Scientia Iranica E (2015) 22(3), 1103–1116

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
www.scientiairanica.com

Review Article

Service centers location problem considering service diversity within queuing framework

F. Partovi and M. Seifbarghy*

Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.

Received 29 December 2012; received in revised form 9 May 2014; accepted 17 June 2014

KEYWORDS
Location;
Queuing systems;
Service diversity;
Particle swarm optimization;
Variable neighborhood search.

Abstract. In this paper, a new model is developed considering diversity of service in service centers location problem. It is assumed that different services can be provided at each service center. The model has three objective functions including: minimizing the sum of customers travel time and waiting time in service centers, balancing service loads among the given centers, and minimizing the total establishment costs of service centers and assignment costs of servers. Different number of servers can be assigned to each service center. Regarding the allocation of customers to the service centers, each customer patronizes with respect to the distance to the center, the attractiveness of each service center's site for the customer and the number of located servers at the service center. Since the proposed model is of nonlinear integer programming type and is of high complexity in solving, two meta-heuristic based heuristics using Particle Swarm Optimization (PSO) and Variable Neighborhood Search (VNS) are proposed in order to solve the problem. Different sizes of numerical examples are designed and solved in order to compare the efficiency of the heuristics.

1. Introduction

Location problems generally deal with finding the location of one or more facilities in such way as to minimize the establishment and transportation costs or to maximize the market share, reliability and so on. One of the attractive problems in location is service centers location problem. Most of the traditional location problems assumed customers demand to be constant such as the well-known p-median problem proposed by Hakimi [1]. Service center location problem belongs to the family of congested location problems in which the inter-arrival time of customers and service times are assumed to be stochastic; the commonly considered probability distribution function for the addressed times is exponential. Application of such problems are finding the location of medical facilities in which the number of medical care staffs should be determined at each location, the location of post offices, and the location of bank branches and "Automatic Teller Machines (ATMs)".

Berman et al. [2] propose a heuristic algorithm in order to solve the service centers location problem considering that the servers are of an M/G/1 queuing system. Berman et al. [3] developed the given heuristic in order to find the optimal location of a set of p servers in congestion networks. Marianov and Serra [4] formulated several maximal coverage models and developed heuristics to solve the problems; an important constraint of the given models is that nobody stands on line for a time longer than a given time limit.

Formulation of service centers location problem while, at most, one server can be located at each po-

*. *Corresponding author. Mobile: +98 9121232308; Fax: +98 21 77207041*
E-mail addresses: fezi_parto@yahoo.com (F. Partovi); m.seifbarghy@qiau.ac.ir (M. Seifbarghy)

tential location is given in [5,6]; the system is modeled as an M/M/1 queuing system. The objective is to minimize the sum of travel time and average waiting time of all customers. Berman and Drezner [7] generalized their research and introduced multiple servers location problem in a stochastic environment. Multiple servers are allowed to be located at each potential location; the system is modeled as an M/M/m queuing system. Aboolian et al. [8] proposed a model in which one or more servers are allowed to be located at each potential location, but the objective is to minimize the maximum travel time plus the average waiting time spent at the service facility for all customers.

Boffy et al. [9] presented a review of congestion location problems with immobile servers. However, there is an excellent coverage of the mobile servers location problem by Berman and Krass [10]. The given model in this paper is concerned with immobile servers. Drezner and Drezner [11] introduced new models on multi-server location problem with gravity. They assumed that each customer patronizes each facility with respect to distance and the attractiveness of each service center's site. The objective is to minimize the sum of travel time and average waiting time spent at the service centers for all customers. Pasandideh and Niaki [12] proposed a bi-objective facility location problem within M/M/1 queuing framework based on the p-median problem. They used the desirability function technique and Genetic Algorithm (GA) to solve the problem. Seifbarghy et al. [13] proposed a model for service centers location problem within M/M/m queuing system framework; the objective is to minimize the average queue length. Allocation of customers to service centers is made considering distance and the number of servers at each service center. Pasandideh et al. [14] proposed a three-objective model with batch arrivals to service centers. They converted this model into a single-objective one using the LP-metric method and then solved it using GA and Simulated Annealing (SA) algorithms.

Konur and Geunes [15] formulated a model on location decisions for competitive firms serving a set of markets. Firms incur firm-specific transportation, congestion, and location costs, and market price is linear and decreasing with the amount shipped to the markets by all firms.

Lakshmi and Iyer [16] proposed a review paper regarding the contributions and applications of queuing theory in the field of health care management problems. This review has proposed a classification of health care problems, which are modeled using queuing models. Jouzdani et al. [17] proposed dynamic dairy facility location and supply chain planning through minimizing the costs of facility location, traffic congestion and transportation of raw/processed milk and dairy products under demand uncertainty.

Vidyarthi and Jayaswal [18] proposed a nonlinear model on location-allocation problem considering the effects of congestions and queuing delays. The problem seeks to simultaneously locate service facilities, equip them with appropriate capacities, and allocate user demand to these facilities in such a way as to minimize the total system costs. Wang et al. [19] proposed queuing problems of bulk arrival and service with the balking and reneging behavior of customers. This study formulates queues of this type using compound Poisson processes and determines some key probabilistic measures.

In all the aforementioned researches, it is assumed that only one type of service is provided at each service center. Most service centers provide different types of services. For example, different types of fuels such as regular petrol, super petrol and gasoil may be offered in a petrol station. In a clinic, there are a number of doctors with different expertise. Different services differ from each other in some criteria such as service rate, demand rate, cost and number of servers. Another issue is balancing the service load among servers in order to prevent both from the huge congestion at a service center and idleness of servers.

In this research, a new model on service centers location with service diversity is proposed. The model is of three objective functions including: minimizing the customers' travel time to service center and waiting time while receiving service, balancing service load among different centers, and minimizing the location costs of service centers beside the assignment costs of servers. A major assumption is the possibility of assigning different number of servers to each service center. On allocating customers to centers, a fraction of customers with respect to distance to centers, the attractiveness of service centers site and the number of servers at the centers patronize the favorite center in order to receive service. Two meta-heuristic based heuristics including Particle Swarm Optimization (PSO) and Variable Neighborhood Search (VNS) are developed to solve the addressed problem.

The rest of this paper is structured as follows: notation and problem formulation are presented in Section 2. In Section 3, the aforementioned heuristics are presented. Some numerical examples are designed in order to evaluate the efficiency of the proposed heuristics. The results are given in Section 4. Conclusions and suggestions for further research are given in Section 5.

2. Problem definition and assumptions

The system under study is a network where arcs are the possible paths between nodes and the nodes are the demand points, which are also candidate locations for locating service centers. The following assumptions are considered:

- The service request of each demand point follows an independent Poisson process;
- Each service center has at least one server with exponential service time;
- Each service center may present different types of services, which are independent from each other;
- Each service center behaves as an $M/M/m_j^k$ queue system (k and j represent the service type and location, respectively).

