

Dynamic capacitated maximal covering location problem by considering dynamic capacity

Jafar Bagherinejad^{a*} and Mahnaz Shoeb^b

^aAssociate Professor, Department of Industrial Engineering, Alzahra University, Tehran, Iran

^bMSc student of Industrial Engineering, Alzahra University, Tehran, Iran

CHRONICLE

Article history:

Received January 15 2017

Received in Revised Format

April 1 2017

Accepted May 28 2017

Available online

May 29 2017

Keywords:

Capacitated MCLP

Multi-period MCLP

Dynamic capacity

Genetic algorithm

Bee algorithm

ABSTRACT

Capacitated maximal covering location problems (MCLP) have considered capacity constraint of facilities but these models have been studied in only one direction. In this paper, capacitated MCLP and dynamic MCLP are integrated with each other and dynamic capacity constraint is considered for facilities. Since MCLP is NP-hard and commercial software packages are unable to solve such problems in a rational time, Genetic algorithm (GA) and bee algorithm are proposed to solve this problem. In order to achieve better performance, these algorithms are tuned by Taguchi method. Sample problems are generated randomly. Results show that GA provides better solutions than bee algorithm in a shorter amount of time.

© 2018 Growing Science Ltd. All rights reserved

1. Introduction

Facility location problem is a special class of optimization problems whose primary goal is to locate a limited number of facilities that satisfy particular constraints (Máximo et al., 2017). Facility location problems have been studied widely during recent years due to their extensive application in real situations (Correia & Captivo, 2006). Location problems can be defined according to two factors; space (planning area) and time (time of location). Space and time issues have been taken into account in static facility location problems and dynamic facility location problems respectively (Bolori Arabani & Zanjirani Farahani, 2012). Bolori Arabani and Zanjirani Farahani (2012) classified different types of static and dynamic location problems that have been studied by the literature review. They studied multi-period facility location problem as a type of dynamic location problems. Static location problem considers only one period. If a time horizon is considered for more than one period, the location problem becomes dynamic (Canel et al., 2001). By considering a time horizon with more than one period, determining the appropriate time for facility location, specifying the best locations and better prediction of favorable and unfavorable fluctuations of demand in time horizon can be achieved; whereas single period models do not have these characteristics (Miller et al., 2006).

* Corresponding author

E-mail: jbagheri@Alzahra.ac.ir (J. Bagherinejad)

Dynamic models can be classified into two categories: explicitly dynamic models and implicitly dynamic models. In explicitly dynamic models, in order to respond to changes in parameters over time, facilities are closed and opened in pre-specified times and locations. In implicitly dynamic models, all facilities are to be open in the beginning of time horizon and remain open throughout the time horizon. These models are considered to be dynamic because they try to consider changes in parameters such as demand changes over time (Current et al., 1998).

Due to largely capital outlaid, facility location problems are frequently long-term in nature. Facilities such as schools, hospitals and dams operate for decades (Current et al., 1998). While, some facilities such as buses may move around in order to meet the demands of the population (Datta, 2012). Decision makers should select locations and consider time of relocation of facilities which are able to response demand fluctuations over the time (Daskin et al., 1992). Therefore, in order to control probable fluctuations in the future as well as parameters fluctuations a dynamic model seems to be necessary (Bolori Arabani & Zanjirani Farahani, 2012). Theoretically, opening/closing of facilities could impose no cost (Hormozi & Khumawala, 1996). One of the objectives in facility location problem is to minimize the total cost for assigning facilities to satisfy the demand nodes (Jahantigh & Malmir, 2016).

One of the traditional location problems is covering location problem. Covering location problem seeks for a solution to cover a subset of customers by considering one or more objective (Davari et al., 2013). Although covering models are not new, due to their application in real cases especially for emergency service facilities, they have always been attractive topics for researchers (Fallah et al., 2009). Static models can be transformed to their equivalent dynamic models. In these models instead of single period, T periods are considered. Therefore, maximum covering location problem could be studeid as a multi-period and dynamic problem (Bolori Arabani & Zanjirani Farahani, 2012).

The rest of the paper is organized as follows: First, a concise literature review of covering problems and related issues are presented in Section 2. Section 3 is dedicated to the definition of the problem. The proposed solution algorithm is presented in Section 4 and numerical examples and parameter setting appear in Section 5. Moreover, results are analyzed and discussions are given in this section. Finally, to bring the paper to a close, conclusions and outlooks for potential future research are given in Section 6.

2. Literature review

Schilling et al. (1993) classified covering location problems in two categories named maximal covering location problem (MCLP) and set covering location problem (SCLP). In covering problems, a demand is said to be covered if at least one facility is located within a predefined distance of it. This predefined distance is often called coverage radius. The objective of SCLP is to cover all demand with the minimum number of facilities. Covering location problem was introduced for the first time by Hakimi (1965). The objective of that model was to determine the minimum number of polices that was necessary to cover demand nodes on a network of highways. SCLP was introduced formally by Toregas et al. (1971) and was extended slightly by Berlin and Liebman (1974). MCLP was introduced for the first time by Church and Reville (1974). The objective of the MCLP is to locate a fixed number of facilities in such a way that the total covered demand is maximized.

The main assumption of covering location problems is that the facilities are uncapacitated (Salari, 2013). But, practically this assumption is not always valid (Pirkul & Schilling, 1991) and usually limits the applications of covering models (Current & Storbeck, 1988). Most service facilities are capacitated (Murray & Gerrard, 1998; Liao & Approach, 2008). Therefore some covering models considered a capacity constraint for facilities. Although incorporating capacity constraint in formulation of location problems is not difficult, it increases computational complexity. Therefore, most research efforts focus on improvement of solving method (Pirkul & Schilling, 1989). Current and Storbeck (1988) incorporated capacity limitation to MCLP and LSCP. There is a theoretical link between these models and capacitated

