# Author's Accepted Manuscript


www.elsevier.com/locate/caor

Bi-objective vibration damping optimization for congested location-pricing problem

Vahid Hajipour, Reza Zanjirani Farahani, Parviz Fattahi

PII: S0305-0548(16)00003-4
DOI: http://dx.doi.org/10.1016/j.cor.2016.01.001
Reference: CAOR3910

To appear in: *Computers and Operation Research*

Received date: 9 June 2014
Revised date: 13 October 2015
Accepted date: 2 January 2016

Cite this article as: Vahid Hajipour, Reza Zanjirani Farahani and Parviz Fattahi Bi-objective vibration damping optimization for congested location-pricin problem, *Computers and Operation Research* http://dx.doi.org/10.1016/j.cor.2016.01.001

# Bi-objective vibration damping optimization for congested location-pricing problem

**Vahid Hajipour[a], Reza Zanjirani Farahani[b*], Parviz Fattahi[a]**

[a] Industrial Engineering Department, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran
[b] Kingston Business School, Kingston University, Kingston Upon Thames, Surrey, The UK

## ABSTRACT

This paper presents a bi-objective mathematical programming model for the restricted facility location problem, under a congestion and pricing policy. Motivated by various applications such as locating server on internet mirror sites and communication networks, this research investigates congested systems with immobile servers and stochastic demand as *M/M/m/k* queues. For this problem, we consider two simultaneous perspectives; (1) customers who desire to limit waiting time for service and (2) service providers who intend to increase profits. We formulate a bi-objective facility location problem with two objective functions: (i) maximizing total profit of the whole system and (ii) minimizing the sum of waiting time in queues; the model type is mixed-integer nonlinear. Then, a multi-objective optimization algorithm based on vibration theory (so-called multi-objective vibration damping optimization (MOVDO)), is developed to solve the model. Moreover, the Taguchi method is also implemented, using a response metric to tune the parameters. The results are analyzed and compared with a non-dominated sorting genetic algorithm (NSGA-II) as a well-developed multi-objective evolutionary optimization algorithm. Computational results demonstrate the efficiency of the proposed MOVDO to solve large-scale problems.

**Keywords:** restricted facility location; computational intelligence; multi-objective optimization; pricing; queuing theory.



## 1. Introduction

The traditional goal in facility location problems (FLPs) is to locate the facilities in the best locations to minimize fixed location and transportation costs. Hakimi [24], Toregas et al. [45], Love et al. [27], Marianov and Revelle [29], and Hodgson and Berman [26] proposed various models and solutions methodologies for FLPs. Farahani and Hekmatfar [17], Melo et al. [31], Farahani et al. [16], Boloori Arabania and Farahani [9], and Farahani et al. [18] provided more detail on FLPs and their solving methodologies. In addition to FLPs, several other streams of research such as queuing, pricing and multi-objective heuristics techniques are related to this paper which will be explained in detail as follows.

* Corresponding. Fax: +44 (0)20 8417 7026; Tel: +44 (0)20 8417 5098; Email: Zanjiranireza@gmail.com; R.Zanjiranifarahani@kingston.ac.uk

In the many real-life applications of FLPs such as gas stations and car parking, customer demand on arrival at facilities is heavy; these facilities are called congested [8]. Therefore, queues are formed in these systems; consequently, waiting time will be a key parameter in such FLPs. Thus, the combination of queuing theory with FLPs emerges to create queuing facility location problems (QFLPs) which are more realistic in some applications. Berman and Larson [3] proposed a nonlinear location problem with congested facilities which behaves like a *M/G/*1 queue. Wang et al. [46] presented a congested FLP to minimize aggregate traveling and waiting times. Wang et al. [47] presented several heuristic algorithms to solve location problems with budget limitations. Berman and Drezner [4] formulated a facility location problem within a *M/M/m* (multi-server) queuing framework. Syam [43] presented a nonlinear multi-server location-allocation problem to minimize total costs of the system. Zarrinpoor and Seifbarghy [50] developed FLPs in competitive environments to determine a specific percentage of the market share in the context of cost minimization. Hajipour and Pasandideh [21] proposed a multi-objective congested FLP which behaves as a $M^{[x]}$/M/1 queuing system. Chambari et al. [11] presented two Pareto-based algorithms based on a genetic algorithm (GA) for a facility location model with two conflicting objectives for *M/M/1/k* queues. Pasandideh and Niaki [38] applied a GA and desirability function approach to solve a bi-objective facility location model with classical queues. Hajipour and Pasandideh [22] optimized a bi-objective congested facility location problem by an adaptive multi-objective particle swarm optimization. Pasandideh et al. [39] presented a multi-objective facility location model in which batch demands arrive on the system; they solved the problem by a multi-objective GA and SA (simulated annealing) algorithm. Rahmati et al. [42] presented multi-objective FLPs considering multiple servers at each facility. They solved the model with multi-objective Pareto-based meta-heuristic algorithms.

Another aspect of real-world applications is that demand nodes are mainly influenced by pricing strategy. Some researchers have focused on hybridizing thepricing concept with FLPs ([19, 15, 28]) and also with queuing theory ([41, 2]). Since these problems are chiefly multi-objective, the majority of solving methods are applied to find Pareto solutions for multi-objective FLPs. Additionally, since exact or hard computing approaches cannot solve NP-hard problems [47, 38, 39], soft computing approaches are applied. Unlike hard computing techniques, soft computing approaches deal with imprecision, uncertainty, and approximation to determine robustness and low-cost solutions. Neural networks, fuzzy logic and evolutionary computations may help in such approaches. Among these, evolutionary algorithms are more popular for developing algorithms to solve the FLP models [10]. Evolutionary algorithms are divided into two classifications, namely single and multi-objective models. Interested readers may refer to Farahani et al. [16] to see a survey of these studies in FLPs.

Berman et al. [5] combined location, pricing and queuing concepts in a single facility location problem on a network to maximize profit. They considered, simultaneously, decision making on location, pricing and service capacity, where the demand depends on price, distance and waiting time at the facilities. They presented an algorithm to achieve the optimal price and capacity. Later, Berman et al. [6] extended their previous research to a multi-facility location model. In these two research works, they assumed that customers have a prior knowledge about the expected waiting times at the facilities. Abouee-Mehrizi et al. [1] extended these models to locating *m* facilities on a network with *n* demand nodes. They assumed a *M/M/1* queuing system in which (a) customers may balk the system upon their arrival and (b) all the facilities charge the same price for service.

Among multi-objective algorithms, the non-dominated sorting genetic algorithm (NSGA-II) is one of the commonly used Pareto-based approaches proposed by Deb et al. [14]. This algorithm is applied to various operations research applications including FLPs and their variations. Bhattacharya and Bandyopadhyay [7] applied NSGA-II to solve FLPs with two conflicting objectives. Chambari et al. [11] solved an *M/M/1/k* queue model by using both NSGA-II and non-dominated ranking genetic algorithms (NRGA). Chambari et al. [12] implemented NSGA-II to optimize cost and reliability of the whole system in a redundancy allocation problem. Mehdizadeh and Tavakkoli-Moghaddam [32] proposed a new meta-heuristic optimization algorithm, namely vibration damping optimization (VDO) to solve the parallel machine scheduling problem; VDO is based on the concept of vibration damping in mechanical vibration. Zhang and Lu [53] proposed a multi-objective decision support system to consider how to help users select and use the proposed algorithms. This algorithm simulates the vibration phenomenon. Mehdizadeh et al. [33] proposed a hybrid VDO algorithm to solve the multiple facilities stochastic-fuzzy capacitated location-allocation problem. Mousavi et al. [35] developed a special type of the VDO algorithm to solve the capacitated multi-facility location-allocation problem with probabilistic customer locations and demands. Recently, Hajipour et al. [23] introduced a multi-objective version of VDO for solving multi-objective optimization problems.

In this paper, a hybrid problem of location, pricing, and queuing in a network with *M* customer nodes and *N* potential server nodes is developed. We model the problem mathematically; the model contains two simultaneous objectives of (i) maximizing the profit and (ii) minimizing the sum of waiting time in the whole network. The model is formulated for a system in which each facility behaves as a *M/M/m/k* queuing system; *m* is the number of servers in each facility and *k* is the capacity in the queuing system. We have assumed that various prices at different service facilities are provided. Furthermore, the capacity constraints are considered to make the problem more realistic. This assumption is known as the “mill pricing”; petrol stations and paid car parking areas are examples of mill pricing application.

The closest research paper to this work is Aboouee Mehrizi et al. [1]. However, the contributions of this research to the research literature are as follows:

- This research considers a *M/M/m/K* queuing system at each facility, whereas the previous literature is mainly based upon a *M/M/1* queuing system;
- Various prices are considered for each facility, while the simplifying assumption in the existing literature is based on the same price for all facilities;
- Our mathematical model contains two objectives and is presented in the form of a bi-objective model; in the literature, only single-objective models have been considered to date;
- A multi-objective VDO (MOVDO) is developed to find Pareto solutions. The VDO algorithm is extended, using fast non-dominated sorting and ranking procedures to find Pareto-optimal solutions for multi-objective optimization problems with conflicting and competing objectives. In fact, fast non-dominated sorting and crowding distances have been used to find and manage the Pareto-optimal front. The MOVDO is also analyzed and compared with the best-developed NSGA-II on some standard metrics. To block the impact of algorithm operators, the Taguchi approach is applied.

The rest of the paper is organized as follows: Section 2 details the problem. Section 3 formulates the problem as a non-linear integer programming mathematical model. Section 4 presents the proposed MOVDO algorithm as well as the NSGA-II. Section 5 discusses the tuning parameters of the algorithms. Section 6 analyzes the computational results and investigates the efficiency of the algorithm. Finally, conclusions and future research directions are provided.

## 2. Problem definition

We consider a firm that intends to locate several multi-server facilities in a region. The system under study contains two networks: (1) a customers' network with demand on nodes and (2) a facilities network in which nodes represent candidate location for facilities. The arcs indicate the allocation of demand nodes to facility nodes. Each customer node has a potential number of users which refers to facilities traveling certain distances to receive the service/goods. In order to receive the service/ goods, users refer to the facility that provides the highest utility. User utility is a function of service/ good pricing and the distance between customer nodes and facility nodes. It is rational to assume that potential users refrain from receiving the service/ goods if a desirable price and distance are not provided. Obviously, sensitivity in each customer node toward price and distance is different. Therefore, pricing for service/ goods and location of facilities are the most important determinants for firms, in order to maximize both profit and customer satisfaction levels. Since waiting time in queues is one of the satisfaction factors, firms intend to optimize profit and waiting time simultaneously. To obtain an efficient waiting time, queue length in each facility is controlled by an appropriate pricing policy to obtain the appropriate number of servers at the facility. Figure 1 illustrates the network that is used in this paper. The idea is to achieve the following objectives:

- Optimal number of facilities;
- Optimal allocation process of customer nodes into the opened facilities;
- Optimal number of servers at each facility;
- Optimal queuing capacity at each facility; and
- Optimal price at each facility.

