Spring 5-31-2017

Multi-population-based differential evolution algorithm for optimization problems

Ishani Chatterjee
New Jersey Institute of Technology

ABSTRACT

MULTI-POPULATION-BASED DIFFERENTIAL EVOLUTION ALGORITHM FOR OPTIMIZATION PROBLEMS

by
Ishani Chatterjee

A differential evolution (DE) algorithm is an evolutionary algorithm for optimization problems over a continuous domain. To solve high dimensional global optimization problems, this work investigates the performance of differential evolution algorithms under a multi-population strategy. The original DE algorithm generates an initial set of suitable solutions. The multi-population strategy divides the set into several subsets. These subsets evolve independently and connect with each other according to the DE algorithm. This helps in preserving the diversity of the initial set. Furthermore, a comparison of combination of different mutation techniques on several optimization algorithms is studied to verify their performance. Finally, the computational results on the arbitrarily generated experiments, reveal some interesting relationship between the number of subpopulations and performance of the DE.

Centralized charging of electric vehicles (EVs) based on battery swapping is a promising strategy for their large-scale utilization in power systems. In this problem, the above algorithm is designed to minimize total charging cost, as well as to reduce power loss and voltage deviation of power networks. The resulting algorithm and several others are executed on an IEEE 30-bus test system, and the results suggest that the proposed algorithm is one of effective and promising methods for optimal EV centralized charging.

MULTI-POPULATION-BASED DIFFERENTIAL EVOLUTION ALGORITHM FOR OPTIMIZATION PROBLEMS

by
Ishani Chatterjee

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Department of Electrical and Computer Engineering

May 2017

Blank Page

APPROVAL PAGE

MULTI-POPULATION-BASED DIFFERENTIAL EVOLUTION ALGORITHM FOR OPTIMIZATION PROBLEMS

Ishani Chatterjee

Dr. MengChu Zhou, Thesis Advisor Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. James M. Calvin, Committee Member Date
Professor of Computer Science, NJIT

Qing Liu, Committee Member Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Ishani Chatterjee

Degree: Master of Science

Date: May 2017

Undergraduate and Graduate Education:

- Master of Science in Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2017

- Bachelor of Science in Electronics and Communication Engineering,
  West Bengal University of Technology, Kolkata, West Bengal, India, 2015

Major: Computer Engineering

Presentations and Publications:

Ishani Chatterjee and Dr. MengChu Zhou "Differential evolution algorithms under multi-population strategy", 26th Wireless and Optical Communications Conference (WOCC 2017), Newark, USA, April 2017

Ishani Chatterjee and Arunava Mukhopadhyay "Design of a microstrip line fed rectangular slot antenna and response under slit loading," 4th International conference of Technical and Managerial innovation in computing and communication in Industry and academia, Kolkata, India, August 2013

This thesis is dedicated to
My Grandparents for inspiring me,
My Parents for teaching me how to dream,
Professor MengChu Zhou for believing in me,
Dr. Sharon Morgan for motivating me,
Dr. Ashish Borgaonkar for guiding me,
My Family for loving me,
My Friends for supporting me.

ACKNOWLEDGMENT

This thesis is done as an engineering Project, as a part of our course. I am really thankful to my course adviser Dr. MengChu Zhou, Distinguished Professor, Department of Electronics and Communication Engineering, New Jersey Institute of Technology, Newark, NJ, for his valuable guidance and assistance, without which the accomplishment of the task would have never been possible. Dr. Zhou has directed me with great enthusiasm and interest and has allowed me to have the freedom to exercise thoughtful and scientific approach towards the problem.

I would also like to thank Prof. Qing Liu, Assistant Professor, New Jersey Institute of Technology, Newark, NJ, for agreeing to be a part of the committee for my thesis Defence.

Finally, all of this would be incomplete without the presence of Dr. James M. Calvin, Professor, New Jersey Institute of Technology, Newark, NJ, who is the third member on the panel. I was a part of the Data Structures & Algorithm taught by Dr. Calvin. Everything that was taught by him has been of immense help towards my thesis.

I also thank the teachers who have been my guide for knowledge for the past two years, without whom I would not have been able to share the knowledge in this project that I was able to.

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

LIST OF TABLES
(continued)

LIST OF FIGURES

LIST OF FIGURES
(continued)

LIST OF SYMBOLS

N_P	Number of Population
F	Scaling Factor
CR	Crossover Rate
T	Termination Criterion
∈	Belongs To
T	Number of iteration
T	Total number of time slots
N	Total number of electric vehicles (EVs)
S	The number of EV groups/sets (EVSs)
M	The number of EVs included in an EVS
N	The number of charging stations
P	Charger power rating of EVs
P_t	Total power consumption for EV charging at the t^{th} time slot
P_{tot}	Total power demand of all EVs.
P_t^{lim}	Maximum permissible power consumption for EV charging at the t^{th} time slot.
P_t^{rem}	Remaining power available for EV charging at the t^{th} time slot.

LIST OF SYMBOLS
(continued)

P_{max}	Maximum residential load demand with EVs being charged.
P_t^{load}	Total residential power consumption at the t^{th} time slot without EV being charged.
Pow_t	Total power consumption of EVs at the t^{th} time slot.

LIST OF DEFINITIONS

Initial vector	A parent vector from the present/current generation
Mutant vector	A vector obtained through the differential mutation operation.
Trial vector	An offspring formed by recombining the donor with the target vector.
Fitness value	The value which represents the quantitative representation of natural selection.
Hill-climbing	A mathematical optimization technique which belongs to the family of local search.
Local search	A heuristic method for solving computationally hard optimization problems.

CHAPTER 1

INTRODUCTION

1.1 Background

Optimization is a procedure through which the best possible values of decision variables are obtained given a set of constraints and in terms of a selected optimization objective function. The most common optimization procedure applies to a design that minimizes the total cost or maximize the possible reliability or any other specific objective. Fields of science and engineering, business decision-making and industry are all rich in problems that require the implementation of optimization approaches.

Since, most real world optimization problems seem to be both fundamentally and practically hard, research into better algorithms remains valuable and continues, such that, one can find the best solution by using an efficient and proper optimization algorithm. The path we choose to travel to work every day; the line we select to stand in for billing in the supermarket; the order we choose for our daily tasks; or even to organize a function are all examples of optimization problems present in our daily lives.

Nowadays, there exist numerous optimization algorithms that work by using gradient-based and heuristic-based search techniques in deterministic and stochastic contexts. In order to widen the applicability of an optimization approach to various problem domains, natural and physical principles are mimicked to develop robust optimization algorithms. Simulated annealing, ant colony optimization, memetic algorithms, and particle swarm optimization are few examples of such algorithms.

Over the last decade, evolutionary algorithms were extensively used in various problem domains and succeeded in effectively finding the optimal or near-optimal solutions. The present thesis provides a detailed description of one such evolutionary algorithm, named as Differential Evolution (DE).

DE has earned a reputation of a very effective global optimizer (Storn and Price, 1995). It is a stochastic search method for solving optimization problems of multi-dimensional real valued functions by repeatedly trying to improve the fit solutions. In general, the job is to optimize some features of a system by suitably selecting the system parameters. In recent years, DE has been widely used because of its stability, robustness, and ability for global search and determination of the optimal solution. Its effectiveness and efficiency have been established successfully in such fields as artificial intelligence, communication, and mechanical engineering.

Electric Vehicles (EVs) are welcoming a rapid development along with progresses of relevant technologies in recent years. As an eco-friendly substitute for traditional vehicle, EV is seen as a promising solution to the ever devastating energy crisis and environmental pollution around the globe, thus drawing increasing attentions from the public, markets, decision-makers, industry and academia. Many countries and cities have proposed plans to promote EV usage or have been preparing to do so, providing a foreseeable vision that EVs will become the major vehicles of the private transportation sector in the near future. Limited battery capacity and long charging time, probably the most widely complained disadvantages, raise mileage anxiety and largely impair EV users' driving experience. As a result, charging convenience has become a top concern affecting potential users' choice between EV and traditional fuel-engine vehicle. Specialized EV charging stations, which provide more than 10 times faster charging speed than domestic charging, are therefore critical to the successful promotion of EVs.

However, uncoordinated EVs charging would exert a tremendous influence on the daily residential load curve if they are widely connected to the power grid for battery charging. Due to the uncertainty of their charging behaviours, uncoordinated random charging of a mass of EVs may lead to unforeseen effects on the normal operation of a power distribution system, i.e., voltage fluctuation, thereby aggravating the load peak and off-peak difference in the network. Without taking the spot pricing into consideration, EV owners may pay much higher cost for battery charging.

1.2 Goal and Objectives

This thesis analyses a DE algorithm with multi-population strategy, based on randomly generated subpopulation. It studies the effect of this strategy on the searching accuracy, optimization ability and convergence speed. It compares a combination of different mutation techniques on several optimization algorithms to verify their performance. An improved population-based heuristic algorithm, multi-population based differential evolution, is designed to find the optimal charging priority and location of EVs in a distribution network.

In this thesis, a novel charging strategy of EVs based on optimal charging priority and charging station is proposed under a spot pricing-based electricity market environment is proposed by taking advantage of a centralized charging strategy. The proposed approach is evaluated via an IEEE 30-bus test system.

1.3 Organization of Thesis

Chapter 2 explains the differential evolution algorithm, and introduces the electric vehicle problem. It also reflects the background and analyses the research work done on the algorithm. Chapter 3 illustrates the algorithm based on multi-population strategy. Results on the effect of multi-population strategy on differential evolution and number of subpopulation on execution time and mean value are discussed in Chapter 4. In Chapter 5 an EV charging rule based on charging priority and locations is set forth and the problem of optimal EV charging priority and locations (bus index) is given. Its application to the IEEE 30-bus system is discussed, and its results are compared with some existing methods. In Chapter 6 conclusion and future works are given.

CHAPTER 2

LITERATURE REVIEW

A differential evolution (DE) algorithm is a genetic population based algorithm, pioneered by Storn and Price in 1995 (Storn and Price, 1995; 1997). The benefit of the algorithm is its simple structure, robustness, speed, ease of use and few control variables. The algorithm is a chief genetictype algorithm for solving real valued problems. If a system is susceptive to being reasonably evaluated, DE provides the best possible performance from it. The algorithm uses mutation as a search technique and selection to direct the search towards a potential section in the feasible region. In general, the job is to optimize certain features of a system by suitably selecting the system parameters.

In recent years, DE has been widely used due to its stability, robustness, ability for global search and several excellent performances. Its effectiveness and efficiency have been established successfully in such fields as artificial intelligence, communications systems, mechanical designs to name a few.

2.1 An Introduction to Differential Evolution

DE maintains a population of fit solutions by combining existing ones according to a simple formula and keeps the solution with best fitness given an optimization problem. The starting iteration of the algorithm consists of four sections – initialization, mutation, crossover, and selection of which only the last three sections repeat themselves into the following iterations. The iterations continue until the termination criterion is satisfied, such as the maximum number of iterations, maximum number of function evaluations, and/ or maximum execution time. The flowchart of the algorithm is shown in Figure 2.1.

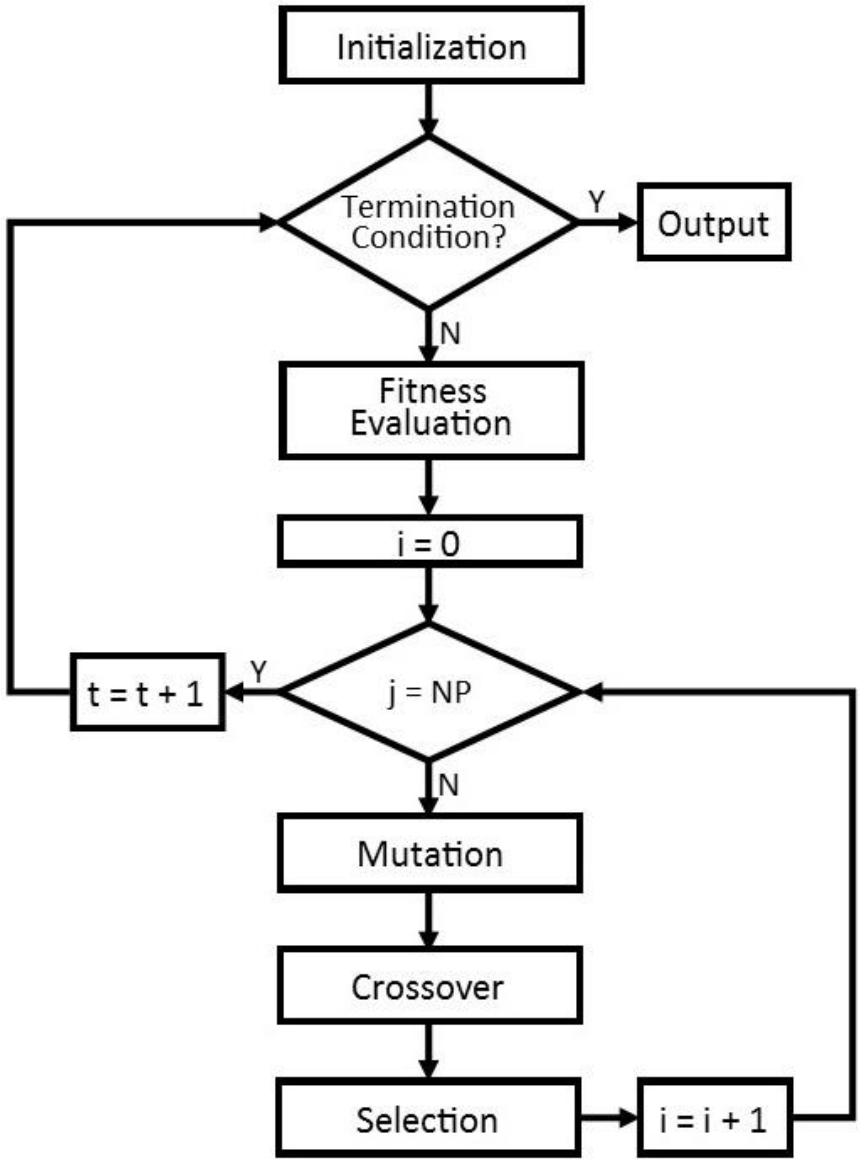

Figure 2.1 Flowchart of a differential evolution algorithm.

2.1.1 Initialization

Let $S \subseteq R^d$ be a d-dimensional search space of the problem taken under consideration. A population of size N_P d-dimensional random individual vectors, is generated where, $x_i^{(t)} = (x_{i,1}^{(t)}, x_{i,2}^{(t)}, \dots, x_{i,d}^{(t)}) \in S$ and $i \in \{1,2,3 \dots N_P\}$. The major parameters including population size (N_P), scaling factor (F), crossover rate (C_R) and termination criterion (T) are initialized. After the population is generated each, individual is encoded as a floating-point vector number. The initial set covers the entire search space by randomly spreading individual vectors with uniform distribution between the upper bound ($x_{\max,j}$) and lower bound ($x_{\min,j}$). The initial vector is generated as follows:

$$x_{i,j}^{(0)} = x_{min,j} + R_{i,j}[0,1] \times \left(x_{max,j} - x_{min,j}\right) (j = 0, 1, 2,3, \dots, d) \tag{2.1}$$

2.1.2 Mutation

Post initialization, a mutant vector ($v_i^{(t)}$) is achieved through the process of mutation, with respect to each population member in the current iteration. The common mutation strategies are as follows:

$$\text{DE/rand/1: } v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot (x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)}) \tag{2.2}$$

$$\text{DE/best/1: } v_i^{(t)} = x_{best}^{(t)} + F \cdot (x_{r_1^i}^{(t)} - x_{r_2^i}^{(t)}) \tag{2.3}$$

$$\text{DE/current to best/1: } v_i^{(t)} = x_i^{(t)} + F \cdot \left(x_{best}^{(t)} - x_i^{(t)}\right) + F \cdot (x_{r_1^i}^{(t)} - x_{r_2^i}^{(t)}) \tag{2.4}$$

$$\text{DE/rand to best/1: } v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot \left(x_{best}^{(t)} - x_{r_1^i}^{(t)}\right) + F \cdot (x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)}) \tag{2.5}$$

$$\text{DE/best/2: } v_i^{(t)} = x_{best}^{(t)} + F \cdot \left(x_{r_1^i}^{(t)} - x_{r_2^i}^{(t)}\right) + F \cdot (x_{r_3^i}^{(t)} - x_{r_4^i}^{(t)}) \tag{2.6}$$

$$\text{DE/rand/2: } v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot \left(x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)}\right) + F \cdot (x_{r_4^i}^{(t)} - x_{r_5^i}^{(t)}) \tag{2.7}$$

where, $r_1^i - r_5^i$ are random integers produced from the set {1, 2, 3, …, N_p}, which is not identical to the current mutant vector index *i*; weighting, or scaling, factor *F* is a user specified constant in the range between 0 and 2 (Gämperle, Sibylle and Petros, 2002); and $x_{best}^{(t)}$ is the individual vector with best fitness value in the current population.

2.1.3 Crossover

To magnify the diversity of the population, crossover is performed. The initial vector is mixed with the mutant vector to create a trial vector $u_i^{(t)}$

$$u_{i,j}^{(t)} = \begin{cases} v_{i,j}^{(t)} \; if \; \left(rand_j \leq C_R\right) or \; (j = \; j_{rand}), \\ x_{i,j}^{(t)} \; otherwise, \end{cases} \tag{2.8}$$

where, j = 1, 2,…, *d*; $rand_j$ is a random number between 0 and 1; C_R is user defined in the range [0, 1) and $j_{rand} \in (1, 2, \ldots, d)$ is the randomly picked index to ensure the trial and initial vector differ from each other by at least one parameter.

