

## The Redundancy Queuing-Location-Allocation Problem: A Novel Approach

— [Source link](#) 

Vahid Hajipour, Vahid Khodakarami, Madjid Tavana

**Institutions:** Bu-Ali Sina University, La Salle University

**Published on:** 08 Jul 2014 - IEEE Transactions on Engineering Management (IEEE)

**Topics:** Redundancy (engineering), Simulated annealing, Nondeterministic algorithm, Location-allocation and Time complexity

Related papers:

- [A fast and elitist multiobjective genetic algorithm: NSGA-II](#)
- [A multi-objective harmony search algorithm to optimize multi-server location–allocation problem in congested systems](#)
- [Evolutionary algorithms for solving multi-objective problems](#)
- [Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study](#)
- [A multi-objective facility location model with batch arrivals: two parameter-tuned meta-heuristic algorithms](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/the-redundancy-queuing-location-allocation-problem-a-novel-3isf4gd2a0>

# The Redundancy Queuing-Location-Allocation Problem: A Novel Approach

Vahid Hajipour, Vahid Khodakarami, and Madjid Tavana

**Abstract**—Redundancy queuing-location-allocation problems (RQLAPs) involve the economical allocation of facilities, each with a number of servers, to suitable locations with appropriate levels of redundancy or reliability. The goal in RQLAPs is to find the facilities which are both inexpensive and reliable. We take into consideration the congestion of the system by modeling each facility as an  $M/M/m$  queuing system and formulate the problem as a multiobjective nonlinear mixed integer programming problem. However, these problems are nondeterministic polynomial-time hard (NP-hard) problems where an exact solution cannot be produced in polynomial time. We propose a novel soft-computing approach based on the vibration theory called vibration damping optimization (VDO) to solve the RQLAP. We develop a multiobjective version of the VDO called multiobjective VDO (MOVDO) based on the fast nondominated sorting and crowding distance concepts in the literature. The performance of the proposed MOVDO is statistically compared with two other commonly used metaheuristic algorithms called the nondominated sorting genetic algorithm and multiobjective simulated annealing. A comparison of the results based on different problem sizes favors the MOVDO method proposed in this study.

**Index Terms**—Computational intelligence, congestion, redundancy queuing-location-allocation problem (RQLAP), reliability, vibration damping optimization (VDO).

## I. INTRODUCTION

THE rqlaps involve the economical allocation of facilities, each with a number of servers, to suitable locations with appropriate levels of redundancy or reliability. The goal in RQLAPs is to find the facilities which are both inexpensive and reliable. We take into consideration the congestion of the system by modeling each facility as an  $M/M/m$  queuing system and formulate the problem as a multiobjective nonlinear mixed integer programming problem. The objectives of the RQLAPs are to: 1) maximize the system reliability, 2) minimize the system cost, and 3) minimize the system waiting time, simultaneously.

Manuscript received January 7, 2014; revised March 17, 2014; accepted April 8, 2014. Date of publication July 7, 2014; date of current version July 15, 2014. Review of this manuscript was arranged by Department Editor B. Jiang.

V. Hajipour and V. Khodakarami are with the Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran (e-mail: v.hajipour@yahoo.com; v\_khodakarami@yahoo.com).

M. Tavana is with the Business Systems and Analytics Department, Lindback Distinguished Chair of Information Systems and Decision Sciences, La Salle University, Philadelphia, PA 19141 USA, and also with the Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn D-33098, Germany (e-mail: tavana@lasalle.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEM.2014.2317760

Several evolutionary algorithms have been proposed to solve these multiobjective nonlinear mixed integer programming models which combine rules and randomness mimicking natural phenomena. These phenomena include biological evolutionary processes [18], the genetic algorithm (GA) [25], [30], animal behavior [24], [46], [39], the physical annealing process [31], and the musical process of searching for a perfect state of harmony [21]. Many researchers have studied these metaheuristics to solve various optimization problems.

The RQLAPs are nondeterministic polynomial-time hard (NP-hard) [43], [51], [53]. In this study, we propose a novel soft-computing approach based on vibration theory called vibration damping optimization (VDO) to solve the NP-hard RQLAPs. We develop a multiobjective version of the VDO model called multiobjective VDO (MOVDO) based on fast nondominated sorting (FNDS) and crowding distance (CD) concepts in the literature. The performance of the MOVDO approach proposed in this study is statistically compared with two other common metaheuristic algorithms called the nondominated sorting genetic algorithm (NSGA-II) and multiobjective simulated annealing (MOSA).

### A. Location-Allocation Problem (LAP)

In the LAP, the question is how to place a number of new facilities in between a number of customers located at fixed points in a feasible area such that the total transportation cost from facilities to customers is minimized, while the customers' demands are satisfied. An example of a LAP involves the design of a distribution network where, in addition to the determination of the locations of the warehouses, determining the allocation of customers to warehouses is desired. Meanwhile, in a stochastic LAP, many parameters, such as the customers' demands, allocations, and even locations of the customers and facilities are stochastic rather than deterministic.

Although LAP is rather an old field of research, its application is still applicable to real-world location planning problems [12], [17] introduced eight basic discrete location allocation models called (1) p-median, (2) p-center, (3) p-dispersion, (4) set covering, (5) maximal covering, (6) fixed charge, (7) hub, and (8) maximum. Ohsawa [40] provided a single facility, quadratic euclidean distance bicriteria model defined in the continuous space, with convex combinations of the minimum and minimax objectives. Other researchers have proposed various methods for solving LAPs in the literature (see, for example, [19] and [33]). For more details about facility location problem as well as various solving methodologies, interested readers should refer to [6] and [15]–[17].

Farhani *et al.* [17] showed that modeling and solving multi-criteria location problems have had a substantial growth in the past decade. Recently, a more realistic hybrid version of the LAP integrated with other models such as supply chain management and queuing theory has received considerable attention [5], [37]. As a result, in this study, we concentrate on a deterministic LAP combined with queuing theory, called the queuing LAP (QLAP).

### B. Queuing-Location-Allocation Problem

Queuing theory includes some analytical techniques related to everyday waiting lines. Reduction of the wasted time because of these waiting lines is the desire of any manufacturer, service provider, system owner, and the like [11], [26]. This desire can be achieved by appropriately allocating the necessary resources to the customers. Some applications of the QLAP model include healthcare and emergency services, public safety, and fire-fighting among others [48]. For a two-server network, the problem of districting using queuing techniques was investigated by Berman and Larson [2]. In order to locate two servers in a congested network, they developed a nonlinear model and a heuristic algorithm to determine the optimal service territories of each server, where each server behaves as an  $M/G/1$  system. Berman and Drezner [4] proposed a facility location model within an  $M/M/m$  queuing framework, in which one or more servers were allowed to be located at any potential location.

Wang *et al.* [53] proposed a facility location model within an open finite  $M/M/1$  queuing network, in which at most one server can be located at any potential location. They considered the sum of the aggregate travel time plus the aggregate waiting time of customers per unit time as the objective functions in their mathematical model. Berman *et al.* [3] proposed a model similar to [53] model, in which more constraints on lost demand were considered and minimizing the number of facilities was the primary objective function. Syam [49] developed a multiple-server location-allocation model for service system design. He solved a comprehensive nonlinear location-allocation model for service system design that incorporated several relevant costs. Aboolian *et al.* [1] proposed a multiple-server center location model, in which each customer selects the nearest facility. Their objective was to minimize the maximum time spent by any customer including travel time and waiting time at the server sites.

Pasandideh and Niaki [41] proposed a biobjective facility location model within an  $M/M/1$  queuing framework and used the desirability function technique to solve the problem with a genetic algorithm. Hajipour and Pasandideh [27] proposed an adaptive version of a multiobjective particle swarm optimization to optimize biobjective facility location models for congested systems. Recently, Pasandideh *et al.* [43] developed a novel multiobjective facility location model with batch arrivals. To solve the model, they proposed two parameter-tuned metaheuristic algorithms named simulated annealing and the genetic algorithm. Following this, Rahmati *et al.* [47] developed a multiserver facility location model and proposed three Pareto-based metaheuristic algorithms to solve the problem at hand.

