International Journal of Supply and Operations Management

IJSOM

February 2017, Volume 4, Issue 1, pp. 90-104
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com

Decentralized Fuzzy P-hub Centre Problem: Extended Model and Genetic Algorithms

Sara Mousavinia [a, *], Majid Khalili [a], Mohammad Shafiee [a]

[a] *Department of industrial engineering, Islamic Azad University of Karaj, Karaj, Iran*

Abstract
This paper studies the incapacitated P-hub centre problem in a network under decentralized management assuming time as a fuzzy variable. In this network, transport companies act independently, each company makes its route choices according to its own criteria. In this model, time is presented by triangular fuzzy number and used to calculate the fraction of users that probably choose hub routes instead of direct routes. To solve the problem, two genetic algorithms are proposed. The computational results compared with LINGO indicate that the proposed algorithm solves large-scale instances within promising computational time and outperforms LINGO in terms of solution quality.

Keywords: Decentralized management; Fuzzy number; Genetic algorithm; P-hub network; Hub location problem.

1. Introduction

Hub location problems (HLP) are of great importance in the field of operation management with a wide range of applications and ever-growing body of literature. (See for example Zarrinpour 2011, Alumur 2012, Campbell 2012, Hernandez 2012). Zanjirani Farahani et al. (2013) provided a complete and detailed discussion of the various aspects of the problem and the state of the art of its different formulations and solution methods.

Table1. Different types of HLPs (Zanjirani Farahani et al. (2013))

Capacity of hub node	Assignment of non-hub node to hub nodes	Type of the HLP	Number of hub nodes
Capacitated (C)	Single allocation (SA)	Median (M)	Single (1)
Incapacitated (U)	Multiple allocation (MA)	Center (T)	More than one (P)
		Covering (V)	
		Set covering (SV)	
		Maximum covering (MV)	

As it was marked in table.1, the problem we discussed in this paper is based on Incapacitated Multiple allocation P hub centre problem. So, in the rest of this section, the recent studies pertinent to our studied DFUMHLP[1]will be reviewed.

According to Zanjirani et al. (2013), majority of studies have dealt with incapacitated cases of HLPs. One of the most recent studies in this area has been done by O'Kelly et al. (2014) who formulated a model to analyse the role of fixed costs in the design of optimal transportation hub networks. Campbell et al. (2015) presented a new model for hub location and network design that uses fixed and variable transportation costs on all arcs, fixed costs for hubs, and also allowed direct arcs.

[1] Decentralized Fuzzy Incapacitated Multiple Hub Location Problem

***Corresponding author email address:** sara.mousavinia@gmail.com

In real world applications, all the parameters of a network may not be known precisely due to uncontrollable factors. Because due to rapid changes, lack of data, and incomplete and/or noisy factors in the available information, if any decision is made based on the deterministic models, demands may not be reached at the right location, at the right time, and at the best costs (2013b). Yang et al. (2011) studied the p-hub centre problem with discrete random travel time. Qin and Gao (2014) formulated a new incapacitated p-hub location model with flows described by uncertain variables. Hult et al. (2014) proposed a reformulation for the p-hub centre problem when the uncertainty of travel times was considered. In short, these studies point out that if a decision maker ignores the uncertainty, it causes huge regrets in long run (2013a).

For many cases, the estimations of probability distributions for decision factors may not be easy due to the lack of data. So this type of imprecise data has not always been well represented by random variable selected from a probability distribution. This kind of data can effectively be presented by fuzzy features (Kaur and Kumar 2011). Yang et al. (2013) presented a new risk aversion p-hub centre problem with fuzzy travel times. Nematian (2016) presented an incapacitated p-hub center problem in case of single allocation and also multiple allocations in which travel times or transportation costs were considered as fuzzy parameters.

Both the facility location and network design problems as sub problems of the facility location–network design problem are NP-hard (Ghaderi2013, Rabbani2015). So, UMHLP is known to be NP-hard, with exception of special cases, for example when matrix of flows W_{ij} is sparse (Kratica 2005). Even though integer programming optimization approaches are applied to solve small hub problems, larger instances of HLPs need to be solved by heuristic procedures or meta-heuristic procedures. As a matter of fact, while large-size instances can be dealt with specialized exact methods (e.g., benders decomposition and branch and price methods), development of meta-heuristics has helped many real-world applications, in which optimal/near-optimal solutions can even be obtained in less computational time (Zanjirani et al. 2013).

The most related and recent GA solution approaches are summarized in table2.

Table 2. Genetic algorithms in HLPs

Article	**Problem**	**Solution algorithm**
Kratica et al. (2005)	incapacitated	genetic algorithm- applied caching technique
Topcuoglu et al. (2005)	incapacitated	genetic algorithm
Eraslan S. (2010)	incapacitated	genetic algorithm
Bashiri et al. (2013)	capacitated	genetic algorithm- hybrid approach
Yang et al. (2013)	no capacities involved	genetic algorithm- hybrid approach
Rabbani et al. (2015)	incapacitated	genetic algorithm and simulated annealing

According to the literature of HLP solution approaches, we think that UMHLP problem under decentralized management has not yet been solved for large-sized scale problems, but it is a very common case in the analysis of networks of regional or greater scope. Besides, few studies have considered hub location problems with uncertain parameters (Contreras 2011) so; another contribution of this paper is that we considered travel time in the network as a fuzzy parameter in order to study the network in a more real situation. It may contribute to estimate the network time and cost more accurately. The existing model is only applicable to networks by the deterministic factors.

In this paper, we are aimed to develop the mathematical formulation of the Vosconcelos's problem (2011)- incapacitated multiple allocation p-hub location problem under decentralized management- with uncertainty in travel time parameter. This parameter is characterized by a triangular fuzzy number while the objective function of this problem minimizes the expected costs. Then we present a novel solution based on a genetic search framework for DFUMHLP. We compared the quality of solutions from our method by comparing the both solutions; by exact time factor and fuzzy time factor. To demonstrate the effectiveness of our method, we compared the quality of solutions from our method and GA algorithm with the solutions from LINGO for small-sized networks. The results showed that LINGO solver fails to efficiently solve large and complicated instances.

The rest of this paper is organized as follows. In Section 2, we extend the Vosconcelos's model. The solution approach and GA operators are discussed in Section 3. Computational results are reported in section 4 by using the proposed algorithm. Finally, Section 5 covers the conclusions.