Thus, the trial vector is the outcome of two parent vectors, an initial vector and the mutant vector against which it competes in this step. The C_R represents that the trial vector inherits the parameter values from the mutant vector. For instance, when $C_R = 1$, every trial vector parameter is certain to come from a mutant vector. On the contrary, if $C_R = 0$, all but one trial vector parameter comes from the initial vector. To ensure a difference between the trial and the initial vectors by at least one parameter, the final trial vector parameter always comes from the mutant vector even when $C_R = 0$.

2.1.4 Selection

The selection process regulates if the initial or trial vector consistently adheres to the next iteration, i.e., at $t + 1$. The "greedy" selection strategy is applied if and only if a trial vector yields a better fitness value compared to the initial vectors. The trial vector is the initial vector for the next iteration; otherwise the initial vector remains the same for the next iteration. The selection operation is as follows:

$$\mathrm{x}_i^{(t+1)} = \begin{cases} \mathrm{u}^{(t)} \text{ if } f\left(\mathrm{u}^{(t)}\right) \leq f\left(\mathrm{x}_i^{(t)}\right), \\ \mathrm{x}_i^{(t)} \text{ otherwise,} \end{cases} \tag{2.9}$$

2.2 Advances in Differential Evolution

The classical DE includes a set of basic mutation strategies along with three possible crossover schemes namely binomial, exponential and arithmetic ones. Considering a vast range of studies on DE, Neri and Tirronen (2009) reviewed various DE-variants for single objective optimization problems and produced an experimental comparison of these variants on a set of standard benchmark functions. The first comprehensive survey on almost all aspects of the DE algorithm was published in 2011 by Das and Suganthan (2011). Dragoi and Dafinescu (2015) surveyed two aspects of DE algorithms, i.e., the self-adaptive and adaptive parameter control approach in DE and the hybridization of DE with diverse algorithms. A recent article in 2016 by Das, Mullick and Suganthan (2016) explained a more comprehensive account of the recent advances in DE including its basic concepts, various structures, and variants for solving constraints, multi-objective, dynamic, and large-scale optimization problems along with applications of DE variants to a variety of practical optimization problems.

DE is a simple, heuristic optimization algorithm with control parameters and learning strategies, which depends on the problem under consideration. To improve the original algorithm, many researchers have made useful efforts for improving its search accuracy, convergence speed, etc. In 2005, Quin and Suganthan (2005) proposed a self-adaptive DE algorithm, which can automatically modify its learning strategies and related parameters during an evolution process. There has been a growing trend of selecting the generation strategies from a pool (Spears, 1995) to DE with Ensemble of Parameters and mutation Strategies (EPSDE) (Mallipeddi *et al.*, 2011). Gong *et al*. (2011a) proposed an adaptive DE algorithm where four mutation strategies are used to create a pool. Yu *et al*. (2014a) introduced an individual dependent control parameter adaptation mechanism by using a two-step process. Wu *et al*. (2015a) presented a multi-population based framework to realize an adaptive ensemble of three mutation strategies into a novel DE variant named MPEDE in which MPE represents multi-population ensemble.

Song and Hou (2015) presented an improved DE with a multi-population strategy for solving high dimensional optimization problems. Tang *et al*. (2014) studied practical dynamic scheduling in a steelmaking continuous casting (SCC) problem by proposing and using an improved DE with a real-coded matrix representation for each individual of the population, a two-step method for generating the initial population, and a new mutation strategy. To further improve the efficiency and effectiveness of the solution process for dynamic use, an incremental mechanism is used to generate a new initial population for the

DE whenever a real-time event arises, based on the final population in the last DE solution process.

Baatar *et al.* (2013) proposed a DE algorithm adopting a λ-best mutation strategy for optimization of electromagnetic devices. Gämperli *et al*. (2002) assessed the selection of strategy parameters for DE over a set of test problems. Huang (2016) suggested a chaotic optimization algorithm with a multi-population strategy and adaptive crossover probability strategy for function optimization problems.

2.2.1 Prominent DE Variants for Bound-constrained Single-objective Global Optimization

DE has been most frequently applied to the global optimization problems involving a single objective function and bound constraints on the decision variables. Melo and Delbem (2012) recommended a Smart Sampling (SS) method to identify promising regions of the search space where an optimum may lie. Firstly, a high number of random solutions are generated covering the search space. Based on the fitness value this initial population is filtered and only better solutions are kept. A collection of new solutions is generated by moving one of the population members towards one of the better solutions, with a random noise. A classifier is trained to distinguish the promising solutions from the non-promising ones which is used to identify the good solutions from the newly introduced collection. The good individuals are added to the population, and the increased population is reduced to maintain a fixed cardinality by deleting the worse members. This process is repeated until a convergence criterion is met. A rule-based classifier is used on the final population to identify the promising region.

Li and Zhang (2011) proposed a DE based on subpopulation by a clustering technique. The clustering technique (agglomerative hierarchical) is simple to implement, can create dynamic subpopulations, and is adaptive to the change of the population although, the raw clusters may not be suitable for evolution (for example they may be too small). To overcome this, they suggested that all the clusters with a single element were combined (called as SPEX). If the current best is not a member of this cluster, then it can be used to maintain the diversity of the population and the explorative capability of the algorithm. For each of the clusters, which has more than one member but does not have enough members to perform mutation, the algorithm maintains a pool of solutions. The pool is updated after every generation and helps a cluster by supplying required members to perform a mutation.

Poikolainen *et al.* (2015) came up with a cluster-based population initialization technique for DE. The initialization is performed as a three-stage process. In the first stage, two local searches are performed on a random collection of uniformly selected points over the search space. The second searcher is the Rosenbrock algorithm (Rosenbrock, 1960), which has been shown to converge towards a local optimum. In the second phase, k-means clustering is applied to the resultant population of the first phase. The third stage is used to generate additional individuals to form the initial population of DE. In the third stage the best fit individuals are collected (forming a set Q) from each of the cluster. A probability is assigned to each of these better individuals based on their fitness values.

Understanding and utilizing surrounding and directional information is vital for a DE population to search efficiently. Cai and Wang (2013) suggested an improved mutation strategy in DE by introducing the guidance of surrounding and directional information.

They first defined a Neighbourhood Guided Selection (NGS) scheme for selecting the base and difference vectors for mutation. For generating the donor of the i^{th} target, NGS first assigns a probability to each vector of the population. The probability for any vector with index *j* is calculated as:

$$p_j = 1 - \frac{d(x_i, x_j)}{\sum_{j=1}^{N_p} d(x_i, x_j)} \tag{2.10}$$

where, $d(x_i, x_j)$ is the Euclidean distance between vectors x_i and x_j. The algorithm uses a roulette wheel method to select three vectors in proportion to their probabilities from the population.

A bi-criteria mutation scheme is recommended by Wang *et al*. (2014) considering both the fitness value and the population diversity to achieve a proper balance between exploitation and exploration. The algorithm uses two objective functions, one is the actual function to be minimized and the other is the summation of the pair-wise Euclidian distance between the individuals of the population. In each generation, the solutions are subjected to a non-dominated sorting defined by Deb *et al*. (2002). This type of sorting generates a set of mutually non-dominated (in Pareto sense) solutions, called the Pareto optimal front. Thus, to successfully rank each individual of the population, another round of sorting is required within each front. This second sorting can be done based on a randomly picked objective function. A simple roulette wheel model is proposed by using the generated probabilities to select the parents for mutation.

To enhance the process of the generation of new population, Cai *et al.* (2011) proposed to use a one-step *k*-means operation alongside the DE trial vector strategy. Liu *et al.* (2012) further modified the proposed algorithm by introducing two multi-parent crossovers over the one-step *k*-means to generate trial vectors. They also introduced a small alteration in the DE/rand/1 scheme by imposing the condition that the base vector, selected from the current population, must be fitter than the target vector.

Zhong *et al.* (2013) presented a dual-population DE (DP-DE) to control exploitation and exploration capabilities of DE. The two populations are used to serve different purposes in the search process. One population (GP) uses an explorative strategy and maintains diversity while the other (LP) performs an exploitative search over the neighbourhoods. A new migration strategy was proposed after a regular selection operator to facilitate an interaction between the two populations. In DP-DE, different mutation strategies are applied to the populations based on the irrespective purposes.

Yang *et al.* (2015) constructed an automatic population enhancement scheme that would check each dimension to identify a convergence and diversify that dimension to a satisfactory level, thereby aiding DE to escape from a local minimum and stagnation. To quantify diversity, mean and standard deviation for each of the dimensions is calculated for the population. A lower standard deviation in a direction indicates lower diversity, so for each dimension, a threshold is maintained, if the standard deviation is found to be below the threshold, the dimension is called converged.

2.2.2 DE in Complex Optimization Scenarios

There has been a significant advance in research to adopt DE for optimization in complex environments that include optimization with nonlinear constraints, multiple objectives, dynamic and noisy fitness landscapes and very high dimensionality of the search space. Mohamed and Sabry (2012) proposed a modified DE to handle constraint optimization problems. This variant of DE comes with a mutation scheme, a strategy for choosing the parameters and a constraint handling policy. A new type of mutation is proposed where the base vector is added with the scaled difference of the global best and worst vector. The selection process is modified to select a trial based on any of the following three criteria (1) if it is fitter than the target (when both are feasible), (2) if it has less penalty for constraint violation than the target (when both are infeasible), or (3) if it is feasible while the target is infeasible. The problem of constrained optimization demands not only to optimize a function but also to respect the constraints imposed upon its dimensions. A way to tackle this kind of problems is to quantify the overall constraint violation of a solution and try to minimize it alongside optimizing the function.

Zhong and Zhang (2011) presented an adaptive multi-objective DE with stochastic coding strategy (AS-MODE) where each individual in the DE population is represented by a multivariate Gaussian with a diagonal covariance matrix. A simple DE/rand/1/bin strategy is used for generating trial vectors. However, the vectors participating in the mutation process are chosen by using a tournament selection instead of picking them at random. The selection process involves a non-dominated sorting followed by the crowding distance based operation to rank the solutions of the set, from where top N_p solutions are picked for the next generation. The algorithm, however, introduces six new parameters apart from the three usual parameters (F, Cr, and N_p) of DE.

Rakshit *et al.* (2014) developed a modified version of a popular multi-objective DE algorithm known as DEM (Robič and Filipič, 2005), which can address MOPs in a noisy environment. The major problem of such type of environment is that the fitness value of an individual changes over sampling. To tackle this issue, the authors proposed a simple alteration of the initialization and selection step of DEMO to apply three strategies. First, an adaptive sample size is suggested to measure the fitness of any individual in a noisy environment. Next, the significance of using of the expected value and variance of fitness rather than simple averaging is established, and lastly, a comparison technique is used to deeply investigate the chance of a slightly worse trial to be placed in the Pareto optimal front.

Several optimization problems in the real world are dynamic in nature. Mukherjee *et al.* (2014) proposed a new dynamic DE algorithm, using clustering to generate sub-population, a crowding-based technique to maintain the diversity and local information, and a new crowding based archive to help the algorithm adapt to a dynamically changing environment. Das *et al.* (2014) suggested a dynamic DE algorithm where they used the popular multi-population approach accompanied with two special types of individuals in each subpopulation to maintain the diversity known as Quantum or Brownian individuals and do not follow the DE rules. The algorithm also employs a neighbourhood-driven double mutation strategy to control the perturbation and thereby prevents the population from converging too quickly with the hope to avoid premature convergence. In addition, an exclusion rule is used to spread the subpopulations over a larger portion of the search space as this enhances the optima tracking ability of the algorithm. Furthermore, an aging mechanism is incorporated to prevent the algorithm from stagnating at any local optimum.

Zhao *et al.* (2009) introduced a hybrid DE algorithm (HtDE) based on the concept of the transform functions and proved the convergence of the same under some restrictive assumptions. He *et al*. (2010) used the so-called Differential Operator (DO) to obtain a random mapping from the decision variable space to the Cartesian product of the former and subsequently investigated the asymptotic convergence of DE by using the random contraction mapping theorem.

Hu *et al*. (2014) derived two sufficient conditions that can assure the convergence of DE in the usual sense. The DE-variants can guarantee convergence to a globally optimal point, provided the probability of generating an actual optimum (or optima) by the reproduction operators in each generation in a certain sub-sequence of the population remains greater than a small positive number. The fundamental problem with this approach is that they consider the distribution of the population in each iteration to be independent of each other, which is generally not the case, as any population in an iteration is completely determined by the previous iteration. Hence, even intuitively it is not acceptable that the probability distribution of the population in each iteration would be independent of the distributions in earlier iterations.

Wang *et al.* (2013) proposed a parallel DE scheme by using an adaptive parameter control and Generalized Opposition Based Learning (GOBL) (Wang *et al.*, 2011), which is useful for high dimensional optimization problems. This variant can also be implemented in a parallel processing environment, for instance in a graphical processing unit (GPU), which can provide a massively large and fast computational power. GOBL for every solution creates an opposite solution and it retains a dynamic range of the dimensions of the population, such that the knowledge of the shrinking search space with generations can be kept in record. The proposed algorithm either applies GOBL or classical DE with a probability. In GOBL, after updating the dynamic range of a solution, opposite solutions are generated to form another population. The N_p best individuals are selected from the union of the current and the opposite population, to form the population for the next generation. While the above algorithm is for executing in CPU, the authors also presented an implementation scheme, by defining the kernel functions, to execute it in a GPU.

2.2.3 Applications of DE to Engineering Optimization Problems

With the growing popularity of DE amongst the practitioners, parallel to the core algorithmic research in DE, the application-specific research on and with DE also spiked over the last 5 years. Multi-objective DE (Basu, 2011) and ECHT-DE (Mallipeddi *et al.*, 2011) have been proposed and used to solve economic dispatch problems. Power distribution reconfiguration is determined by applying discrete DE (Prado *et al.*, 2014). In artificial neural network, self-adaptive DE (Dragoi *et al.*, 2013) is considered for an optimal network topology problem. Liao *et al.* (2012) proposed two hybrid DE variants for transport sequencing in cross docking systems. Real time object tracking is done by DE with modified mutation and crossover (Nyirarugira and Kim, 2013).

There has been a vast application of DE on engineering optimization problems. In pattern recognition, adaptive DE with multiple strategies (Dong *et al.,* 2014) is used for clustering while hybrid of self-adaptive PSO and DE (Zhai and Jiang, 2015) is applied to classification problems and MOEA/D-DE by Paul and Das (2015) is employed to feature selection problems. MOEA/D-DE (Sengupta *et al.*, 2012) is also implemented for sleep-scheduling in wireless sensor networks.

In the fields of robotics and expert systems, Vasile *et al.* (2011) proposed a variant of DE known as inflationary DE and applied it to space trajectory optimization. Chen *et al*. (2015), a variant of multi-objective DE is used in satellite orbit reconfiguration. Moving object detection is solved by a distributed DE with neighbourhood based mutation

(Ghosh *et al*., 2014). For Hypoglycaemia detection, multi-objective DE, with double wavelet mutation (Lai *et al*., 2013) is used.

2.3 Electric Vehicle Charging Problem

Electric Vehicles (EVs) are welcoming a rapid development along with progresses of relevant technologies in recent years. As an eco-friendly substitute for traditional fuel-engine vehicle, EV is a promising solution to the ever-devastating energy crisis and environmental pollution around the globe, and thus has drawn increasing attentions from the public, markets, decision-makers, and academia. Their large-scale utilization has the potential to reduce greenhouse gas emission, save fuel costs for EV drivers, and increase the use of renewable energy (Zhao *et al.,* 2011).

While the EV charging station location problem is a very new topic area, some important strides are made in the past few years. Morrow *et al*. (2008) show how an EV-based transport system's overall cost can be reduced by providing more charging infrastructure instead of investing in bigger batteries (with greater range). They estimated that the marginal cost of increasing a car's all-electric range (AER) from 10 miles to 40 miles is $8,268, and the cost of installing an additional level-2 commercial charging station (including administrative and circuit installation costs, assuming 10 charge cords per facility) is $18,520.

Wang *et al*. (2010) created a numerical method for the layout of charging stations using a multi-objective planning model. Accounting for charging station attributes, distribution of gas-station demands (rather than parking decisions, as a proxy for charging demands), and power grid infrastructure, among other variables, they tested and verified their model using data from Chengdu, China. Sweda and Klabjan (2011) used an agent-based decision support system to identify patterns of residential EV ownership and driving activities to determine strategic locations for new charging infrastructure, with the Chicago region as a case study. Most station location problems are based on existing optimization routines/heuristics.

Worley *et al*. (2012) formulated the problem of locating stations and optimal EV routings as a discrete integer programming problem, based on the classic Vehicle Routing Problem (VRP). Ge *et al*. (2011) proposed a method based on grid partition using genetic algorithms. Their routine focuses on minimizing users' loss or cost to access charging stations after zoning the planning area with a grid partition method by choosing the best location within each partition, to reflect traffic density and station capacity constraints (which include charging power, efficiency, and number of chargers per station).

Knezović and Marinelli (2016) proposed a voltage-dependent EV reactive power control for grid support to raise the minimum phase-to-neutral voltage magnitudes and to improve voltage dispersion. However, it needs local voltage measurements. Another local control technique is also proposed by Richardson *et al.* (2013) whereby individual electric vehicle charging units attempt to maximize their own charging rate along with the information about the instantaneous voltage of their own point and loading of the service cable.

Li *et al*. (2011) also used genetic algorithms to identify top locations for charging infrastructure. Their method is based on the conservation theory of regional traffic flows, taking EVs within each district as fixed load points for charging stations. The number and distribution of EVs are forecasted, and the cost-minimizing charging station problem is (heuristically) solved using genetic algorithms.