2.1. Notation and problem formulation

The notations used in this model are as follows:

i	Index of demand nodes;
j	Index of candidate locations for locating service centers;
k	Index of service types;
N	Set of network nodes;
M	Set of different types of services;
P^k	Total number of the servers of type k ($k \in M$) which should be located at service centers;
f_j	Fixed cost of locating a new service center at node j ($j \in N$);
A_j	Attractiveness rate of node j for the customers of demand points;
c_j^k	Fixed operating cost of a server of type k at node j;
h_i^k	Demand rate of service of type k at node i ($i \in N$);
w_j^k	Average waiting time of customers while receiving service of type k at node j;
λ_j^k	Arrival rate of customers at each service center j in order to receive service of type k;
μ^k	Service rate of a server of type k;
v	Speed of a customer when moving to receive a service at a service center;
d_{ij}	Distance between nodes i and j;
t_{ij}	Travel time between nodes i and j;
x_{ij}^k	The probability of a customer at node i patronizing a service center at node j for service of type k;
R	Maximal productivity rate considered for all types of services at each service center ($0 < R < 1$);
γ	Constant parameter of the distance exponential function;
m_j^k	Integer decision variable, which represents the number of servers of type k located at node j;
y_j	Binary decision variable, which represents locating or not locating a service center at node j.

Based on the definition given for h_i^k and given the fact that demand generation process at each node is Poisson, the demand rate at node j can be obtained by:

$$\lambda_j^k = \sum_{i \in N} h_i^k x_{ij}^k, \qquad \forall\, j \in N, \qquad \forall\, k \in M. \tag{1}$$

As mentioned earlier, three factors including travel time to service centers, the attractiveness of the service centers site and the number of servers at the centers influence customer's behavior while choosing a service center. The probability x_{ij}^k of a customer at node i chooses a service center at node j, for service of type k can be defined as:

$$x_{ij}^k = \frac{m_j^k A_j e^{-\gamma t_{ij}}}{\sum_{j \in N} m_j^k A_j e^{-\gamma t_{ij}}}, \qquad \forall\, i, j \in N, \qquad \forall\, k \in M. \tag{2}$$

The constant parameter γ is assumed to be computed as $\frac{\pi}{\sigma\sqrt{6}}$, where σ is the standard deviation in "taste" of the customers as given by McFadden in [20]. If γ turns out to be a big value, all customers at a demand node will usually patronize the same service center. As γ decreases, the dispersion in service centers selection increases, which means that customers at the demand node i will not always choose the service center located at the same node. This may happen because of the customers' access to some types of information or the customers' experience of the congestion at the service center located at the same node. Furthermore, the average waiting time for getting service of type k at node j in an $M/M/m_j^k$ queuing system is as follows [21]:

$$w_j^k = \left(\frac{1}{\mu^k}\right) + \left(\frac{\mu^k \left(\frac{\lambda_j^k}{\mu^k}\right)^{m_j^k}}{\left(m_j^k \mu^k - \lambda_j^k\right)^2 \left(m_j^k - 1\right)!}\right) \left(\left(\frac{1}{m_j^k!}\right)\left(\frac{\lambda_j^k}{\mu^k}\right)^{m_j^k}\left(\frac{\mu^k m_j^k}{m_j^k \mu^k - \lambda_j^k}\right) + \sum_{r=0}^{m_j^k - 1}\left(\frac{1}{r!}\right)\left(\frac{\lambda_j^k}{\mu^k}\right)^{r}\right)^{-1}, \qquad \forall\, j \in N, \qquad \forall\, k \in M. \tag{3}$$

One of the major contributions of the proposed model is balancing the service load at the service centers. The difference between the real demand rate of service of type k at node j (λ_j^k) and the estimated average demand rate of the same service ($\frac{\sum_{i\in N} h_i^k}{P^k}$)($m_j^k$) is proposed for this purpose noting that the statement $\left(\frac{\sum_{i\in N} h_i^k}{P^k}\right)$ gives the average demand rate at each server of type k.

The proposed model can be stated by the following equations:

$$\min Z_1 = \sum_{k\in M}\sum_{j\in N}\sum_{i\in N}\lambda_j^k t_{ij} + \sum_{j\in N}\sum_{k\in M}\lambda_j^k w_j^k, \quad (4)$$

$$\min Z_2 = \sum_{j\in N}\sum_{k\in M}\left(\lambda_j^k - \left(\frac{\sum_{i\in N} h_i^k}{P^k}\right)(m_j^k)\right)^2, \quad (5)$$

$$\min Z_3 = \sum_{j\in N}\sum_{k\in M}(f_j y_j + c_j^k m_j^k). \quad (6)$$

S.t.:

$$\lambda_j^k = \sum_{i\in N} h_i^k x_{ij}^k, \qquad \forall\, j\in N, \qquad \forall\, k\in M, \quad (7)$$

$$\sum_{j\in N} m_j^k = P^k, \qquad \forall\, k\in M, \quad (8)$$

$$x_{ij}^k = \frac{m_j^k A_j e^{-\gamma t_{ij}}}{\sum\limits_{j\in N} m_j^k A_j e^{-\gamma t_{ij}}},$$

$$\forall\, i,j\in N, \qquad \forall\, k\in M, \quad (9)$$

$$w_j^k = \left(\frac{1}{\mu^k}\right) + \left(\frac{\mu^k\left(\frac{\lambda_j^k}{\mu^k}\right)^{m_j^k}}{(m_j^k\mu^k - \lambda_j^k)^2(m_j^k-1)!}\right)$$

$$\left(\left(\frac{1}{m_j^k!}\right)\left(\frac{\lambda_j^k}{\mu^k}\right)^{m_j^k}\left(\frac{\mu^k m_j^k}{m_j^k\mu^k - \lambda_j^k}\right)\right.$$

$$\left. + \sum_{r=0}^{m_j^k-1}\left(\frac{1}{r!}\right)\left(\frac{\lambda_j^k}{\mu^k}\right)^r\right)^{-1},$$

$$\forall\, j\in N, \qquad \forall\, k\in M, \quad (10)$$

$$t_{ij} = \frac{d_{ij}}{v}, \qquad \forall\, i,j\in N, \quad (11)$$

$$\frac{\lambda_j^k}{u^k m_j^k} \le R, \qquad \forall\, j\in N, \qquad \forall\, k\in M, \quad (12)$$

$$m_j^k \le P^k y_j, \qquad \forall\, j\in N, \qquad \forall\, k\in M, \quad (13)$$

$$\sum_{k\in M} m_j^k \ge y_j, \qquad \forall\, j\in N, \quad (14)$$

$$m_j^k \ge 0, \quad \text{Integer}, \qquad \forall\, j\in N, \qquad \forall\, k\in M, \quad (15)$$

$$y_j \in \{0,1\}, \qquad \forall\, j\in N. \quad (16)$$

The first objective function in Eq. (4) represents the sum of travel time of customers to service centers and waiting times of customers at service centers. Eq. (5) gives the second objective function, which balances the service load at service centers. The third objective function given by Eq. (6) minimizes the sum of fixed costs of locating service centers and operation of the servers. Eq. (7) gives the demand rate of service of type k at each service center located at node j. Eq. (8) ensures the predetermined number of servers of each service type to be located at the nodes. Eq. (9) gives the probability of choosing a service center located at node j in order to receive a service of type k by customer at node i. Eq. (10) presents the average waiting times of customers, while receiving service of type k at node j. Eq. (11) gives the formulae of computing travel time from node i to node j. Eq. (12) ensures the productivity rate of service to be less than or equal to its maximal value at each service center for each service type. Eq. (13) implies that assigning a server to a given node must be done after selecting the node as a service center. Eq. (14) guarantees that at least one server is located at a given service center. Eqs. (15) and (16) indicate the status of the decision variables.