plant location problem, the capacitated P-median location problem and generalized assignment problem. Small and moderately sized problems can be solved with existing solution methods. Theoretical links give insight into developing new heuristics for large sized capacitated covering problems. Pirkul and Schilling (1991) developed the model by considering workload limits on the facilities and quality of service delivered to the uncovered demand zones. Facility's workload limits the demand amount which a facility can serve. In this model, all demands are allocated to facilities regardless of whether they are in covering radius or not. The quality of service is modeled as the total distance from uncovered demand zones to the nearest facility. This model is solved by an approach based on the Lagrangian relaxation (Pirkul & Schilling, 1991). Haghani (1996) proposed a capacitated maximal covering location problem in which weighted covered demand is maximized and average distance from the uncovered demands to the located facilities is minimized. They solved the problem with two heuristic methods. The first one was based on greedy adding technique and the second one was based on Lagrangian relaxation. Correia and Captivo (2003) considered modular capacitated location problems. In this model, instead of considering only one fixed capacity level for each facility, they considered a discrete and limited set including available capacity levels. Capacity level of facility is selected from this set. This model can be applied in schools, warehouses and other public facilities. Griffin et al. (2008) proposed capacitated maximal covering location problem by considering three capacity levels for each facility. In their model, there is no composing relationship (such as that between the number of ambulances and emergency stations) between facilities' capacity levels. Yin and Mu (2012) proposed modular capacitated maximal covering location problem (MCMCLP) in two situations. In these models, it is assumed that each facility has a capacity which is related to number of vehicles assigned to that facility. Vehicles have a fixed capacity but the capacity of each facility is equal to the total capacity of vehicles assigned to that facility. In the first model, the number of vehicles is predefined but in the second model, number of vehicles as well as number of facilities is predefined. Yin and Mu (2012) stated that this is a static model and disregard dynamic factors such as daily population movement. On the other hand, since MCMCLP is NP-hard, proposing a heuristic for this problem is important (Yin & Mu, 2012). Although the papers surveyed above, considered capacitated MCLP in only one period, the concept of dynamic (multi-period) covering location problem is not new in the literature (Fazel Zarandi et al., 2013). Schilling (1980) is among the first researchers who considered dynamic maximal covering location problem. Also, other researchers have taken into account this problem. Fazel Zarandi et al. (2013) considered large scale dynamic MCLP. Dell'Olmo et al. (2014) proposed a multi period MCLP for the optimal location of intersection safety cameras on an urban traffic network. Vatsa and Jayaswal (2016) present a new formulation for a multi-period MCLP with server uncertainty, motivated by its relevance with respect to primary health centers.

In this paper, by integrating modular capacitated maximal covering location problem and multi-period maximal covering location problem, a developed model is proposed.

3. Problem definition

In the proposed model, a time horizon consisted of T periods is considered. The objective of this model is to find optimal location of q facilities in a time horizon in such a way that with locating at most p_t vehicles in period t , the maximum covering is achieved in the whole time horizon. This model can be applied in location facilities such as ambulance bases and vehicles such as ambulances. In this model, it is assumed that each vehicle has a fixed capacity (Yin & Mu, 2012) equal to maximum amount of demand that it can serve in each period. Capacity of each facility in each period is related to the number of vehicles stationed in that facility (Yin & Mu, 2012). So, facilities' capacity is changing periodically and we call it as dynamic capacity. For example, if capacity of ambulance is C and Z_{jt} is the number of ambulances located in ambulance base in location j in period t , the capacity of that ambulance base will be CZ_{jt} .

Another assumption is that potential locations are identical in all periods. In each potential location, only one facility can be located and if a facility were located in location j in period t , facility would serve in

this location until the end of the time horizon. In other words, the cost/penalty of closing facilities is so high that prevents closing facilities (for instance buildings such as hospital). Since the importance of costs in public sectors is inconsequential compared to provided services (Fazel Zarandi et al., 2013), it is assumed that opening and closing of vehicles and relocation of them has no cost. Therefore, vehicles are closed at the end of each period and are relocated again in the next period (if they were available).

Since some vehicle may become unavailable in each period because of being out of use, etc. ($d_t < 0$) or some new facilities are added ($d_t > 0$), the number of vehicles are not considered to be identical in all periods. In especial situation, the number of facilities is identical in each period ($p_1 = p_2 = \dots = p_T$). In the proposed model, each facility in each period is as a potential location for stationing of vehicles. If there is no facility in a period, there would be no potential location for stationing of vehicles. Therefore, to maintain the feasibility of the problem, the constraint on the number of vehicles in each period is considered as the maximum number of vehicles. The maximum number of vehicles is given in the beginning of each period and no limitation on the number of vehicles which can be stationed in a facility is considered. In this model, a constraint on the number of facilities is considered in the whole time horizon. If a constraint on the minimum number of new facilities which can be located in period t is not imposed, the minimum number of new facilities in period t will be zero ($m_t = 0$), if a constraint on the maximum number of new facilities which can be located in period t is not imposed, the maximum number of new facilities in period t will be q ($n_t = q$). q is the total number of facilities in the time horizon. In some periods, a constraint on the minimum or maximum number of new facilities located might be imposed in each period. In this situation, the decision maker determines the minimum number of facilities in each period in such a way that sum of minimum number of facilities in the time horizon would not exceed the total number of facilities and considers the maximum number of facilities in each period more than the minimum number of facilities in each period. It is assumed that the minimum and maximum number of new facilities in each period is certain and predefined. It is assumed that at the end of each time horizon, all facilities and vehicles are closed. So, in the beginning of each time horizon, no facilities are located in potential locations ($x_{j0} = 0$). Hereby, the proposed dynamic capacitated MCLP is presented. First, problem parameters and variables are defined.

Sets and parameters

- i, I : The index and set of demand nodes
- j, J : The index and set of eligible facility locations
- t, T : The index and set of time periods
- a_{it} : The population or demand at node i in period t
- d : The Euclidean distance from demand node i to facility at j
- S : The distance (or time) standard within which coverage is desired
- $N = \{j | d \leq S\}$: the set of nodes that are within a distance less than S from node i
- p_t : Maximum number of vehicles in period t
- q : The number of facilities to be located throughout the time horizon
- m_t : Minimum number of new facility in period t ($\sum_{t=1}^T m_t \leq q$)
- n_t : Maximum number of new facility in period t ($n_t \geq m_t$)
- c : capacity of each vehicle

Variables

- x_{jt} : A binary variable which equals one if a facility is sited at location j in period t .
- y_{it} : A binary variable which equals one if node i in period t is covered by one or more facilities stationed within a distance of S .
- z_{jt} : An integer variable which indicates the number of vehicles which are located in period t and site j . ($0 \leq z_{jt} \leq p_t$).