Frade *et al*. (2011) used Lisbon, Portugal as a case study, for application of a maximal covering location model to maximize the EV charging demand served by an acceptable level of service. They determined not just the locations, but also the capacity of stations to be installed at each location. Finally, Kameda and Mukai (2011) developed an optimization routine for locating charging stations, relying on taxi data and focusing on stations for Japan.

Locment *et al*. (2015) presented an evaluation on PV micro-grid power architecture for efficient charging of plug-in EVs from the aspects of theoretical and numerical. Aziz *et al.* (2015) studies showed that the application of EVs and used EV batteries in supporting certain small-scale energy management systems is feasible. Liu *et al*. (2015) established multi-objective economic dispatch models of a microgrid with EVs charging under the autonomous charging mode.

Honarmand *et al*. (2014) proposed a method to solve this problem by considering practical constraints, renewable power forecasting errors, spinning reserve requirements, and EV owner satisfaction. The modeling results indicate that EV owners can profit by either discharging the batteries of their vehicles or providing the reserve capacity during departure time. Zakariazadeh *et al*. (2014) proposed a multi-objective operational scheduling method for EV charging in a smart distribution system. V2G capability and actual driver patterns are considered in this method. The findings show that the proposed method can lower both operation cost and air pollutant emissions.

Wong *et al*. (2010) proposed a muti-objective planning model for the placement of EV charging stations in Chengdu, China, with a solution based on demand and usage of existing gas stations. Chen *et al*. (2013) particularly considered EV users' costs for accessing charging stations, and minimizing the costs and penalizing unmet demand. Moreover, He *et al*. (2013) took a broader view and emphasized the impact on overall efficiency of a transportation system when optimizing the placement.

Most of the existing researches have focused on developing plug-in charging strategies, which could be divided into two classes, i.e., centralized and decentralized ones. The former determines at when and where and what rate every vehicle should be charged such that they use less expensive electricity and EV load is shifted to off-peak hours. All decisions in this class of strategies could be made based on the system-level concerns such as mitigating total losses and feeder congestion.

Zou *et al*. (2011) propose a centralized charging strategy of PHEVs by employing a dynamic estimation interpolation (DEI) based algorithm. It considers the valley-filling effect of the supply side and minimizes the users' cost by developing a price discount scheme. Besides, some algorithms have been developed to coordinate a practical number of EVs in a power system with different concentrations. However, the challenge is its difficulty to implement the centralized charging control under the plug-in mode by considering the stochastic charging behaviour of EV users.

CHAPTER 3

DE ALGORITHM WITH MULTI-POPULATION STRATEGY

A DE algorithm is a population-based algorithm like genetic algorithms by using the similar operations; crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operation. This main operation is based on the differences of randomly sampled pairs of solutions in the population.

A basic idea behind DE is generating trial vectors by adding weighted difference vector between two population members to a third member. If the resulting vector yields a better objective function value than an initial vector, the trial vector replaces the vector with which it is compared. In addition, the best parameter vector is evaluated for every iteration to keep track of the progress that is made during the optimization process.

The DE algorithm also uses a non-uniform crossover that can take child vector parameters from one parent more often than it does from others. By using the components of the existing population members to construct trial vectors, the recombination (crossover) operator efficiently shuffles information about successful combinations, enabling the search for a better solution space.

An optimization task consisting of D parameters can be represented by a D-dimensional vector. In DE, a population of N_P solution vectors is randomly created at the start. This population is successfully improved by applying mutation, crossover and selection operators. The main steps of a DE algorithm are given below:

Initialization

Evaluation

Repeat

Mutation

Recombination

Evaluation

Selection

Until (*termination criteria are met*)

DE maintains two arrays, the primary array holds the current vector population, while the secondary array accumulates vectors that are selected for the next generation. Every pair of vectors (X_a, X_b) defines a vector differential: (X_a - X_b). When X_a and X_b are chosen randomly, their weighted differential is used to perturb another randomly chosen vector X_c. This process can be mathematically expressed as:

$$X_c' = X_c + F \cdot (X_a - X_b \tag{3.1}$$

The weighting, or scaling, factor F is a user supplied constant in the optimal range between 0.5 and 1.0 (Evan *et al.*, 2008). In every generation, each primary array vector X_i is targeted for crossover with a vector like X_c' to produce a trial vector X_t. Thus, the trial vector is the child of two parents, a noisy random vector and the target vector against which it must compete.

Uniform crossover is used with a crossover constant (C_R), in the optimal range of 0.5 to 1.0, which represents the probability that the child vector inherits the parameter values from the noisy random vector. When C_R = 1, for example, every trial vector parameter certainly comes from X'_c. On the other hand, if $C_R = 0$, all but one trial vector parameter comes from the target vector.

To ensure that X_t differs from X_i by at least one parameter, the final trial vector parameter always comes from the noisy random vector even when $C_R = 0$. Then the objective function corresponding to the trial vector is compared with that of the target vector, and the vector that has the better objective function value of the two would survives into the next generation. This process is continued until a termination criterion is met and difference in objective function values between two consecutive generations reaches a small value. Figure 3.1 shows how a DE algorithm works.

Price & Storn (1997) gave the working principle of DE with a single strategy. Later, they suggested ten different strategies for DE. These strategies can be adopted in a DE algorithm depending upon the type of problems to which DE is applied. The strategies can vary based on the vector to be perturbed, number of difference vectors considered for perturbation, and finally the type of crossover used. The ten different working strategies are given in Table 3.1.

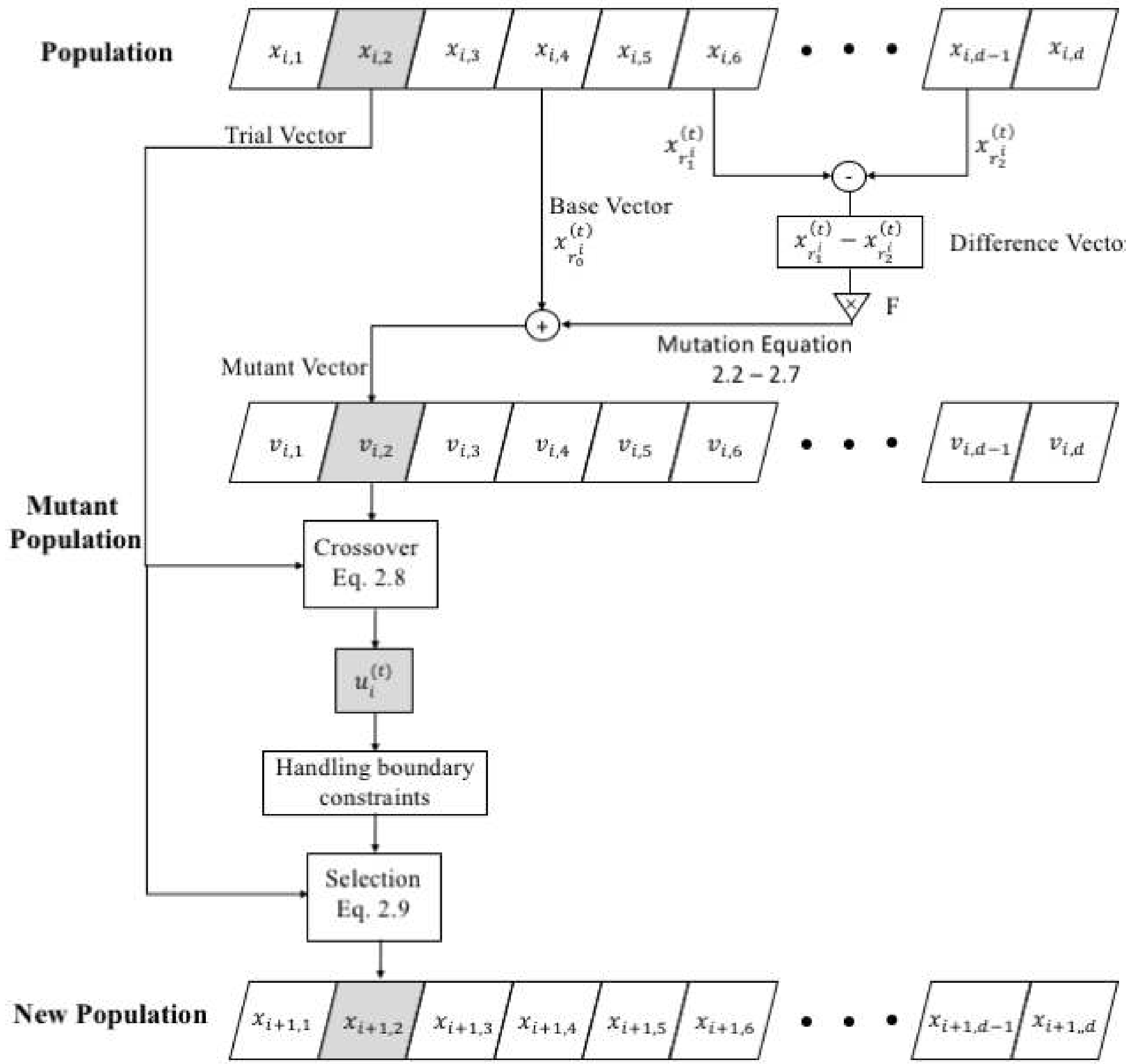

Figure 3.1 Schematic of a differential evolution algorithm.

Table 3.1 Mutation Strategies with expression

Mutation Strategy Name	**Expression**
DE/best/1/exp	$v_i^{(t)} = x_{best}^{(t)} + F \cdot (x_{r_1^i}^{(t)} - x_{r_2^i}^{(t)})$
DE/rand/1/exp	$v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot (x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)})$
DE/rand-to-best/1/exp	$v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot \left(x_{best}^{(t)} - x_{r_1^i}^{(t)}\right) + F \cdot (x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)})$
DE/best/2/exp	$v_i^{(t)} = x_{best}^{(t)} + F \cdot \left(\square_{r_1^i}^{(t)} - x_{r_2^i}^{(t)}\right) + F \cdot (x_{r_3^i}^{(t)} - x_{r_4^i}^{(t)}$
DE/rand/2/exp	$v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot \left(x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)}\right) + F \cdot (x_{r_4^i}^{(t)} - x_{r_5^i}^{(t)}$
DE/best/1/bin	$v_i^{(t)} = x_{best}^{(t)} + F \cdot (x_{r_1^i}^{(t)} - x_{r_2^i}^{(t)})$
DE/rand/1/bin	$v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot (x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)})$
DE/rand-to-best/1/bin	$v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot \left(x_{best}^{(t)} - x_{r_1^i}^{(t)}\right) + F \cdot (x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)})$
DE/best/2/bin	$v_i^{(t)} = x_{best}^{(t)} + F \cdot \left(x_{r_1^i}^{(t)} - x_{r_2^i}^{(t)}\right) + F \cdot (x_{r_3^i}^{(t)} - x_{r_4^i}^{(t)}$
DE/rand/2/bin	$v_i^{(t)} = x_{r_1^i}^{(t)} + F \cdot \left(x_{r_2^i}^{(t)} - x_{r_3^i}^{(t)}\right) + F \cdot (x_{r_4^i}^{(t)} - x_{r_5^i}^{(t)}$

The general convention used above is *DE/x/y/z*. DE stands for Differential Evolution, *x* represents a string denoting the vector to be perturbed, *y* is the number of different vectors considered for perturbation of *x*, and *z* stands for the type of crossover being used (*exp*: exponential; *bin*: binomial). Hence, the perturbation can be either in the best vector of the previous generation or in any randomly chosen vector. Similarly, either single or two vector differences can be used for perturbation. Through perturbation with a single vector difference, the weighted vector differential of any two vectors out of the three distinct randomly chosen vectors, is added to the third one. Through five distinct vectors, other than the target vector are chosen randomly from the current population. Out of these, the weighted vector difference of each pair of any four vectors is added to the fifth one for perturbation.

Exponential crossover, is performed on D variables in one loop until it is within the C_R bound. The first time a randomly picked number between 0 and 1 goes beyond the C_R value, no crossover is performed and the remaining D variables are left intact. Binomial crossover, is performed on each of the D variables whenever a randomly picked number between 0 and 1 is within the C_R value. So for high values of C_R, the exponential and binomial crossover methods yield similar results.

A strategy that works out to be the best for a given problem may not work well when applied to a different problem. Also, the strategy and key parameters to be adopted for a problem are to be determined by trial and error. However, strategy-7 (*DE/rand/1/bin*) appears to be the most successful and the most widely used strategy.

All solutions in the population have the same chance of being selected as parents without dependence of their fitness value. The child produced after the mutation and crossover operations is evaluated. Then, the performance of the child vector and its parent is compared and the better one is selected. If the parent is still better, it is retained in the population.

3.1 Idea behind Multi-population Strategy

In a multi-population approach, the entire population is divided into a predefined number of sub-populations. The size and population members of these sub-populations are kept unchanged during the algorithm's execution. Each sub-population can exchange information with any other sub-population.

In most of the multi-population evolutionary algorithms, migration is used as a means of communication between sub-populations. Different from these algorithms, sub-populations in our multi-population approach exchange information via the mutation operation. Various multi-population approaches for DE have been designed to solve different kinds of optimization problems. Most of these approaches maintain population diversity via information exchange among different sub-populations. Tasoulis *et al.* (2004) parallelized DE in a virtual parallel environment so as to improve its computing performance. In order to promote information sharing, the best individuals from each sub-population are allowed to migrate to other sub-populations based on a ring topology. Another migration scheme for multi-population was proposed by Kozlov *et al.* (2006). The authors suggested substituting the oldest individual of the target sub-population instead of a randomly chosen one.

Song and Hou (2015) proposed a multi-population multi-strategy improved differential evolution (MPMSIDE) algorithm in which, the population is divided into three different subpopulations according to the fitness value, standard deviation of fitness and distance between two individuals, which are best population with the better fitness of individuals, worst population with the poor fitness of individuals and general population with the rest individuals. The best population is responsible for local search and improves the convergence speed and precision. The worst population is responsible for global search, jumps out the local optimum and avoids premature convergence. The general population is responsible for balancing the global search ability and local search ability.

The local optimization strategy is used to avoid the local extreme point and improve the local hill-climbing ability in the local search. The self-adaptive update strategy determines the similarity between the best individual and the general individual according to the individual similarity coefficient for reducing the adverse effects of the linear adjusting scaling factors and making the parameter sensitivity of a DE algorithm and improving the stability and robustness.

Yu and Zhang (2011) suggested a multi-population approach for the DE variant known as DE/best/1, which uses the best solution information to guide the search. The DE/best/1 strategy has a fast convergence rate but easily suffers from premature convergence due to early loss of population diversity. The entire population is divided into multiple sub-populations, which evolve on their own. The size and number of the sub-populations are predefined and kept unchanged after initialization. During the evolutionary process, each sub-population can exchange information with any other sub-population.

Most of the multi-population EAs use migration as a means of communication among different sub-populations. However, their performance is sensitive to the choice of control parameters such as migration size and rate. Instead of using migration, the sub-populations can communicate with each other by means of a novel mutation operation, which involves a best vector and a difference vector. The former is selected from the corresponding sub-population instead of the entire population, which can balance the fast convergence and population diversity.

On the other hand, the difference vector generated by two vectors is selected from the entire population. Therefore, it may contain information from two different sub-populations and can be used as a medium of information exchange.

3.2 Multi-population Strategy Applied to DE

Due to the weak global search ability, the stability and time consumption of optimization algorithms in solving a high dimensional optimization problem, an improved differential evolution (DE) algorithm with multi-population strategy for solving high dimensional optimization problems is proposed.

The aim of this work is to analyse a parallel implementation of the differential evolution based on a multi-population model. The reasons of choosing a multi-population model are: (i) it is inspired from the spatial structure of natural populations; and (ii) its ability of preserving the population diversity through the migration process.

The multi-population strategy implemented in the former paper (Song and Hou, 2015) illustrates the generation of three subpopulations based on the fitness value of the individual vectors in the initial set. The three mutation techniques applied are best/1 to the subpopulations with the higher fitness values, rand/1 to the lower fitness values and current to best to the subset with average fitness value.

This thesis examines a DE-based on multi-population strategy for solving high dimensional optimization problems. Following the initialization operation, the initial population is randomly divided into multiple subpopulations. In each iteration, the mutation, crossover and selection component are sequentially executed until the algorithm fulfils its termination criterion.

The mutation is the key operation of a DE algorithm; the selected mutation strategy determines population direction in the process of evolution. The multiple-mutation strategy is introduced to avoid possible stagnation in a local minimum value for dealing with complex functions with high dimension multimodal optimization problems, and the premature loss of population diversity.

The multiple-mutation strategy is initiated to increase the global optimization ability of a greedy algorithm. The selected techniques used for mutation are "DE/rand/1" and "DE/rand to best/1". The individuals from both subpopulations are compared to obtain optimal individual as the result. The flowchart of the proposed 2-subpopulation DE algorithm is shown in Figure 3.2. The multi-population strategy helps in maintaining the evolution of the best individual and enhances the local hill-climbing ability in the local search. It contributes to avoiding the local extreme points and premature convergence in an optimization process.