2.2. Converting the multi-objective model

One of the most widely used techniques for solving multi-objective optimization problems is LP-metric. In order to transform the given three-objective model into a single one, the differences between each objective function and the corresponding optimum value are minimized as follows [22]:

$$\min Z(x) = \left[\sum_{i=1}^{Q}\left[\eta_i\left|\frac{z_i(x) - z_i^*}{z_i^{\wedge} - z_i^*}\right|\right]^p\right]^{\frac{1}{p}}, \quad (17)$$

where z_i^* and $z_i^{\wedge}$ represent the optimum and worst values of the ith objective functions subject to the given constraints, respectively; η_i represents the weight of objective function i, and Q is the number of objective functions. Considering $P=\infty$, the single objective function for the current model will be equal to a Min-Max problem with objective function of α as in Relations (18) to (21) adding with Constraints (7)-(16).

$$\min Z = \alpha, \quad (18)$$

s.t.:

$$\alpha \geq \eta_1 \left| \frac{z_1 - z_1^*}{z_1^{\wedge} - z_1^*} \right|, \tag{19}$$

$$\alpha \geq \eta_2 \left| \frac{z_2 - z_2^*}{z_2^{\wedge} - z_2^*} \right|, \tag{20}$$

$$\alpha \geq \eta_3 \left| \frac{z_3 - z_3^*}{z_3^{\wedge} - z_3^*} \right|. \tag{21}$$

It should be noted that $\eta_1 + \eta_2 + \eta_3$ is equal to one. It is proven that different values of η_1, η_2 and η_3 lead to generating efficient solutions of the initial model.

3. Solution algorithms

Since the problem belongs to nonlinear integer programming (NLIP) problems, using the common solution methodologies will be time-consuming, especially when the problem size increases. This is the reason why we propose two meta-heuristic-based heuristics to solve the addressed problem. There are several examples in which such heuristics are proposed for NLIP. Aboolian et al. [8] and Drezner and Drezner [11] used meta-heuristic algorithms for solving service centers location problem including PSO and VNS.

3.1. Solution representation

We present a given solution by a matrix with M rows and N columns. Each column represents a potential location and each row represents a type of service. Each element of the matrix represents the number of servers of the corresponding service type and location. For example in Figure 1, considering $M = 3$ and $N = 4$, the solution matrix indicates that service centers are located at all potential locations except for the second one.

$$\begin{pmatrix} 1 & 0 & 2 & 1 \\ 2 & 0 & 0 & 3 \\ 3 & 0 & 3 & 0 \end{pmatrix}$$

Figure 1. A sample solution of the problem.

3.2. Particle swarm optimization algorithm

Initially, PSO was introduced by Kennedy and Eberhart [23] as a member of swarm intelligence techniques. PSO is a population based search algorithm founded on the simulation of the social behavior of birds, bees or a school of fishes. PSO can be easily implemented and has been useful for solving a great number of problems [24].

Initially, a swarm of particles with random positions and velocities are generated. The velocity and position of all particles are updated based on the inertia of direction, personal best experience and global best experience of swarm.

The velocity and position of each particle can be stated as in :

$$\vec{v}_{k+1} = \vec{w}_k.\vec{v}_k + \vec{c}_1.\vec{r}_1.(\vec{p}_1 - \vec{x}_k) + \vec{c}_2.\vec{r}_2.(\vec{p}_2 - \vec{x}_k), \tag{22}$$

$$\vec{x}_{k+1} = \vec{x}_k + \vec{v}_{k+1}. \tag{23}$$

The sign point (.) in the aforementioned Eq. (22) represents the multiplication of the element by element of the vectors. Parameter $\vec{w}_k$ represents inertia coefficient, which is attributed to the current position of each particle. Parameter $\vec{c}_1$ represents the coefficient, which is attributed to the personal best position of each particle. Vector $\vec{p}_1$ represents the particle's best position found so far. Parameter $\vec{c}_2$ represents the coefficient, which is attributed to the global best position of swarm. Vector $\vec{p}_2$ represents the best known position found by any particle in the swarm so far. Coefficients $\vec{r}_1$ and $\vec{r}_2$ are considered as random numbers with values between 0 and 1. Index k represents the current iteration and $\vec{v}_k$ and $\vec{x}_k$ represent the velocity and position of each particle in kth iteration. It should be noted that the inertia coefficient at each iteration is obtained as :

$$\vec{w}_{k+1} = \vec{w}_k \times wdamp, \tag{24}$$

in which, $wdamp$ is the reduction coefficient of inertia. Since the large inertia coefficient (w_k) in Eq. (22) is used for global search and small inertia coefficient is used for local search, it is better to gradually reduce the value of inertia coefficient. In this way, search is done globally at first, and then gradually goes towards local search. In this paper, the reduction coefficient of inertia ($wdamp$) is assumed as one of the input parameters of the PSO algorithm [25].

The general outline of the proposed PSO-based heuristic is as follows:

Step 1. **Initialization:** Generate swarm of particles with random velocities and positions;

Step 2. **Evaluation:** Compute fitness value (objective function) for each particle;

Step 3. **Comparison:** For each particle, compare its fitness value with its best fitness value obtained from previous iterations (the fitness value of personal best experience). If the new position has a better fitness value, personal best experience is replaced with the new position. Furthermore, compare the fitness value of each particle with the best fitness value of the swarm obtained from the previous iteration (the fitness value of global best experience). If the particle is of better fitness value, global best experience is replaced with the particle's position;

Step 4. **Convergence:** Stop algorithm if no improvement occurs in several successive iterations, otherwise go to Step 5;

Step 5. **Updating:** Calculate new inertia coefficient, velocity and position of each particle from Eqs. (22), (23) and (24) and go to Step 2.

3.3. Variable neighborhood search algorithm

VNS was introduced by Hansen and Mladenovic' in [26]. VNS is a recent meta-heuristic algorithm, based on systematic changes in neighborhood structures. Finding solutions with high quality within reasonable time and beside the simplicity are of major characteristics of this algorithm.

The idea of VNS is based on the neighborhood structure changes during the search process. The first step in VNS algorithm is to define neighborhood structures for generating neighborhood solutions. Then, VNS utilizes two main phases including shake procedure and local search. Shake procedure as an innovation process is used for local search loop in order to prevent from falling in local optimum.

