Cost	Optimized Cost	Cost Saving (%)	Actual Cost	Optimized Cost	Cost Saving (%)
Highest	4	3315.5	2977.7	10.19	1460.4	1395.9	4.42
Median	5	3315.5	3038	8.37	1460.4	1387.7	4.98
Lowest	6	3315.5	3106.6	6.3	1460.4	1334.7	8.61
			Average	**8.29 %**		**Average**	**6.0 %**

Solution	Label	Number of Pump Switching	Reservoir Level (ft)	Chlorine Residual (mgL)
Highest	4	**143**	1.1	1.2
Median	5	**95**	-1.8	1.9
Lowest	6	**73**	2.7	1.5

Table 6.5: Experimental Results for the Saskatoon West WDS – Objective 3

Based on the Range of the Reservoir Level Variation							
		Cost of Electrical Energy ($)			Cost of Maximum Power Peak ($)		
Solution	Label	Actual Cost	Optimized Cost	Cost Saving (%)	Actual Cost	Optimized Cost	Cost Saving (%)
Highest	7	3315.5	2998.5	9.56	1460.4	1386.3	5.08
Median	8	3315.5	3046.3	8.12	1460.4	1392.8	4.63
Lowest	9	3315.5	3115.2	6.04	1460.4	1398.9	4.21
			Average	**7.91 %**		**Average**	**4.64 %**

Solution	Label	Number of Pump Switching	Reservoir Level (ft)	Chlorine Residual (mgL)
Highest	7	122	**3.4**	1.9
Median	8	119	**0.5**	1.6
Lowest	9	81	**-2.9**	2.1

Table 6.6: Experimental Results for the Saskatoon West WDS – Objective 4

Based on the Range of the Maximum Power Peak Cost							
		Cost of Electrical Energy ($)			Cost of Maximum Power Peak ($)		
Solution	Label	Actual Cost	Optimized Cost	Cost Saving (%)	Actual Cost	Optimized Cost	Cost Saving (%)
Highest	10	3315.5	3112.6	6.12	1460.4	**1405.2**	3.79
Median	11	3315.5	3075.8	7.23	1460.4	**1367.7**	6.35
Lowest	12	3315.5	3059.4	7.72	1460.4	**1323.5**	9.37
			Average	**7.02 %**		**Average**	**6.48 %**

Solution	Label	Number of Pump Switching	Reservoir Level (ft)	Chlorine Residual (mgL)
Highest	10	79	-2.7	1.9
Median	11	88	2.4	1.6
Lowest	12	113	2.8	2.1

Table 6.7: Experimental Results for the Saskatoon West WDS – Objective 5

Based on the Range of the Chlorine Concentration							
		Cost of Electrical Energy ($)			Cost of Maximum Power Peak ($)		
Solution	**Label**	**Actual Cost**	**Optimized Cost**	**Cost Saving (%)**	**Actual Cost**	**Optimized Cost**	**Cost Saving (%)**
Highest	13	3315.5	3076.5	7.21	1460.4	1389.9	4.83
Median	14	3315.5	3087.7	6.87	1460.4	1382.7	5.32
Lowest	15	3315.5	3082.4	7.03	1460.4	1387.7	4.98
			Average	**7.04 %**		**Average**	**5.03%**

Solution	**Label**	**Number of Pump Switching**	**Reservoir Level (ft)**	**Chlorine Residual (mgL)**
Highest	13	84	2.1	**2.4**
Median	14	119	1.7	**1.6**
Lowest	15	89	2.9	**1.1**

6.3.2.2 Discussion of Results

The results showed the feasibility of using the APC-MOPSO framework for the optimization of system-operating problem. The main aspects needed to implement an APC-MOPSO algorithm, EPANET, SFG, and SBOP for the Saskatoon West WDS, namely the representation of solutions as operating-modes, the reduction in electrical energy consumption, the improvement of network reliability, the enhancement of water quality, and the new Operating-Mode Pointer (OMP) used by APC-MOPSO to generate new solutions. An extensive analysis of the solutions obtained by the energy optimization strategy is carried out, examining, and comparing in detail the outcome of each solution with regard to the cost function of the problem (i.e., five objectives). By means of statistical criterion, the highest, median, and lowest configuration of values pertaining to each objective are selected as solutions in this analysis. One conclusion of this analysis is that the five objectives are fundamentally different, a compromise is therefore sought which aimed to represent the solution that has the best configuration values on one of them has little resemblance with the best values for the others. Hence, from the fifteen solutions, solutions 4, 5, and 3 are chosen to investigate the performance of the energy optimization strategy.

The final part of this discussion is to compare the solutions 4, 5, and 3 with their corresponding values of electrical energy cost ($), number of pump switching, cost of maximum power peak ($), reservoir level (ft), and chlorine concentration (mgL). Table 6.8 shows that solution 4 produced lower reservoir level (1.2 ft) and chlorine concentration (1.1 mgL) (i.e., this considers better that the optimization problem is designed as a minimization of all objective). Whereas there is no significant difference between using solution 4 or the others in terms of

energy cost. Therefore, it seems more beneficial to prefer solution 4 for the best result. For the rest of this section, the analysis is mainly made on solution 4.

Table 6.8: Summary of Experimental Results

Solution	Savings in Electricity Cost (%)	Savings in Max. Power Peak Cost (%)	Improvement in Pumping Reliability (%)
4	10.19	4.42	3.57
5	8.37	4.98	4.56
3	11.73	4.17	3.84
Solution	**Reservoirs Level Variation (ft)***	**Chlorine Concentration Range (mg/L)****	**Total Pump Switches Range*****
4	1.2	1.1	143
5	-1.8	1.9	95
3	-2.7	1.4	131

* *Small variations do not make solutions unacceptable* [200]

** *The range of chlorine concentration at demand nodes is set to [1.1 mg/L –to- 2.47 mg/L]*

*** *The total allowable pumps switches considered in this work is equal to 420 times*

To ensure the hydraulic periodicity of the Saskatoon West WDS, the reservoirs must finish their optimization period at the same level from which they started. In this work, the differences between initial and final levels in all reservoirs are determined and shown in Figures 6.13 to 6.16. It should be noted that the reservoir level that is largely affected is that of Agrium (see Figure 6.14) due to its large storage capacity. Also, it is important to mention that the capacity constraints of this reservoir (and other reservoirs in the network) are not exceeded. For all reservoirs, the initial levels are restored at the end of the optimization period of time.

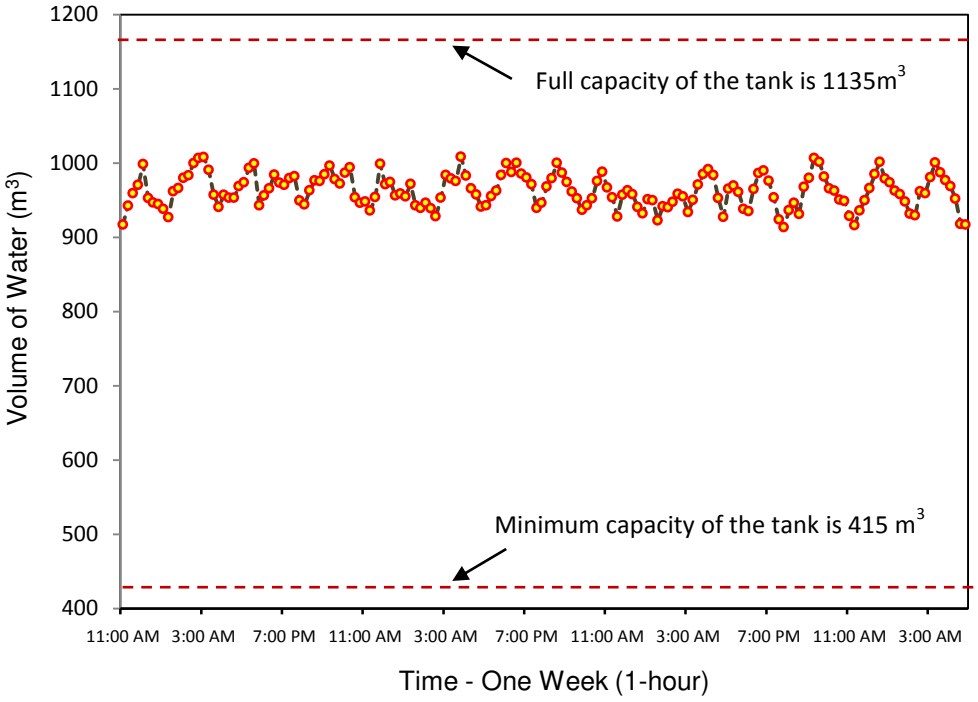


Figure 6.13: Optimized Reservoir Level for PCS Cory (One-Week)

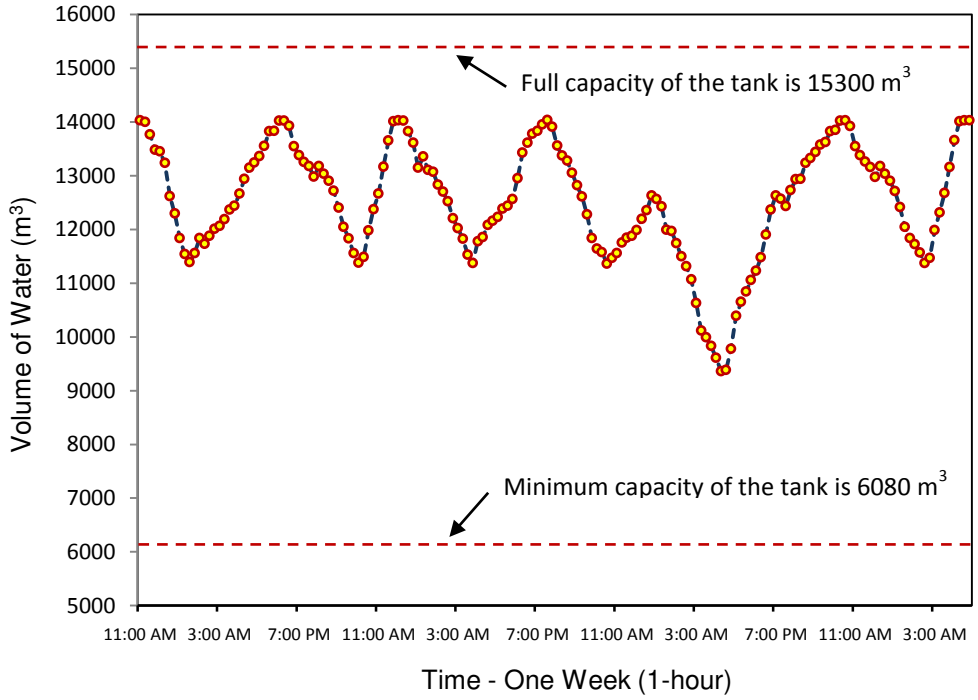


Figure 6.14: Optimized Reservoir Level for Agrium (One-Week)

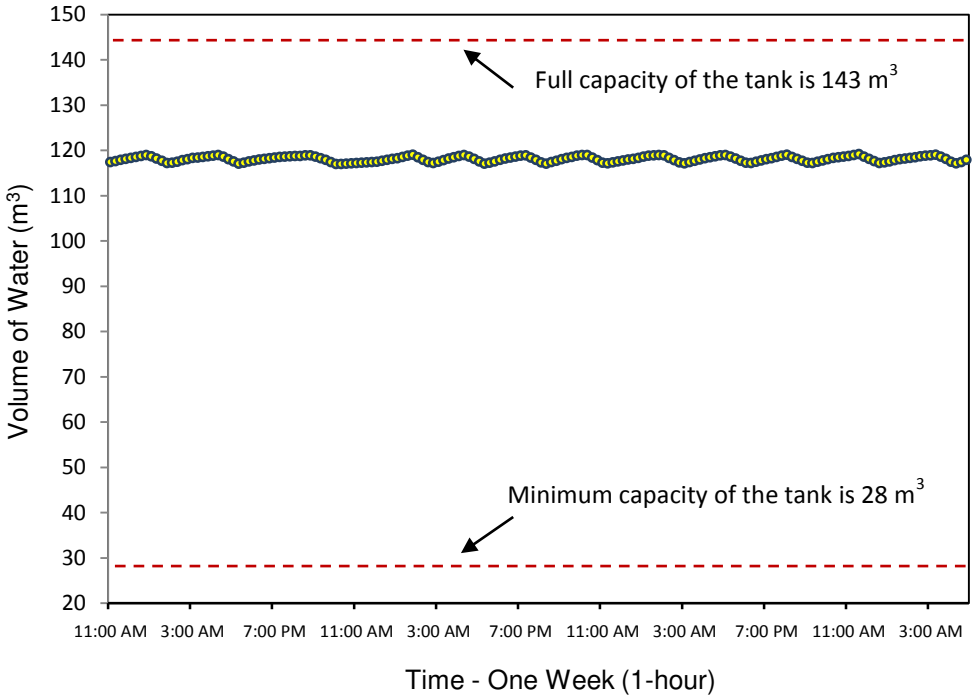


Figure 6.15: Optimized Reservoir Level for Corman Park (One-Week)

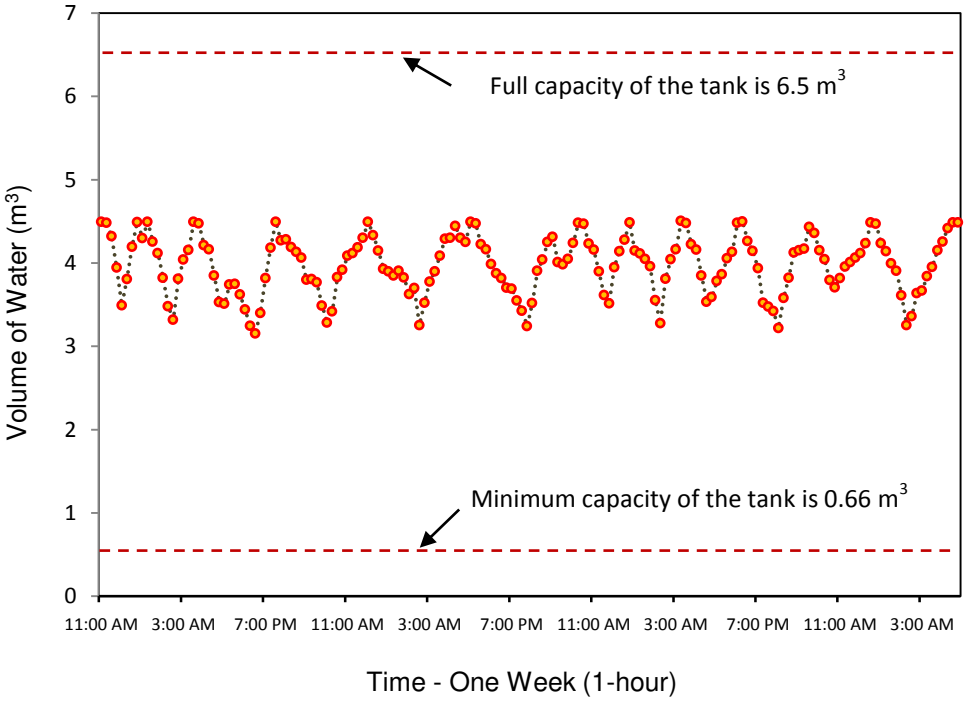
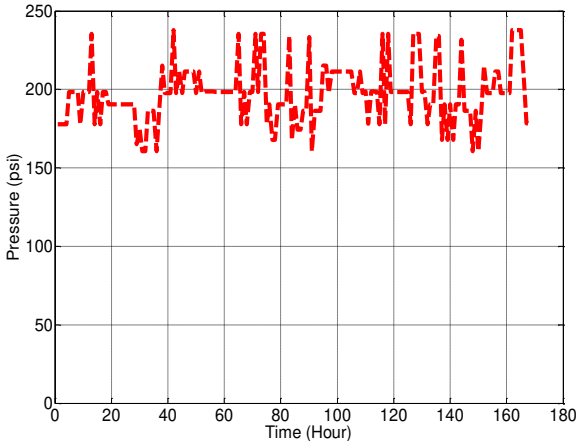
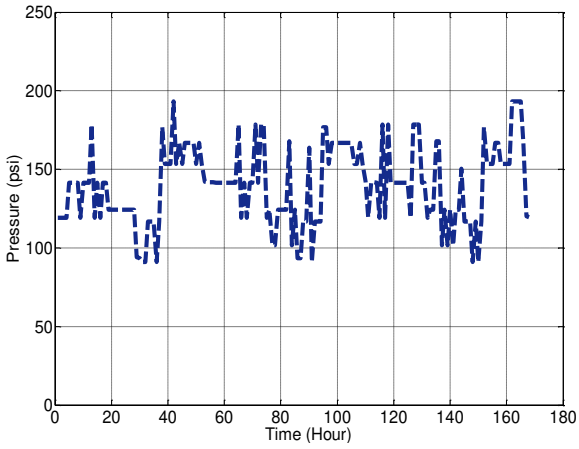


Figure 6.16: Optimized Reservoir Level for Vanscoy Villa (One-Week)

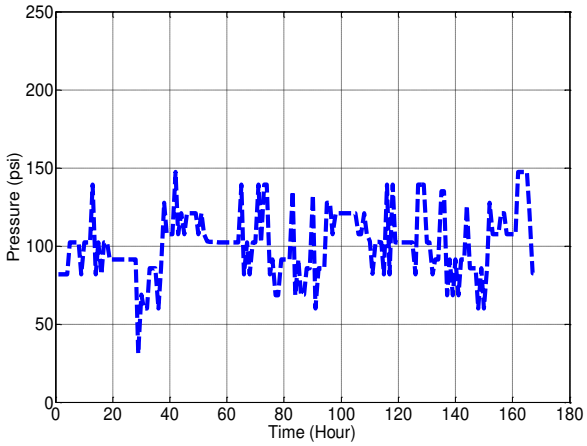
The corresponding pump discharge, node pressure, pump scheduling, and operating-modes for the optimized solutions are tabulated and provided in Appendix C. The pressure and flow rate at demand nodes and pumping stations are shown in Figures 6.17 and 6.18. It should be noted that under the optimized pump schedules, both the consumption profile of the customers as well as the reservoirs constraints are not violated. Furthermore, for all solutions obtained by APC-MOPSO, EPANET simulations confirm that the pressure constraints are satisfied.



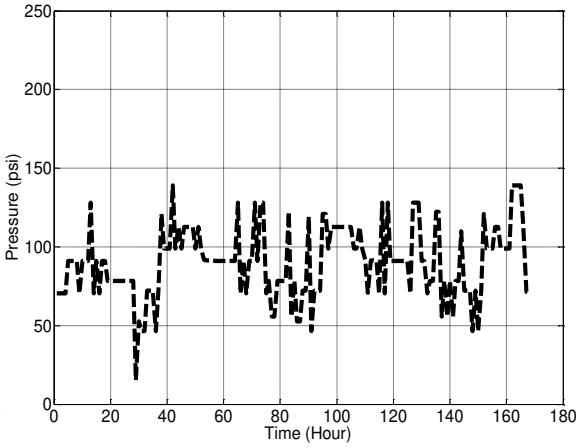
a. Pressure at QE Pumping Station



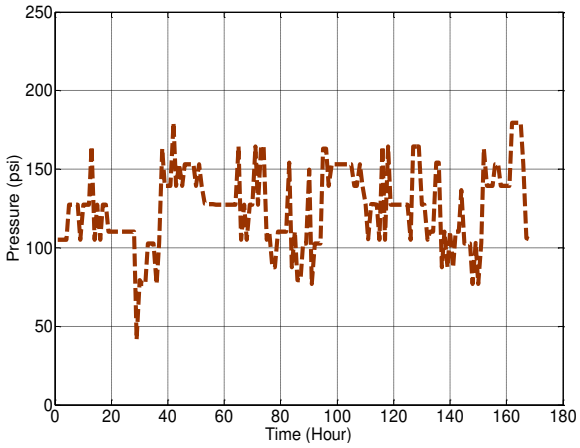
b. Pressure at Aurora Pumping Station



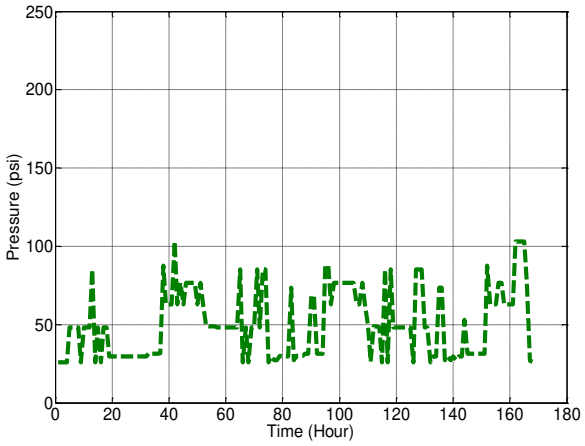
c. Node Pressure at Corman Park



d. Node Pressure at PCS

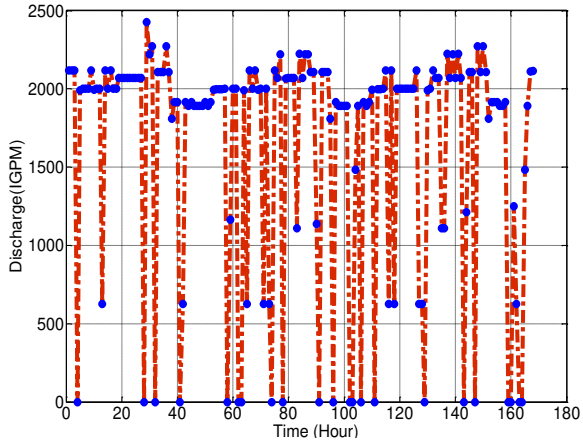


e. Node Pressure at Villa of Vanscoy

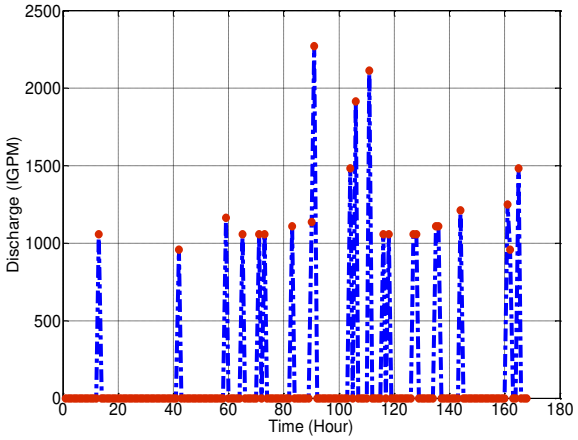


f. Node Pressure at Agrium

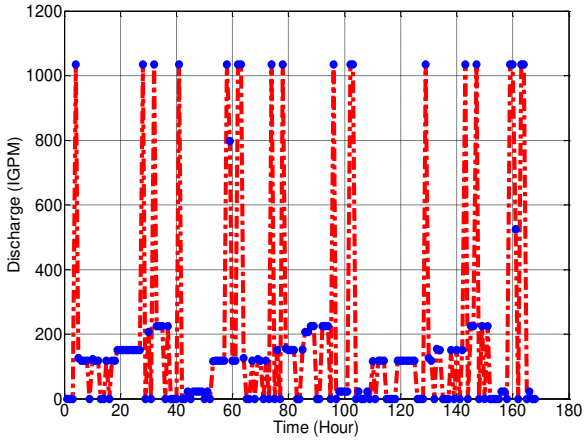
Figure 6.17: Corresponding Nodes Pressures for the Optimized Operating-Modes



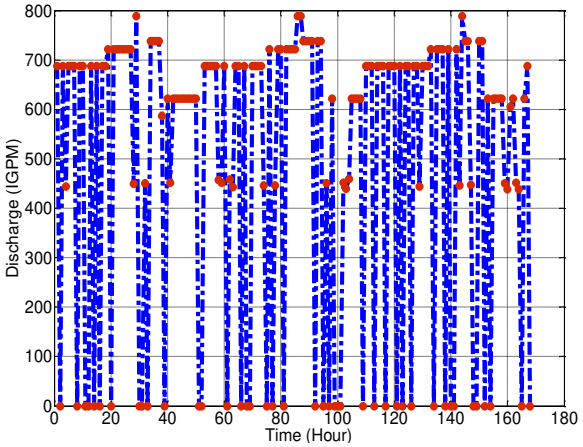
a. Discharge at Pump 1 at QE Station



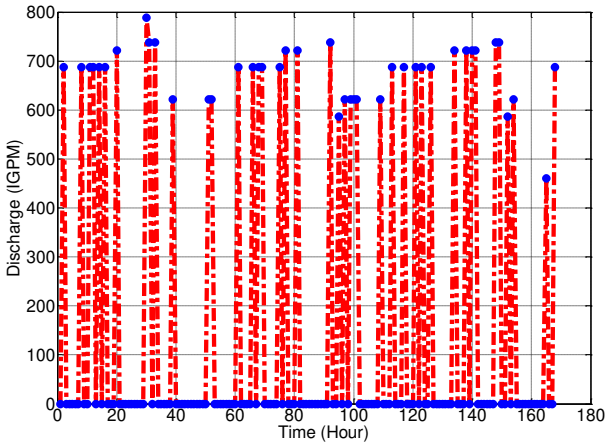
b. Discharge at Pump 2 at QE Station



c. Discharge at Pump 3 at QE Station



d. Discharge at Pump 1 at aurora Station



e. Discharge at Pump 2 at Aurora Station

Figure 6.18: Corresponding Pumps Discharge for the Optimized Operating-Modes

Changing the demand profile of the customers is considered as one of the cost-effective factor that reduces the energy costs. The revised demand profiles of the five major customers are illustrated in Figures 6.19 to 6.22.

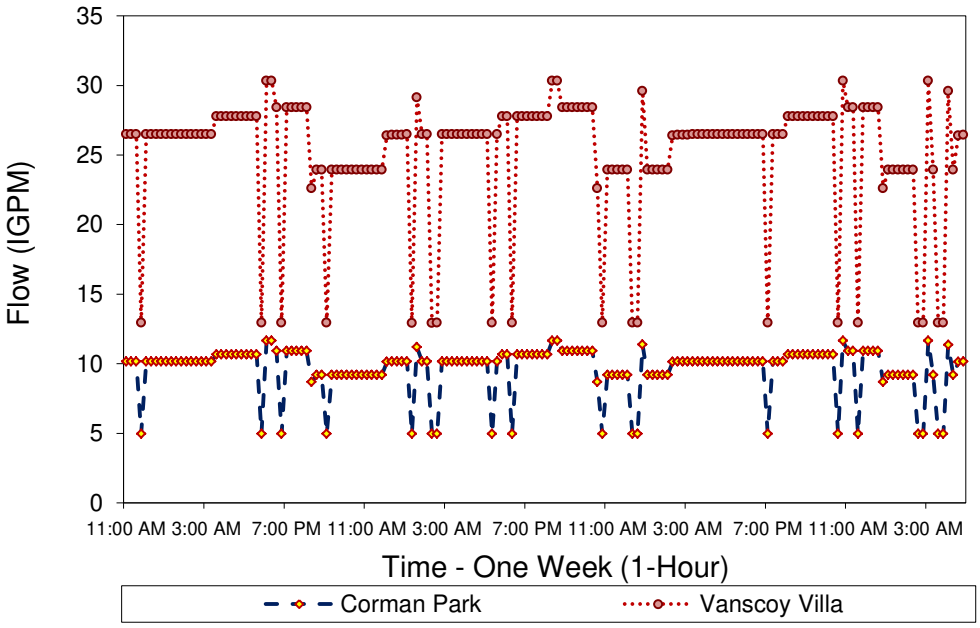


Figure 6.19: Revised Demand Profiles of Corman Park and Vanscoy Villa (One-Week)

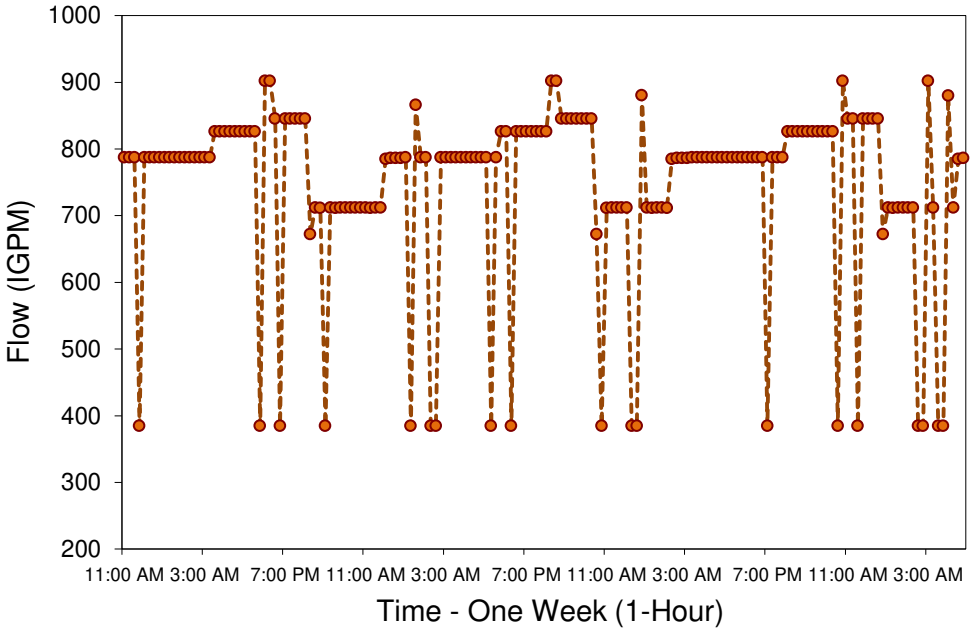


Figure 6.20: Revised Demand Profile of PCS Cogen (One-Week)

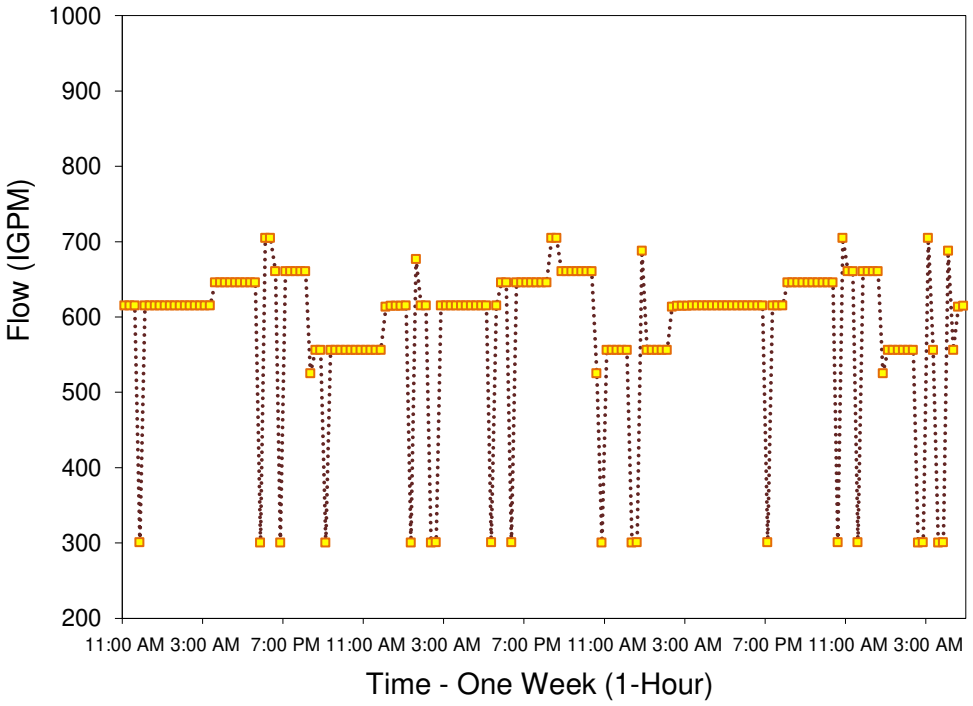


Figure 6.21: Revised Demand Profile of Agrium (One-Week)

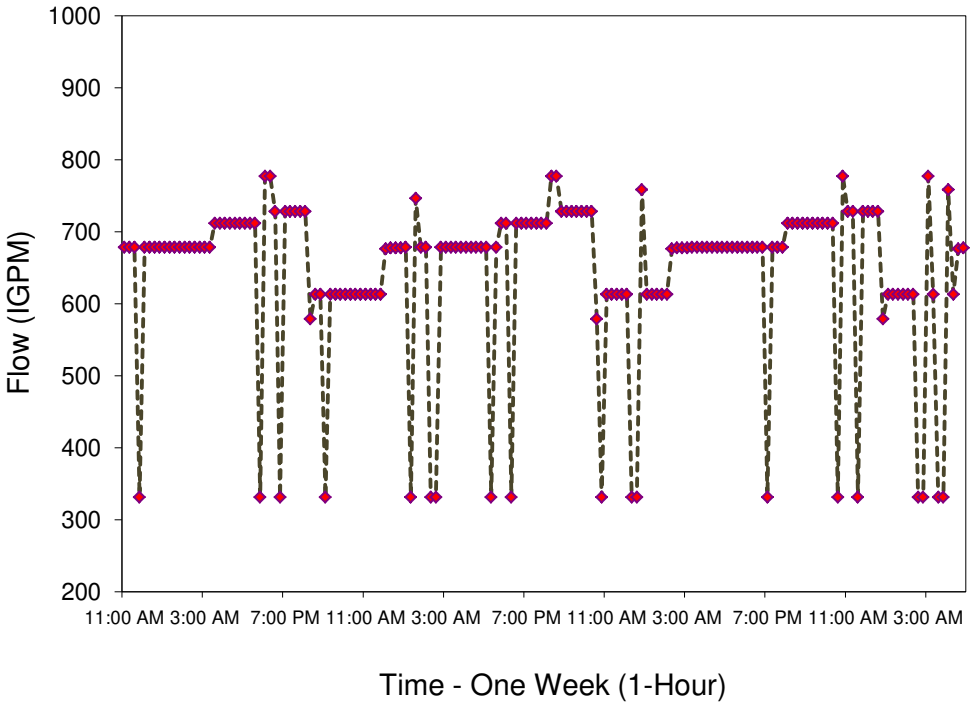
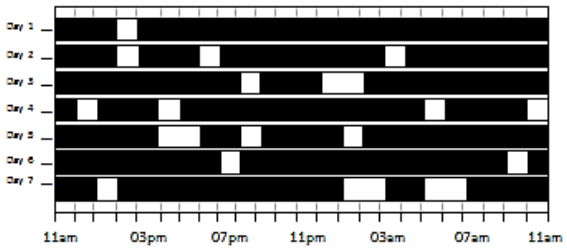
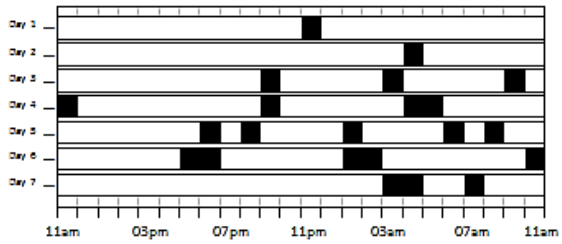


Figure 6.22: Revised Demand Profile of PCS Cory (One-Week)

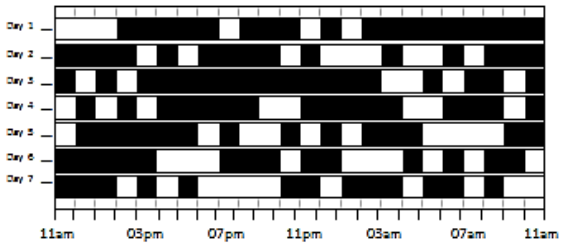
Figure 6.23 shows the pump schedules obtained from solution 3 for the Saskatoon West WDs using a constraint of 420 pump switches.