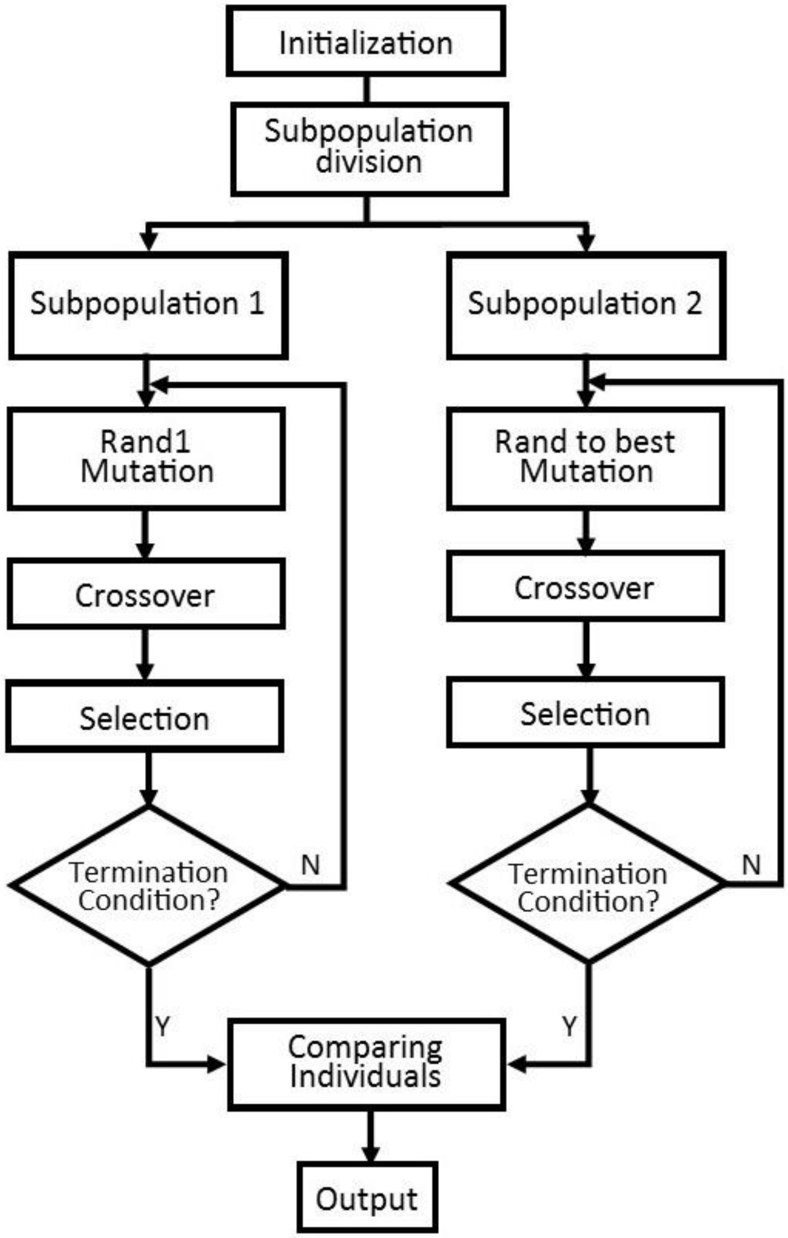

Figure 3.2 Flowchart of 2-subpopulation differential evolution algorithm.

The process is executed to keep the best individuals to achieve the dynamic exchange information. The multi-population strategy keeps the evolutionary stability of the best individuals while avoiding the premature convergence in the evolutionary process.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Testing Functions and Algorithms

Eleven benchmark functions are chosen to verify the performance of the proposed algorithm. The classic functions from the benchmark testing set include Ackley function, Rastrigin function, Whitley function, Schaffer function, Rosenbrock's function, Modified double sum, Sphere function, Ridge function, Schwefel 2.21 function, Lunacek's bi-Rastrigin function and Levy function. These particular function expressions along with their global minimum value (opt.) and range are shown in Table 4.1.

Table 4.1 Benchmark Testing Functions

Index	Function	Expression	Opt.	Range
f_1	Ackley	$f(x) = -a\, exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d} cos(cx_i)\right) + a + \exp(1)$	0	[-32.768, 32.768]
f_2	Rastrigin	$f(x) = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10cos(2\pi x_i)\right)$	0	[-5.12, 5.12]
f_3	Whitley	$f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\left(100\left(x_i^2 - x_j\right)^2 + \left(1 - x_j\right)^2\right)^2}{4000} - \cos\left(100\left(x_i^2 - x_j\right)^2 + \left(1 - x_j\right)^2\right) + 1\right)$	0	[-10.24, 10.24]

*f*4	Schaffer	$f(x) = 0.5 + \frac{sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	-1	[-100, 100]
*f*5	Rosenbrock's	$f(x) = \sum_{i=1}^{n} \begin{pmatrix} 100(x_i - x_{i-1}^2)^2 + \\ (x_{i-1} - 1)^2 \end{pmatrix}$	0	[-30, 30]
*f*6	Modified double sum	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} (x_j - j)^2 \right)$	0	[-10.24, 10.24]
*f*7	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	0	[-5.12, 5.12]
*f*8	Ridge	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	0	[-64, 64]
*f*9	Schwefel 2.21	$f(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	0	[-30, 30]
*f*10	Lunacek's bi-Rastrigin	$f(x) = min \begin{pmatrix} \left\{ \sum_{i}^{n} (x_i - 2.5)^2 \right\}, \\ \left\{ d.n + s. \sum_{i}^{n} (x_i - \mu_2)^2 \right\} \end{pmatrix}$ $+10 \sum_{i}^{n} (1 - cos2\pi(x_i - 2.5))$ $where, \mu_2 = -\sqrt{\frac{\mu_1^2 - d}{s}}$	0	[-5.12, 5.12]
*f*11	Levy	$f(x) = sin^2(\pi\omega_1)$ $+ \sum_{i=1}^{d-1} (\omega_i - 1)^2[1 + 10sin^2(\pi\omega_i + 1)]$ $+ (\omega_d - 1)^2[1 + sin^2(2\pi\omega_d)]$ $where, \omega_i = 1 + \frac{x_i - 1}{4}$	1	[-10, 10]

The initial population is divided into various subpopulations and several mutation strategies are applied to each subpopulation. These different multi-population strategies along with number of subpopulations and mutation strategies applied are shown in Table 4.2.

Table 4.2 Multi-population Strategies Used

Index	Number of subpopulations	Mutation Strategies
a_1	2	rand/1, rand to best
a_2	2	rand/1, best/1
a_3	3	rand/1, best/1, rand to best
a_4	4	rand/1, best/1, rand to best, current to best
a_5	5	rand/1, best/1, rand to best, current to best, rand/2
a_6	6	rand/1, best/1, rand to best, current to best, rand/2, best/2

4.2 Experimental Results and Analysis

The experimental parameters used for all the multi-population strategy differential evolution algorithms are given as follows: population size N_P = 500, functional dimension = 30, crossover probability factor C_R = 0.6, and scaling factor (F) = 0.6. Each algorithm terminates when the number of iterations reaches not more than 580. Each algorithm is run independently 30 times for all the eleven functions.

4.2.1 Experimental Results

The experimental results of all algorithms in Table 4.2 along with the multi-population strategy implemented in the former paper (*old*), applied to the functions mentioned in Table 4.1 with their maximum value, minimum value, mean value and standard deviation are shown in Table B1.

Eleven benchmark functions are optimized with the multi-population strategy algorithms mentioned in Table 4.2. The obtained maximum value, mean, minimum value and standard deviation are chosen to analyse the performance. From Table B1 it can be observed that a_1 has the most optimal mean value for the functions f_1 - f_5, f_7 - f_8 and f_{10} - f_{11}. Hence the multi-population strategy of dividing the population into two subpopulations and performing rand/1 and rand to best with the individual subpopulations achieves a better global convergence ability in solving high dimensional optimization problems for nine out of eleven benchmark functions.

4.2.2 Effect of Number of Subpopulations on Execution Time and Mean Value

The execution time and mean for the multi-population DE are mapped in a graph where the population size (N_P) is 2000 and the initial population is divided from 2 to 40 subpopulations with different mutation strategies applied to each subpopulation in each experiment. Figures 4.1-4.11 show the normalized execution time and normalized mean values plotted against the number of subpopulations (N_s) for f_1 - f_{11} functions respectively.

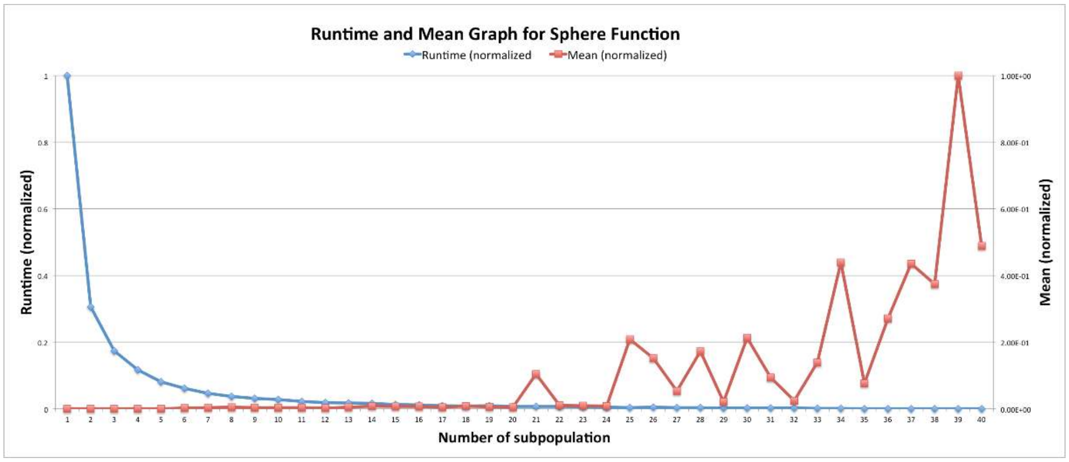

Figure 4.1 Execution time and mean against N_s for Sphere function.

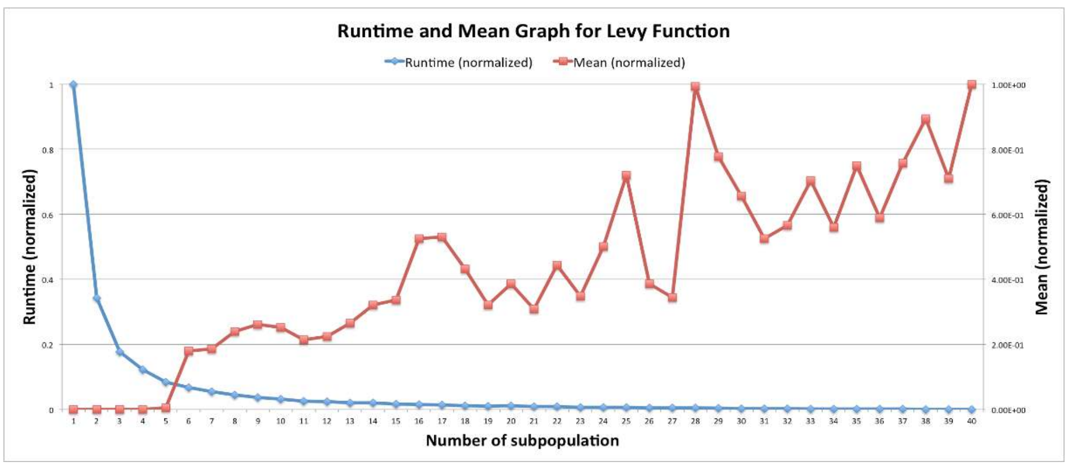

Figure 4.2 Execution time and mean against N_s for Levy function.

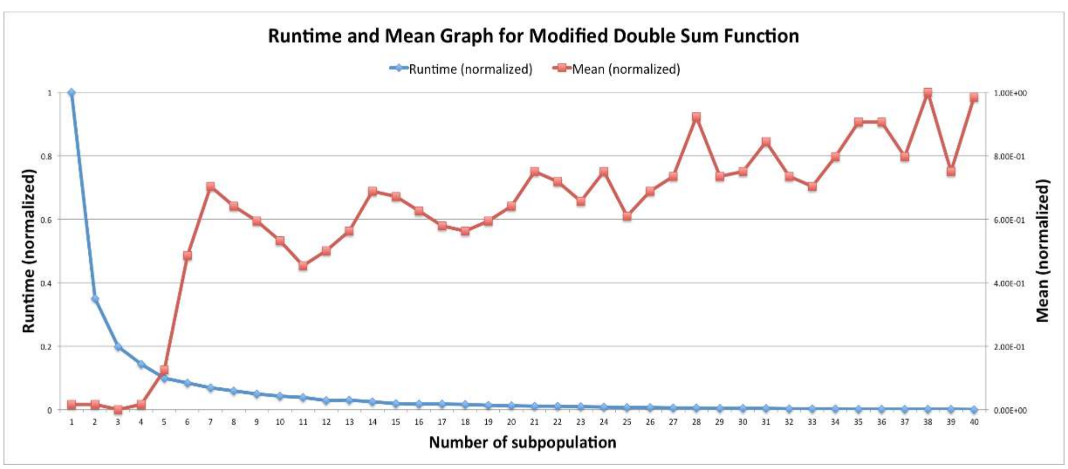

Figure 4.3 Execution time and mean against N_s for Modified double sum function.

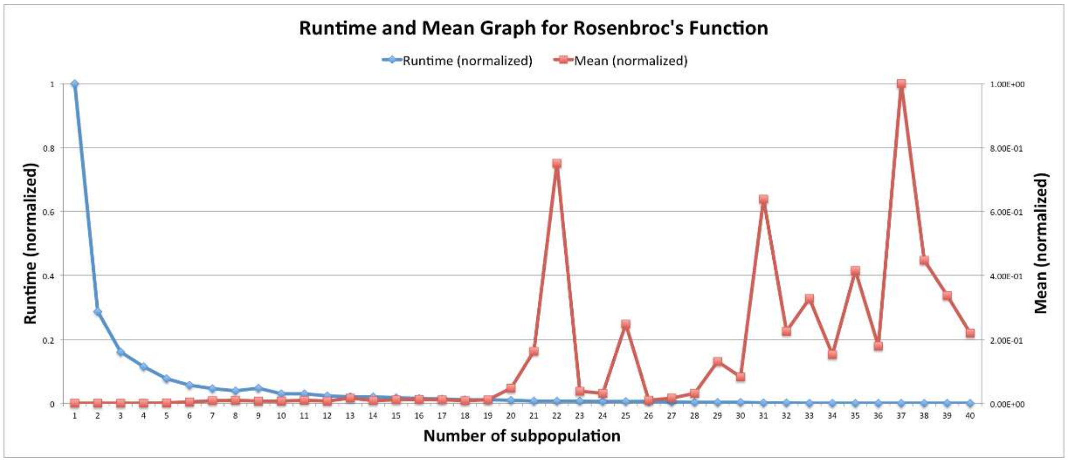

Figure 4.4 Execution time and mean against N_s for Rosenbroc's function.

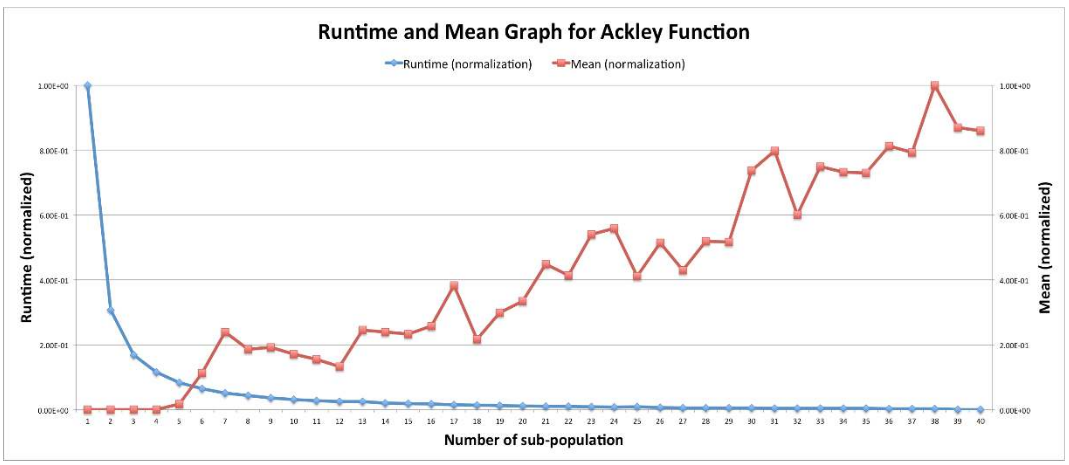

Figure 4.5 Execution time and mean against N_s for Ackley function.

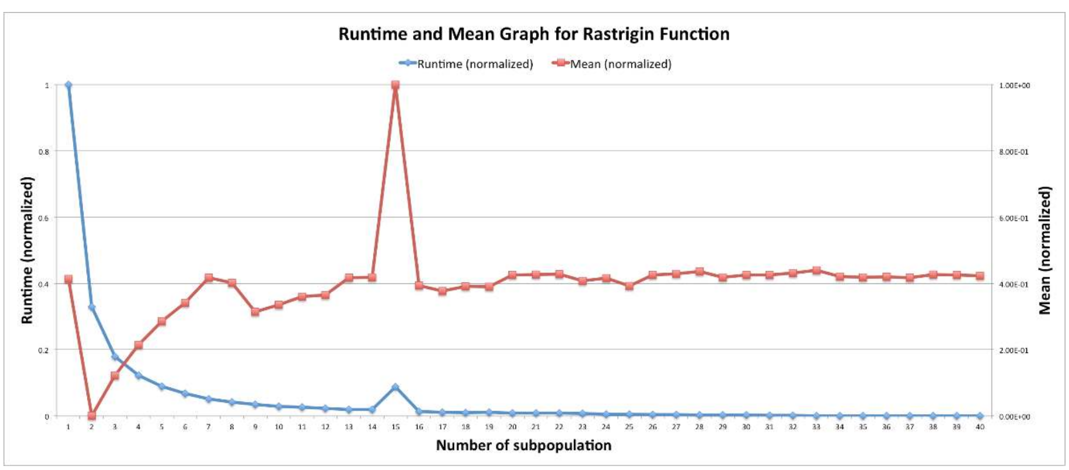

Figure 4.6 Execution time and mean against N_s for Rastrigin function.