2. Mathematical formulation

The studied problem deals with a network with n nodes in which there are some pre-established hubs ($H^* \in N$) and P hubs to be established. The traffic between any pair of non-hub nodes is routed via one or two hubs at most, and the multiple allocation schemes are assumed. In this model, a non-limited amount of flow can be collected in a hub. The goal is to minimize the sum of the overall transportation costs in the network. The traffic in the network consists of

collection (from origin nodes to hubs), transfer (between hubs) and distribution (from hubs to destination nodes). Consider G=(N, A) as a hub-and-spoke network in which each vertex of the set N corresponds to a point of origin and destination of flows and can be chosen for establishment of a hub. The arcs of set A are the elements that constitute the routes. Let $W = |w_{ij}|$ be the flow demand matrix between pairs of nodes ij (with $i, j \in N$).

Vasconcelos et al (2011) defined C^0_{ijkm} as the sum of the transport costs of the direct flows and the flows via hubs of a node i to another node j (according to the following equation):

$$C^0_{ijkm} = P_{ijkm} C_{ijkm} + (1 - P_{ijkm}) C_{ij} \quad (1)$$

The equation (1) indicates that the flows from vertex i to vertex j are distributed between two route types (a direct one and another through hubs). To estimate the transportation cost matrix we used the formulation of Vasconselos et al. (2011) For highway transport they created (by regression analysis) two curves to model the average freight rate per ton-kilometer (US$/t km) as a function of the distance traveled (dist), which are $C^1_{ij} = 1.1364(dist)^{-0.481}$ and $C^2_{ij} = 3.0675(dist)^{-0.562}$. The first function models the freight rate for hauling services to collect or distribute cargoes to or from hubs, while the second function refers to direct highway transport between an O–D pair.

The parameters and variables used in this model are as follows.

Table3. Parameters

n	The number of nodes
H^*	Pre-established hubs
α, $\aleph$, γ	Discount factors due to economies of scale
a^0_k	A constant of the hinterland of the origin hub (k)
$\bar{a}_k$	The vector of parameters of the hinterland of the origin hub (k)
P_{ijkm}	The probability that a user of the pair ij will opt for the ijkm hub route when it is offered the choice only between this route and the direct route
w_{ij}	Demand matrix between pairs of nodes ij
C_{ijkm}	The transport cost of demand w_{ij} over a route ijkm
C_{ij}	The cost of a direct route (ij)
C^o_{ijkm}	The sum of the transport costs of the direct flows and the flows via hubs of a node i to another node j
$\bar{h}_{ijkm}$	The vector of variables related to the route from i to j that goes through hubs k and m
$\bar{d}_{ij}$	The vector of variables related to the direct route

Table4. Decision variables

Y_k	Binary taking value 1 if a hub is established at vertex k; and 0 otherwise.
X_{ij}	Binary taking value 1 if no hub is used for the transportation; and 0 otherwise.
X_{ijkm}	Binary taking value 1 if a flow from i to j is routed via hub nodes (k,m); and 0 otherwise.

The objective function is written as (2):

$$\min_{x,y} F(x,y) = \left[\sum_{i\in N}\sum_{j\in N}\left(C_{ij}x_{ij} + \sum_{k\in(H\cup H^*)}\sum_{m\in(H\cup H^*)} C^o_{ijkm}x_{ijkm}\right)\right] + \left(\sum_{h\in N} C_h y_h\right) \quad (2)$$

Where $H^* \subseteq N$ is the subset consisting of points where hubs have already been installed and $H \subseteq N$ is the subset consisting of feasible points at which new hubs may be installed. The transport cost of demand w_{ij} over a route ijkm is given by

$$C_{ijkm} = w_{ij}(\aleph C_{ik} + \alpha C_{km} + \delta C_{mj}) \quad (3)$$

C_{ik}, C_{km} and C_{mj} are unit transport costs for each route sub segment in which $\aleph$, α and δ are discount factors due to economies of scale. Each of these three factors ranges from 0 to 1, and α is expected to be smaller than the other two since the most significant cost reductions are achieved over the interhub links, where the flows are higher than on the spokes. So $C_{ij}=w_{ij}.\ c_{ij}.\ C_h$ is the cost of establishing a new hub terminal at vertex $h \in N$.

The constraints are:

$$x_{ij} + \sum_{k\in(H\cup H^*)} \sum_{m\in(H\cup H^*)} x_{ijkm} = 1 \quad \forall i,j \in N \quad (4)$$

$$y_h = 1 \quad \forall h \in H^* \quad (5)$$

$$X_{i,j} \in \{0,1\} \quad \forall i,j \in N \quad (6)$$

$$Z_{ijkm} \in \{0,1\} \quad \forall i,j,k,m \in (H \cup H^*) \quad (7)$$

$$x_{ijkm} = 0 \quad \forall i,j,k,m \in (H \cup H^*) \big| P_{ijkm} = 0 \quad (8)$$

$$\sum_{m\in(H\cup H^*)} x_{ijkm} + \sum_{\substack{m\in(H\cup H^*)\\ m\neq k}} x_{ijmk} \leq y_k \quad \forall i,j,k \in (H \cup H^*) \quad (9)$$

$$\sum_{\substack{k\in(H\cup H^*)\\ k\neq i}} \sum_{m\in(H\cup H^*)} x_{ijkm} + y_i \leq 1 \quad \forall i,j \in N, i \neq j \quad (10)$$

$$\sum_{k\in(H\cup H^*)} \sum_{\substack{m\in(H\cup H^*)\\ m\neq j}} x_{ijkm} + y_j \leq 1 \quad \forall i,j \in N, i \neq j \quad (11)$$

$$y_i + \sum_{\substack{(k,m)\in A\\ (k,m)\neq(i,j)}} x_{iikm} \leq 1 \quad \forall i \in N \quad (12)$$

$$x_{kkkm} = x_{kkmk} = 0 \quad \forall k,m \in (H \cup H^*), m \neq k \quad (13)$$

$$x_{kjmk} = x_{jkkm} = 0 \quad \forall k,m,j \in (H \cup H^*), m \neq k, j \neq k \quad (14)$$

$$Y_k \in \{0,1\} \quad \forall k \in N \quad (15)$$

The constraints in (4) guarantee that each flow ij is totally routed. Those of (5) type define the vertices h of the network that are existing hubs, such that their respective C_h values are nil. Constraints in (8) guarantee that x_{ijkm} is null and x_{ij} equals 1 when P_{ijkm} vanishes. The constraints in (9) guarantee that a new route that goes through a hub k, only can be used if this concentrate or terminal is established. The constraints in (10) determine that a flow starting from one hub must go directly to a distribution hub without passing through a collection hub because the origin is already considered to be a collection hub. The constraints in (11) follow the same logic as that in (10) but analogously. The constraint in (12) represents the case where hub i is both the origin and destination of a flow, but this flow may not pass through any other vertex because hub i itself is the collection and distribution hub. The constraints in (13) and (14) follow the same logic as those in (10), (11) and (12), which impede flows over redundant routes.