### C. Redundancy Allocation Problem (RAP)

Fyffe *et al.* [20] introduced the RAP with weight and cost constraints and solved it by using dynamic programming. Coit and Liu [10] presented system designs which consisted of multiple subsystems by considering a predetermined redundancy strategy for each subsystem. They used a genetic algorithm to solve the optimization problem. In another study, Coit [9] proposed a zero-one integer programming method in which the selection of active or cold-standby redundancy could be selectively chosen for individual subsystems. The proposed method was specifically developed to accommodate the case where there was a choice of a redundancy strategy. Snyder and Daskin [51] proposed models for choosing facility locations to minimize cost, while also taking into account the expected transportation cost after failures of facilities. The goal was to choose facility locations that were both inexpensive under traditional objective functions and also reliable. They presented an optimal Lagrangian relaxation algorithm to solve these reliability problems. For more details about RAP, interested readers can refer to [7] and [32].

The remainder of the paper is organized as follows: In the next section, the problem is first described and then the proposed mathematical model is stated. In Section III, three metaheuristic algorithms including MOVDO, NSGA-II, and MOSA are proposed to solve the model. Section IV analyzes the results and statistical comparisons of the proposed algorithms on several problems of different sizes. The conclusions are made and possible future research works are recommended in Section V.

## II. REDUNDANCY QUEUING-LOCATION-ALLOCATION PROBLEM

In this section, the problem of the redundancy queuing-location-allocation is first introduced. Then, the parameters and decisions variables of the model are introduced in Sections 2.2 and 2.3. Finally, the proposed mathematical formulation is presented in Section 2.4.

### A. RQLAP

The problem of redundancy queuing-location-allocation is introduced to determine:

- 1) the optimal number of facilities;
- 2) the optimal allocation process for demand nodes into the facilities; and
- 3) the optimal number of servers at each facility.

The goal is to find the facilities which are both inexpensive and also reliable. In the problem, we model the congestion of the facilities by means of multiserver queuing systems. The proposed RQLAP is a multiobjective mathematical model designed to simultaneously optimize the goal of the LAP, QLAP, and RAP. A multiobjective optimization problem can be defined as a vector of decision variables. This vector satisfies constraints and optimizes a vector function of the objective functions, forms a mathematical description of the performance criteria that are usually in conflict with each other, and considers the concept of the optimization word, which means acceptable values of

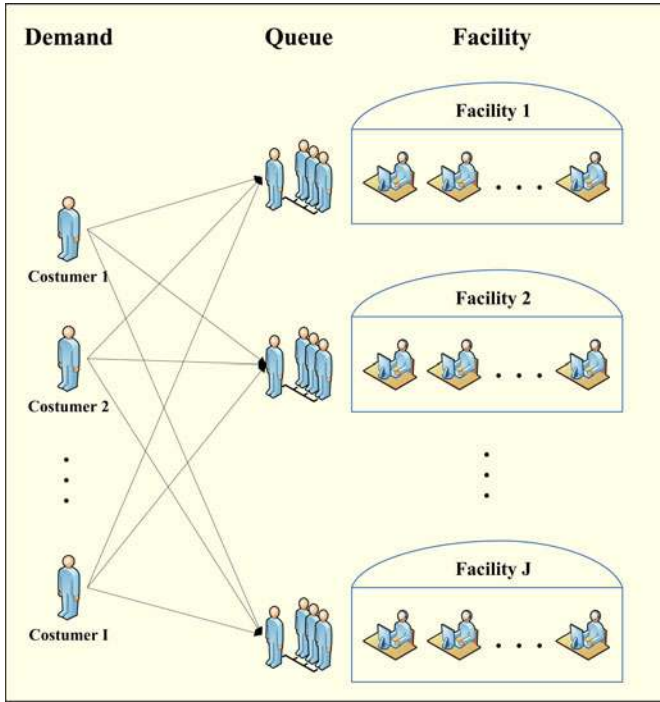


Fig. 1. RQLAP framework.

all the objective functions from the viewpoint of the decision makers [8]. The three objectives considered here are: maximizing reliability, minimizing cost of system, and waiting time. The applications of such a model consist of medical facilities, post offices, automated teller machines, vending machines, and intercity service centers [1].

In this model, the customers' arrivals are assumed to independently follow a Poisson process and service times are considered to follow an exponential distribution. Fig. 1 schematically presents the RQLAP.

It is also assumed the servers in a given facility to have a unique service rate, whereas the service rates of the servers belonging to different facilities are assumed to be different. Moreover, demand nodes travel to each facility to receive service, where each demand node can only be assigned to a single facility.

### B. Parameters

- $I$ : Number of customer nodes indexed by  $i$ ;  $i = 1, 2, \dots, I$ .
- $J$ : Number of service-site nodes indexed by  $j$ ;  $j = 1, 2, \dots, J$ .
- $P$ : Maximum number of servers that are on-duty; ( $P \leq J$ ).
- $\lambda_i$ : Demand rate of service requests from customer node  $i$ .
- $\mu_j$ : Service rate of server  $j$ .
- $v_j$ : Scale parameter for gamma distribution in server-site  $j$ .
- $k_j$ : Shape parameter for gamma distribution in server-site  $j$ .
- $w_j$ : Expected waiting time at open facility  $j$ .
- $r_j$ : Reliability of each server in server-site  $j$ .
- $R_j$ : Total reliability of server-site  $j$ .

- $\psi_j$ : Demand rate at open facility node  $j$ .
- $t_{ij}$ : Traveling time of demand node  $i$  into facility  $j$ .
- $c_j$ : Fixed cost of establishing a facility at potential node  $j$ .
- $cs_j$ : Unit staffing cost at facility  $j$ .
- $u$ : Maximum number of servers that can be used at each facility.
- $\xi$ : A large number.
- $\pi_{0j}$ : Idle probability of the open facility  $j$ .

### C. Decision Variables

$$y_{ij} = \begin{cases} 1, & \text{if customer } i \text{ is allocated to server-site } j \\ 0, & \text{otherwise} \end{cases}$$

$$h_j = \begin{cases} 1, & \text{if server-site } j \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

$$m_j: \text{ Number of servers at server-site } j$$

### D. RQLAP Model

The parameters  $\psi_j$  and  $\pi_{0j}$  denote the demand rate and the idle probability at open facility  $j$ , respectively. In an  $M/M/m$  queuing system, let  $w_j$  be the expected waiting time at open facility  $j$ . These parameters can be obtained as follows [26]:

$$\psi_j = \sum_{i=1}^I \lambda_i y_{ij}; \quad j = 1, \dots, J \quad (1)$$

$$w_j = \left[ \frac{\pi_{0,j}}{m_j!} \left( \frac{\psi_j}{\mu_j} \right)^{m_j} \frac{m_j \mu_j}{(m_j \mu_j - \psi_j)^2} + \frac{1}{\mu_j} \right]. \quad (2)$$

Then, the first objective function of the model ( $f_1$ ), the aggregate waiting times of the customers per time unit (to be minimized), is obtained as follows:

$$f_1 = \sum_{i=1}^I \sum_{j=1}^J \lambda_i y_{ij} \left[ \frac{\pi_{0,j}}{m_j!} \left( \frac{\psi_j}{\mu_j} \right)^{m_j} \frac{m_j \mu_j}{(m_j \mu_j - \psi_j)^2} + \frac{1}{\mu_j} \right]. \quad (3)$$

The second objective function considers the goal of the customers in which the minimum probability of a facility to fail (system reliability) is maximized. At each facility, we assume the servers to be parallel structures. Using this fact and the gamma distributions, it is possible to compute the reliability of each server and the total reliability of the facilities [see (5) and (6) below]. In addition, at each facility, scale and shape parameters are recorded. Larger values of the parameters of the gamma distribution increase the reliability of the server site. Thus, the second objective function ( $f_2$ ) that needs to be maximized is:

$$\text{Max } f_2 = \text{Min } \{R_j(t)\} \quad (4)$$

where

$$r_j(t) = e^{(-v_j \cdot t)} \cdot \sum_{l=0}^{k_j-1} \frac{(v_j \cdot t)^l}{l!}; \quad j = 1, \dots, J \quad (5)$$

$$R_j(t) = (1 - (1 - r_j(t))^{m_j}); \quad j = 1, \dots, J. \quad (6)$$

The third objective function, presented in (7), considers the total cost of the system including the fixed costs associated with establishing facilities as well as the staffing costs that needs to