Then, the proposed model will be as follows:

$$\text{Max}Z = \sum_{t=1}^T \sum_{i=1}^I a_{it} y_{it} \quad (1)$$

$$\sum_{j=1}^J \sum_{t=1}^T (x_{jt} - x_{jt-1}) = q \quad (2)$$

$$\sum_{j=1}^J Z_{jt} x_{jt} \leq p_t \quad \text{for } \forall t \quad (3)$$

$$m_t \leq \sum_{j=1}^j (x_{jt} - x_{jt-1}) \leq \min\{n_t, q\} \quad t = 1 \quad (4)$$

$$m_t \leq \sum_{j=1}^j (x_{jt} - x_{jt-1}) \leq \min\left\{n_t, \left(q - \sum_{j=1}^J \sum_{t=1}^{t-1} (x_{jt} - x_{jt-1})\right)\right\} \quad t = 2 \dots T \quad (5)$$

$$Y_{it} a_{it} \leq \sum_{j \in N_i} C X_{jt} Z_{jt} \quad \text{for } \forall t, i \quad (6)$$

$$\sum_{j=1}^J Z_{jt} x_{jt} C \geq \sum_{i=1}^I a_{it} y_{it} \quad \text{for } \forall t \quad (7)$$

$$x_{jt-1} \leq x_{jt} \quad \forall j, t \quad (8)$$

$$Z_{jt} \leq P_t x_{jt} \quad \forall j, t \quad (9)$$

$$x_{jt}, y_{it} \in \{0,1\} \quad \forall j, i, t \quad (10)$$

$$0 \leq Z_{jt} \leq P_t \text{ and int} \quad \forall j, t \quad (11)$$

The objective function (1) maximizes the overall covered demand. Constraint (2) shows that q facilities are to be established in T periods. Constraint (3) specifies the maximum number of vehicles to be located in each period. Constraint (4) assures that if minimum or maximum number of new facilities in period 1 be predefined, the number of new facilities in period 1 ($t=1$) will be in the predetermined interval. Otherwise, it will be between zero and q (the constraint on the minimum and maximum number of new facilities in $t=1$). Constraint (5) specifies that if minimum or maximum number of new facilities for $t>1$ be predefined, the number of new facilities will be in the predetermined interval. Otherwise, it will be between zero and the number of available facilities (number of facilities is not located until period t). Constraint (6) specifies that the demand point i in period t is covered, if it does not exceeds the total capacity of facilities which can cover this demand point. Constraint (7) ensures that the total covered demand in each period cannot exceed total capacity of facilities in that period. Constraint (8) specifies that if a facility is located in period t , it will remain open until the end of the time horizon. Constraint (9) specifies that if a facility is located in period t in site j , the number of vehicles assigned to this facility cannot exceed p_t (the maximum number vehicles in period t). In other words, in each period vehicles can be stationed in site j if a facility is located in that site and this cannot be more than the number of vehicles

predefined for each period. Constraint (10) specifies that decision variables x_{jt} and y_{it} are binary. Constraints (11) restrict the integer decision variable z_{jt} , which ranges from zero to p_t .

Linearization of the proposed model

If we have a non-linear constraint in the form of $y \leq \min(x_1, x_2)$, it could be linearized by definition of a binary variable (δ) and a large enough positive value ($G \geq q$) as follows (linearization of constraints (4) and (5) by Eq. (12-15)):

$$x_1 \leq x_2 + G\delta \quad (12)$$

$$x_2 \leq x_1 + G(1 - \delta) \quad (13)$$

$$y \leq x_2 + G(1 - \delta) \quad (14)$$

$$y \leq x_1 + G\delta \quad (15)$$

By considering $z_{jt}x_{jt} = O_{jt}$, constraints (3), (6) and (7) are linearized as follows (linearization of constraints (3), (6) and (7) by Eq. (16-19)):

$$O_{jt} \leq z_{jt} \quad (16)$$

$$O_{jt} \geq z_{jt} - p_t \times (1 - x_{jt}) \quad (17)$$

$$O_{jt} \leq p_t \times x_{jt} \quad (18)$$

$$0 \leq O_{jt} \leq p_t, \text{ int} \quad (19)$$

4. Solution methods

4.1. Genetic algorithm (GA)

4.1.1. Review of GA

GA was first proposed by Holland (1975) as an evolutionary algorithm. It is based on Darwinian evolution: good traits survive and mix to form new while the bad traits are eliminated from the population (Zanjirani Farahani & Hekmatfar, 2009). Beasley and Chu (1996) seem to be the first to apply GA for covering model. The Simple GA is as follows:

1. Generate an initial population mostly in a random way.
2. Select individuals for reproduction.
3. Perform genetic operations and generate a new generation.
4. Insert offspring into population and form the new population.
5. If the predefined stopping criteria are met, stop the algorithm, otherwise, return to step 2.

The rest of this section is devoted to elaboration of the proposed GA.

4.1.2. Encoding scheme

An appropriate chromosome representation must be defined for a GA. Encoding very depends on the problem. Most previously adopted representations, such as the bit string, are linear or one-dimensional. Some real problems are naturally suitable for two-dimensional representation. In this paper, GA with

multi chromosome representation is applied and a separate chromosome is considered for each variable (Matthias et al., 2013). Therefore, three chromosomes are defined which are two-dimensional. The first chromosome is binary matrix which has I rows and T columns. Each bit in this matrix represents the status (covered/uncovered) of the node i in period t . The second chromosome is a binary matrix which has J rows and T column and each bit in this matrix represents the status (located/unlocated) of facility at location j in period t . In other words, one value in the i th bit means that there is a facility at location j in period t . The third chromosome is a matrix consisted of integer number which has J rows and T column and each bit in this matrix represents the number of vehicles at location j in period t . Each bit of third chromosome is an integer number between zero and p_t in such a way that total number of vehicles in each period is not more than p_t . Therefore, an upper and lower bound are considered for the number of vehicles in each period. At first, an initial population is generated randomly for each chromosome.

4.1.3. Selection

Selection is the stage of GA in which chromosomes are selected from the population to be parented for the next generation. There are many methods in selecting the best chromosomes. In this paper, the Roulette Wheel Selection (RWS) has been used as the selection method. Each individuals of the population is allocated a section of an imaginary roulette wheel. The size of each sections is directly proportional to their fitness values, such that the fittest individual has the biggest slice of the wheel and the weakest individual has the smallest. The wheel is then spun and the individual associated with the winning section is selected.

4.1.4. Crossover

Crossover and mutation are two basic genetic operators used to make new off springs. Type and implementation of operators depend on encoding and also on a problem. In this paper, crossover in first and second chromosomes is performed using one of these two methods: one-point crossover and two-point crossover. But in the third chromosome, the arithmetic crossover is applied. Fig.1 shows different types of crossover.

4.1.4.1. One-point crossover

One-point crossover randomly select a crossover point and then copy everything before this point from the first parent and then everything after the crossover point copy from the second parent. Here chromosomes are two-dimensional and crossover point can be a bit or column. In one-point crossover (column), third column in each parent is selected randomly and exchanged. In one-point crossover (bit), two bits in each parent selected randomly and exchanged (Fig.1).