In this model, the author defined P_{ijkm} as the fraction of users that probably choose route ijkm if this is the best one via hubs. Vasconselos et al. (2011) estimated P_{ijkm} utilizing choice model (as proposed by Domencich and McFadden (1975) and discussed in Ortuzar and Willumsen (1995) :

$$P_{ijkm} = \frac{1}{1+\exp(\bar{a}_k(\bar{d}_{ij} - \bar{h}_{ijkm}) + a_k^0)} \quad (16)$$

where $\bar{a}_k$ is the vector of parameters of the hinterland of the origin hub (k); a_k^0 is a constant of the hinterland of the origin hub (k); $\bar{h}_{ijkm}$ is the vector of variables related to the route from i to j that goes through hubs k and m; and $\bar{d}_{ij}$ is the vector of variables related to the direct route.

To obtain P_{ijkm}, it was necessary to estimate the various characteristics of the transport process which could be considered as variables of vectors $\bar{h}_{ijkm}$ and $\bar{d}_{ij}$ (e.g., cost, time, variance of time, etc.). Thus, for each O–D pair ij, we assumed $\bar{d}_{ij}$ and $\bar{h}_{ijkm}$ as the time of transportation for direct and hub routes, respectively. Basis on the physical laws, time is directly proportional to distance when speed is constant and inversely is proportional to speed when distance is constant. Combining these two rules together gives definition of time in symbolic form:

$$t = \frac{x}{v} \quad (17)$$

In each transportation model, road geometry standards, type of vehicles, speeds restriction, distances between nodes etc were necessary to be studied. Gathering such data was possible for a real world problem. But in this case it needed financial and human resources which had not been considered in the scope of this study. So, for each solution, the distances between nodes were generated randomly and speed parameter was defined based on road type and permissible speed. The speed estimation was covered in more detail in this Section.

As we mentioned before, we applied fuzzy theory as an innovative aspect of our study in order to obtain a more real situation in the network. Considering the difference between driving speeds applied by different drivers in different road types, we define the speed variable as a triangle fuzzy number, consequently the time is calculated as a fuzzy function. Utilizing the fuzzy time variable in P_{ijkm} formulation causes this probability is calculated as a fuzzy function, after defuzzification, it is used in C^{o}_{ijkm} estimation. In this study, establishing a new hub is not feasible for an O–D pair with a distance of less than 400 km nor for an O–D pair where the distance from its origin or from its destination to the respective closer port was greater than 1000 km. It was possible to eliminate the first type of O–D pair because in those cases there are no scale economies and the second types were out of the aim of the study, so, we assume that those cases are considered in the other networks such as airline, waterway etc.

Taking the above explanation into account, the main parameters are defined in this way; $\bar{x}_{ij}$ was defined as the distance between pairs of nodes, ij, $\tilde{v}_{ij}$ and $\tilde{t}_{ij}$ are defined as the speed and time of transportation between pairs of nodes, ij, when a flow starting from one hub and going directly to a distribution hub without passing through a collection hub, and $\tilde{v}_{ijkm}$ and $\tilde{t}_{ijkm}$ are defined as the speed and time of transportation between pairs of nodes, ij, when a flow starting from one node and going to another node with passing through a collection hub, k and m ($k, m \in N$). In order to estimate $\tilde{v}_{ij}$ defined as the triangular fuzzy number; we apply the Pereira and Widmer's classification (2013). They classified the roads of Brazil according to Minimum Radius, maximum grade, width of traffic lanes, and with of shoulders. However, the valid speed in each class is shown in the Table 5.

In the studied problem, three paths were assumed; from one hub to another hub, from a non-hub node to a hub (and vice versa), and from a non-hub node to another none hub node.

Table5. Road Classification

Road Classification	**Valid speed (km/h)**		
	Mountainous	**Rolling hills**	**Level**
Class Zero	80	100	120
Class One	60	80	100
Class Two	50	70	100
Class Three	40	60	80
Class Four	30	50	60

We merged two classes, "one" and "two", because their upper and lower bounds were approximately closed. In the new interval named class "one", the bounds were replaced by the average speeds of the two previous classes.

Considering the upper bound of class "four" which was low, the fifth class road was thought not to be acceptable from the design standards point of view and considering the objective function, cost minimization in the network, it was not applied for transportation. On the basis of these assumptions, Table 5 was changed as bellow.

Table 6. Proposed road classification

Road Classification	**Valid speed (km/h)**		
	Mountainous	**Rolling hills**	**Level**
Class Zero	80	100	120
Class One	55	75	100
Class Two	40	60	80

With regard to Table 6, these assumptions had been made:

The road class for hub-hub pairs in the network was "Zero"

The road class for hub-spoke or spoke-hub pairs in the network was “one”

The road class for spoke-spoke pairs in the network was “two”

A driver, who drives at speed of less than the lower bound or more than the upper bound of the defined ranges for each class of road, would be penalized.

The valid speed was defined as a triangular fuzzy number as it is illustrated in Figure 1:

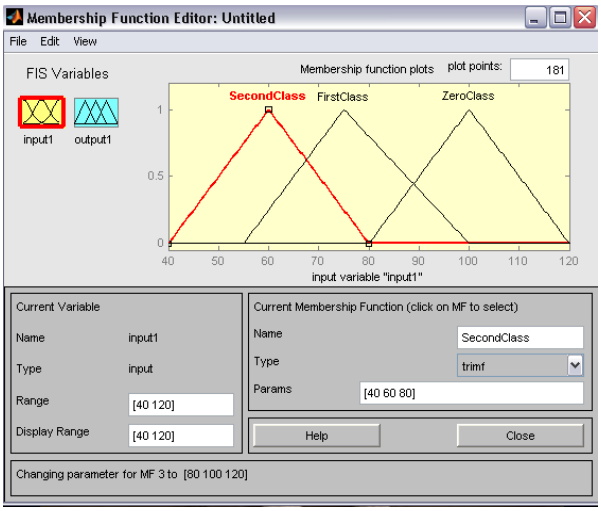

Figure1. Speed definition as a triangular fuzzy number

For each O-D pair ij, transferring time was calculated via the fuzzy function:

$$\tilde{t}_{ij} = \frac{\bar{x}_{ij}}{\tilde{v}_{ij}} \tag{18}$$

It is proved that the inverse and multiplication of a fuzzy number, is a fuzzy number, so the above equation is acceptable (Kwang 2005). Since, the results from addition between fuzzy numbers result also fuzzy numbers (Kwang 2005), the transferring time was estimated for each path via this equation:

$$\tilde{t}_{ijkm} = \tilde{t}_{ik} + \tilde{t}_{km} + \tilde{t}_{mj} \tag{19}$$

Then, equation of P_{ijkm} was modified as is noted bellow:

$$\tilde{P}_{ijkm} = \frac{1}{1+\exp(\bar{a}_k(\tilde{t}_{ij}-\tilde{t}_{ijkm})+a_k^0)} \tag{20}$$

After calculating $\tilde{P}_{ijkm}$, it was defuzzified using COG (center of gravity) method. Some other necessary parameters (that can be obtained) were the followings:

O–D matrix for the general cargo flows between the hinterlands (w_{ij});

Transportation cost matrix for each pairs of nodes;

The cost of the establishing a new hub (C_h);

Discount factors; χ , α , γ.

