Expert Systems With Applications 119 (2019) 1–19

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Particle Swarm Optimization for Uncapacitated Multiple Allocation Hub Location Problem under Congestion

Çağrı Özgün-Kibiroğlu[a,*], Mehmet Nahit Serarslan[b], Yusuf İlker Topcu[b]

[a] Faculty of Engineering, Department of Industrial Engineering, Haliç University, Beyoğlu, İstanbul, 34445, Turkey
[b] Faculty of Management, Department of Industrial Engineering, İstanbul Technical University, Maçka, İstanbul, 34367, Turkey

A R T I C L E I N F O

Article history:
Received 11 June 2018
Revised 1 September 2018
Accepted 10 October 2018
Available online 11 October 2018

Keywords:
Network design
Hub location problem
Congestion
Particle swarm optimization

A B S T R A C T

In this paper we address the hub location problem in which capacity restrictions are introduced into the objective function as a penalty cost to represent their congestion effects on respective hubs. The goal of hub location problems is to locate an optimal set of hubs while minimizing sum of the transportation and opening costs under various constraints such as number of hubs, delay time and capacities of hubs. Moreover, we propose a more realistic variant of the problem where the capacities of hubs are not only a limitation in selecting the hubs. In practice, capacities of hubs are predicted by a strategic plan, but these predictions generally fail to meet the expectations due to changes in the flow in future and competitive nature of air transportation. A particle swarm optimization (PSO) is proposed to handle the complex nature of this problem and shorten the computing times. The performance of the proposed PSO is analyzed comparatively on the Australia Post (AP) data sets. The numerical results show that the proposed method can find the optimal solutions in less computation time in comparison with the earlier multiple allocation hub location models.

1. Introduction

In air transportation networks, the hub location problem became an important subject of research since the last quarter of 20th century. In general, hub location problems aim to design networks where the flows of origin-destination pairs are accumulated and distributed through some central nodes named as hubs. The benefits of the economies of scale by flow consolidation between hubs fulfill the expectations of the sector and competitive, progressive motivations of the actors in the network. In other words, in case of non-hub network design, direct connections between nodes cost more and the network becomes more complicated to design.

In air transportation, as well as in other transportation networks, it is important to extend the network as large as possible to satisfy the customer needs and to obtain competitive advantage. The hub location problem (HLP) also takes the advantage of a discount factor while minimizing the transportation and opening costs. However, the more nodes are inserted to the network, the more capacity restrictions are imposed to balance the flows between nodes. Essentially, in recent times, the air transportation companies seek to resolve these undesired consequences and optimize traffic flow density of their hub airports. Hubs' capacities tend to be sensitive against the increasing volume of traffic flows in time. Hence, capacity features of the hub location problem had to be included in the existing literature in order to prevent the violation related to the restrictions of the problem. Some of the important abbrevations used in this paper relating to hub location problem are listed as footnotes[1234567].

In the literature, hub location problems are classified as capacitated and uncapacitated types. In the capacitated class of the hub location problems, capacity is considered as a limitation on hubs. However, restricting possible hub nodes with capacity levels could not represent a realistic approach to the problem. Luckily, in recent years, a new term 'congestion' has been introduced as an additional flow in excess of the capacity which is imposed on hubs. This surplus of the capacity could be represented as a cost or a delay time to minimize in appropriately defined objective functions.

* Corresponding author.
E-mail addresses: cagrikibiroglu@halic.edu.tr (Ç. Özgün-Kibiroğlu), serarslann@itu.edu.tr (M.N. Serarslan), topcuil@itu.edu.tr (Y.İ. Topcu).

[1] Australia Post: AP
[2] Civil Aeronautics Board: CAB
[3] Freight Analysis Framework: FAF
[4] Hub Location Problem: HLP
[5] Particle Swarm Optimization: PSO
[6] Uncapacitated Multiple Allocation Hub Location Problem: UMAHLP
[7] Uncapacitated Multiple Allocation *p*-median Hub Location Problem: UMA*p*HLP

https://doi.org/10.1016/j.eswa.2018.10.019
0957-4174/

While the problem focuses on obtaining a minimum set of opening and transportation costs, a relatively small number of hubs are selected. These hubs are likely to be overloaded in comparison to non-hub nodes. In the diverse range of literature congestion effect on the networks is represented implicitly by constraints which are included in the model, similar to capacity constraints. Furthermore, the congestion is represented explicitly in the literature as a delay cost or delay time in the objective function. For recent and up-to-date reviews and classifications of the problem, see Alumur and Kara (2008); Campbell (1994); O'Kelly and Miller (1994); Campbell (1996); Campbell and O'Kelly (2012); Campbell, Ernst, and Krishnamoorthy (2002) and Contreras, Laporte, Nickel, and Gama (2015).