TABLE I  
SIMILARITIES BETWEEN THE VIBRATIONS DAMPING PROCESS AND THE OPTIMIZATION PROCESS

Vibration damping process		Optimization process
System states	↔	Feasible solution
Energy	↔	Cost
Change of state	↔	Neighboring solution
Amplitude	↔	Control parameter
Vibration damping	↔	Heuristic solution

be minimized:

$$\text{Min} f_3 = \sum_{j=1}^J c_j h_j + cs_j m_j. \quad (7)$$

To make the model more realistic, we also consider the constraints such as service capacity and nearest facility selection. As a result, the proposed RQLAP model can be formulated into a constrained nonlinear mixed integer programming model which is presented in the next section:

$$\text{Min} f_1 = \sum_{i=1}^I \sum_{j=1}^J \lambda_i y_{ij} \left[ \frac{\pi_{0,j}}{m_j!} \left( \frac{\psi_j}{\mu_j} \right)^{m_j} \frac{m_j \mu_j}{(m_j \mu_j - \psi_j)^2} + \frac{1}{\mu_j} \right] \quad (8)$$

$$\text{Max} f_2 = \text{Min} \{1 - (1 - r_j(t))^{m_j}\} \quad (9)$$

$$\text{Min} f_3 = \sum_{j=1}^J c_j h_j + cs_j m_j \quad (10)$$

Subject to:

$$\sum_{j=1}^J h_j \leq P \quad (11)$$

$$\sum_{j=1}^J y_{ij} = 1; \quad i = 1, \dots, I \quad (12)$$

$$y_{ij} \leq h_j; \quad i = 1, \dots, I, \quad j = 1, \dots, J \quad (13)$$

$$\sum_{i=1}^I \lambda_i y_{ij} \leq m_j \mu_j; \quad j = 1, \dots, J \quad (14)$$

$$1 \leq m_j \leq u \quad (15)$$

$$\sum_{j=1}^J t_{ij} y_{ij} \leq (t_{ij} - \xi) h_j + \xi; \quad i = 1, \dots, I, j = 1, \dots, J \quad (16)$$

$$h_j \in \{0, 1\}; \quad j = 1, \dots, J$$

$$y_{ij} \in \{0, 1\}; \quad i = 1, \dots, I, \quad j = 1, \dots, J \quad (17)$$

$$m_j \text{ integer}; \quad j = 1, \dots, J$$

$$\psi_j = \sum_{i=1}^I \lambda_i y_{ij}; \quad j = 1, \dots, J.$$

Constraint (11) ensures that the maximum number of candidate facilities is on-duty. Constraints (12) ensure that each

customer must be assigned to one and only one facility. Constraints (13) requires the server-site  $j$  to be open for the  $i$ th customer to be assigned to it. Constraints (14) ensure that the service capacity is greater than the demand rate of each facility. Constraints (15) bound the number of servers at each facility. Constraints (16) ensure that the demand node select the nearest facility. Constraints (17) impose the range of decision variables in the model.

### III. NOVEL METAHEURISTIC ALGORITHM

In this section, we first introduce a multiobjective version of the VDO algorithm that we will use to solve the proposed multiobjective optimization problems. We will then employ two well-developed Pareto-based MOEAs, namely, NSGA-II and MOSA, to demonstrate the performance of the MOVDO approach.

The VDO algorithm was first introduced by Mehdizadeh and Tavakkoli-Moghaddam [35] to solve parallel machine scheduling problems. It is a stochastic search method based on the concept of the vibration damping in mechanical vibration. VDO is a neighborhood search technique that begins with an initial solution, initial amplitude, and an iteration number. The amplitude controls the possibility of the acceptance of a deteriorating solution, and the iteration number determines the number of repetitions until a solution reaches a stable state under the amplitude. The vibrations damping process and the optimization process are similar in the sense that: 1) the states of an oscillation system represents feasible solutions of the optimization problem; 2) the energies of the states correspond to the objective function value (OFV) computed at those solutions; and 3) the minimum energy state corresponds to the optimal solution to the problem and rapid quenching can be viewed as local optimization. Table I highlights the similarities between the vibrations damping process and the optimization process.

In vibration theory, the concept of vibration is identified with the one of oscillation. If the damping is small, it has very little influence on the natural frequencies of the system, and hence the calculation of the natural frequencies is made primarily on the basis of no damping. In the VDO algorithm, a scope of a solution is bigger and the probability of obtaining of a new solution is greater at high amplitudes. In other words, in the VDO algorithm, the damping coefficient has influence on the maximum amplitude. At high amplitudes, the scope of a solution is larger and the probability of obtaining a new solution is higher. When the amplitude is reduced, the probability of obtaining a new solution decreases. This process is similar to the simulated annealing algorithm where instead of amplitude,

we deal with temperature. Therefore, when the amplitude is reduced, the solution space and consequently the probability of obtaining a new solution decreases. Finally, the algorithm stops when the amplitude reaches zero [35], [36], [38].

In the analogy between an optimization problem and the vibration damping process, the states of the oscillation system represent feasible solutions of the optimization problem, the energies of the states correspond to the OFV computed at those solutions, the minimum energy state corresponds to the optimal solution to the problem, and rapid quenching can be viewed as local optimization.

The VDO algorithm starts by generating random solutions in the search space. Then, the algorithm parameters including initial amplitude ( $A_0$ ), max of iteration at each amplitude ( $L$ ), damping coefficient ( $\gamma$ ), and standard deviation ( $\sigma$ ) are initialized. The solutions are then evaluated by means of the OFV. The algorithm consists of two main loops. The first loop generates a solution randomly and then using a neighborhood structure a new solution is obtained and chosen as the best solution. The neighborhood structure is based on the swap strategy [28]. However, similarly to the SA algorithm, the solution with a lower OFV can be selected with respect to the Rayleigh distribution function [38]. In fact, the new solution is accepted if  $\Delta = \text{OFV}(\text{New Solution}) - \text{OFV}(\text{Current Solution}) < 0$ . Besides, if  $\Delta > 0$ , we generate a random number  $r$  between (0, 1) [31]. The current solution is selected with respect to the following criterion:

$$r < 1 - \exp\left(-\frac{A^2}{2\sigma^2}\right). \quad (18)$$

The second loop reduces the amplitude in each iteration. The algorithm is stopped when the amplitude ( $A$ ) reaches zero:

$$A_t = A_0 \exp\left(-\frac{\gamma t}{2}\right). \quad (19)$$

After the above brief illustration of the VDO algorithm, we introduce the first procedure for applying a multiobjective version of the VDO algorithm called MOVDO to solve and manage Pareto-optimal solutions. To do so, we apply two main concepts of multiobjective metaheuristics; namely, FNDS and CD, to compare the solutions. In FNDS,  $R$  initial populations are compared and sorted. The chromosomes are chosen and evaluated using the concept of domination [13]. In order to do this, all chromosomes in the first nondominated front are first found. Then, in order to find the chromosomes in the next nondominated front, the solutions of the previous fronts are disregarded temporarily. This procedure is repeated until all solutions are set into fronts.

After sorting the populations, a CD measure is defined to evaluate solution fronts of populations in terms of the relative density of individual solutions [13]. To do this, consider  $Z$  and  $f_k; k = 1, 2, \dots, M$  as the number of nondominated solutions in a particular front ( $F$ ) and the objective functions, respectively. Besides, let  $d_i$  and  $d_j$  be the value of the CD on the solution  $i$

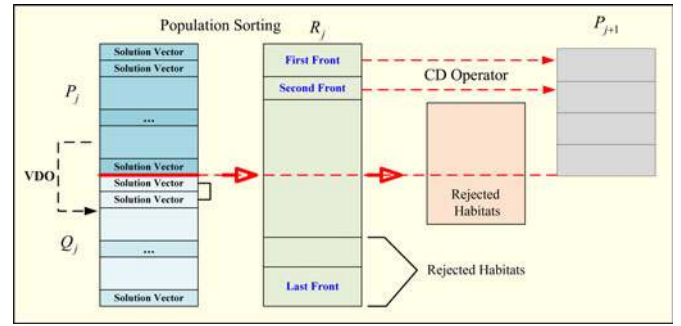


Fig. 2. Evolution process in the proposed MOVDO approach.

and  $j$ , respectively. Then, the CD is obtained using the following steps:

- 1) set  $d_i = 0$  for  $i = 1, 2, \dots, Z$ ;
- 2) sort all objective functions  $f_k; k = 1, 2, \dots, M$  in ascending order;
- 3) the CD for end solutions in each front ( $d_1$  and  $d_Z$ ) are  $d_1 = d_Z \rightarrow \infty$ ;
- 4) the CD for  $d_j; j = 2, 3, \dots, (Z - 1)$  are  $d_j = d_j + (f_{k_{j+1}} - f_{k_{j-1}})$ .