Parameters of the probability functions; $\bar{a}_k$, a_k^0.

In our research, we did not find a general cargo O–D matrix for Brazil that could be used in this study. On the other hand, the main aim of this work was to extend of the previous model. Thus, an exact estimation of the O-D matrix considering substantial human and financial resources for field surveys was not possible within the scope of this study. Therefore, a random O-D matrix (n×n) between two assumptive numbers was obtained by software in each stage of the solution. In real world problems, it can be generated by methods such as regression analysis using the annual demand from i to j.

By using the generated distance matrix and cost equation as it was explained before, the cost for each route could be calculated and the cost matrix would be made in this way. Since we had not considered any special nodes, we couldn’t estimate the establishment costs. Therefore C_h for each node should be considered as a random number. $\bar{a}_k$ was the

vector of parameters of the hinterland of the origin hub (k); a_k^0 was a constant of the hinterland of the origin hub(k). Since, we didn't considered a special network, it could not be calculated, so, in order to estimate $\bar{a}_k$ we created a random matrix with arrays between -2 to +2 and according to the main model we set $a_k^0 = 0$. χ,α,γ are discount factors estimated in a way to be more efficient for this study which is demonstrated in more details, in section 4.

3. Genetic algorithm

As we discussed in pervious sections, to the best of our knowledge, the UMApHLP in a decentralized management has not been solved for large scale problems. So, another contribution of this study is applying an efficient metaheuristics algorithm to solve this problem for large sized networks. The closest study to our research is the one carried out by Rabbani et al. (2015). They applied two well-known metaheuristic algorithms, Simulated Annealing (SA) and Genetic Algorithm (GA) to solve the incapacitated multiple allocation p-hub center problem for small scale and large scale standard data sets. The numerical results of running the GA and SA for standard test problems show that for smaller scale test problems, SA showed greater performance versus GA but for larger scales of data sets the GA generally yielded more desirable solutions. As we applied LINGO to solve small scale problems, it would be possible to test our results for small sized networks by comparing the computational results by LINGO outputs. But when it comes to large sized networks we preferred to apply more efficient algorithm, so we selected GA.

Genetic algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. It is used to find the near-optimal solutions in large spaces, which is inspired from population genetics (Haupt 2004). The method was developed by John Holland (1975) and popularized by one of his students, David Goldberg (1989). GA have been successfully applied to a variety of problems in operation research including HLP problems; several samples are found in Topcuoglu et al. (2005), Kartica et al. (2005), Eraslan (2010), Yang (2013), Bashiri et al. (2013), Rabbani et al. (2015) etc.

The basic idea is to start from a population of initial solutions and search for successive improvements by examining neighboring solutions. GA chooses a direction to move, then moves in that direction until the cost function begins to decrease. Next the procedure repeats in another direction (Haupt 2004).

3.1. Representation and objective function

Following this line, we propose an efficient GA with two phases. Phase1, called GA1, aims at providing an initiative network in which there are some hubs and spokes while Phase2, called GA2, aims at improving solutions of this network with some pre-established hubs. Note that the same operators are applied for both algorithms, GA1 and GA2.

GA works with a population of individuals, each representing a possible solution to a given problem. In both phases, the binary encoding of the individuals is used in GA implementation. Each solution is represented by the binary string of length n. Digit 1 in the genetic code denotes that particular hub was established while 0 shows it is not. For instance, the genetic code A=[101101] represents a network with 6 nodes in which the first, third, fourth, and sixth ones are hubs. Since users can be assigned only to the opened hub facilities, only array Y_k is obtained from the genetic code. There are no capacities, so the values of x_{ijkm} can be calculated during the evaluation of the objective function. In each solution, after finding the paths between all pairs of nodes, the objective function can be computed by summing up the distances origin-hub, hub-hub and hub-destination, multiplied by flows and corresponding parameters α, $\aleph$, γ parameters and adding fixed cost Y_h of established hubs (Y_k = 1).

3.2. Initialization process

The initial solutions of GA1 are generated a heuristic approach, while those of GA2 are generated randomly.

The heuristic, called "nearest neighbor ordering", applied for GA1 is as follows. The "closer" hubs are favored for each non-hub node. Therefore, while generating the initial population, the first bit and the remaining ones of each genetic code segment are generated with different probabilities. Each individual in the initial population is generated as follows. The first bit in each gene takes the value of 1 with the probability of $\frac{H^*}{n}$ .The second bit in each gene is generated with the probability of $\frac{1}{n}$, while the following bits takes the value 1 with two times smaller probability than the previous ones $\left(\frac{1}{2n}, \frac{1}{4n}, \frac{1}{8n}, \ldots\right)$. The applied strategy ensures that in the initial population each non-hub node is frequently assigned to its closest/closer hub and rarely to a distant hub. For more detailed explanation see (Kratica 2005).

Although the probability of $\frac{H^*}{n}$ for generating the first bits in each gene leads to establishing p hubs, it might be slightly different in practice. If an individual had a number of ones in the genetic code that is different from H^* (denoted as k, $k \neq H^*$), it is corrected by adding/erasing $|H^* - k|$ hubs going backwards from the end of the genetic

code. In this way, the initial population becomes feasible. Genetic operators implemented in the GA1 preserves the fixed number of hubs and keep them distinct. Therefore, all individuals in the following generations remained feasible.

The initial population of the GA2 is randomly generated. The specific characteristic of this procedure is that the generation occurs in a given string already contained some pre-established hubs that will be frozen during the GA2. The number of hubs to be located is in GA2 predetermined (P).