Congestion has been mentioned for the first time in the paper of Grove and O'Kelly (1986). They have demonstrated the effect of the accumulated flows on a single hub triggered by scheduled delays on a network. Yet the first consideration of the congestion within a hub-spoke network model was that of Marianov and Serra (2003). They have designed a hub network as an M/D/c queuing network. They proposed a probabilistic model as referring to capacity constraints due to the waiting customers in the network and subsequently linearized the model. A tabu search algorithm has been used to handle the computational complexity of the model. Another solution procedure, simulated annealing, has been used to solve a similar problem by Rodriguez, Alvarez and Barcos (2007). On the other hand, some studies of the capacitated hub location problem have considered congestion as a constraint. Costa, Captivo and Climaco (2008) have suggested a multi-criteria model which aimed to minimize both transportation cost and maximum waiting time at hubs. Thereby, congestion has been utilized as a parameter which determines waiting times at hubs according to its level related to capacity. Rodriguez-Martin and Salazar-Gonzalez (2008) have proposed a Benders decomposition algorithm to tackle a capacitated hub location problem based on a multi-commodity formulation where the arcs connecting the hubs are not assumed to create a complete graph. Their model has been similar to the one presented by Sohn and Park (2000) for the fixed-charge capacitated network design problem.

Camargo, Miranda, and Ferreira (2009) have stated that modeling congestion through capacity constraints on the flows does not mimic the exponential nature of congestion effect. Congestion introduced in the model as a capacity constraint causes the model to disregard the realistic structure of the network. Limiting flows according to capacities avoids achieving credible and sustainable allocation of hubs and network designs. Some studies in the literature, such as hub arc location problem have achieved the purpose of avoiding congestion and excessive delays at hubs (Elhedhli & Hu, 2005). Hub arc location problems try to assign hubs while balancing their flows. Thereby, they simplified the congestion problem.

The paper of Elhedhli and Hu (2005) has been the first work where congestion is introduced in the objective function. They have modeled a single assignment hub location problem with a convex cost function, previously discussed in de Palma and Marshall (1998) and Gillen and Levinson (1999), in which the delay cost is assumed to be a power-law function of capacity usage. They linearized the non-linear model by using a piecewise linear function and then solved it by Lagrangean relaxation. Consecutively, the multiple allocation version of this problem was studied by Elhedhli and Hu (2005). They have introduced a Benders Decomposition algorithm to deal with the non-linear part of the problem.

Both studies have proposed congestion as a function of total consolidated flows at selected hubs. Afterward, Elhedhli and Wu (2010) have modified the congestion as the ratio of total flow to the surplus capacity by modeling the network as M/M/1 queues. They proposed a Lagrangean heuristic algorithm where subproblems are solved by the cutting plane method and subgradient optimization. Camargo, Miranda, and Ferreira (2011) considered a different cost function called Kleinrock average delay function and compared this function with power-law function in the modeling single allocation hub location problem. Camargo and Miranda (2012) presented newer congestion perspectives according to user or owner networks and evaluated their results and network designs. Due to the fact that their problem possesses a highly combinatorial nature with the non-linearity associated to congestion, they have worked on a generalized Benders decomposition algorithm. Also, their next study focused on solving large scale instances by that algorithm. Ishfaq and Sox (2012) followed up on this idea and solved the problem with a row generation procedure.

Rahimi, Tavakkoli-Moghaddam, Mohammadi, and Sadeghi (2016) have presented a new bi-objective model for a multi-modal hub location problem under uncertainty considering congestion in the hubs. Alumur, Nickel, Rohrbeck, and Saldanha-da gama (2018) have proposed a model which is determining location and capacities of hubs, allocation of non-hub nodes and routes of flow through the network while providing the delivery between all of the pairs within the given service time. Limiting the service time on the network helps to balance the flows and optimize the congestion effects.

In this paper, we propose a model allocating hubs whose capacities are larger than a desired level and still congestion on hubs is considered as a penalty cost in the objective function. Therefore, hubs are chosen between nodes which have higher capacities in order to reduce the penalty costs arising from surplus flows on hubs. Furthermore, this paper explores the congestion effects written as a convex cost function similar to the one defined by Elhedhli and Hu (2005) and Kleinrock function defined by Camargo et al. (2011). We also propose a new congestion function whose delay cost generated by transportation costs.