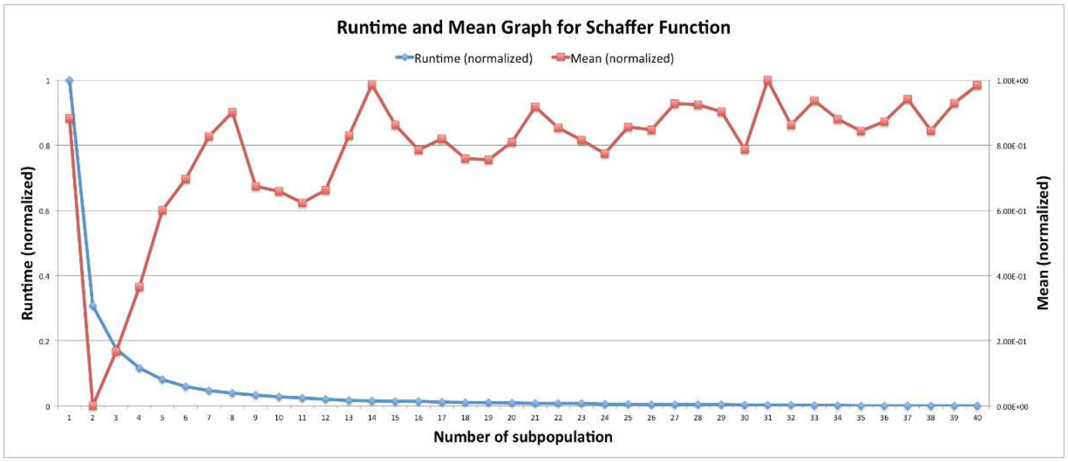

Figure 4.7 Execution time and mean against N_s for Schaffer function.

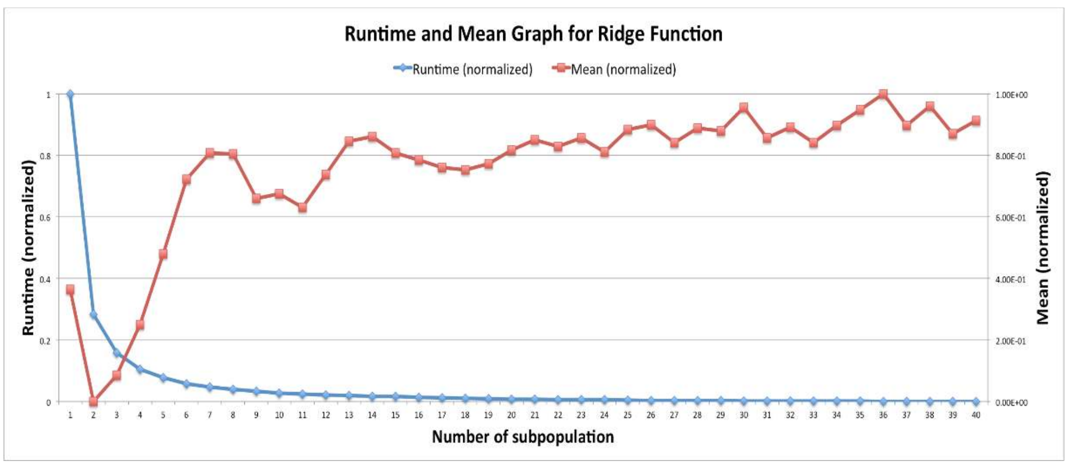

Figure 4.8 Execution time and mean against N_s for Ridge function.

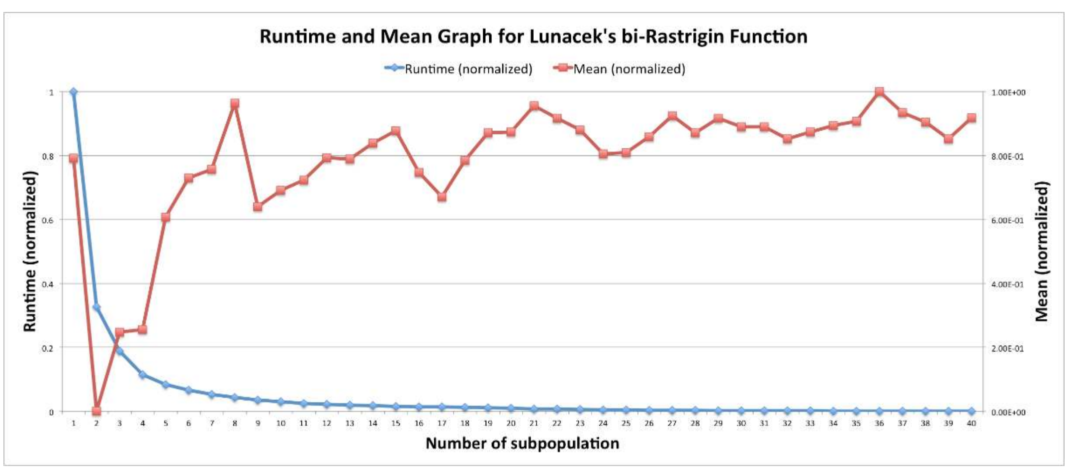

Figure 4.9 Execution time and mean against N_s for Lunacek's bi-Rastrigin function.

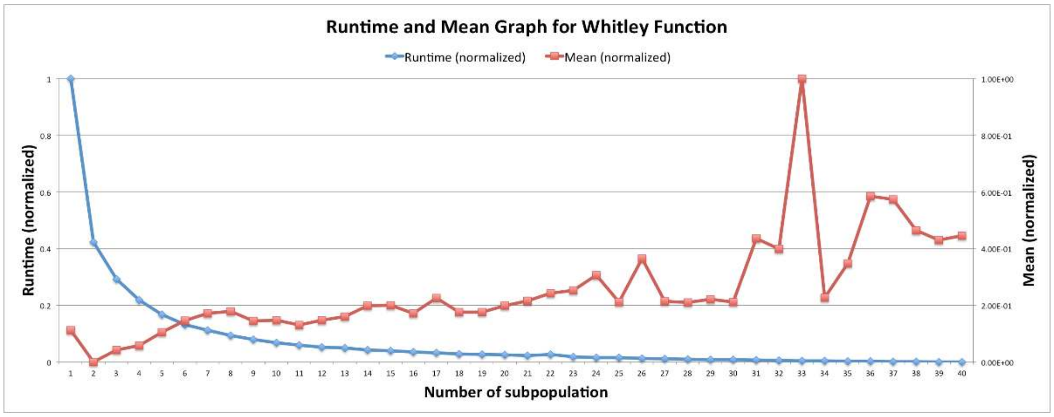

Figure 4.10 Execution time and mean against N_s for Whitley function.

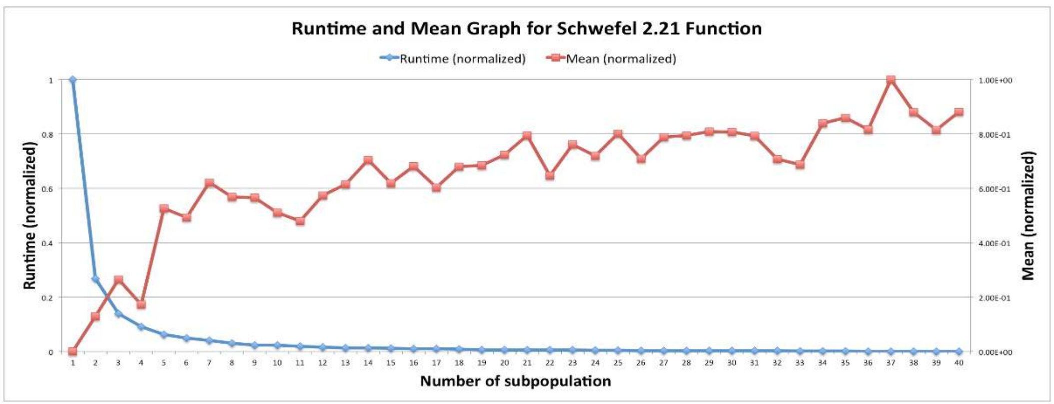

Figure 4.11 Execution time and mean against N_s for Schwefel 2.21 function.

Figures 4.1-4.11 exhibit that the execution time reduces drastically when N_s is less than or equal to 8; otherwise the reduction is nominal. Figures 4.1-4.5 show that the optimization result degrades negligibly until $N_s = 6$, and thereafter the result drops noticeably. Furthermore, Figures 4.6–4.10 indicate substantial improvement in optimization results when N_s changes from 1 to 2 and then the result decreases distinctly. Figure 4.11 reflects an exception to both the above cases by producing the optimum value when there is no multi-population strategy applied, on the contrary weakening the performance as N_s increases.

CHAPTER 5

ELECTRIC VEHICLE CHARGING PROBLEM

Electric vehicles (EVs) have emerged as one of the most interesting and promising solutions to reduce the levels of greenhouse gas emissions. With rapid development of high-capacity Li-ion batteries, high-efficiency motor drives, and power electronics, and integrated EV control and management, EVs have entered the large-scale commercialization stage (Liu *et al.*, 2014). To support a large number of EVs, high-capacity and high-efficiency charging infrastructures are mandatory to sustain the growing charging demands and to improve pure electric driving mileages and operational economy of EVs (Alonso *et al.*, 2014).

According to Zhang *et al.* (2011), EVs parking at home account for more than 75% of the daily parking time, and the average parking duration at night is more than 10. Delayed charging is better than immediate charging at home, and non-home charging increases peak grid loads. A simulation model is presented in (Zhang *et al.*, 2013) to analyse economic and environmental performance of EVs operating under different conditions, including electricity generation mix, smart charging control strategies, and real-time pricing mechanisms. Its results show that 100 kWh excess electricity can be reduced annually per vehicle when the smart charging method is employed to replace the off-peak charging method.

At present, EV charging strategies can be mainly classified into centralized and distributed control. A common feature of a centralized control system is that it directionally communicates with all EVs and manages charging time and power to optimize certain objective functions, such as minimizing carbon dioxide emissions, power loss, cost, or and achieving desired "valley-filling", by using EV data (the connection time to the grid, charge demand, rated voltage, and charger power) (Abdelaziz *et al.*, 2014). Such control strategies require extensive real-time bi-directional communications, with increased cost on communications equipment and resources and, consequently, they are not desirable to charging service providers. Commonly used algorithms in centralized control, including linear programming, quadratic programming, dynamic programming, stochastic programing, robust optimization, and model predictive control, are summarized and presented by Hu *et al.* (2016).

In distributed methods, a central control system broadcasts a common electricity price or a reference power signal to all EVs. Then each EV decides individually, and locally, its charging power and time, based on its own parameters and associated optimization criteria (Soares *et al*., 2017). Katarina and Mattia (2016) propose a voltage-dependent EV reactive power controller for grid support to raise the minimum phase-to-neutral voltage magnitude and to improve voltage dispersion. However, it needs local voltage measurements. A local control technique is proposed (Richardson *et al.*, 2013) whereby individual EV charging units attempt to maximize its own charging rate along with the information about the instantaneous voltage of its own point and loading of the service cable.

From these existing centralized control strategies or distributed control strategies, we can see that they usually need a central unit to control EV charging or broadcast a common reference signal such as electricity price, loading of the service cable, and network constraints, or at least it needs voltage or other local variable measurements for local control strategies.

Electric vehicles (EVs) have an enormous consequence on the daily residential load curve if they are widely connected to the power grid for battery charging (Zou *et al.*, 2011). Due to the uncertainty of their charging behaviours, uncoordinated random charging of a mass of EVs may lead to an unanticipated effect on the normal operation of a power distribution system, i.e., voltage fluctuation, thereby increasing the load peak and off-peak difference in the network. Without taking the spot pricing into consideration, EV owners may pay much higher cost for battery charging. The appropriate dispatching of EVs in a distribution system for their charging represents a challenging demand side management problem (Masoum *et al.*, 2011).

A unique charging strategy of EVs based on optimal charging priority and charging station is suggested under a spot pricing-based electricity market environment (Kang *et al.*, 2016). It is an improved population-based heuristic algorithm which is designed to find the optimal charging priority and location of EVs in a distribution network. It inherits a hybrid algorithm of PSO and GA.

5.1 Model Formulation

Under a spot pricing-based electricity market environment, uncoordinated charging of many EVs may cause sincere impacts on the security and economy of power system operations, such as increasing power losses, overload, voltage fluctuation and charging cost.

To reduce such impacts, various locations and time slots for EV charging may have different influences on power quality. To ensure high power quality, i.e., low voltage fluctuation and power loss, applicable locations and time slots should be found. To subside the vacancy of electric power, EVs should be scheduled for charging during the off-peak time zone. The charging cost should be as low as possible by considering spot electric price and maximum power consumption should be set for every time slot to prevent an overload condition from EV charging. Shao *et al*. (2011) have suggested that a new peak demand may emerge if all EV owners have a preference for the exact time when the electricity price is the lowest. Thus, when the charging load of EVs reach a given limit, other EVs should be enabled to connect to the distribution system.

To explain the problem more specifically and clearly, some assumptions and explanations are made similar to Kang *et al.* (2016). EVs are divided into several groups. Each group is scheduled as a whole, and thus treated as an EV set, simply named as "EVS" (basic unit of scheduling). To facilitate the simulation, each EVS has the same number of EVs, and the same type of vehicles. However, if each EVS has different number of various types of EVs, the strategy still works.

In order to determine the number of time slots for EVS charging, a statistical distribution of power demand of EVS charging load is established during a dispatch cycle. Power demand of an EV after one day's travel exhibits a good deal of randomness, but it is largely determined by the distance in miles driven by an EV, irrespective of some secondary factors, such as road condition, climate and battery age.

EV aggregator is a communication and control agent between the grid and EVs, it is employed to solve plug-in storming problems. The implementation of centralized charging strategy is performed by EV aggregator. All information of EVs and control signals generated by aggregators can be delivered immediately between EVs and aggregators

(Lan *et al.,* 2013). Aggregator owners are known as price takers, which mean that the total power consumption of EVs do not have a large share to affect the electricity price.

This thesis intends to address an EV centralized charging problem on spot pricing under a battery swapping scenario. The charging locations and time slots in a distribution network are viewed as decision variables of a charging strategy. It can be considered as a multi-objective optimization problem to minimize charging cost, power loss, and voltage deviation.

5.1.1 Charging Rule

The maximum demand level is considered as the maximal value of residential load during a scheduling period. The power for EV charging is expressed as:

$$P_t \leq P_t^{lim}, t \in \mathrm{N_T} = \{1, 2, \dots, \mathrm{T}\} \tag{5.1}$$

$$P_t^{lim} = P_{max} - P_t^{load} \tag{5.2}$$

All vehicles under the centralized charging strategy are fully charged once their charging starts:

$$\sum_{t=1}^{\mathrm{T}} P_t = P_{tot} \tag{5.3}$$

The easiest strategy to handle constraints in population-based heuristic approaches is to assign infeasible individuals an arbitrarily low fitness. A better way to deal with such constraints is to map the search space so as to decrease the number of infeasible individuals or design some strategies/rules to avoid infeasible ones (Fonseca *et al*., 1998).

The load scheduling is transformed into an EV charging rule, if EV charging priority and locations are known. The basic concept is to charge each vehicle at a time slot where the lowest electricity price occurs and power consumption satisfies equation (5.1). The main idea of the EV charging rule states that the time slots are ranked according to their price from the lowest to highest. EVs with high priority have privilege to choose their charging time slot from the lowest to the highest until no empty time slot is left. P_t^{lim} is set as the maximum acceptable power consumption for EV charging at a time slot. If the total power demand at a time slot does not exceed P_t^{lim}, the slot can be chosen for EV charging. Hence, the peak load can be avoided and the charging cost is reduced to a large extent. Then, the amount of power at per time slot and each charging location can be obtained.

5.1.2 Power Flow

It is important for planning future expansion of power systems as well as in determining the best operation of the existing systems. The power flow calculation is necessary to obtain the variation of power and voltage distributions when EVs are connected to a distribution system. The traditional formulation of power flow can be denoted as power balance equation $g(x) = 0$, which is split into its active and reactive components as follows:

$$\begin{cases} g_P(\theta, V_m, P_g) = P_{los} + P_d + P_g = 0 \\ g_Q(\theta, V_m, Q_g) = Q_{los} + Q_d + Q_g = 0 \end{cases} \tag{5.4}$$

where, θ is a voltage angle, V_m is the voltage magnitude, P_g and Q_g are generator injections, P_d and Q_d are load injections and assumed to be constant, P_{los} and Q_{los} are the active power loss and reactive power loss, respectively (Kang *et al.*, 2013). This work uses a Newton method to calculate power flow.

5.1.3 Objectives

The objectives include the minimization of charging cost, power loss, and voltage deviations. Therefore, the following objective function is obtained:

$$f = \min(P_{los} + \sigma . V_{dev} + \gamma . \text{Cost}) \quad (5.5)$$

where, P_{los} is active power loss, V_{dev} denotes load bus voltage deviation from 1.0 p.u., and Cost is the total electricity cost; σ and γ are the non-negative weighting factors.

P_{los} can be obtained with power flow calculation and is represented as $P_{los} = \sum_{t=1}^{T}\sum_{l=1}^{L}|I_{lt}|^2 R_l$ where T is the number of time slots, L is the number of lines in the power system, I_{lt} is the current of the l^{th} line at the t^{th} time slot, and R_l is the resistance of the l^{th} line.