To select individuals of the next generation, the crowded tournament selection operator  $\succ$  is applied [13]. In order to do that, the following steps are required.

- Step 1) choose  $n$  individuals in the population randomly.
- Step 2) the nondominated rank of each individual should be obtained and the CD of the solutions having equal nondominated rank is calculated.
- Step 3) the solutions with the least rank are the selected ones. Moreover, if more than one individual share the least rank, the individual with the highest CD should be selected.

In other words, the comparison criterion of MOVDO algorithm's solutions can be written as follows: If  $q_x < q_y$  or ( $q_x = q_y$  and  $d_x > d_y$ ) then  $x \succ y$  where  $q_x$  and  $q_y$  are the ranks and  $d_x$  and  $d_y$  are the CDs. In this paper, a polynomial neighborhood structure for the selected chromosome is performed.

After operating the aforementioned concepts and operators, the parents and offspring population should be combined to ensure the elitism. Since the combined population size is naturally greater than the original population size  $N$ , once more, nondomination sorting is performed. In fact, chromosomes with higher ranks are selected and added to the populations until the population size becomes  $N$ . The algorithm stops when a predetermined number of iterations (or any stopping criteria) are reached.

Fig. 2 illustrates the evolution process of the proposed MOVDO schematically. The process is started by initializing the initial population of the solution vectors  $P_j$ . Then, the new operators are implemented on  $P_j$  to create a new population  $Q_j$ . The combination of  $P_j$  and  $Q_j$  creates  $R_j$  for keeping elitism in the algorithm. In this step, vectors of  $R_j$  are sorted in several fronts based on FNDS and CD. Using the proposed selection method, a population of the next iteration  $P_{j+1}$  is chosen to have a predetermined size.

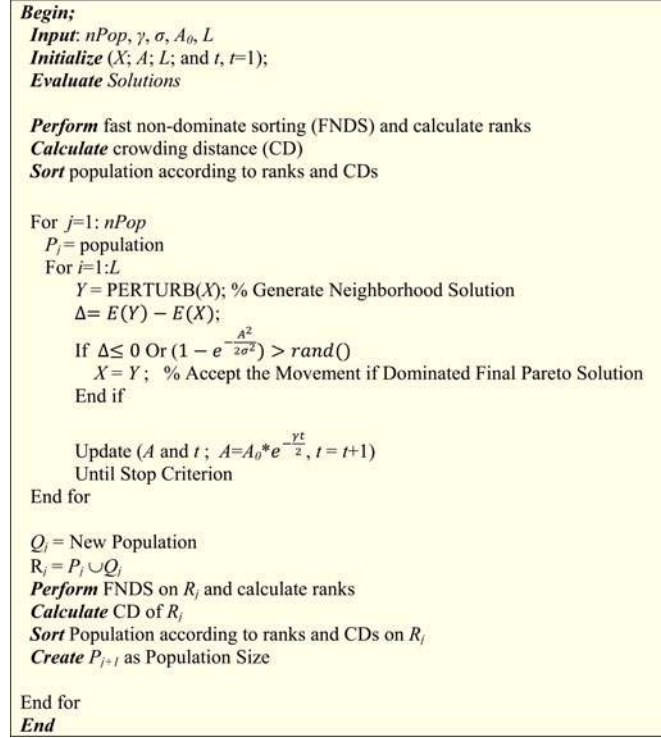


Fig. 3. MOVDO pseudocode.

In the main loop of the MOVDO algorithm, the parents and offspring populations should be combined to ensure the elitism. Since the combined population size is naturally greater than the original population size ( $N$ ), the nondomination sorting operator is applied. In fact, solutions with higher ranks are selected and added to the populations until the population size becomes  $N$ . The last front also consists of the population based on the CD. The solutions that are excluded are called rejected habitats.

It is worthwhile to mention that using Pareto dominance solutions, the algorithm is a computationally efficient algorithm implementing the idea of a selection method based on classes of dominance of all the solutions. In order to improve the clarity of the proposed algorithm, we present the pseudocode and a flowchart for the MOVDO method in Figs. 3 and 4, respectively. The chromosome structure is based on the chromosome proposed by Hajipour and Pasandideh [27] and Rahmati *et al.* [47].

To demonstrate the performance of the proposed MOVDO, two Pareto-based MOEAs, that is, NSGA-II and MOSA are applied. The main difference between these two MOEAs and MOVDO is in the evolution process of the algorithms from  $P_t$  to  $Q_t$ . Furthermore, NSGA-II and MOSA are different in their selection strategies. NSGA-II uses a binary tournament selection and MOSA uses the roulette wheel selection strategy. Accordingly, after generating or modifying populations by means of single-objective operators of the algorithms (e.g., GA, SA, or VDO), the population is dealt with in a multiobjective way in a similar fashion in all the algorithms. Besides, to minimize the impact of using different operators on the performance

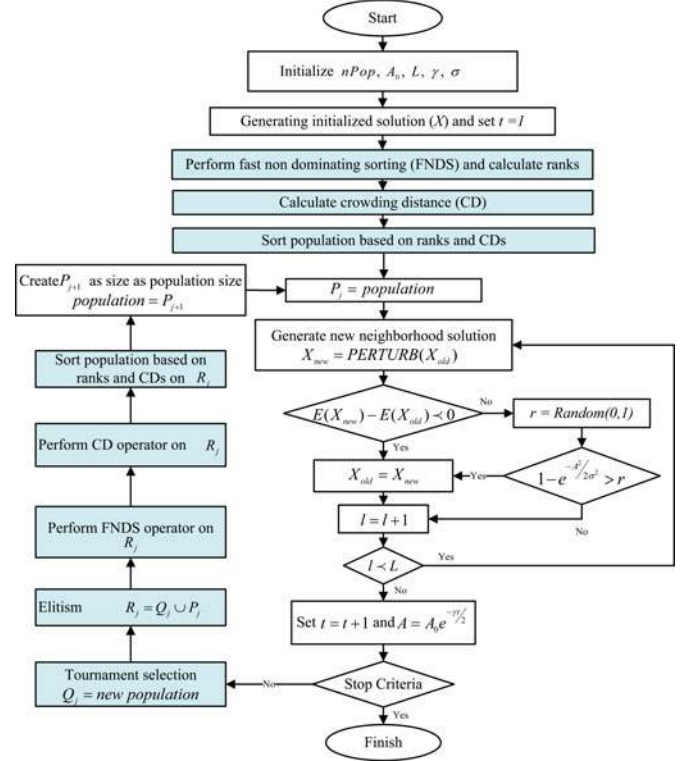


Fig. 4. MOVDO flowchart.

comparison process of the algorithms, operators are designed identically.

To do so, the neighborhood structure of the proposed MOVDO is designed similar to the mutation operator of NSGA-II and the neighborhood structure of MOSA. Moreover, in NSGA-II, the crossover operator is also designed similarly using a uniform crossover operator [28]. A flowchart of the proposed NSGA-II is depicted in Fig. 5. Moreover, to clarify the process of the proposed MOSA, Fig. 6 represents a pseudocode of this algorithm.

In the next section, we solve the RQLAP. We analyze the result and demonstrate the effectiveness of the proposed MOVDO in the area of multiobjective optimization problems.