4.1.4.2. Two-point crossover

Two-point crossover randomly selects two crossover points within a chromosome then interchanges the two parent chromosomes between these points to produce two new offspring. In Fig.1, in the case of two point crossover first and forth columns in each parent are selected and they are swapped.

4.1.4.3. Arithmetic crossover

Arithmetic crossover is a linear combination of two chromosomes as follows:

$$C_i^{new} = \alpha \cdot C_i^{old} + (1 - \alpha)C_j^{old} \quad (20)$$

$$C_j^{new} = (1 - \alpha)C_i^{old} + \alpha C_j^{old}, \quad 0 \leq \alpha \leq 1 \quad (21)$$

where C^{old} is a parent, C^{new} is an offspring, and α is random matrix (between 0 and 1). This operation is performed for each bit (Köksoy & Yalcinoz, 2008). In Fig.1, random matrix is given. The number of bits are selected randomly. According to the Eq. (20) and Eq. (21) arithmetic crossover is operated.

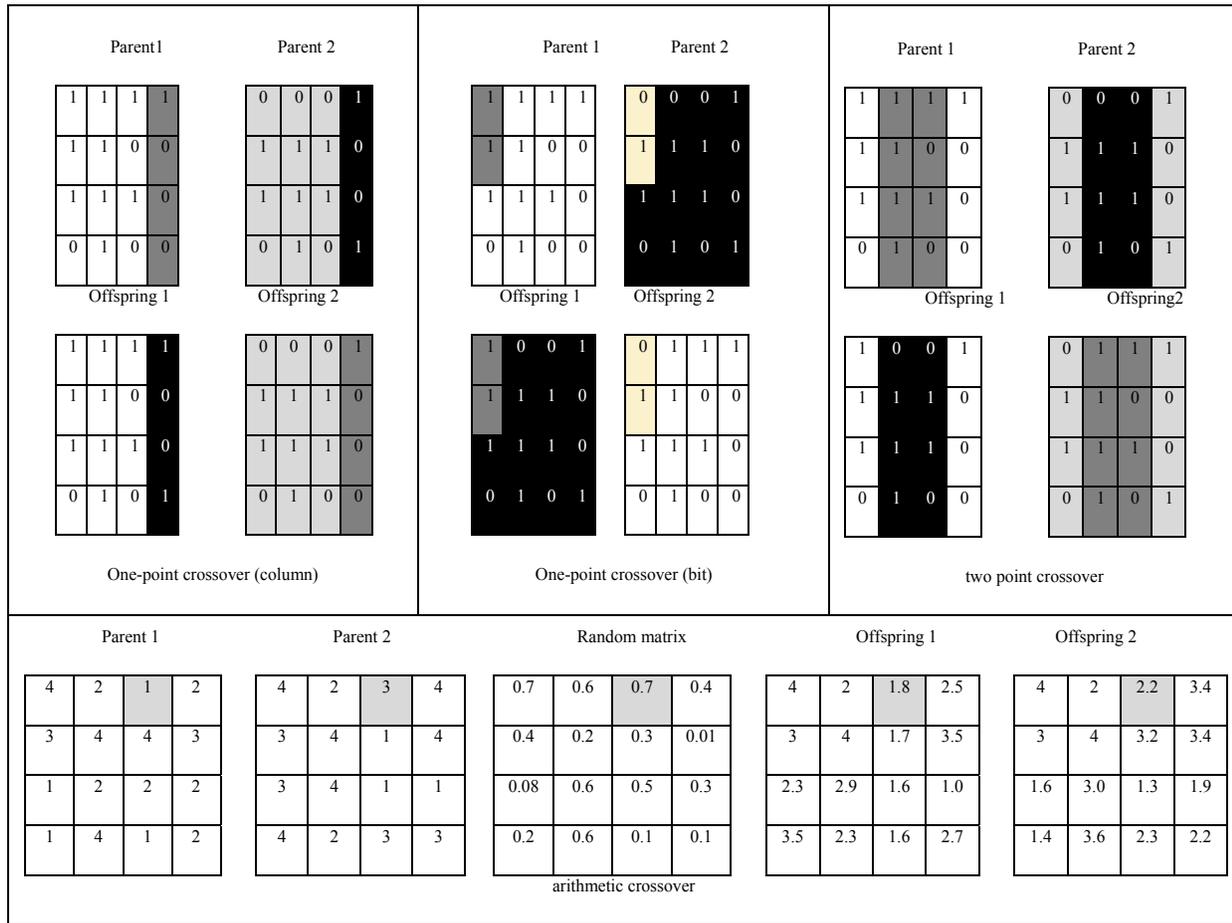


Fig. 1. Crossover

4.1.5. Mutation

Mutation is the genetic operator that randomly changes one or more of the chromosome's gene. The purpose of the mutation operator is to prevent the genetic population from converging to a local minimum and to introduce to the population of new possible solutions. The mutation is carried out according to the mutation probability. Mutation rate is usually set to a very low level. However, different references have found that a higher mutation rate is necessary when the GA has converged (Jaramillo et al., 2002). In this paper, crossover in first and second chromosomes is performed using one of these methods: swap, binary and reversion mutations. In third chromosome, the mutation method is only arithmetic mutation. Fig. 2 shows different types of mutation.

4.1.5.1. Binary mutation

In Binary mutation, some bits are selected randomly and their values are changed from zero to one and one to zero. In Fig. 2, in case of binary mutation, colored bits are selected randomly and binary mutated.

4.1.5.2. Swap mutation

In Swap mutation, two bits are selected randomly and are exchanged with each other. In Fig.2, in the case of swap mutation, colored bits are selected randomly and swapped.

4.1.5.3. Reversion mutation

In reversion mutation, two bits are selected randomly and bits between them are reversed. In Fig. 2, in the case of reversion mutation, colored bits are selected randomly and reversed.

4.1.5.4. Arithmetic mutation

The definition of arithmetic mutation is as follows:

$$\begin{cases} C_i^{new} = C_i^{old} + \alpha_i & \text{If } C_i^{new} \leq C_i^{old} + \alpha_i \\ u_i & \text{If } C_i^{new} > C_i^{old} + \alpha_i \end{cases} \quad (22)$$

where u_i is upper bound, C^{old} is a parent, C^{new} is an offspring, and α is random matrix (between 0 and u_i). At each iteration, offspring as many as the population size are created. In order to produce new generation, these offspring replace less fit individuals in the existing population. In Fig. 2, in the case of arithmetic mutation, upper bound and random matrix is given ($u_i = 10$). Certain number of bits are selected randomly. According to the Eq. (22) arithmetic mutation is operated.