V_{dev} can be denoted as $V_{\text{dev}} = \sum_{t=1}^{\text{T}}\sum_{b=1}^{\text{N}}|V_{bt} - 1.0|(\text{p.u.})$ where T plays the same role as that in P_{los}, N is the number of buses in a power system, and V_{bt} is node voltage (p.u) of the b^{th} bus at the t^{th} time slot (Kang *et al.*, 2012).

Cost is denoted as $\text{Cost} = \sum_{t=1}^{\text{T}} P_t \cdot \delta \cdot \varphi_t$, where P_t is the power consumption of EVs at the t^{th} time slot, φ_t is the electricity price at the t^{th} time slot, and δ is time span of a time slot.

5.2 Simulations and Result

5.2.1 Parameter Scheduling

The IEEE 30-bus system is used which has 6 generator buses 1, 2, 13, 22, 23 and 27. Besides, 20 load buses at buses 2, 3, 4, 7, 8, 10, 12, 14-21, 23, 24, 26, 29, and 30. The buses can be divided into PV buses (at buses 2, 13, 22, 23, and 27, the range of bus voltage from 0.95 to 1.1 p.u.), balance bus (at bus 1, the range of bus voltage from 0.95 to 1.05 p.u.) and PQ buses (at the rest of buses, the range of buses voltage from 0.95 to 1.05 p.u.).

In this thesis, 10 buses, i.e., buses 3–12, are chosen as EV charging nodes. These nodes are the locations where EVs are arranged to be charged and the place of battery swapping stations are located. According to the EV charging rule, the charging priority and locations of EV are coded in a particle. The objective here is to optimize the charging cost, power loss and voltage deviation. The Differential Evolution (DE) based on multi-population strategy is used as a solver of this optimization problem of charging priority. The following equations are used for calculating active power loss, voltage deviation and charging cost:

$$P_{los} = \sum_{t=1}^{\mathrm{T}} \sum_{l=1}^{\mathrm{L}} |I_{lt}|^2 R_l \tag{5.6}$$

$$V_{\mathrm{dev}} = \sum_{t=1}^{\mathrm{T}} \sum_{b=1}^{\mathrm{N}} |V_{bt} - 1.0| (\mathrm{p.u.}) \tag{5.7}$$

$$\mathrm{Cost} = \sum_{t=1}^{\mathrm{T}} P_t \cdot \delta \cdot \varphi_t, \tag{5.8}$$

The active power loss ($P_{los,}$) is expressed in equation (5.6), where T is the number of time slots, L is the number of lines in the power system, I_{lt} is the current of the l^{th} line at the t^{th} time slot, and R_l is the resistance of the l^{th} line. The value range for current I is 0A – 300A and resistance R is 0.025Ω – 0.75Ω. Random values are generated for current and resistance for D sets of EVs and a muli-population strategy-based DE is used as a solver for optimizing all the three functions.

Voltage deviation (V_{dev}) is indicated in equation (5.7), where T plays the same role in P_{los}, N is the number of buses in a power system, and V_{bt} is node voltage (p.u) of the b^{th}

bus at the t^{th} time slot. Voltage is V = I · R. So, V_{dev} is calculated for D set of EVs using the same value range mentioned above.

Equation (5.8) demonstrates the charging cost, where T plays the same role in P_{los}, P_t is the power consumption of EVs at the t^{th} time slot, which ranges from 10kW to 20kW, φ_t is the electricity price at the t^{th} time, which slot spans from \$50/MW - \$75/MW, and δ is time span of a time slot, which varies from 5min – 10min time slot.

5.2.2 Experimental Outcome

All algorithms in Table 4.2 and PSO-GA+ that is a heuristic algorithm proposed by Kang *et al.* (2016), are applied to the problem to minimize active power loss, voltage deviation and charging cost. By running the algorithms for 30 times independently, the statistical results obtained are shown in Tables 5.1-5.4.

Table 5.1 Experimental Result of $P_{los,}$ (MW)

Multi-population strategy	Optimum Value	Mean	Worst Value	Standard Deviation
PSO-GA+	1398.7	1429.1	1513.5	42.68
a_1	1018	**1247**	1552	83
a_2	1041	1415	1834	114
a_3	1709	1890	1945	90
a_4	1465	1504	1531	14
a_5	1241	1670	2711	266
a_6	1279	1352	1635	75

Table 5.2 Experimental Result of V_{dev} (p.u.)

Multi-population strategy	Optimum Value	Mean	Worst Value	Standard Deviation
PSO-GA+	37.92	**37.86**	39.85	1
a_1	40	52	78	4
a_2	37	59	80	3
a_3	38	54	59	3
a_4	34	53	57	3
a_5	40	51	54	3
a_6	36	51	54	3

Table 5.3 Experimental Result of Cost ($)

Multi-population strategy	Optimum Value	Mean	Worst Value	Standard Deviation
PSO-GA+	130592.5	130623.5	130663.9	23.59
a_1	135134	**130545**	135160	5
a_2	135134	135158	135196	10
a_3	138126	139849	144945	822
a_4	136930	137045	137189	42
a_5	136290	136520	136640	79
a_6	137420	137620	137780	65

Table 5.4 Experimental Result of f

Multi-population strategy	Optimum Value	Mean	Worst Value	Standard Deviation
PSO-GA+	133504.7	**133570.0**	133733.9	84.61
a_1	137752	138472	139832	248
a_2	137655	138933	140230	244
a_3	141355	143899	149250	1032
a_4	139755	140669	141000	176
a_5	139131	140230	141511	465
a_6	140139	141012	141575	260

The results show that PSO-GA+ outperforms all the other six algorithms in terms of V_{dev}, and f. The multi-population strategy a_1 surpasses all the other algorithms in terms of P_{los} and Cost, but its optimization capability is worse than PSO-GA+. In this case, we can argue reason that the operations in the algorithm, such as crossover and mutation operation, may play a significant role in solving this problem.

Based on the statistic results, PSO-GA+ is the best in convergence rate and precision for this problem. Furthermore, in case of differential evolution with multi-population strategy, a_1 has better global search ability than the other DE algorithms. In addition, a_1 has lower computational complexity.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis investigates a multi-population DE that divides the initial population into two different subpopulations and executes rand/1 and rand to best/1t on each. The construction and use of battery swapping technology-based facilities for EVs, especially battery swapping stations, offers great opportunities to promote the deployment of EVs. This thesis focuses on a centralized EVs charging strategy with battery swapping under a spot pricing-based market environment. The thesis has made the following contribution:

(1) A literature review is conducted on the DE algorithm and EV charging problem. It is observed that DE is widely implemented due to its stability, robustness, ability for global search and several excellent performances. It is also verified that the construction and use of battery swapping technology-based facilities for EVs, especially battery swapping stations, offers great opportunities to promote the deployment of EVs.

(2) A multi-population strategy based DE algorithm is proposed. It is observed from the previous study that the multi-population strategy helps in maintaining the evolution of the best individual and enhances the local hill-climbing ability in the local search. It contributes to avoiding the local optima and premature convergence in an optimization process.

(3) A comparative study is performed with various numbers of subpopulations and several combinations of mutation strategies. The results demonstrate that the execution time of a multi-population algorithm is inversely proportional to the number of its subpopulations. This multi-population strategy affects its optimization performance either

by improving its solution with increase in the number of subpopulations or by producing the best solution when the population is divided into 2-subpopulations followed by decrease in the optimization accuracy. Hence, the trade-off range for the algorithm to achieve better accuracy and accelerated speed is when the number of subpopulations lies between 2 and 8 inclusive.

(4) The performance of differential evolution algorithm based on a multi-population strategy is studied to optimally determine the EVs charging priority and locations in a distribution network, by minimizing the total charging cost, power loss and voltage deviation for the first time. Its effectiveness has been verified via an IEEE 30-bus test system. The results show that the existing PSO-GA+ algorithm outperforms the proposed algorithms in terms of the weighted optimization objective and voltage deviation while the 2-subpopulation DE performs better in power loss and cost than PSO-GA+ algorithm and the other multi-population DEs.

Finally, it can be summarized that, 2-subpopulation DE, the DE that divides the initial population into two different subpopulations and executes rand/1 and rand to best on each has a higher searching accuracy and faster convergence speed in solving high dimensional optimization problems.

6.2 Future Work

Although during the last two decades, research on and with DE reached an impressive state, there are still some interesting open problems and new application areas that are continually emerging for the algorithm. Recently, population size adaptation has been demonstrated to yield improved performance. Naturally, a larger population is required to perform exploration of the search space at the early stage of the search while a smaller population is necessary to conduct fine search near the best regions at the end of the search process. Further research is needed in the population size adaptation in multiple objective and other optimization scenarios.

Even before the advent of population-based EAs, the concept of convergence was prevalent for single point (only one candidate solution) based search methods. If the single point does not converge, there will be no solution. However, in the context of population based algorithms, the best scenario is that even after one population member discovers the global solution, the other members are well distributed in the search space. Hence, in the context of population-based search methods, the focus of theoretical research can be controlling diversity and convergence behaviours while avoiding chaotic search behaviour.

With the increasing use of uncontrollable distributed generators based on such renewable energy sources as wind and solar power, the complex system constraints caused by the high uncertainty in power outputs should be taken into account when solving the EV charging problems. The future work is required to be oriented to dispatching EVs for charging in a grid with distributed generations using renewable energy by using various intelligent optimization and mathematical programming methods.

More benchmark studies should be created and more evolutionary algorithms should be compared. The sufficiency issues related to how many samples are required for such algorithms should be addressed. Besides, we need to model energy requirements more accurately according to the vehicle travel behaviour. Hence, it is apparent that there are numerous issues to be investigated in the context of differential evolution and electric vehicle charging problems.

APPENDIX A
BENCHMARK OPTIMIZATION FUNCTIONS

The functions listed below are some of the common test functions and datasets used for testing optimization algorithms with the aim of giving an idea about the different situations that optimization algorithms have to face when coping with these kinds of problems.

Ackley Function

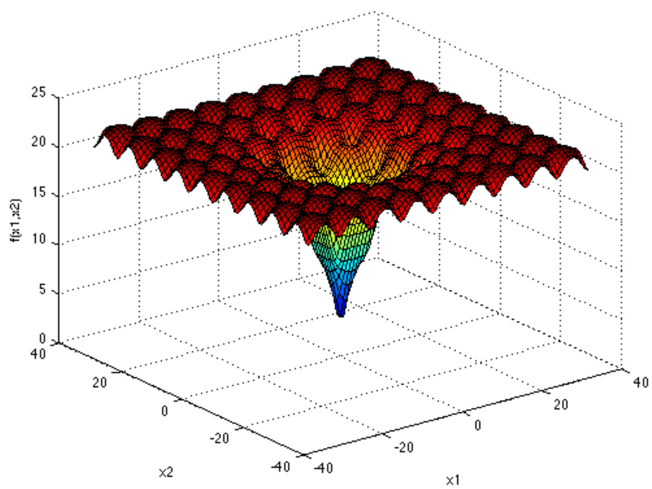

Figure A.1 Ackley function.

Source: "Ackley function," in Virtual Library of Simulation Experiments: Test Functions and Datasets. [Online]. Available: https://www.sfu.ca/~ssurjano/rastr.html. Accessed: Mar. 1, 2017.

$$f(x) = -a\, exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d} cos(cx_i)\right) + a + \exp(1) \qquad \text{(A.1)}$$

Description:

Dimensions:d

The Ackley function is widely used for testing optimization algorithms. In its two-dimensional form, as shown in the plot above, it is characterized by a nearly flat outer region, and a large hole at the centre. The function poses a risk for optimization algorithms, particularly hillclimbing algorithms, to be trapped in one of its many local minima. Recommended variable values are: a = 20, b = 0.2 and c = 2π.

Input Domain:

The function is usually evaluated on the hypercube $x_i \in [-32.768, 32.768]$, for all i = 1,…,d, although it may also be restricted to a smaller domain.

Global Minimum:

$f(x^*) = 0, \text{at } x^* = (0, \ldots, 0)$

Rastrigin Function

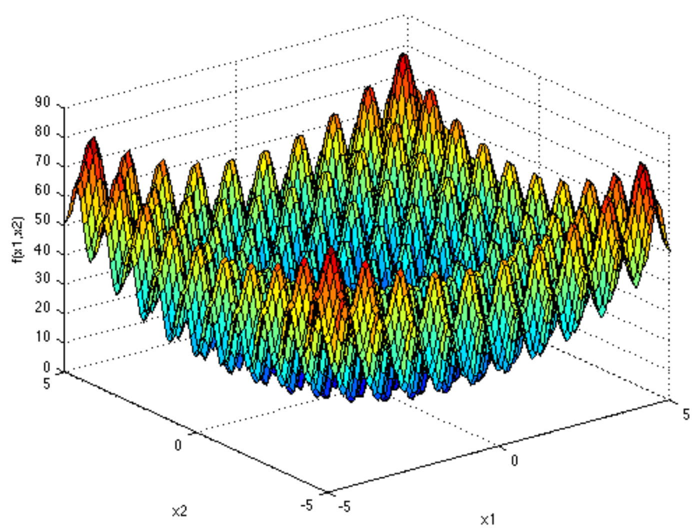

Figure A.2 Rastrigin function.

Source: "Rastrigin function," in Virtual Library of Simulation Experiments: Test Functions and Datasets. [Online]. Available: https://www.sfu.ca/~ssurjano/rastr.html. Accessed: Mar. 1, 2017.

$$f(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)] \quad \text{(A.2)}$$

Description:

Dimensions:d

The Rastrigin function has several local minima. It is highly multimodal, but locations of the minima are regularly distributed. It is shown in the plot above in its two-dimensional form.

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-5.12, 5.12], for all i = 1,…,d.

Global Minimum:

$f(x^*) = 0, \text{at } x^* = (0, \dots, 0)$

Whitley Function

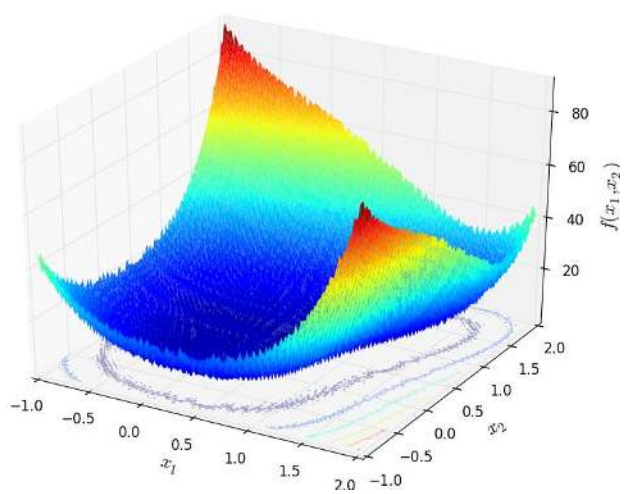

Figure A.3 Whitley function.

Source: "N-D Test Functions W — AMPGO 0.1.0 documentation", Infinity77.net, 2017. [Online]. Available: http://infinity77.net/global_optimization/test_functions_nd_W.html. [Accessed: 23- Mar- 2017].

$$f(x) = \sum_{i=1}^{d}\sum_{j=1}^{d}\left(\frac{\left(100\left(x_i^2 - x_j\right)^2 + \left(1 - x_j\right)^2\right)^2}{4000} - cos\left(100\left(x_i^2 - x_j\right)^2 + \left(1 - x_j\right)^2\right) + 1\right) \tag{A.3}$$

Description:

Dimensions:d

This class defines the Whitley global optimization problem. This is a multimodal minimization problem.

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-10.24, 10.24], for all i = 1,…,d.

Global Minimum:

$f(x^*) = 0, \text{at } x^* = (1, \ldots, 1)$

Schaffer Function N.2

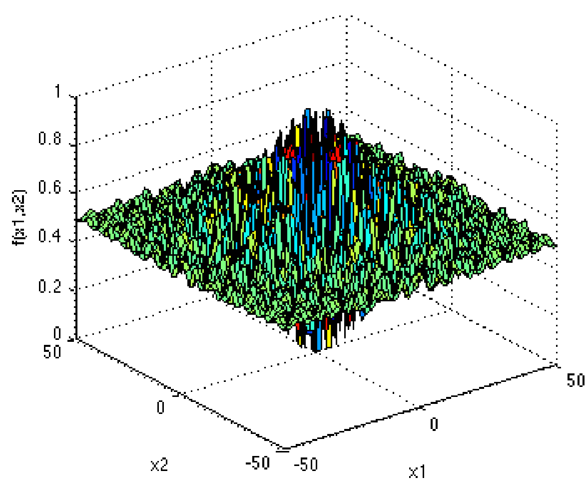

Figure A.4 Schaffer function N.2.

Source: "Schaffer Function N. 2", Sfu.ca, 2017. [Online]. Available: https://www.sfu.ca/~ssurjano/schaffer2.html. [Accessed: 25- Mar- 2017].

$$f(x) = 0.5 + \frac{sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} \quad \text{(A.4)}$$

Description:

Dimensions: 2

Input Domain:

The function is usually evaluated on the square xi ∈ [-100, 100], for all i = 1, 2.