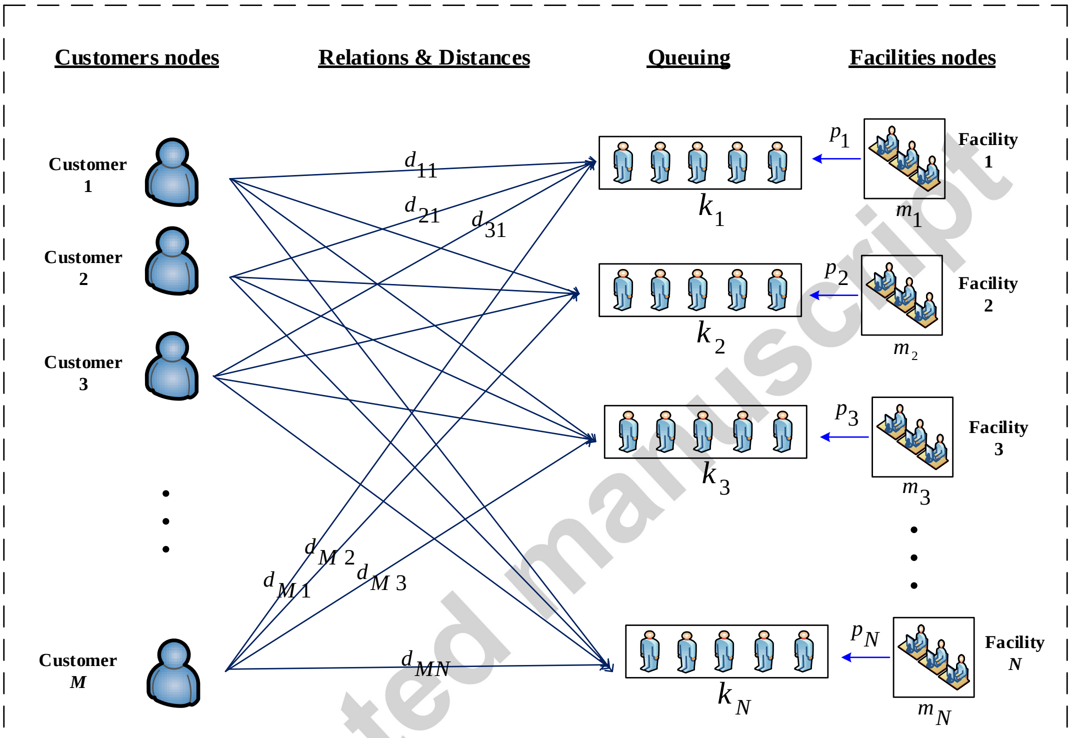


**Figure 1.** Facility location problems behaving as *M/M/m/k* queues.

There are many applications for such a problem in real-life systems, as follows: health systems (including local clinics, hospitals and medical centres, relief distribution centres and reconstruction centre locations), educational systems (such as kindergartens, guidance schools and high schools), police stations, truck terminals, hotels, vending machine locations, city logistics terminals, bus stops, post boxes, air ports, telecommunication systems, petrol stations, blood banking centres, libraries, automatic teller machines location and so forth.

## 3. The proposed mathematical model

We mathematically formulate the location-pricing-queuing problem by using the following notations:

***Sets and indices***

$M$: Set of customer nodes

$N$: Set of locations for potential facilities

$V$: Maximum number of servers which can be on-duty; $(V \leq N)$

$i$: Customer node index $(i = 1, 2, ..., M)$

$j$: Potential facility location index $(j = 1, 2, ..., N)$

***Parameters***

$d_{ij}$ : The travelling distance from costumer $i$ to facility node $j$ ; $(i \in M,\ j \in N)$

$\tau_j$ : The demand rate at open facility node $j$ ; $(j \in N)$

$w_j$ : The expected waiting time of customer batches assigned to facility node $j$ ; $(j \in N)$

$\lambda_i$ : The demand rate of service requests from customer node $i$ ; $(i \in M)$

$e_j$ : Fixed cost of establishing a facility at potential node $j$ ; $(j \in N)$

$cs_j$ : Unit cost of service/goods at facility $j$ ; $(j \in N)$

$\mu_j$ : The service rate for server $j$ ; $(j \in N)$

$g_i$ : Potential number of users in customer node $i$ ; $(i \in M)$

$\alpha_i$ **:** Price sensitivity coefficient in customer node $i$ ; $(i \in M)$

$\beta_i$ **:** Distance sensitivity coefficient in customer node $i$ ; $(i \in M)$

***Decision variable***

$$x_{ij} = \begin{cases} 1 & \text{; if customer } i \text{ is assigned to facility } j \\ 0 & \text{; otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{; if facility } j \text{ is opened} \\ 0 & \text{; otherwise} \end{cases}$$

$m_j$ : The number of servers at opened facility $j$

$k_j$ : Queuing capacity at opened facility $j$

$p_j$ : Price of service/ goods at open facility $j$

Users in each customer node are sensitive to price and distance; the user behavior for receiving the service/goods can be modeled as a linear function of price and distance, as follows [54]:

$$\lambda_{i,j}(p,d) = g_i - p_j \alpha_i - d_{ij} \beta_i \tag{1}$$

$\lambda_{i,j}(p,d)$ is the number of users in customer node $i$ patronized by price $p_j$ in potential facility $j$. Therefore, total demands that potential facility $j$ may encounter is calculated as follows:

$$\tau_j = \sum_{i=1}^{M} \lambda_{i,j}(p,d) x_{ij} \tag{2}$$

The first objective function, profit of facility $j$ by providing the service/ goods to users, is obtained by $(p_j - cs_j)\tau_j$. In the second objective, we assume that each facility acts as a *M/M/m/K* queue, considering Poisson arrival with a mean rate $\tau$, exponentially distributed service time with mean $\mu$, $m$ server to be on-duty at each facility, and the queue capacity to be restricted to $k$ users [20, 50]. In most pricing-queuing problems, the effect of queuing parameters (such as waiting time) is considered in the form of cost reduction revenue [44, 1]. It is assumed that customers get information about the queue and then make a decision about balking or staying in the queue. In this paper, we consider waiting time as customer satisfaction and optimize it simultaneously with profit function. The mathematical formulation of the problem is as follows:

$$\text{Maximize} \quad \sum_{j=1}^{N} (p_j - cs_j)\tau_j - e_j y_j \tag{3}$$

$$\text{Minimize} \quad \sum_{j=1}^{N} \frac{\pi_{0,j}}{m_j!} \left(\frac{\tau_j}{\mu_j}\right)^{m_j} \frac{r_j}{(1-r_j)^2} \left[ 1 - r_j^{\,k_j - m_j + 1} - (1-r_j)(k_j - m_j + 1) r_j^{\,k_j - m_j} \right] \tag{4}$$

$$\lambda_{i,j}(p,d) = g_i - p_j \alpha_i - d_{ij}\beta_i \qquad i=1,\ldots,M \ \& \ j=1,\ldots,N \tag{5}$$

$$\tau_j = \sum_{i=1}^{M} \lambda_{i,j}(p,d) x_{ij} \qquad j=1,\ldots,N \tag{6}$$

$$\tau_j (1 - \pi_{k_j}) \le m_j k_j \qquad j=1,\ldots,N \tag{7}$$

$$r_j = \frac{\tau_j}{m_j \mu_j} \qquad j=1,\ldots,N \tag{8}$$

$$\pi_{0,j} = \frac{1}{\left[ \sum_{s=0}^{m_j - 1} \left(\frac{\tau_j}{\mu_j}\right)^s \frac{1}{s!} + \left(\frac{\tau_j}{\mu_j}\right)^{m_j} \frac{1}{m_j!} \sum_{s=m_j}^{k_j} r_j^{\,s-m_j} \right]} \qquad j=1,\ldots,N \tag{9}$$

$$c_{k_j} = \left(\frac{\tau_j}{\mu_j}\right) \frac{1}{m_j!} \frac{1}{m_j^{\,k_j - m_j}} \qquad j=1,\ldots,N \tag{10}$$

$$\pi_{k_j} = c_{k_j} \pi_{0,j} \qquad j=1,\ldots,N \tag{11}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i = 1,\dots,M \qquad (12)$$

$$x_{ij} \le y_j \qquad i = 1,\dots,M \;\&\; j = 1,\dots,N \qquad (13)$$

$$\sum_{j=1}^{N} y_j \le V \qquad (14)$$

$$y_j \in \{0,1\}$$
$$x_{ij} \in \{0,1\}$$
$$k_j \ge 0,\ Integer$$
$$m_j \ge 0,\ Integer$$
$$p_j \ge 0 \qquad i = 1,\dots,M \;\&\; j = 1,\dots,N \qquad (15)$$

Objective function (3) maximizes the total profit of the whole system and objective function (4) minimizes the total waiting time of customers. Constraint (5) turns out the demand rate that facility *j* receives from customer node *i*. Constraint (6) defines the arrival rate at each facility located at node *j*. Constraint (7) insures the service capacity for each facility. Constraints (8)-(11) provide queuing equations for the *M/M/m/k* queuing system. This queue is a variation of a multi-server system and only a maximum of *k* customers are allowed to stay in the system. The number of customers in the system is a birth-death process with appropriate rates and for a steady-state distribution. The main queuing performance measures can be obtained by a continuous time Markov chain with transition rate matrix and applying Little's law [20]. Constraint (12) ensures that all demands at node *i* will be served by the facilities. Constraint (13) imposes that customers can be captured only by open facilities. Constraint (14) provides an upper bound for the number of open facilities. Finally, Constraint (15) identifies the type of decision variables on the model.

## 4. The Pareto-based meta-heuristics

Numerous exact and heuristic solution approaches have been developed to solve location problems. Many location problems can be modeled by integer programming and solved by conventional techniques such as branch-and-bound. However, for real-life large-sized problems, heuristic approaches need to be used. Since the premise of this paper is a multi-objective optimization problem (or specifically, bi-objective), a novel Pareto-based algorithm called MOVDO is presented to find and manage Pareto solutions of the designed bi-objective mathematical formulation. NSGA-II is also applied to demonstrate the performance of the proposed MOVDO. First, some essential multi-objective concepts are explained in the next section.

### *4.1. Fundamental concept of multi-objective algorithms*

Consider a multi-objective model with a set of conflicting objectives $f(\vec{x}) = [f_1(\vec{x}), ..., f_m(\vec{x})]$ subject to $g_i(\vec{x}) \le 0$ , $i = 1, 2, ..., c$ , $\vec{x} \in X$ , where $\vec{x}$ denotes $n$-dimensional decision variable that can take real, integer, or Boolean value. $X$ is the feasible region, and $f(\vec{x})$ is a set of $m$ objective functions. Then, for a minimization model, solution $\vec{a}$ dominates solution $\vec{b}\,(\vec{a}, \vec{b} \in X)$ if:

1) $f_i(\vec{a}) \le f_i(\vec{b}), \quad \forall i = 1, 2, ..., m$ and

2) $\exists i \in \{1, 2, ..., m\}: f_i(\vec{a}) < f_i(\vec{b})$

In Pareto-based algorithms, the concept of Pareto solutions set (or so-called Pareto front) is the key concept; it is a set of solutions that cannot dominate one another. An efficient Pareto front includes an appropriate convergence and diversity of solutions. Further, the Pareto optimal front (the front obtained in the last iteration of algorithm) has the most convergence and the highest diversity [14]. Figure 2 represents Pareto front solutions in multi-objective concepts for a problem with two minimization-type objective functions.

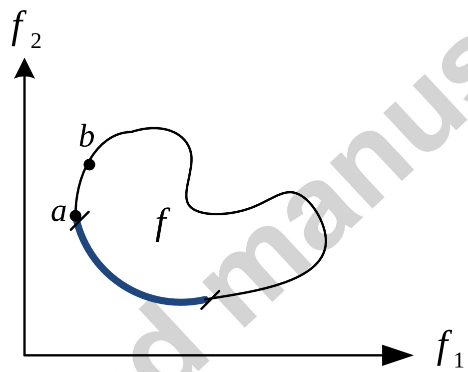


**Figure 2.** Pareto optimal set and domination concept.

### *4.2. Multi-objective vibration damping optimization (MOVDO) algorithm*

VDO is a meta-heuristic based on the concept of vibration damping in mechanical vibration [32]. Hajipour et al. [23] introduced MOVDO for solving multi-objective optimization problems. In this subsection, the multi-objective version of the VDO algorithm has been developed for discrete environments of location-queuing-pricing problems.

### *Solution coding*

To code the solutions, we have used the solution structure in [39]. Our model has three extra decision variables, which add three new vectors to the solution representation structure. Therefore, this coding approach avoids infeasible solution generation. Thus, the majority of restrictions are satisfied and the rest are penalized. The solution representation structure of the problem with $i^{th}$ customer gene ($i^{(i)}$) and $j^{th}$ facility gene ($j^{(j)}$) is represented in Figure 3, where:

- The fourth vector is a new vector indicating the number of servers at all facilities.
- The fifth vector is a new vector indicating the queuing capacity at all facilities.
- The sixth vector is a new vector indicating the price of service/ goods at all facilities.