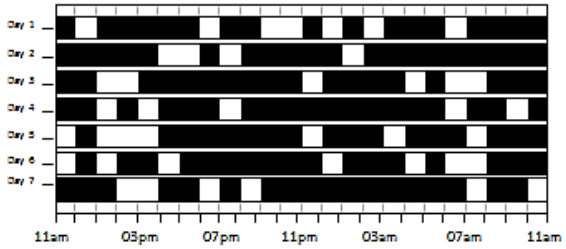
a. Duty Pump 1 at QE Pumping Station



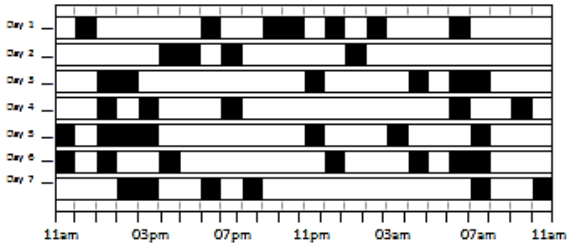
b. Duty Pump 2 at QE Pumping Station



c. Jockey Pump at QE Pumping Station



d. Duty Pump at Aurora Pumping Station



: Pump is OFF

: Pump is ON

- Pump Schedules extracting from Solution 1

e. Standby Pump at Aurora Pumping Station

Figure 6.23: Pump Schedules for the Saskatoon West WDS – Solution 4

To illustrate the advantages of implementing Selecting-Best-Operating Point (SBOP) technique, Table 6.9 and Figure 6.24 together show the number of modified operating-modes for the fifteen solutions specified in this case, and the changes that take place on the discharge profile for the two duty pumps at QE. Figure 6.24 shows the changes in the pumps discharge over the optimization period obtained by solution 4 (e.g., pumps 1 and 2 at the QE pumping station).

Table 6.9: Implementing SBOP Method on the Selected Fifteen Solutions

Solution	Number of Operating-Modes	Number of Modified Operating-Modes	Improvement %
1	104	4	3.84
2	115	6	5.22
3	123	13	10.57
4	112	5	3.57
5	132	6	4.56
6	112	13	11.6
7	131	4	3.05
8	121	6	4.96
9	104	13	12.5
10	101	4	3.96
11	115	6	5.22
12	108	13	12.04
13	121	7	5.79
14	111	8	7.21
15	117	5	4.27

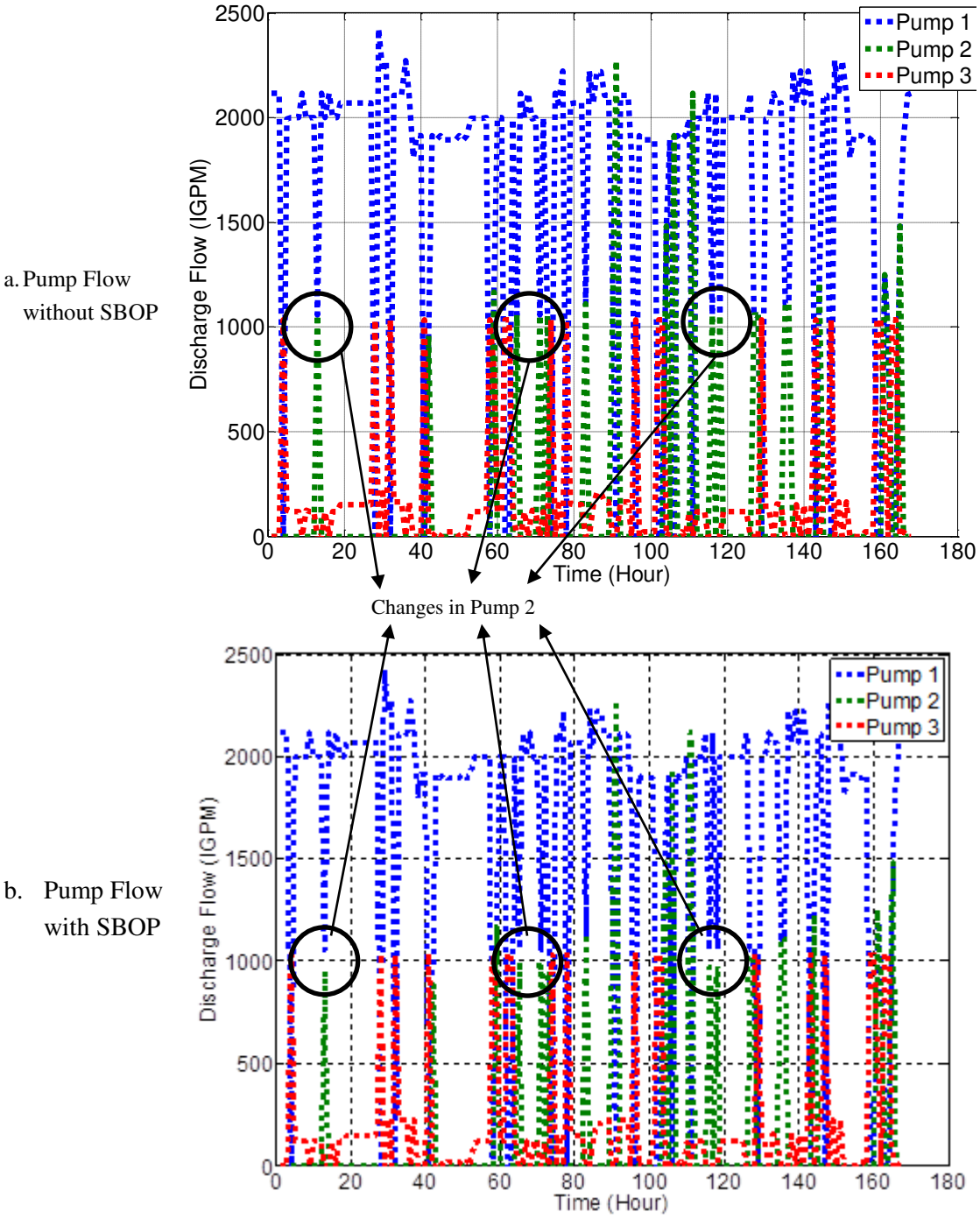


Figure 6.24: Optimized Discharge Rate at QE Pumping Station - Solution 4

One of the consequences of adopting the OMP and SBOP to the APC-MOPSO framework rather than creating new proven operating-modes is that the two approaches can be straightforwardly extended to include more complex and modern water distribution systems. The

point made in this chapter is that the application of the energy optimization strategy has potential savings with the goal of further improving the safety and reliability of the operating-modes.

6.4 Conclusion

In 1980s and 1990s, most pump-scheduling models only considered the objectives of minimizing the energy-consumption cost and the maintenance cost, [192,187,291]. While recent optimization methodologies consider goals other than the aforementioned objectives for pump-scheduling problems, including network reliability, cost of maximum power peak, system rehabilitation[30], water quality, and tank capacity and operation, [200,205,292,293,190,294]. In this work, the system-operational problem is formulated using five objectives that are: the electrical energy cost, the cost of maximum power peak, the total number of pump switches, network reliability, and water quality. Moreover, the hydraulic analysis of the Saskatoon West WDS for the demand pattern and pump combinations is performed using EPANET. The APC-MOPSO algorithm is then used to search for approximately optimal solutions. It is implemented by using a combination of MATLAB and C++ codes.

Due to the problem of the high dimensionality, the proposed energy optimization strategy is implemented for an optimization time period of one week (168 hours). However, the proposed strategy can be applied to a longer optimization.

Selecting-Best-Operating Point (SBOP) is used to correct inefficiencies in pump operation and the necessary calculations to compute the *head-discharge-speed-efficiency* relationships. The results of these calculations showed that variable-speed pumps operate efficiently over the full range of speeds. It should be noted that the changes in head and discharge cause minor degradation in the optimality of the solutions obtained by the energy optimization.

The results of the compromised best three solutions show potential for considerable cost reductions in electricity and maximum power peak costs (see Table 6.8). The implementation of the APC-MOPSO, SFG, SBOP, and a novel Operating-Mode Pointer (OMP) on the Saskatoon West WDS improve its total operational costs by using the pump schedules.

Finally, using the potential savings for the month of March 2011, the projected reduction in the total energy cost in 2011 is as follows: (using the savings obtained by solution 4)

[30] System rehabilitation is defined as the application of repairing, renewing, and replacing the infrastructure (e.g., pipes, valves, etc.) of water distribution systems and wastewater collection systems to return its functionality.

Table 6.10: Projected Savings for the Total Energy Cost for the Saskatoon West WDS in 2011

Annual Total Energy Cost ($)			Projected Saving ($)		Monthly Cost ($)		Projected Monthly Saving ($)
Month	Electrical Energy	Max. Power Peak	Electrical Energy (10.19%)	Max. Power Peak (4.42%)	Actual Cost	Projected Cost	
JAN	16,817	6,589	1714	291	27,142	24,739	2,403
FEB	17,602	6,212	1794	275	27,614	25,136	2,478
MAR	14,339	6,259	1461	277	23,899	21,815	2,084
APR	17,785	5,680	1812	251	27,210	24,740	2,471
MAY	14,408	5,772	1468	255	23,416	21,350	2,066
JUN	16,438	6,463	1675	286	26,559	24,209	2,350
JUL	15,327	5,999	1562	265	24,740	22,550	2,190
AUG	14,661	5,767	1494	255	23,703	21,606	2,096
SEP*	13,150	9,245	1340	409	25,975	23,870	2,104
OCT*	13,551	8,781	1381	388	25,902	23,775	2,127
NOV*	17,388	5,767	1772	255	26,852	24,425	2,427
DEC*	17,005	5,535	1733	245	26,142	23,774	2,368
Total	188,470	78,068	19,205	3,451	309,154	**281,989**	**27,166**

* *Note that the electricity bills for the months of September, October, November, and December are not available and have been estimated using the bills of the previous year with a reasonable adjustments.*

Chapter 7

Summary, Conclusions & Future Considerations

This dissertation is concerned with the development of a practical and efficient energy optimization strategy for rural water distribution systems. The work can be divided into six main parts: *first*, the fundamentals of water distribution systems; *second*, the formulation of system-operation as a constrained multi-objective optimization problem; *third*, the development of the APC-MOPSO approach and its application to well-known benchmark test problems; *fourth*, the development of auxiliary techniques that are integral to the APC-MOPSO algorithm, including operating-mode pointer, searching-for-gaps, and selecting-best-operating point; *fifth*, the application of energy optimization strategy to the Saskatoon West WDS; and *sixth*, the exhaustive experimental analysis and comparison of collected results. A conclusion to these parts are summarized as follows.

7.1 Fundamentals of Water Distribution Systems

Chapter 3 starts by giving an overview of the main elements in water distribution systems, defines the nature, functions, and purposes of water distribution system modeling. This chapter is structured to explain the fundamental of modeling and simulating of water distribution systems.

Saskatoon West Water Distribution System is a rural water distribution system used for the proposed energy optimization strategy. This system has all the typical features of a water distribution system, yet is small enough to provide a tangible test case.

7.2 Formulation of System-Operational Problem

The formulation of a system-operational problem is explained in Chapter 3. The formulation of the objective functions and the constraints are presented. The cost function includes the consideration of electrical energy cost, total number of pump switches (i.e., maintenance cost), cost of maximum power peak (i.e., demand charge), network reliability (i.e., reservoir water level variation), and water quality (i.e., free chlorine residual). Chapter 3 also

summarized improvements to the hydraulic simulator, EPANET, for its use in optimization within a MATLAB framework.

7.3 Development of APC-MOPSO Algorithm

A new Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) algorithm is proposed in Chapter 4. The proposed algorithm is distinguished from today's MOEAs and MOPSOs by having a set of new features, including a dynamic model to update the position and velocity of flying particles, an adaptive mutation operator, an adaptive variable-size of repository, an adaptive mutation model, a migration process to exchange the good particles amongst the sub-swarms, an adaptive search space boundary, and adaptive social, cognitive, contiguous, position, and velocity factors. In addition to these new features, a parallel computing process is proposed to perform concurrent searches using several subgroupings of the population (i.e., sub-swarms) to obtain non-dominated solutions to broadly cover the search space. The proposed APC-MOPSO is then tested to verify its performance using fifteen well-known benchmark problems and four different performance metrics.

7.4 Development of Auxiliary Techniques

Searching-for-Gaps (SFG) is a technique proposed in Chapter 5 to improve the performance of the APC-MOPSO when solving multi-objective real-world problems. This is achieved by incorporating a new adaptive grid scheme as well as a secondary external repository. Accordingly, the search space is divided into hyper-cubes and non-dominated solutions in each cube are counted. Then, the secondary repository is used to archive the neighborhood best solutions that APC-MOPSO finds at each iteration which in turn used to cover the entire global fronts.

Based on the previous work proposed by Habibi [295], a new representation of pump scheduling, referred to as the "operating-mode", is developed in Chapter 5. This representation is based on proven pumping operations that are extracted from the network daily supply and consumption profiles. The operating-mode representation explicitly contains system settings, loading conditions, and flow demands. The operating-modes are scheduled and combined to represent system operations in this case over one week of an optimization period of time.

The Selecting-Best-Operating Point (SBOP) is also proposed. In this technique, a new reliability curve is introduced to assess the reliability of the pump characteristics in terms of efficiency, head pressure, and flow rate. Two processes are suggested to improve the reliability, namely speed-up and speed-down. In the first process, the speed of the pump is increased to move the current operating point of a pump close to its best efficiency point (BEP). Conversely, in the latter process, the speed of the pump is reduced so that the required head pressure and flow rate of the operating point become close to those of the BEP.

7.5 Application of Energy Optimization Strategy to the Saskatoon West WDS

The implementation of the new energy optimization strategy to the Saskatoon West WDS is provided in Chapter 6. The APC-MOPSO algorithm with SFG and EPANET is used to optimize the system-operating problem that is formulated and presented in Chapter 3. For this problem, five objectives are considered including, electrical energy consumption, total number of pump switches, network reliability, maximum power peak, and water quality. Fifteen optimized solutions are selected and analyzed in which the savings in electrical cost, number of pump switching, reservoir level, cost of maximum power peak, and chlorine concentration are tabulated and provided. The criterion of selecting these solutions is done based on the highest, median, and lowest values returned by the optimized solution with respect to each objective.

7.6 Experimental Analysis and Comparison

7.6.1 Experiments on APC-MOPSO Algorithm

To identify the robustness, accuracy, and flexibility of the proposed APC-MOPSO algorithm, three different sets of experiments are performed using fifteen well-known benchmark test problems and four different performance metrics. The first set of experiments is held using two-objective test problems, a second set used three-objective test problems, while a third set considered five-objective test problems. An exhaustive experimental analysis is then carried out to evaluate and compare the results obtained by APC-MOPSO with other state-of-the-art algorithms in the heuristic literature.

The optimization of the two-objective test problems is empirically studied to assess the performance of the proposed APC-MOPSO. Selected test problems are chosen from different perspectives, including continuous, discontinuous, discrete, convex, concave, unimodal, and multi-modal. The results showed that non-dominated solutions obtained by APC-MOPSO against ZDTs, DTLZs, OKA2, KURSAWE, and FONSCA typically have the best average values of Two-sets Covering (C), Error Ratio (ER), Spacing (S), and Generational Distance (GD) metrics. It is also concluded that the adaptive techniques, K-Means clustering tool, parallel structure of search process, new dynamic (i.e., flying) model have a great effect on the obtained solutions in terms of converging and covering the Pareto-optimal front. For the same problems, five different MOEAs and MOPSOs are tested for comparison. In most experiments, APC-MOPSO generated better solutions than those of the other MOEAs and MOPSOs. Only in for the DTLZ2 test problem, APC-MOPSO is ranked second in the ER, GD, and S metrics. Nonetheless, with the use of an auxiliary technique referred to as Searching-for-Gaps (SFG), APC-MOPSO produced better results on all benchmark test problems.

With regard to the three-objective benchmark test problems, the proposed APC-MOPSO produced the best solutions in terms of returning the best metrics' values and having the best coverage of the Pareto-optimal front. Optimizing with three or more objectives increasingly requires longer computational time due to the exponential growth of the search space.

Next, the APC-MOPSO algorithm is applied to the five-objective benchmark test problems. Minimizing five-objective DTLZ2and DTLZ5 problems is considered. The results of the APC-MOPSO approach outperformed all others in terms of the quality of the Pareto-front set and the computation time.

The overall conclusion from the above experiments indicate that APC-MOPSO is a flexible search approach that can easily be adapted to most of real-world optimization problems, typically producing high quality non-dominated solutions with reduced computational time.

7.6.2 Experiments on Computation Effort

The computation time of the proposed APC-MOPSO approach is discussed in Chapter 4. The results from the benchmark test problems indicate that the computation time (i.e., CPU time) of the APC-MOPSO algorithm is significantly lower than when compared with the time spent by MOPSOs and some of MOEAs (e.g., NSGA-II) considered as state-of-the-art

7.7 Contributions of this Thesis

The main contributions of this thesis are summarized as follows:

- A new formulation for a system-operational problem is developed by simultaneously minimizing the electrical energy cost, the total number of pump switches, the reservoir water level variation, the cost of maximum power peak, and the free chlorine residual.
- A new Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (ACP-M OPSO) Algorithm is developed.
- A new Searching-for-Gaps technique to strengthen the performance of the APC-MOPSO is proposed.
- The proposed APC-MOPSO algorithm is linked to a new modified version of a hydraulic simulator (EPANET) for WDS optimization.
- The effect of the constraint on the total number of pump switches is empirically investigated in the Saskatoon West WDS. The results suggest that schedules with low electrical cost typically have a moderate number of pump switches. This observation contradicts the intuitive notion that a higher number of pump switches would provide a lower cost.
- The application of APC-MOPSO for solving well-known benchmark multi-objective test problems demonstrated its ability to outperform other state-of-the-art algorithms proposed in the literature such as NSGA-II [233], 2LBMOPSO [243], SPEA2 [110,74], PEGA [253], and original MOPSO [52].

- The new Searching-for-Gaps (SFG) technique effectively enhanced the search properties of the APC-MOPSO algorithm by guaranteeing uniform and even distribution along the Pareto-optimal front. It also is found to preserve diversity.
- The APC-MOPSO algorithm is applied to rural WDS, specifically Saskatoon West WDS by using the Operating-Mode Pointer (OMP).
- A modified version of EPANET is developed to simulate WDS within a MATLAB platform and to interface with APC-MOPSO to accept and evaluate new pump curves at each optimization iteration.
- The APC-MOPSO algorithm is implemented using a parallel search model (i.e., the concept of parallel islands).
- Speed-up and speed-down are two post-processes applied to the solutions obtained by a combination of APC-MOPSO and EPANET. The results showed a considerable improvement in pumping characteristics (e.g., efficiency, head pressure, and flow rate) for all solutions, regardless of the total number of pump switches or the problem formulation.
- The proposed energy optimization strategy has benefits other than saving energy costs, including improving system safety (i.e., all the hydraulic and operational constraints are satisfied and all new solutions (operating-modes) have been already validated by the operators of the Saskatoon West WDS).

Significant average reductions of 10.19% in the electrical energy cost and 4.42% in the cost of maximum power peak are achieved when the new energy optimization strategy is applied to the Saskatoon West WDS.

Currently, in Canada, water is highly subsidized (water is priced at one-tenth of the actual cost, [296]). Hence, it is imperative to reduce the costs involved in operating / maintaining water distribution systems. A considerable saving of 10% or CAD$27,000 per annum in total operational cost is achieved by implementing a new energy optimization strategy on the Saskatoon West WDS. The new Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization (APC-MOPSO) with Searching-for-Gaps (SFG) is developed to find feasible solutions that fulfill the desired objectives as well as the system and operating constraints. To overcome the disadvantages of today's encodings, which clearly limit their functionality to represent multidisciplinary systems and systems with large variables, a novel Operating-Mode Pointer (OMP) technique is proposed. Within the Operating-Mode-Pointer (OMP), the operational safety, reliability, and ability to meet customer demands are guaranteed for the Saskatoon West WDS. The new operating scenarios obtained by the proposed energy optimization strategy are made up of existing operating-modes and chosen only after extensive practical verification. For a real-time implementation, the proposed energy optimization strategy can be interfaced to the control system that is used for operating the system. The success of the work presented in this dissertation can lead to a new technology that actively manages and minimizes the energy consumption in water distribution systems. This work would also allow a

way of practically implementing a more efficient pumping operation in a safe and controlled manner.

7.8 Future Directions

The following ideas and research trends deserve further considerations:

- In the dissertation, the steady-state (SS) and the extended-period (EPS) hydraulic simulations are used. The hydraulic simulation should be extended to include the transient behavior of the network as well as unusual operating scenarios (e.g., fire, burst damages).
- Since the localization problems involve high strain gradients, adaptive meshing techniques are encouraged to be used.
- New applications of the proposed optimization strategy should be considered, namely in power plant, manufacturing, and wastewater treatment facilities.
- Applications of energy optimization should be extended to include system-design parameters such as pipeline sizes, reservoir geometries, valve locations, and demand profiles.
- Characterization and modeling of the pump operating point and energy consumption is required particularly for the problem of pumping operation in water distribution systems.
- Further investigations are required to include water age and chlorine residual in the network model when solving system-operational problem.

Appendix A

Derivation of Equation (4.5) for C_p and C_v Coefficients

The proposed dynamic model to update the positions of the particles in the decision space (i.e., known also variable space) is given as follows:

$$x_{t+1} = C_p\, x_t + C_v\, (x_t + V_{t+1}) \quad (1)$$

where

$$C_p + C_v = 1 \quad (2)$$

The condition of Eq. (2) effectively results in the implementation of a low-pass filter as follows. For example, the transfer function of a first-order low-pass filter is:

$$\frac{Output}{Input} = H(s) = \frac{b}{\tau s + 1} \quad (3)$$

where s is the Laplace variable, τ is the time constant, and b is the DC gain of the digital filter.

Applying the z-transformation to Eq. (3) provides a discrete representation as:

$$\frac{V_f(z)}{V(z)} = H(z) = \frac{b}{1 - z^{-1}\, e^{-T/\tau}} \quad (4)$$

where V_f represents the filtered variable (i.e., output signal), V is the original variable (i.e., input signal), and T is the sampling period.

In the time domain, the above can be implemented as:

$$V_{f\,n} = b\, V_n + e^{-T/\tau}\, V_{f\,n-1} \quad (5)$$

If $e^{-T/\tau}$ is replaced with k, solve Eq. (4) to get unity gain, then the difference equation can be simplified to

$$V_{f\,n} = (1 - k)\, V_n + k\, V_{f\,n-1} \quad (6)$$

Then, the above equation can be written as:

$$V_{f\,n+1} = (1-k)\,V_{n+1} + k\,V_{f\,n} \quad (7)$$

Generally, the velocity is defined as the change in position over a period of time, which is expressed as follows:

$$V_{n+1} = \frac{x_{n+1} - x_n}{\Delta T} \quad (8)$$

Substitute V_{n+1} from Eq. (8) into Eq. (7), yields:

$$V_{f\,n+1} = (1-k)\left(\frac{x_{n+1} - x_n}{\Delta T}\right) + k\,V_{f\,n}$$

This can be simplified to:

$$x_{n+1} = x_n + \frac{\Delta T}{(1-k)}\left[V_{f\,n+1} - k\,V_{f\,n}\right]$$

$$x_{n+1} = x_n + \frac{\Delta T}{(1-k)}\,(1-k)\,V_{n+1}$$

$$x_{n+1} = x_n + \Delta T(\,V_{n+1}) \quad (9)$$

According to Eq. (1), the proposed dynamic model can be re-written as:

$$x_{n+1} = \left(C_p + C_v\right)\,x_n + C_v\,(V_{n+1}) \quad (10)$$

Comparing Eqs. (9) and (10), then:

$$C_p + C_v = 1 \quad (11)$$

$$C_v = \Delta T \quad (12)$$

Appendix B

Pipe Flow and Node Pressures for the Existing Operating-Modes

The Operating-Mode Pointer (OMP) is used to analyze and extract the existing operating-modes of the Saskatoon West WDS over the three-month period from March to May 2011. The OMP is applied by taking snap-shots (i.e., hourly basis) of the system, referred to as operating-modes. Each mode contains the status of all network components, including the pump discharge, pump combination, pump scheduling, and demands of the network's clients. A total of 297 operating-modes are extracted and provided as follows.