The general outline of the proposed basic VNS-based heuristic is as follows:

Step 1. **Selection of neighborhood structures:** Select the set of neighborhood structures;

Step 2. **Initialization:** Generate an initial solution randomly and let it as the best solution. Then, start from first neighborhood structure and first iteration;

Step 3. **Shake procedure:** Generate a neighbor solution for best solution by the kth neighborhood structure;

Step 4. **Local search:** Generate neighbor solutions for solution produced in Step 3 by kth neighborhood structure. Perform a local search and find the best solution;

Step 5. **Comparison:** Compare the solution obtained from local search in Step 4 with the best solution; if it is better than best solution, replace it with the best solution; then, continue with new best solution and first neighborhood structure from Step 3. If it is not better than best solution, go to the second neighborhood structure from Step 3. Continue this process until all neighborhood structures are checked and no improvement occurs in best solution, then go to Step 6;

Step 6. **Convergence:** Stop algorithm if no improvement occurs in several successive iterations, otherwise go to Step 7;

Step 7. **Updating:** Go to the next iteration of the algorithm starting from Step 3 and first neighborhood structure with the best solution found.

The local search method that is used in VNS algorithm is simple local search. At this method, a number of neighbor solutions are generated by each neighborhood structure, and then the best solution is selected.

In the proposed VNS-based heuristic, each solution is represented as Figure 1. Initially, eight neighborhood structures are defined by operational structures such as exchanging two columns, reversing the orders of rows' elements, inserting the total of all row's elements in a column. For example, if $M = 3$ and $N = 4$, then Figure 2 illustrates eight defined neighborhood structures. For better performance and shorter execution time in algorithm, a certain number of neighborhood structures of these eight neighborhood structures were selected in order to implement the algorithm. The way to select effective neighborhood structures and the corresponding orders in VNS algorithm will be described in the next sections.

3.4. Penalty function

In the presence of the constraints of the given model, a number of generated solutions may be infeasible. We use penalty function in order to tackle the addressed problem. When a solution is feasible, the penalty value will be zero, otherwise, it will be considered as a non-zero value. According to general form of constraints as $g(x) = b$, the penalty value of a solution is given as in [27]:

$$p(x) = U \times \max\left\{\left(\frac{g(x)}{b} - 1\right), 0\right\}, \qquad (25)$$

where $P(x)$, U and $g(x)$ represent the penalty value of solution x, a large number and the addressed constraint, respectively. We consider normalization policy within penalty function framework in order to normalize all constraints.

4. Numerical examples and the results

Initially, we design numerical examples; then, the major parameters of the PSO based heuristic are tuned. We also define the neighborhood structures of the VNS based heuristic. The numerical examples with three different sizes are solved using the two heuristics and the results are analyzed in order to asses the performance of the given heuristics.

Both the corresponding algorithms of the heuristics are coded in MATLAB 7.11 (R2010b) and we run each iteration of the program on a laptop with a 4 GB of RAM and 2.30 GHz processor.

4.1. Designing numerical examples

Numerical examples with three different sizes of small, medium and large are designed in which the number of service types and demand nodes are as in Table 1.

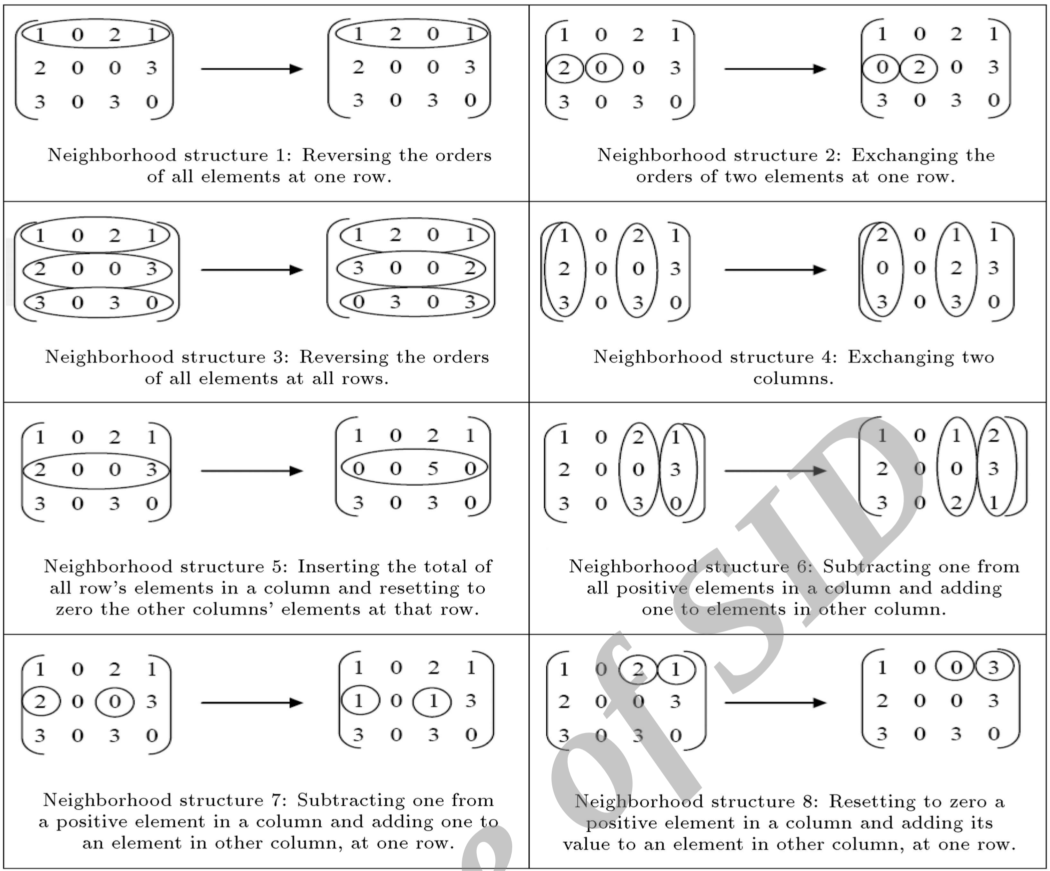

Figure 2. Neighborhood structures for the basic VNS algorithm.

Table 1. Structure of the numerical examples with small, medium and large sizes.

Problem size	Number of service types (M)	Number of demand nodes (N)
Small	2	10
Medium	5	50
Large	10	100

The input parameters' values of the numerical examples are given as in Table 2. The values of P^k ($\forall\ k \in M$) are given in three parts from left to right for small, medium and large sizes, respectively. The numerical examples are generated randomly so that most of the parameters' values are originated from [14]. Furthermore, we let parameter R take value of 0.9; this can be a reasonable value for the maximal productivity rate. We have uniformly generated A_j values from the interval of [0,1]; this can also be a reasonable interval for the attractiveness rate. The value of γ takes 1 according to [28].