Since the hub location problems are NP-hard, developing solution procedures for these problems is a major concern. Determining both location of hubs and allocation of non-hub nodes make the solution techniques complicated for HLPs. Since the middle of 1990s, hub location problem literature has shown significant advance in developing algorithms to obtain high quality solutions in a reasonable time. Specific real world requirements make the problem more complicated to model and solve. Thus, a novel approach is needed to develop better performing solution procedures. Exact, heuristic, meta-heuristic and combinatorial algorithms are employed to solve HLPs. Besides, exact solution techniques are common in the literature; many heuristic algorithms attempt to either solve them more efficiently or to obtain the best solution, not both at the same time. Genetic algorithms, tabu search, simulated annealing, greedy randomized adaptive search procedure and variable neighborhood search are some of the heuristic methodologies used frequently to solve these problems. However, congestion is one of the recent research subjects to develop more realistic and insightful hub location models. There is a very small number of studies focusing on hub location problem with congestion in the literature where only exact solution procedures are proposed. As far as known, this paper is the first study developing a heuristic solution procedure on multiple allocation HLP with congestion in the literature. Table 1 summarizes the solution techniques used to solve the hub location problems with congestion. The data sets in Table 1 and other tables of this work, Australia Post (AP), Civil Aeronautics Board (CAB) can be obtained from OR-Library (Beasley, 1990a). Freight Analysis Framework (FAF) database is maintained by the Federal Highway Administration of the U.S. Department of Transportation (Administration, 2002).

Considering that even simple hub location models have proven difficult to solve, Bryan (1998) pointed out that the research in this area will soon have focused on development of new heuristic algorithms. Currently, developing heuristic algorithms are competing

Table 1
Congestion Literature on the Hub Location Problem.

Authors	Data	Range of Instance	Solution Algorithms
Elhedhli and Hu (2005)	CAB	10-25	Lagrangean Relaxation
Camargo et al. (2009)	CAB-AP	10-50	Generalized Benders Decompositions
Elhedhli and Wu (2010)	CAB	10-25	Lagrangean Relaxation
Camargo et al. (2011)	AP	10-200	Outer-Approximation/ Benders Decomposition algorithm
Camargo and Miranda (2012)	AP	10-200	Outer-Approximation/ Benders Decomposition algorithm
Ishfaq and Sox (2012)	FAF database	25	Row Generation
Camargo and Miranda (2012)	AP	10-100	Generalized Benders Decompositions
Alumur et al. (2018)	AP	10-40	IBM ILOG CPLEX Optimization Studio

for large scale instances, short computational time and minimum gap to optimum. Accordingly, an additional purpose of this paper is to develop an efficient particle swarm optimization algorithm to solve our congested uncapacitated hub location problem.

The remainder of the paper is organized as follows: In the next section, we summarize three mathematical formulations:

- one based on 3-index variables to model uncapacitated problem suggested by Ernst and Krishnamoorthy (1998),
- second one again with 3-index variables suggested by Ebery, Krishnamoorthy, Ernst, and Boland (2000) to model capacitated hub location problem
- third one suggested by Elhedhli and Hu (2005) which is based on 4-index variables.

Then, we propose a revised mathematical formulation with different congestion cost function in Section 3. Section 4 presents a general framework of particle swarm optimization. In Section 5, we develop particle swarm optimization algorithm for the uncapacitated multiple allocation hub location problem (UMAHLP). The computational results in Section 6 show the performance of the heuristic algorithm. Final section is devoted to conclusions.

2. Model Formulation for Uncapacitated Multiple Allocation Hub Location Problem (UMAHLP)

In the multiple allocation hub location problem, each node can be assigned to more than one hub. Given a complete graph $G = (N, P)$, where $N = \{1, 2, \ldots, n\}$ is the set of nodes and $P = \{1, 2, \ldots, p\}$ is the set of potential hubs, $P \subseteq N$. There is a flow traffic, W_{ij} between each pair of origin-destination nodes i and j ($i, j \in N$, $i \neq j$).

Also, a transportation cost, C_{ij} , is assigned to unit flow from i to j. In general, transportation cost is calculated proportional to distance between origin-destination pairs and therefore it is euclidian. Thus, by referring a hub-spoke network consisting of interacting points and three types of links for collection, transfer, and distribution, the following assumptions are required for UMAHLP:

- Hubs are fully connected, and all flow between pairs of spokes go through hubs, i.e. direct connection between spokes is not allowed.
- Hubs have finite capacities, they can have higher capacities from desired level.
- Each spoke can be assigned to more than one hub.