Table B.1: Pump Discharge and Node Demands

Mode	Flow	Pump	Pump	Pump	Agrium	Vanscoy	Cory	Cogen	Corman
	(IGPM)	1	2	3	(IGPM)	(IGPM)	(IGPM)	(IGPM)	(IGPM)
1	2419	1	1	0	702.72	30.24	775.05	899.38	11.61
2	2525	1	1	0	733.51	31.56	809.01	938.80	12.12
3	2647	1	1	0	768.95	33.09	848.10	984.15	12.71
4	1563	0	1	0	454.05	19.54	500.79	581.12	7.50
5	1371	0	1	0	398.28	17.14	439.27	509.74	6.58
6	2652	1	1	0	770.41	33.15	849.70	986.01	12.73
7	2494	0	1	0	724.51	31.18	799.08	927.27	11.97
8	2721	0	1	1	790.45	34.01	871.81	1011.6	13.06
9	2486	0	1	1	722.18	31.08	796.51	924.29	11.93
10	2637	0	1	1	766.05	32.96	844.89	980.44	12.66
11	2492	0	1	1	723.93	31.15	798.44	926.53	11.96
12	1774	0	1	1	515.35	22.18	568.39	659.57	8.52
13	1854	0	1	0	538.59	23.18	594.02	689.32	8.90
14	1855	0	1	1	538.88	23.19	594.34	689.69	8.90

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
15	1856	0	1	0	539.17	23.20	594.66	690.06	8.91
16	2517	0	1	1	731.19	31.46	806.45	935.82	12.08
17	2498	0	1	1	725.67	31.23	800.36	928.76	11.99
18	2562	0	1	1	744.26	32.03	820.86	952.55	12.30
19	2381	0	1	1	691.68	29.76	762.87	885.26	11.43
20	1337	0	1	0	388.40	16.71	428.37	497.10	6.42
21	2473	0	1	1	718.41	30.91	792.35	919.46	11.87
22	2390	0	1	1	694.30	29.88	765.76	888.60	11.47
23	2604	0	1	1	756.46	32.55	834.32	968.17	12.50
24	2657	0	1	1	771.86	33.21	851.30	987.87	12.75
25	2289	0	1	1	664.95	28.61	733.40	851.05	10.99
26	1811	0	1	1	526.10	22.64	580.24	673.33	8.69
27	1810	0	1	0	525.81	22.63	579.92	672.96	8.69
28	2072	0	1	0	601.92	25.90	663.87	770.37	9.95
29	1966	0	1	0	571.12	24.58	629.91	730.96	9.44
30	2515	0	1	1	730.61	31.44	805.81	935.08	12.07
31	1543	0	1	0	448.24	19.29	494.38	573.69	7.41
32	1953	0	1	0	567.35	24.41	625.74	726.13	9.37
33	2528	0	1	1	734.38	31.60	809.97	939.91	12.13
34	2246	0	1	1	652.46	28.08	719.62	835.06	10.78
35	1544	0	1	1	448.53	19.30	494.70	574.06	7.41
36	1544	0	1	1	448.53	19.30	494.70	574.06	7.41
37	2335	0	1	0	678.32	29.19	748.13	868.15	11.21
38	2337	0	1	1	678.90	29.21	748.77	868.90	11.22
39	1457	0	1	1	423.26	18.21	466.82	541.71	6.99
40	1558	0	1	0	452.60	19.48	499.18	579.26	7.48
41	1122	0	1	0	325.94	14.03	359.49	417.16	5.39
42	2034	0	1	0	590.88	25.43	651.69	756.24	9.76
43	2346	0	1	1	681.51	29.33	751.66	872.24	11.26
44	2348	1	1	0	682.09	29.35	752.30	872.99	11.27
45	2368	1	1	0	687.90	29.60	758.71	880.42	11.37
46	1848	1	1	0	536.84	23.10	592.10	687.09	8.87
47	1245	0	1	0	361.67	15.56	398.90	462.89	5.98
48	2221	1	0	0	645.20	27.76	711.61	825.77	10.66
49	2535	1	0	0	736.42	31.69	812.21	942.51	12.17
50	2539	1	0	0	737.58	31.74	813.50	944.00	12.19

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
51	2350	1	1	0	682.68	29.38	752.94	873.73	11.28
52	2127	1	1	0	617.89	26.59	681.49	790.82	10.21
53	2124	1	0	0	617.02	26.55	680.53	789.70	10.20
54	2125	1	1	0	617.31	26.56	680.85	790.08	10.20
55	1446	1	1	0	420.06	18.08	463.30	537.62	6.94
56	1247	1	0	0	362.25	15.59	399.54	463.63	5.99
57	2393	1	0	0	695.17	29.91	766.72	889.72	11.49
58	2395	1	1	0	695.75	29.94	767.36	890.46	11.50
59	2509	1	1	0	728.86	31.36	803.88	932.85	12.04
60	2279	1	1	0	662.05	28.49	730.19	847.33	10.94
61	1514	1	1	0	439.82	18.93	485.09	562.91	7.27
62	1514	1	1	0	439.82	18.93	485.09	562.91	7.27
63	1877	1	1	0	545.27	23.46	601.39	697.87	9.01
64	1875	1	0	0	544.69	23.44	600.75	697.13	9.00
65	1879	1	0	0	545.85	23.49	602.03	698.61	9.02
66	1900	1	1	0	551.95	23.75	608.76	706.42	9.12
67	1761	1	1	0	511.57	22.01	564.22	654.74	8.45
68	1760	1	0	0	511.28	22.00	563.90	654.37	8.45
69	2601	1	0	0	755.59	32.51	833.36	967.05	12.48
70	1750	1	1	0	508.38	21.88	560.70	650.65	8.40
71	1724	1	1	0	500.82	21.55	552.37	640.98	8.28
72	2118	1	1	0	615.28	26.48	678.61	787.47	10.17
73	2120	1	0	0	615.86	26.50	679.25	788.22	10.18
74	2115	1	1	0	614.41	26.44	677.65	786.36	10.15
75	1848	1	1	0	536.84	23.10	592.10	687.09	8.87
76	1850	1	0	0	537.43	23.13	592.74	687.83	8.88
77	1845	1	1	0	535.97	23.06	591.14	685.97	8.86
78	2280	1	1	0	662.34	28.50	730.51	847.70	10.94
79	2285	1	0	0	663.79	28.56	732.11	849.56	10.97
80	2540	1	0	0	737.87	31.75	813.82	944.37	12.19
81	2428	1	1	0	705.33	30.35	777.93	902.73	11.65
82	2501	1	1	0	726.54	31.26	801.32	929.87	12.00
83	1391	1	1	0	404.09	17.39	445.68	517.17	6.68
84	2101	1	1	0	610.34	26.26	673.16	781.15	10.08
85	2105	1	0	0	611.50	26.31	674.44	782.64	10.10
86	2396	1	0	0	696.04	29.95	767.68	890.83	11.50

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
87	2185	1	0	0	634.74	27.31	700.07	812.38	10.49
88	2190	1	1	0	636.20	27.38	701.68	814.24	10.51
89	1477	1	1	0	429.07	18.46	473.23	549.15	7.09
90	1607	1	1	0	466.83	20.09	514.88	597.48	7.71
91	1605	1	0	0	466.25	20.06	514.24	596.74	7.70
92	1982	1	0	0	575.77	24.78	635.03	736.91	9.51
93	1980	1	1	0	575.19	24.75	634.39	736.16	9.50
94	1985	1	0	0	576.64	24.81	635.99	738.02	9.53
95	2427	1	0	0	705.04	30.34	777.61	902.36	11.65
96	2491	1	0	0	723.64	31.14	798.12	926.15	11.96
97	2490	1	1	0	723.35	31.13	797.80	925.78	11.95
98	2292	1	1	0	665.83	28.65	734.36	852.17	11.00
99	1359	1	1	0	394.79	16.99	435.42	505.28	6.52
100	2160	1	1	0	627.48	27.00	692.06	803.09	10.37
101	2161	1	0	0	627.77	27.01	692.38	803.46	10.37
102	1965	1	0	0	570.83	24.56	629.59	730.59	9.43
103	2579	1	0	0	749.20	32.24	826.31	958.87	12.38
104	2580	1	1	0	749.49	32.25	826.63	959.24	12.38
105	1559	1	1	0	452.89	19.49	499.50	579.64	7.48
106	1560	1	0	0	453.18	19.50	499.82	580.01	7.49
107	1555	1	1	0	451.73	19.44	498.22	578.15	7.46
108	1489	1	1	0	432.55	18.61	477.08	553.61	7.15
109	2310	1	1	0	671.06	28.88	740.12	858.86	11.09
110	2311	1	0	0	671.35	28.89	740.44	859.23	11.09
111	2214	1	0	0	643.17	27.68	709.37	823.17	10.63
112	2407	1	0	0	699.23	30.09	771.20	894.92	11.55
113	2410	1	1	0	700.11	30.13	772.16	896.04	11.57
114	2381	1	1	0	691.68	29.76	762.87	885.26	11.43
115	2405	1	1	0	698.65	30.06	770.56	894.18	11.54
116	2526	1	1	0	733.80	31.58	809.33	939.17	12.12
117	1220	1	1	0	354.41	15.25	390.89	453.60	5.86
118	1059	1	1	0	307.64	13.24	339.30	393.74	5.08
119	1405	1	1	0	408.15	17.56	450.16	522.38	6.74
120	1406	1	0	0	408.44	17.58	450.48	522.75	6.75
121	2439	1	0	0	708.53	30.49	781.46	906.82	11.71
122	2666	1	0	0	774.47	33.33	854.19	991.22	12.80

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
123	2665	1	1	0	774.18	33.31	853.87	990.85	12.79
124	2318	1	1	0	673.38	28.98	742.69	861.83	11.13
125	2158	1	1	0	626.90	26.98	691.42	802.34	10.36
126	2602	1	1	0	755.88	32.53	833.68	967.42	12.49
127	1951	1	1	0	566.77	24.39	625.10	725.38	9.36
128	1167	1	1	0	339.01	14.59	373.91	433.89	5.60
129	1143	1	1	0	332.04	14.29	366.22	424.97	5.49
130	1283	1	1	0	372.71	16.04	411.07	477.02	6.16
131	1280	1	0	0	371.84	16.00	410.11	475.90	6.14
132	2309	1	0	0	670.76	28.86	739.80	858.49	11.08
133	2455	1	0	0	713.18	30.69	786.58	912.77	11.78
134	2360	1	0	0	685.58	29.50	756.14	877.45	11.33
135	2343	1	1	0	680.64	29.29	750.70	871.13	11.25
136	2266	1	1	0	658.27	28.33	726.03	842.50	10.88
137	2138	1	1	0	621.09	26.73	685.02	794.91	10.26
138	1514	1	1	0	439.82	18.93	485.09	562.91	7.27
139	2178	1	1	0	632.71	27.23	697.83	809.78	10.45
140	2175	0	1	0	631.84	27.19	696.87	808.67	10.44
141	1896	0	1	0	550.79	23.70	607.48	704.93	9.10
142	1900	0	1	1	551.95	23.75	608.76	706.42	9.12
143	2350	0	1	1	682.68	29.38	752.94	873.73	11.28
144	2272	0	1	0	660.02	28.40	727.95	844.73	10.91
145	2373	0	1	0	689.36	29.66	760.31	882.28	11.39
146	2375	0	1	1	689.94	29.69	760.95	883.03	11.40
147	1478	0	1	1	429.36	18.48	473.55	549.52	7.09
148	2400	0	1	1	697.20	30.00	768.96	892.32	11.52
149	2399	0	1	0	696.91	29.99	768.64	891.95	11.52
150	2259	0	1	0	656.24	28.24	723.78	839.90	10.84
151	2339	0	1	0	679.48	29.24	749.42	869.64	11.23
152	2340	1	1	0	679.77	29.25	749.74	870.01	11.23
153	2288	1	1	0	664.66	28.60	733.08	850.68	10.98
154	2305	1	1	0	669.60	28.81	738.52	857.00	11.06
155	2488	1	1	0	722.76	31.10	797.16	925.04	11.94
156	1610	1	1	0	467.71	20.13	515.84	598.60	7.73
157	1184	1	1	0	343.95	14.80	379.35	440.21	5.68
158	1484	1	1	0	431.10	18.55	475.47	551.75	7.12

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
159	2468	0	1	0	716.95	30.85	790.75	917.60	11.85
160	2513	0	1	0	730.03	31.41	805.17	934.33	12.06
161	2515	1	1	0	730.61	31.44	805.81	935.08	12.07
162	2259	1	1	0	656.24	28.24	723.78	839.90	10.84
163	2481	1	1	0	720.73	31.01	794.91	922.44	11.91
164	2436	1	1	0	707.66	30.45	780.49	905.70	11.69
165	1627	1	1	0	472.64	20.34	521.29	604.92	7.81
166	1625	0	1	0	472.06	20.31	520.65	604.18	7.80
167	1995	0	1	0	579.55	24.94	639.20	741.74	9.58
168	1928	0	1	0	560.08	24.10	617.73	716.83	9.25
169	2245	1	1	0	652.17	28.06	719.30	834.69	10.78
170	2243	0	1	0	651.59	28.04	718.66	833.95	10.77
171	2353	0	1	0	683.55	29.41	753.90	874.85	11.29
172	2383	0	1	0	692.26	29.79	763.51	886.00	11.44
173	2460	1	1	0	714.63	30.75	788.18	914.63	11.81
174	2340	1	1	0	679.77	29.25	749.74	870.01	11.23
175	2051	1	1	0	595.82	25.64	657.14	762.56	9.84
176	1576	1	1	0	457.83	19.70	504.95	585.96	7.56
177	2674	1	1	0	776.80	33.43	856.75	994.19	12.84
178	2645	0	0	1	768.37	33.06	847.46	983.41	12.70
179	2479	0	1	1	720.15	30.99	794.27	921.69	11.90
180	2476	0	0	1	719.28	30.95	793.31	920.58	11.88
181	2830	0	0	1	822.12	35.38	906.73	1052.2	13.58
182	2835	0	1	1	823.57	35.44	908.33	1054.1	13.61
183	2465	0	1	1	716.08	30.81	789.79	916.49	11.83
184	1499	0	1	1	435.46	18.74	480.28	557.33	7.20
185	1500	1	1	0	435.75	18.75	480.60	557.70	7.20
186	2029	1	1	0	589.42	25.36	650.09	754.38	9.74
187	2030	0	1	0	589.72	25.38	650.41	754.75	9.74
188	2035	1	1	0	591.17	25.44	652.01	756.61	9.77
189	2665	1	1	0	774.18	33.31	853.87	990.85	12.79
190	2663	0	1	0	773.60	33.29	853.23	990.10	12.78
191	2438	0	1	0	708.24	30.48	781.14	906.45	11.70
192	2645	0	1	0	768.37	33.06	847.46	983.41	12.70
193	2650	1	1	0	769.83	33.13	849.06	985.27	12.72
194	2481	1	1	0	720.73	31.01	794.91	922.44	11.91

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
195	1670	1	1	0	485.14	20.88	535.07	620.91	8.02
196	2015	1	1	0	585.36	25.19	645.61	749.18	9.67
197	3056	1	1	0	887.77	38.20	979.14	1136.2	14.67
198	3050	0	1	0	886.03	38.13	977.22	1134	14.64
199	2531	0	1	0	735.26	31.64	810.93	941.03	12.15
200	2504	0	1	0	727.41	31.30	802.28	930.99	12.02
201	2510	1	1	0	729.16	31.38	804.20	933.22	12.05
202	2333	1	1	0	677.74	29.16	747.49	867.41	11.20
203	2410	1	1	0	700.11	30.13	772.16	896.04	11.57
204	2440	1	1	0	708.82	30.50	781.78	907.19	11.71
205	1654	1	1	0	480.49	20.68	529.94	614.96	7.94
206	1647	0	1	0	478.45	20.59	527.70	612.35	7.91
207	2549	0	1	0	740.48	31.86	816.70	947.72	12.24
208	2560	1	1	0	743.68	32.00	820.22	951.81	12.29
209	2383	1	1	0	692.26	29.79	763.51	886.00	11.44
210	2592	1	1	0	752.98	32.40	830.48	963.71	12.44
211	1497	1	1	0	434.88	18.71	479.64	556.58	7.19
212	2064	1	1	0	599.59	25.80	661.31	767.40	9.91
213	2060	0	1	0	598.43	25.75	660.02	765.91	9.89
214	2464	0	1	0	715.79	30.80	789.47	916.12	11.83
215	2708	1	1	0	786.67	33.85	867.64	1006.8	13.00
216	2463	1	1	0	715.50	30.79	789.15	915.74	11.82
217	2377	1	1	0	690.52	29.71	761.59	883.77	11.41
218	1721	1	1	0	499.95	21.51	551.41	639.87	8.26
219	1710	1	1	0	496.76	21.38	547.88	635.78	8.21
220	2058	0	1	0	597.85	25.73	659.38	765.16	9.88
221	2055	0	1	0	596.98	25.69	658.42	764.05	9.86
222	1825	0	1	0	530.16	22.81	584.73	678.54	8.76
223	1740	1	1	0	505.47	21.75	557.50	646.93	8.35
224	2377	1	1	0	690.52	29.71	761.59	883.77	11.41
225	2375	0	1	0	689.94	29.69	760.95	883.03	11.40
226	2270	0	1	0	659.44	28.38	727.31	843.99	10.90
227	2271	1	1	0	659.73	28.39	727.63	844.36	10.90
228	2738	0	1	0	795.39	34.23	877.26	1018	13.14
229	2922	1	1	0	848.84	36.53	936.21	1086.4	14.03
230	2610	1	1	0	758.21	32.63	836.24	970.40	12.53

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
231	1333	1	1	0	387.24	16.66	427.09	495.61	6.40
232	1404	1	1	0	407.86	17.55	449.84	522.01	6.74
233	1882	1	1	0	546.72	23.53	602.99	699.73	9.03
234	1880	0	1	0	546.14	23.50	602.35	698.98	9.02
235	1881	1	1	0	546.43	23.51	602.67	699.36	9.03
236	2560	1	1	0	743.68	32.00	820.22	951.81	12.29
237	2561	0	1	0	743.97	32.01	820.54	952.18	12.29
238	2676	0	1	0	777.38	33.45	857.39	994.94	12.84
239	2890	1	1	0	839.55	36.13	925.96	1074.5	13.87
240	2775	1	1	0	806.14	34.69	889.11	1031.8	13.32
241	2538	1	1	0	737.29	31.73	813.18	943.63	12.18
242	2425	1	1	0	704.46	30.31	776.97	901.62	11.64
243	2677	1	1	0	777.67	33.46	857.71	995.31	12.85
244	700	1	1	0	203.35	8.75	224.28	260.26	3.36
245	2158	1	1	0	626.90	26.98	691.42	802.34	10.36
246	2413	1	1	0	700.98	30.16	773.13	897.15	11.58
247	2410	0	1	0	700.11	30.13	772.16	896.04	11.57
248	2453	0	1	0	712.60	30.66	785.94	912.03	11.77
249	2626	1	1	0	762.85	32.83	841.37	976.35	12.60
250	2359	1	1	0	685.29	29.49	755.82	877.08	11.32
251	2325	1	1	0	675.41	29.06	744.93	864.44	11.16
252	1860	1	1	0	540.33	23.25	595.94	691.55	8.93
253	1527	1	1	0	443.59	19.09	489.25	567.74	7.33
254	1729	1	1	0	502.27	21.61	553.97	642.84	8.30
255	2848	1	1	0	827.34	35.60	912.50	1058.9	13.67
256	2850	0	1	0	827.93	35.63	913.14	1059.6	13.68
257	2452	0	1	0	712.31	30.65	785.62	911.65	11.77
258	1539	0	1	0	447.08	19.24	493.10	572.20	7.39
259	1540	1	1	0	447.37	19.25	493.42	572.57	7.39
260	1988	1	1	0	577.51	24.85	636.96	739.14	9.54
261	1985	0	1	0	576.64	24.81	635.99	738.02	9.53
262	2800	0	1	0	813.40	35.00	897.12	1041.0	13.44
263	2805	1	1	0	814.85	35.06	898.72	1042.9	13.46
264	2633	1	1	0	764.89	32.91	843.61	978.95	12.64
265	2668	1	1	0	775.05	33.35	854.83	991.96	12.81
266	1738	1	1	0	504.89	21.73	556.86	646.19	8.34

Mode	Flow (IGPM)	Pump 1	Pump 2	Pump 3	Agrium (IGPM)	Vanscoy (IGPM)	Cory (IGPM)	Cogen (IGPM)	Corman (IGPM)
267	1272	1	1	0	369.52	15.90	407.55	472.93	6.11
268	1463	1	1	0	425.00	18.29	468.75	543.94	7.02
269	1564	1	1	0	454.34	19.55	501.11	581.50	7.51
270	2543	0	1	0	738.74	31.79	814.78	945.49	12.21
271	2839	0	1	0	824.73	35.49	909.62	1055.6	13.63
272	2841	1	1	0	825.31	35.51	910.26	1056.3	13.64
273	2555	1	1	0	742.23	31.94	818.62	949.95	12.26
274	2649	1	1	0	769.53	33.11	848.74	984.90	12.72
275	2438	1	1	0	708.24	30.48	781.14	906.45	11.70
276	2199	1	1	0	638.81	27.49	704.56	817.59	10.56
277	1759	1	1	0	510.99	21.99	563.58	654.00	8.44
278	1512	1	1	0	439.24	18.90	484.44	562.16	7.26
279	1510	0	1	0	438.66	18.88	483.80	561.42	7.25
280	2354	0	1	0	683.84	29.43	754.22	875.22	11.30
281	2355	1	1	0	684.13	29.44	754.54	875.59	11.30
282	2330	1	1	0	676.87	29.13	746.53	866.29	11.18
283	2758	1	1	0	801.20	34.48	883.66	1025.4	13.24
284	1954	0	1	0	567.64	24.43	626.06	726.50	9.38
285	1617	0	1	0	469.74	20.21	518.09	601.20	7.76
286	1620	1	1	0	470.61	20.25	519.05	602.32	7.78
287	1718	1	1	0	499.08	21.48	550.45	638.75	8.25
288	1715	0	1	0	498.21	21.44	549.49	637.64	8.23
289	2275	0	1	0	660.89	28.44	728.91	845.85	10.92
290	2592	1	1	0	752.98	32.40	830.48	963.71	12.44
291	2545	1	1	0	739.32	31.81	815.42	946.23	12.22
292	2755	1	1	0	800.33	34.44	882.70	1024.3	13.22
293	2151	1	1	0	624.87	26.89	689.18	799.74	10.32
294	2150	0	1	0	624.58	26.88	688.86	799.37	10.32
295	1655	0	1	0	480.78	20.69	530.26	615.33	7.94
296	2594	0	1	0	753.56	32.43	831.12	964.45	12.45
297	2600	1	1	0	755.30	32.50	833.04	966.68	12.48

Table B. 2: Pump Combination (ON = 1, OFF = 0)

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
1	1	1	0	0	1	25
2	1	1	0	1	0	26
3	1	1	0	1	0	26
4	0	1	0	0	0	8
5	0	1	0	1	0	10
6	1	1	0	1	0	26
7	0	1	0	1	0	10
8	0	1	1	1	0	14
9	0	1	1	1	0	14
10	0	1	1	1	0	14
11	0	1	1	1	0	14
12	0	1	1	1	0	14
13	0	1	0	1	0	10
14	0	1	1	1	0	14
15	0	1	0	1	0	10
16	0	1	1	1	0	14
17	0	1	1	1	0	14
18	0	1	1	1	0	14
19	0	1	1	1	0	14
20	0	1	0	1	0	10
21	0	1	1	1	0	14
22	0	1	1	1	0	14
23	0	1	1	1	0	14
24	0	1	1	1	0	14
25	0	1	1	1	0	14
26	0	1	1	1	0	14
27	0	1	0	1	0	10
28	0	1	0	0	0	8
29	0	1	0	1	0	10
30	0	1	1	1	0	14
31	0	1	0	1	0	10
32	0	1	0	1	0	10
33	0	1	1	1	0	14
34	0	1	1	1	0	14
35	0	1	1	1	0	14
36	0	1	1	1	0	14

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
37	0	1	0	1	0	10
38	0	1	1	1	0	14
39	0	1	1	0	0	12
40	0	1	0	1	0	10
41	0	1	0	1	0	10
42	0	1	0	1	0	10
43	0	1	1	1	0	14
44	1	1	0	1	0	26
45	1	1	0	1	0	26
46	1	1	0	1	0	26
47	0	1	0	1	0	10
48	1	0	0	1	0	18
49	1	0	0	1	0	18
50	1	0	0	1	0	18
51	1	1	0	1	0	26
52	1	1	0	1	0	26
53	1	0	0	1	0	18
54	1	1	0	1	0	26
55	1	1	0	0	0	24
56	1	0	0	1	0	18
57	1	0	0	1	0	18
58	1	1	0	1	0	26
59	1	1	0	1	0	26
60	1	1	0	1	0	26
61	1	1	0	1	0	26
62	1	1	0	1	0	26
63	1	1	0	1	0	26
64	1	0	0	1	0	18
65	1	0	0	1	0	18
66	1	1	0	0	0	24
67	1	1	0	1	0	26
68	1	0	0	1	0	18
69	1	0	0	1	0	18
70	1	1	0	1	0	26
71	1	1	0	1	0	26
72	1	1	0	1	0	26
73	1	0	0	1	0	18

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
74	1	1	0	1	0	26
75	1	1	0	1	0	26
76	1	0	0	1	0	18
77	1	1	0	0	0	24
78	1	1	0	1	0	26
79	1	0	0	1	0	18
80	1	0	0	1	0	18
81	1	1	0	1	0	26
82	1	1	0	1	0	26
83	1	1	0	1	0	26
84	1	1	0	1	0	26
85	1	0	0	1	0	18
86	1	0	0	1	0	18
87	1	0	0	1	0	18
88	1	1	0	1	0	26
89	1	1	0	1	0	26
90	1	1	0	1	0	26
91	1	0	0	1	0	18
92	1	0	0	1	0	18
93	1	1	0	1	0	26
94	1	0	0	1	0	18
95	1	0	0	1	0	18
96	1	0	0	1	0	18
97	1	1	0	0	0	24
98	1	1	0	0	0	24
99	1	1	0	1	0	26
100	1	1	0	1	0	26
101	1	0	0	1	0	18
102	1	0	0	1	0	18
103	1	0	0	1	0	18
104	1	1	0	1	0	26
105	1	1	0	1	0	26
106	1	0	0	1	0	18
107	1	1	0	1	0	26
108	1	1	0	1	0	26
109	1	1	0	1	0	26
110	1	0	0	1	0	18

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
111	1	0	0	1	0	18
112	1	0	0	1	0	18
113	1	1	0	1	0	26
114	1	1	0	1	0	26
115	1	1	0	1	0	26
116	1	1	0	1	0	26
117	1	1	0	0	0	24
118	1	1	0	1	0	26
119	1	1	0	1	0	26
120	1	0	0	1	0	18
121	1	0	0	1	0	18
122	1	0	0	1	0	18
123	1	1	0	1	0	26
124	1	1	0	1	0	26
125	1	1	0	1	0	26
126	1	1	0	1	0	26
127	1	1	0	0	0	24
128	1	1	0	0	0	24
129	1	1	0	1	0	26
130	1	1	0	1	0	26
131	1	0	0	1	0	18
132	1	0	0	1	0	18
133	1	0	0	1	0	18
134	1	0	0	1	0	18
135	1	1	0	1	0	26
136	1	1	0	1	0	26
137	1	1	0	1	0	26
138	1	1	0	1	0	26
139	1	1	0	1	0	26
140	0	1	0	1	0	10
141	0	1	0	1	0	10
142	0	1	1	1	0	14
143	0	1	1	1	0	14
144	0	1	0	1	0	10
145	0	1	0	1	0	10
146	0	1	1	0	0	12
147	0	1	1	1	0	14

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
148	0	1	1	1	0	14
149	0	1	0	1	0	10
150	0	1	0	1	0	10
151	0	1	0	1	0	10
152	1	1	0	1	0	26
153	1	1	0	1	0	26
154	1	1	0	1	0	26
155	1	1	0	1	0	26
156	1	1	0	0	0	24
157	1	1	0	1	0	26
158	1	1	0	1	0	26
159	0	1	0	1	0	10
160	0	1	0	1	0	10
161	1	1	0	1	0	26
162	1	1	0	1	0	26
163	1	1	0	1	0	26
164	1	1	0	0	0	24
165	1	1	0	0	0	24
166	0	1	0	0	0	8
167	0	1	0	0	0	8
168	0	1	0	1	0	10
169	1	1	0	1	0	26
170	0	1	0	1	0	10
171	0	1	0	1	0	10
172	0	1	0	1	0	10
173	1	1	0	1	0	26
174	1	1	0	1	0	26
175	1	1	0	0	0	24
176	1	1	0	1	0	26
177	1	1	0	1	0	26
178	0	0	1	1	0	6
179	0	1	1	1	0	14
180	0	0	1	1	0	6
181	0	0	1	1	0	6
182	0	1	1	1	0	14
183	0	1	1	1	0	14
184	0	1	1	1	0	14

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
185	1	1	0	1	0	26
186	1	1	0	0	0	24
187	0	1	0	1	0	10
188	1	1	0	1	0	26
189	1	1	0	1	0	26
190	0	1	0	1	0	10
191	0	1	0	1	0	10
192	0	1	0	1	0	10
193	1	1	0	1	0	26
194	1	1	0	1	0	26
195	1	1	0	1	0	26
196	1	1	0	1	0	26
197	1	1	0	1	0	26
198	0	1	0	1	0	10
199	0	1	0	1	0	10
200	0	1	0	1	0	10
201	1	1	0	1	0	26
202	1	1	0	1	0	26
203	1	1	0	1	0	26
204	1	1	0	0	0	24
205	1	1	0	1	0	26
206	0	1	0	1	0	10
207	0	1	0	1	0	10
208	1	1	0	1	0	26
209	1	1	0	1	0	26
210	1	1	0	1	0	26
211	1	1	0	1	0	26
212	1	1	0	1	0	26
213	0	1	0	1	0	10
214	0	1	0	1	0	10
215	1	1	0	1	0	26
216	1	1	0	1	0	26
217	1	1	0	1	0	26
218	1	1	0	0	0	24
219	1	1	0	0	0	24
220	0	1	0	0	1	9
221	0	1	0	0	1	9

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
222	0	1	0	0	1	9
223	1	1	0	0	1	25
224	1	1	0	0	1	25
225	0	1	0	0	1	9
226	0	1	0	0	1	9
227	1	1	0	0	1	25
228	0	1	0	0	1	9
229	1	1	0	0	1	25
230	1	1	0	0	1	25
231	1	1	0	0	1	25
232	1	1	0	0	1	25
233	1	1	0	0	1	25
234	0	1	0	0	1	9
235	1	1	0	0	1	25
236	1	1	0	0	1	25
237	0	1	0	0	1	9
238	0	1	0	0	1	9
239	1	1	0	0	1	25
240	1	1	0	0	1	25
241	1	1	0	0	1	25
242	1	1	0	0	0	24
243	1	1	0	0	1	25
244	1	1	0	0	1	25
245	1	1	0	0	1	25
246	1	1	0	0	1	25
247	0	1	0	0	1	9
248	0	1	0	0	1	9
249	1	1	0	0	1	25
250	1	1	0	0	1	25
251	1	1	0	0	1	25
252	1	1	0	0	1	25
253	1	1	0	0	1	25
254	1	1	0	0	0	24
255	1	1	0	0	1	25
256	0	1	0	0	1	9
257	0	1	0	0	1	9
258	0	1	0	0	1	9

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
259	1	1	0	0	1	25
260	1	1	0	0	1	25
261	0	1	0	0	1	9
262	0	1	0	0	1	9
263	1	1	0	0	1	25
264	1	1	0	0	1	25
265	1	1	0	0	1	25
266	1	1	0	0	1	25
267	1	1	0	0	1	25
268	1	1	0	0	1	25
269	1	1	0	0	1	25
270	0	1	0	0	1	9
271	0	1	0	0	1	9
272	1	1	0	0	1	25
273	1	1	0	0	0	24
274	1	1	0	0	1	25
275	1	1	0	0	1	25
276	1	1	0	0	1	25
277	1	1	0	0	1	25
278	1	1	0	0	1	25
279	0	1	0	0	1	9
280	0	1	0	0	1	9
281	1	1	0	0	1	25
282	1	1	0	0	1	25
283	1	1	0	0	1	25
284	0	1	0	0	1	9
285	0	1	0	0	1	9
286	1	1	0	0	1	25
287	1	1	0	0	1	25
288	0	1	0	0	1	9
289	0	1	0	0	1	9
290	1	1	0	0	1	25
291	1	1	0	0	1	25
292	1	1	0	0	1	25
293	1	1	0	0	1	25
294	0	1	0	0	1	9
295	0	1	0	0	1	9

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5	Combination
296	0	1	0	0	1	9
297	1	1	0	0	1	25

Appendix C

Pipe Flow and Node Pressures for the Optimized Operating-Modes

The corresponding pump discharge, node pressures obtained from the optimized operating-modes are tabulated and provided as follows.