4.2. Tuning the parameters of the PSO

In this section, we use Response Surface Method (RSM) in order to tune the PSO parameters. RSM is a mathematical tool for modeling and analyzing of problems in which several independent variables affect a dependent variable (response), and we aim to optimize the response. The first step in RSM is to determine a suitable estimation for the response. This can be suggested by [29]:

$$y = f(x_1, x_2, \cdots, x_k) + e_r. \tag{26}$$

Table 2. Input parameters' values of the numerical examples.

Parameter	Parameter's values
h_i^k	Uniform [1,15]
μ^k	Uniform [50,80]
d_{ij}	Uniform [50,100]
V	10
f_j	Uniform [100,500]
c_j^k	Uniform [1,9]
P^k	Discrete uniform [4,7] Discrete uniform [20,30] Discrete uniform [50,60]
A_j	Uniform [0,1]
R	0.9
γ	1
η_1, η_2, η_3	0.5, 0.2, 0.3

Function f is called surface (function) response between the response variable y and independent variables $x_1, x_2, \cdots, x_k$ of k quantitative factors. The additional e_r measures the experimental errors. We define two levels of high and low for the different parameters of the algorithm as in Table 3 in which, $nPop$ represents the number of particles in the swarm. The addressed levels is obtained by try and error.

Moreover, the type of experimental design is cubic and we use Central Composite Designs (CCD) for the experiments; therefore, the distance α of the axial points from the design center to generate a face-centered design is utilized with $\alpha = 1$. Furthermore, we consider cube points, since the factors setting represent the cube points in the design [30].

The CCD used in this research, is the design of fractional in which we have 2^{k-p} factorial points, $2k$ axial points and 5 center points; k represents the number of factors and is equal to 5. Considering $p = 1$, there are 31 combinational runs of experiments. The results for medium size are given as in Table 4. The

Table 3. High and low levels of input parameters of the PSO for different size problems.

Problem size	Level	$nPop$	W	$wdamp$	C_1	C_2
Small	Low	10	0.4	0.9000	2	2
	High	30	0.9	0.9999	4	4
Medium	Low	20	0.4	0.9000	2	2
	High	40	0.9	0.9999	4	4
Large	Low	30	0.4	0.9000	2	2
	High	50	0.9	0.9999	4	4

Table 4. Input parameters and response values for the medium size problems.

Row	Design point	$nPop$	w	$wdamp$	C_1	C_2	Z (response)
1	Factorial	10	0.40	0.9000	2	4	0.0263
2	Factorial	30	0.40	0.9000	2	2	0.0275
3	Factorial	10	0.90	0.9000	2	2	0.0276
4	Factorial	30	0.90	0.9000	2	4	0.0299
5	Factorial	10	0.40	0.9999	2	2	0.0249
6	Factorial	30	0.40	0.9999	2	4	0.0247
7	Factorial	10	0.90	0.9999	2	4	0.0253
8	Factorial	30	0.90	0.9999	2	2	0.0232
9	Factorial	10	0.40	0.9000	4	2	0.0176
10	Factorial	30	0.40	0.9000	4	4	0.0228
11	Factorial	10	0.90	0.9000	4	4	0.0250
12	Factorial	30	0.90	0.9000	4	2	0.0232
13	Factorial	10	0.40	0.9999	4	4	0.0224
14	Factorial	30	0.40	0.9999	4	2	0.0218
15	Factorial	10	0.90	0.9999	4	2	0.0204
16	Factorial	30	0.90	0.9999	4	4	0.0265
17	Axial	10	0.65	0.9500	3	3	0.0253
18	Axial	30	0.65	0.9500	3	3	0.0170
19	Axial	20	0.40	0.9500	3	3	0.0213
20	Axial	20	0.90	0.9500	3	3	0.0215
21	Axial	20	0.65	0.9000	3	3	0.0247
22	Axial	20	0.65	0.9999	3	3	0.0203
23	Axial	20	0.65	0.9500	2	3	0.0225
24	Axial	20	0.65	0.9500	4	3	0.0243
25	Axial	20	0.65	0.9500	3	2	0.0241
26	Axial	20	0.65	0.9500	3	4	0.0207
27	Center	20	0.65	0.9500	3	3	0.0209
28	Center	20	0.65	0.9500	3	3	0.0258
29	Center	20	0.65	0.9500	3	3	0.0232
30	Center	20	0.65	0.9500	3	3	0.0239
31	Center	20	0.65	0.9500	3	3	0.0225

result of each experiment (Response) is the average of five iterations.

The same procedure is done for the small and large size problems. Then, the Regression equations are estimated for the three sizes considering the results. The Regression equations for small, medium and large size problems are given as in Eqs. (27)-(29), respectively:

$$\begin{aligned}R_{\text{PSO-Small}} &= -1.43594 + 0..00272nPop \\ &+ 0.22310w + 3.14469wdamp - 0.07211c_1 \\ &- 0.00794c_2 + 0.00007nPop^2 - 0.03436w^2 \\ &- 1.62646wdamp^2 + 0.00029c_1^2 + 0.00250c_2^2 \\ &+ 0.00072nPop \times w - 0.00504nPop \times wdamp \\ &- 0.00017nPop \times c_1 - 0.00036nPop \times c_2 \\ &- 0.18704w \times wdamp - 0.00196w \times c_1 \\ &- 0.00557w \times c_2 + 0.06285wdamp \times c_1 \\ &- 0.00575wdamp \times c_2 + 0.00386c_1 \times c_2, \qquad (27)\end{aligned}$$

$$\begin{aligned}R_{\text{PSO-Medium}} &= 0.329809 - 0.000002nPop \\ &+ 0.019435w - 0.514096wdamp - 0.029017c_1 \\ &- 0.013725c_2 - 0.0000077nPop^2 - 0.007358w^2 \\ &+ 0.232338wdamp^2 + 0.001497c_1^2 + 0.000488c_2^2 \\ &- 0.000205nPop \times w + 0.000406nPop \times wdamp \\ &+ 0.000093nPop \times c_1 - 0.000044nPop \times c_2 \\ &- 0.014795w \times wdamp + 0.003705w \times c_1 \\ &+ 0.000207w \times c_2 + 0.010927wdamp \times c_1 \\ &+ 0.009509wdamp \times c_2 + 0.001103c_1 \times c_2, \qquad (28)\end{aligned}$$

$$\begin{aligned}R_{\text{PSO-Large}} &= -0.303772 + 0.001989nPop \\ &+ 0.093382w + 0.756076wdamp - 0.010714c_1 \\ &- 0.049036c_2 - 0.000015nPop^2 - 0.019425w^2 \\ &- 0.439688wdamp^2 - 0.003295c_1^2 + 0.006519c_2^2 \\ &- 0.000583nPop \times w - 0.000358nPop \times wdamp \\ &- 0.000021nPop \times c_1 - 0.000021nPop \times c_2 \\ &- 0.050442w \times wdamp + 0.000613w \times c_1 \\ &+ 0.000436w \times c_2 + 0.028687wdamp \times c_1 \\ &+ 0.008718wdamp \times c_2 + 0.000708c_1 \times c_2. \qquad (29)\end{aligned}$$

Table 5. Optimal levels of the parameters of PSO for each problem size.