#### IV. RESULT ANALYSIS AND COMPARISONS

In order to solve the proposed RQLAP, we implemented three multiobjective Pareto-based metaheuristic algorithms. The algorithms are calibrated by statistical approaches in the literature including response surface methodology and the Taguchi method which are provided in Table II. In a single objective algorithm, the objective function is the representation of the efficiency and the computational (CPU) time is the representation of the effectiveness, although in a multiobjective algorithm for assessing the efficiency, different metrics such as: mean to ideal (MID), *diversity*, and *spacing* metrics can be applied to develop a good measure of performance [56]. Meanwhile, *time* can be still used as a good metric for evaluating the effectiveness. To



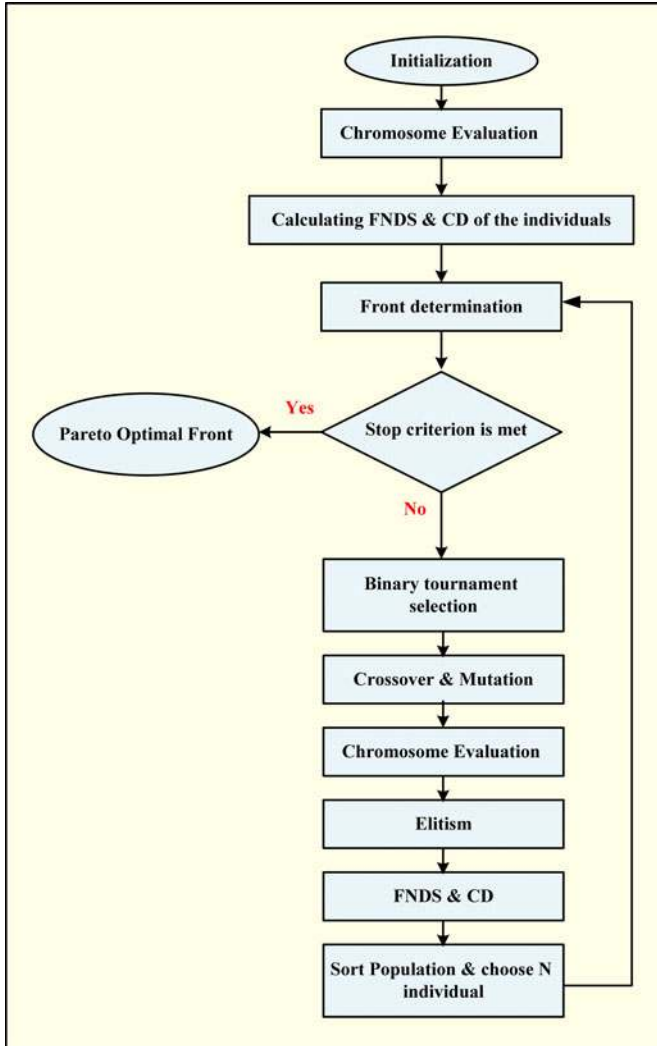


Fig. 5. NSGA-II flowchart.

evaluate the performances of the proposed MOVDO, we applied four standard metrics including spacing, number of Pareto solution, computational time, and the multiobjective coefficient of variation.

#### A. Multiobjective Coefficient of Variation (MOCV)

Measures the division of diversity into MID which considered convergence and diversity of the Pareto solutions [47].

#### B. Spacing

Measures the standard deviation of the distances among solutions of the Pareto front in which smaller values are better [56].

#### C. Number of Found Solutions (NOS)

Counts the number of the Pareto solutions in Pareto optimal front in which bigger values are better [56].

1) *Computational Time (CPU)*: he CPU time of running the algorithms to reach near optimum solutions.

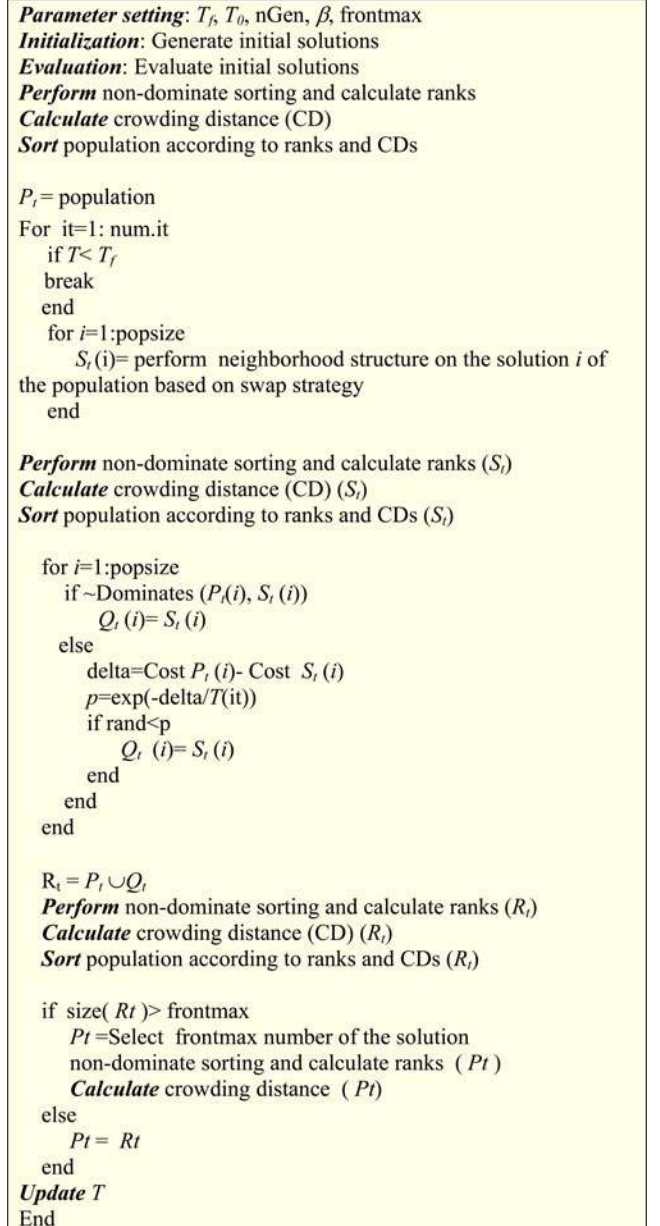


Fig. 6. MOSA pseudocode.

TABLE II  
PARAMETER VALUES OF THE ALGORITHMS ALONG WITH THEIR TUNING PROCEDURE

Multi-Objective Algorithms	Algorithm Parameters	Optimum Amount	Selected Policy
NSGA-II	$nPop$	25	RSM (Pasandideh et al., 2013)
	$P_c$	0.6	
	$P_m$	0.4	
	$Ngen$	100	
MOSA	$T_0$	500	RSM (Pasandideh et al., 2013)
	$Popsize$	5	
	$Ngen$	500	
	$\beta$	0.99	
MOVDO	$nPop$	5	Taguchi by (Mousavi et al., 2013)
	$A_0$	6	
	$L$	40	
	$\sigma$	1.5	
	$\gamma$	0.05	



TABLE III  
MULTIOBJECTIVE CALCULATIONS OF THE ALGORITHM METRICS

Problem No.	MOCV			NOS			Spacing			CPU Time		
	NSGA-II	MOSA	MOVDO	NSGA-II	MOSA	MOVDO	NSGA-II	MOSA	MOVDO	NSGA-II	MOSA	MOVDO
1	9.93	17.12	7.19	21	9	9	3.21E+08	3.12E+08	2.78E+07	21.81	19.81	12.91
2	3.32	16.84	4.83	22	8	9	4.12E+08	4.82E+08	4.01E+07	31.84	22.81	21.93
3	3.81	7.91	6.93	24	9	10	4.74E+08	9.18E+09	7.16E+08	32.81	28.71	22.91
4	1.10	8.48	5.91	23	7	11	3.76E+09	3.40E+09	4.18E+08	39.91	31.84	29.81
5	7.50	11.87	7.93	21	3	4	6.57E+08	5.88E+08	3.17E+08	41.99	38.81	38.71
6	5.43	12.92	6.94	24	6	5	3.17E+08	3.79E+09	5.19E+07	41.19	34.91	31.91
7	12.76	9.19	5.83	19	5	4	3.11E+10	3.31E+09	3.15E+09	50.11	45.91	44.91
8	29.73	22.91	25.93	25	7	6	3.17E+10	5.40E+08	4.14E+08	51.75	44.18	31.91
9	2.81	5.09	2.85	23	9	7	5.18E+09	2.10E+07	8.42E+06	49.52	39.01	30.01
10	1.98	4.98	9.60	21	3	8	4.43E+09	7.56E+07	5.68E+07	62.05	38.91	31.91
11	0.61	3.82	3.11	19	6	4	4.19E+08	3.16E+09	5.14E+07	68.94	41.74	32.28
12	4.11	9.10	6.91	21	6	5	6.52E+08	3.13E+10	4.14E+08	66.91	48.94	39.01
13	1.91	3.92	1.83	24	5	7	3.49E+09	3.98E+10	3.32E+07	80.51	51.74	41.84
14	13.91	19.95	22.91	23	8	6	5.18E+08	4.17E+09	3.18E+07	88.01	52.74	47.81
15	8.49	14.84	7.86	25	8	6	6.49E+10	6.42E+09	2.44E+07	99.81	65.83	49.81
16	1.83	10.04	4.83	19	7	7	2.13E+09	5.19E+07	2.18E+08	91.91	71.91	52.81
17	4.93	15.93	5.92	25	4	8	3.11E+10	7.67E+09	5.45E+10	110.85	77.81	61.81
18	4.98	8.93	9.83	18	3	4	1.42E+10	7.42E+09	5.64E+10	120.95	81.74	65.32
19	NAS	12.95	11.94	NAS	5	7	NAS	4.15E+10	6.45E+10	205.73	101.84	82.91
20	NAS	NAS	12.85	NAS	NAS	8	NAS	NAS	6.34E+11	319.74	204.81	137.73