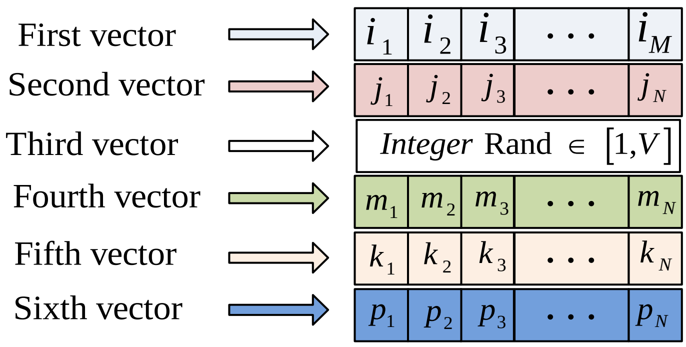


**Figure 3.** Solution representation.

To encode the decision variables, specifically the number of required facilities and their allocation process of customers, the decoding process of the first three vectors is provided in [39]. In this structure, when the cell values of the second vector are zero, the corresponding cell values of the fourth, fifth, and sixth vectors will be zero. No capacity and price can be assigned to the inactive facilities.

After the decoding process, the solutions should be evaluated. Since some constraints are likely to be violated, they are penalized and infeasible solutions are fined using Eq. (16) [48].

$$P(x) = M \times \left[ \left(\frac{g(x)}{b}\right) - 1 \geq 0 \right] \tag{16}$$

where $M$ , *g(x)*, *P(x)* , and *f(x)* represent a big number, the constraint, the penalty function, and the value of chromosome $x$ , respectively. This equation is designed in form of a $g(x) \leq b$ constraint; an additional function is utilized to evaluate the infeasible solutions as follows [42]:

$$F(x) = \begin{cases} f(x) & ; \quad x \in \text{feasible region} \\ f(x) + P(x) & ; \quad x \notin \text{feasible region} \end{cases} \tag{17}$$

***MOVDO main loop***

In the vibration theory, the concept of vibration can be considered to be the oscillation. If the damping is small, it has little influence on the natural frequencies of the system and calculation for the natural frequencies is made on the basis of no damping. In the VDO algorithm, at high amplitudes, the solution scope is larger and a new solution is more likely to be obtained. Therefore, when the amplitude is reduced, the probability of obtaining a new solution decreases; then the trend continues until the amplitude fades away [32, 36].

In order to make an analogy between the vibration damping process and an optimization problem, the states of the oscillation system represent feasible solutions of the optimization problem; the energies of the states correspond to the objective function value computed at those solutions, the minimum energy state corresponds to the optimal solution of the problem, and rapid quenching can be viewed as local optimization. The VDO algorithm starts by generating random solutions in search space. Then, the algorithm parameters, including initial amplitude ($A_0$), maximum number of iterations at each amplitude ($L$), damping coefficient ($\gamma$) and standard deviation ($\sigma$), are initialized. Then, the solutions are evaluated by means of the objective function value (OFV). The algorithm contains two main loops. The first loop generates a solution randomly and then, using neighborhood structure, a new solution is obtained and the best one is selected. However, similar to the simulated annealing (SA) algorithm, the solution with a lower OFV can be selected regarding the Rayleigh distribution function. In fact, the new solution is accepted if *Δ=OFV (New Solution) – OFV (Current Solution)*<0 ([39]). Besides, if Δ>0, then a random number $r$ between (0, 1) is generated. The current solution is selected with regard to the following criterion:

$$r < 1 - \exp(-\frac{A^2}{2\sigma^2}) \tag{18}$$

The second loop adjusts the amplitude, which is used for reducing amplitude at each iteration. The algorithm is stopped when the stopping criterion is met as follows:

$$A_t = A_0 \exp(-\frac{\gamma t}{2}) \tag{19}$$

After a brief illustration of the VDO algorithm, we developed the MOVDO algorithm to handle Pareto-optimal solutions. To do so, we applied two main concepts of multi-objective meta-heuristics to compare solutions; namely, fast non-dominated sorting (FNDS) and crowding distance (CD). In FNDS, to sort the population, each solution should be compared with every other solution in the population to find if it is dominated. First, all chromosomes in the first non-dominated front are found. In this case, $x_i$ is the non-dominated solution within the solution set $\{x_i, x_j\}$; otherwise, it is not. Then, in order to find the chromosomes in the next non-dominated front, the solutions of the previous fronts are disregarded temporarily. This procedure is repeated until all solutions are set into fronts.

After sorting the population, a CD measure is defined to evaluate solution fronts of population in terms of relative density of individual solutions [14]. To this aim, consider $Z$ and $f_k;\ k=1,2$ to be the number of non-dominated solutions in a particular front ($F$) and the objective functions, respectively. Besides, let $d_i$ and $d_j$ be the value of the CD for solutions $i$ and $j$, respectively. Then, CD is obtained using the following steps:

I. Set $d_i$=0 for $i$=1, 2, …, $Z$

II. Sort all objective functions $f_k$; $k$=1,2 in ascending order

III. The CD for boundary solutions in each front ($d_1$ and $d_Z$) are $d_1 = d_Z \rightarrow \infty$

IV. The CDs for $d_j$; $j$=2, 3, …, ($Z$-1) are $d_j = d_j + (f_{k_{j+1}} - f_{k_{j-1}})$

In order to select individuals of the next generation, the crowded tournament selection operator ">" is applied [13] through the following steps:

***Step 1*****:** Choose $n$ individuals in the population randomly.

***Step 2*****:** Non-dominated ranks of each individual should be obtained and the CDs of the solutions having equal non-dominated rank calculated.

***Step 3*****:** The solutions with the least rank are selected. If more than one individual shares the least rank, the individual with the highest CD should be selected.

In other words, the comparison criterion of MOVDO algorithm solutions is considered as follows: If $r_x<r_y$ or ($r_x=r_y$ and $d_x<d_y$) then $r_x>r_y$ where $r_x$ and $r_y$ are the ranks and $d_x$ and $d_y$ are CDs. In this paper, a polynomial neighborhood structure for the selected chromosome is performed.

After performing the aforementioned operators and concepts, the parents and offspring population should be combined to ensure the elitism. On the other hand, the offspring population is combined with the current generation of population and selection is performed to set the individuals of the next generation. Since all previous and current best individuals are added to the population, elitism is ensured. This concept leads to keeping the best individuals from the parent and child population for the next generation. Since the combined population size is naturally greater than the original population size $N$, non-dominating sorting is again performed. In fact, chromosomes with higher ranks are selected and added to the population until the population size becomes $N$. The last front also consists of the population based on the CD. The algorithm stops when a predetermined number of iterations (or any stopping criteria) is reached.



***Evolution process of MOVDO***

The process starts working by initializing the first population of the solution vectors $P_j$. Later, the operators are implemented on $P_j$ to get a new population $Q_j$. The combination of $P_j$ and $Q_j$ creates $R_j$ for the elitism process [14]. Besides, solutions in $R_j$ are categorized in different fronts based on FNDS and CD. In the end, a population of the next iteration $P_{j+1}$ is selected to have a predetermined size. Figure 4 illustrates the evolution process of the proposed MOVDO.

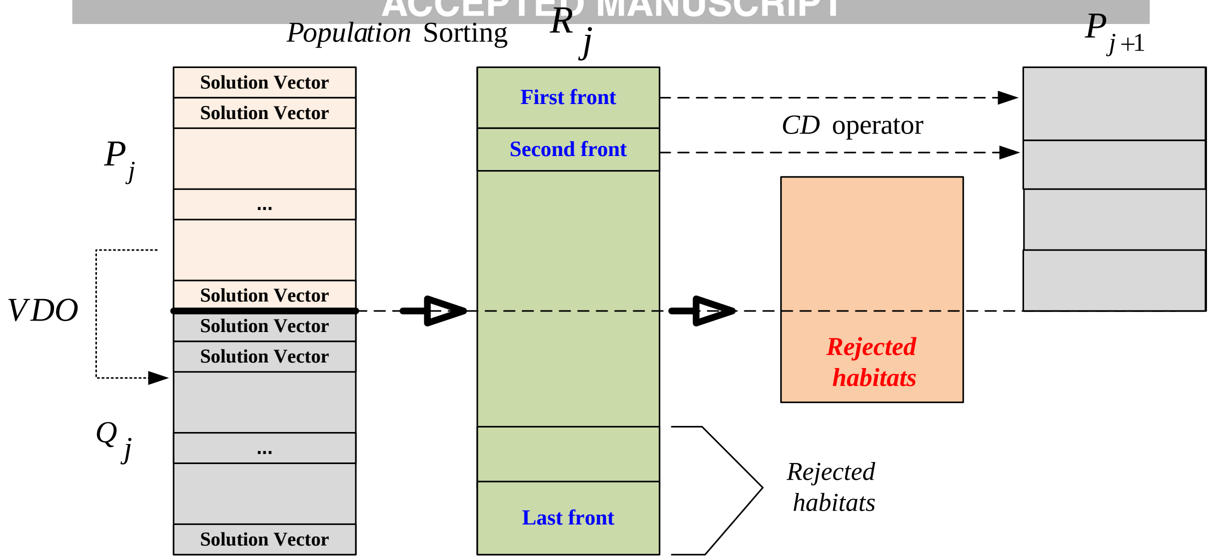


**Figure 4.** MOVDO evolution process.

Figure 5 illustrates the flowchart of MOVDO in which the multi-objective parts are highlighted. In the flowchart, we use Perturb function as representative of the objective function. *PERTURB*(*X*) represents the objective function values of solution *X*. Figure 6 provides Pseudo code of MOVDO.