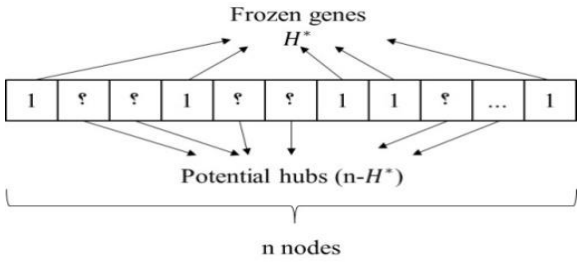

Figure2. The proposed chromosome

3.3. Selection mechanism

The selection operator chooses the individuals that produce offsprings in the next generation, according to their fitness. The term fitness is extensively used to designate the output of the objective function in the GA literature (Haupt 2004). The fitness function represents the objective function which describes the optimization problem. It is taken the same as the objective function, hence, the fitness of each individual can be calculated by first solving the problem and computing the objective value (Doerner et al. 2007); hence, the minimum value of the fitness function was desired. Low fitness-valued individuals have less chance to be selected than high fitness-valued ones.

We use the rank weighting roulette-wheel strategy for our both genetic algorithms. This approach is problem-independent and found the probability from the rank, n, of the chromosome (Haupt 2004). In this method, the ranks of the chromosomes are calculated via their fitness function value which is used instead of the fitness values. The selection probability of the chromosome i is calculated through this equation:

$$p_i = \frac{Rank_i}{\sum_{j=1}^{n} Rank_j} \tag{21}$$

In which, rank j was calculated using the following equation:

$$\sum_{j=1}^{n} Rank_j = \frac{n(n+1)}{2} \tag{22}$$

Before ranking, elitist strategy by replacing the worst solutions with the best solutions had been done.

3.4. Crossover mechanism

After a pair of parents is selected, the crossover operator is applied to them producing two offsprings. The operator we use in both GA1 and GA2 implementation is one-point crossover. It is performed by exchanging segments of two parents' genetic codes after randomly chosen crossover point. Consider a 10-node network, as it is illustrated in Figure4, F genes are the pre-established hubs which had been created in GA1 and are frozen in GA2.

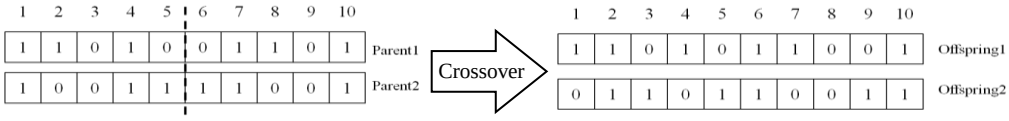

Figure3. The crossover of GA1

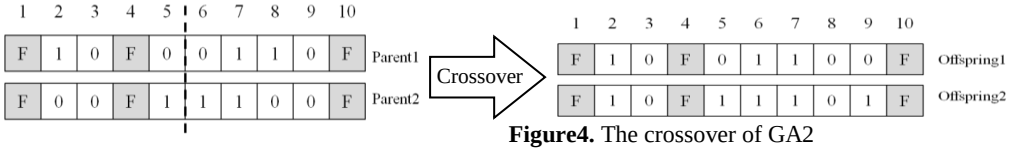

Figure4. The crossover of GA2

3.5. Mutation mechanism

The simple mutation operator used in this GA concept is performed. Note that, the hubs which are resulted from the GA1 are frozen; hence, if the number of frozen genes is H^*,as it is illustrated in Figure 6 in which the frozen nodes are named F, the search space concludes n-H^* genes.

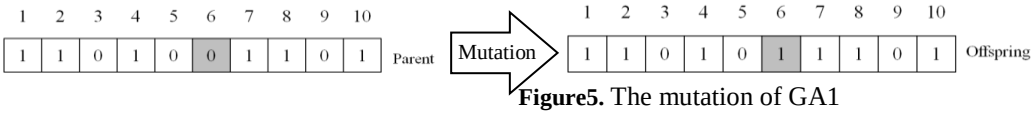

Figure5. The mutation of GA1

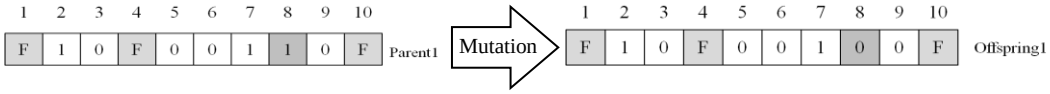

Figure6. The mutation of GA2

3.6. Generation replacement strategy

Considering (Kratica 2005), two thirds of the populations are directly passing in the next generation (elite individuals). Genetic operators are applied on the rest of the population, so that only one third of the population was replaced in every generation. The objective value of each elite individual is calculated only once, and this provides significant time savings.

Regarding (Kratica 2005), duplicated individuals are removed in each generation. This is very effective method for saving the diversity of genetic material and keeping the algorithm away from premature convergence.

4. Computational results

In this section, we present the results of our GA, tested on an Intel ® Core™ i3-4030 processor and 4.00 GB of RAM memory under Microsoft Windows 7 operating system. The code is written in Matlab R2013b v8.2 programming language. We conduct two sets of numerical experiments. Firstly, a simulated problem involves 10, 20, and 30 nodes. Secondly, two new data sets consisting of 50 and 100 nodes are generated to demonstrate that our algorithm can effectively solve large-scaled problems. In order to verify the effectiveness of the proposed GA, we applied two methods, explained in section4.3.

4.1. Parameters tuning: Taguchi method

In order to tune algorithm's parameters, a number of statistical methods can be used in the design of experiments. Taguchi's method is a systematic, simple, and yet efficient approach to optimize design parameters of each algorithm using a limited set of tests. The major tools used in Taguchi method to change and systematically test different levels of parameters include: 1) designing experiments particularly orthogonal arrays (OAs), 2) single-to-noise ratio (S/N), and 3) calculating relative percentage of deviation (RPD). OA matrix is composed of numbers arranged in rows and columns. Each row represents the level of parameters in each execution and each column refers to a specific level of a parameter that can be changed in each execution. S/N ratio is an indicator. Taguchi tests are aimed at finding the best level of each parameter so that the S/N ratio of each parameter at the same level becomes maximized. The relative percentage of deviation indicator seeks to reduce solutions' deviation from the optimal value of the objective function. Taguchi method can be divided into several different steps. These steps are described below:

Step 1: Various parameters and their levels that affect the performance of the given process are first identified. In other words, the parameters that can have positive impact on the performance of the algorithm are selected.

The following parameters are selected in the present study:

1) Crossover probability; 2) mutation probability; 3) initial population size; and 4) maximum iteration.

The considered parameters and their levels are shown in the table8. In other words, the method is used for four parameters at three levels. The numbers mentioned in table 7 are based on the trial and error method.

Table7. The Selected Values for Each Parameter at Each Level

Parameter	Level 0	Level 1	Level 2
% Cross	0.7	0.8	0.9
% Mut	0.1	0.2	0.3
Pop size	10	20	30
Max iter	10	20	30

Given the above table, we use Taguchi design in our tests that consists of nine tests with 4 three-level parameters.