Table C.1: Pump Discharges Obtained From Solution 4

Mode	Time Period		Discharge (igpm)				
	From	To	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
181	3/1/11 11:00 AM	3/1/11 12:00 PM	2118	0	0	688	0
216	3/1/11 12:00 PM	3/1/11 1:00 PM	2118	0	0	0	688
128	3/1/11 1:00 PM	3/1/11 2:00 PM	2118	0	0	688	0
151	3/1/11 2:00 PM	3/1/11 3:00 PM	0	0	1035	444	0
72	3/1/11 3:00 PM	3/1/11 4:00 PM	1992	0	126	688	0
148	3/1/11 4:00 PM	3/1/11 5:00 PM	2000	0	118	688	0
223	3/1/11 5:00 PM	3/1/11 6:00 PM	2000	0	118	688	0
137	3/1/11 6:00 PM	3/1/11 7:00 PM	2000	0	118	0	688
99	3/1/11 7:00 PM	3/1/11 8:00 PM	2118	0	0	688	0
144	3/1/11 8:00 PM	3/1/11 9:00 PM	1995	0	123	688	0
190	3/1/11 9:00 PM	3/1/11 10:00 PM	2000	0	118	0	688
124	3/1/11 10:00 PM	3/1/11 11:00 PM	2000	0	118	0	688
42	3/1/11 11:00 PM	3/2/11 12:00 AM	1059	1059	0	688	0
173	3/2/11 12:00 AM	3/2/11 1:00 AM	2118	0	0	0	688
90	3/2/11 1:00 AM	3/2/11 2:00 AM	2000	0	118	688	0
155	3/2/11 2:00 AM	3/2/11 3:00 AM	2118	0	0	0	688
119	3/2/11 3:00 AM	3/2/11 4:00 AM	2000	0	118	688	0
170	3/2/11 4:00 AM	3/2/11 5:00 AM	2000	0	118	688	0
124	3/2/11 5:00 AM	3/2/11 6:00 AM	2071	0	151	722	0
167	3/2/11 6:00 AM	3/2/11 7:00 AM	2071	0	151	0	722

Mode	Time Period		Discharge (igpm)				
	From	To	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
49	3/2/11 7:00 AM	3/2/11 8:00 AM	2071	0	151	722	0
143	3/2/11 8:00 AM	3/2/11 9:00 AM	2071	0	151	722	0
157	3/2/11 9:00 AM	3/2/11 10:00 AM	2071	0	151	722	0
195	3/2/11 10:00 AM	3/2/11 11:00 AM	2071	0	151	722	0
47	3/2/11 11:00 AM	3/2/11 12:00 PM	2071	0	151	722	0
86	3/2/11 12:00 PM	3/2/11 1:00 PM	2071	0	151	722	0
164	3/2/11 1:00 PM	3/2/11 2:00 PM	2071	0	151	722	0
91	3/2/11 2:00 PM	3/2/11 3:00 PM	0	0	1035	450	0
141	3/2/11 3:00 PM	3/2/11 4:00 PM	2426	0	0	788	0
126	3/2/11 4:00 PM	3/2/11 5:00 PM	2219	0	207	0	788
175	3/2/11 5:00 PM	3/2/11 6:00 PM	2273	0	0	0	738
95	3/2/11 6:00 PM	3/2/11 7:00 PM	0	0	1035	450	0
155	3/2/11 7:00 PM	3/2/11 8:00 PM	2107	0	225	0	738
169	3/2/11 8:00 PM	3/2/11 9:00 PM	2108	0	225	738	0
174	3/2/11 9:00 PM	3/2/11 10:00 PM	2108	0	225	738	0
182	3/2/11 10:00 PM	3/2/11 11:00 PM	2273	0	0	738	0
126	3/2/11 11:00 PM	3/3/11 12:00 AM	2107	0	225	738	0
71	3/3/11 12:00 AM	3/3/11 1:00 AM	1807	0	0	587	0
174	3/3/11 1:00 AM	3/3/11 2:00 AM	1914	0	0	0	622
143	3/3/11 2:00 AM	3/3/11 3:00 AM	1914	0	0	622	0
79	3/3/11 3:00 AM	3/3/11 4:00 AM	0	0	1034	452	0
157	3/3/11 4:00 AM	3/3/11 5:00 AM	625	957	0	622	0
196	3/3/11 5:00 AM	3/3/11 6:00 AM	1914	0	0	622	0
148	3/3/11 6:00 AM	3/3/11 7:00 AM	1893	0	22	622	0
91	3/3/11 7:00 AM	3/3/11 8:00 AM	1914	0	0	622	0
122	3/3/11 8:00 AM	3/3/11 9:00 AM	1893	0	22	622	0
63	3/3/11 9:00 AM	3/3/11 10:00 AM	1893	0	22	622	0
83	3/3/11 10:00 AM	3/3/11 11:00 AM	1893	0	22	622	0
181	3/3/11 11:00 AM	3/3/11 12:00 PM	1893	0	22	622	0
144	3/3/11 12:00 PM	3/3/11 1:00 PM	1914	0	0	622	0
141	3/3/11 1:00 PM	3/3/11 2:00 PM	1893	0	22	0	622
189	3/3/11 2:00 PM	3/3/11 3:00 PM	1914	0	0	0	622
80	3/3/11 3:00 PM	3/3/11 4:00 PM	1996	0	116	687	0
133	3/3/11 4:00 PM	3/3/11 5:00 PM	1998	0	117	687	0
152	3/3/11 5:00 PM	3/3/11 6:00 PM	1998	0	117	687	0
116	3/3/11 6:00 PM	3/3/11 7:00 PM	1998	0	117	687	0
88	3/3/11 7:00 PM	3/3/11 8:00 PM	2000	0	118	688	0

Mode	Time Period		Discharge (igpm)				
	From	To	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
85	3/3/11 8:00 PM	3/3/11 9:00 PM	0	0	1034	457	0
132	3/3/11 9:00 PM	3/3/11 10:00 PM	1166	1166	798	452	0
92	3/3/11 10:00 PM	3/3/11 11:00 PM	2000	0	118	688	0
157	3/3/11 11:00 PM	3/4/11 12:00 AM	2000	0	118	0	688
62	3/4/11 12:00 AM	3/4/11 1:00 AM	0	0	1034	458	0
92	3/4/11 1:00 AM	3/4/11 2:00 AM	0	0	1035	443	0
137	3/4/11 2:00 AM	3/4/11 3:00 AM	1992	0	126	688	0
204	3/4/11 3:00 AM	3/4/11 4:00 AM	1059	1059	0	688	0
199	3/4/11 4:00 AM	3/4/11 5:00 AM	2118	0	0	0	688
172	3/4/11 5:00 AM	3/4/11 6:00 AM	2000	0	118	688	0
168	3/4/11 6:00 AM	3/4/11 7:00 AM	2118	0	0	0	688
117	3/4/11 7:00 AM	3/4/11 8:00 AM	1995	0	123	0	688
217	3/4/11 8:00 AM	3/4/11 9:00 AM	2000	0	118	688	0
169	3/4/11 9:00 AM	3/4/11 10:00 AM	1059	1059	0	688	0
157	3/4/11 10:00 AM	3/4/11 11:00 AM	2000	0	118	688	0
138	3/4/11 11:00 AM	3/4/11 12:00 PM	1059	1059	0	688	0
132	3/4/11 12:00 PM	3/4/11 1:00 PM	0	0	1035	446	0
126	3/4/11 1:00 PM	3/4/11 2:00 PM	2118	0	0	0	688
177	3/4/11 2:00 PM	3/4/11 3:00 PM	2071	0	151	722	0
184	3/4/11 3:00 PM	3/4/11 4:00 PM	2222	0	0	0	722
189	3/4/11 4:00 PM	3/4/11 5:00 PM	0	0	1035	447	0
222	3/4/11 5:00 PM	3/4/11 6:00 PM	2067	0	156	722	0
92	3/4/11 6:00 PM	3/4/11 7:00 PM	2071	0	151	722	0
140	3/4/11 7:00 PM	3/4/11 8:00 PM	2071	0	151	0	722
124	3/4/11 8:00 PM	3/4/11 9:00 PM	2071	0	151	722	0
123	3/4/11 9:00 PM	3/4/11 10:00 PM	1111	1111	0	722	0
85	3/4/11 10:00 PM	3/4/11 11:00 PM	2222	0	0	722	0
94	3/4/11 11:00 PM	3/5/11 12:00 AM	2071	0	152	722	0
124	3/5/11 12:00 AM	3/5/11 1:00 AM	2219	0	207	788	0
76	3/5/11 1:00 AM	3/5/11 2:00 AM	2219	0	207	788	0
86	3/5/11 2:00 AM	3/5/11 3:00 AM	2108	0	225	738	0
124	3/5/11 3:00 AM	3/5/11 4:00 AM	2108	0	225	738	0
79	3/5/11 4:00 AM	3/5/11 5:00 AM	1137	1137	0	738	0
156	3/5/11 5:00 AM	3/5/11 6:00 AM	0	2273	0	738	0
113	3/5/11 6:00 AM	3/5/11 7:00 AM	2108	0	225	0	738
155	3/5/11 7:00 AM	3/5/11 8:00 AM	2108	0	225	738	0
207	3/5/11 8:00 AM	3/5/11 9:00 AM	2108	0	225	738	0

Mode	Time Period		Discharge (igpm)				
	From	To	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
121	3/5/11 9:00 AM	3/5/11 10:00 AM	1807	0	0	0	587
157	3/5/11 10:00 AM	3/5/11 11:00 AM	0	0	1035	450	0
205	3/5/11 11:00 AM	3/5/11 12:00 PM	1914	0	0	0	622
209	3/5/11 12:00 PM	3/5/11 1:00 PM	1893	0	22	622	0
138	3/5/11 1:00 PM	3/5/11 2:00 PM	1893	0	22	0	622
158	3/5/11 2:00 PM	3/5/11 3:00 PM	1893	0	22	0	622
87	3/5/11 3:00 PM	3/5/11 4:00 PM	1893	0	22	0	622
123	3/5/11 4:00 PM	3/5/11 5:00 PM	0	0	1034	452	0
190	3/5/11 5:00 PM	3/5/11 6:00 PM	0	0	1035	439	0
127	3/5/11 6:00 PM	3/5/11 7:00 PM	1484	1484	0	459	0
156	3/5/11 7:00 PM	3/5/11 8:00 PM	1893	0	22	622	0
70	3/5/11 8:00 PM	3/5/11 9:00 PM	0	1914	0	622	0
101	3/5/11 9:00 PM	3/5/11 10:00 PM	1914	0	0	622	0
94	3/5/11 10:00 PM	3/5/11 11:00 PM	1893	0	22	622	0
92	3/5/11 11:00 PM	3/6/11 12:00 AM	1914	0	0	0	622
215	3/6/11 12:00 AM	3/6/11 1:00 AM	1996	0	116	687	0
167	3/6/11 1:00 AM	3/6/11 2:00 AM	0	2116	0	687	0
137	3/6/11 2:00 AM	3/6/11 3:00 AM	1998	0	117	687	0
117	3/6/11 3:00 AM	3/6/11 4:00 AM	1998	0	117	0	687
132	3/6/11 4:00 AM	3/6/11 5:00 AM	2000	0	118	688	0
100	3/6/11 5:00 AM	3/6/11 6:00 AM	2118	0	0	688	0
114	3/6/11 6:00 AM	3/6/11 7:00 AM	1059	1059	0	688	0
67	3/6/11 7:00 AM	3/6/11 8:00 AM	2118	0	0	0	688
129	3/6/11 8:00 AM	3/6/11 9:00 AM	1059	1059	0	688	0
81	3/6/11 9:00 AM	3/6/11 10:00 AM	2000	0	118	688	0
176	3/6/11 10:00 AM	3/6/11 11:00 AM	2000	0	118	688	0
151	3/6/11 11:00 AM	3/6/11 12:00 PM	2000	0	118	0	688
171	3/6/11 12:00 PM	3/6/11 1:00 PM	2000	0	118	688	0
116	3/6/11 1:00 PM	3/6/11 2:00 PM	2000	0	118	0	688
197	3/6/11 2:00 PM	3/6/11 3:00 PM	2000	0	118	688	0
117	3/6/11 3:00 PM	3/6/11 4:00 PM	2000	0	118	688	0
118	3/6/11 4:00 PM	3/6/11 5:00 PM	2118	0	0	0	688
89	3/6/11 5:00 PM	3/6/11 6:00 PM	1059	1059	0	688	0
105	3/6/11 6:00 PM	3/6/11 7:00 PM	1059	1059	0	688	0
130	3/6/11 7:00 PM	3/6/11 7:00 PM	0	0	1035	444	0
57	3/6/11 7:00 PM	3/6/11 8:00 PM	1992	0	126	688	0
152	3/6/11 8:00 PM	3/6/11 9:00 PM	2000	0	118	688	0

Mode	Time Period		Discharge (igpm)				
	From	To	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
197	3/6/11 9:00 PM	3/6/11 10:00 PM	2118	0	0	688	0
107	3/6/11 10:00 PM	3/6/11 11:00 PM	2069	0	153	722	0
81	3/6/11 11:00 PM	3/7/11 12:00 AM	2071	0	151	0	722
120	3/7/11 12:00 AM	3/7/11 1:00 AM	1111	1111	0	722	0
211	3/7/11 1:00 AM	3/7/11 2:00 AM	1111	1111	0	722	0
187	3/7/11 2:00 AM	3/7/11 3:00 AM	2222	0	0	722	0
129	3/7/11 3:00 AM	3/7/11 4:00 AM	2071	0	151	0	722
118	3/7/11 4:00 AM	3/7/11 5:00 AM	2222	0	0	722	0
78	3/7/11 5:00 AM	3/7/11 6:00 AM	2071	0	151	0	722
70	3/7/11 6:00 AM	3/7/11 7:00 AM	2222	0	0	0	722
118	3/7/11 7:00 AM	3/7/11 8:00 AM	2071	0	151	722	0
116	3/7/11 8:00 AM	3/7/11 9:00 AM	0	0	1035	446	0
70	3/7/11 9:00 AM	3/7/11 10:00 AM	1213	1213	0	788	0
124	3/7/11 10:00 AM	3/7/11 11:00 AM	2108	0	225	738	0
195	3/7/11 11:00 AM	3/7/11 12:00 PM	2108	0	225	738	0
121	3/7/11 12:00 PM	3/7/11 1:00 PM	0	0	1035	448	0
110	3/7/11 1:00 PM	3/7/11 2:00 PM	2273	0	0	0	738
65	3/7/11 2:00 PM	3/7/11 3:00 PM	2107	0	225	0	738
163	3/7/11 3:00 PM	3/7/11 4:00 PM	2273	0	0	738	0
177	3/7/11 4:00 PM	3/7/11 5:00 PM	2107	0	225	738	0
243	3/7/11 5:00 PM	3/7/11 6:00 PM	1807	0	0	0	587
89	3/7/11 6:00 PM	3/7/11 7:00 PM	1914	0	0	622	0
127	3/7/11 7:00 PM	3/7/11 8:00 PM	1914	0	0	0	622
86	3/7/11 8:00 PM	3/7/11 9:00 PM	1914	0	0	622	0
131	3/7/11 9:00 PM	3/7/11 10:00 PM	1893	0	22	622	0
125	3/7/11 10:00 PM	3/7/11 11:00 PM	1893	0	22	622	0
130	3/7/11 11:00 PM	3/8/11 12:00 AM	1914	0	0	622	0
151	3/8/11 12:00 AM	3/8/11 1:00 AM	0	0	1035	451	0
110	3/8/11 1:00 AM	3/8/11 2:00 AM	0	0	1035	438	0
157	3/8/11 2:00 AM	3/8/11 3:00 AM	1251	1251	525	605	0
167	3/8/11 3:00 AM	3/8/11 4:00 AM	625	957	0	622	0
175	3/8/11 4:00 AM	3/8/11 5:00 AM	0	0	1035	451	0
111	3/8/11 5:00 AM	3/8/11 6:00 AM	0	0	1035	439	0
122	3/8/11 6:00 AM	3/8/11 7:00 AM	1484	1484	0	0	460
182	3/8/11 7:00 AM	3/8/11 8:00 AM	1893	0	22	622	0
100	3/8/11 8:00 AM	3/8/11 9:00 AM	2111	0	0	687	0
73	3/8/11 9:00 AM	3/8/11 10:00 AM	2116	0	0	0	687

Table C.2: Node Pressures Obtained From Solution 4

Mode	Time Period		Pressures (psi)					
	From	To	QE	Corman	PCS	Aurora	Vanscoy	Agrium
181	3/1/11 11:00 AM	3/1/11 12:00 PM	178	82	71	119	105	26
216	3/1/11 12:00 PM	3/1/11 1:00 PM	178	82	71	119	105	26
128	3/1/11 1:00 PM	3/1/11 2:00 PM	178	82	71	119	105	26
151	3/1/11 2:00 PM	3/1/11 3:00 PM	178	82	71	119	105	26
72	3/1/11 3:00 PM	3/1/11 4:00 PM	198	102	91	142	128	48
148	3/1/11 4:00 PM	3/1/11 5:00 PM	198	102	91	142	128	48
223	3/1/11 5:00 PM	3/1/11 6:00 PM	198	102	91	142	128	48
137	3/1/11 6:00 PM	3/1/11 7:00 PM	198	102	91	142	128	48
99	3/1/11 7:00 PM	3/1/11 8:00 PM	178	82	71	119	105	26
144	3/1/11 8:00 PM	3/1/11 9:00 PM	198	102	91	142	128	48
190	3/1/11 9:00 PM	3/1/11 10:00 PM	198	102	91	142	128	48
124	3/1/11 10:00 PM	3/1/11 11:00 PM	198	102	91	142	128	48
42	3/1/11 11:00 PM	3/2/11 12:00 AM	235	139	128	178	164	85
173	3/2/11 12:00 AM	3/2/11 1:00 AM	178	82	71	119	105	26
90	3/2/11 1:00 AM	3/2/11 2:00 AM	198	102	91	142	128	48
155	3/2/11 2:00 AM	3/2/11 3:00 AM	178	82	71	119	105	26
119	3/2/11 3:00 AM	3/2/11 4:00 AM	198	102	91	142	128	48
170	3/2/11 4:00 AM	3/2/11 5:00 AM	198	102	91	142	128	48
124	3/2/11 5:00 AM	3/2/11 6:00 AM	191	91	79	124	110	30
167	3/2/11 6:00 AM	3/2/11 7:00 AM	191	91	79	124	110	30
49	3/2/11 7:00 AM	3/2/11 8:00 AM	191	91	79	124	110	30
143	3/2/11 8:00 AM	3/2/11 9:00 AM	191	91	79	124	110	30
157	3/2/11 9:00 AM	3/2/11 10:00 AM	191	91	79	124	110	30
195	3/2/11 10:00 AM	3/2/11 11:00 AM	191	91	79	124	110	30
47	3/2/11 11:00 AM	3/2/11 12:00 PM	191	91	79	124	110	30
86	3/2/11 12:00 PM	3/2/11 1:00 PM	191	91	79	124	110	30
164	3/2/11 1:00 PM	3/2/11 2:00 PM	191	91	79	124	110	30
91	3/2/11 2:00 PM	3/2/11 3:00 PM	191	91	79	124	110	30
141	3/2/11 3:00 PM	3/2/11 4:00 PM	165	31	15	94	42	30
126	3/2/11 4:00 PM	3/2/11 5:00 PM	175	69	53	94	80	30
175	3/2/11 5:00 PM	3/2/11 6:00 PM	161	60	47	91	77	30
95	3/2/11 6:00 PM	3/2/11 7:00 PM	161	60	47	91	77	30
155	3/2/11 7:00 PM	3/2/11 8:00 PM	187	86	72	117	103	32
169	3/2/11 8:00 PM	3/2/11 9:00 PM	187	86	72	117	103	32
174	3/2/11 9:00 PM	3/2/11 10:00 PM	187	86	72	117	103	32
182	3/2/11 10:00 PM	3/2/11 11:00 PM	161	60	47	91	77	32

Mode	Time Period		Pressures (psi)					
	From	To	QE	Corman	PCS	Aurora	Vanscoy	Agrium
126	3/2/11 11:00 PM	3/3/11 12:00 AM	187	86	72	117	103	32
71	3/3/11 12:00 AM	3/3/11 1:00 AM	215	128	121	177	163	88
174	3/3/11 1:00 AM	3/3/11 2:00 AM	198	107	99	153	139	63
143	3/3/11 2:00 AM	3/3/11 3:00 AM	198	107	99	153	139	63
79	3/3/11 3:00 AM	3/3/11 4:00 AM	198	107	99	153	139	63
157	3/3/11 4:00 AM	3/3/11 5:00 AM	238	148	139	193	179	103
196	3/3/11 5:00 AM	3/3/11 6:00 AM	198	107	99	153	139	63
148	3/3/11 6:00 AM	3/3/11 7:00 AM	211	121	113	167	153	77
91	3/3/11 7:00 AM	3/3/11 8:00 AM	198	107	99	153	139	63
122	3/3/11 8:00 AM	3/3/11 9:00 AM	211	121	113	167	153	77
63	3/3/11 9:00 AM	3/3/11 10:00 AM	211	121	113	167	153	77
83	3/3/11 10:00 AM	3/3/11 11:00 AM	211	121	113	167	153	77
181	3/3/11 11:00 AM	3/3/11 12:00 PM	211	121	113	167	153	77
144	3/3/11 12:00 PM	3/3/11 1:00 PM	198	107	99	153	139	63
141	3/3/11 1:00 PM	3/3/11 2:00 PM	211	121	113	167	153	77
189	3/3/11 2:00 PM	3/3/11 3:00 PM	198	107	99	153	139	63
80	3/3/11 3:00 PM	3/3/11 4:00 PM	199	103	92	142	128	49
133	3/3/11 4:00 PM	3/3/11 5:00 PM	199	103	91	142	128	49
152	3/3/11 5:00 PM	3/3/11 6:00 PM	199	103	91	142	128	49
116	3/3/11 6:00 PM	3/3/11 7:00 PM	199	103	91	142	128	49
88	3/3/11 7:00 PM	3/3/11 8:00 PM	198	102	91	142	128	48
85	3/3/11 8:00 PM	3/3/11 9:00 PM	198	102	91	142	128	48
132	3/3/11 9:00 PM	3/3/11 10:00 PM	198	102	91	142	128	48
92	3/3/11 10:00 PM	3/3/11 11:00 PM	198	102	91	142	128	48
157	3/3/11 11:00 PM	3/4/11 12:00 AM	198	102	91	142	128	48
62	3/4/11 12:00 AM	3/4/11 1:00 AM	198	102	91	142	128	48
92	3/4/11 1:00 AM	3/4/11 2:00 AM	198	102	91	142	128	48
137	3/4/11 2:00 AM	3/4/11 3:00 AM	198	102	91	142	128	48
204	3/4/11 3:00 AM	3/4/11 4:00 AM	235	139	128	178	164	85
199	3/4/11 4:00 AM	3/4/11 5:00 AM	178	82	71	119	105	26
172	3/4/11 5:00 AM	3/4/11 6:00 AM	198	102	91	142	128	48
168	3/4/11 6:00 AM	3/4/11 7:00 AM	178	82	71	119	105	26
117	3/4/11 7:00 AM	3/4/11 8:00 AM	198	102	91	142	128	48
217	3/4/11 8:00 AM	3/4/11 9:00 AM	198	102	91	142	128	48
169	3/4/11 9:00 AM	3/4/11 10:00 AM	235	139	128	178	164	85
157	3/4/11 10:00 AM	3/4/11 11:00 AM	198	102	91	142	128	48
138	3/4/11 11:00 AM	3/4/11 12:00 PM	235	139	128	178	164	85

Mode	Time Period		Pressures (psi)					
	From	To	QE	Corman	PCS	Aurora	Vanscoy	Agrium
132	3/4/11 12:00 PM	3/4/11 1:00 PM	235	139	128	178	164	85
126	3/4/11 1:00 PM	3/4/11 2:00 PM	178	82	71	119	105	26
177	3/4/11 2:00 PM	3/4/11 3:00 PM	191	91	79	124	110	30
184	3/4/11 3:00 PM	3/4/11 4:00 PM	168	69	56	102	88	27
189	3/4/11 4:00 PM	3/4/11 5:00 PM	168	69	56	102	88	27
222	3/4/11 5:00 PM	3/4/11 6:00 PM	191	91	79	124	110	30
92	3/4/11 6:00 PM	3/4/11 7:00 PM	191	91	79	124	110	30
140	3/4/11 7:00 PM	3/4/11 8:00 PM	191	91	79	124	110	30
124	3/4/11 8:00 PM	3/4/11 9:00 PM	191	91	79	124	110	30
123	3/4/11 9:00 PM	3/4/11 10:00 PM	234	135	122	168	154	73
85	3/4/11 10:00 PM	3/4/11 11:00 PM	168	69	56	102	88	27
94	3/4/11 11:00 PM	3/5/11 12:00 AM	191	91	79	124	110	30
124	3/5/11 12:00 AM	3/5/11 1:00 AM	175	69	53	94	80	30
76	3/5/11 1:00 AM	3/5/11 2:00 AM	175	69	53	94	80	30
86	3/5/11 2:00 AM	3/5/11 3:00 AM	187	86	72	117	103	32
124	3/5/11 3:00 AM	3/5/11 4:00 AM	187	86	72	117	103	32
79	3/5/11 4:00 AM	3/5/11 5:00 AM	233	133	119	164	150	68
156	3/5/11 5:00 AM	3/5/11 6:00 AM	161	60	47	91	77	68
113	3/5/11 6:00 AM	3/5/11 7:00 AM	187	86	72	117	103	32
155	3/5/11 7:00 AM	3/5/11 8:00 AM	187	86	72	117	103	32
207	3/5/11 8:00 AM	3/5/11 9:00 AM	187	86	72	117	103	32
121	3/5/11 9:00 AM	3/5/11 10:00 AM	215	128	121	177	163	88
157	3/5/11 10:00 AM	3/5/11 11:00 AM	215	128	121	177	163	88
205	3/5/11 11:00 AM	3/5/11 12:00 PM	198	107	99	153	139	63
209	3/5/11 12:00 PM	3/5/11 1:00 PM	211	121	113	167	153	77
138	3/5/11 1:00 PM	3/5/11 2:00 PM	211	121	113	167	153	77
158	3/5/11 2:00 PM	3/5/11 3:00 PM	211	121	113	167	153	77
87	3/5/11 3:00 PM	3/5/11 4:00 PM	211	121	113	167	153	77
123	3/5/11 4:00 PM	3/5/11 5:00 PM	211	121	113	167	153	77
190	3/5/11 5:00 PM	3/5/11 6:00 PM	211	121	113	167	153	77
127	3/5/11 6:00 PM	3/5/11 7:00 PM	211	121	113	167	153	77
156	3/5/11 7:00 PM	3/5/11 8:00 PM	211	121	113	167	153	77
70	3/5/11 8:00 PM	3/5/11 9:00 PM	198	107	99	153	139	63
101	3/5/11 9:00 PM	3/5/11 10:00 PM	198	107	99	153	139	63
94	3/5/11 10:00 PM	3/5/11 11:00 PM	211	121	113	167	153	77
92	3/5/11 11:00 PM	3/6/11 12:00 AM	198	107	99	153	139	63
215	3/6/11 12:00 AM	3/6/11 1:00 AM	199	103	92	142	128	49

Mode	Time Period		Pressures (psi)					
	From	To	QE	Corman	PCS	Aurora	Vanscoy	Agrium
167	3/6/11 1:00 AM	3/6/11 2:00 AM	178	82	71	120	106	26
137	3/6/11 2:00 AM	3/6/11 3:00 AM	199	103	91	142	128	49
117	3/6/11 3:00 AM	3/6/11 4:00 AM	199	103	91	142	128	49
132	3/6/11 4:00 AM	3/6/11 5:00 AM	198	102	91	142	128	48
100	3/6/11 5:00 AM	3/6/11 6:00 AM	178	82	71	119	105	26
114	3/6/11 6:00 AM	3/6/11 7:00 AM	235	139	128	178	164	85
67	3/6/11 7:00 AM	3/6/11 8:00 AM	178	82	71	119	105	26
129	3/6/11 8:00 AM	3/6/11 9:00 AM	235	139	128	178	164	85
81	3/6/11 9:00 AM	3/6/11 10:00 AM	198	102	91	142	128	48
176	3/6/11 10:00 AM	3/6/11 11:00 AM	198	102	91	142	128	48
151	3/6/11 11:00 AM	3/6/11 12:00 PM	198	102	91	142	128	48
171	3/6/11 12:00 PM	3/6/11 1:00 PM	198	102	91	142	128	48
116	3/6/11 1:00 PM	3/6/11 2:00 PM	198	102	91	142	128	48
197	3/6/11 2:00 PM	3/6/11 3:00 PM	198	102	91	142	128	48
117	3/6/11 3:00 PM	3/6/11 4:00 PM	198	102	91	142	128	48
118	3/6/11 4:00 PM	3/6/11 5:00 PM	178	82	71	119	105	26
89	3/6/11 5:00 PM	3/6/11 6:00 PM	235	139	128	178	164	85
105	3/6/11 6:00 PM	3/6/11 7:00 PM	235	139	128	178	164	85
130	3/6/11 7:00 PM	3/6/11 7:00 PM	235	139	128	178	164	85
57	3/6/11 7:00 PM	3/6/11 8:00 PM	198	102	91	142	128	48
152	3/6/11 8:00 PM	3/6/11 9:00 PM	198	102	91	142	128	48
197	3/6/11 9:00 PM	3/6/11 10:00 PM	178	82	71	119	105	26
107	3/6/11 10:00 PM	3/6/11 11:00 PM	191	91	79	124	110	30
81	3/6/11 11:00 PM	3/7/11 12:00 AM	191	91	79	124	110	30
120	3/7/11 12:00 AM	3/7/11 1:00 AM	234	135	122	168	154	73
211	3/7/11 1:00 AM	3/7/11 2:00 AM	234	135	122	168	154	73
187	3/7/11 2:00 AM	3/7/11 3:00 AM	168	69	56	102	88	27
129	3/7/11 3:00 AM	3/7/11 4:00 AM	191	91	79	124	110	30
118	3/7/11 4:00 AM	3/7/11 5:00 AM	168	69	56	102	88	27
78	3/7/11 5:00 AM	3/7/11 6:00 AM	191	91	79	124	110	30
70	3/7/11 6:00 AM	3/7/11 7:00 AM	168	69	56	102	88	27
118	3/7/11 7:00 AM	3/7/11 8:00 AM	191	91	79	124	110	30
116	3/7/11 8:00 AM	3/7/11 9:00 AM	191	91	79	124	110	30
70	3/7/11 9:00 AM	3/7/11 10:00 AM	232	126	110	151	137	53
124	3/7/11 10:00 AM	3/7/11 11:00 AM	187	86	72	117	103	32
195	3/7/11 11:00 AM	3/7/11 12:00 PM	187	86	72	117	103	32
121	3/7/11 12:00 PM	3/7/11 1:00 PM	187	86	72	117	103	32