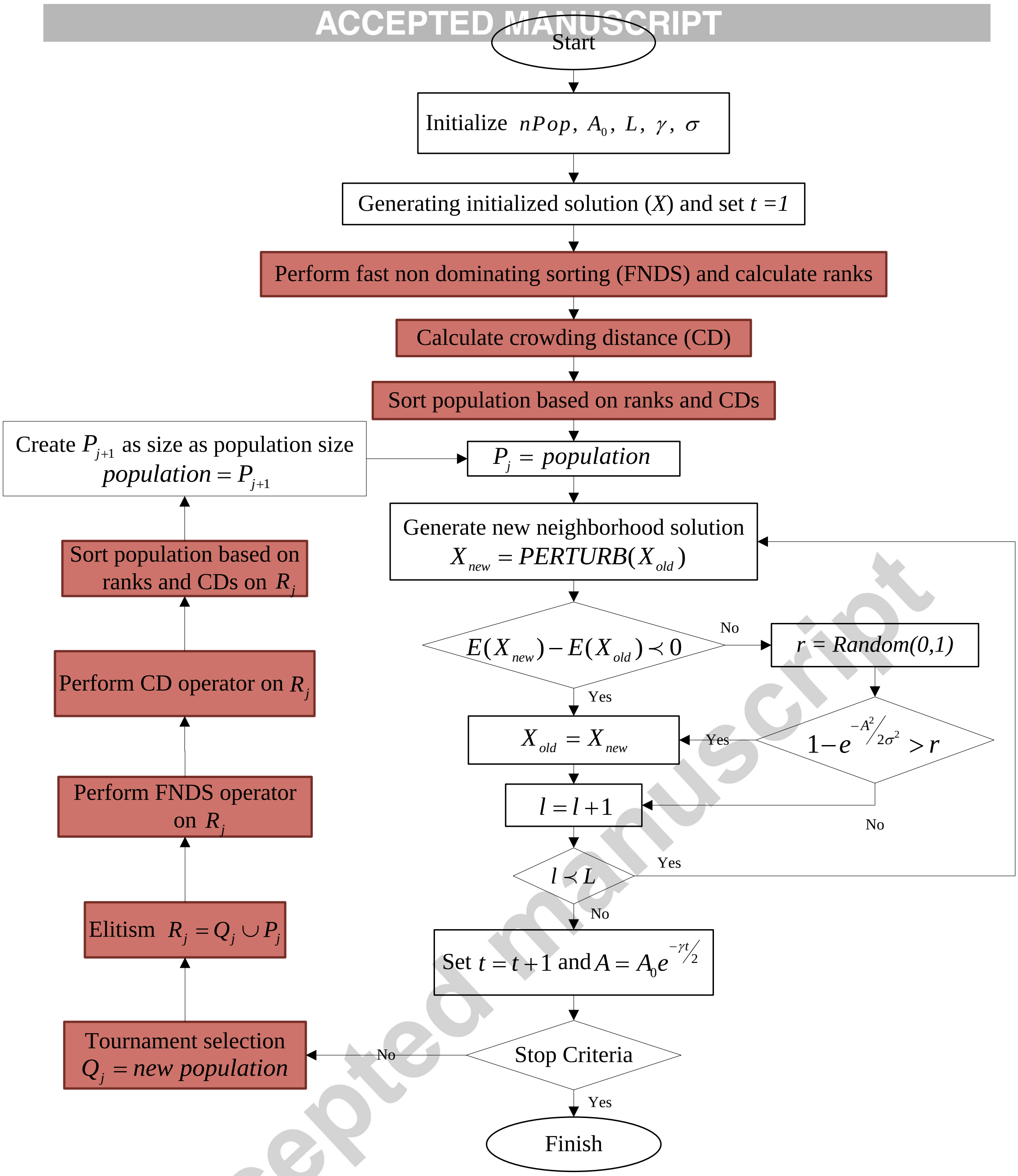


**Figure 5.** Flowchart of the proposed MOVDO.

***Begin;***

***Input***: *nPop* (Population Number), $\gamma$ (Damping Coefficient) and $\sigma$ (Raleigh Distribution Constant);
***Initialize*** ($X$; $A$; $L$ and $t$, $t$=1);
***Evaluate*** *Solutions*

***Perform*** fast non-dominate sorting (FNDS) and calculate ranks
***Calculate*** crowding distance (CD)
***Sort*** population according to ranks and CDs

For $j$=1: *nPop*
$P_j$ = Population
For $i$=1:$L$
$Y$ = PERTURB($X$); {Generate Neighborhood Solution}
$\Delta = E(Y) - E(X)$;
If $\Delta \leq 0$ Or $(1 - e^{-\frac{A^2}{2\sigma^2}} > Random(0,1))$
Then $X = Y$ ; {Accept the Movement if Dominated Final Pareto Solution}
End if

Update ($A$ and $t$ ; $A = A_0 * e^{-\frac{\gamma t}{2}}$, $t = t+1$)
Until (Stop-Criterion)
End for

$Q_j$ = New Population
$R_j = P_j \cup Q_j$
***Perform*** fast non-dominate sorting (FNDS) on $R_j$ and calculate ranks
***Calculate*** crowding distance (CD) of $R_j$
***Sort*** Population according to ranks and CDs on $R_j$
***Create*** $P_{j+1}$ as Size as Population Size (Population= $P_{j+1}$)

End for
***End***

**Figure 6.** MOVDO Pseudo-code.

### *4.3. The NSGA-II*

To demonstrate performance of the proposed MOVDO, a well-developed Pareto-based multi-objective evolutionary algorithm (MOEA) called NSGA-II is applied. The main difference of the NSGA-II with the MOVDO is the evolution process of the algorithm from $P_t$ to $Q_t$. The evolution of MOVDO is based on Figure 4 but in NSGA-II the evolution process of a GA is applied. In order to minimize the impact of algorithms operators in comparing the two algorithms, the neighborhood operator of MOVDO is designed similar to the mutation operator of NSGA-II. Moreover, in NSGA-II, the crossover operator is also performed by uniform crossover operator [25]. The NSGA-II framework is depicted in Figure 7.

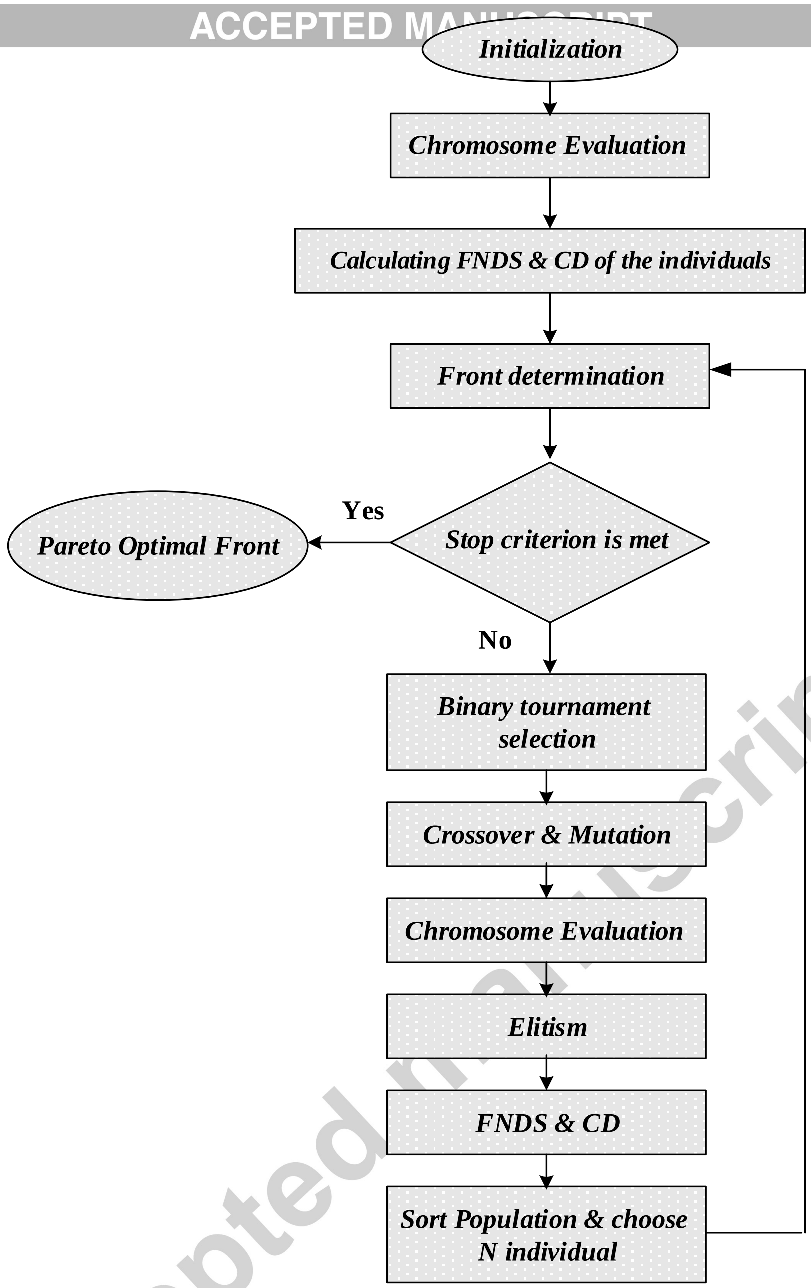


**Figure 7.** Flowchart of NSGA-II.

## 5. Parameters

This section is classified into two subsections to set input parameters of the model and the two algorithms parameter setting.

### *5.1. Input parameters*

Twenty test problems are generated randomly based on literature [39]. These problems are classified according to the number of costumers (*M*), the number of facilities (*N*), and the maximum number of on-duty servers (*V*). Each test problem is run thirty times and the average solution values are computed. The travel distance $d_{ij}$ is calculated as being a proportion of the Euclidean distance among the location of customers and potential facilities, i.e., $d_{ij} \sim Uniform$ [100,500]. The service rate of server $j$ has a uniform distribution function, i.e., $\mu_{ij}$~*Uniform* [100,1000]. Fixed cost of establishing potential facility $j$ follows a uniform distribution, i.e., $e_j$~*Uniform* [1000,6000]. Unit cost of service/goods at facility $j$ has a uniform distribution, i.e., $cs_j$~*Uniform* [100,500]. At each facility, the upper bound for the number of servers at each facility, the upper bound of queuing capacity at each facility and the upper bound of price are set to be 10, 300 and 1000, respectively. Potential number of users in customer node $i$ has a uniform distribution, i.e., $g_i$~*Uniform* [5000,10000]. Price and distance sensitivity coefficients in customer node $i$ have a uniform distribution, i.e., $\alpha_i$ and $\beta_i$~*Uniform* [1,10], respectively.

### *5.2. Algorithm parameter tuning*

Different approaches, such as response surface methodology [37, 39] and the Taguchi method, [35] have already been utilized for algorithm calibration. In this subsection, to tune the parameters of both algorithms, the Taguchi approach is applied. The Taguchi approach exploits orthogonal arrays to manage and adjust experiences in the presence of a group of decision variables or factors [40]. The method attempts to minimize the effect of noise and to obtain the optimal level of signal factors. Since the nature of our response is minimization, the smaller the response the better. The following equation formulates signal-to-noise ratio (*S/N*):

$$S/N = -10 \times \log\left(S(Y^2)/n\right) \tag{20}$$

$Y$ denotes response value, $n$ denotes the number of the orthogonal arrays, and $S(Y)$ is the representative objective function value.

For conducting the method, a criterion is applied [42]. The two main goals are i) appropriate convergence and ii) diversity of Pareto solutions. Among standard multi-objective metrics, computational time and mean ideal distance (MID) are two metrics for modeling the convergence of the algorithms and other metrics are applied for formulating diversity of the Pareto solution. In the proposed metric, diversity and MID (as representatives of their multi-objective feature group) are combined through Eq. (21). The two features of the Pareto-based meta-heuristics are considered simultaneously. Therefore, by using this metric as the response of the Taguchi method, a combination of major signals can be proposed. Thus, we expect to obtain precise outputs. This metric is called the multi-objective coefficient of variation (MOCV) [42]:

$$MOCV = \frac{MID}{Diversity} \tag{21}$$

In order to implement the Taguchi method, the level of each factor is reported in Table 1.

**Table 1.** Algorithm parameter ranges and levels of the factors.

| Multi-objective algorithm | Algorithm parameter | Parameter range | Low (1) | Medium (2) | High (3) |
|---|---|---|---|---|---|
| *NSGA-II* | $nPop_{NSGA\text{-}II}$ | 25-100 | 25 | 50 | 100 |
| | $P_c$ | 0.6-0.9 | 0.6 | 0.8 | 0.9 |
| | $P_m$ | 0.1-0.4 | 0. 1 | 0.25 | 0.4 |
| | *MaxIT* | 100-500 | 100 | 300 | 500 |
| *MOVDO* | $A_0$ | 6-10 | 6 | 8 | 10 |
| | $\sigma$ | 1-2 | 1 | 1.5 | 2 |
| | $\gamma$ | 0.005-0.5 | 0.005 | 0.05 | 0.5 |
| | $L$ | 50-100 | 50 | 75 | 100 |
| | $nPop_{MOVDO}$ | 4-12 | 4 | 8 | 12 |

In each algorithm, three levels (i.e. low, medium and high) are considered for each factor. Then, by using Minitab Software, L9 design is used for NSGA-II and L27 design is exploited for MOVDO. The orthogonal arrays of these designs and our obtained responses are presented in Table 2 (for NSGA-II) and Table 3 (for MOVDO).