After this step, in order to compute the S/N and RPD ratios, the normalized values of objective functions with respect to every trial are calculated.

Table 8. The OA Matrix used in the Selected Taguchi Design

Sen 1	Sen 2	Sen 3	Sen 4	Sen 5	Sen 6	Sen 7	Sen 8	Sen 9
0	0	0	1	1	1	2	2	2
0	0	2	0	1	2	0	1	2
0	2	1	2	1	0	1	0	2
0	1	2	2	0	1	1	2	0

Step 2: S/N ratio of each test is calculated. In general, there are three types of S/N ratio in Taguchi method. Here, S/N ratio of Eq. (23) is used.

$$S/N = -10 \log\left(\frac{1}{k}\sum_{i=1}^{k}(\text{objectivefunction})_i^2\right) \quad (23)$$

Where K is the number of tests. Eq. (24) is used to calculate the relative percentage deviation.

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \quad (24)$$

Alg_{sol} and Min_{sol} are the algorithm's solution and the best solution, respectively. When all the S/N and RPD ratios are calculated for each test, Taguchi method uses a diagram approach to analyse the data. According to this approach, the average diagrams for S/N and RPD ratios of each parameter at different levels are drawn. The optimal level of each factor is where the S/N diagram and RPD diagram are maximized and minimized, respectively. Thus, the optimal value of parameter is first determined according to diagram S/N. When at least two levels have an equal S/N ratio, we refer to the RPD diagram and compare the specific parameters according to it. Figure 7 shows the results.

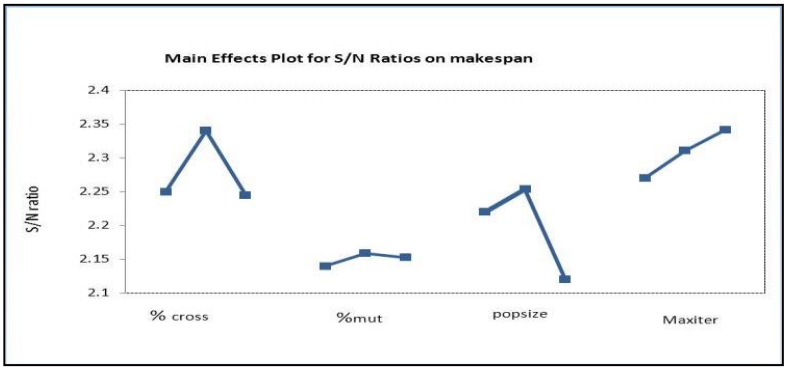

Figure7. S/N Diagram

The selection of optimal levels based on one of the Figures guarantees the optimal values in the other. Given the Figures, the optimal values are shown in table 9.

Table9. Optimal Values of Parameters

Parameter	Optimal Value
%Cross	0.8
%Mut	0.2
Popsize	20
Maxiter	30

4.2. Solution result of the proposed GA algorithm

The proposed GA algorithm is executed 20 times for each problem size with H^* pre-established hubs using different seed values and the solution with the minimum cost is selected. Our experimental study is grouped into two categories including small-scale and large-scale problems; for small ones $n \in \{10, 20, 30\}$ and for large ones $n \in \{50, 100\}$ are assumed. In order to provide the best initial state of the network, for each problem size n and P (assumed pre-established hubs, GA1 was run 30 times and the solution with the lowest cost value obtained becomes the initial network.

We assumed that each of these three factors $\aleph$, α and δ are ranges from 0 to 1, and $\alpha < \aleph, \delta$ (2011). In order to obtain the best amount of these parameters we use sensitive analysis. A small instance is solved with different discount factors and the best ones are selected for the next solutions. In this instance, pop size=20, N=10, Hub=3, N_{gen}=30,

Pc=0.8 and Pm=0.2 are considered. At first, for $\alpha = 0.5$ we solve the instance with $0.5 < \gamma \text{ and } \aleph < 1$, with increasing in $\gamma \text{ and } \aleph$, the cost is worse, so we assume $\gamma = \aleph = 0.5$ and solve the instance with $0 < \alpha < 1$, with the reduction of α, the cost becomes better. Hence, we assumed $\alpha = 0.1$, and solve the problem with $0.1 < \gamma \text{ and } \aleph < 0.5$, in that case, the best cost is obtained with $\gamma \,\&\aleph = 0.4$, basis of this analysis the discount factors were assumed $\alpha = 0.1, \gamma = 0.4, \aleph = 0.4$.

For each instance, O-D matrix, establishment costs, economic coefficients, vector of parameters of the hinterland of the origin hub, distance matrix and consequently the cost matrix had been generated once for GA1 and then they are used in related GA2. The results of small instances are condensed in Table 10 and for large instances they are shown in Table 11.

Table 10 presents the results for GA based on the small problems and table 11 presents the result of executing the large size problems. For each problem instance, n denotes the number of nodes, pre-hub denote the initial state of network, $\mathrm{Opt_{cost}}1$, the optimal values of the objective function (initiate cost of the network before assuming the time as a fuzzy parameter), the numbers in the column labelled "$\mathrm{Opt_{cost}}2$", the final cost of the objective function after applying the proposed model. "Num of new hubs" shows the number of established hubs in the network after applying the fuzzy method via our proposed model, while $\mathrm{NO_p}$, denotes the new-hub numbers. Total running time is shown in the last column. As we can see from the Table 10, the network with n=10 nodes and $\mathrm{H^*} = 1$and $\mathrm{NO_p} = 0$ show the minimum network cost.

Table10. The results of small instances

n	case	Primary network			Final network			
		Pre-Hub	$Opt_{cost}1$		Num of new hubs	NO_p	$Opt_{cost}2$	t
	10H1	3	6578.14		0	0	4223.66	32.4
10	10H2	3,6	7640.16		0	0	6017.29	45.6
	10H3	2,3,6	7737.06		1	5	7507.22	45.0
	10H4	2,3,6,7	7905.52		1	4	7528.98	50.9
	10H5	3,4,5,6,7	8755.83		0	0	7536.68	56.4
	20H1	12	11184.58		0	0	9986.24	130.4
	20H2	2,12	10772.30		1	7	9577.21	165.5
20	20H3	2,7,8	11246.05		2	3,5	11253.12	179.7
	20H4	2,5,7,12	10036.71		1	8	10025.98	190.3
	20H5	2,5,8,12,13	12544.03		0	0	12143.32	200.8
	30H3	3,12,22	13834.01		3	13,18,22	10142.26	299.2
	30H4	12,17,18,22	11222.32		1	3	99787.93	286.1
30	30H5	3,12,17,18,22	11387.80		0	0	11325.22	280.2
	30H6	2,3,17,18,22,25	10279.41		0	0	10022.44	273.3
	30H7	3,15,17,18,20,22,25	16998.92		1	5	162529.12	272.1

By increasing the number of hub, the network cost is increased too. The results for the network with n=20 can be analyzed in the same way. For the networks with n=30 the minimum cost is obtained with $\mathrm{H^*} = 4$ and $\mathrm{NO_p} = 1$. Note that for each problem size, $\mathrm{H^*} \in$ {1,2,3,….,n/2} was assumed. But we show 5 solutions around the optimum one. As an example, for n=20, the problem was solved by 1to10 pre-established hubs.