Global Minimum:

$f(x^*) = -1, \text{at } x^* = (0,0)$

Rosenbrock's Function

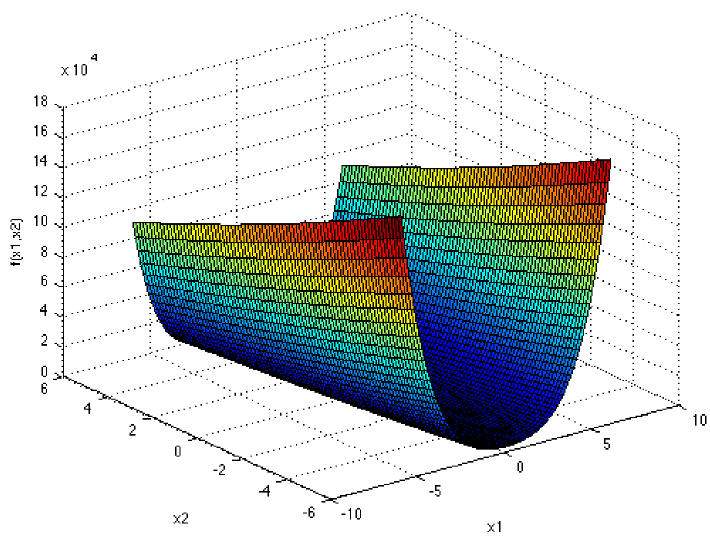

Figure A.5 Rosenbrock's function.

Source: "Rosenbrock Function", Sfu.ca, 2017. [Online]. Available: https://www.sfu.ca/~ssurjano/rosen.html. [Accessed: 25- Mar- 2017].

$$f(x) = \sum_{i=1}^{n} (100(x_i - x_{i-1}^2)^2 + (x_{i-1} - 1)^2) \quad \text{(A.5)}$$

Description:

Dimensions: d

The Rosenbrock function, also referred to as the Valley or Banana function, is a popular test problem for gradient-based optimization algorithms. It is shown in the plot above in its two-dimensional form. The function is unimodal, and the global minimum lies in a narrow, parabolic valley.

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-5, 10], for all i = 1, …, d, although it may be restricted to the hypercube xi ∈ [-2.048, 2.048], for all i = 1, …, d.

Global Minimum:

$f(x^*) = 0, \text{at } x^* = (1, \ldots, 1)$

Modified double sum

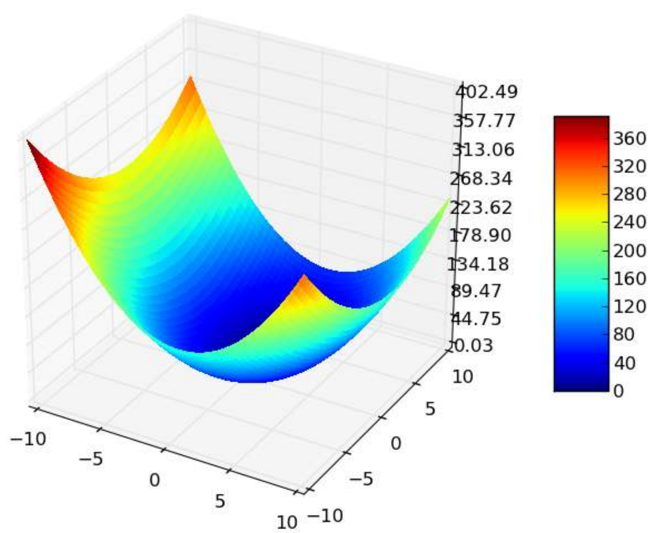

Figure A.6 Modified double sum function.

Source: N. Holtschulte, "Modified Double Sum", Cs.unm.edu, 2017. [Online]. Available: http://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/modDouble.html. [Accessed: 25- Mar- 2017].

$$f(x) = \sum_{i=1}^{d}\left(\sum_{j=1}^{i}(x_j - j)^2\right) \tag{A.6}$$

Description:

Dimensions:d *d*

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-10.24, 10.24], for all i = 1,…,d.

Global Minimum:

$f(x^*) = 0, \text{at } x^* = (0, \ldots, 0)$

Sphere Function

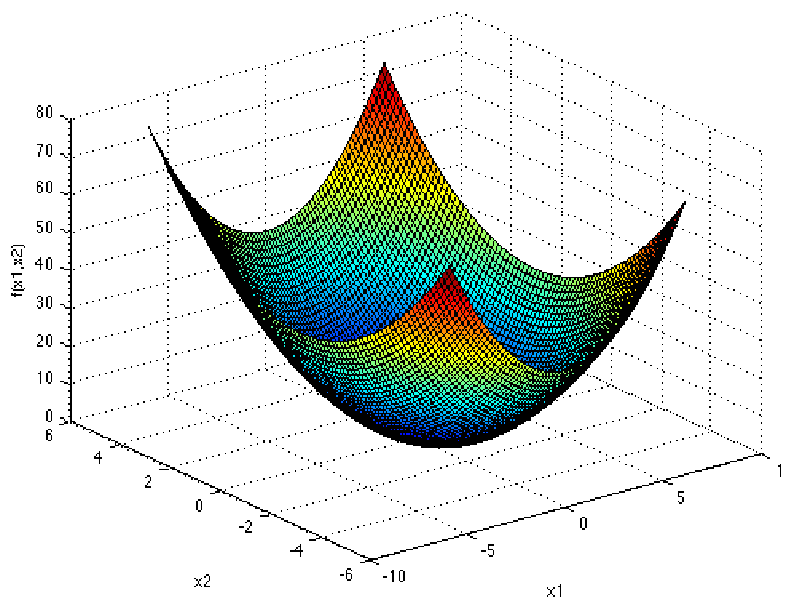

Figure A.7 Sphere function.

Source: "Sphere Function", Sfu.ca, 2017. [Online]. Available: https://www.sfu.ca/~ssurjano/spheref.html. [Accessed: 25- Mar- 2017].

$$f(x) = \sum_{i=1}^{d} x_i^2 \qquad \text{(A.7)}$$

Description:

Dimensions: d

The Sphere function has d local minima except for the global one. It is continuous, convex and unimodal. The plot shows its two-dimensional form.

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-5.12, 5.12], for all i = 1, …, d.

Global Minimum:

$f(x^*) = 0, \text{at } x^* = (0, \ldots, 0)$

Ridge Function

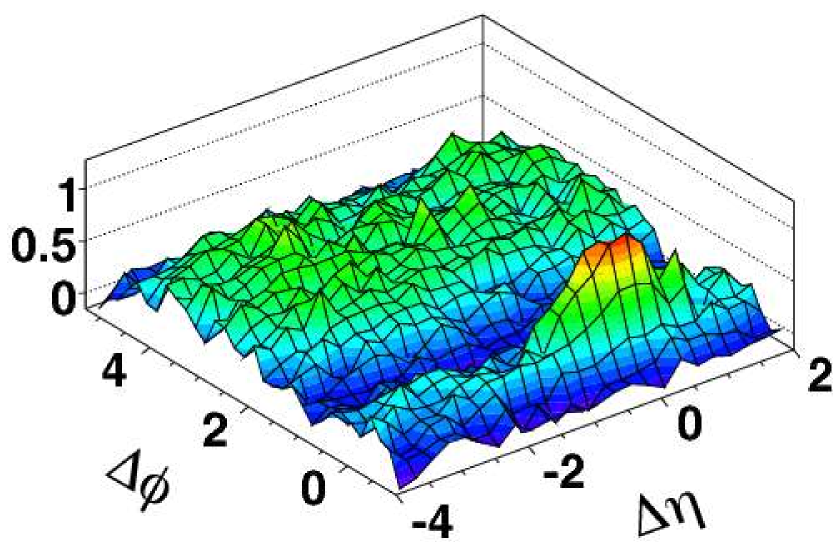

Figure A.8 Ridge function.

Source: W. LI, "OBSERVATION OF A RIDGE CORRELATION STRUCTURE IN HIGH MULTIPLICITY PROTON–PROTON COLLISIONS: A BRIEF REVIEW", 2017. .

$$f(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2 \tag{A.8}$$

Description:

Dimensions: d

Ridge functions and ridge function approximation are studied in statistics. In general, linear combinations of ridge functions with fixed directions occur in the study of hyperbolic partial differential equations with constant coefficients.

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-64, 64], for all i = 1, …, d.

Global Minimum:

$f(x^*) = 0, \text{at } x^* = (0, \dots, 0)$

Schwefel 2.21 Function

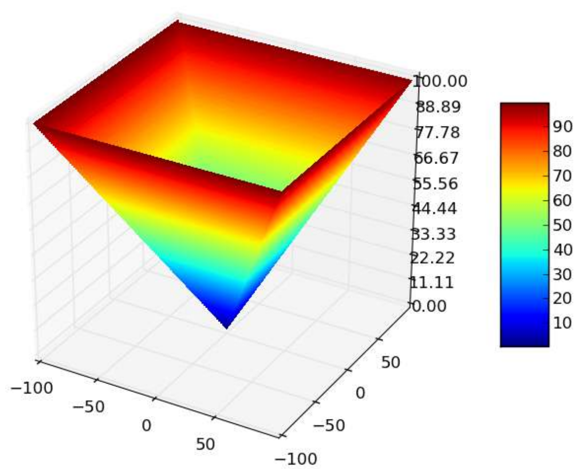

Figure A.9 Ridge function.

Source: W. LI, "OBSERVATION OF A RIDGE CORRELATION STRUCTURE IN HIGH MULTIPLICITY PROTON–PROTON COLLISIONS: A BRIEF REVIEW", 2017. .

$$f(x) = max_i\{|x_i|, 1 \leq i \leq d\} \quad \textbf{(A.9)}$$

Description:

Dimensions: d

The Schwefel 2.21 function is complex, with many local minima. The plot shows the two-dimensional form of the function.

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-500, 500], for all i = 1, …, d.

Global Minimum:

$f(x^*) = 0, at\ x^* = (420.9687, \ldots, 420.9687)$

Lunacek's bi-Rastrigin Function

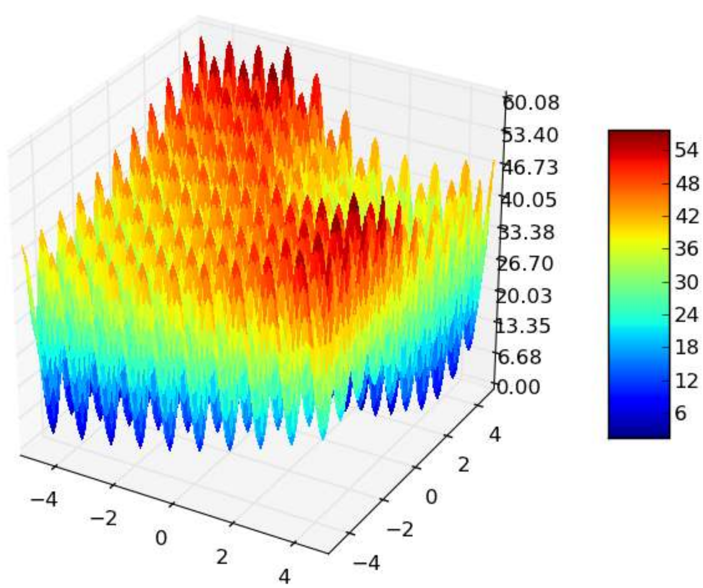

Figure A.10 Lunacek's bi-Rastrigin function.

Source: N. Holtschulte, "Lunacek", Cs.unm.edu, 2017. [Online]. Available: http://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/lunacek.html. [Accessed: 25- Mar- 2017].

$$f(x) = min\left(\left\{\sum_{i}^{n}(x_i - 2.5)^2\right\}, \left\{d.n + s.\sum_{i}^{n}(x_i - \mu_2)^2\right\}\right) + 10\sum_{i}^{n}\left(1 - cos2\pi(x_i - 2.5)\right) \quad \text{(A.10)}$$

$$where, \mu_2 = -\sqrt{\frac{\mu_1^2 - d}{s}}$$

Description:

Dimensions: d

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-5.12, 5.12], for all i = 1, …, d.

Global Minimum:

$f(x^*) = 0, at\ x^* = (0, \dots, 0)$

Levy Function

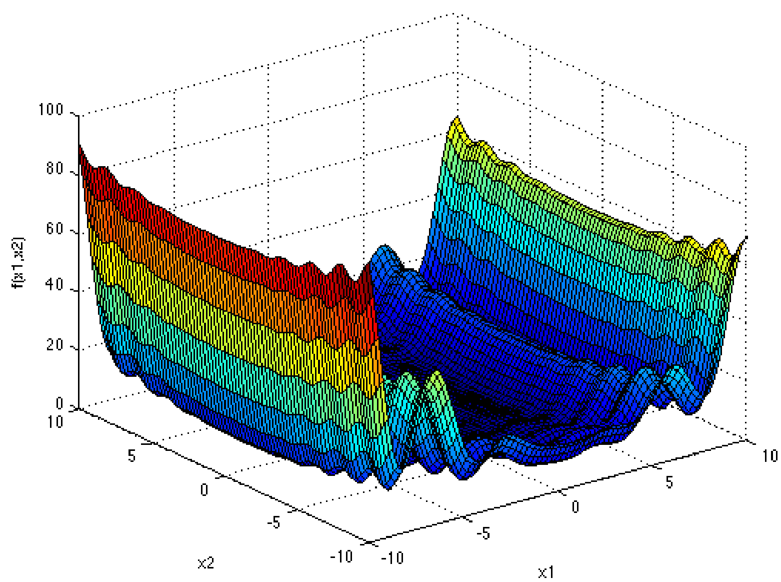

Figure A.11 Levy function.

Source: "Levy Function", Sfu.ca, 2017. [Online]. Available: https://www.sfu.ca/~ssurjano/levy.html. [Accessed: 26- Mar- 2017].

$$f(x) = sin^2(\pi\omega_1) + \sum_{i=1}^{d-1}(\omega_i - 1)^2[1 + 10sin^2(\pi\omega_i + 1)] + (\omega_d - 1)^2[1 + sin^2(2\pi\omega_d)] \quad \textbf{(A.11)}$$

$$where, \omega_i = 1 + \frac{x_i - 1}{4}$$

Description:

Dimensions: d

Input Domain:

The function is usually evaluated on the hypercube xi ∈ [-10, 10], for all i = 1, …, d.

Global Minimum:

$\mathrm{f}(\mathrm{x}^*) = 0, \mathrm{at}\ \mathrm{x}^* = (1, \dots, 1)$

APPENDIX B
EXPERIMENTAL RESULT

Table B1 below shows the experimental results of all algorithms in Table 4.2 along with the multi-population strategy implemented in the former paper (*old*), applied to the functions mentioned in Table 4.1 with their maximum value, minimum value, mean value and standard deviation.

Table B1 Experimental Results

Function	Multi-population strategy	Maximum Value	Mean	Minimum Value	Standard Deviation
f_1	a_1	5.617E-12	**2.953E-09**	1.097E-09	1.394E-10
	old	3.703E-11	1.239E-07	4.294E-07	1.409E-08
	a_2	1.286E-11	1.353E-07	3.581E-07	1.534E-08
	a_3	3.174E-11	8.018E-08	3.297E-07	9.995E-09
	a_4	2.477E-09	2.328E-01	9.313E-01	1.245E-08
	a_5	2.840E-09	3.150E-01	1.475E+00	2.373E-03
	a_6	2.839E-09	6.153E-01	2.532E+00	2.251E-02
f_2	a_1	4.723E+01	**9.654E+01**	1.863E+02	6.493E+00
	old	3.349E+01	1.083E+02	1.835E+02	1.006E+01
	a_2	5.923E+01	1.080E+02	1.942E+02	7.965E+00
	a_3	5.212E+01	1.154E+02	1.904E+02	9.872E+00
	a_4	4.111E+01	1.271E+02	1.874E+02	1.144E+01

	a_5	8.292E+01	1.402E+02	2.192E+02	1.280E+01
	a_6	8.711E+01	1.553E+02	2.393E+02	1.8334E+01
f_3	a_1	2.143E+02	**3.552E+02**	6.143E+02	2.075E+01
	old	4.165E+02	5.455E+02	6.823E+02	2.815E+01
	a_2	3.117E+02	4.434E+02	6.698E+02	2.319E+01
	a_3	3.191E+02	4.696E+02	6.668E+02	2.989E+01
	a_4	3.084E+02	5.050E+02	7.157E+02	3.626E+01
	a_5	3.728E+02	5.469E+02	9.025E+02	3.573E+01
	a_6	3.876E+02	6.537E+02	9.979E+02	3.789E+01
f_4	a_1	1.020E-01	**2.858E-01**	8.233E-01	5.488E-02
	old	2.439E-01	5.240E-01	9.240E-01	8.665E-02
	a_2	1.742E-01	3.479E-01	1.009E+00	6.526E-02
	a_3	5.481E-02	4.095E-01	1.025E+00	9.001E-02
	a_4	9.225E-02	4.076E-01	1.005E+00	8.769E-02
	a_5	1.968E-01	5.121E-01	1.203E+00	1.024E-01
	a_6	2.193E-01	5.350E-01	1.288E+00	1.132E-01
f_5	a_1	2.323E+01	**2.534E-01**	2.739E+01	2.804E-01
	old	2.353E+01	2.544E+01	2.906E+01	5.668E-01
	a_2	1.742E-01	3.479E+01	1.009E+00	6.526E-02
	a3	2.352E+01	2.577E+01	2.729E+01	4.930E-01
	a_4	1.931E+01	2.401E+01	2.692E+01	3.942E-01
	a_5	2.341E+01	2.659E+01	3.121E+01	4.041E-01