**Table 2.** Taguchi procedure for NSGA-II.

| Run order | Algorithm parameters | | | | Proposed response for NSGA-II |
|---|---|---|---|---|---|
| | ***nPop*** | $\boldsymbol{P_c}$ | $\boldsymbol{P_m}$ | ***MaxIT*** | |
| 1 | 1 | 1 | 1 | 1 | 4.9384 |
| 2 | 1 | 2 | 2 | 2 | 8.3948 |
| 3 | 1 | 3 | 3 | 3 | 28.9484 |
| 4 | 2 | 1 | 2 | 3 | 58.2983 |
| 5 | 2 | 2 | 3 | 1 | 93.1287 |
| 6 | 2 | 3 | 1 | 2 | 81.1273 |
| 7 | 3 | 1 | 3 | 2 | 78.8743 |
| 8 | 3 | 2 | 1 | 3 | 69.3492 |
| 9 | 3 | 3 | 2 | 1 | 94.2983 |

**Table 3.** Taguchi procedure for MOVDO.

| Run order | Algorithm parameters | | | | | Proposed response for MOVDO |
|---|---|---|---|---|---|---|
| | $A_0$ | $\sigma$ | $\gamma$ | $L$ | $nPop_{MOVDO}$ | |
| 1 | 1 | 1 | 1 | 1 | 1 | 2.9834 |
| 2 | 1 | 1 | 1 | 1 | 2 | 1.5434 |
| 3 | 1 | 1 | 1 | 1 | 3 | 0.7293 |
| 4 | 1 | 2 | 2 | 2 | 1 | 0.7533 |
| 5 | 1 | 2 | 2 | 2 | 2 | 0.5522 |
| 6 | 1 | 2 | 2 | 2 | 3 | 0.6728 |
| 7 | 1 | 3 | 3 | 3 | 1 | 0.7627 |
| 8 | 1 | 3 | 3 | 3 | 2 | 0.6282 |
| 9 | 1 | 3 | 3 | 3 | 3 | 0.5739 |
| 10 | 2 | 1 | 2 | 3 | 1 | 1.7363 |
| 11 | 2 | 1 | 2 | 3 | 2 | 0.7363 |
| 12 | 2 | 1 | 2 | 3 | 3 | 2.3983 |
| 13 | 2 | 2 | 3 | 1 | 1 | 1.7485 |
| 14 | 2 | 2 | 3 | 1 | 2 | 2.9834 |
| 15 | 2 | 2 | 3 | 1 | 3 | 0.5562 |
| 16 | 2 | 3 | 1 | 2 | 1 | 0.8171 |
| 17 | 2 | 3 | 1 | 2 | 2 | 4.3921 |
| 18 | 2 | 3 | 1 | 2 | 3 | 0.6843 |
| 19 | 3 | 1 | 3 | 2 | 1 | 0.7273 |
| 20 | 3 | 1 | 3 | 2 | 2 | 0.4202 |
| 21 | 3 | 1 | 3 | 2 | 3 | 1.9384 |
| 22 | 3 | 2 | 1 | 3 | 1 | 0.9312 |
| 23 | 3 | 2 | 1 | 3 | 2 | 0.9343 |
| 24 | 3 | 2 | 1 | 3 | 3 | 0.9384 |
| 25 | 3 | 3 | 2 | 1 | 1 | 5.9743 |
| 26 | 3 | 3 | 2 | 1 | 2 | 0.7832 |
| 27 | 3 | 3 | 2 | 1 | 3 | 0.8584 |

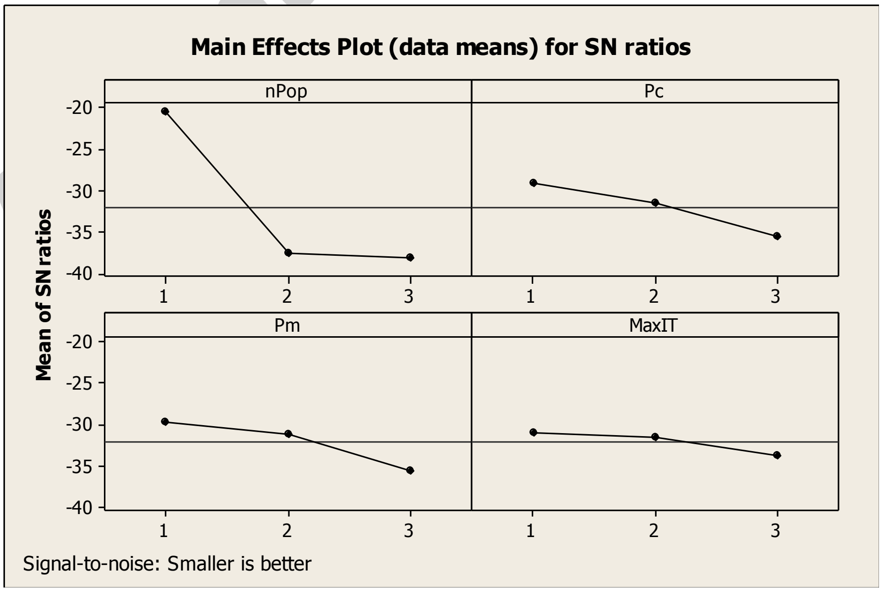


**Figure 8.** Outputs of Taguchi ratio for NSGA-II.

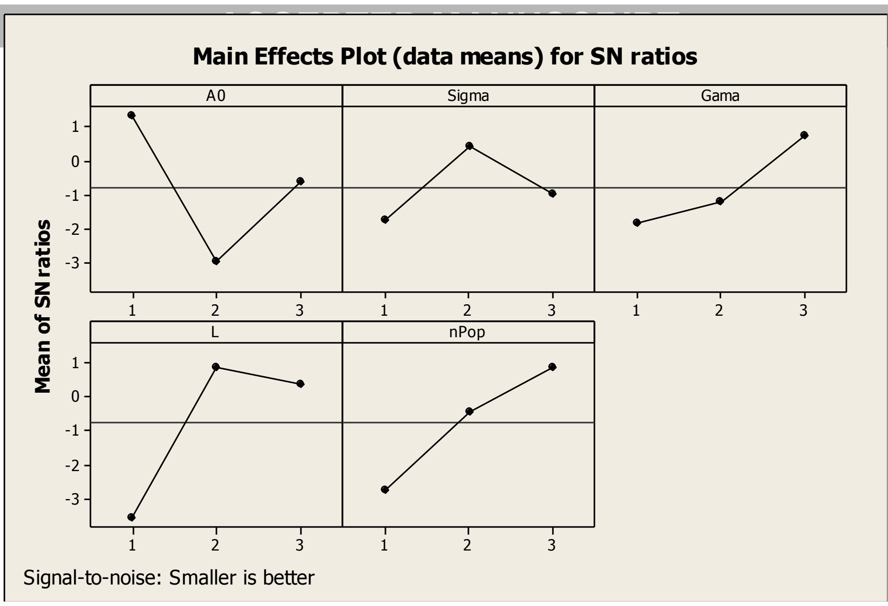


**Figure 9.** Outputs of Taguchi ratio for MOVDO.

For each algorithm, the effect plots for *S/N* ratio are presented in Figures 8 and 9. By using these results for each algorithm, the optimum values of all parameters are obtained and reported in Table 4.

**Table 4.** The calibrated values of both algorithms.

| Multi-objective algorithms | Algorithm parameters | Optimum value |
|---|---|---|
| *NSGA-II* | $nPop_{NSGA\text{-}II}$ | 25 |
| | $P_c$ | 0.6 |
| | $P_m$ | 0.1 |
| | *MaxIT* | 100 |
| *MOVDO* | $A_0$ | 6 |
| | $\Sigma$ | 1.5 |
| | $\gamma$ | 0.5 |
| | $L$ | 75 |
| | $nPop_{MOVDO}$ | 12 |

## 6. Computational results and analysis

In order to evaluate performance of each algorithm, two features should be assessed: efficiency and effectiveness. In a single objective algorithm, the objective function represents effectiveness and the computational time is a proxy of efficiency. However, in a multi-objective algorithm for assessing the efficiency and effectiveness, different metrics can be used to have a comprehensive picture of the algorithm's capability. Given this notion, in Pareto-based multi-objective algorithms, we adopt the two following strategies to fulfill the above-mentioned features: 1) convergence, and 2) diversity. In this paper, computational time is also used as a metric. Computational time can still be used as a good metric for evaluating effectiveness.

In summary, the performance of the proposed MOVDO algorithm is evaluated using five multi-objective performance metrics as follows [51, 52]:

- ***Diversity*** computes the extension of the Pareto front.
- ***Spacing*** computes the standard deviation of the solutions distances in the front.
- ***MID*** computes the convergence rate of Pareto fronts to a certain point (0, 0).
- ***The number of Pareto solutions (NOS)*** enumerates the number of the Pareto solutions in optimal front.
- The ***run time*** shows executing time of the algorithm to attain best solutions.

While in terms of the diversity and NOS metrics, larger values are desirable, for spacing, *MID*, and CPU time, smaller values are desired. Table 5 reports the computational results of implementing the algorithms on the 20 test problems. In the table, the test problems in which the algorithm cannot find Pareto front in the reported time are identified by "NAN". In order to code the proposed meta-heuristic algorithms, MATLAB Software [30] has been exploited and the programs have been implemented on a Core i7, 2 GHz laptop with 8 GB RAM.

**Table 5.** Multi-objective metrics computed by MOVDO and NSGA-II.

| Problem No. | *M* | *N* | *V* | Diversity | | NOS | | MID | | Spacing | | Time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* |
| 1 | 16 | 7 | 5 | 2.11E+08 | 1.25E+07 | 22 | 8 | 2.27E+07 | 2.42E+07 | 1.95E+08 | 1.25E+07 | 15.13 | 10.84 |
| 2 | 20 | 9 | 6 | 1.22E+09 | 2.66E+07 | 19 | 9 | 4.48E+08 | 3.55E+08 | 2.75E+08 | 3.26E+05 | 25.43 | 12.12 |
| 3 | 35 | 12 | 9 | 2.23E+09 | 4.22E+07 | 23 | 5 | 6.55E+08 | 4.92E+09 | 4.35E+08 | 7.64E+06 | 39.31 | 19.75 |
| 4 | 42 | 15 | 11 | 3.23E+07 | 3.77E+07 | 20 | 11 | 3.70E+08 | 2.15E+09 | 3.65E+06 | 4.71E+06 | 44.12 | 18.65 |
| 5 | 57 | 17 | 12 | 4.54E+09 | 7.67E+08 | 25 | 6 | 4.22E+08 | 9.65E+09 | 3.80E+08 | 4.36E+06 | 46.31 | 21.65 |
| 6 | 62 | 21 | 14 | 1.44E+09 | 3.60E+09 | 23 | 8 | 2.46E+08 | 6.71E+09 | 4.68E+08 | 5.75E+07 | 45.64 | 22.35 |
| 7 | 77 | 25 | 18 | 4.31E+10 | 3.66E+09 | 24 | 9 | 4.54E+08 | 8.92E+09 | 3.48E+10 | 5.49E+06 | 49.35 | 20.45 |
| 8 | 81 | 30 | 20 | 2.87E+11 | 8.20E+08 | 23 | 6 | 5.65E+09 | 6.72E+09 | 3.49E+10 | 6.92E+06 | 51.43 | 29.53 |
| 9 | 90 | 38 | 22 | 3.00E+08 | 2.19E+09 | 22 | 8 | 1.57E+08 | 4.45E+08 | 5.67E+06 | 5.69E+06 | 54.57 | 28.53 |
| 10 | 105 | 42 | 25 | 3.71E+08 | 3.14E+09 | 23 | 10 | 4.55E+08 | 3.61E+08 | 6.55E+07 | 3.59E+07 | 53.43 | 31.84 |
| 11 | 128 | 45 | 25 | 2.17E+08 | 2.12E+08 | 25 | 8 | 3.68E+08 | 2.21E+09 | 4.54E+08 | 4.58E+07 | 58.45 | 29.13 |
| 12 | 147 | 53 | 30 | 1.69E+09 | 2.33E+08 | 23 | 10 | 4.36E+08 | 3.81E+08 | 6.76E+08 | 3.47E+07 | 61.19 | 26.86 |
| 13 | 175 | 68 | 38 | 2.15E+08 | 2.91E+09 | 20 | 7 | 8.79E+08 | 5.73E+08 | 5.88E+08 | 2.38E+07 | 84.12 | 35.55 |
| 14 | 196 | 70 | 45 | 2.95E+10 | 4.90E+10 | 24 | 9 | 2.16E+09 | 2.46E+09 | 5.73E+08 | 3.74E+07 | 86.83 | 36.68 |
| 15 | 220 | 88 | 52 | 3.35E+10 | 4.12E+10 | 25 | 9 | 4.52E+09 | 6.44E+08 | 6.73E+09 | 2.35E+07 | 101.13 | 59.74 |
| 16 | 250 | 95 | 70 | 3.88E+09 | 9.52E+09 | 24 | 12 | 3.46E+09 | 3.45E+09 | 2.87E+08 | 6.54E+07 | 119.43 | 65.35 |
| 17 | 350 | 105 | 75 | 3.73E+10 | 1.99E+10 | 25 | 7 | 4.52E+09 | 5.70E+10 | 2.32E+09 | 3.56E+09 | 121.13 | 63.32 |
| 18 | 800 | 190 | 140 | 2.16E+12 | 3.66E+11 | 21 | 11 | 6.54E+10 | 4.70E+09 | 2.33E+12 | 3.48E+10 | 118.64 | 69.44 |
| 19 | 2200 | 750 | 570 | NAN | 2.82E+10 | NAN | 8 | NAN | 7.74E+09 | NAN | 2.47E+10 | 228.74 | 110.84 |
| 20 | 3500 | 1100 | 700 | NAN | 1.70E+10 | NAN | 12 | NAN | 4.19E+10 | NAN | 3.71E+10 | 331.77 | 189.63 |

In order to demonstrate performance of the proposed MOVDO, various analyses are carried out as follows.