As it is clear in table 11, the optimal costs for network with n=50, $\mathrm{H^*} = 7$and $\mathrm{NO_p} = 3$ and for network with n=100, $\mathrm{H^*} = 11$and $\mathrm{NO_p} = 2$ are obtained.

4.3. Comparison of Computational results

In order to measure the effectiveness of our method, we solved an instance with n=10, 20 node with deterministic and fuzzy time parameter and then compare the solution quality and running times in the both models.

Table11. The results of large instances

	case	Primary network				Final network	
		Pre-Hub	Opt_{cost}	New hubs	NO_p	Opt_{cost}	t
50	50H5	3,4,7,8,13	81332.8	0	0	74915.30	1208.7
	50H6	3,4,7,8,41,45	76694.3	0	0	72515.23	1001.3
	50H7	4,7,8,9,26,37,45	69931.8	5	2,3,6	61122.65	985.2
	50H8	8,13,25,28,32,37,39,45	65334.1	2	12,17	61741.78	979.4
	50H9	13,18,23,26,28,32,37,45,48	61225.4	2	10,15	86959.11	944.6
	50H10	5,8,13,18,32,35,39,41,45,48	95132.5	4	1,27,38,42	102888.00	900.2
100	100H10	12,18,32,45,52,68,70,75,82,95	157222.8	5	4,5,10,14	198478.88	3241.5
	100H11	12, 15, 17, 19, 26, 32, 45,52, 67, 75,82	175325.7	2	5,8	151784.24	2999.4
	100H12	3,5,8,12,18,32,43,55,62,70,82,95	195342.0	0	0	162214.33	2514.6
	100H13	1,5,8,12,14,18,32,45,52,67,70,82,95	152340.3	3	22,40,58	201782.14	2387.8
	100H14	8,12,15,18,26,32,39,43,52,55,62,67,82,92	200142.3	1	35	199852.92	1978.2

We also compared the solution results with those had obtained by LINGO solver 15. Both GA and LING were run on the same system with the same parameters.

To perform the comparison of experiments, the population size P=20 and the number of generation for GA are set to 30 and $P_c = 0.8$, and $P_m = 0.2$ and the discount factors are assumed as $\alpha = 0.1$, $\gamma = \aleph = 0.4$. Tables 12 gives the results of comparison experiment for two kinds of networks (n=10, 20) with various hubs ($P = 1,2,3,4,5$).

The third, fifth and seventh columns of the table 12 present set of hubs. The fourth and sixth columns of the table5 shows the corresponding costs from the results generated by our GA-based framework assuming the time as a deterministic and as a fuzzy parameter respectively. The eighth columns show the corresponding costs generated by LINGO. The running time of the three methods are given in the CPU (Time) columns which are measured in CPU seconds.

As we can see from table 12, the results demonstrate that the solutions of the fuzzy model because of more accurate seeking the solution space would be better than the model with deterministic parameters.

Table12. Effectiveness of the proposed algorithm

n	case	GA (Deterministic)		GA (Fuzzy)		LINGO (Fuzzy)		CPU(Time)		
		Hubs	Cost	Hubs	Cost	Hubs	Cost	GA_{Dtr}	GA_{Fuzzy}	LINGO
	10H1	3	6578.14	3	4223.66	3	4223.66	35.2	32.4	1300
10	10H2	3,6	7640.16	3,6	6017.29	3,6	6017.29	50.4	45.6	2070
	10H3	2,3,6	7737.06	2,3,5,6	7507.22	2,3,5,6	7507.22	52.9	45.0	2550
	10H4	2,3,6,7	7905.52	2,3,4,6,7	7528.98	2,3,4,6,7	7528.98	57.8	50.9	2490
	10H5	3,4,5,6,7	8755.83	3,4,5,6,7	7536.68	3,4,5,6,7	7536.68	60.2	56.4	2360
	20H1	12	11184.58	12	9986.24	12	9986.24	133.4	130.4	7821
	20H2	2,12	10772.30	2,7,12	9577.21	2,7,12	9577.21	184.6	165.5	10245
20	20H3	2,7,8	11246.05	2,3,5,7,8	11253.12	2,3,5,7,8	11253.12	194.8	179.7	9482
	20H4	2,5,7,12	10036.71	2,5,7,8,12	10025.98	2,5,7,8,12	10025.98	203.9	190.3	11254
	20H5	2,5,8,12,13	12544.03	2,5,8,12,13	12143.32	2,5,8,12,13	12143.32	213.0	200.8	8759

Moreover, comparison between the sixth and eighth columns of Table 7 illustrates that GA concept reaches optimal solution in significantly shorter CPU time compared to LINGO. Specifically, when the problem size is 20, the average

running time of GA method is 173s. LINGO solver achieves the same optimum values by an average running time of 9512 s. In addition, LINGO solver fails to efficiently solve large and complicated instances. It is due to the fact that LINGO solver achieves its optimal solution using the branch and bound procedure by examining nodes, where number of nodes becomes very large when the test problem size increases. Due to the memory requirements, LINGO solver is unavailable for large problem size.

5. Conclusions and future research

Among all of network parameters, travel time is one of the most important factors that cannot be considered deterministic since its values may vary because of traffic conditions, speed, and time of day, climate conditions, and land and road types. Due to the uncertainty of relevant data, the estimated travel time by automobile between two points has fuzzy features due to the measurement imprecision and perception (Yang et al. 2013).

In this paper we extended the model of the incapacitated hub location problem with fixed costs on networks under decentralized management (UHLP-DM) which was created by Vasconcelos et al. (2011). In our proposed extension, the time had been supposed as a fuzzy parameter and then the model was solved by genetic algorithm.

The results revealed that, because of exploring the more pieces of the solution space, using the fuzzy parameters contributed to more accurate system analysis. Using non-deterministic parameters has large practical real-world application in network design. Since, in the model, we sought to solve the large-sized UHLP-DM problem, it was solved using two genetic algorithms; the first one, GA1, was applied to create the initial network in which there were some pre-established hubs and the second one, GA2, was applied to find new hubs in such a network with regarding to the objective function. The proposed model was implemented for small-sized and large-sized problems using Matlab optimization software (R2013b v8.2) and it was run on the Windows 7 operating system with an Intel ® Core™ i3-4030 processor and 4.00 GB of RAM. The described GA method used a binary encoding of the individuals and an appropriate objective function. However, the proposed GA algorithm surpasses the LINGO solver with respect to efficiency. It obtains optimal values in significantly less time than the LINGO solver for the same test problems. The proposed GA was also able to solve practical size problems that were out of reach for exact methods.