Problem size	**Swarm size ($nPop$)**	w	$wdamp$	C_1	C_2
Small	10	0.9	0.9000	4	2
Medium	20	0.4	0.9999	4	2
Large	30	0.4	0.9000	4	2.9

Then aforementioned equations are optimized subject to the defined intervals for the input parameters. The optimal combinations of the parameters are given in Table 5 for each size.

The stopping criterion for the PSO is considered getting no improvement for the integrated objective function value (Z) within 10, 15 and 20 successive iterations for small, medium and large size problems, respectively.

4.3. Determining the type and order of neighborhood structures for VNS algorithm

Developing neighborhood structures is of an important effect on the performance of the VNS [26]. Initially, eight different neighborhood structures, which are indicated in Figure 2, are proposed. Then, based on the given structures, we run the program for each numerical example as five times, taking into account all the neighborhood structures and the given orders in Figure 2. The average of the integrated objective function value (Z) for five runs turns out to be 0.0386 with solution time 4.8453 seconds for small size, 0.0125 with solution time 283.7753 seconds for medium size, and 0.0112 with solution time 3573.6304 seconds for large size problems. One neighborhood structure is eliminated for each problem size, alternatively. The solution is obtained for the both cases of with and without a neighborhood structure. The difference between two solutions is calculated and is illustrated in Table 6. If the difference is zero or less than zero, that neighborhood structure is removed; otherwise, the neighborhood structures is arranged in order to obtain the greatest to least positive differences. It should be noted that each solution is the average of five run times.

From Table 6, neighborhood structures 1 and 5 for small size, neighborhood structures 2, 4 and 8 for medium size and neighborhood structures 1, 2, 4 and 7 for large size are removed according to the given procedure, since there is no improvement or negative difference in the integrated objective function

Table 6. Computational results to determine the type and order of neighborhood structures for the VNS.

Problem size	The number of neighborhood structures for eliminating	Integrated objective function values (Z) obtained from eliminating a neighborhood structure	Integrated objective function values (Z) obtained from all eight neighborhood structures	Difference between two solutions
Small	1	0.0360	0.0386	-0.0026
	2	0.0405	0.0386	0.0019
	3	0.0454	0.0386	0.0068
	4	0.0432	0.0386	0.0046
	5	0.0383	0.0386	-0.0003
	6	0.0462	0.0386	0.0076
	7	0.0409	0.0386	0.0023
	8	0.0434	0.0386	0.0048
Medium	1	0.0135	0.0125	0.0010
	2	0.0092	0.0125	-0.0033
	3	0.0162	0.0125	0.0037
	4	0.0107	0.0125	-0.0018
	5	0.0940	0.0125	0.0815
	6	0.0216	0.0125	0.0091
	7	0.0201	0.0125	0.0076
	8	0.0122	0.0125	-0.0003
Large	1	0.0110	0.0112	-0.0002
	2	0.0094	0.0112	-0.0018
	3	0.0119	0.0112	0.0007
	4	0.0098	0.0112	-0.0014
	5	0.0472	0.0112	0.0360
	6	0.0114	0.0112	0.0002
	7	0.0110	0.0112	-0.0002
	8	0.0145	0.0112	0.0033

value (Z). Other neighborhood structures are arranged in order of descending difference values. Thus, the arrangement of neighborhood structures for the VNS are as follows:

For the small size:
Neighborhood structure 6 → neighborhood structure 3 → neighborhood structure 8 → neighborhood structure 4 → neighborhood structure 7 → neighborhood structure 2;

For the medium size:
Neighborhood structure 5 → neighborhood structure 6 → neighborhood structure 7 → neighborhood structure 3 → neighborhood structure 1;

For the large size:
Neighborhood structure 5 → neighborhood structure 8 → neighborhood structure 3 → neighborhood structure 6.

The stopping criterion for VNS is considered getting no improvement on the integrated objective function value (Z) within 10, 15 and 20 successive iterations for small, medium and large size problems, respectively.

4.4. Analysis of results

In this section, numerical examples consisting of ten small size, ten medium size and ten large size problems, which are generated randomly, are solved and the results are analyzed. Furthermore, each problem is solved for five times so that the final result is the average of the run times.

It is necessary to define some criteria to analyze the solutions in order to assess the performance of the given heuristics; in this paper, two criteria including Relative Percentage Index (RPI) and solution time are utilized for this purpose. Regarding RPI, having obtained the solutions of the two heuristics, the best and the worst solutions, called Best_{sol} and $\text{Worst}_{\text{sol}}$, are obtained and then, RPI is computed from Eq. (30) as given by Naderi et al. in [31]:

$$\text{RPI} = \left| \frac{\text{Best}_{\text{sol}} - \text{Alg}_{\text{sol}}}{\text{Best}_{\text{sol}} - \text{Worst}_{\text{sol}}} \right| \times 100, \tag{30}$$

in which Alg_{sol} is the solutions obtained for solving an instance of each problem. RPI takes value between 0 and 100. Clearly, lower values of RPI are preferred. We also use solution time as other performance measure to compare the performance of the heuristics. It is clear that lower values of time solution are preferred. Therefore, we compare the heuristics using the given performance measures for three sizes of problems including small, medium and large sizes. Table 7 gives the results for RPI and solution time for the problems of the three given sizes. It should be noted that the solution times are regarding the integrated objective

Table 7. Performance measures results for small, medium and large size problems.

Problem size	Problem number	Proposed VNS		Proposed PSO	
		PRI	Time (second)	PRI	Time (second)
Small	1	46.2465	0.7968	17.9048	2.9981
	2	24.6578	0.7163	8.1870	3.5192
	3	20.0000	0.6250	19.9406	3.6550
	4	26.1345	0.7148	20.0000	3.1336
	5	62.7325	0.7439	29.6256	3.0383
	6	37.5380	0.7860	23.3209	4.1408
	7	29.1736	0.7840	1.2645	4.6294
	8	58.7099	0.5931	8.3208	2.5058
	9	34.6496	0.7633	14.9198	3.6592
	10	46.5677	1.0187	10.1124	4.5928
	Average	38.6410	0.7542	15.3596	3.5872
Medium	1	39.6331	56.0510	3.6958	121.7189
	2	44.4542	81.6468	1.8006	81.9951
	3	73.1701	58.4505	37.1754	153.9089
	4	56.1810	75.7589	40.8264	87.5870
	5	72.2546	64.6167	35.0439	100.3495
	6	56.6740	71.0616	19.9877	116.8319
	7	67.4515	66.2246	26.2448	93.9695
	8	63.3214	57.2004	37.5155	121.0114
	9	55.9641	69.7544	15.1105	108.1125
	10	38.4036	64.2211	12.5040	113.6038
	Average	56.7508	66.4986	22.9905	109.9089
Large	1	61.3471	1039.1484	36.5919	1394.2684
	2	68.2506	760.6648	4.4838	905.6184
	3	68.4546	1195.6272	13.2063	953.5308
	4	52.5654	825.0824	3.5557	1425.4338
	5	68.4120	917.9315	7.0129	1775.0087
	6	76.5279	1366.8287	5.6881	772.0368
	7	77.9441	858.0823	7.6240	2040.0848
	8	62.8718	755.2276	4.7663	2234.6916
	9	48.9008	799.8815	2.3388	899.5159
	10	84.0510	894.2079	16.5409	1780.2962
	Average	66.9325	941.2682	10.1809	1418.0485

Table 8. The values of decision variables and objective functions for a small size problem.