### *6.1. Statistical analysis*

In order to show the difference of both algorithms in terms of five metrics, the algorithms are statistically analyzed according to obtained solutions via analysis of variance (ANOVA) tests [34]. The procedure of ANOVA including F-test value and also P-value on each metric is summarized in Table 6. To visualize statistical outputs for the cases, *NOS* and computational time metrics are drawn in Figures 10 and 11. In Figure 12-15, the algorithms are also compared schematically for all test problems.

**Table 6.** The results of ANOVA test.

| Metrics | Source | DF | SS | MS | F-test | P-value | Test results |
|---|---|---|---|---|---|---|---|
| ***Diversity*** | Algorithms<br>Error<br>Total | 1<br>36<br>37 | 1.30529E+23<br>4.50046E+24<br>4.63099E+24 | 1.30529E+23<br>1.25013E+23 | 1.04 | 0.314 | Null hypothesis is not rejected |
| ***NOS*** | Algorithms<br>Error<br>Total | 1<br>36<br>37 | 1905.79<br>131.05<br>2036.84 | 1905.79<br>3.64 | 523.53 | 0.000 | Null hypothesis is rejected |
| ***MID*** | Algorithms<br>Error<br>Total | 1<br>36<br>37 | 8.69201E+19<br>8.01466E+21<br>8.10158E+21 | 8.69201E+19<br>2.22630E+20 | 0.39 | 0.536 | Null hypothesis is not rejected |
| ***Spacing*** | Algorithms<br>Error<br>Total | 1<br>36<br>37 | 1.57744E+23<br>5.11057E+24<br>5.26831E+24 | 1.57744E+23<br>1.41960E+23 | 1.11 | 0.299 | Null hypothesis is not rejected |
| ***Time*** | Algorithms<br>Error<br>Total | 1<br>38<br>39 | 17385<br>139687<br>157071 | 17385<br>3676 | 4.73 | 0.036 | Null hypothesis is rejected |

The results are analyzed in a 95% confidence level. Table 6 shows that when the metrics are statistically compared, the algorithms have significant differences in terms of *NOS* and CPU Time. According to Figure 11, the algorithms perform similarly in terms of diversity, spacing, and *MID* metrics. Moreover, in large-sized test problems (i.e. test problems 19 and 20), NSGA-II cannot find Pareto front but MOVDO can. This fact demonstrates better performance of the proposed MOVDO in large-sized problems. In order to investigate the performance of MOVDO in large-sized problems, a separate analysis is performed.