Mode	Time Period		Pressures (psi)					
	From	To	QE	Corman	PCS	Aurora	Vanscoy	Agrium
110	3/7/11 1:00 PM	3/7/11 2:00 PM	161	60	47	91	77	32
65	3/7/11 2:00 PM	3/7/11 3:00 PM	187	86	72	117	103	32
163	3/7/11 3:00 PM	3/7/11 4:00 PM	161	60	47	91	77	32
177	3/7/11 4:00 PM	3/7/11 5:00 PM	187	86	72	117	103	32
243	3/7/11 5:00 PM	3/7/11 6:00 PM	215	128	121	177	163	88
89	3/7/11 6:00 PM	3/7/11 7:00 PM	198	107	99	153	139	63
127	3/7/11 7:00 PM	3/7/11 8:00 PM	198	107	99	153	139	63
86	3/7/11 8:00 PM	3/7/11 9:00 PM	198	107	99	153	139	63
131	3/7/11 9:00 PM	3/7/11 10:00 PM	211	121	113	167	153	77
125	3/7/11 10:00 PM	3/7/11 11:00 PM	211	121	113	167	153	77
130	3/7/11 11:00 PM	3/8/11 12:00 AM	198	107	99	153	139	63
151	3/8/11 12:00 AM	3/8/11 1:00 AM	198	107	99	153	139	63
110	3/8/11 1:00 AM	3/8/11 2:00 AM	198	107	99	153	139	63
157	3/8/11 2:00 AM	3/8/11 3:00 AM	198	107	99	153	139	63
167	3/8/11 3:00 AM	3/8/11 4:00 AM	238	148	139	193	179	103
175	3/8/11 4:00 AM	3/8/11 5:00 AM	238	148	139	193	179	103
111	3/8/11 5:00 AM	3/8/11 6:00 AM	238	148	139	193	179	103
122	3/8/11 6:00 AM	3/8/11 7:00 AM	238	148	139	193	179	103
182	3/8/11 7:00 AM	3/8/11 8:00 AM	211	121	113	167	153	77
100	3/8/11 8:00 AM	3/8/11 9:00 AM	179	83	72	120	106	27
73	3/8/11 9:00 AM	3/8/11 10:00 AM	178	82	71	120	106	26

Table C.3: Pump Discharge and Node Demands Obtained From Solution 4

Mode	Time Period		Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
	From	To					
181	3/1/11 11:00 AM	3/1/11 12:00 PM	0	1	0	1	0
216	3/1/11 12:00 PM	3/1/11 1:00 PM	0	1	0	0	1
128	3/1/11 1:00 PM	3/1/11 2:00 PM	1	1	0	1	0
151	3/1/11 2:00 PM	3/1/11 3:00 PM	0	1	0	1	0
72	3/1/11 3:00 PM	3/1/11 4:00 PM	1	1	0	1	0
148	3/1/11 4:00 PM	3/1/11 5:00 PM	1	1	0	1	0
223	3/1/11 5:00 PM	3/1/11 6:00 PM	1	1	0	0	1
137	3/1/11 6:00 PM	3/1/11 7:00 PM	0	1	0	1	0
99	3/1/11 7:00 PM	3/1/11 8:00 PM	1	1	0	1	0
144	3/1/11 8:00 PM	3/1/11 9:00 PM	1	1	0	1	0
190	3/1/11 9:00 PM	3/1/11 10:00 PM	1	1	0	1	0
124	3/1/11 10:00 PM	3/1/11 11:00 PM	1	1	0	1	0
42	3/1/11 11:00 PM	3/2/11 12:00 AM	1	1	0	1	0
173	3/2/11 12:00 AM	3/2/11 1:00 AM	0	1	0	1	0
90	3/2/11 1:00 AM	3/2/11 2:00 AM	1	0	0	1	0
155	3/2/11 2:00 AM	3/2/11 3:00 AM	0	1	0	1	0
119	3/2/11 3:00 AM	3/2/11 4:00 AM	1	1	0	1	0
170	3/2/11 4:00 AM	3/2/11 5:00 AM	1	1	0	1	0
124	3/2/11 5:00 AM	3/2/11 6:00 AM	1	1	0	1	0
167	3/2/11 6:00 AM	3/2/11 7:00 AM	1	1	0	1	0
49	3/2/11 7:00 AM	3/2/11 8:00 AM	1	1	0	1	0
143	3/2/11 8:00 AM	3/2/11 9:00 AM	1	1	0	1	0
157	3/2/11 9:00 AM	3/2/11 10:00 AM	1	1	0	1	0
195	3/2/11 10:00 AM	3/2/11 11:00 AM	1	1	0	1	0
47	3/2/11 11:00 AM	3/2/11 12:00 PM	1	0	0	1	0
86	3/2/11 12:00 PM	3/2/11 1:00 PM	1	0	0	1	0
164	3/2/11 1:00 PM	3/2/11 2:00 PM	0	1	1	1	0
91	3/2/11 2:00 PM	3/2/11 3:00 PM	1	1	0	1	0
141	3/2/11 3:00 PM	3/2/11 4:00 PM	1	1	0	1	0
126	3/2/11 4:00 PM	3/2/11 5:00 PM	1	1	0	1	0
175	3/2/11 5:00 PM	3/2/11 6:00 PM	1	1	0	1	0
95	3/2/11 6:00 PM	3/2/11 7:00 PM	1	0	0	1	0
155	3/2/11 7:00 PM	3/2/11 8:00 PM	0	1	0	1	0
169	3/2/11 8:00 PM	3/2/11 9:00 PM	1	1	0	1	0
174	3/2/11 9:00 PM	3/2/11 10:00 PM	1	1	0	1	0
182	3/2/11 10:00 PM	3/2/11 11:00 PM	1	1	0	1	0

Mode	Time Period		Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
	From	To					
126	3/2/11 11:00 PM	3/3/11 12:00 AM	1	1	0	1	0
71	3/3/11 12:00 AM	3/3/11 1:00 AM	1	0	0	1	0
174	3/3/11 1:00 AM	3/3/11 2:00 AM	1	1	0	1	0
143	3/3/11 2:00 AM	3/3/11 3:00 AM	1	1	0	1	0
79	3/3/11 3:00 AM	3/3/11 4:00 AM	1	0	0	1	0
157	3/3/11 4:00 AM	3/3/11 5:00 AM	1	1	0	1	0
196	3/3/11 5:00 AM	3/3/11 6:00 AM	1	1	0	1	0
148	3/3/11 6:00 AM	3/3/11 7:00 AM	1	1	0	1	0
91	3/3/11 7:00 AM	3/3/11 8:00 AM	1	1	0	1	0
122	3/3/11 8:00 AM	3/3/11 9:00 AM	1	0	0	1	0
63	3/3/11 9:00 AM	3/3/11 10:00 AM	1	0	0	1	0
83	3/3/11 10:00 AM	3/3/11 11:00 AM	1	1	0	1	0
181	3/3/11 11:00 AM	3/3/11 12:00 PM	0	1	0	1	0
144	3/3/11 12:00 PM	3/3/11 1:00 PM	1	1	0	1	0
141	3/3/11 1:00 PM	3/3/11 2:00 PM	1	1	0	1	0
189	3/3/11 2:00 PM	3/3/11 3:00 PM	1	1	0	1	0
80	3/3/11 3:00 PM	3/3/11 4:00 PM	1	0	0	1	0
133	3/3/11 4:00 PM	3/3/11 5:00 PM	0	1	0	1	0
152	3/3/11 5:00 PM	3/3/11 6:00 PM	1	1	0	1	0
116	3/3/11 6:00 PM	3/3/11 7:00 PM	1	1	0	1	0
88	3/3/11 7:00 PM	3/3/11 8:00 PM	1	0	0	1	0
85	3/3/11 8:00 PM	3/3/11 9:00 PM	1	0	0	1	0
132	3/3/11 9:00 PM	3/3/11 10:00 PM	0	1	1	1	0
92	3/3/11 10:00 PM	3/3/11 11:00 PM	1	1	0	1	0
157	3/3/11 11:00 PM	3/4/11 12:00 AM	1	1	0	1	0
62	3/4/11 12:00 AM	3/4/11 1:00 AM	1	1	0	1	0
92	3/4/11 1:00 AM	3/4/11 2:00 AM	1	1	0	1	0
137	3/4/11 2:00 AM	3/4/11 3:00 AM	0	1	0	1	0
204	3/4/11 3:00 AM	3/4/11 4:00 AM	0	1	0	0	1
199	3/4/11 4:00 AM	3/4/11 5:00 AM	0	1	0	0	1
172	3/4/11 5:00 AM	3/4/11 6:00 AM	0	1	0	1	0
168	3/4/11 6:00 AM	3/4/11 7:00 AM	0	1	0	1	0
117	3/4/11 7:00 AM	3/4/11 8:00 AM	1	1	0	1	0
217	3/4/11 8:00 AM	3/4/11 9:00 AM	1	1	0	0	1
169	3/4/11 9:00 AM	3/4/11 10:00 AM	1	1	0	1	0
157	3/4/11 10:00 AM	3/4/11 11:00 AM	1	1	0	1	0
138	3/4/11 11:00 AM	3/4/11 12:00 PM	0	1	0	1	0

Mode	Time Period		Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
	From	To					
132	3/4/11 12:00 PM	3/4/11 1:00 PM	0	1	1	1	0
126	3/4/11 1:00 PM	3/4/11 2:00 PM	1	1	0	1	0
177	3/4/11 2:00 PM	3/4/11 3:00 PM	1	1	0	1	0
184	3/4/11 3:00 PM	3/4/11 4:00 PM	1	1	0	1	0
189	3/4/11 4:00 PM	3/4/11 5:00 PM	1	1	0	1	0
222	3/4/11 5:00 PM	3/4/11 6:00 PM	1	1	0	0	1
92	3/4/11 6:00 PM	3/4/11 7:00 PM	1	1	0	1	0
140	3/4/11 7:00 PM	3/4/11 8:00 PM	1	1	0	1	0
124	3/4/11 8:00 PM	3/4/11 9:00 PM	1	1	0	1	0
123	3/4/11 9:00 PM	3/4/11 10:00 PM	1	0	0	1	0
85	3/4/11 10:00 PM	3/4/11 11:00 PM	1	0	0	1	0
94	3/4/11 11:00 PM	3/5/11 12:00 AM	1	0	0	1	0
124	3/5/11 12:00 AM	3/5/11 1:00 AM	1	1	0	1	0
76	3/5/11 1:00 AM	3/5/11 2:00 AM	1	1	0	1	0
86	3/5/11 2:00 AM	3/5/11 3:00 AM	1	0	0	1	0
124	3/5/11 3:00 AM	3/5/11 4:00 AM	1	1	0	1	0
79	3/5/11 4:00 AM	3/5/11 5:00 AM	1	0	0	1	0
156	3/5/11 5:00 AM	3/5/11 6:00 AM	1	1	0	1	0
113	3/5/11 6:00 AM	3/5/11 7:00 AM	1	0	0	1	0
155	3/5/11 7:00 AM	3/5/11 8:00 AM	0	1	0	1	0
207	3/5/11 8:00 AM	3/5/11 9:00 AM	1	1	0	0	1
121	3/5/11 9:00 AM	3/5/11 10:00 AM	1	0	0	1	0
157	3/5/11 10:00 AM	3/5/11 11:00 AM	1	1	0	1	0
205	3/5/11 11:00 AM	3/5/11 12:00 PM	1	1	0	0	1
209	3/5/11 12:00 PM	3/5/11 1:00 PM	1	1	0	0	1
138	3/5/11 1:00 PM	3/5/11 2:00 PM	0	1	0	1	0
158	3/5/11 2:00 PM	3/5/11 3:00 PM	1	1	0	1	0
87	3/5/11 3:00 PM	3/5/11 4:00 PM	1	1	0	1	0
123	3/5/11 4:00 PM	3/5/11 5:00 PM	1	0	0	1	0
190	3/5/11 5:00 PM	3/5/11 6:00 PM	1	1	0	1	0
127	3/5/11 6:00 PM	3/5/11 7:00 PM	1	1	0	1	0
156	3/5/11 7:00 PM	3/5/11 8:00 PM	1	1	0	1	0
70	3/5/11 8:00 PM	3/5/11 9:00 PM	1	1	0	1	0
101	3/5/11 9:00 PM	3/5/11 10:00 PM	1	1	0	1	0
94	3/5/11 10:00 PM	3/5/11 11:00 PM	1	0	0	1	0
92	3/5/11 11:00 PM	3/6/11 12:00 AM	1	1	0	1	0
215	3/6/11 12:00 AM	3/6/11 1:00 AM	0	1	0	0	1

Mode	Time Period		Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
	From	To					
167	3/6/11 1:00 AM	3/6/11 2:00 AM	1	1	0	1	0
137	3/6/11 2:00 AM	3/6/11 3:00 AM	0	1	0	1	0
117	3/6/11 3:00 AM	3/6/11 4:00 AM	1	1	0	1	0
132	3/6/11 4:00 AM	3/6/11 5:00 AM	0	1	1	1	0
100	3/6/11 5:00 AM	3/6/11 6:00 AM	1	1	0	1	0
114	3/6/11 6:00 AM	3/6/11 7:00 AM	1	1	0	1	0
67	3/6/11 7:00 AM	3/6/11 8:00 AM	1	1	0	1	0
129	3/6/11 8:00 AM	3/6/11 9:00 AM	0	1	0	1	0
81	3/6/11 9:00 AM	3/6/11 10:00 AM	1	0	0	1	0
176	3/6/11 10:00 AM	3/6/11 11:00 AM	1	1	0	1	0
151	3/6/11 11:00 AM	3/6/11 12:00 PM	0	1	0	1	0
171	3/6/11 12:00 PM	3/6/11 1:00 PM	0	1	0	1	0
116	3/6/11 1:00 PM	3/6/11 2:00 PM	1	1	0	1	0
197	3/6/11 2:00 PM	3/6/11 3:00 PM	1	1	0	1	0
117	3/6/11 3:00 PM	3/6/11 4:00 PM	1	1	0	1	0
118	3/6/11 4:00 PM	3/6/11 5:00 PM	1	1	0	1	0
89	3/6/11 5:00 PM	3/6/11 6:00 PM	1	0	0	1	0
105	3/6/11 6:00 PM	3/6/11 7:00 PM	1	1	0	1	0
130	3/6/11 7:00 PM	3/6/11 7:00 PM	0	1	0	1	0
57	3/6/11 7:00 PM	3/6/11 8:00 PM	1	1	0	1	0
152	3/6/11 8:00 PM	3/6/11 9:00 PM	1	1	0	1	0
197	3/6/11 9:00 PM	3/6/11 10:00 PM	1	1	0	1	0
107	3/6/11 10:00 PM	3/6/11 11:00 PM	1	1	0	1	0
81	3/6/11 11:00 PM	3/7/11 12:00 AM	1	0	0	1	0
120	3/7/11 12:00 AM	3/7/11 1:00 AM	1	0	0	1	0
211	3/7/11 1:00 AM	3/7/11 2:00 AM	1	1	0	0	1
187	3/7/11 2:00 AM	3/7/11 3:00 AM	0	1	0	1	0
129	3/7/11 3:00 AM	3/7/11 4:00 AM	0	1	0	1	0
118	3/7/11 4:00 AM	3/7/11 5:00 AM	1	1	0	1	0
78	3/7/11 5:00 AM	3/7/11 6:00 AM	1	1	0	1	0
70	3/7/11 6:00 AM	3/7/11 7:00 AM	1	1	0	1	0
118	3/7/11 7:00 AM	3/7/11 8:00 AM	1	1	0	1	0
116	3/7/11 8:00 AM	3/7/11 9:00 AM	1	1	0	1	0
70	3/7/11 9:00 AM	3/7/11 10:00 AM	1	1	0	1	0
124	3/7/11 10:00 AM	3/7/11 11:00 AM	1	1	0	1	0
195	3/7/11 11:00 AM	3/7/11 12:00 PM	1	1	0	1	0
121	3/7/11 12:00 PM	3/7/11 1:00 PM	1	0	0	1	0

Mode	Time Period		Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
	From	To					
110	3/7/11 1:00 PM	3/7/11 2:00 PM	1	1	0	1	0
65	3/7/11 2:00 PM	3/7/11 3:00 PM	1	1	0	1	0
163	3/7/11 3:00 PM	3/7/11 4:00 PM	0	0	1	1	0
177	3/7/11 4:00 PM	3/7/11 5:00 PM	1	1	0	1	0
243	3/7/11 5:00 PM	3/7/11 6:00 PM	1	1	0	0	1
89	3/7/11 6:00 PM	3/7/11 7:00 PM	1	0	0	1	0
127	3/7/11 7:00 PM	3/7/11 8:00 PM	1	1	0	1	0
86	3/7/11 8:00 PM	3/7/11 9:00 PM	1	0	0	1	0
131	3/7/11 9:00 PM	3/7/11 10:00 PM	0	1	1	1	0
125	3/7/11 10:00 PM	3/7/11 11:00 PM	1	1	0	1	0
130	3/7/11 11:00 PM	3/8/11 12:00 AM	0	1	0	1	0
151	3/8/11 12:00 AM	3/8/11 1:00 AM	0	1	0	1	0
110	3/8/11 1:00 AM	3/8/11 2:00 AM	1	1	0	1	0
157	3/8/11 2:00 AM	3/8/11 3:00 AM	1	1	0	1	0
167	3/8/11 3:00 AM	3/8/11 4:00 AM	1	1	0	1	0
175	3/8/11 4:00 AM	3/8/11 5:00 AM	1	1	0	1	0
111	3/8/11 5:00 AM	3/8/11 6:00 AM	1	0	0	1	0
122	3/8/11 6:00 AM	3/8/11 7:00 AM	1	0	0	1	0
182	3/8/11 7:00 AM	3/8/11 8:00 AM	1	1	0	1	0
100	3/8/11 8:00 AM	3/8/11 9:00 AM	1	1	0	1	0
73	3/8/11 9:00 AM	3/8/11 10:00 AM	1	0	0	1	0

Table C.4: Pump Scheduling (ON = 1, OFF = 0) Obtained From Solution 4

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
181	0	1	0	1	0
216	0	1	0	0	1
128	1	1	0	1	0
151	0	1	0	1	0
72	1	1	0	1	0
148	1	1	0	1	0
223	1	1	0	0	1
137	0	1	0	1	0
99	1	1	0	1	0
144	1	1	0	1	0
190	1	1	0	1	0
124	1	1	0	1	0
42	1	1	0	1	0
173	0	1	0	1	0
90	1	0	0	1	0
155	0	1	0	1	0
119	1	1	0	1	0
170	1	1	0	1	0
124	1	1	0	1	0
167	1	1	0	1	0
49	1	1	0	1	0
143	1	1	0	1	0
157	1	1	0	1	0
195	1	1	0	1	0
47	1	0	0	1	0
86	1	0	0	1	0
164	0	1	1	1	0
91	1	1	0	1	0
141	1	1	0	1	0
126	1	1	0	1	0
175	1	1	0	1	0
95	1	0	0	1	0
155	0	1	0	1	0
169	1	1	0	1	0
174	1	1	0	1	0
182	1	1	0	1	0
126	1	1	0	1	0

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
71	1	0	0	1	0
174	1	1	0	1	0
143	1	1	0	1	0
79	1	0	0	1	0
157	1	1	0	1	0
196	1	1	0	1	0
148	1	1	0	1	0
91	1	1	0	1	0
122	1	0	0	1	0
63	1	0	0	1	0
83	1	1	0	1	0
181	0	1	0	1	0
144	1	1	0	1	0
141	1	1	0	1	0
189	1	1	0	1	0
80	1	0	0	1	0
133	0	1	0	1	0
152	1	1	0	1	0
116	1	1	0	1	0
88	1	0	0	1	0
85	1	0	0	1	0
132	0	1	1	1	0
92	1	1	0	1	0
157	1	1	0	1	0
62	1	1	0	1	0
92	1	1	0	1	0
137	0	1	0	1	0
204	0	1	0	0	1
199	0	1	0	0	1
172	0	1	0	1	0
168	0	1	0	1	0
117	1	1	0	1	0
217	1	1	0	0	1
169	1	1	0	1	0
157	1	1	0	1	0
138	0	1	0	1	0
132	0	1	1	1	0
126	1	1	0	1	0

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
177	1	1	0	1	0
184	1	1	0	1	0
189	1	1	0	1	0
222	1	1	0	0	1
92	1	1	0	1	0
140	1	1	0	1	0
124	1	1	0	1	0
123	1	0	0	1	0
85	1	0	0	1	0
94	1	0	0	1	0
124	1	1	0	1	0
76	1	1	0	1	0
86	1	0	0	1	0
124	1	1	0	1	0
79	1	0	0	1	0
156	1	1	0	1	0
113	1	0	0	1	0
155	0	1	0	1	0
207	1	1	0	0	1
121	1	0	0	1	0
157	1	1	0	1	0
205	1	1	0	0	1
209	1	1	0	0	1
138	0	1	0	1	0
158	1	1	0	1	0
87	1	1	0	1	0
123	1	0	0	1	0
190	1	1	0	1	0
127	1	1	0	1	0
156	1	1	0	1	0
70	1	1	0	1	0
101	1	1	0	1	0
94	1	0	0	1	0
92	1	1	0	1	0
215	0	1	0	0	1
167	1	1	0	1	0
137	0	1	0	1	0
117	1	1	0	1	0

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
132	0	1	1	1	0
100	1	1	0	1	0
114	1	1	0	1	0
67	1	1	0	1	0
129	0	1	0	1	0
81	1	0	0	1	0
176	1	1	0	1	0
151	0	1	0	1	0
171	0	1	0	1	0
116	1	1	0	1	0
197	1	1	0	1	0
117	1	1	0	1	0
118	1	1	0	1	0
89	1	0	0	1	0
105	1	1	0	1	0
130	0	1	0	1	0
57	1	1	0	1	0
152	1	1	0	1	0
197	1	1	0	1	0
107	1	1	0	1	0
81	1	0	0	1	0
120	1	0	0	1	0
211	1	1	0	0	1
187	0	1	0	1	0
129	0	1	0	1	0
118	1	1	0	1	0
78	1	1	0	1	0
70	1	1	0	1	0
118	1	1	0	1	0
116	1	1	0	1	0
70	1	1	0	1	0
124	1	1	0	1	0
195	1	1	0	1	0
121	1	0	0	1	0
110	1	1	0	1	0
65	1	1	0	1	0
163	0	0	1	1	0
177	1	1	0	1	0

Mode	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
243	1	1	0	0	1
89	1	0	0	1	0
127	1	1	0	1	0
86	1	0	0	1	0
131	0	1	1	1	0
125	1	1	0	1	0
130	0	1	0	1	0
151	0	1	0	1	0
110	1	1	0	1	0
157	1	1	0	1	0
167	1	1	0	1	0
175	1	1	0	1	0
111	1	0	0	1	0
122	1	0	0	1	0
182	1	1	0	1	0
100	1	1	0	1	0
73	1	0	0	1	0

Table C.5: Operating-Modes for the Solutions (1-7) Chosen in the Case Study

Time Period		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7
From	To							
3/1/11 11:00 AM	3/1/11 12:00 PM	173	191	138	181	188	90	179
3/1/11 12:00 PM	3/1/11 1:00 PM	151	182	234	216	185	120	151
3/1/11 1:00 PM	3/1/11 2:00 PM	26	87	38	128	84	124	32
3/1/11 2:00 PM	3/1/11 3:00 PM	60	106	81	151	103	224	57
3/1/11 3:00 PM	3/1/11 4:00 PM	27	53	48	72	56	63	27
3/1/11 4:00 PM	3/1/11 5:00 PM	44	158	101	148	161	89	44
3/1/11 5:00 PM	3/1/11 6:00 PM	194	230	176	223	227	116	194
3/1/11 6:00 PM	3/1/11 7:00 PM	152	149	244	137	152	101	152
3/1/11 7:00 PM	3/1/11 8:00 PM	129	153	168	99	150	84	129
3/1/11 8:00 PM	3/1/11 9:00 PM	140	113	195	144	110	272	140
3/1/11 9:00 PM	3/1/11 10:00 PM	168	120	244	190	117	140	168
3/1/11 10:00 PM	3/1/11 11:00 PM	132	162	250	124	159	237	129
3/1/11 11:00 PM	3/2/11 12:00 AM	24	61	15	42	64	160	24
3/2/11 12:00 AM	3/2/11 1:00 AM	181	205	260	173	202	75	178
3/2/11 1:00 AM	3/2/11 2:00 AM	95	164	148	90	161	68	98
3/2/11 2:00 AM	3/2/11 3:00 AM	32	95	232	155	92	196	32
3/2/11 3:00 AM	3/2/11 4:00 AM	45	51	49	119	54	152	51
3/2/11 4:00 AM	3/2/11 5:00 AM	198	184	177	170	187	48	198
3/2/11 5:00 AM	3/2/11 6:00 AM	139	124	110	124	121	88	136
3/2/11 6:00 AM	3/2/11 7:00 AM	69	120	231	167	123	175	66
3/2/11 7:00 AM	3/2/11 8:00 AM	123	75	117	49	78	111	123
3/2/11 8:00 AM	3/2/11 9:00 AM	62	87	62	143	90	111	68
3/2/11 9:00 AM	3/2/11 10:00 AM	201	126	183	157	123	204	198
3/2/11 10:00 AM	3/2/11 11:00 AM	178	118	119	195	121	109	163
3/2/11 11:00 AM	3/2/11 12:00 PM	180	90	149	47	93	181	180
3/2/11 12:00 PM	3/2/11 1:00 PM	95	85	99	86	82	248	86
3/2/11 1:00 PM	3/2/11 2:00 PM	86	122	83	164	125	202	86
3/2/11 2:00 PM	3/2/11 3:00 PM	52	115	86	91	118	104	52
3/2/11 3:00 PM	3/2/11 4:00 PM	70	69	154	141	72	245	70
3/2/11 4:00 PM	3/2/11 5:00 PM	64	115	251	126	118	11	58
3/2/11 5:00 PM	3/2/11 6:00 PM	241	160	234	175	157	163	238
3/2/11 6:00 PM	3/2/11 7:00 PM	89	140	89	95	143	68	89
3/2/11 7:00 PM	3/2/11 8:00 PM	237	171	257	155	168	252	237
3/2/11 8:00 PM	3/2/11 9:00 PM	242	212	158	169	209	272	242
3/2/11 9:00 PM	3/2/11 10:00 PM	180	127	157	174	130	180	180
3/2/11 10:00 PM	3/2/11 11:00 PM	175	205	187	182	202	5	163

Time Period		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7
From	To							
3/2/11 11:00 PM	3/3/11 12:00 AM	119	191	128	126	194	202	119
3/3/11 12:00 AM	3/3/11 1:00 AM	39	125	124	71	122	228	36
3/3/11 1:00 AM	3/3/11 2:00 AM	185	153	204	174	150	209	185
3/3/11 2:00 AM	3/3/11 3:00 AM	100	97	171	143	94	38	97
3/3/11 3:00 AM	3/3/11 4:00 AM	71	119	89	79	116	226	71
3/3/11 4:00 AM	3/3/11 5:00 AM	149	194	164	157	191	105	149
3/3/11 5:00 AM	3/3/11 6:00 AM	233	231	172	196	234	37	236
3/3/11 6:00 AM	3/3/11 7:00 AM	179	146	157	148	143	234	182
3/3/11 7:00 AM	3/3/11 8:00 AM	17	76	14	91	79	147	17
3/3/11 8:00 AM	3/3/11 9:00 AM	12	1	57	122	1	118	24
3/3/11 9:00 AM	3/3/11 10:00 AM	17	85	109	63	88	7	17
3/3/11 10:00 AM	3/3/11 11:00 AM	150	186	126	83	183	272	147
3/3/11 11:00 AM	3/3/11 12:00 PM	240	222	56	181	219	270	240
3/3/11 12:00 PM	3/3/11 1:00 PM	180	176	155	144	173	98	180
3/3/11 1:00 PM	3/3/11 2:00 PM	197	214	222	141	211	46	197
3/3/11 2:00 PM	3/3/11 3:00 PM	197	239	215	189	242	218	197
3/3/11 3:00 PM	3/3/11 4:00 PM	82	112	126	80	109	142	82
3/3/11 4:00 PM	3/3/11 5:00 PM	223	182	141	133	179	57	223
3/3/11 5:00 PM	3/3/11 6:00 PM	161	116	167	152	119	88	161
3/3/11 6:00 PM	3/3/11 7:00 PM	22	86	66	116	83	197	19
3/3/11 7:00 PM	3/3/11 8:00 PM	44	71	75	88	74	62	44
3/3/11 8:00 PM	3/3/11 9:00 PM	28	88	86	85	91	251	28
3/3/11 9:00 PM	3/3/11 10:00 PM	137	137	149	132	134	266	140
3/3/11 10:00 PM	3/3/11 11:00 PM	155	167	149	92	164	82	152
3/3/11 11:00 PM	3/4/11 12:00 AM	131	125	266	157	128	121	125
3/4/11 12:00 AM	3/4/11 1:00 AM	22	103	93	62	106	269	34
3/4/11 1:00 AM	3/4/11 2:00 AM	48	83	157	92	86	170	36
3/4/11 2:00 AM	3/4/11 3:00 AM	18	82	21	137	79	272	15
3/4/11 3:00 AM	3/4/11 4:00 AM	272	272	269	204	272	171	272
3/4/11 4:00 AM	3/4/11 5:00 AM	207	171	192	199	168	201	204
3/4/11 5:00 AM	3/4/11 6:00 AM	257	272	260	172	272	47	257
3/4/11 6:00 AM	3/4/11 7:00 AM	244	214	256	168	211	130	244
3/4/11 7:00 AM	3/4/11 8:00 AM	71	93	142	117	96	92	71
3/4/11 8:00 AM	3/4/11 9:00 AM	158	194	161	217	197	235	155
3/4/11 9:00 AM	3/4/11 10:00 AM	82	169	100	169	172	272	91
3/4/11 10:00 AM	3/4/11 11:00 AM	60	105	131	157	108	1	63
3/4/11 11:00 AM	3/4/11 12:00 PM	29	134	104	138	131	184	29