Algorithm name	Y	M	Z_1	Z_2	Z_3	Z
PSO	[0 1 0 0 1 0 0 0 1 0]	$\begin{pmatrix} 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \end{pmatrix}$	10806.2590	595.6060	1276	0.0660
VNS	[1 0 1 0 0 0 0 0 0 0]	$\begin{pmatrix} 3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	10527.3580	49.0340	911	0.0427

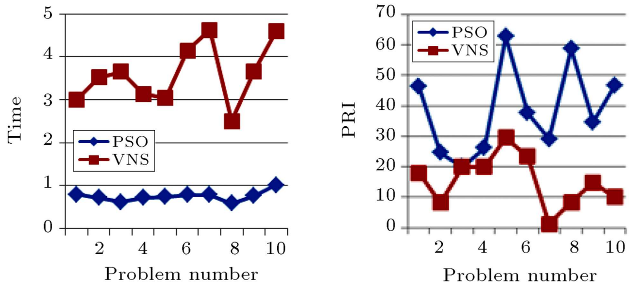

Figure 3. Time and RPI results for the small size problems.

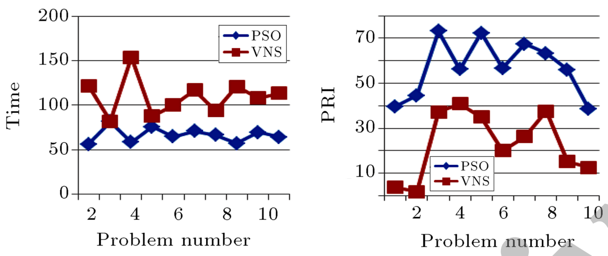

Figure 4. Time and RPI results for the medium size problems.

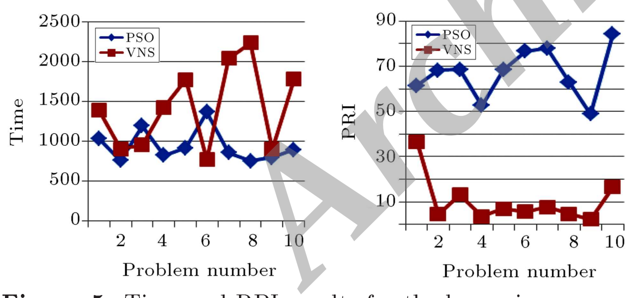

Figure 5. Time and RPI results for the large size problems.

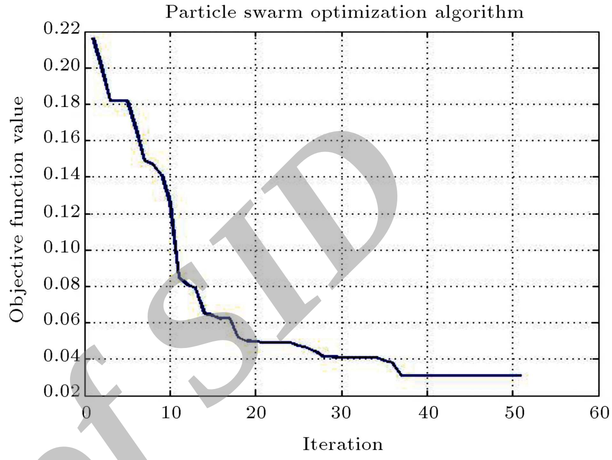

Figure 6. Convergence diagram of PSO for problem 8 in medium size.

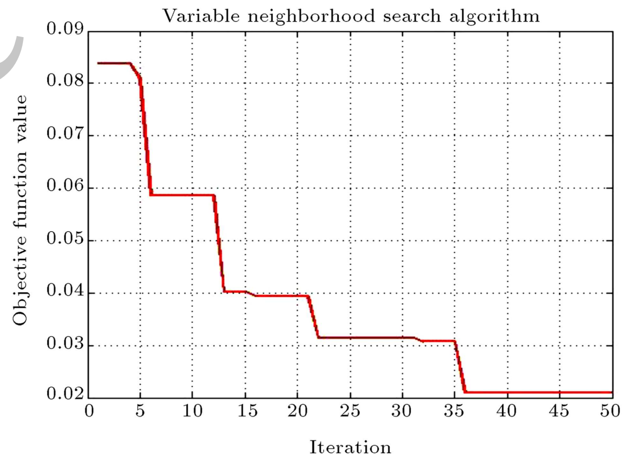

Figure 7. Convergence diagram of VNS for problem 8 in medium size.

function in terms of second unit. The values of decision variables and objective functions for a small size problem are given in Table 8 as an instance.

The computational results from each heuristic are graphically indicated by Figures 3-5 for small, medium and large size problems regarding the two performance measures. The convergence diagrams of the proposed heuristics are indicated as in Figures 6 and 7 for problem 8 in the medium size as an instance.

The results from solving numerical examples with different sizes in Table 7 indicate that the VNS based heuristic outperforms the PSO for all the problems considering RPI values. Figures 3-5 also confirms this statement. The objective functions values in Table 8 indicate that the VNS works better than the PSO. In all, the VNS gives better solutions; however, the PSO solves in shorter times.

5. Conclusions and further research

In this paper, a new model in service centers location with several servers considering service diversity was proposed. We assumed that different and independent service types can be given at each service center.

Service diversity has very important role in locating service centers, because many real world service centers provide different service types. Different services differ from each other from the viewpoint of waiting time, service rate, demand for service, efficiency coefficient, cost and number of servers. The differences are of great importance in correct location of service centers and finding suitable number of servers. Balancing service loads among the service centers can prevent either from huge congestion or from idleness of the servers. In this research, service centers were located considering the possibility of establishing different types of services in a service center with three objective functions including

1. Minimizing the sum of customers' travel and waiting time in service centers;
2. Balancing service loads;
3. Minimizing the locating costs of service centers and assignment costs of servers.

Customers were assigned to the service centers whose servers had been assigned before. Customers were assumed to patronize the favorite service center, considering closeness, attractiveness and the number of servers of each service center. Two meta-heuristic algorithms including PSO and VNS were used in order to solve a number of numerical examples. The VNS gives better solutions; however, the PSO solves in shorter times.

The given model in this research can be applied to the location of service centers, petrol stations, clinics and hospitals, terminals, banks and ATMs.