As mentioned before, the proposed multiobjective algorithm is applied to solve the multiobjective facility location problems in the literature [43]. The experiments are implemented on 20 test problems. Furthermore, to eliminate uncertainties of the solutions, each problem is analyzed three times under different random environments. Then, the averages of these three runs are treated as the ultimate responses. The MOVDO algorithm compares with MOSA and NSGA-II as most applicable Pareto-based MOEAs in the literature to demonstrate the capability of the proposed algorithm to solve the multiobjective optimization problems.

To evaluate the performance of the proposed MOVDO, Table III reports the multiobjective metrics results on the 20 test problems, in which “NAS” indicates that the algorithm cannot find the Pareto front in the reported time. MATLAB Software [34] has been used to code the proposed metaheuristic algorithms, and the programs have been executed on a 2 GHz laptop with an 8 GB RAM.

The algorithms are statistically compared based on the analysis of variance (ANOVA) test. These outputs are reported in Table IV in terms of the defined metrics. In order to clarify our statistical results, Box-plots are represented in Figs. 7–10.

Based on the statistical outputs presented in Table IV and Figs. 8 and 10, NSGA-II shows better performances in terms of NOS, while MOVDO has better performance in terms of the CPU time. In addition, according to MOCV and spacing, we represent the comparability of MOVDO in comparison with NSGA-II and MOSA, in which the algorithms have no significant differences and are statistically equivalent. We should note that this conclusion is confirmed at the 95% confidence level. Moreover, the graphical comparisons of all the metrics on 20 test problems are shown in Fig. 11. Based on these graphical outputs, by increasing the size of the problems, the MOVDO

TABLE IV  
ANALYSIS OF VARIANCE RESULTS

MOCV Metric					
Source	DF	SS	MS	F	P
Algorithms	2	215.2	107.6	2.81	0.069
Error	54	2066.2	38.3		
Total	56	2281.4			
NOS Metric					
Source	DF	SS	MS	F	P
Algorithms	2	2987.87	1493.93	328.13	0.000
Error	54	245.85	4.55		
Total	56	3233.72			
Spacing Metric					
Source	DF	SS	MS	F	P
Algorithms	2	1.25787E+22	6.28934E+21	0.88	0.422
Error	54	3.87529E+23	7.17646E+21		
Total	54	4.00108E+23			
CPU Time Metric					
Source	DF	SS	MS	F	P
Algorithms	2	15482	7741	3.21	0.048
Error	57	137322	2409		
Total	59	152804			

algorithm performs well especially in the terms of the NOS and CPU time metrics. Furthermore, in test problems 19 and 20, NSGA-II and MOSA could not find the Pareto front. However, in these large size problems, MOVDO can find the Pareto front. These features conclude the robustness of the proposed

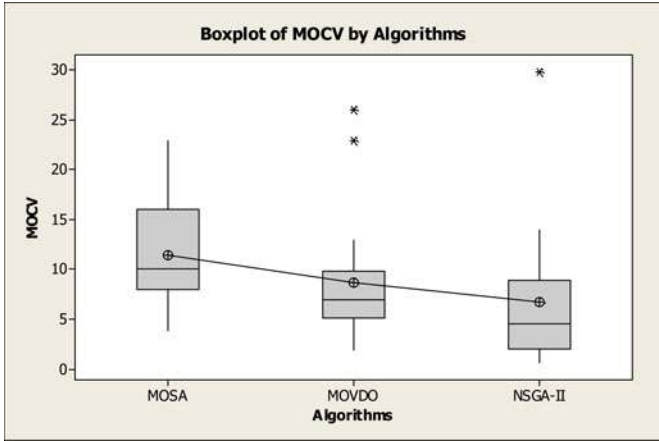


Fig. 7. Box-plot of the algorithm comparisons based on the MOCV metric.

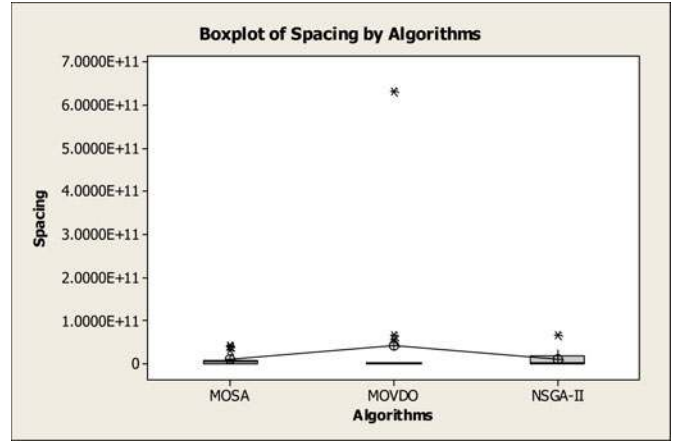


Fig. 9. Box-plot of the algorithm comparisons based on the spacing metric.

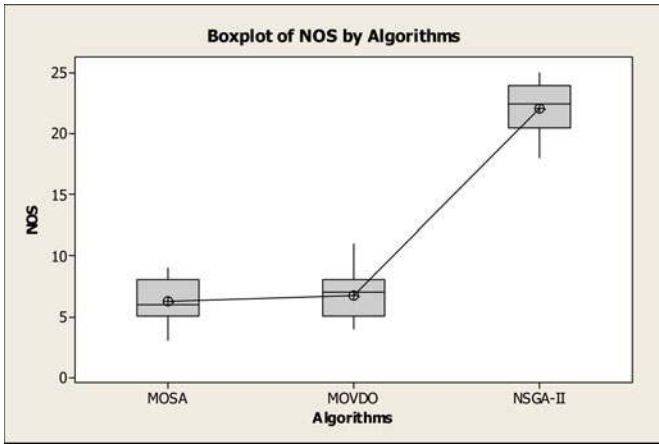


Fig. 8. Box-plot of the algorithm comparisons based on the NOS metric.

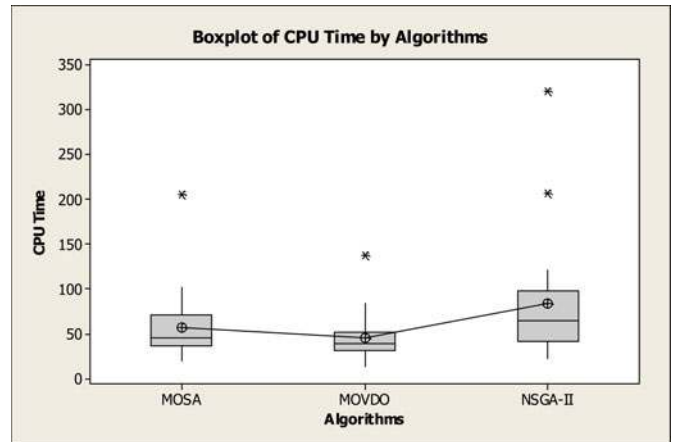


Fig. 10. Box-plot of the algorithm comparisons based on the CPU time metric.

MOVDO in solving large-size multiobjective optimization problems.

To increase the readability of the proposed MOVDO approach, Fig. 12 is used to represent the nondominated solutions of a single run of the proposed MOVDO algorithm at the final iteration.