Time Period		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7
From	To							
3/4/11 12:00 PM	3/4/11 1:00 PM	41	80	212	132	83	228	41
3/4/11 1:00 PM	3/4/11 2:00 PM	166	142	83	126	139	163	169
3/4/11 2:00 PM	3/4/11 3:00 PM	244	147	247	177	144	247	244
3/4/11 3:00 PM	3/4/11 4:00 PM	132	146	61	184	143	96	132
3/4/11 4:00 PM	3/4/11 5:00 PM	173	182	188	189	179	200	176
3/4/11 5:00 PM	3/4/11 6:00 PM	186	171	183	222	168	170	180
3/4/11 6:00 PM	3/4/11 7:00 PM	202	154	211	92	151	191	202
3/4/11 7:00 PM	3/4/11 8:00 PM	217	196	100	140	193	97	220
3/4/11 8:00 PM	3/4/11 9:00 PM	100	139	33	124	136	117	82
3/4/11 9:00 PM	3/4/11 10:00 PM	174	153	180	123	150	95	177
3/4/11 10:00 PM	3/4/11 11:00 PM	21	47	69	85	44	9	36
3/4/11 11:00 PM	3/5/11 12:00 AM	33	126	66	94	129	125	33
3/5/11 12:00 AM	3/5/11 1:00 AM	103	106	124	124	109	242	106
3/5/11 1:00 AM	3/5/11 2:00 AM	39	72	83	76	75	211	39
3/5/11 2:00 AM	3/5/11 3:00 AM	52	67	110	86	70	121	49
3/5/11 3:00 AM	3/5/11 4:00 AM	77	77	23	124	80	31	77
3/5/11 4:00 AM	3/5/11 5:00 AM	112	126	162	79	123	118	109
3/5/11 5:00 AM	3/5/11 6:00 AM	119	122	207	156	125	79	119
3/5/11 6:00 AM	3/5/11 7:00 AM	104	104	116	113	107	87	107
3/5/11 7:00 AM	3/5/11 8:00 AM	101	170	119	155	167	212	89
3/5/11 8:00 AM	3/5/11 9:00 AM	217	199	205	207	196	260	217
3/5/11 9:00 AM	3/5/11 10:00 AM	44	62	47	121	65	206	44
3/5/11 10:00 AM	3/5/11 11:00 AM	178	221	200	157	218	141	178
3/5/11 11:00 AM	3/5/11 12:00 PM	118	167	118	205	170	97	118
3/5/11 12:00 PM	3/5/11 1:00 PM	236	169	236	209	166	74	230
3/5/11 1:00 PM	3/5/11 2:00 PM	216	173	236	138	176	272	228
3/5/11 2:00 PM	3/5/11 3:00 PM	223	193	217	158	196	271	217
3/5/11 3:00 PM	3/5/11 4:00 PM	33	43	50	87	40	265	36
3/5/11 4:00 PM	3/5/11 5:00 PM	19	112	79	123	115	180	19
3/5/11 5:00 PM	3/5/11 6:00 PM	260	221	187	190	224	148	263
3/5/11 6:00 PM	3/5/11 7:00 PM	88	63	84	127	66	117	88
3/5/11 7:00 PM	3/5/11 8:00 PM	180	180	162	156	177	245	168
3/5/11 8:00 PM	3/5/11 9:00 PM	23	82	26	70	85	87	23
3/5/11 9:00 PM	3/5/11 10:00 PM	140	113	57	101	110	261	140
3/5/11 10:00 PM	3/5/11 11:00 PM	224	167	206	94	164	206	221
3/5/11 11:00 PM	3/6/11 12:00 AM	69	94	54	92	97	244	54
3/6/11 12:00 AM	3/6/11 1:00 AM	255	185	162	215	182	251	240

Time Period		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7
From	To							
3/6/11 1:00 AM	3/6/11 2:00 AM	118	160	41	167	163	112	118
3/6/11 2:00 AM	3/6/11 3:00 AM	258	237	261	137	234	217	252
3/6/11 3:00 AM	3/6/11 4:00 AM	67	139	91	117	142	216	64
3/6/11 4:00 AM	3/6/11 5:00 AM	171	204	180	132	207	241	171
3/6/11 5:00 AM	3/6/11 6:00 AM	123	138	135	100	141	203	123
3/6/11 6:00 AM	3/6/11 7:00 AM	88	97	255	114	94	178	88
3/6/11 7:00 AM	3/6/11 8:00 AM	1	43	15	67	46	20	1
3/6/11 8:00 AM	3/6/11 9:00 AM	138	156	178	129	153	256	120
3/6/11 9:00 AM	3/6/11 10:00 AM	108	139	117	81	136	65	105
3/6/11 10:00 AM	3/6/11 11:00 AM	206	176	235	176	179	14	206
3/6/11 11:00 AM	3/6/11 12:00 PM	189	162	195	151	165	186	195
3/6/11 12:00 PM	3/6/11 1:00 PM	178	223	209	171	220	56	178
3/6/11 1:00 PM	3/6/11 2:00 PM	123	57	114	116	54	216	123
3/6/11 2:00 PM	3/6/11 3:00 PM	178	139	203	197	136	21	178
3/6/11 3:00 PM	3/6/11 4:00 PM	237	144	28	117	147	74	237
3/6/11 4:00 PM	3/6/11 5:00 PM	138	172	165	118	169	60	138
3/6/11 5:00 PM	3/6/11 6:00 PM	118	115	50	89	112	272	118
3/6/11 6:00 PM	3/6/11 7:00 PM	76	80	82	105	77	47	76
3/6/11 7:00 PM	3/6/11 7:00 PM	81	54	87	130	57	51	81
3/6/11 7:00 PM	3/6/11 8:00 PM	77	26	147	57	29	272	77
3/6/11 8:00 PM	3/6/11 9:00 PM	262	262	176	152	259	272	262
3/6/11 9:00 PM	3/6/11 10:00 PM	230	191	261	197	188	137	230
3/6/11 10:00 PM	3/6/11 11:00 PM	229	188	31	107	185	167	229
3/6/11 11:00 PM	3/7/11 12:00 AM	86	128	131	81	125	230	80
3/7/11 12:00 AM	3/7/11 1:00 AM	119	122	134	120	119	70	119
3/7/11 1:00 AM	3/7/11 2:00 AM	262	233	265	211	230	139	265
3/7/11 2:00 AM	3/7/11 3:00 AM	156	132	138	187	135	231	153
3/7/11 3:00 AM	3/7/11 4:00 AM	176	77	248	129	74	229	176
3/7/11 4:00 AM	3/7/11 5:00 AM	19	25	234	118	22	128	13
3/7/11 5:00 AM	3/7/11 6:00 AM	57	81	108	78	84	81	57
3/7/11 6:00 AM	3/7/11 7:00 AM	180	84	120	70	81	158	180
3/7/11 7:00 AM	3/7/11 8:00 AM	49	26	25	118	29	243	40
3/7/11 8:00 AM	3/7/11 9:00 AM	91	136	100	116	139	195	97
3/7/11 9:00 AM	3/7/11 10:00 AM	138	111	157	70	114	272	138
3/7/11 10:00 AM	3/7/11 11:00 AM	127	130	112	124	133	115	118
3/7/11 11:00 AM	3/7/11 12:00 PM	173	188	216	195	191	209	173
3/7/11 12:00 PM	3/7/11 1:00 PM	223	163	220	121	160	229	223

Time Period		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7
From	To							
3/7/11 1:00 PM	3/7/11 2:00 PM	71	101	130	110	104	135	71
3/7/11 2:00 PM	3/7/11 3:00 PM	89	62	84	65	65	182	86
3/7/11 3:00 PM	3/7/11 4:00 PM	180	207	205	163	204	207	180
3/7/11 4:00 PM	3/7/11 5:00 PM	193	204	187	177	207	177	196
3/7/11 5:00 PM	3/7/11 6:00 PM	221	137	212	243	140	129	221
3/7/11 6:00 PM	3/7/11 7:00 PM	67	55	19	89	52	253	61
3/7/11 7:00 PM	3/7/11 8:00 PM	168	100	185	127	97	183	168
3/7/11 8:00 PM	3/7/11 9:00 PM	121	95	76	86	92	227	130
3/7/11 9:00 PM	3/7/11 10:00 PM	143	158	146	131	155	18	143
3/7/11 10:00 PM	3/7/11 11:00 PM	76	112	60	125	115	96	76
3/7/11 11:00 PM	3/8/11 12:00 AM	98	176	46	130	179	176	89
3/8/11 12:00 AM	3/8/11 1:00 AM	129	135	141	151	138	199	138
3/8/11 1:00 AM	3/8/11 2:00 AM	159	96	13	110	93	154	159
3/8/11 2:00 AM	3/8/11 3:00 AM	71	89	79	157	86	227	71
3/8/11 3:00 AM	3/8/11 4:00 AM	103	133	94	167	130	120	103
3/8/11 4:00 AM	3/8/11 5:00 AM	179	164	204	175	161	272	179
3/8/11 5:00 AM	3/8/11 6:00 AM	193	85	184	111	88	272	193
3/8/11 6:00 AM	3/8/11 7:00 AM	181	166	187	122	169	131	181
3/8/11 7:00 AM	3/8/11 8:00 AM	219	141	234	182	138	39	234
3/8/11 8:00 AM	3/8/11 9:00 AM	62	122	7	100	125	272	65
3/8/11 9:00 AM	3/8/11 10:00 AM	81	162	102	73	159	50	84

Table C.6: Operating-Modes for the Solutions (8-14) Chosen in the Case Study

Time Period		Solution 8	Solution 9	Solution 10	solution 11	Solution 12	Solution 13	Solution 14
From	To							
3/1/11 11:00 AM	3/1/11 12:00 PM	150	166	265	265	209	158	106
3/1/11 12:00 PM	3/1/11 1:00 PM	213	235	250	242	268	238	160
3/1/11 1:00 PM	3/1/11 2:00 PM	59	192	118	144	219	104	181
3/1/11 2:00 PM	3/1/11 3:00 PM	57	272	69	97	272	87	114
3/1/11 3:00 PM	3/1/11 4:00 PM	30	223	178	178	134	30	48
3/1/11 4:00 PM	3/1/11 5:00 PM	44	60	240	241	220	85	172
3/1/11 5:00 PM	3/1/11 6:00 PM	173	242	272	272	170	161	118
3/1/11 6:00 PM	3/1/11 7:00 PM	238	177	243	243	173	209	89
3/1/11 7:00 PM	3/1/11 8:00 PM	132	58	76	77	61	149	124
3/1/11 8:00 PM	3/1/11 9:00 PM	192	195	145	148	162	173	108
3/1/11 9:00 PM	3/1/11 10:00 PM	238	238	196	197	141	180	94
3/1/11 10:00 PM	3/1/11 11:00 PM	129	115	124	121	133	90	100
3/1/11 11:00 PM	3/2/11 12:00 AM	24	162	185	185	105	60	85
3/2/11 12:00 AM	3/2/11 1:00 AM	178	220	110	110	92	181	245
3/2/11 1:00 AM	3/2/11 2:00 AM	142	104	176	176	233	143	72
3/2/11 2:00 AM	3/2/11 3:00 AM	32	42	84	85	38	56	125
3/2/11 3:00 AM	3/2/11 4:00 AM	69	91	5	9	170	105	105
3/2/11 4:00 AM	3/2/11 5:00 AM	201	184	78	57	272	159	122
3/2/11 5:00 AM	3/2/11 6:00 AM	116	166	126	118	188	139	58
3/2/11 6:00 AM	3/2/11 7:00 AM	222	232	166	168	48	180	88
3/2/11 7:00 AM	3/2/11 8:00 AM	120	270	146	146	272	111	123
3/2/11 8:00 AM	3/2/11 9:00 AM	71	126	155	155	229	101	155
3/2/11 9:00 AM	3/2/11 10:00 AM	207	134	141	141	270	192	178
3/2/11 10:00 AM	3/2/11 11:00 AM	169	272	262	262	145	181	99
3/2/11 11:00 AM	3/2/11 12:00 PM	137	30	245	241	145	123	83
3/2/11 12:00 PM	3/2/11 1:00 PM	99	181	77	68	26	122	56
3/2/11 1:00 PM	3/2/11 2:00 PM	101	272	170	170	35	119	83
3/2/11 2:00 PM	3/2/11 3:00 PM	58	268	180	181	255	88	210
3/2/11 3:00 PM	3/2/11 4:00 PM	148	138	151	148	236	154	180
3/2/11 4:00 PM	3/2/11 5:00 PM	64	227	69	84	58	70	129
3/2/11 5:00 PM	3/2/11 6:00 PM	237	272	202	197	233	181	148
3/2/11 6:00 PM	3/2/11 7:00 PM	68	226	214	175	177	116	224
3/2/11 7:00 PM	3/2/11 8:00 PM	254	267	269	269	258	261	225
3/2/11 8:00 PM	3/2/11 9:00 PM	242	260	194	189	170	212	190
3/2/11 9:00 PM	3/2/11 10:00 PM	154	127	181	181	189	150	105
3/2/11 10:00 PM	3/2/11 11:00 PM	193	101	181	181	189	199	261

Time Period		Solution 8	Solution 9	Solution 10	solution 11	Solution 12	Solution 13	Solution 14
From	To							
3/2/11 11:00 PM	3/3/11 12:00 AM	113	173	181	181	104	143	140
3/3/11 12:00 AM	3/3/11 1:00 AM	115	99	178	180	114	122	149
3/3/11 1:00 AM	3/3/11 2:00 AM	201	255	192	191	14	194	105
3/3/11 2:00 AM	3/3/11 3:00 AM	109	81	117	117	77	111	127
3/3/11 3:00 AM	3/3/11 4:00 AM	80	202	171	171	250	95	55
3/3/11 4:00 AM	3/3/11 5:00 AM	149	120	249	246	160	149	88
3/3/11 5:00 AM	3/3/11 6:00 AM	215	252	220	220	200	209	160
3/3/11 6:00 AM	3/3/11 7:00 AM	163	191	210	209	203	161	166
3/3/11 7:00 AM	3/3/11 8:00 AM	14	105	268	272	35	53	133
3/3/11 8:00 AM	3/3/11 9:00 AM	30	245	227	227	98	66	41
3/3/11 9:00 AM	3/3/11 10:00 AM	29	28	218	250	180	61	186
3/3/11 10:00 AM	3/3/11 11:00 AM	138	41	114	117	52	120	196
3/3/11 11:00 AM	3/3/11 12:00 PM	243	272	125	125	142	218	216
3/3/11 12:00 PM	3/3/11 1:00 PM	158	193	159	159	138	141	216
3/3/11 1:00 PM	3/3/11 2:00 PM	197	266	159	159	90	149	166
3/3/11 2:00 PM	3/3/11 3:00 PM	194	272	248	248	126	200	126
3/3/11 3:00 PM	3/3/11 4:00 PM	85	224	258	258	31	121	191
3/3/11 4:00 PM	3/3/11 5:00 PM	223	72	148	162	165	190	182
3/3/11 5:00 PM	3/3/11 6:00 PM	164	134	210	212	205	146	56
3/3/11 6:00 PM	3/3/11 7:00 PM	66	184	54	55	202	58	212
3/3/11 7:00 PM	3/3/11 8:00 PM	81	26	72	72	101	80	148
3/3/11 8:00 PM	3/3/11 9:00 PM	28	161	158	156	122	70	177
3/3/11 9:00 PM	3/3/11 10:00 PM	146	265	178	178	181	146	156
3/3/11 10:00 PM	3/3/11 11:00 PM	140	112	188	193	86	113	124
3/3/11 11:00 PM	3/4/11 12:00 AM	248	105	191	181	128	152	165
3/4/11 12:00 AM	3/4/11 1:00 AM	99	114	183	181	101	112	14
3/4/11 1:00 AM	3/4/11 2:00 AM	39	272	159	164	178	61	161
3/4/11 2:00 AM	3/4/11 3:00 AM	15	175	229	229	180	56	135
3/4/11 3:00 AM	3/4/11 4:00 AM	260	137	221	213	85	266	227
3/4/11 4:00 AM	3/4/11 5:00 AM	180	223	202	204	214	180	225
3/4/11 5:00 AM	3/4/11 6:00 AM	257	111	188	190	180	203	144
3/4/11 6:00 AM	3/4/11 7:00 AM	253	165	101	101	201	208	183
3/4/11 7:00 AM	3/4/11 8:00 AM	74	188	190	190	272	71	107
3/4/11 8:00 AM	3/4/11 9:00 AM	176	106	218	218	70	182	124
3/4/11 9:00 AM	3/4/11 10:00 AM	106	64	178	178	79	124	54
3/4/11 10:00 AM	3/4/11 11:00 AM	66	219	97	97	78	57	81
3/4/11 11:00 AM	3/4/11 12:00 PM	32	263	158	158	136	61	81

Time Period		Solution 8	Solution 9	Solution 10	solution 11	Solution 12	Solution 13	Solution 14
From	To							
3/4/11 12:00 PM	3/4/11 1:00 PM	212	243	124	128	29	146	228
3/4/11 1:00 PM	3/4/11 2:00 PM	89	196	170	153	80	94	105
3/4/11 2:00 PM	3/4/11 3:00 PM	241	40	185	181	218	151	70
3/4/11 3:00 PM	3/4/11 4:00 PM	132	85	157	157	181	114	88
3/4/11 4:00 PM	3/4/11 5:00 PM	188	120	197	197	212	206	138
3/4/11 5:00 PM	3/4/11 6:00 PM	189	204	172	172	120	171	135
3/4/11 6:00 PM	3/4/11 7:00 PM	193	84	171	173	98	148	92
3/4/11 7:00 PM	3/4/11 8:00 PM	220	136	105	106	141	208	128
3/4/11 8:00 PM	3/4/11 9:00 PM	85	207	134	134	155	115	142
3/4/11 9:00 PM	3/4/11 10:00 PM	180	206	181	181	181	150	63
3/4/11 10:00 PM	3/4/11 11:00 PM	39	168	272	241	96	75	70
3/4/11 11:00 PM	3/5/11 12:00 AM	33	238	152	154	253	84	170
3/5/11 12:00 AM	3/5/11 1:00 AM	127	142	152	154	175	160	156
3/5/11 1:00 AM	3/5/11 2:00 AM	39	255	197	198	186	117	118
3/5/11 2:00 AM	3/5/11 3:00 AM	49	191	197	201	218	67	65
3/5/11 3:00 AM	3/5/11 4:00 AM	35	180	143	143	109	126	105
3/5/11 4:00 AM	3/5/11 5:00 AM	109	127	76	69	158	70	66
3/5/11 5:00 AM	3/5/11 6:00 AM	116	93	128	128	107	128	116
3/5/11 6:00 AM	3/5/11 7:00 AM	110	130	91	92	73	110	104
3/5/11 7:00 AM	3/5/11 8:00 AM	113	196	157	181	68	74	155
3/5/11 8:00 AM	3/5/11 9:00 AM	217	166	230	230	250	190	126
3/5/11 9:00 AM	3/5/11 10:00 AM	44	272	242	242	142	83	124
3/5/11 10:00 AM	3/5/11 11:00 AM	181	241	70	70	181	139	177
3/5/11 11:00 AM	3/5/11 12:00 PM	112	272	183	181	93	85	154
3/5/11 12:00 PM	3/5/11 1:00 PM	224	89	243	243	197	188	116
3/5/11 1:00 PM	3/5/11 2:00 PM	230	89	67	72	16	192	192
3/5/11 2:00 PM	3/5/11 3:00 PM	205	179	197	196	127	166	146
3/5/11 3:00 PM	3/5/11 4:00 PM	42	247	214	180	14	87	58
3/5/11 4:00 PM	3/5/11 5:00 PM	25	97	220	220	60	58	138
3/5/11 5:00 PM	3/5/11 6:00 PM	202	269	258	258	42	197	96
3/5/11 6:00 PM	3/5/11 7:00 PM	94	73	70	70	236	124	158
3/5/11 7:00 PM	3/5/11 8:00 PM	177	95	192	192	39	147	156
3/5/11 8:00 PM	3/5/11 9:00 PM	32	178	190	181	44	62	94
3/5/11 9:00 PM	3/5/11 10:00 PM	60	79	128	127	48	35	74
3/5/11 10:00 PM	3/5/11 11:00 PM	200	131	143	148	223	170	127
3/5/11 11:00 PM	3/6/11 12:00 AM	63	229	158	159	235	111	188
3/6/11 12:00 AM	3/6/11 1:00 AM	171	266	159	158	213	180	140

Time Period		Solution 8	Solution 9	Solution 10	solution 11	Solution 12	Solution 13	Solution 14
From	To							
3/6/11 1:00 AM	3/6/11 2:00 AM	121	155	185	221	84	103	181
3/6/11 2:00 AM	3/6/11 3:00 AM	255	97	97	97	29	216	251
3/6/11 3:00 AM	3/6/11 4:00 AM	64	208	59	67	252	43	162
3/6/11 4:00 AM	3/6/11 5:00 AM	180	153	143	144	252	178	108
3/6/11 5:00 AM	3/6/11 6:00 AM	120	157	157	158	209	84	144
3/6/11 6:00 AM	3/6/11 7:00 AM	249	19	165	174	27	184	40
3/6/11 7:00 AM	3/6/11 8:00 AM	33	131	161	163	43	19	94
3/6/11 8:00 AM	3/6/11 9:00 AM	178	74	139	143	115	162	82
3/6/11 9:00 AM	3/6/11 10:00 AM	114	213	180	180	228	159	132
3/6/11 10:00 AM	3/6/11 11:00 AM	200	240	270	265	204	202	166
3/6/11 11:00 AM	3/6/11 12:00 PM	192	272	222	224	50	210	153
3/6/11 12:00 PM	3/6/11 1:00 PM	178	190	242	240	154	170	236
3/6/11 1:00 PM	3/6/11 2:00 PM	123	175	213	224	91	111	101
3/6/11 2:00 PM	3/6/11 3:00 PM	194	254	208	208	27	160	150
3/6/11 3:00 PM	3/6/11 4:00 PM	28	170	268	252	181	126	110
3/6/11 4:00 PM	3/6/11 5:00 PM	171	105	128	133	240	138	134
3/6/11 5:00 PM	3/6/11 6:00 PM	115	119	178	178	206	112	200
3/6/11 6:00 PM	3/6/11 7:00 PM	88	117	211	206	268	106	72
3/6/11 7:00 PM	3/6/11 7:00 PM	87	227	150	149	220	70	163
3/6/11 7:00 PM	3/6/11 8:00 PM	133	263	178	181	252	101	159
3/6/11 8:00 PM	3/6/11 9:00 PM	191	129	77	77	76	177	125
3/6/11 9:00 PM	3/6/11 10:00 PM	264	272	182	181	106	219	197
3/6/11 10:00 PM	3/6/11 11:00 PM	229	66	101	103	72	215	109
3/6/11 11:00 PM	3/7/11 12:00 AM	86	99	178	178	48	92	212
3/7/11 12:00 AM	3/7/11 1:00 AM	140	157	196	196	50	122	84
3/7/11 1:00 AM	3/7/11 2:00 AM	262	166	223	225	134	215	215
3/7/11 2:00 AM	3/7/11 3:00 AM	153	246	202	201	181	117	160
3/7/11 3:00 AM	3/7/11 4:00 AM	245	185	199	178	216	128	117
3/7/11 4:00 AM	3/7/11 5:00 AM	13	197	168	169	81	61	172
3/7/11 5:00 AM	3/7/11 6:00 AM	54	30	121	121	27	99	147
3/7/11 6:00 AM	3/7/11 7:00 AM	123	49	262	265	27	120	145
3/7/11 7:00 AM	3/7/11 8:00 AM	22	201	178	178	187	58	95
3/7/11 8:00 AM	3/7/11 9:00 AM	97	210	245	245	258	133	197
3/7/11 9:00 AM	3/7/11 10:00 AM	148	163	181	181	272	108	64
3/7/11 10:00 AM	3/7/11 11:00 AM	118	52	180	180	130	148	127
3/7/11 11:00 AM	3/7/11 12:00 PM	167	233	180	180	233	182	191
3/7/11 12:00 PM	3/7/11 1:00 PM	214	272	181	181	178	145	109

Time Period		Solution 8	Solution 9	Solution 10	solution 11	Solution 12	Solution 13	Solution 14
From	To							
3/7/11 1:00 PM	3/7/11 2:00 PM	130	151	134	139	83	141	92
3/7/11 2:00 PM	3/7/11 3:00 PM	86	197	207	208	117	75	130
3/7/11 3:00 PM	3/7/11 4:00 PM	202	257	258	272	180	201	136
3/7/11 4:00 PM	3/7/11 5:00 PM	199	180	207	220	184	181	115
3/7/11 5:00 PM	3/7/11 6:00 PM	224	168	202	202	270	212	246
3/7/11 6:00 PM	3/7/11 7:00 PM	61	268	213	213	126	62	72
3/7/11 7:00 PM	3/7/11 8:00 PM	168	36	96	96	210	165	200
3/7/11 8:00 PM	3/7/11 9:00 PM	79	163	69	93	272	58	113
3/7/11 9:00 PM	3/7/11 10:00 PM	131	77	144	159	266	114	110
3/7/11 10:00 PM	3/7/11 11:00 PM	73	196	119	128	253	109	147
3/7/11 11:00 PM	3/8/11 12:00 AM	52	59	119	119	232	109	195
3/8/11 12:00 AM	3/8/11 1:00 AM	138	211	76	76	77	135	142
3/8/11 1:00 AM	3/8/11 2:00 AM	153	214	110	118	201	123	127
3/8/11 2:00 AM	3/8/11 3:00 AM	79	171	108	108	169	104	130
3/8/11 3:00 AM	3/8/11 4:00 AM	103	272	97	108	198	142	80
3/8/11 4:00 AM	3/8/11 5:00 AM	198	272	192	181	162	197	155
3/8/11 5:00 AM	3/8/11 6:00 AM	193	192	101	117	181	181	93
3/8/11 6:00 AM	3/8/11 7:00 AM	178	272	110	110	169	142	186
3/8/11 7:00 AM	3/8/11 8:00 AM	228	252	178	178	85	207	156
3/8/11 8:00 AM	3/8/11 9:00 AM	53	90	159	130	68	95	94
3/8/11 9:00 AM	3/8/11 10:00 AM	99	181	145	148	185	102	91

Table C.7: Operating-Modes for the Solution (15) Chosen in the Case Study

Time Period		Solution 15
From	To	
3/1/11 11:00 AM	3/1/11 12:00 PM	154
3/1/11 12:00 PM	3/1/11 1:00 PM	230
3/1/11 1:00 PM	3/1/11 2:00 PM	100
3/1/11 2:00 PM	3/1/11 3:00 PM	161
3/1/11 3:00 PM	3/1/11 4:00 PM	70
3/1/11 4:00 PM	3/1/11 5:00 PM	153
3/1/11 5:00 PM	3/1/11 6:00 PM	132
3/1/11 6:00 PM	3/1/11 7:00 PM	163
3/1/11 7:00 PM	3/1/11 8:00 PM	162
3/1/11 8:00 PM	3/1/11 9:00 PM	152
3/1/11 9:00 PM	3/1/11 10:00 PM	182
3/1/11 10:00 PM	3/1/11 11:00 PM	173
3/1/11 11:00 PM	3/2/11 12:00 AM	67
3/2/11 12:00 AM	3/2/11 1:00 AM	170
3/2/11 1:00 AM	3/2/11 2:00 AM	197
3/2/11 2:00 AM	3/2/11 3:00 AM	44
3/2/11 3:00 AM	3/2/11 4:00 AM	82
3/2/11 4:00 AM	3/2/11 5:00 AM	209
3/2/11 5:00 AM	3/2/11 6:00 AM	123
3/2/11 6:00 AM	3/2/11 7:00 AM	236
3/2/11 7:00 AM	3/2/11 8:00 AM	163
3/2/11 8:00 AM	3/2/11 9:00 AM	119
3/2/11 9:00 AM	3/2/11 10:00 AM	189
3/2/11 10:00 AM	3/2/11 11:00 AM	188
3/2/11 11:00 AM	3/2/11 12:00 PM	23
3/2/11 12:00 PM	3/2/11 1:00 PM	39
3/2/11 1:00 PM	3/2/11 2:00 PM	204
3/2/11 2:00 PM	3/2/11 3:00 PM	133
3/2/11 3:00 PM	3/2/11 4:00 PM	121
3/2/11 4:00 PM	3/2/11 5:00 PM	72
3/2/11 5:00 PM	3/2/11 6:00 PM	114
3/2/11 6:00 PM	3/2/11 7:00 PM	149
3/2/11 7:00 PM	3/2/11 8:00 PM	202
3/2/11 8:00 PM	3/2/11 9:00 PM	181
3/2/11 9:00 PM	3/2/11 10:00 PM	208
3/2/11 10:00 PM	3/2/11 11:00 PM	199

Time Period		Solution 15
From	To	
3/2/11 11:00 PM	3/3/11 12:00 AM	105
3/3/11 12:00 AM	3/3/11 1:00 AM	152
3/3/11 1:00 AM	3/3/11 2:00 AM	134
3/3/11 2:00 AM	3/3/11 3:00 AM	173
3/3/11 3:00 AM	3/3/11 4:00 AM	145
3/3/11 4:00 AM	3/3/11 5:00 AM	110
3/3/11 5:00 AM	3/3/11 6:00 AM	170
3/3/11 6:00 AM	3/3/11 7:00 AM	144
3/3/11 7:00 AM	3/3/11 8:00 AM	107
3/3/11 8:00 AM	3/3/11 9:00 AM	118
3/3/11 9:00 AM	3/3/11 10:00 AM	11
3/3/11 10:00 AM	3/3/11 11:00 AM	132
3/3/11 11:00 AM	3/3/11 12:00 PM	171
3/3/11 12:00 PM	3/3/11 1:00 PM	151
3/3/11 1:00 PM	3/3/11 2:00 PM	170
3/3/11 2:00 PM	3/3/11 3:00 PM	151
3/3/11 3:00 PM	3/3/11 4:00 PM	180
3/3/11 4:00 PM	3/3/11 5:00 PM	124
3/3/11 5:00 PM	3/3/11 6:00 PM	95
3/3/11 6:00 PM	3/3/11 7:00 PM	147
3/3/11 7:00 PM	3/3/11 8:00 PM	139
3/3/11 8:00 PM	3/3/11 9:00 PM	137
3/3/11 9:00 PM	3/3/11 10:00 PM	179
3/3/11 10:00 PM	3/3/11 11:00 PM	203
3/3/11 11:00 PM	3/4/11 12:00 AM	143
3/4/11 12:00 AM	3/4/11 1:00 AM	88
3/4/11 1:00 AM	3/4/11 2:00 AM	127
3/4/11 2:00 AM	3/4/11 3:00 AM	157
3/4/11 3:00 AM	3/4/11 4:00 AM	233
3/4/11 4:00 AM	3/4/11 5:00 AM	141
3/4/11 5:00 AM	3/4/11 6:00 AM	272
3/4/11 6:00 AM	3/4/11 7:00 AM	208
3/4/11 7:00 AM	3/4/11 8:00 AM	130
3/4/11 8:00 AM	3/4/11 9:00 AM	127
3/4/11 9:00 AM	3/4/11 10:00 AM	84
3/4/11 10:00 AM	3/4/11 11:00 AM	161
3/4/11 11:00 AM	3/4/11 12:00 PM	74

Time Period		Solution 15
From	To	
3/4/11 12:00 PM	3/4/11 1:00 PM	107
3/4/11 1:00 PM	3/4/11 2:00 PM	26
3/4/11 2:00 PM	3/4/11 3:00 PM	182
3/4/11 3:00 PM	3/4/11 4:00 PM	126
3/4/11 4:00 PM	3/4/11 5:00 PM	158
3/4/11 5:00 PM	3/4/11 6:00 PM	122
3/4/11 6:00 PM	3/4/11 7:00 PM	81
3/4/11 7:00 PM	3/4/11 8:00 PM	160
3/4/11 8:00 PM	3/4/11 9:00 PM	241
3/4/11 9:00 PM	3/4/11 10:00 PM	107
3/4/11 10:00 PM	3/4/11 11:00 PM	112
3/4/11 11:00 PM	3/5/11 12:00 AM	121
3/5/11 12:00 AM	3/5/11 1:00 AM	151
3/5/11 1:00 AM	3/5/11 2:00 AM	101
3/5/11 2:00 AM	3/5/11 3:00 AM	25
3/5/11 3:00 AM	3/5/11 4:00 AM	52
3/5/11 4:00 AM	3/5/11 5:00 AM	76
3/5/11 5:00 AM	3/5/11 6:00 AM	141
3/5/11 6:00 AM	3/5/11 7:00 AM	117
3/5/11 7:00 AM	3/5/11 8:00 AM	207
3/5/11 8:00 AM	3/5/11 9:00 AM	106
3/5/11 9:00 AM	3/5/11 10:00 AM	150
3/5/11 10:00 AM	3/5/11 11:00 AM	203
3/5/11 11:00 AM	3/5/11 12:00 PM	156
3/5/11 12:00 PM	3/5/11 1:00 PM	209
3/5/11 1:00 PM	3/5/11 2:00 PM	73
3/5/11 2:00 PM	3/5/11 3:00 PM	97
3/5/11 3:00 PM	3/5/11 4:00 PM	70
3/5/11 4:00 PM	3/5/11 5:00 PM	4
3/5/11 5:00 PM	3/5/11 6:00 PM	149
3/5/11 6:00 PM	3/5/11 7:00 PM	70
3/5/11 7:00 PM	3/5/11 8:00 PM	111
3/5/11 8:00 PM	3/5/11 9:00 PM	91
3/5/11 9:00 PM	3/5/11 10:00 PM	119
3/5/11 10:00 PM	3/5/11 11:00 PM	215
3/5/11 11:00 PM	3/6/11 12:00 AM	84
3/6/11 12:00 AM	3/6/11 1:00 AM	104