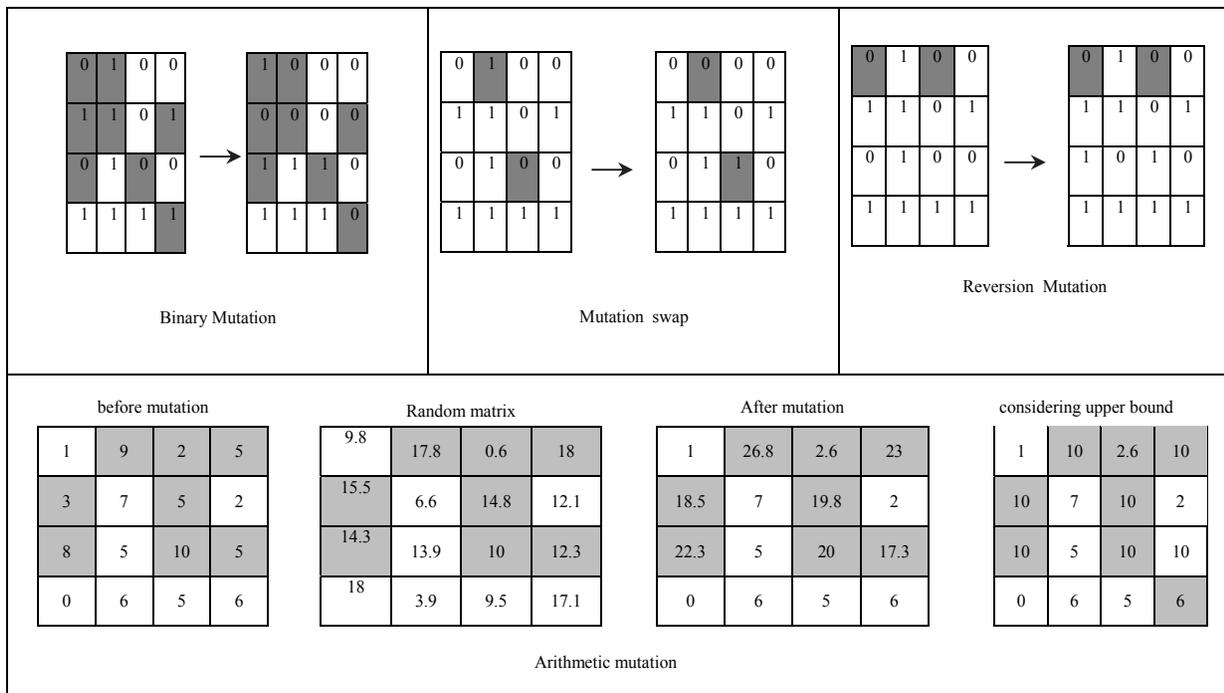


Fig. 2. Mutation

4.1.6. Termination criteria

The algorithm will iterate until the maximum number of iterations is attained.

4.2. Bee algorithm

4.2.1. Review of Bee algorithm

Bee algorithm is one of the swarm-based algorithms which imitate the food foraging behavior of

honeybee swarms (Tsai, 2014). The basic version of the algorithm performs a kind of local search combined with random search and it can be used for combinatorial optimization and functional optimization (Pham et al., 2006). The foraging process begins in a colony by scout bees being sent to search for promising flower patches. At first, these bees spread out in flower patches randomly. Each selected food source represents a feasible solution of the problem. The nectar amount in the food source represents the quality of the problems' solution. When the foraging process is finished, based on the nectar amount, food sources are ranked into three categories: elite, good (average) and bad (unselected) sites. Each scout bee performs a dancing known as the waggle dance above a certain quality threshold deposit their nectar. This way, it transfers information of that region (in comparison to the hive), its distance from the hive and its quality rating to other bees. This information helps the colony to send follower bees to flower patches. Most follower bees go to region where are more promising to find nectar. In other words, more follower bees are assigned to elite sites than those of good sites. Assigning follower bees is as generating a neighbor for a solution. In each iteration of the algorithm for each elite site, a certain number of neighbors are generated. In this paper, the method for generating neighbors is similar to mutation methods in GA. The fitness of these neighbors is calculated and for each site the best neighbor is recognized. The quality of the best neighbor is compared to that of current bee. If the quality of the best neighbor bee is better than that of current bee, it will be replaced. Otherwise, current bee stays in its site with no movement. This process is performed on the good site. The difference is that the number of neighbors generated for good site is less than those of elite sites. Bad sites are abandoned and scout bees assigned to bad sites are assigned to other sites randomly. Therefore, the new population is generated. This process continues until the stop criteria is met. In this algorithm, chromosomes defined in GA are considered as bee and Mutation operator is performed to generate neighbors. Bee algorithm is as follow:

1. The bees of initial population are randomly generated (n : number of initial bees).
2. The fitness are calculated.
3. Certain number of best bees (m) are selected to finding neighborhood. Among selected bees (m), a certain number of them (e) are considered as elite bees. The rest of them ($m-e$) are good bees.
4. By mutation method, the neighborhood are generated for elite and good bees and their fitness are calculated. The number of neighborhood are generated for each elite bees are more than good bees.
5. Among neighborhoods of each bees, the best bee is selected. These bees are transferred to the next generation.
6. Good bees ($n-m$) are used to random search and their fitness are calculated. These bees also transmitted to the next generation.
7. If a stop condition is met, the algorithm stops. Otherwise, go to step 3.

4.2.2. Termination criteria

The algorithm will iterate until the maximum number of iterations is attained.

5. Numerical examples

5.1. Test problems

To generate test problems, a similar approach to Revelle et al. (2008) was employed. In this approach, the locations of nodes were randomly generated using a uniform distribution between 0 and 30 for both x and y coordinates. The distances between the nodes are then defined as their Euclidean distance. Populations on the demand nodes for each time period were randomly generated using a uniform distribution between 0 and 100. Fazel Zarandi et al. (2013) by considering 5 periods use this approach for generating dynamic sample problems. This paper by considering time scale uses this method for generating 30 sample problems. The minimum number of new facilities in each period is randomly generated using a uniform distribution between 0 and q in such a way that sum of minimum number of new facilities in the whole planning horizon be less than or equal to q ($\sum_t^T m_t \leq q$). The maximum number of new facilities in each period is generated by random numbers larger than or equal to the

minimum number of new facilities in each period ($n_t \geq m_t$). Lingo 8.0 was used to solve these problems and results were compared against those obtained using the GA and bees algorithm.