In summary, a decision maker can select the best algorithm based on his/her preference(s). MOVDO is the best method if computational time is the most important metric for the decision maker. In addition, MOVDO is more effective than NSGA-II and MOSA in solving large problems. NSGA-II is the best method if the NOS in the most important metric for the decision maker.

V. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this paper, we introduced the RQLAP in the literature and formulated the problem as a multiobjective nonlinear integer mathematical programming model. The congestion and reliability concepts were considered in the problem formulation. The RQLAP was solved to determine the number of optimal facilities, the optimal allocation of the demand nodes into the facilities, and the optimal number of servers at each facility.

The goals were to optimize the waiting time, costs, and reliability, simultaneously. Due to the complexity of the problem, we developed the multiobjective version of a novel metaheuristic algorithm based on vibration theory called MOVDO. We used NSGA-II and MOSA, two common metaheuristic algorithms, to optimize the model and demonstrate the performance of the algorithm. We developed three metaheuristic algorithms, including MOVDO, NSGA-II, and MOSA to solve this NP-hard problem. We used the standard metrics for multiobjective comparison to show that the MOVDO approach works better than the other two algorithms, especially for larger size problems and the metric of computation time.

For future research, the model can be extended by considering capacitated queue as *M/M/m/k* queuing systems. In the proposed model, the service capacity is unlimited, while, in real world situations, the service may be capacitated. The performance of the proposed MOVDO can be evaluated in different optimization problems. We also propose the application of the proposed method to real-world problems including automated teller machines location, police department stations, and telecommunication switching centers, among others.

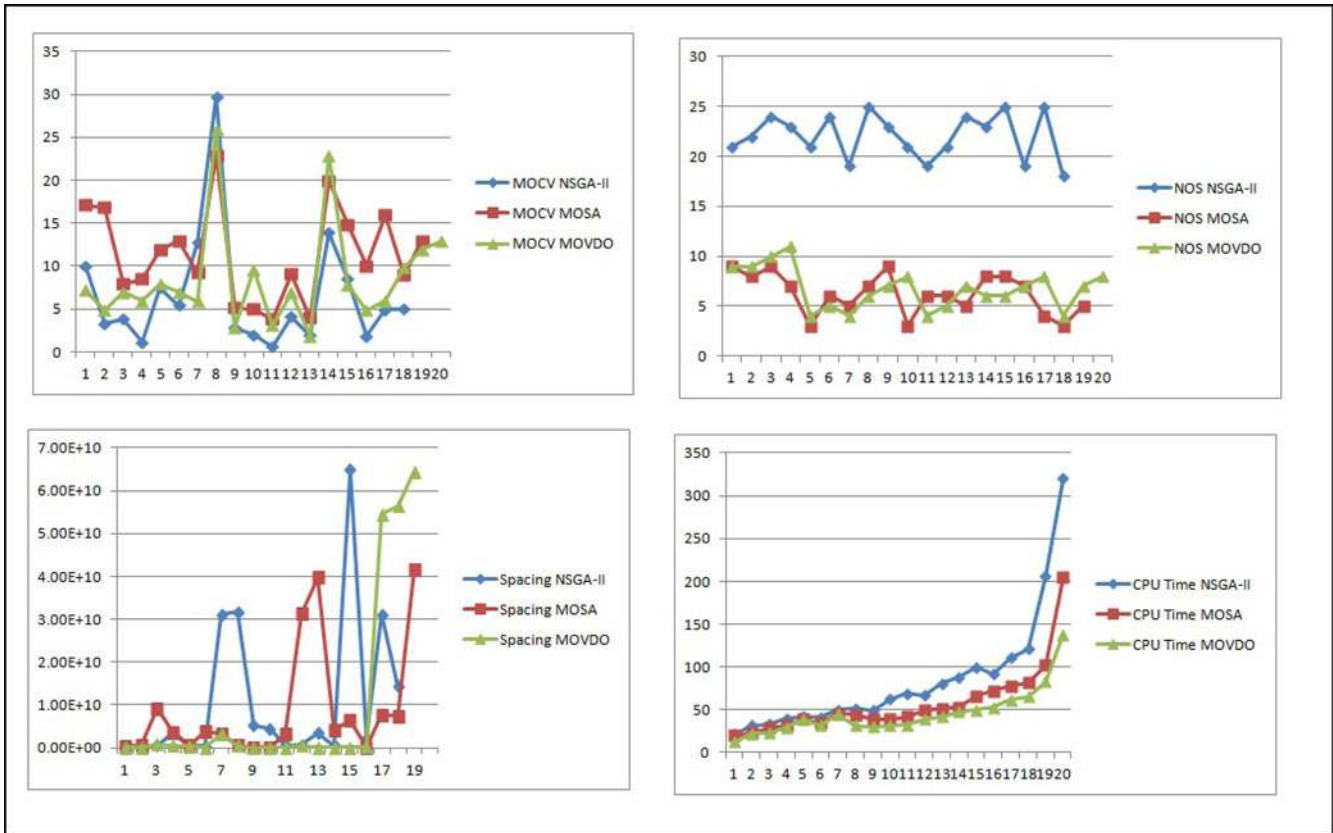


Fig. 11. Graphical comparison of the algorithms based on the four metrics.

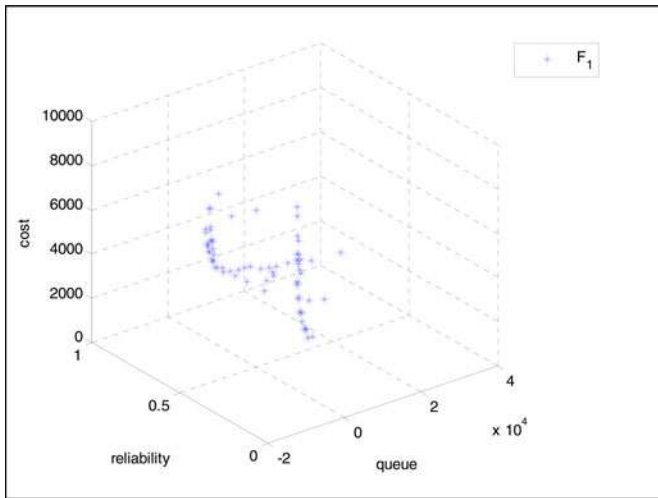


Fig. 12. Pareto solutions obtained from the proposed MOVDO approach.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the editor-in-chief for their constructive comments and suggestions.

REFERENCES

[1] R. Aboolian, O. Berman, and Z. Drezner, "The multiple server center location problem," *Ann. Oper. Res.*, vol. 167, pp. 337–352, 2009.

[2] O. Berman and R. Larson, "Optimal 2-facility network districting in the presence of queuing," *Transp. Sci.*, vol. 19, pp. 261–277, 1985.

[3] O. Berman, D. Krass, and J. Wang, "Locating service facilities to reduce lost demand," *IIE Trans.*, vol. 38, pp. 933–946, 2006.

[4] O. Berman and Z. Drezner, "The multiple server location problem," *J. Oper. Res. Soc.*, vol. 58, pp. 91–99, 2007.

[5] B. Boffey, R. Galvao, and L. Espejo, "A review of congestion models in the location of facilities with immobile servers," *Eur. J. Oper. Res.*, vol. 178, pp. 643–662, 2007.

[6] B. A. Arabania and R. Z. Farahani, "Facility location dynamics: An overview of classifications and applications," *Comput. Ind. Eng.*, vol. 62, pp. 408–420, 2012.

[7] A. Chambari, S. H. A. Rahmati, A. A. Najafi, and A. Karimi, "A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies," *Comput. Ind. Eng.*, vol. 63, pp. 109–119, 2012.

[8] C. A. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. Berlin, Germany: Springer, 2007.

[9] D. W. Coit, "Maximization of system reliability with a choice of redundancy strategies," *IIE Trans.*, vol. 35, no. 6, pp. 535–543, 2003.

[10] D. W. Coit and Liu, "System reliability optimization with k-out-of-n subsystems," *Quality Safety Eng.*, vol. 7, no. 2, pp. 129–143, 2000.

[11] R. B. Cooper, *Introduction to Queuing Theory*, 2nd ed. New York, NY, USA: Elsevier North, 1980.