The followings can be considered as future researches:

- Budget constraint can be considered for location of service centers;
- Other queuing systems can be considered for the servers;
- Pareto-based multi-objective solution techniques can be applied;
- Some of the parameters can be considered as fuzzy numbers such as service and demand rate.

References

1. Hakimi, Sl. "Optimal locations of switching centers and the absolute centers and medians of a graph", *Operational Research*, **12**, pp. 450-459 (1964).
2. Berman, O., Larson, R. and Chiu, S. "Optimal server location on a network operating as an M/G/1 queue", *Operational Research*, **12**(4), pp. 746-77 (1985).
3. Berman, O., Larsen, R. and Parken, C. "The stochastic queue P-median location problem", *Transportation Society*, **21**, pp. 207-216 (1987).
4. Marianov, V. and Serra, D. "Probabilistic maximal covering location-allocation for congested system", *Journal of Regional Science*, **38**, pp. 401-424 (1998).
5. Wang, Q., Batta, R. and Rump, CM. "Algorithms for a facility location problem with stochastic customer demand and immobile servers", *Annals of Operations Research*, **111**, pp. 17-34 (2002).
6. Berman, O. and Krass, D. "Recent developments in the theory and applications of location models, Part II", *Annals of Operations Research*, **111**, pp. 15-16 (2002).
7. Berman, O. and Drezner, Z. "The multiple server location problem", *Journal of Operational Research Society*, **58**, pp. 91-99 (2007).
8. Aboolian, R., Berman, O. and Drezner, Z. "The multiple server center location problem", *Annals of Operations Research*, **167**, pp. 337-352 (2008).
9. Boffy, B.T., Galvao, R.D. and Espejo, L. "A review of congestion models in the location of facilities with immobile servers", *European Journal of Operational Research*, **178**, pp. 643-662 (2008).
10. Berman, O. and Krass, D. "Facility location problems with stochastic demands and congestions", In *Facility Location: Applications and Theory*, Z. Drezner and H.W. Hamacher, Eds., Springer, New York (2001).
11. Drezner, T. and Drezner, Z. "The gravity multiple server location problem", *Computer and Operation Research*, **38**, pp. 694-701 (2011).
12. Pasandideh, S.H.R. and Niaki, S.T.A. "Genetic application in a facility location problem with random demand within queuing framework", *Journal of Intelligent Manufacturing*, **23**(3), pp. 651-659 (2012).
13. Seifbarghy, M., Rabieian, R. and Karimi, F. "Location service centers optimizing customers' perspective criteria", *International Journal of Advanced Manufacturing Technology*, **54**, pp. 811-819 (2011).
14. Pasandideh, S.H.R., Niaki, S.T.A. and Hajipour, V. "A multi-objective facility location model with batch arrivals: Two parameter-tuned meta-heuristic algorithms", *Journal of Intelligent Manufacturing*, **24**(2), pp. 331-348 (2011).
15. Konur, D. and Geunes, J. "Competitive multi-facility location games with non-identical firms and convex traffic congestion costs", *Transportations Research Part E: Logistics and Transportation Review*, **48**, pp. 373-385 (2012).
16. Lakshmi, C. and Lyer, S.A. "Application of queueing theory in health care: A literature review", *Operations Research for Health Care*, **2**, pp. 25-39 (2013).
17. Jouzdani, J., Sadjadi, S.J. and Fathian, M. "Dynamic dairy facility location and supply chain planning under traffic congestion and demand uncertainty: A case study of Tehran", *Applied Mathematical Modelling*, **37**, pp. 8467-8483 (2013).
18. Vidyarthi, N. and Jayaswal, S. "Efficient solution of a class of location-allocation problems with stochastic demand and congestion", *Computers and Operations Research*, **48**, pp. 20-30 (2014).

19. Wang, Y., Guo, J., Ceder, A., Currie, G., Dong, W. and Yuan, H. "Waiting for public transport services: Queueing analysis with balking and reneging behaviors of impatient passengers", *Transportation Research Part B: Methodological*, **63**, pp. 53-76 (2014).

20. MacFadden, D. "Conditional logit analysis of qualitative choice behavior", In *Frontiers in Econometrics*, P. Zarembka, Ed., Academic, New York (1974).

21. Klienrock, L., *Queuing Systems*, Wiley, New York (1975).

22. Asgharpour, M.J., *Multi-Criteria Decision Making*, 6th Ed., Tehran University Press, Iran (2009).

23. Kennedy, J. and Eberhart, R.C. "Particle swarm optimization", In *Proceedings of IEEE International Conference on Neural Network*, **IV**, pp. 1942-1948 (1995).

24. Sadeghpour, M., Salarieh, H., Vossoughi, GH. and Alasty, A. "Multi-variable control of chaos using PSO-based minimum entropy control", *Commun. Nonlinear Sci. Numer. Simulat.*, **16**, pp. 2397-2404 (2011).

25. Noori Darvish, S., Tavakkoli-Moghaddam, R. and Javadian, N. "A multi-objective particle swarm optimization algorithm for a possible open shop problem to minimize weighted mean tardiness and Weighted mean completion times", *Iranian Journal of Operations Research*, **3**(1), pp. 61-88 (2012).

26. Hansen, P. and Mladenovic', N. "Variable neighborhood search", *Computers and Operations Research*, **24**(11), pp. 1097-1100 (1997).

27. Yeniay, O. and Ankare, B. "Penalty function methods for constrained optimization with genetic algorithms", *Mathematical and Computational Application*, **10**, pp. 45-56 (2005).

28. Marianov, V., Rios, M. and Icaza, M.J. "Facility location for market capture when users rank facilities by shorter travel and waiting times", *European Journal of Operational Research*, **191**, pp. 32-44 (2008).

29. Montgomery, D.C., *Design and Analysis of Experiments*, 6th Ed., New York, Wiley, USA (2005).

30. Najafi, A.A. Niaki, S.T.A. and Shahsavar, M. "A parameter-tuned genetic algorithm for resource investment problems with discounted cash flows and generalized precedence relations", *Computers and Operations Research*, **36**, pp. 2994-3001(2009).

31. Naderi, B., Zandieh, M. and Shirazi, M.A.H.A. "Modeling and scheduling a case of flexible flowshop: Total weighted tardiness minimization", *Computers and industrial Engineering*, **57**, pp. 1258-1267 (2009).

Biographies

Fezzeh Partovi is an MS graduated from the Department of Industrial Engineering, Islamic Azad University of Qazvin, Iran. She received her MS degree in Industrial Engineering in 2013, Qazvin, Iran. Her research interests include Modeling, Optimization, Location and Supply Chain Management.

Mehdi Seifbarghy is a professor at the Department of Industrial Engineering, Alzahra University, Iran, and presently serves as the Head of the Faculty of Technical and Engineering. He received his PhD degree in Industrial Engineering from Sharif University of Technology, Iran. In teaching, he has been focusing on Location and Facility Layout problems and Supply Chain Management. In research, his current interests include Location and Supply Chain Management.