5.2. Parameter setting

Heuristic and metaheuristic algorithms are sensitive to their parameters; A small change can affect the quality of the solution. So, tuning algorithms are necessary to find a better solutions (Pasandideh et al., 2015). The most widely used method to tune the algorithms is a full factorial design (Chan et al., 2015). This method does not seem effective when the number of parameters significantly increases, since it requires arduous task to conduct the experiment. A family of matrices is used to reduce the number of experiments. In Taguchi method, we utilize the orthogonal arrays to investigate a large number of decision variables with a small number of experiments (Raju et al., 2014).

In this method, factors are classified into two groups: Controllable factors (signal) and uncontrollable factors (noise). Also, objective functions are categorized into three groups: “the smaller the better”, “the larger the better” and “the nominal value is expected”. The objective functions of the proposed model is “the larger the better”. S/N ratio (the large the better) is calculated by Eq. (23).

$$S/N \text{ ratio} = -10 \log \left(\frac{1}{n} \sum_{i=1}^n \frac{1}{x_i^2} \right), \tag{23}$$

where x_i = observed response value and n = number of replications. According to Taguchi's design of experiments, for 4 parameters and 3 levels L_9 Taguchi orthogonal array was selected (Table 1 and Table 2). For calibration of each algorithm, 6 sample problems are iterated for 5 times in each scenario. Since sample problems' dimension is not identical, so the differences between their objective functions are large and using raw data cause to wrong analyses. In other words, the dimension of the problem should be excluded form data. So after conversion of raw data to relative deviation index (RDI), the S/N ratio is calculated. Relative deviation index is calculated by Eq. (24):

$$RDI_{ijk} = \begin{cases} \left| \frac{OF_{ijk} - u_i}{u_i - l_i} \right| & u_i \neq l_i \\ 0 & u_i = l_i \end{cases} \quad i = 1..6, \quad j = 1..5, \quad k = 1..9 \tag{24}$$

where OF_{ijk} is the objective function value related to iteration j in sample problem i in scenario k . l_i and u_i are minimum and maximum values for i th sample problem respectively. The S/N rate for scenario k can be calculated (using relative deviation index of objectives function) by Eq. (25).

$$S/N_k = \begin{cases} -10 \log \left(\frac{1}{n} \sum_{i=1}^I \sum_{j=1}^J \frac{1}{RDI_{ijk}^2} \right) & RDI_{ijk} \neq 0 \quad k = 1 \dots 9 \\ 0 & RDI_{ijk} = 0 \end{cases} \tag{25}$$

In Taguchi method, S/N rate is considered as the first criterion. There could be no meaningful difference between different S/N levels, so, another criterion named RDI_k is introduced for scenario k which is calculated by Eq. (23). \overline{RDI}_k is considered as the smaller the better.

$$\overline{RDI}_k = \frac{1}{n} \sum_{i=1}^I \sum_{j=1}^J RPI_{ijk} \quad k = 1 \dots 9 \tag{26}$$

It is time consuming to set all effective parameters in bee algorithm. Therefore, we set the most important effective parameters and other parameters have been determined by try and error. According to S/N (Fig. 3 and Fig. 5) and RDI (Fig. 4 and Fig. 6) the selected levels are colorful.

Table 1
Parameter and their levels in GA

Parameters	Level 1	Level 2	Level 3
% crossover	0.6	0.65	0.7
% mutation	0.2	0.25	0.3
Population size	50	100	150
Max iteration	30	50	100

Table 2
Parameters and their levels in Bee algorithms

Parameters	Level 1	Level 2	Level 3
% good site	0.6	0.65	0.7
% elite site	0.01	0.1	0.3
n scout bee	10	50	80
Max iteration	20	30	50

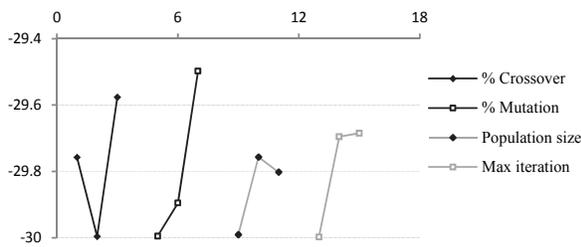


Fig. 3. S/N ratio (GA)

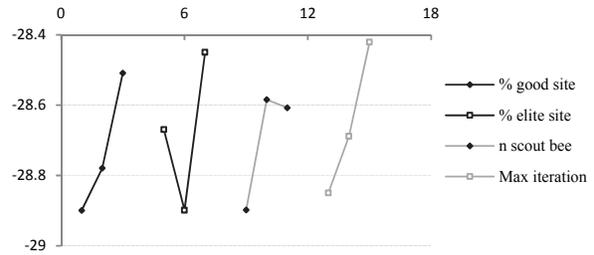


Fig. 5. S/N ratio (bee algorithm)

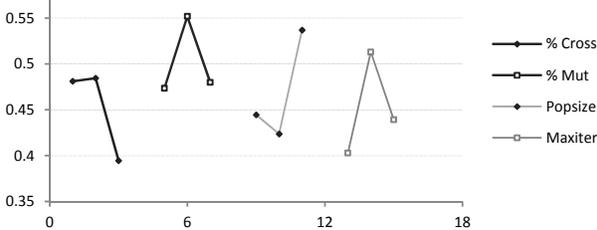


Fig. 4. RDI (GA)

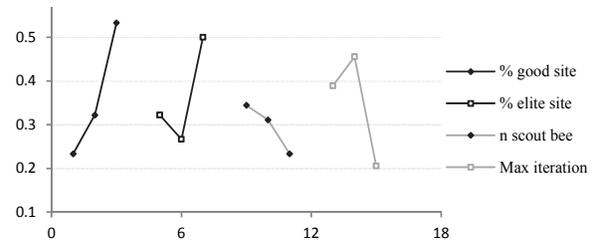


Fig. 6. RDI (bee algorithm)

5.4. Results and discussions

In this paper, Lingo software package has been applied to find the exact solutions for some sample problems. Lingo uses branch and bound method to solve the problems. Objective bound specifies the theoretical bound of objective function. This bound specifies how much a solver can improve the objective function. The best solution cannot exceed the objective bound. Colored rows specify problems in which Lingo cannot find the optimal solution in one hour. In such cases instead of optimal value, the objective bound and the best solution found in one hour is reported (Niroomand, 2008). In such cases, metaheuristic/heuristic algorithms might find better solution than what lingo finds in one hour. In such situation, the gap will be negative (Mehdizadeh & Afrabandpei, 2012). Gap is calculated as follow:

$$Gap = \frac{Exact\ solution - Metaheuristic\ solution}{Exact\ solution} \times 100 \tag{27}$$