[12] J. Current, M. Daskin, and D. Schilling, "Discrete network location models," in *Facility Location: Applications and Theory*, Z. Drezner and H. W. Hamacher, Eds. Heidelberg, Germany: Springer, 2002.

[13] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[14] G. Derringer and R. Suich, "Simultaneous optimization of several response variables," *J. Quality Technol.*, vol. 12, pp. 214–219, 1980.

[15] R. Z. Farahani, N. Asgari, N. Heidari, M. Hosseinini, and M. Goh, "Covering problems in facility location: A review," *Comput. Ind. Eng.*, vol. 62, pp. 368–407, 2012.

[16] R. Z. Farahani and M. Hekmatfar, "Facility location: Concepts, models, algorithms and case studies," in *Contributions to Management Science*, Heidelberg, Germany: Physica-Verlag, 2009.



- [17] R. Z. Farahani, M. SteadieSeifi, and N. Asgari, "Multiple criteria facility location problems: A survey," *Appl. Math. Modelling*, vol. 34, pp. 1689–1709, 2010.
- [18] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*. Chichester, U.K.: Wiley, 1966.
- [19] R. L. Francis, L. F. McGinnis, and J. A. White, *Facility Layout and Location: An Analytical Approach*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.
- [20] D. E. Fyffe, W. W. Hines, and N. K. Lee, "System reliability allocation and a computational algorithm," *IEEE Trans. Rel.*, vol. R-17, no. 2, pp. 64–69, Jun. 1968.
- [21] Z. W. Geem, J.-H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, pp. 60–68, 2001.
- [22] Z. W. Geem, J.-H. Kim, and G. V. Loganathan, "Harmony search optimization: Application to pipe network design," *Int. J. Modeling Simul.*, vol. 22, pp. 125–133, 2002.
- [23] Z. W. Geem, Harmony Search Algorithm for Solving Sudoku. B. Apolloni, R. J. Howlett, and L. Jain, eds., KES 2007, Part I. LNCS (LNAI), Berlin, Germany: Springer, vol. 4692, 2007, pp. 371–378.
- [24] F. Glover, "Heuristic for integer programming using surrogate constraints," *Decision Sci.*, vol. 8, no. 1, pp. 156–166, 1977.
- [25] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison Wesley, 1989.
- [26] D. Gross and C. M. Harris, *Fundamental of Queuing Theory*, 3rd ed. New York, NY, USA: Wiley Inter science, 1988.
- [27] V. Hajipour and S. H. R. Pasandideh, "Proposing an adaptive particle swarm optimization for a novel bi-objective queuing facility location model," *Econ. Comput. Econ. Cybern. Studies Res.*, vol. 3, pp. 223–240, 2012.
- [28] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, 2nd ed. New York, NY, USA: Wiley, 2004.
- [29] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, vol. 82, pp. 781–798, 2004.
- [30] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: Univ. Michigan Press, 1975.
- [31] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [32] W. Kuo and V. R. Prasad, "An annotated overview of system-reliability optimization," *IEEE Trans. Rel.*, vol. 49, no. 2, pp. 176–187, Jun. 2000.
- [33] R. L. Love, J. G. Morris, and G. O. Wesolowsky, *Facility Location: Models and Methods*. New York, NY, USA: North-Holland Publishing Company, 1988.
- [34] MATLAB Version 7.10.0.499 (R2010a). The MathWorks, Inc. Protected by U.S. and international patents, 2010
- [35] E. Mehdizadeh and R. Tavakkoli-Moghaddam, "Vibration damping optimization algorithm for an identical parallel machine scheduling problem," in *Proc. 2nd Int. Conf. Iranian Oper. Res. Soc.*, Babolsar, Iran, May 20–22, 2009. Available: [http://www.civilica.com/EnPaper-ICIORS02-ICIORS02\\_230.html](http://www.civilica.com/EnPaper-ICIORS02-ICIORS02_230.html)
- [36] E. Mehdizadeh, M. R. Tavarroth, and V. Hajipour, "A new hybrid algorithm to optimize stochastic-fuzzy capacitated multi-facility location-allocation problem," *J. Optimization Ind. Eng.*, vol. 7, pp. 71–80, 2011.
- [37] M. T. Melo, S. Nickel, and F. Saldanha-da-Gama, "Facility location and supply chain management—A review," *Eur. J. Oper. Res.*, vol. 196, pp. 401–412, 2009.
- [38] S. M. Mousavi, S. T. A. Niaki, E. Mehdizadeh, and M. R. Tavarroth, "The capacitated multi-facility location—Allocation problem with probabilistic customer location and demand: Two hybrid meta-heuristic algorithms," *Int. J. Syst. Sci.*, vol. 44, pp. 1897–1912, 2013.
- [39] R. Oftadeh, M. J. Mahjoob, and M. Shariatpanahi, "A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search," *Comput. Math. Appl.*, vol. 60, pp. 2087–2098, 2010.
- [40] Y. Ohsawa, "A geometrical solution for quadratic bicriteria location models," *Eur. J. Oper. Res.*, vol. 114, pp. 380–388, 1999.
- [41] S. H. R. Pasandideh and S. T. A. Niaki, "Genetic application in a facility location problem with random demand within queuing framework," *J. Intell. Manuf.*, vol. 23, no. 3, pp. 651–659, 2012.
- [42] S. H. R. Pasandideh, S. T. A. Niaki, and J. A. Yeganeh, "A parameter-tuned genetic algorithm for multi-product economic production quantity model with space constraint, discrete delivery orders and shortages," *Advances Eng. Softw.*, vol. 41, pp. 306–314, 2010.
- [43] S. H. R. Pasandideh, S. T. A. Niaki, and V. Hajipour, "A multi-objective facility location model with batch arrivals: Two parameter-tuned meta-heuristic algorithms," *J. Intell. Manuf.*, vol. 24, no. 2, pp. 331–348, 2013.
- [44] S. H. R. Pasandideh, S. T. A. Niaki, and N. Tokhmehchi, "A parameter-tuned genetic algorithm to optimize two-echelon continuous review inventory systems," *Expert Syst. Appl.*, vol. 38, pp. 11708–11714, 2011.
- [45] S. H. R. Pasandideh and S. T. A. Niaki, "Multi-response simulation optimization using genetic algorithm within desirability function framework," *Appl. Math. Comput.*, vol. 175, pp. 366–382, 2006.
- [46] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization An overview," *Swarm Intell.*, vol. 1, pp. 33–57, 2007.
- [47] S. H. A. Rahmati, V. Hajipour, and S. T. A. Niaki, "A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem," *Appl. Soft Comput.*, vol. 13, pp. 1728–1740, 2013.
- [48] H. Shavandi and H. Mahlooji, "A fuzzy queuing location model with a genetic algorithm for congested systems," *Appl. Math. Comput.*, vol. 181, pp. 440–456, 2006.
- [49] S. S. Syam, "A multiple server location-allocation model for service system design," *Comput. Oper. Res.*, vol. 35, pp. 2248–2265, 2008.
- [50] L. V. Snyder, M. P. Scaparra, M. S. Daskin, and R. L. Church, "Planning for disruptions in supply chain networks," *Tuts. Oper. Res. Inform.*, pp. 400–416, 2006.
- [51] L. V. Snyder and M. S. Daskin, "Reliability models for facility location: The expected failure cost case," *Transp. Sci.*, vol. 39, no. 3, pp. 416–440, 2005.
- [52] A. A. Taleizadeh, S. T. A. Niaki, and F. Barzinpour, "Multiple-buyer multiple-vendor multi-product multi-constraint supply chain problem with stochastic demand and variable lead-time: A harmony search algorithm," *Appl. Math. Comput.*, vol. 217, pp. 9234–9253, 2011.
- [53] Q. Wang, R. Batta, and C. Rump, "Algorithms for a facility location problem with stochastic customer demand and immobile servers," *Ann. Oper. Res.*, vol. 111, pp. 17–34, 2002.
- [54] O. Yeniay and B. Ankare, "Penalty function methods for constrained optimization with genetic algorithms," in *Math. Comput. Appl.* vol. 10, pp. 45–56, 2005.
- [55] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving From Nature* Berlin, Germany: Springer, 1998, 292–301.
- [56] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*. Barcelona, Spain: CIMNE, 2001, pp. 95–100.