	a_6	2.358E+01	4.030E+01	9.383E+01	9.199E-01
f_6	a_1	1.544E+04	1.544E+04	1.544E+04	3.705E-03
	old	1.544E+04	1.545E+04	1.545E+04	3.299E-01
	a_2	3.152E-19	**1.279E-13**	4.541E-13	2.880E-14
	a_3	1.544E+04	1.545E+04	1.546E+04	4.753E-01
	a_4	1.344E+04	1.545E+04	1.546E+04	3.704E-01
	a_5	1.544E+04	1.552E+04	1.589E+04	7.088E+00
	a_6	1.544E+04	1.577E+04	1.731E+04	4.312E+01
f_7	a_1	4.036E-24	**5.705E-20**	1.919E-19	1.351E-20
	old	2.111E-23	4.322E-16	2.105E-15	9.784E-17
	a_2	1.547E-22	1.997E-15	6.266E-15	4.352E-16
	a_3	4.938E-23	1.017E-16	4.971E-16	2.175E-17
	a_4	1.916E-19	1.484E-02	5.936E-02	2.693E-16
	a_5	3.268E-19	1.105E-02	5.504E-02	7.732E-06
	a_6	2.226E-19	3.406E-03	1.533E-02	1.768E-04
f_8	a_1	8.093E+02	**3.341E+03**	1.071E+04	8.153E+02
	old	1.432E+03	4.391E+03	1.898E+04	1.000E+03
	a_2	4.466E+03	1.046E+04	3.226E+04	2.064E+03
	a_3	2.679E+02	4.841E+03	1.651E+04	1.501E+03
	a_4	1.467E+03	6.685E+03	2.309E+04	1.190E+03
	a_5	2.165E+03	9.331E+03	2.839E+04	1.830E+03
	a_6	1.643E+03	9.156E+03	2.729E+04	1.985E+03

f_9	a_1	9.611E-05	6.515E+00	1.303E+01	1.030E-05
	old	1.387E-03	6.347E+00	1.902E+01	8.073E-04
	a_2	1.613E-02	1.477E+01	2.952E+01	8.580E-04
	a_3	1.678E-04	5.943E+00	1.782E+01	3.392E-04
	a_4	2.054E-04	**4.922E+00**	1.967E+01	2.084E-04
	a_5	1.656E-02	1.042E+01	3.638E+01	2.773E-01
	a_6	3.503E-02	1.151E+01	3.177E+01	5.146E-01
f_{10}	a_1	8.799E+01	**1.127E+02**	2.090E+02	6.796E+00
	old	8.871E+01	1.529E+02	2.262E+02	9.964E+00
	a_2	6.444E+01	1.305E+02	1.976E+02	8.024E+00
	a_3	7.897E+01	1.456E+02	2.234E+02	1.012E+01
	a_4	1.200E+01	1.594E+02	2.162E+02	1.126E+01
	a_5	9.150E+01	1.701E+02	2.667E+02	1.280E+01
	a_6	1.011E+02	1.904E+02	3.100E+02	1.312E+01
f_{11}	a_1	1.772E-23	**9.532E-19**	3.113E-18	2.223E-19
	old	8.018E-23	1.010E-03	5.032E-01	1.298E-02
	a_2	7.821E-14	4.999E-02	9.998E-02	1.796E-14
	a_3	4.617E-21	3.391E-14	1.683E-13	7.761E-15
	a_4	1.061E-17	5.779E-01	2.311E+00	6.989E-15
	a_5	3.001E-18	1.577E-01	7.655E-01	9.828E-03
	a_6	8.100E-18	3.077E-01	1.348E+00	2.440E-02

REFERENCES

M. M. A. Abdelaziz, M. F. Shaaban, H. E. Farag, and E. F. El-Saadany, "A Multistage Centralized Control Scheme for Islanded Microgrids With PEVs," IEEE Transactions on Sustainable Energy, vol. 5, no. 3, pp. 927–937, 2014.

M. Alonso, H. Amaris, J. Germain, and J. Galan, "Optimal Charging Scheduling of Electric Vehicles in Smart Grids by Heuristic Algorithms," *Energies*, vol. 7, no. 4, pp. 2449–2475, 2014.

M. Aziz, T. Oda, T. Mitani, Y. Watanabe, and T. Kashiwagi, "Utilization of Electric Vehicles and Their Used Batteries for Peak-Load Shifting," *Energies*, vol. 8, no. 5, pp. 3720–3738, 2015.

N. Baatar, D. Zhang, and C.-S. Koh, "An Improved Differential Evolution Algorithm Adopting λ-Best Mutation Strategy for Global Optimization of Electromagnetic Devices," *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 2097–2100, 2013.

M. Basu, "Economic environmental dispatch using multi-objective differential evolution," *Applied Soft Computing*, vol. 11, no. 2, pp. 2845–2853, 2011.

Z. Cai, W. Gong, C. X. Ling, and H. Zhang, "A clustering-based differential evolution for global optimization," *Applied Soft Computing*, vol. 11, no. 1, pp. 1363–1379, 2011.

Y. Cai and J. Wang, "Differential Evolution With Neighborhood and Direction Information for Numerical Optimization," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2202–2215, 2013.

T. D. Chen, K. M. Kockelman, M. Khan, et al. The electric vehicle charging station location problem: a parking-based assignment method for seattle, *In Proceedings of 92nd Annual Meeting of the Transportation Research Board (TRB)*, 2013.

C.-W. Chiang, W.-P. Lee, and J.-S. Heh, "A 2-Opt based differential evolution for global optimization," *Applied Soft Computin*g, vol. 10, no. 4, pp. 1200–1207, 2010.

S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

S. Das, A. Mandal, and R. Mukherjee, "An Adaptive Differential Evolution Algorithm for Global Optimization in Dynamic Environments," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 966–978, 2014.

S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

E.-N. Dragoi and V. Dafinescu, “Parameter control and hybridization techniques in differential evolution: a survey,” *Artificial Intelligence Review*, vol. 45, no. 4, pp. 447–470, 2015.

E. G. R. Davies and S. P. Simonovic, An integrated system dynamics model for analyzing behaviour of the social-economic-climatic system: model description and model use guide. *London: University of Western Ontario, Dept. of Civil and Environmental Engineering*, 2008.

C. Fonseca and P. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms—I: A unified formulation,” IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 28, no. 1, pp. 26–37, Jan. 1998.

I. Frade, R. Anabela, G. Goncalves, and A. P. Antunes. Optimal location of charging stations for electric vehicles in a neighborhood in lisbon, Portugal, *Transportation Research Record: Journal of the Transportation Research Board*, 2252:91–98, 2011.

R. Gämperle, M. D. Sibylle, and K. Petros, “A parameter study for differential evolution.,” *Advances in intelligent systems, fuzzy systems, evolutionary computation*, vol. 10, pp. 293–298, 2002.

S. Ge, L. Feng, and H. Liu, “The planning of electric vehicle charging station based on Grid partition method,” *2011 International Conference on Electrical and Control Engineering*, 2011.

W. Gong, Z. Cai, C. X. Ling, and H. Li, “Enhanced Differential Evolution With Adaptive Strategies for Numerical Optimization,” *IEEE Transactions on Systems, Man, and Cybernetics*, Part B (Cybernetics), vol. 41, no. 2, pp. 397–413, 2011.

Y.-C. He, X.-Z. Wang, K.-Q. Liu, and Y.-Q. Wang, “Convergent Analysis and Algorithmic Improvement of Differential Evolution,” *Journal of Software*, vol. 21, no. 5, pp. 875–885, 2010.

F. He, D. Wu, Y. Yin, and Y. Guan, Optimal deployment of public charging stations for plug-in hybrid electric vehicles, *Transportation Research Part B: Methodological*, 47:87–101, 2013.

M. Honarmand, A. Zakariazadeh, and S. Jadid, “Integrated scheduling of renewable generation and electric vehicles parking lot in a smart microgrid,” *Energy Conversion and Management*, vol. 86, pp. 745–755, 2014.

Z. Hu, S. Xiong, Q. Su, and Z. Fang, “Finite Markov chain analysis of classical differential evolution algorithm,” *Journal of Computational and Applied Mathematics*, vol. 268, pp. 121–134, 2014.

L. Huang, “An Improved Adaptive Differential Evolution based on Hybrid Method for Function Optimization,” *International Journal of Hybrid Information Technology*, vol. 9, no. 3, pp. 95–104, 2016.

H. Kameda and N. Mukai, “Optimization of Charging Station Placement by Using Taxi Probe Data for On-Demand Electrical Bus System,” *Knowledge-Based and*

Intelligent Information and Engineering Systems Lecture Notes in Computer Science, pp. 606–615, 2011.

Q. Kang, J. Wang, M. Zhou, and A. C. Ammari, “Centralized Charging Strategy and Scheduling Algorithm for Electric Vehicles Under a Battery Swapping Scenario,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 659–669, 2016.

Q. Kang, M. C. Zhou, J. An, and Q. D. Wu, “Swarm intelligence approaches to optimal power flow problem with distributed generator failures in power networks,” IEEE Trans. Autom. Sci. Eng., vol. 10, no. 2, pp. 343–353, Apr. 2013.

Q. Kang, T. Lan, Y. Yan, L. Wang, and Q. D. Wu, “Group search optimizer based optimal location and capacity of distributed generations,” *Neurocomputing*, vol. 78, no. 1, pp. 55–63, Feb. 2012.

K. Knezović and M. Marinelli, “Phase-wise enhanced voltage support from electric vehicles in a Danish low-voltage distribution grid,” *Electric Power Systems Research*, vol. 140, pp. 274–283, 2016.

K. N. Kozlov and A. M. Samsonov, “A new migration scheme for parallel differential evolution”, in *Proc. Fifth International Conference on Bioinformatics of Genome Regulation and Structure (BGRS 06)*, Novosibirsk, Russia, July 16-22, 2006.

T. Lan et al., “Optimal control of an electric vehicle’s charging schedule under electricity markets,” Neural Comput. Appl., vol. 23, no. 7/8, pp. 1865–1872, Dec. 2013.

Y. Li, L. Li, J. Yong, Y. Yao, and Z. Li, “Layout Planning of Electrical Vehicle Charging Stations Based on Genetic Algorithm,” *Lecture Notes in Electrical Engineering Electrical Power Systems and Computers*, pp. 661–668, 2011.

Y.-L. Li and J. Zhang, “A new differential evolution algorithm with dynamic population partition and local restart,” *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*, 2011.

H. Liu, Y. Ji, H. Zhuang, and H. Wu, “Multi-Objective Dynamic Economic Dispatch of Microgrid Systems Including Vehicle-to-Grid,” *Energies*, vol. 8, no. 5, pp. 4476–4495, 2015.

Z. Liu, Q. Wu, A. Nielsen, and Y. Wang, “Day-Ahead Energy Planning with 100% Electric Vehicle Penetration in the Nordic Region by 2050,” *Energies*, vol. 7, no. 3, pp. 1733–1749, 2014.

G. Liu, Y. Li, X. Nie, and H. Zheng, “A novel clustering-based differential evolution with 2 multi-parent crossovers for global optimization,” *Applied Soft Computing*, vol. 12, no. 2, pp. 663–681, 2012.

F. Locment and M. Sechilariu, “Modeling and Simulation of DC Microgrids for Electric Vehicle Charging Stations,” *Energies*, vol. 8, no. 5, pp. 4335–4356, 2015.

R. Mallipeddi, P. Suganthan, Q. Pan, and M. Tasgetiren, “Differential evolution algorithm with ensemble of parameters and mutation strategies,” *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

A. Masoum, S. Deilami, and P. Moses, "Smart load management of plug-in electric vehicles in distribution and residential networks with charging stations for peak shaving and loss minimization considering voltage regulation*," IET Gen., Transmiss. Distrib*., vol. 5, no. 8, pp. 877–888, Aug. 2011.

V. C. A. D. V. D. Melo and A. C. B. Delbem, "Investigating Smart Sampling as a population initialization method for Differential Evolution in continuous problems," *Information Sciences*, vol. 193, pp. 36–53, 2012.

A. W. Mohamed and H. Z. Sabry, "Constrained optimization based on modified differential evolution algorithm," *Information Sciences*, vol. 194, pp. 171–208, 2012.

K. Morrow, D. Darner, and J. Francfort, "U.S. Department of Energy Vehicle Technologies Program -- Advanced Vehicle Testing Activity -- Plug-in Hybrid Electric Vehicle Charging Infrastructure Review," Jan. 2008.

R. Mukherjee, G. R. Patra, R. Kundu, and S. Das, "Cluster-based differential evolution with Crowding Archive for niching in dynamic environments," *Information Sciences*, vol. 267, pp. 58–82, 2014.

F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2009.

I. Poikolainen, F. Neri, and F. Caraffini, "Cluster-Based Population Initialization for differential evolution frameworks," *Information Sciences*, vol. 297, pp. 216–235, 2015.

A. Qin and P. Suganthan, "Self-adaptive Differential Evolution Algorithm for Numerical Optimization," *2005 IEEE Congress on Evolutionary Computation,* 2015.

P. Rakshit, A. Konar, S. Das, L. C. Jain, and A. K. Nagar, "Uncertainty Management in Differential Evolution Induced Multiobjective Optimization in Presence of Measurement Noise," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 7, pp. 922–937, 2014.

P. Richardson, D. Flynn, and A. Keane, "Local versus centralized charging strategies for electric vehicles in low voltage distribution systems," *2013 IEEE Power & Energy Society General Meeting*, 2013.

T. Robič and B. Filipič, "DEMO: Differential Evolution for Multiobjective Optimization," *Lecture Notes in Computer Science Evolutionary Multi-Criterion Optimization*, pp. 520–533, 2005.

H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, Jan. 1960.

S. Shao, M. Pipattanasomporn, and S. Rahman, "Demand response as a load shaping tool in an intelligent grid with electric vehicles," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 624–631, Nov. 2011.

C. Song and Y. Hou, "An Improved Differential Evolution Algorithm for Solving High Dimensional Optimization Problem," *International Journal of Hybrid Information Technology*, vol. 8, no. 10, pp. 177–186, 2015.

W. M. Spears, "Adapting crossover in evolutionary algorithms," *The 4th Annual Conference on Evolutionary Programming*, pp. 367–384, 1995.

R. Storn and K. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces". *Berkeley, CA: ICSI*, 1995.

R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

T. Sweda and D. Klabjan, "An agent-based decision support system for electric vehicle charging infrastructure deployment," *2011 IEEE Vehicle Power and Propulsion Conference*, 2011.

L. Tang, Y. Zhao, and J. Liu, "An Improved Differential Evolution Algorithm for Practical Dynamic Scheduling in Steelmaking-Continuous Casting Production," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 209–225, 2014.

D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Parallel differential evolution," in *Proc. Congr. Evol. Comput.*, 2004.

J. Wang, J. Liao, Y. Zhou, and Y. Cai, "Differential Evolution Enhanced With Multiobjective Sorting-Based Mutation Operators," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2792–2805, 2014.

H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian Bare-Bones Differential Evolution," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.

Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 55–66, 2011.

H. Wang, Z. Wu, and S. Rahnamayan, "Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems," *Soft Computing*, vol. 15, no. 11, pp. 2127–2140, 2010.

H. Wang, Q. Huang, C. Zhang, and A. Xia. "A novel approach for the layout of electric vehicle charging station", *In Proceedings of International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA),* pages 64–70, 2010.

O. Worley, D. Klabjan, and T. M. Sweda, "Simultaneous vehicle routing and charging station siting for commercial Electric Vehicles," *2012 IEEE International Electric Vehicle Conference*, 2012.

G. Wu, R. Mallipeddi, P. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329–345, 2016.

G. Wu, W. Pedrycz, P. Suganthan, and R. Mallipeddi, "A variable reduction strategy for evolutionary algorithms handling equality constraints," *Applied Soft Computing*, vol. 37, pp. 774–786, 2015.

M. Yang, C. Li, Z. Cai, and J. Guan, "Differential Evolution With Auto-Enhanced Population Diversity," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 302–315, 2015.

W. Yu, M. Shen, W. Chen, Z. Zhan, Y. Gong, Y. Lin, O. Liu, and J. Zhang, "Differential evolution with two-level parameter adaptation," *IEEE Transaction on Cybernetics*, vol. 44(7), Pp: 1080-1099, 2014.

W.-J. Yu and J. Zhang, "Multi-population differential evolution with adaptive parameter control for global optimization," *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*, 2011.

A. Zakariazadeh, S. Jadid, and P. Siano, "Multi-objective scheduling of electric vehicles in smart distribution system," *Energy Conversion and Management*, vol. 79, pp. 43–53, 2014.

L. Zhang, T. Brown, and G. S. Samuelsen, "Fuel reduction and electricity consumption impact of different charging scenarios for plug-in hybrid electric vehicles," *Journal of Power Sources*, vol. 196, no. 15, pp. 6559–6566, 2011.

Q. Zhang, B. C. Mclellan, T. Tezuka, and K. N. Ishihara, "A methodology for economic and environmental analysis of electric vehicles with different operational conditions," *Energy*, vol. 61, pp. 118–127, Nov. 2013.

Y. Zhao, J. Wang, and Y. Song, "An improved differential evolution to continuous domains and its convergence," *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation - GEC '09*, 2009.

J. Zhao, F. Wen, and A. Yang, "Impacts of electric vehicles on power systems as well as the associated dispatching and control problem," *Autom. Elect. Power Syst.* vol. 35, no. 14, pp. 2–10, 2011.

J.-H. Zhong and J. Zhang, "Adaptive multi-objective differential evolution with stochastic coding strategy," *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*, 2011.

J.-H. Zhong, M. Shen, J. Zhang, H. S.-H. Chung, Y.-H. Shi, and Y. Li, "A Differential Evolution Algorithm With Dual Populations for Solving Periodic Railway Timetable Scheduling Problem," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 512–527, 2013.

W. Zou, B. Wu, and Z. Liu, "Centralized charging strategies of plug-in hybrid electric vehicles under electricity markets based on spot pricing," *Autom. Elect. Power Syst.*, vol. 35, no. 14, pp. 62–67, 2011.