According to computational results of 30 sample problems, it could be concluded that Lingo can find the optimal solution for only one thirds of the problems. In half of the sample problems, GA finds a solution

better than or equal to what lingo finds in one hour ($Gap_{GA} < 0$). In other problems except one problem, GA can achieve a gap less than 1.9%. The run time of GA in the largest problems is less than 2.5 minutes. Bee algorithm finds a solution better than or equal to what lingo finds in one hour in less than half of the sample problems ($Gap_{BA} < 0$). In other problems except two problems, Bee algorithm can achieve a gap less than 2.9%. The run time of Bee algorithm is less than 4 minutes in the largest problems. Totally, it can be stated as following:

$$Time_{GA} < Time_{BA}$$

$$Gap_{GA} \leq Gap_{BA}$$

Therefore, GA can find better solution in a shorter time. The average run time and objective function value in 5 iterations is reported in Table 3.

Table 3
Computational results

Sample problems			Lingo			GA			BA		
I	J	T	Z*	Objective bound	Time (s)	Z*	Time (s)	Gap (%)	Z*	Time (s)	Gap (%)
100	50	3	550	-	197	548.8	14.33	0.21	545.2	21.96	0.87
		5	950	-	316	932.6	22.85	1.83	922.2	35.11	2.92
		7	1297	1300	2600	1271.2	35.03	1.98	1240.4	48.38	4.36
100	100	3	550	-	337	547.8	15.09	0.4	547	23.29	0.54
		5	1000	-	2816	991.8	24.17	0.82	986.2	41.32	1.38
		7	1177	1200	3600	1163.8	33.26	1.12	1141.8	50.29	2.99
150	150	3	550	-	319	549	25.18	0.18	546.6	35.55	0.61
		5	999	1000	3600	992.2	40.93	0.68	985.6	56.97	1.34
		7	1119	1200	3600	1144.6	58.09	-2.28	1160.4	79.24	-3.69
200	100	3	550	-	548	549.6	28.68	0.07	548.6	43.31	0.25
		5	996	1000	3600	995.2	46.85	0.08	987.4	70.82	0.86
		7	1099	1200	3600	1150.4	65.10	-4.67	1145	98.96	-4.18
200	200	3	550	-	945	549.2	35.94	0.14	546.6	49.63	0.61
		5	986	1000	3600	992.6	58.79	-0.66	991.2	81.25	-0.52
		7	1108	1200	3600	1141.6	83.86	-3.03	1147.8	112.94	-3.59
250	250	3	550	-	1153	549.8	50.86	0.036	549.4	70.18	0.10
		5	888	1000	3600	996.4	83.47	-12.2	991.2	115.8	-11.62
		7	1096	1200	3600	1169.2	116.99	-6.67	1156.2	162.06	-5.49
300	100	3	550	-	1193	549.2	43.13	0.14	547.6	65.42	0.43
		5	999	1000	3600	995.6	57.34	0.34	982.6	107.89	1.64
		7	1195	1200	3600	1153.4	108.97	4.98	1132	150.01	5.27
300	200	3	549	550	3600	549.2	52.44	-0.03	549	77.69	0
		5	879	1000	3600	989.6	86.79	-12.58	981	128.81	-11.60
		7	1027	1200	3600	1147.8	121.14	-11.76	1143.2	180.71	-11.31
300	250	3	548	550	3600	550	68.03	-0.36	548.8	91.92	-0.14
		5	926	1000	3600	992.4	117.26	-7.17	991.6	151.01	-7.08
		7	993	1200	3600	1166	151.51	-17.42	1161.4	211.24	-16.95
300	300	3	550	-	1760	550	70.71	0	549	104.14	0.18
		5	980	1000	3600	991.8	117.34	-1.20	993	137.22	-1.32
		7	1037	1200	3600	1161	131.00	-11.95	931.8	192.7	10.14

C=10, S=10, q=10, $p_t \in \{10, 20, 25\}$

6. Conclusion and future research areas

In this paper, capacitated MCLP and dynamic MCLP were integrated to each other and dynamic capacity constraint was considered for facilities. Therefore, the MCLP has been extended to the capacitated dynamic MCLP. The developed model was solved by GA and bee algorithm and the results were compared to the exact solutions of Lingo. We have shown that while GA and bee algorithm are superior to the exact method in terms of runtime, there are negligible errors compared to the optimal solutions. GA found better solutions in a shorter amount of time than the bee algorithm. Although GA shows great performance to solve this model, one may assess the performance of other methods in finding solutions to this problem. Another opportunity for research is to add a constraint on the number of vehicles which can be located in each facility. A possible future study could be to integrate this model with gradual covering location problem. Another future research is to consider cost for each vehicle. Cost can be dynamic and changes in each period. Objective function could be maximization covered demands while

the cost of vehicles is minimized. Some parameters can be fuzzy such as covering radius. Covering radius can be dynamic, too.