The presented solution is applicable to analyse the transport networks in free markets. Additionally, there are many other points for future research. Some of the points are:

- Consideration of demand in P_{ijkm} calculation.
- Consideration of the transferring cost in network cost calculation.
- Consideration the demand as a fuzzy parameter
- Creation of a new constraint to cover the demand in any hinterland.

References

Alumur S. A., Kara B. Y., and Karasan O. E. (2012). Multimodal hub location and hub network design, *Omega*,Vol. 40(6), pp. 927–939.

Bashiri M., Mirzaei M. and Randall M. (2013). Modeling fuzzy capacitated p-hub center problem and a genetic algorithm solution, *Applied Mathematical Modelling*,Vol. 37, pp. 3513–3525.

Campbell JF. and O'Kelly ME. (2012). Twenty-five years of hub location research, *Transp Sci*, Vol.46, pp. 153–169.

Campbell J.F., Miranda J.R., Camargo R.S. and O'Kelly M.E. (2015). Hub Location and Network Design with Fixed and Variable Costs. *Annual Hawaii International Conference on System Sciences, Computer Society Press*,No. *48*, pp. 1059-1067.

Contreras I., Cordeau J.F. and Laporte G. (2011). Benders Decomposition for Large-Scale Uncapacitated Hub Location, *operation research*, Vol.6, pp. 1477-1490.

Doerner K.F., Gendreau M., Greistorfer P., Gutjahr W., Hartl RF., and Reimann M. (2007). *Metaheuristics-progress in complex systems optimization*, Berlin: Springer.

Domencich T. and McFadden D.L. (1975). *Urban travel demand: a behavioral analysis*, Amsterdam: North-Holland Publishing Co.

Eraslan S. (2010). A genetic algorithm for the p-hub center problem with stochastic service level constraints, Dissertation, *Middle East technical university.*

Ghaderi A. and Jabalameli M.S. (2013). Modeling the budget-constrained dynamic uncapacitated facility location–network design problem and solving it via two efficient heuristics: a case study of health care, *Math Comput Model,* Vol.57, pp. 382–400.

Goldberg D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning,* Reading, MA: Addison-Wesley.

Haupt R.L. and Haupt S.E. (2004). *Practical Genetic Algorithms,* New York: Wiley.

Hernández P., Alonso-Ayuso A., Bravo F., Escudero LF., Guignard M., Marianov V., and Weintraub A. (2012). A branch-and-cluster coordination scheme for selecting prison facility sites under uncertainty, *Comput Oper Res,* Vol.39, pp. 2232–2241.

Holland J.H. (1975). *Adaptation in natural and artificial systems,* Michigan: University of Michigan Press.

Hult E., Jiang H. and Ralph D. (2014). Exact computational approaches to a stochastic uncapacitated single allocation p-hub center problem, *Computational Optimization and Applications,* Vol.59, pp. 85-200.

Kaur A. and Kumar A. (2011). A new method for solving fuzzy transportation problems using ranking function, *Applied Mathematical Modeling,* Vol.35, pp. 5652–5661.

Kratica J., Stanimirović Zorica., Tosic D. and Filipović. (2005). Genetic algorithm for solving uncapacitated multiple allocation hub location problems, *Computing and Informatics,* Vol.24, pp. 415–426.

Kwang D.r. and Lee H. (2005). *First Course on Fuzzy Theory and Applications,* Berlin: Springer.

Nematian J. and Musavi M. (2016). Uncapacitated phub center problem under uncertainty, *Journal of Industrial and Systems engineering (Iranian Institute of Industrial Engineering)* ,Vol. 9, pp. 23-39.

O'Kelly M.E., Campbell J.F., Camargo R.S. and Miranda J.R. (2014). Multiple Allocation Hub Location Model with Fixed Arc Costs, *Geographical Analysis,* Vol.47, pp. 73–96.

Ortuzar J.D. and Willumsen L.G. (1995). *Modelling transport,* New York, Wiley.

Qin Z. and Gao Y. (2014). Uncapacitated p-hub location problem with fixed costs and uncertain flows, *Journal of Intelligent Manufacturing,* pp. 1-12.

Rabbani M. and Kazemi M. (2015). Solving uncapacitated multiple allocation p-hub center problem by Dijkstra's algorithm-based genetic algorithm and simulated annealing, *International Journal of Industrial Engineering Computations,* Vol.6, pp. 405–418.

Rahmaniani R., Saidi M. and Ashouri H. (2013a). Robust Capacitated Facility Location Problem: Optimization Model and Solution Algorithms, *J Uncertain Syst* Vol.7(1), pp. 22–35.

Rahmaniani R., Ghaderi A., Mahmoudi N. and Barzinepour S. (2013b). Stochastic p–robust uncapacitated multiple allocation p–hub location problems, *Int J Ind Syst Eng* Vol.14, pp. 296–314.

Topcuoglu H., Corut F., Ermis M. and Yilmaz G. (2005). Solving the uncapacitated hub location using genetic algorithms, *Computers and Operational Research,* Vol.32, pp. 467–984.

Vasconcelos A.D., Nassi C.D. and Lopes L.S. (2011). The uncapacitated hub location problem in networks under decentralized management, *Computers & Operations Research,* Vol.38, pp. 1656–1666.

Waldemiro P.N. and Widmer J. (2013). *Compatibility of Long and Heavy Cargo Vehicles With the Geometric Design Standards of Brazilian Rural Roads and Highways,* New York: Wiley.

Yang K., Liu Y.K. and Zhang X. (2011). Stochastic p-hub center problem with discrete time distributions, Lecture Notes, *Computer Science,* Vol.6676, pp. 182–191.

Yang K., Liu Y.K. and Yang G.Q. (2013). Solving fuzzy p-hub center problem by genetic algorithm incorporating local search, *Applied Soft Computing*, Vol.13(5), pp. 2624–2632.

Zarrinpoor N. and Seifbarghy M. (2011). A competitive location model to obtain a specific market share while ranking facilities by shorter travel time, *Int J Adv Manuf Technol* Vol.55, pp. 807–816.

Zanjirani Farahani R. , Hekmatfar M., Boloori Arabani A. and Nikbakhsh E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications, *Computers & Industrial Engineering*, Vol.64, pp. 1096–1109.