Time Period		Solution 15
From	To	
3/6/11 1:00 AM	3/6/11 2:00 AM	106
3/6/11 2:00 AM	3/6/11 3:00 AM	224
3/6/11 3:00 AM	3/6/11 4:00 AM	110
3/6/11 4:00 AM	3/6/11 5:00 AM	144
3/6/11 5:00 AM	3/6/11 6:00 AM	24
3/6/11 6:00 AM	3/6/11 7:00 AM	150
3/6/11 7:00 AM	3/6/11 8:00 AM	52
3/6/11 8:00 AM	3/6/11 9:00 AM	195
3/6/11 9:00 AM	3/6/11 10:00 AM	162
3/6/11 10:00 AM	3/6/11 11:00 AM	174
3/6/11 11:00 AM	3/6/11 12:00 PM	144
3/6/11 12:00 PM	3/6/11 1:00 PM	161
3/6/11 1:00 PM	3/6/11 2:00 PM	115
3/6/11 2:00 PM	3/6/11 3:00 PM	199
3/6/11 3:00 PM	3/6/11 4:00 PM	157
3/6/11 4:00 PM	3/6/11 5:00 PM	167
3/6/11 5:00 PM	3/6/11 6:00 PM	136
3/6/11 6:00 PM	3/6/11 7:00 PM	143
3/6/11 7:00 PM	3/6/11 7:00 PM	60
3/6/11 7:00 PM	3/6/11 8:00 PM	75
3/6/11 8:00 PM	3/6/11 9:00 PM	175
3/6/11 9:00 PM	3/6/11 10:00 PM	272
3/6/11 10:00 PM	3/6/11 11:00 PM	161
3/6/11 11:00 PM	3/7/11 12:00 AM	209
3/7/11 12:00 AM	3/7/11 1:00 AM	156
3/7/11 1:00 AM	3/7/11 2:00 AM	272
3/7/11 2:00 AM	3/7/11 3:00 AM	148
3/7/11 3:00 AM	3/7/11 4:00 AM	210
3/7/11 4:00 AM	3/7/11 5:00 AM	150
3/7/11 5:00 AM	3/7/11 6:00 AM	90
3/7/11 6:00 AM	3/7/11 7:00 AM	57
3/7/11 7:00 AM	3/7/11 8:00 AM	62
3/7/11 8:00 AM	3/7/11 9:00 AM	101
3/7/11 9:00 AM	3/7/11 10:00 AM	83
3/7/11 10:00 AM	3/7/11 11:00 AM	124
3/7/11 11:00 AM	3/7/11 12:00 PM	79
3/7/11 12:00 PM	3/7/11 1:00 PM	170

Time Period		Solution 15
From	To	
3/7/11 1:00 PM	3/7/11 2:00 PM	102
3/7/11 2:00 PM	3/7/11 3:00 PM	107
3/7/11 3:00 PM	3/7/11 4:00 PM	239
3/7/11 4:00 PM	3/7/11 5:00 PM	79
3/7/11 5:00 PM	3/7/11 6:00 PM	176
3/7/11 6:00 PM	3/7/11 7:00 PM	48
3/7/11 7:00 PM	3/7/11 8:00 PM	143
3/7/11 8:00 PM	3/7/11 9:00 PM	130
3/7/11 9:00 PM	3/7/11 10:00 PM	166
3/7/11 10:00 PM	3/7/11 11:00 PM	66
3/7/11 11:00 PM	3/8/11 12:00 AM	185
3/8/11 12:00 AM	3/8/11 1:00 AM	93
3/8/11 1:00 AM	3/8/11 2:00 AM	140
3/8/11 2:00 AM	3/8/11 3:00 AM	106
3/8/11 3:00 AM	3/8/11 4:00 AM	166
3/8/11 4:00 AM	3/8/11 5:00 AM	221
3/8/11 5:00 AM	3/8/11 6:00 AM	95
3/8/11 6:00 AM	3/8/11 7:00 AM	185
3/8/11 7:00 AM	3/8/11 8:00 AM	208
3/8/11 8:00 AM	3/8/11 9:00 AM	94
3/8/11 9:00 AM	3/8/11 10:00 AM	72

Bibliography

[1] Q. S. M. Yang, D. Guo-Sha, and Y. Ji- Luo, "A Novel Evolution Strategy for Multi-objective Optimization Problems," *Applied Mathematics and Computation*, vol. 170, p. 850–873, 2005.

[2] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial Intelligence Through a Simulation of Evolution," in *In Proceddings of Biophysics and Cybernetic Sciences Symposium*, 1965, pp. 131-155.

[3] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*. Addison Wesley, 1989.

[4] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Michigan: University of Michigan Press, 1975.

[5] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," *Global Optimization*, vol. 11, p. 341–359, 1997.

[6] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies," in *In Proceedings of Actes de la Première Conférence Européenne Sur la Vie Artificielle*, Paris, 1991, pp. 134-142.

[7] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in *Proceeding of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.

[8] B. B. Li, L. Wang, and B. Liu, "An Effective PSO-based Hybrid Algorithm for Multi-objective Permutation Slow shop Scheduling," *IEEE Transition System, Man, Cybern. A, System, Humans*, vol. 38, no. 4, p. 818–831, 2008.

[9] M. J. Sasena, P. Papalambros, and P. Goovaerts, "Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization," *Engineering Optimization*, vol. 34, p. 263–278, 2002.

[10] A. Prudius and S. Andradóttir, "Simulation Optimization Using Balanced Explorative and Exploitative Search," in *In Proceedings of the Winter Simulation*, Washington DC, 2004, p. 545–549.

[11] A. Auger, M. Schoenauer, and O. Teytaud, "Local and Global Order 3/2 Convergence of a Surrogate Evolutionary Algorithm," in *In Proceedings of the Genetic Evolutionary Computation*, Washington DC, 2005, p. 857–864.

[12] Pump Systems Matter, *Optimizing Pumping Systems: A Guide for Improved Energy, Efficiency, Reliability and Profitability*. Hydraulic Institute, 2008.

[13] B. S. Jung, B. W. Karney, P. F. Bolous, and D. J. Wood, "The Need for Comprehensive Transient Analysis of Distribution Systems," *American Water Work Association*, pp. 112-123, 2007.

[14] P. F. Bolous, B. W. Karney, D. J. Wood, and S. Lingireddy, "Hydraulic Transient Guidelines for Protecting Water Distribution Systems," *American Water Work Associationand*, pp. 111-124, 2005.

[15] T. Ming-Yen, T. C. Frank, Tsai, and W. G. William, "Optimization of Water Distribution and Water Quality by Hybrid Genetic Algorithm," *Water Resources Planning and Management, ACSE,*, pp. 431-440, 2005.

[16] C. Darwin , *On the Origin of Species by Means of Natural Selection*, 1st ed. London, UK: Murray, 1859.

[17] A. I. F. Vaz and L. N. Vicents, "A Particle Swarm Pattern Search Method for Bound Constrained Global Optimization," *Global Optimization*, vol. 39, pp. 197-219, 2007.

[18] I. Osman and J. Kelly, *Meta-Heuristics: Theory and Applications*. Dordrecht, Netherlands: Kluwer Academic, 1996.

[19] G. Dueck and T. Scheuer, "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing," *Computer Science*, vol. 90, pp. 161-175, 1990.

[20] F. Glover, "Tabu Search – Part I," *ORSA Journal on Computing*, vol. 1, pp. 190-206, 1989.

[21] M. L. Eusuff and F. K. Pasha, "Shuffled Frog-Leaping Algorithm: a Memetic Meta-heuristic for Discrete Optimization," *Engineering Optimization*, vol. 38, pp. 129-154, 2006.

[22] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new Heuristic Optimization Algorithm: Harmony Search," *Simulation*, vol. 76, pp. 60-68, 2001.

[23] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. New York: Springer-Verlag, 2004.

[24] F. Glover, M. Laguna, and R. Martí , "Fundamentals of Scatter Search and Path Relinking," *Control and Cybernetics*, vol. 39, pp. 653-684, 2000.

[25] J. Geddes, "Modern Heuristics," The University of Reading Thesis, 2010.

[26] M. Gilli, "An Intorduction to Optimization Heuristics," *Optimization Heuristic*, vol. 9, pp. 1-93, 2004.

[27] R. Poli, J. Kennedy, and T. Blackwell, "Particle Swarm Optimization: An Overview," *Swarm Intelligent*, vol. 1, pp. 33-57, 2007.

[28] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2004.

[29] C. A. C. Coello and M. S. Lechuga, "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 2, 2002, pp. 1051-1056.

[30] M. Millonas, "Swarms, Phase Transitions, and Collective Intelligence," *Artificial Life III*, 1994.

[31] J. D. Schaffer, L. J. Eshelman, and D. Offutt, "Spurious Correlations and Premature Convergence in Genetic Algorithms," in *In Proceedings of the 1st Workshop on Foundations of Genetic Algorithms*, 1990, p. 102–112.

[32] R. K. Ursem, "Models for Evolutionary Algorithms and their Applications in System Identification and Control Optimization," PhD thesis, Department of Computer Science, University of Aarhus, 2003.

[33] W. Thomas, Z. Michael, C. Raymond, and J. Antonio, "Why Is Optimization Difficult?," *Nature-Inspired Algorithms for Optimization*, vol. 193, pp. 1-50, 2009.

[34] I. Paenke, J. Branke, and Y. Jin, "On the Influence of Phenotype Plasticity on Genotype Diversity," in *In Proceedings of First IEEE Symposium on Foundations of Computational Intelligence*, 2007, pp. 33-40.

[35] R. D. Routledge, "Diversity Indices, Which ones are admissible?," *Theoretical Biology*, vol. 76, p. 503–515, 1979.

[36] A. E. Magurran, "Biological Diversity," *Current Biology Magazine*, vol. 15, p. 116–118, 2005.

[37] R. W. Morrison and K. A. De-Jong, "Measurement of Population Diversity," in *In Proceediings of the 5th European Conference on Artificial Evolution*, vol. 2310, 2002, pp. 1047-1074.

[38] D. N. Wilke, S. Kok, and A. A. Groenwold, "Comparison of Linear and Classical Velocity Update Rules in Particle Swarm Optimization: Notes on Diversity," *International Numerical Methods in Engineering*, vol. 70, pp. 962-984, 2007.

[39] T. Krink, J. S. Vesterstrom, and J. Riget, "Particle Swarm Optimization with Spatial Particle Extension," in *In Proceedings of IEEE Congress on Evolutionary Computation*, Honolulu, Hawaii, USA, 2002, pp. 1474-1479.

[40] Y. Shi and R. C. Eberhart, "A Modified Particle Swarm Optimizer," in *In Proceedings of IEEE International Conference on Evolutionary Computation*, 1998, pp. 4-9.

[41] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," Evolutionary Computation Group, EVOCINV, Technical Report, 2006.

[42] S. S. Madeiro, C. J. A. Bastos-Filho, F. B. Neto, and E. M. N. Figueiredo, "Adaptative Clustering Particle Swarm Optimization," in *Proceedings of IEEE International Conference on Parallel and Distributed Processing*, 2009.

[43] J. Kennedy and R. Mendes, "Population Structure and Particle Swarm Performance," in *In Proceedings of IEEE Congress on Evolutionary Computation*, vol. 2, 2002, pp. 1671-1676.

[44] D. F. Carvalho and C. J. A. Bastos-Filho, "Clan Particle Swarm Optimization," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 3044-3051.

[45] P. N. Sugagthan, "Particle Swarm optimizer with Neighborhood Operator," in *In Proceedings of the Congress of Evolutionary Computational*, vol. 3, 1999, pp. 1958-1962.

[46] L. B. Zhang, et al., "Solving Multi-objective Optimization Problems Using Particle Swarm Optimization,"

in *Proceedings of IEEE Congress on Evolutionary Computation*, 2003, pp. 2400-2405.

[47] J. Lis and A. E. Eiben, "A Multi-Sexual Genetic Algorithm for Multi-objective Optimization," in *In Proceedings of the International Conference on Evolutionary Computation*, 1996, pp. 59-64.

[48] G. Toscano-Pulido, C. A. C. Coello, and L. V. Santana-Quintero, "EMOPSO: A Multi-objective Particle Swarm Optimizer with Emphasis on Efficiency," in *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization*, vol. 4403,, 2007, pp. 272-285.

[49] A. J. Nebro, et al., "SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization," in *Proceedings of IEEE Symposium on Computational Intelligence in Multi-criteria Decision-Making*, 2009, pp. 66-73.

[50] Y. Qin and W. Danyang, "An Improved Particle Swarm Optimization," in *Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics*, 2009.

[51] M. Reyes-Sierra and C. A. C. Coello, "On-line Adaptation in Multi-objective Particle Swarm Optimization," in *Proceedings of IEEE Press in Swarm Intelligence Symposium*, 2006, pp. 61-68.

[52] C. A. C. Coello, G. Toscano-Pulido, and M. S. Lechuga, "Handling Multiple Objectives with Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 256-279, 2004.

[53] T. Bartz-Beielstein, et al., "Particle Swarm Optimizers for Pareto Optimization with Enhanced Archiving Techniques," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 3, 2003, pp. 1780-1787.

[54] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "A Particle Swarm Optimizer for Constrained Numerical Optimization," in *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature*, vol. 4193, 2006, pp. 910-919.

[55] J. E. Alvarez-Benitez, R. M. Everson, and J. E. Fieldsend, "A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts," in *Proceedings of the 3rd International Conference on Evolutionary Multi-criterion Optimization*, vol. 3410, 2005, pp. 459-473.

[56] V. L. Huang, P. N. Suganthan, and J. J. Liang, "Comprehensive Learning Particle Swarm Optimizer for Solving Multi-objective Optimization Problems," *International Journal of Intelligent Systems*, vol. 21, pp. 209-226, 2006.

[57] M. Reyes-Sierra and C. A. C. Coello, "Fitness Inheritance in Multi-objective Particle Swarm Optimization," in *Proceedings of IEEE Swarm Intelligence Symposium*, vol. 5, 2005, pp. 119-126.

[58] G. Dun-Wei, Z. Yong, and Z. Jian-Hua, "Multi-objective Particle Swarm Optimization Based on Minimal Particle Angle," in *Proceedings of the International Conference on Advances Intelligent Computing*, vol. 3644, 2005, pp. 571-580.

[59] M. Reyes-Sierra and C. A. C. Coello, "A Study of Techniques to Improve the Efficiency of a Multi-objective Particle Swarm Optimizer," *Evolutionary Computation in Dynamic and Uncertain Environments*, pp. 269-296, 2007.

[60] X. H. Zhang, H. Y. Meng, and L. C. Jiao, "Intelligent Particle Swarm Optimization in Multi-objective Optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 714-719.

[61] L. Zhen-Su, H. Zhi-Rong, and D. Huan, "Particle Swarm Optimization with Adaptive Mutation," *Frontiers of Electrical and Electronic Engineering in China*, vol. 1, pp. 99-104, 2006.

[62] A. Chatterjee and P. Siarry, "Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization," *Computers and Operations Research*, vol. 33, pp. 859-871, 2006.

[63] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, p. 281–295, 2006.

[64] J. Kennedy, "Bare Bones Particle Swarms," in *In Proceedings of the IEEE Swarm Intelligence Symposium*, 2003, pp. 80-87.

[65] C. K. Monson and K. D. Seppi, "The Kalman Swarm," in *In Proceedings of the Genetic and Evolutionary Computation Conference*, 2004.

[66] M. Mahfouf, M. Y. Chen, and D. A. Linkens, "Adaptive Weighted Particle Swarm Optimization for Multi-objective Optimal Design of Alloy Steels," *Parallel Problem Solving from Nature*, vol. 3242, pp. 762-771,

2004.

[67] V. F. Bergh and A. P. Engelbrecht, "A New Locally Convergent Particle Swarm Optimizer," in *In Proceedings of IEEE Conference on Systems, Man and Cybernetics*, 2002.

[68] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing Hierarchical Particle Swarm optimizer with Time-varying Acceleration Coefficients," *IEEE Trans Evolutionary Computation,*, vol. 8, pp. 240-255, 2004.

[69] J. Kennedy and R. C. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm," in *In Proceedings of IEEE International Conference on In Systems, Man, and Cybernetics*, vol. 5, 4104-4108, p. 1997.

[70] B. Al-Kazemi and C. K. Mohan, "Multi-phase Discrete Particle Swarm Optimization," in *In Proceedings of 4th International Workshop on Frontiers in Evolutionary Algorithms*, 2002.

[71] C. K. Monson and K. D. Seppi, "Bayesian Optimization Models for Particle Swarm," in *in Proceedings of Genetic and Evolutionary Computation Conference*, 2005.

[72] M. Reyes-Sierra and C. A. C. Coello, "Improving PSO-based Multi-objective Optimization using Crowding, Mutation and E-dominance," in *In Proceedings of 3rd International Conference on Evolutionary Multi-criterion Optimization*, 2005, pp. 505-519.

[73] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," *IEEE Transactions of Evolutionary Computation*, vol. 6, pp. 182-197, 2002.

[74] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multi-objective Optimization," in *In Proceedings of the EUROGEN Conference*, 2002, p. 95–100.

[75] C. A. C. Coello and G. T. Pulido, "Multi-objective Optimization using a micro-Genetic Algorithm," in *In Proceedings of the Genetic and Evolutionary Computation Conference*, 2001, pp. 274-282.

[76] J. D. Knowles and D. W. Corne, "Approximating the Non-dominated Front using the Pareto Archived Evolution Strategy," *IEEE Transaction and Evolutionary Computation*, vol. 8, pp. 149-172, 2000.

[77] C. F. C. Juan and C. A. C. Coello, "Micro-MOPSO: A Multi-objective Particle Swarm Optimizer that Uses a Very Small Population Size," *Multi-objective Swarm Intelligent Systems*, 2010.

[78] J. E. Fieldsend and S. Singh, "A Multi-objective Algorithm based Upon Particle Swarm Optimization, an Efficient Data Structure and Turbulence," in *In Proceedings of the Computational Intelligence Workshop*, 2002, pp. 37-44.

[79] X. D. Li, "A Non-dominated Sorting Particle Swarm Optimizer for Multi-objective Optimization," in *In Proceedings of Genetic and Evolutionary Computation*, 2003, pp. 37-48.

[80] M. Jacqueline and C. Richard, "Application of Particle Swarm to Multi-objective Optimization," Auburn University - Department of Computer Science,, Technical Report, 1999.

[81] M. Reyes-Sierra and C. A. C. Coello, "A New Multi-objective Particle Swarm Optimizer with Improved Selection and Diversity Mechanisms," EVOCINV, Technical Report, 2004.

[82] U. Baumgartner, C. Magele, and W. Renhart, "Pareto Optimality and Particle Swarm Optimization," *IEEE Transactions on Magnetics*, vol. 40, pp. 1172-1175, 2004.

[83] G. Toscano-Pulido and C. A. C. Coello, "Using Clustering Techniques to Improve The Performance of A Multi-objective Particle Swarm Optimizer," in *Proceedings of the Genetic and Evolutionary Computation*, vol. 3102, 2004, pp. 225-237.

[84] P. Alessandro and S. Antonina, "Particle Swarm Optimization for Multimodal Functions: A Clustering Approach," *Artificial Evolution and Applications*, pp. 1-15, 2008.

[85] X. Li, "Adaptively Choosing Neighbourhood Bests Using Species in a Particle swarm Optimizer for Multimodal Function Optimization," in *In Proceedings of the Conference on Genetic and Evolutionary Computation*, vol. 3102, 2004, pp. 105-116.

[86] S. Bird and X. Li, "Adaptively Choosing Niching Parameters in a PSO," in *In Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, vol. 1, 2006, pp. 3-10.

[87] S. Das, A. Abraham, and A. Konar, "Automatic Kernel Clustering with A Multi-elitist Particle Swarm Optimization Algorithm," *Pattern Recognition*, vol. 29, pp. 688-699, 2008.

[88] M. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic Clustering Using Particle Swarm Optimization with Application in Unsupervised Image Classification," in *In Proceedings of the 5th World Enformatika Conference*, 2005.

[89] S. Mostaghim and J. Teich, "Covering Pareto-Optimal Fronts by Subswarms in Multi-objective Particle Swarm Optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1404-1411.

[90] O. Schutze, S. Mostaghim, M. Dellnik, and L. Teich, "Covering Pareto Sets by Multilevel Evolutionary Subdivision Techniques," in *In Proceedings of 2nd International Conference on Evolutionary Multi-Criterion Optimization*, 2003, pp. 118-132.

[91] H. Sheng-Ta, S. Tsung-Ying, C. Shih-Yuan, L. Chan-Cheng, and L. Cheng-Wei, "Cluster Based Solution Exploration Strategy for Multi-objective Particle Swarm Optimization," in *Proceedings of the 25th International Multi-Conference: Artificial Intelligence and Applications*, 2007, pp. 295-300.

[92] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multi-objective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, pp. 173-195, 2000.

[93] C. Xiaohui and E. P. Thomas, "Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm," in *Proceedings of IEEE Swarm Intelligence Symposium*, vol. 8, 2005, pp. 185-191.

[94] A. Passaro and A. Starita, "Clustering Particles for Multimodal Function Optimization," in *Proceedings of ECAI Workshop on Evolutionary Computation*, 2006.

[95] J. F. Schutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka, and A. D. George, "Parallel Global Optimization with the Particle Swarm Algorithm," *Numerical Methods in Engineering*, vol. 61, pp. 2296-2315, 2004.

[96] R. Brits, A. Engelbrecht, and F. Van-Den-Bergh, "A Niching Particle Swarm Optimizer," in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, vol. 2, 2002, pp. 692-696.

[97] S. K. S. Fan and J. M. Chang, "A Parallel Particle Swarm Optimization Algorithm for Multi-Objective Optimization Problems," *Engineering Optimization*, vol. 41, pp. 673-697, 2009.

[98] K. E. Parsopoulos and M. N. Vrahatis, "Multi-objective Optimization Using Parallel Vector Evaluated Particle Swarm Optimization," in *Proceedings of IASTED International Conference on Artificial Intelligence and Applications*, vol. 2, 2004, pp. 823-828.

[99] D. Schaffer, "Multiple Objective Optimizations with Vector Evaluated Genetic Algorithms," in *In Proceedings of the 1st International Conference on Genetic Algorithms and their Applications*, 1985, pp. 93-100.

[100] K. E. Parsopóulos and M. N. Vrahatis, "Particle Swarm Optimization Method in Multi-objective Problems," in *Proceedings of the ACM Symposium on Applied Computing*, 2002, pp. 603-607.

[101] D. Gies and Y. Rahmat-Samii, "Particle Swarm Optimization for Reconfigurable Phase-differentiated Array Design," *Microwave Opt. Technol.*, vol. 38, pp. 168-175, 2003.

[102] S. Mostaghim, J. Branke, and H. Schmeck, "Multi-objective Particle Swarm Optimization on Computer Grids," in *Proceedings of Genetic and Evolutionary Computation Conference*, 2007, pp. 869-875.

[103] J. Nanbo and R. Yahya, "Parallel Particle Swarm Optimization and Finite Difference Time-domain Algorithm for Multi-band and Wide-band Patch Antenna Designs," *IEEE Transaction on Antenna and Propagation*, vol. 53, 2005.

[104] J. F. Chang, S. C. Chu, J. F. Roddick, and J. S. Pan, "A Parallel Particle Swarm Optimization Algorithm with Communication Strategies," *Information Science and Engineering*, vol. 21, p. 809–818, 2005.

[105] S. C. Esquivel and C. A. C. Coello, "On the Use of Particle Swarm Optimization with Multimodal Functions," *Evolutionary Computation*, vol. 2, pp. 1130-1136, 2003.

[106] J. Stefan and M. Daniel, "A New Multi-objective Particle Swarm Optimization Algorithm Using Clustering Applied to Automated Docking," *Computer Science - Hybrid Metaheuristics*, vol. 3636, pp. 124-141, 2005.

[107] G. M. Morris, et al., "Automated Docking using A Lamarckian Genetic Algorithm and An Empirical Binding Free Energy Function," *Computational Chemistry*, vol. 19, pp. 1639-1662.

[108] H. Shing-Jang, K. Wen-Yuan, J. Jun-Wun, H. Ming-Hao, and H. Shinn-Ying, "Intelligent Particle Swarm Optimization in Multi-objective Problems," *Computer Science - Advances in Knowledge Discovery and*

Data Mining, vol. 3928, pp. 790-800, 2006.

[109] T. Ching-Shih, F. Hsiao-Hua, C. Hsu-Hwa, and K. Chia-Hung, "An Improved Particle Swarm Pareto Optimizer with Local Search and Clustering," in *Proceeding of the Simulated Evolution and Learning (SEAL)*, vol. 4247, 2006, pp. 400-407.

[110] J. Knowles and D. Corne, "The Pareto Archived Evolution Strat Strategy," in *In Proceedings of the Congress on Evolutionary Computation*, Washington D.C., USA, 1999, p. 98–105.

[111] M. Salazar-Lechuga and J. E. Rowe, "Particle Swarm Optimization and Fitness Sharing to Solve Multi-objective Optimization Problems," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 2, 2005, pp. 1204-1211.

[112] E. Zitzler and L. Thiele, "Multi-objective Evolutionary Algorithms: a Comparative Case Study and Strengthen Pareto Approach," *IEEE Transaction on Evolutionary Computation*, vol. 3, pp. 257-271, 1999.

[113] W. R. M. Wickramasinghe and X. Li, "Choosing Leaders for Multi-objective PSO Algorithms Using Differential Evolution," in *Proceedings of the 7th International Conference on Simulated Evolution and Learning*, vol. 5361, 2008, pp. 249-258.

[114] L. V. Santana-Quintero, N. Ramírez-Santiago, and C. A. C. Coello, "Towards a More Efficient Multi-objective Particle Swarm Optimizer," *Multi-objective Optimization in Computational Intelligence: Theory and Practice*, pp. 76-105, 2008.

[115] M. R. Chen, X. Li, X. Zhang, and L. Yong-Zai, "A Novel Particle Swarm Optimizer Hybridized with External Optimization," *Applied Soft Computing*, vol. 10, pp. 367-373, 2010.

[116] H. Liu and A. Abraham, "Fuzzy Adaptive Turbulent Particle Swarm Optimization," in *In Proceedings of 5th International Conference on Hybrid Intelligent Systems*, p. 2005.

[117] R. Poli, W. B. Langdon, and O. Holland, "Extending Particle Swarm Optimization via Genetic Programming," in *In Proceedings of 8th European Conference*, 2005.

[118] M. Jian and Y. Chen, "Introducing Recombination with Dynamic Linkage Discovery to Particle Swarm Optimization," in *In Proceedings of the Genetic and Evolutionary Computation Conference*, 2006, pp. 85-86.

[119] P. J. Angeline, "Using Selection to Improve Particle Swarm Optimization," in *In Proceedings of the IEEE Congress on Evolutionary Computation*, 1998.

[120] Z. W. Geem, "Particle-Swarm Harmony Search for Water Network Design," *Engineering Optimization*, vol. 41, pp. 297-311, 2009.

[121] M. Meissner, M. Schmuker, and G. Schneider, "Optimized Particle Swarm Optimization (OPSO) and its Application to Artificial Neural Networks Training," *BMC Bioinformatics*, vol. 7, 2006.

[122] S. Dipti and S. Tian-Hou, "Particle Swarm Inspired Evolutionary Algorithm (PS-EA) for Multi-objective Optimization Problems," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 4, 2003, pp. 2292-2297.

[123] V. S. Q. Luis, R. S. Noel, C. A. C. Coello, M. L. Julian, and H. D. Alfredo-Garcia, "A New Proposal for Multi-objective Using Particle Swarm Optimization and Rough Sets Theory," *Parallel Problem Solving from Nature - Multi-objective Optimization*, vol. 4193, pp. 483-492, 2006.

[124] T. Krink and M. Løvbjerg, "The Lifecycle Model: Combining Particle Swarm Optimization, Genetic Algorithms and Hill-climbers," in *In Proceedings of Parallel Problem Solving from Nature Conference*, vol. 2439, 2002, pp. 621-630.

[125] W. J. Zhang and X. F. Xie, "DEPSO: Hybrid Particle Swarm with Differential Evolution Operator," in *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2003, pp. 3816-3821.

[126] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle Swarm, Genetic Algorithm, and their Hybrids: Optimization of a Profiled Corrugated Horn Antenna," in *In Proceedings IEEE International Symposium on Antennas and Propagation*, 2002, p. 314–317.

[127] J. S. Vesterstrøm, J. Riget, and T. Krink, "Division of Labor in Particle Swarm Optimization," in *In Proceedings of the IEEE Congress on Evolutionary Computation*, 2002, p. 1570–1575.

[128] T. Hendtlass, "A Combined Swarm Differential Evolution Algorithm for Optimization Problems," in *In Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial*

Intelligence and Expert Systems, 2001, p. 11–18.

[129] H. S. Lopes and L. S. Coelho, "Particle Swarm Optimization with Fast Local Search for the Blind Travelling Salesman Problem," in *In Proceedings of 5th International Conference on Hybrid Intelligent Systems*, 2005, p. 6–9.

[130] J. Habibi, S. A. Zonouz, and M. Saneei, "A Hybrid PS-based Optimization Algorithm for Solving Traveling Salesman Problem," in *In Proceedings of IEEE Symposium on Frontiers in Networking with Applications*, 2006, pp. 18-20.

[131] J. Branke, E. Salihoglu, and S. Uyar, "Towards an Analysis of Dynamic Environments," in *In Proceedings of Genetic and Evolutionary Computation Conference*, 2005, pp. 1433-1440.

[132] J. Branke, "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems," in *In Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 3, 1999, pp. 1875-1882.

[133] J. Branke, "The Moving Peaks Benchmark," Institute AIFB, University of Karlsruhe, Technical Report, 1999.

[134] R. W. Morrison and K. A. De-Jong, "A Test Problem Generator for Non-stationary Environments," in *In Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 3, 1999, pp. 2047-2053.

[135] X. Li, J. Branke, and T. Blackwell, "Particle Swarm with Speciation and Adaptation in a Dynamic Environment," in *In Proceedings of Genetic and Evolutionary Computation Conference*, 2006, p. 51–58.