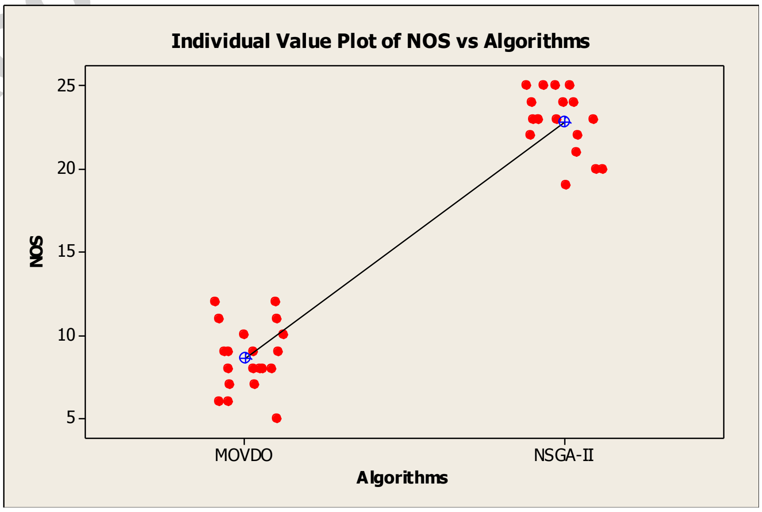


**Figure 10.** Individual value plot of NOS metric.

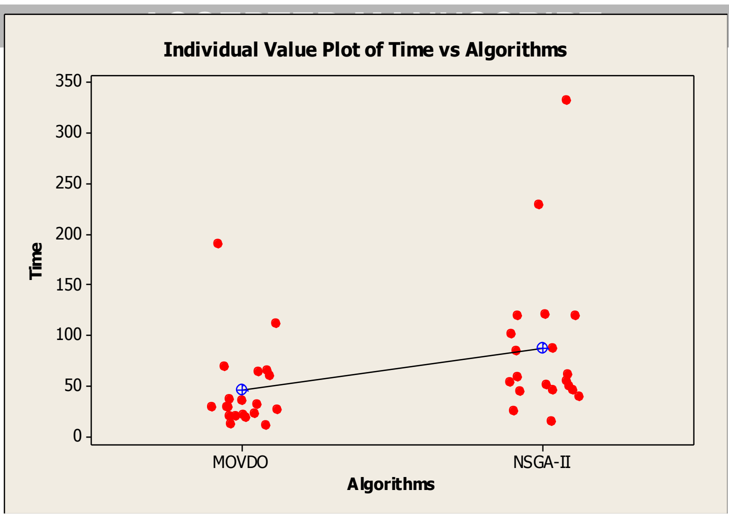


**Figure 11.** Individual value plot of computational time metric.

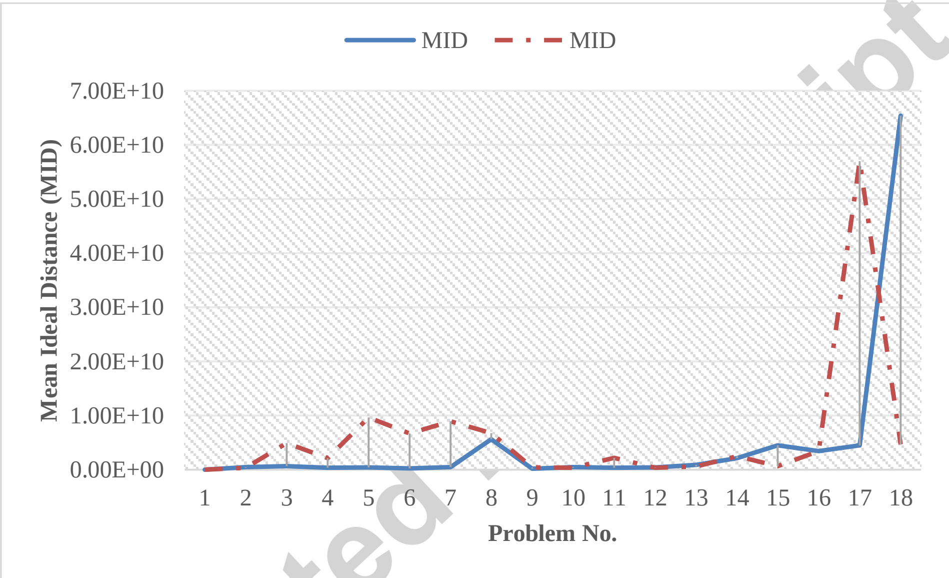


**Figure 12.** Graphical comparisons of MID metric for both algorithms on all test problems.

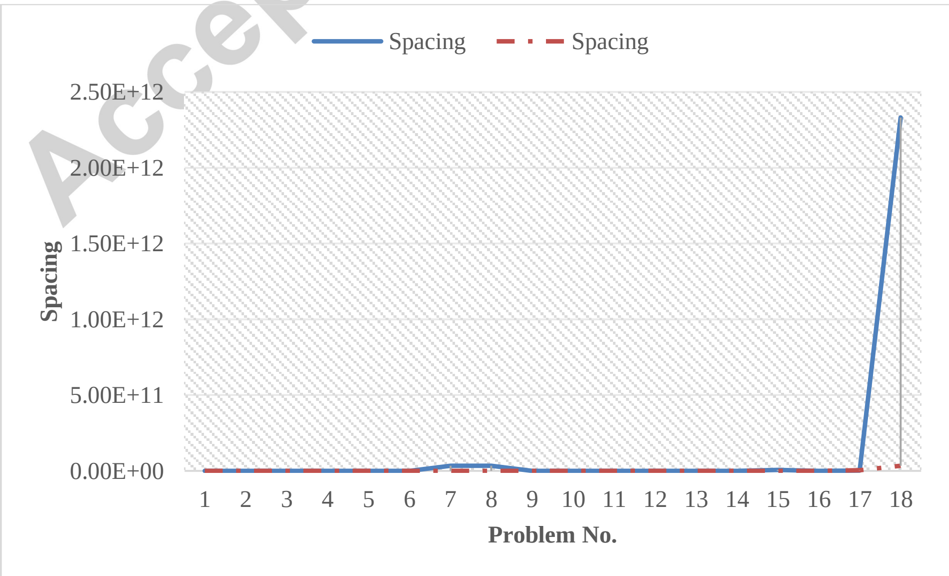


**Figure 13.** Graphical comparisons of Spacing metric for both algorithms on all test problems.

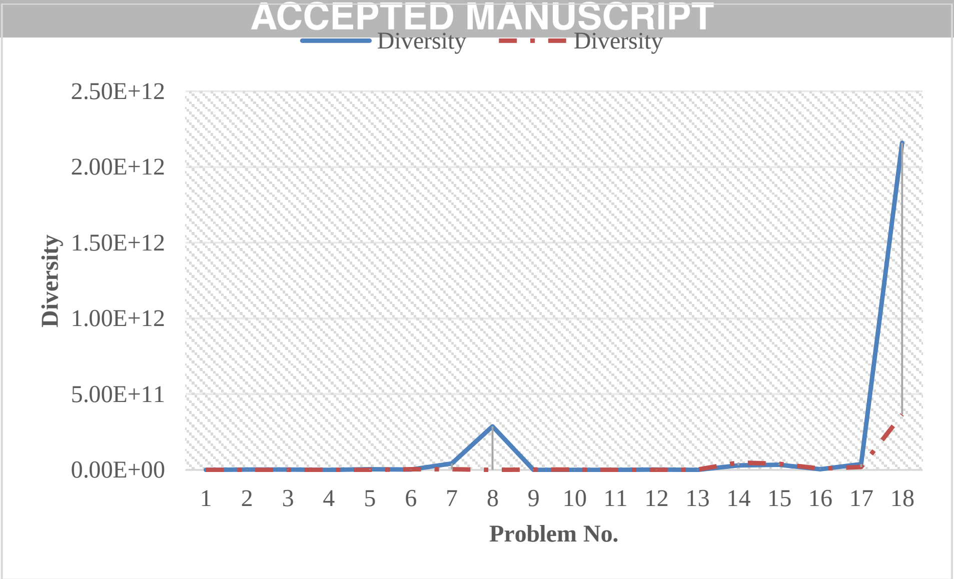


**Figure 14.** Graphical comparisons of Diversity metric for both algorithms on all test problems.

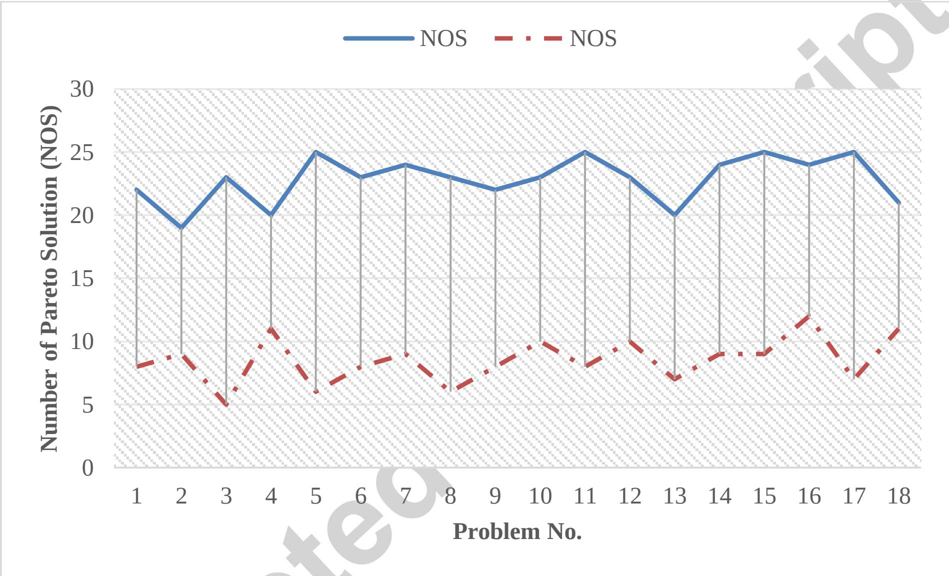


**Figure 15.** Graphical comparisons of NOS metric for both algorithms on all test problems.

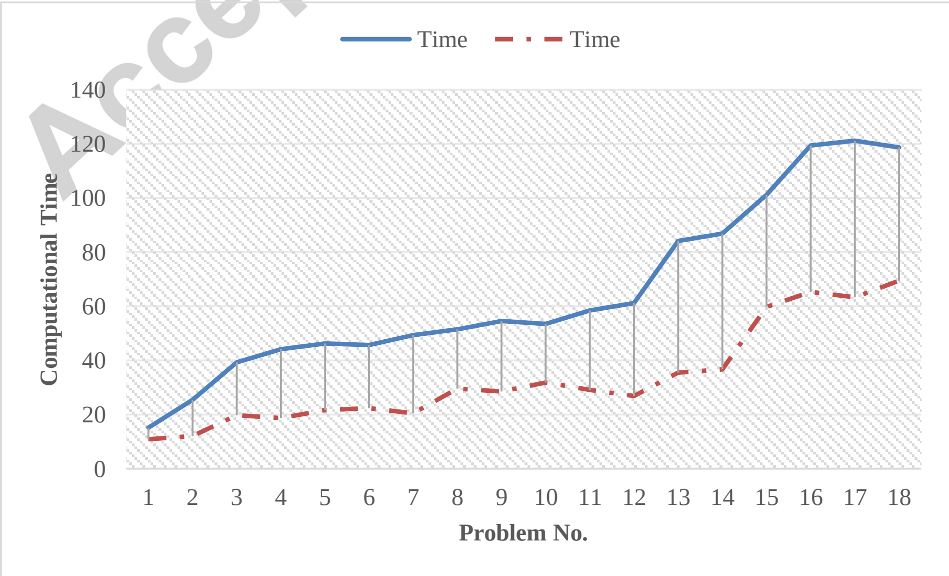


**Figure 16.** Graphical comparisons of Time metric for both algorithms on all test problems.

### *6.2. Optimized solution*

To validate the problem representations, for test problem no. 1, we have reported the optimized solutions obtained by the proposed MOVDO as follows:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \qquad m = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \\ 2 \end{bmatrix} \qquad k = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 5 \\ 4 \\ 0 \\ 6 \end{bmatrix}$$

### *6.3. Relative Performance of MOVDO and NSGA-II*

In order to compare the results of NSGA-II versus the proposed MOVDO with more visibility, we have analyzed non-dominated solutions obtained by both algorithms. Table 7 represents the results of NSGA-II that are dominated by the results of MOVDO in test problem 13. For example, result no. 2 of NSGA-II is dominated by results nos. 1 and 2 of MOVDO. Therefore, six Pareto solutions of NSGA-II algorithm are dominated by MOVDO and no MOVDO is dominated by NSGA-II.

**Table 7.** Pareto solutions of problem no. 13 along with non-domination analysis.

| Pareto Solution Number | NSGA-II | | MOVDO | | The solutions of MOVDO dominate each solution |
|---|---|---|---|---|---|
| | *Objective 1* | *Objective 2* | *Objective 1* | *Objective 2* | |
| 1 | 6.45E+06 | 4.17E+04 | 8.94E+07 | 5.20E+04 | - |
| 2 | 1.98E+07 | 4.90E+05 | 6.55E+08 | 1.42E+05 | 1 and 2 |
| 3 | 9.85E+07 | 9.85E+05 | 8.85E+08 | 5.17E+05 | 2, 3, 4, and 5 |
| 4 | 2.45E+08 | 6.85E+06 | 9.88E+08 | 6.84E+05 | 2, 3, 4, 5 and 6 |
| 5 | 4.19E+09 | 7.97E+06 | 3.21E+09 | 6.95E+05 | 6 |
| 6 | 8.52E+09 | 2.56E+07 | 6.58E+09 | 9.85E+05 | 7 |
| 7 | 8.69E+09 | 3.52E+07 | 9.45E+09 | 9.90E+06 | 7 |
| 8 | 9.75E+09 | 8.65E+07 | | | - |
| 9 | 5.70E+10 | 8.86E+07 | | | - |
| 10 | 9.87E+10 | 9.18E+07 | | | - |
| 11 | 9.89E+10 | 1.65E+08 | | | - |
| 12 | 1.69E+11 | 1.69E+08 | | | - |
| 13 | 2.85E+11 | 6.58E+08 | | | - |
| 14 | 5.51E+11 | 8.47E+08 | | | - |
| 15 | 6.48E+11 | 9.65E+08 | | | - |
| 16 | 8.79E+11 | 9.85E+08 | | | - |
| 17 | 5.86E+12 | 1.65E+09 | | | - |
| 18 | 8.85E+12 | 6.58E+09 | | | - |
| 19 | 8.95E+12 | 6.77E+09 | | | - |
| 20 | 9.13E+12 | 6.85E+09 | | | - |

In order to investigate the performance of MOVDO in large-sized problems with further analysis, 10 large-sized problems are tested and reported. The computational results of these problems, in terms of the five multi-objective metrics, are summarized in Table 8. As can be observed, NSGA-II is unable to find non-dominated solutions in large-scale problems. Generally, for good performance of each algorithm, two features should be applied, (i.e., efficiency and effectiveness). As results show, in large-sized problems, NSGA-II has a lower efficiency. Therefore, MOVDO dominates NSGA-II in large-scale problems.

**Table 8.** Multi-objective metrics computed by MOVDO and NSGA-II.

| Large Scale Problem No. | *M* | *N* | *V* | Diversity | | NOS | | MID | | Spacing | | Time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* | *NSGA-II* | *MOVDO* |
| LS-01 | 4000 | 1400 | 850 | NAN | 2.22E+10 | NAN | 14 | NAN | 3.52E+10 | NAN | 3.66E+10 | 480.98 | 253.21 |
| LS-02 | 4600 | 2200 | 1200 | NAN | 3.75E+10 | NAN | 12 | NAN | 2.55E+09 | NAN | 3.77E+10 | 592.84 | 293.77 |
| LS-03 | 5200 | 3000 | 1800 | NAN | 4.32E+11 | NAN | 13 | NAN | 4.66E+12 | NAN | 7.22E+10 | 793.51 | 398.74 |
| LS-04 | 6300 | 3600 | 2800 | NAN | 5.65E+10 | NAN | 9 | NAN | 3.81E+09 | NAN | 4.38E+10 | 952.55 | 520.33 |
| LS-05 | 7000 | 4500 | 3600 | NAN | 9.22E+12 | NAN | 9 | NAN | 6.73E+11 | NAN | 6.71E+10 | 1963.27 | 963.29 |
| LS-06 | 8800 | 6000 | 4900 | NAN | 5.72E+10 | NAN | 8 | NAN | 5.79E+11 | NAN | 4.22E+10 | 2548.59 | 1366.41 |
| LS-07 | 9400 | 7700 | 6200 | NAN | 3.22E+11 | NAN | 7 | NAN | 5.21E+10 | NAN | 9.28E+10 | 3669.88 | 1932.53 |
| LS-08 | 12000 | 9500 | 7800 | NAN | 9.66E+11 | NAN | 12 | NAN | 9.37E+10 | NAN | 1.22E+10 | 4902.78 | 2530.69 |
| LS-09 | 15000 | 12200 | 10500 | NAN | 8.50E+10 | NAN | 10 | NAN | 5.74E+10 | NAN | 5.39E+10 | 8935.22 | 3726.10 |
| LS-10 | 20000 | 16800 | 14500 | NAN | 1.32E+11 | NAN | 11 | NAN | 3.11E+11 | NAN | 2.24E+10 | 12632.88 | 6358.09 |

Figure 13 plots MID vs. time, illustrating the multi-objective evolution of the MOVDO algorithm over time. This figure shows that MOVDO has a better convergence performance in MID. Therefore, it can be shown that in spite of all the above- mentioned differences, MOVDO has a significantly greater capability. To sum up, in small and medium-sized problems, NSGA-II shows better performance for those metrics which are related to a diversity feature, while MOVDO performs better on convergence-based metrics. In large-scale problems, MOVDO can be considered a well-developed algorithm to find Pareto solutions in multi-objective optimization.

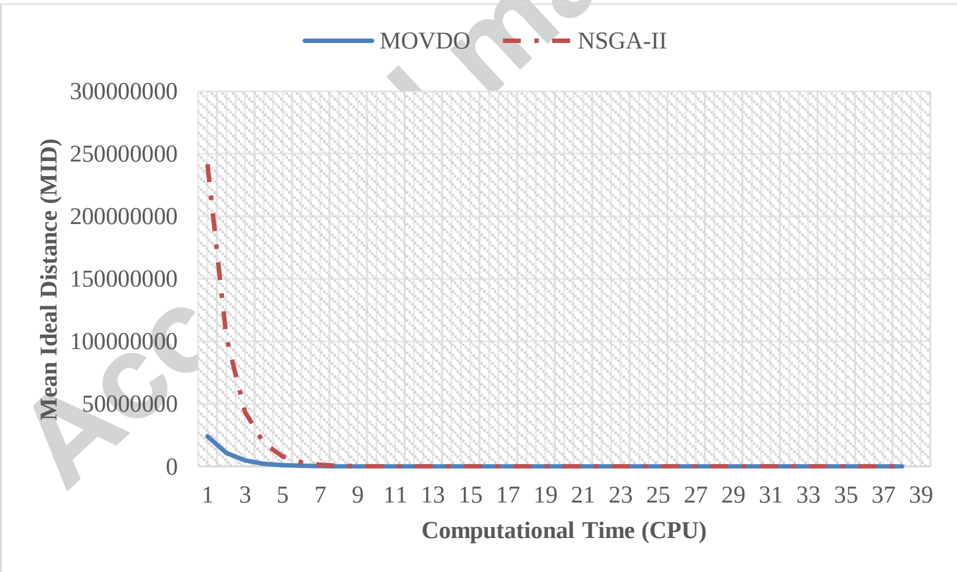


**Figure 13.** Multi-objective convergence plot of MOVDO and NSGA-II

## 7. Conclusion

In this paper, a bi-objective FLP with multiple servers, pricing policies and immobile servers was formulated. Since FLPs are basically NP-hard, a Pareto-based meta-heuristic called MOVDO was developed to solve the model. In the bi-objective model, we maximize total profit of the whole system and simultaneously minimize the sum of waiting time in queues.The results are compared with NSGA-II as well-developed, multi-objective, evolutionary optimization algorithms. To do so, first, the algorithms were tuned by means of the Taguchi method. Then, MOVDO was statistically tested over 20 test problems using five metrics. The results showed that the performance of MOVDO with respect to diversity, spacing, and MID is similar to the performance of NSGA-II. Moreover, MOVDO significantly performs better than NSGA-II in terms of CPU time and it is dominated by NSGA-II in terms of the number of Pareto solutions. According to the computational analysis, the proposed MOVDO is able to generate well-distributed Pareto optimal solutions for multi-objective optimization problems especially in large-sized problems. As future research, multi-layer services with multiple objectives in QFLPs could be modeled. Moreover, the proposed MOVDO could be applied in different fields of multi-objective optimization problems to explore the effectiveness of the technique against problems which were previously solved using NSGA-II.