Acknowledgement

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

- Beasley, J. E., & Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2), 392–404.
- Berlin, G. N., & Liebman, J. C. (1974). Mathematical analysis of emergency ambulance location. *Socio-Economic Planning Sciences*, 8(6), 323–328.
- Bolori Arabani, A., & Zanjirani Farahani, R. (2012). Facility location dynamics: An overview of classifications and applications. *Computers & Industrial Engineering*, 62(1), 408–420.
- Canel, C., Khumawala, B. M., Law, J., & Loh, A. (2001). An algorithm for the capacitated, multi-commodity multi-period facility location problem. *Computer & Operation Research*, 28(5), 411–427.
- Chan, K. Y., Rajakaruna, N., Engelke, U., Murray, I., & Abhayasinghe, N. (2015). Alignment parameter calibration for IMU using the Taguchi method for image deblurring. *Measurement*, 65(Apr 2015), 207–219.
- Church, R., & Reville, C. (1974). The maximal covering location problem. *Papers in Regional Science*, 32(1), 101–118.
- Correia, I., & Captivo, M. E. (2003). A lagrangean heuristic for a modular capacitated location problem. *Annals of Operations Research*, 122(1-4), 141–161.
- Correia, I., & Captivo, M. E. (2006). Bounds for the single source modular capacitated plant location problem. *Computers & Operations Research*, 33(10), 2991–3003.
- Current, J. R., & Storbeck, J. E. (1988). Capacitated covering models. *Environment and Planning B: Planning and Design*, 15(2), 153–163.
- Current, J., Ratick, S., & Reville, C. (1998). Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. *European Journal of Operational Research*, 110(3), 597–609.
- Daskin, M. S., Hopp, W. J., & Medina, B. (1992). Forecast horizons and dynamic facility location planning. *Annals of Operations Research*, 40(1), 125–151.
- Datta, S. (2012). Multi-criteria multi-facility location in Niwai block, Rajasthan. *IIMB Management Review*, 24(1), 16–27.
- Davari, S., Fazel Zarandi, M. H., & Turksen, I. B. (2013). A greedy variable neighborhood search heuristic for the maximal covering location problem with fuzzy coverage radii. *Knowledge-Based Systems*, 41(March 2013), 68–76.
- Dell’Olmo, P., Ricciardi, N., & Sgalambro, A. (2014). A multiperiod maximal covering location model for the optimal location of intersection safety cameras on an urban traffic network. *Procedia-Social and Behavioral Sciences*, 108, 106–117.
- Fallah, H., Naimi Sadigh, A., & Aslanzadeh, M. (2009). *Covering problem*, in: Zanjirani Farahani, R., & Hekmatfar, M. (Eds.), *Facility Location: Concepts, Models, Algorithms and Case studies*. Berlin: Springer-Verlag, pp. 145–176.
- Fazel Zarandi, M. H., Davari, S., & Haddad Sisakht, S. A. (2013). The large-scale dynamic maximal covering location problem. *Mathematical and Computer Modelling*, 57(3), 710–719.
- Griffin, P. M., Scherrer, C. R., & Swann, J. L. (2008). Optimization of community health center locations and service offerings with statistical need estimation. *IIE Transactions*, 40(9), 880–892.
- Haghani, A. (1996). Capacitated maximum covering location models: Formulations and solution procedures. *Journal of Advanced Transportation*, 30(3), 101–136.

- Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3), 462-475.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with application to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press.
- Hormozi, A. M., & Khumawala, B. M. (1996). An improved algorithm for solving a multi-period facility location problem. *IIE transactions*, 28(2), 105-114.
- Jahantigh, F. F., & Malmir, B. (2016, March). A Hybrid Genetic Algorithm for Solving Facility Location Allocation Problem. In *Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia*.
- Jaramillo, J. H., Bhadury, J., & Batta, R. (2002). On the use of genetic algorithms to solve location problems. *Computers & Operations Research*, 29(6), 761-779.
- Köksoy, O., & Yalcinoz, T. (2008). Robust design using pareto type optimization: A genetic algorithm with arithmetic crossover. *Computers & Industrial Engineering*, 55(1), 208-218.
- Liao, A., & Approach, D. G. (2008). A clustering-based approach to the capacitated facility location problem. *Transactions in GIS*, 12(3), 323-339.
- Matthias, K., Severin, T., & Salzwedel, H. (2013). Variable mutation rate at genetic algorithms: Introduction of chromosome fitness in connection with multi-chromosome representation. *International Journal of Computer Applications*, 72(17), 31-38.
- Máximo, V. R., Nascimento, M. C., & Carvalho, A. C. (2017). Intelligent guided adaptive search for the maximum covering location problem. *Computers & Operations Research*, 78(Feb 2017), 129-137.
- Mehdizadeh, E., & Afrabandpei, F. (2012). Design of a mathematical model for logistic network in a multi-stage multi-product supply chain network and developing a metaheuristic algorithm. *Journal of Optimization in Industrial Engineering*, 5(10), 35-43.
- Miller, T. C., Friesz, T. L., Tobin, R. L., & Kwon, C. (2006). Reaction function based dynamic location modeling in Stackelberg-Nash-Cournot competition. *Networks and Spatial Economics*, 7(1), 77-97.
- Murray, T., & Gerrard, R. A. (1998). Capacitated service and regional constraints in location-allocation modeling. *Location Science*, 5(2), 103-118.
- Niroomand, I. (2008). *Modeling and analysis of the generalized warehouse location problem with staircase costs* (Doctoral dissertation, Concordia University Montreal, Quebec, Canada).
- Pasandideh, S. H. R., Akhavan Niaki, S. T., & Asadi, K. (2015). Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: NSGA-II and NPGA. *Information Sciences*, 292(Jan 2015), 57-74.
- Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2011, July). The bees algorithm-A novel tool for complex optimisation. In *Intelligent Production Machines and Systems-2nd I* PROMS Virtual International Conference (3-14 July 2006)*. sn.
- Pirkul, H., & Schilling, D. (1989). The capacitated maximal covering location problem with backup service. *Annals of Operations Research*, 18(1), 141-154.
- Pirkul, H., & Schilling, D. A. (1991). The maximal covering location problem with capacities on total workload. *Management Science*, 37(2), 233-248.
- Raju, B. S., Shekar, U. C., Venkateswarlu, K., & Drakashayani, D. N. (2014). Establishment of Process model for rapid prototyping technique (Stereolithography) to enhance the part quality by Taguchi method. *Procedia Technology*, 14(Jan 2014), 380-389.
- Revelle, C., Scholssberg, M., & Williams, J. (2008). Solving the maximal covering location problem with heuristic concentration. *Computers & Operations Research*, 35(2), 427-435.
- Salari, M. (2013). An iterated local search for the budget constrained generalized maximal covering location problem. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 13(3), 301-313.
- Schilling, D. A. (1980). Dynamic location modeling for public-sector facilities: A multicriteria approach. *Decision Sciences*, 11(4), 714-724.
- Schilling, D. A., Jayaraman, V., & Barkhi, R. (1993). A review of covering problem in facility location. *Location Science*, 1(1), 25-55.

- Toregas, C., Swain, R., ReVelle, C., & Bergman, L. (1971). The location of emergency service facilities. *Operations Research*, 19(6), 1363-1373.
- Tsai, H. (2014). Novel bees algorithm: Stochastic self-adaptive neighborhood. *Applied Mathematics and Computation*, 247(Nov 2014), 1161–1172.
- Vatsa, A. K., & Jayaswal, S. (2016). A new formulation and Benders decomposition for the multi-period maximal covering facility location problem with server uncertainty. *European Journal of Operational Research*, 251(2), 404-418.
- Yin, P., & Mu, L. (2012). Modular capacitated maximal covering location problem for the optimal siting of emergency vehicles. *Applied Geography*, 34(May 2012), 247–254.
- Zanjirani Farahani, R., & Hekmatfar, M. (2009). *Facility location: Concept, Models, Algorithms and Case Studies*. Heidelberg: Physica-Verlag.



© 2017 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).