[136] S. Kiranyaz, J. Pulkkinen, and M. Gabbouj, "Multi-dimensional Particle Swarm Optimization for Dynamic Environments," in *In Proceedings of International Conference on Innovations in Information Technology*, 2008, pp. 34-38.

[137] A. J. Carlisle and G. V. Dozier, "Tracking Changing Extrema with Adaptive Particle Swarm Optimizer," in *In Proceedings of the 5th Biannual World Automation Congress*, vol. 13, 2002, p. 265–270.

[138] G. Pan, Q. Dou, and X. Liu, "Performance of Two Improved Particle Swarm Optimization in Dynamic Optimization Environments," in *In Proceedings of the 6th International Conference on Intelligent Systems Design and Applications*, vol. 2, 2006, pp. 1024-1028.

[139] D. Parrott and X. Li, "A Particle Swarm Model for Tracking Multiple Peaks in a Dynamic Environment Using Speciation," in *In Proceeding of the 2004 Congress on Evolutionary Computation*, 2004, pp. 98-103.

[140] J. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A Species Conserving Genetic Algorithm for Multimodal Function Optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 207-234, 2002.

[141] A. Carlisle and G. Dozier, "Adapting Particle Swarm Optimization to Dynamic Environments," in *In Proceedings of International Conference on Artificial Intelligence*, vol. 1, 2000, pp. 429-434.

[142] J. Kennedy, "The Particle Swarm: Social Adaptation of Knowledge," in *In Proceedings of International Conference Evolutionary Computation*, 1997, pp. 303-308.

[143] T. Peram, K. Veeramachaneni, and C. Mohan, "Fitness-distance Ratio based Particle Swarm Optimization," in *In Proceedings of the IEEE Swarm Intelligence Symposium*, 2003, p. 174–181.

[144] J. J. Liang and P. N. Suganthan, "Dynamic Multi-swarm Particle Swarm Optimizer," in *In Proceedings of the IEEE Swarm Intelligence Symposium*, 2005, p. 124–129.

[145] R. C. Eberhart and Y. Shi, "Tracking and Optimizing Dynamic Systems with Particle Swarms," in *In Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1, 2001, pp. 94-100.

[146] S. Janson and M. Middendorf, "A Hierarchical Particle Swarm Optimizer and its Adaptive Variant," *IEEE Transactions on System Man and Cybernetics B*, vol. 35, p. 1272–1282, 2005.

[147] G. Yen and H. Lu, "Dynamic Population Strategy Assisted Particle Swarm Optimization," in *In Proceedings of IEEE International Symposium on Intelligent Control*, 2003.

[148] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A Regularity Model-based Multi-objective Estimation of Distribution Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 41-63, 2008.

[149] F. Pettersson, N. Chakraborti, and H. Saxén, "A Genetic Algorithms based Multi-objective Neural Net Applied to Noisy Blast Furnace Data," *Applied Soft Computing*, vol. 7, pp. 387-397, 2007.

[150] C. Poloni and V. Pediroda, "GA Coupled with Computationally Expensive Simulations: Tools to Improve

Efficiency," in *In Proceedings of the Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, 1997, p. 267–288.

[151] L. Shao and M. Ehrgott, "Finding Representative Non-dominated Points in Multi-objective Linear Programming," in *In Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision Making*, 2007, p. 245–252.

[152] J. T. Dickersbach, *Supply Chain Management with APO*. Springer, 2005.

[153] O. Bandte and S. Malinchik, "A Broad and Narrow Approach to Interactive Evolutionary Design – an Aircraft Design Example," *GECCO*, vol. 3103, pp. 883-895, 2004.

[154] T. J. Stewart, R. Janssen, and V. M. Herwijnen, "A Genetic Algorithm Approach to Multi-objective Land Use Planning," *Computers and Operations Research*, vol. 32, pp. 2293-2313, 2004.

[155] H. Nakayama, S. Kaneshige, S. Takemoto, and Y. Wata, "An Application of a Multi-objective Programming Technique to Construction Accuracy Control of Cable-stayed Bridges," *European Operational Research*, vol. 87, p. 731–738, 1995.

[156] F. Huiyuan, "A Modification to Particle Swarm Optimization Algorithm," *Engineering Computations*, vol. 19, pp. 970-989, 2002.

[157] C. A. C. Coello, G. Toscano-Pulido, and M. S. Lechuga, "An Extension of Particle Swarm Optimization that Can Handle Multiple Objectives," in *Multiple Objective Metaheuristics*, Paris - France, 2002.

[158] L. Deming, "A Pareto Archive Particle Swarm Optimization for Multi-objective Job Shop Scheduling," *Computers & Industrial Engineering*, vol. 54, pp. 960-971, 2008.

[159] G. Toscano-Pulido and C. A. C. Coello, "A Constraint-Handling Mechanism for Particle Swarm Optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1396-1403.

[160] G. T. Pulido and C. A. C. Coello, "Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer," in *In Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 3102, 2004, pp. 225-237.

[161] M. Eghbal, N. Yorino, and Y. Zoka, "Application of Multi-objective Evolutionary Optimization Algorithms to Reactive Power Planning Problem," *IEEE Transactions on Electrical and Electronic Engineering*, vol. 4, pp. 625-632, 2008.

[162] L. V. Santana-Quintero, N. Ramírez, and C. A. C. Coello, "A Multi-objective Particle Swarm Optimizer Hybridized with Scatter Search," in *Proceedings of the 5th International Conference in Artificial Intelligence*, vol. 4293, 2006, pp. 294-304.

[163] N. K. Khalid, I. Zuwairie, T. B. Kurniawan, K. Marzuki, and A. P. Engelbrecht, "Implementation of Binary Particle Swarm Optimization for DNA Sequence Design," *Computer Science - Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, vol. 5518, pp. 450-457, 2009.

[164] M. El-Abd and M. Kamel, "Cooperative Particle Swarm Optimizers: A Powerful and Promising Approach," *Computational Intelligence - Stigmergic Optimization*, vol. 31, pp. 239-259, 2006.

[165] G. H. Christopher and N. Vlahopoulos, "An Integrated Multidisciplinary Particle Swarm Optimization Approach to Conceptual Ship Design," in *Proceeding of the SAE World Congress on Structural and Multidisciplinary Optimization*, 2009.

[166] Q. K. Pan, L. Wang, and B. Qian, "A Novel Multi-objective Particle Swarm Optimization Algorithm for No-Wait Flow Shop Scheduling Problems," *Institution of Mechanical Engineers - Part B: Engineering Manufacture*, vol. 222, pp. 519-539, 2008.

[167] S. Chamaani, S. A. Mirtaheri, M. Teshnehlab, and M. A. Shooredeli, "Modified Multi-objective Particle Swarm Optimization for Electromagnetic Absorber Design," in *Proceeding of IEEE - Asia-Pacific Conference on Applied Electromagnetics*, 2007, pp. 1-5.

[168] C. Jiejin, M. Xiaoqian, L. Qiong, L. Lixiang, and P. Haipeng, "A Multi-objective Chaotic Particle Swarm Optimization for Environmental/Economic Dispatch," *Energy Conversion and Management*, vol. 50, pp. 1318-1325, 2009.

[169] M. E. H. Pedersen and A. J. Chipperfield, "Simplifying Particle Swarm Optimization," *Applied Soft*

Computing, vol. 10, pp. 618-628, 2010.

[170] Y. Li, R. Liao, and D. Zhang, "A Modified Particle Swarm Optimization and Simulation," in *Proceedings of the Asia-Pacific Conference on Information Processing*, vol. 2, 2009, pp. 387-390.

[171] Z. Tang and D. Zhang, "A Modified Particle Swarm Optimization with an Adaptive Acceleration Coefficients," in *Proceedings of the Asia-Pacific Conference on Information Processing*, vol. 2, 2009, pp. 330-332.

[172] Y. Gao and Y. Duan, "A New Particle Swarm Optimization Algorithm with Adaptive Mutation Operator," in *Proceedings of the 2nd International Conference on Information and Computing Science*, vol. 1, 2009, pp. 58-61.

[173] L. Mo and H. Zheng, "Improved PSO Algorithm with Adaptive Inertia Weight and Mutation," in *Proceedings of the WRI World Congress on Computer Science and Information Engineering*, vol. 4, 2009, pp. 622-625.

[174] T. M. Walski, et al., *Advanced Water Distribution Modeling and Management*. Waterbury, CT, USA: Bentley Institute Press, 2003.

[175] M. W. Larry, *Water Supply Systems Security*, 1st ed. USA: McCRAW HILL PROFESSIONAL ENGINEERING, 2004.

[176] E. Shudifat, "Use of Hydraulic Modeling and Simulation Software to Optimize the Operation of Water Distribution Networks," *ENGREF*, Jan. 2007.

[177] P. K. Swamee and A. K. Jain, "Explicit Equations for Pipe-Flow Problems," *Hydraulics Division (ASCE)*, vol. 102, no. 5, p. 657–664, 1976.

[178] S. Gilmore. (2008, Sep.) EcoSmartElectricians. [Online]. http://www.ecosmartelectricians.com.au/starter-kit/c01_66.html

[179] J. P. Rance, B. Ulanicki, B. Coulbeck, S. Kosov, and P. Bounds, *FINESSE: A Comprehensive Software Package for Water Network Modeling and Optimization*. Leicester, UK: Water Software Systems, De Montfort University, 2002.

[180] W. M. Grayman, R. M. Clark, and R. M. Males, "Modeling Distribution System Water Quality:Dynamic Approach," *Water Resources Planning and Management, ASCE*, p. 114, 1988.

[181] L. A. Rossman and P. F. Boulos, "Numerical Methods for Modeling Water Quality in Distribution Systems: A Comparison," *Water Resources Planning and Management, ASCE*, pp. 122-137, 1996.

[182] L. E. Ormsbee and S. Lingireddy, "Calibrating Hydraulic Network Models," *American Water Works Association*, pp. 1-44, 1997.

[183] L. Baumberger, V. Hart, and S. Darkwah, "Effect of GIS-Based Demand Allocation on Water Distribution System Modeling," *Florida Water Resources* , pp. 1-3, 2007.

[184] I. C. Goulter, "Systems Analysis in Water Dstribution Network Design: From Theory to Practice," *Water Resources Planning and Management, ASCE*, vol. 118, no. 3, p. 238–248, 1992.

[185] K. E. Lansey, "Optimal Design of Water Distribution Systems," in *Water Distribution System Handbook*. McGraw-Hill, 2000.

[186] A. V. Babayan, D. A. Savic, and A. Godfrey, "Multi-objective Optimization of Water Distribution System Design Under Uncertain Demand and Pipe Roughness," *Water Resources Planning and Management - ASCE*, vol. 131, pp. 467-476, 2004.

[187] D. A. Savic, G. A. Walters, and M. Schwab, "Multiobjective Genetic Algorithms for Pump Scheduling in Water Supply," in *In the Proceedings of the Artificial Intelligence and Simulation of Behaviour* , vol. 1305, Springer-Verlag, Berlin, 1997, p. 227–236.

[188] D. Halhal, G. A. Walters, D. Ouazar, and D. Savic, "Water Network Rehabilitation with a Structured Messy Genetic Algorithm," *Water Resources Planning and Management, ASCE*, vol. 123, no. 3, p. 137–146, 1997.

[189] T. A. Doby, S. V. Kumar, J. W. Baugh, E. D. Brill, and S. R. Ranjithan, "Genetic Algorithm Search for Least Cost Design of Looped Pipe Networks Using Age as a Quality Surrogate and Dfferent Levels of Redundancy," in *In Proceedings of the 2001 World Water and Environmental Resources Congress. ASCE,*, Orlando, FL, 2001.

[190] R. Farmani, G. A. Walters, and D. A. Savic, "Trade-off Between Total Cost and Reliability For Anytown Water Distribution Network," *Water Resources Planning and Management, ASCE*, pp. 161-171, 2005.

[191] Water Research Center, *Pump Scheduling in Water Supply*. London, UK: Wiltshire, 1985.

[192] K. E. Lansey and A. Kofi, "Optimal Pump Operation Considering Pump Switches," *Water Resources Planning and Management*, vol. 120, pp. 1-17, 1994.

[193] E. Fred. and W. Larry, *Water Distribution System Operation: Application of Simulated Annealing*. Phoenix: Arizona State University, 2004.

[194] L. Mauel, T. Devi, and P. Ben, "Multi Objectives Optimization of the Pump Scheduling Problem Using SPEA2," *Evolutionary Computation, IEEE Congress*, pp. 435-442, 2005.

[195] D. Mackle, A. Savic, and G. A. Walters, "Application of Genetic Algorithms to Pump Scheduling for Water Supply," in *In the proceddings of the GALESIA 95*, London, UK, 1995.

[196] A. Nafi, C. Werey, and P. Lierena, "Scheduling of Water Distribution Systems Using a Multi-Objective Approach," *Multiobjective Programming and Goal Programming*, vol. 618, pp. 231-241, 2009.

[197] G. Yu, R. S. Powel, and J. H. Sterling, "Optimized Pump Scheduling in Water Distribution Systems," *Optimization theory and Application*, vol. 83, pp. 463-488, 1994.

[198] I. C. Goulter and C. Xu, "Reliability-based Optimal Design of Water Distribution Networks," *Water Resources Planning and Management, ASCE*, vol. 125, no. 6, p. 352–362, 1999.

[199] R. Gupta and P. R. Bhave, "Reliability Analysis of Water-Distribution Systems," *Environmental Engineering*, pp. 120-130, 1994.

[200] C. V. Lucken, B. Baran, and A. Sotelo, "Pump Scheduling Optimization Using Asynchronous Parallel Evolutionary Algorithms," *CLEI ELECTRONIC*, vol. 7, no. 2, pp. 40-60, Dec. 2004.

[201] EPANET, *Water Quality in Small Community Distribution Systems: A Reference Guide for Operation*. Cincinnati, Ohio: U.S. Environmental Protection Agency, 2008.

[202] C. M. Fonseca and P. J. Fleming, "Multiobjective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction," in *In Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems*, Sheffield, UK, 1995, pp. 42-52.

[203] System Bentley, "WaterCAD," Bentley Sustaining Infrstructure - Hydraulics Model and Water Quality Manuel User, 2008.

[204] L. A. Rossman, "EPANET: Risk Reduction Engineering Laboratory," U.S. Environmental Protection Agency - Water Supply and Water Resources Division Users Manual, 2000.

[205] L. I. Manual, T. D. Prasad, and B. Paecht, "Multi-Objective Optimization of the Pump Scheduling Problem using SPEA2," *IEEE Congress on Evolutionary Computation*, vol. 1, pp. 435-442, 2005.

[206] E. Fred and W. Larry , "Application of Simulated Annealing," in *Water Distribution System Operation*, 2004, ch. 5.

[207] S. Elad and A. Ostfeld, "Optimal Operation of Multi-Quality Water Distribution Systems: Unsteady Conditions," *Engineering Optimization*, vol. 36, pp. 672-681, 2004.

[208] T. Liang and S. Nahaji, "Managing Water Quality by Mixing Water From Different Sources," *Water Resources Planning and Management Division - ASCE*, vol. 109, no. 3, p. 48–57, 1983.

[209] A. Ostfeld and U. Shamir, "Optimal Operation of Multiquality Networks I: Steady State Conditions," *Water Resources Planning and Management Division, ASCE*, vol. 119, no. 6, p. 645–662, 1993.

[210] L. A. Rossman, "EPANET," U.S. Environmental Protection Agency Manuel, 2003.

[211] A. Ostfeld, "A Review of Modeling Water Quality in Water Distribution Systems," *Urban Water* , vol. 2, pp. 233-245, 2005.

[212] P. F. Boulos, et al., "Optimal Pump Operation of Water Distribution Systems using Genetic Algorithms," *AWWA Distribution System Symposium*, 2001.

[213] T. C. Jones, "Evolutionary Algorithms, Fitness Landscapes and Search," *Optimization theory and Applications*, 1995.

[214] C. G. Langton, "Artificial Life," *Artificial Life 1*, pp. 1-47, 1989.

[215] R. Storn, "On the Usage of Differential Evolution for Function Optimization," in *In Proceedings of Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, 1996, p. 519–523.

[216] P. Bienert, "Aufbau Einer Optimierungsautomatik f¨ur Drei Parameter," *Engineering Optimization*, 1967.

[217] T. Back, D. Fogel, and Z. Michalewicz, *The Handbook of Evolutionary Computation*. Oxford University Press., 1997.

[218] J. Heitkotter and D. Beasley. (2000) The HitchHiker's Guide to Evolutionary Computation: A List of Frequently Asked Questions (FAQ). [Online]. http://www.aip.de/~ast/EvolCompFAQ/

[219] L. J. Fogel, *On the Organization of Intellect*. Los Angeles, , 1964., California, USA: University of California, 1964.

[220] L. Davis, "Adapting Operator Probabilities in Genetic Algorithms," in *In Proceedings of the Third International Conference on Genetic Algorithms*, Arlington, VA, 1989, p. 61–69.

[221] G. Beni and J. Wang, "Swarm Intelligence in Cellular Robotic Systems," in *In Proceedings of the NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy, 1989.

[222] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems by*. New York, NY, USA: Oxford University Press, 1999.

[223] M. Dorigo, "Optimization, Learning and Natural Algorithms," *Intelligence Systems*, 1992.

[224] Y. Shi and R. C. Eberhart, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," in *In Proceedings of the Congress on Evolutionary Computation*, vol. 1, 2000, pp. 84-88.

[225] Y. Shi and R. C. Eberhart, "Parameter Selection in Particle Swarm Optimization," in *In Proceedings of Evolutionary Programming VII*, 1998, pp. 591-600.

[226] A. Carlisle and G. Dozier, "An Off-The-Shelf PSO," in *In Proceedings of the Particle Swarm Optimization Workshop*, 2001, pp. 1-6.

[227] V. D. F. Bergh, "An Analysis of Particle Swarm Optimizers," PhD thesis, University of Pretoria, Faculty of Natural and Agricultural Science, 2001.

[228] M. Clerc and J. Kennedy, "The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 58-73, 2002.

[229] I. C. Trelea, "The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection," *Information Processing*, vol. 85, pp. 317-325, 2003.

[230] D. Bratton and T. Blackwell, "A Simplified Recombinant PSO," *Artificial Evolution and Applications*, 2008.

[231] Z. Zhi-Hui and Z. Jun, "Adaptive Particle Swarm Optimization," *Computer Science*, vol. 5217, no. Ant Colony Optimization and Swarm Intelligence, p. 227–234, 2008.

[232] X. Wang, Y. Yang, Q. Zeng, and J. Wang, "Particle Swarm Optimization with Adaptive Inertia Weight and Its Application in Optimization Design," *Advanced Materials Research*, vol. 97, pp. 3484-3488, 2010.

[233] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182-197, 2001.

[234] J. Tang and X. Zhao, "Particle Swarm Optimization with Adaptive Mutation," in *In Proceedings of IEEE 2009 WASE International Conference on Information Engineering*, 2009, pp. 234-237.

[235] N. Higashi and H. Iba, "Particle Swarm Optimization with Gaussian Mutation," in *In Proceedings of IEEE Swarm Intelligence Symposium*, 2003, p. 72–79.

[236] A. Stacey, M. Jancic, and I. Grundy, "Particle Swarm Optimization with Mutation," in *In Proceedings of the Congress Evolutionary Computation*, 2003, p. 1425–1430.

[237] J. Chen, Z. Ren, and X. Fan, "Particle Swarm Optimization with Adaptive Mutation and its Application Research in Tuning of PID Parameters," in *In Proceedings 1st International Symposium on Systems and Control in Aerospace and Astronautics*, 2006, p. 990–994.

[238] C. Li, S. Yang, and I. Korejo, "An Adaptive Mutation Operator for Particle Swarm Optimization," in *In Proceedings of the 2008 UK Workshop on Computational Intelligence*, 2008, pp. 165-170.

[239] X. Li, J. Zhuang, S. Wang, and Y. Zhang, "A Particle Swarm Optimization Algorithm Based on Adaptive Periodic Mutation," in *In Proceedings of IEEE 4th International Conference on Natural Computation*, 2008, pp. 150-155.

[240] C. Li, Y. Liu, L. Kang, and A. Zhou, "A Fast Particle Swarm Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy," in *In Proceedings of the 2nd International Symposium on Intelligence Computation and Applications*, 2007, pp. 334-343.

[241] H. Wang, Y. Liu, C. Li, and S. Zeng, "A Hybrid Particle Swarm Algorithm with Cauchy Mutation," in *In Proceedings of IEEE Swarm Intelligence Symposium*, 2007.

[242] J. B. Park, K. S. Lee, J. R. Shin, and K. Y. Lee, "A Particle Swarm Optimization for Economic Dispatch with Non-smooth Cost Functions," *IEEE Transactions on Power Systems*, vol. 20, pp. 34-45, 2005.

[243] S. Z. Zhao and P. N. Suganthan, "Two-lbests Based Multi-objective Particle Swarm Optimizer," *Engineering Optimization*, pp. 1-17, 2010.

[244] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *In Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, California, 1967, p. 281–297.

[245] Weston.pace. (2007) Wikipedia the Free Encyclopedia. [Online]. http://en.wikipedia.org/wiki/K-means algorithm

[246] A. David and V. Sergei , "K-Means++: The Advantages of Carful Seeding," in *ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[247] M. Tomassini, "Parallel and Distributed Evolutionary Algorithms: A Review," *Evolutionary Algorithms in Engineering and Computer Science*, pp. 113-138, 1999.

[248] E. Alba and J. Troya, "Analyzing Synchronous and Asynchronous Parallel Distributed Genetic Algorithms," *Future Generation Computer Systems*, 2001.

[249] L. D. Whitley and T. Starkweather, "GENITOR II: A Distributed Genetic Algorithm," *Experimental and Theoretical Artificial Intelligence*, 1990.

[250] R. Tanese, "Distributed Genetic Algorithms," in *In Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989, pp. 434-439.

[251] M. Gorges-Schleuter, "Explicit Parallelism of Genetic Algorithms through Population Structures," *Parallel Problem Solving from Nature*, pp. 150-159, 1991.

[252] T. C. Belding, "The Distributed Genetic Algorithm Revisited," in *In proceedings of the 6th International Conference on Genetic Algorithms*, 1995, pp. 114-121.

[253] H. Hsu-Chih, T. Ching-Chih, and L. Shui-Chun, "SoPC-Based Parallel Elite Genetic Algorithm for Global Path Planning of an Autonomous Omnidirectional Mobile Robot," in *In Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, TX, USA, 2009, pp. 1956-1962.

[254] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Z¨urich, Switzerland, Technical Report 103, 2001.

[255] D. G. Terrell and D. W. Scott , "Variable Kernel Density Estimation," *Annals of Statistics*, vol. 3, no. 20, p. 1236–1265, 1992.

[256] B. L. Miller and D. E. Goldberg, "Genetic Algorithms, Tournament Selection, and the Effects of Noise," University of Illinois at Urbana Champaign - Department of General Engineering, Urbana, IL, IlliGAL Report, 1995.

[257] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-objective Optimization," *Theoretical Advances and Applications*, pp. 105-145, 2005.

[258] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On Test Functions for Evolutionary Multi-objective Optimization," *Parallel Problem Solving from Nature*, vol. 3242, pp. 792-802, 2004.

[259] F. Kursawe, "A Variant of Evolution Strategies for Vector Optimization," in *In Proceedings of the 1st Workshop Parallel Problem Solving from Nature*, vol. 496, 1991, pp. 193-197.

[260] C. A. C. Coello, G. A. Lamont, and D. A. Van-Veldhuizen, *Evolutionary Algorithms for Solving Multi-objective Problems*, 2nd ed. Springer, 2007.

[261] N. C. Chauhan, M. V. Kartikeyan, and A. Mittal, "Design of RF Window Using Multi-objective Particle Swarm Optimization," in *Proceedings of the International Conference of Recent Advances in Microwave Theory and Applications*, 2008, pp. 34-37.

[262] P. Yang, "Hybrid Particle Swarm Optimization for Vehicle Routing Problem with Reverse Logistics," in *Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2009, pp. 462-465.

[263] T. Ray and H. M. Tsai, "Swarm Algorithm for Single and Multi-objective Airfoil Design Optimization," *AIAA*, vol. 42, pp. 366-373, 2004.

[264] B. Coulbeck and M. Sterling, "Optimal Control of Water Distribution Systems," in *In Proceedings of the Institute of Electrical Engineers*, 1978, pp. 1025-1039.

[265] D. V. Chase and L. E. Ormsbee, "Optimal Pump Operation of Water Distribution Systems with Multiple Tanks," in *In Proceedings of the American Water Works Association*, 1989, p. 205.

[266] M. L. Zessler and U. Shamir, "Optimal Operation of Water Distribution Systems," *Water Resources Planning and Management, ASCE*, pp. 118-125, 1989.

[267] K. E. Lansey and Q. Zhong, "A Methodology for Optimal Control of Pump Stations," in *In Proceedings of the Water Resources Planning and Management Specialty Conference, American Society of Civil Engineers*, 1990, p. 58.

[268] D. Reardon, "Audit Manual for Water-Wastewater Facilities," *CR-104300, EPRI*, 1994.

[269] H. Arora and M. W. LeChevallier, "Energy Management Opportunities," *the American Water Works Association*, p. 40, 1998.

[270] G. Dandy and C. Hewitson, "Optimising Hydraulics and Water Quality in Water Distribution Networks using Genetic Algorithms," in *In Proceedings of the ASCE Joint Conference on Water Resources*, Minneapolis, Minnesota, 2000.

[271] F. E. Goldman, B. Sakarya, L. E. Ormsbee, J. Uber, and L. W. Mays, *Optimization Models for Operations - Water Distribution Systems Handbook*. New York: McGraw-Hill, 2000.

[272] G. Hovstadius, "Pump System Effectiveness," *Pumps and Systems*, p. 48, 2001.

[273] O. Schutze, M. Laumanns, C. A. C. Coello, M. Dellnitz, and E. G. Talbi, "Convergence of stochastic search algorithms to finite size Pareto Set Approximations," *Global Optimization*, vol. 41, no. 4, p. 559–577, 2008.

[274] O. Schutze, "Set Oriented Methods for Global Optimization," PhD thesis, University of Paderborn, 2004.

[275] T. Hiroyasu, M. Miki, and S. Watanabe, "The New Model of Parallel Genetic Algorithm in Multi-Objective Optimization Problems–Divided Range Multi-Objective Genetic Algorithm," in *In Proceddings of the 2000 Congress on Evolutionary Computation*, Piscataway, New Jersey, 2000, p. 333–340.

[276] A. L. Jaimes and C. A. C. Coello, "Applications of Parallel Platforms and Models in Evolutionary Multi-Objective Optimization," *Computational Intelligenc*, vol. 210, pp. 23-49, 2009.

[277] J. D. Knowles, "Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization," PhD Thesis, The University of Reading - Department of Computer Science, Reading, UK, 2002.

[278] O. Schutze, M. Laumanns, E. Tantar, C. A. C. Coello, and T. El-Ghazali, "Computing Gap Free Pareto Front Approximations with Stochastic Search Algorithms," *Evolutionary Computation*, vol. 18, no. 1, pp. 65-96, 2010.

[279] J. D. Knowles and D. W. Corne, "Approximating the Non-dominated Front using the Pareto Archived Evolution Strategy," *IEEE Transaction and Evolutionary Computation*, vol. 8, pp. 149-172, 2000.

[280] J. D. Knowles and D. W. Corne, "Propoerties of An Adaptive Grid Algorithmfor Sorting Non-dominated Vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100-116, 2003.

[281] S. F. Shu-Kai and C. Ju-Ming, "A Parallel Particle Swarm Optimization Algorithm for Multi-objective Optimization Problems," *Engineering Optimization*, vol. 41, pp. 673-697, 2009.

[282] G. E. Liepins and M. D. Vose, "Representational Issue in Genetic Algorithms ," *Experimental and*

Theoretical Artificial Intelligence, 1990.

[283] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithm*, 1st ed. New York, US: Springer, 2008.

[284] R. A. Grauana and J. D. Schaffer, "Representation and hidden Bias: Cray vs. Binary Coding for Genetic Algorithms," in *In Proceedings of the 5th International Conference on Machine Learning*, 1988, pp. 153-161.

[285] C. C. Palmer and A. Kershenbaum, "Representing Trees in Genetic Algorithms," in *In the Proceedings of IEEE Conference on Evolutionary Computation*, Bristol, USA , 1994, p. 379–384.

[286] D. E. Goldberg , "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking," *Complex Systems*, vol. 5, pp. 139-167, 1990.

[287] Europump and Hydraulic Institute, *Variable Speed Pumping Guide*, E. Ltd, Ed. Oxford, UK: Pump Systems Matter, 2004.

[288] H. P. Barringer, "Reliability Issues From A Management Perspective," in *In Proceedings of the 2nd API Pipeline Conference*, San Antonio, TX, 2001, pp. 1-12.

[289] L. Nelik , *Centrifugal & Rotary Pumps Fundamentals With Applications*. Boca Raton, FL, US: CRC Press, 1999.

[290] L. Manuel, "Operational Optimisation of Water Distribution Networks," PhD Thesis, Edinburgh Napier University, Edinburgh, UK, 2009.

[291] Z. Kapelan, D. A. Savic, and G. A. Walters, "Multi-objective Sampling Design for Water Distribution Model Calibration," *Water Resources Planning and Management, ASCE*, vol. 6, no. 129, p. 466–479, 2003

[292] R. Farmani, G. Walters, and D. Savic, "Evolutionary Multi-objective Optimization of the Design and Operation of Water Distribution Network: Total Cost vs. Reliability vs. Water Quality," *Hydroinformatics*, vol. 80, no. 6, pp. 165-179, 2008.

[293] C. J. Rouhiainen, M. O. Tade, and G. West, "Multi-objective Genetic Algorithm for Optimal Scheduling of Chlorine Dosing in Water Distribution Systems," *Advances in Water Supply Management* , p. 459–469, 2003.

[294] T. A. Doby, S. V. Kumar, J. W. Baugh, E. D. Brill, and S. R. Ranjithan, "Genetic Algorithm Search for Last Cost Design of Looped Pipe Networks Using Age as a Quality Surrogate and Different Levels of Redundancy," in *In Proceedings of the 2001 World Water and Environmental Resources Congress, ASCE*, Orlando, FL, 2001.

[295] S. Habibi, "Minimization of Pumping Cost in Water Supply Operations," University of Saskatchewan, Saskatoon, Canada, Technical Report, 2000.

[296] Environment Canada. (2010, Sep.) www.ec.gc.ca. [Online]. http://www.ec.gc.ca/eau-water/default.asp?lang=En&n=F25C70EC-1

[297] J. Sanclement, P. Webster, J. Thomas, and H. Ramadan, "Bacterial Biofilms in Surgical Specimens of Patients with Chronic Rhinosinusitis," *Laryngoscope*, vol. 115, no. 4, p. 578–582, 2005.

[298] S. W. Mahfoud, "Niching Methods for Genetic Algorithms," PhD Thesis, Illinois Genetic Algorithms Laboratory (IlliGAL), Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1995.

[299] R. Dawkins, *The Selfish Gene*. Oxford, U.K.: Oxford University Press, 1976.