## ACKNOWLEDGEMENTS

The authors would like to thank the editor-in-chief, editor and anonymous referees for their valuable comments which helped them improve the paper significantly.

## REFERENCES

[1] Abouee-Mehrizi H, Babri S, Berman O, Shavandi H. Optimizing capacity, pricing and location decisions on a congested network with balking, *Mathematical Methods of Operations Research*; 74(2), (2011) 233-255.

[2] Basarcil E, Karaesmen F, Lerzan Örmeci E. Dynamic pricing and scheduling in a multi-class single-server queueing system, *Queueing Systems*; 67, (2011) 305-331.

[3] Berman O, Larson R. Optimal 2-facility network districting in the presence of queuing. *Transportation Science*;19, (1985) 261-277.

[4] Berman O, Drezner Z. The multiple server location problem. *Journal of the Operational Research Society*; 58, (2007) 91-99.

[5] Berman O, Tong D, Krass D. Pricing, location and capacity planning with elastic demand and congestion; Working paper, University of Toronto, (2010).

[6] Berman O, Tong D, Krass D. Pricing, location and capacity planning with equilibrium driven demand and congestion; Working paper, University of Toronto, (2010).

[7] Bhattacharya R, Bandyopadhyay S. Solving conflicting bi-objective facility location problem by NSGA-II evolutionary algorithm, *The International Journal of Advanced Manufacturing Technology*; 51, (2010) 397-414.

[8] Boffey B, Galvao R, Espejo L. A review of congestion models in the location of facilities with immobile servers. *European Journal of Operational Research*; 178, (2007) 643-662.

[9] Boloori Arabania A, Farahani RZ. Facility location dynamics: An overview of classifications and applications. *Computers & Industrial Engineering*; 62, (2012) 408-420.

[10] Cadenas JM, Canós MJ, Garrido MC, Ivorra C, Liern V. Soft-computing based heuristics for location on networks: The p-median problem. *Applied Soft Computing*; 11(2) (2011) 1540-1547.

[11] Chambari AH, Rahmaty SH, Hajipour V, Karimi A. A bi-objective model for location-allocation problem within queuing framework. *World Academy of Science. Engineering and Technology*; 78, (2011) 138-145.

[12] Chambari A, Rahmati SHR, Najafi AA, Karimi A. A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies. *Computers & Industrial Engineering*; 63, (2012) 109-119.

[13] Coello CA, Lamont GB, Van Veldhuizen DA. *Evolutionary algorithms for solving multiobjective problems*. 2nd ed., Springer, Berlin (2007).

[14] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*; 6, (2002) 182-197.

[15] Diakova Z, Kochetov Yu, A double VNS heuristic for the facility location and pricing problem, *Electronic Notes in Discrete Mathematics*; 39, (2012) 29-34.

[16] Farahani RZ, SteadieSeifi M, Asgari N. Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*; 34, (2010) 1689-1709.

[17] Farahani RZ, Hekmatfar M. Facility location: Concepts, models, algorithms and case studies. *Contributions to management science* (2009). Heidelberg: Physica- Verlag. doi:10.1007/978-3-7908-2151-2 11.

[18] Farahani RZ, Asgari N, Heidari N, Hosseinini M, Goh M. Covering problems in facility location: A review. *Computers & Industrial Engineering*; 62, (2012) 368-407.

[19] Fernández P, Pelegrín B, GarcíaPérez M.D, Peeters P.H, A discrete long-term location-price problem under the assumption of discriminatory pricing: Formulations and parametric analysis, *European Journal of Operational Research*; 179(3), (2007) 1050-1062.

[20] Gross D, Harris CM. *Fundamental of queuing theory*, 3rd ed., Wiley Interscience, New York, (1988).

[21] Hajipour V, Pasandideh SHR. A new multi-objective facility location model within batch arrival queuing framework. *World Academy of Science, Engineering and Technology*; 78, (2011) 1665-1673.

[22] Hajipour V, Pasandideh SHR. Proposing an adaptive particle swarm optimization for a novel bi-objective queuing facility location model. *Economic Computation and Economic Cybernetics Studies and Research*; 3, (2012) 1-18.

[23] Hajipour, V., Mehdizadeh, E., Tavakkoli-Moghaddam, R. A novel Pareto-based Multi-objective Vibration Damping Optimization Algorithm to Solve Multi-objective Optimization Problems, *Scientia Iranica: Transaction E*; 21(6), (2014) 2368-2378.

[24] Hakimi SL. Optimal locations of switching centers and the absolute centers and medians of a graph. *Operation Research*; 12, (1964) 450- 459

[25] Haupt RL, Haupt SE. Practical genetic algorithms, 2nd Ed., John Wiley & Sons, (2004).

[26] Hodgson MJ, Berman O. A billboard location model. *Geographical and Environmental Modeling*; 1, (1997) 25-43.

[27] Love RL, Morris JG, Wesolowsky, GO. Facility location: Models and methods. *North-Holland Publishing Company, New York*, 1988. (Cited by Francis RL, Megginis LF, White JA. Facility layout and location: An analytical approach (2nd ed.). *Englewood Cliffs, NJ: Prentice-Hall*, (1992).

[28] Lüer-Villagra A, Marianov V, A competitive hub location and pricing problem, *European Journal of Operational Research*; 231, (2013) 734-744.

[29] Marianov V., ReVelle C. Siting emergency services in facility Location: A survey of applications and methods. *Springer Series in Operations Research*, (1995).

[30] MATLAB Version 7.10.0.499 (R2010a). The MathWorks, Inc. Protected by U.S. and international patents, (2010).

[31] Melo MT, Nickel S, Saldanha-da-Gama F. Facility location and supply chain management - A review. *European Journal of Operational Research*; 196, (2009) 401-412.

[32] Mehdizadeh E, Tavakkoli-Moghaddam R. Vibration damping optimization algorithm for an identical parallel machine scheduling problem. Proc. of the 2nd Int. Conf. of Iranian Operations Research Society, Babolsar, Iran, (2009) 20-22,.

[33] Mehdizadeh E, Tavarroth MR, Hajipour V. A new hybrid algorithm to optimize stochastic-fuzzy capacitated multi-facility location-allocation problem, *Journal of Optimization in Industrial Engineering*; 7, (2011) 71-80.

[34] Montgomery, D.C. and Runger, G.C., Applied Statistics and Probability for Engineer, 3rd ed., (2003) (Wiley: New York).

[35] Mousavi SM, Niaki STA, Mehdizadeh E, Tavarroth MR. The capacitated multi-facility location-allocation problem with probabilistic customer location and demand: two hybrid meta-heuristic algorithms. *International Journal of Systems Science*; 44(10), (2013) 1897-1912.

[36] Mousavi M, Hajipour V, Niaki STA, Aalikar N. Optimizing Multi-item Multi-period Inventory Control System with Discounted Cash Flow and Inflation Approaches: Two Calibrated Meta-heuristic Algorithms, *Applied Mathematical Modelling*; 37(4), (2013) 2241-2256.

[37] Najafi AA, Niaki STA, Shahsavar A. A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations. *Computer and Operations Research*; 36, (2009) 2994-3001.

[38] Pasandideh SHR, Niaki STA. Genetic application in a facility location problem with random demand within queuing framework. *Journal of Intelligent Manufacturing*; 23(3), (2012) 651-659.

[39] Pasandideh SHR, Niaki STA, Hajipour V. A multi-objective facility location model with batch arrivals: two parameter-tuned meta-heuristic algorithms. *Journal of Intelligent Manufacturing*; 24, (2013) 331-348.

[40] Peace GS. Taguchi Methods. Addison-Wesley Publishing Company, (1993).

[41] Ray S, Jewkes EM. Customer lead time management when both demand and price are lead time sensitive. *European Journal of Operational Research*; 153(3), (2004) 769-781.

[42] Rahmati S.H.A, Hajipour V, Niaki S.T.A, A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem, *Applied Soft Computing*; 13, (2013) 1728-1740.

[43] Syam SS. A multiple server location-allocation model for service system design. *Computers and Operations Research*; 35, (2008) 2248-2265.

[44] Teimoury E, Modarres M, Kazeruni Monfared A, Fathi M. Price, delivery time, and capacity decisions in an M/M/1 make-to-order/service system with segmented market, *The International Journal of Advanced Manufacturing Technology*; 57, (2011) 235-244.

[45] Toregas C, Swain R, ReVelle C, Bergman L. The location of emergency service facilities. *Operations Research*; 19, (1971) 1363-1373.

[46] Wang Q, Batta R, Rump C. Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of Operations Research*; 111, (2002)17-34.

[47] Wang Q, Batta R. Bhadury J, Rump C. Budget constrained location problem with opening and closing of facilities, *Computers & Operations Research*; 30, (2003) 2047-2069.

[48] Yeniay O, Ankare B. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Application*; 10, (2005) 45-56.

[49] Yoon SH, Lewis ME. Optimal pricing and admission control in a queuing system with periodically varying parameters. *Queueing Systems*; 47, (2004) 177-199.

[50] Zarrinpoor N, Seifbarghy M. A competitive location model to obtain a specific market share while ranking facilities by shorter travel time. *The International Journal of Advanced Manufacturing Technology*; 55, (2011) 807-816.

[51] Zitzler E, Thiele L, Multi-objective optimization using evolutionary algorithms a comparative case study. In A. E. Eiben, T. Back, M. Schoenauer and H. P. Schwefel (Eds.), *Fifth International Conference on Parallel Problem Solving from Nature* (PPSN-V), Berlin, Germany; (1998) 292 - 301.

[52] Zitzler E., Evolutionary Algorithms for multi-objective optimization: method and applications. P.h.D Thesis, dissertation ETH NO. 13398, *Swaziland Federal Institute of Technology Zorikh*, Switzerland, (1999).

[53] Zhang G, Lu J .A linguistic intelligent user guide for method selection in multi-objective decision support systems, *Information Sciences*; 179 (14), (2009) 2299-2308.

[54] Zhao W., Wang W, Coordination of joint pricing-production decisions in a supply chain, *IIE Transaction*; 34, (2002) 701-715.

# Research highlights

- A b-objective mathematical model is presented for a location-queuing-pricing problem
- This paper considers *M/M/m/K* queuing system at each facility while *M/M/*1 is usual in the literature.
- Unlike previously published papers which consider the same price at all facilities we consider different prices.
- We developed a multi-objective vibration damping optimization to find Pareto solutions.
- Taguchi method is also implemented using a response metric to tune the parameters.
- Some test problems are generated to compare the algorithm with non-dominated sorting genetic algorithm (NSGA-II)