We next present the first formulation for the uncapacitated multiple allocation p-median hub location problem (UMApHLP) published by Ernst and Krishnamoorthy (1998). The parameters of their model are given as follows, with minor notational changes:

- F_k: the cost of establishing a hub at node k
- α, γ, δ: the collection, transfer and distribution coefficients respectively
- O_k: the capacity of hub at node k

The variables in the formulations are defined as:

- Z_{ik}: the flow from node i to hub k
- Y_{kl}^{i}:the flow from node i via hubs k and l ($k, l \in N$, $k \leq l$)
- X_{lj}^{i}: the flow from node i to node j via hub l
- $H_k = \begin{cases} 1 & \text{if node } k \text{ is a hub} \\ 0 & \text{otherwise} \end{cases}$

With these definitions, Ernst and Krishnamoorthy (1998) stated UMApHLP as follows:

$$Min \quad \sum_i \left(\gamma \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^{i} + \delta \sum_k \sum_l C_{lj} X_{lj}^{i} \right) \tag{1}$$

$s.t.$

$$\sum_k H_k = p, \tag{2}$$

$$\sum_k Z_{ik} = \sum_j W_{ij}, \qquad \forall i \tag{3}$$

$$\sum_l X_{lj}^{i} = W_{ij}, \qquad \forall i, j \tag{4}$$

$$\sum_l Y_{kl}^{i} + \sum_j X_{kj}^{i} - \sum_l Y_{lk}^{i} - Z_{ik} = 0, \qquad \forall i, k \tag{5}$$

$$Z_{ik} \leq \sum_j W_{ij} H_k, \qquad \forall i, k \tag{6}$$

$$X_{lj}^{i} \leq W_{ij} H_l, \qquad \forall i, j, l \tag{7}$$

$$X_{lj}^{i}, Y_{kl}^{i}, Z_{ik} \geq 0, H_k \in \{0, 1\}, \qquad \forall i, j, k, l \tag{8}$$

As the second formulation, we summarize the formulation suggested by Ebery et al. (2000). They proposed a formulation of capacitated multiple allocation version which does not only decides on the optimal location of the hubs and the allocation of non-hub nodes to hubs, but also the number of hubs that are required. Thus, there is no need to predetermine the number of hubs. Capacity constraint is added to limit the flows at selected hub.

$$\text{Min} \quad \sum_i (\gamma \sum_k C_{ik} Z_{ik} + \alpha \sum_k \sum_l C_{kl} Y_{kl}^{i} + \delta \sum_k \sum_l C_{lj} X_{lj}^{i}) + \sum_k F_k H_k \tag{9}$$

s.t.

$$\sum_k Z_{ik} = \sum_j W_{ij}, \quad \forall i \tag{10}$$

$$\sum_l X^i_{lj} = W_{ij}, \quad \forall i, j \tag{11}$$

$$\sum_i Z_{ik} \le O_k H_k, \quad \forall k \tag{12}$$

$$\sum_l Y^i_{kl} + \sum_j X^i_{kj} - \sum_l Y^i_{lk} - Z_{ik} = 0, \quad \forall i, k \tag{13}$$

$$Z_{ik} \le \sum_j W_{ij} H_k, \quad \forall i, k \tag{14}$$

$$X^i_{lj} \le W_{ij} H_l, \quad \forall i, j, l \tag{15}$$

$$X^i_{lj}, Y^i_{kl}, Z_{ik} \ge 0, H_k \in \{0, 1\}, \quad \forall i, j, k, l \tag{16}$$

In order to represent the congestion effect on multiple allocation hub location problems, a formulation with usage of four-index variables is suggested by Elhedhli and Hu (2005). Although 3-index variable usage has substantially fewer variables than this formulation, the model is reputed to have a tight linear programming bound. So, it probably becomes a stronger formulation than the others in solving large scale HLPs. This formulation uses four index variables and parameters which are defined below:

- X_{ijkm}: the flow departing from node i and ending at node j that is routed through hubs k and m in that order $(i, j, k, m \in N)$,
- C_{ijkm}: the accumulated transportation cost where the cost between hubs are discounted,
- $\Gamma_k (= e g_k^b)$: the power-law congestion cost function where e and b are positive constants.
- g_k: the total flow through hub k.

The congestion cost function in the formulation is calculated by the total flow through each hub. This function is responsible for distributing the flow load along each installed hub, thus reducing the overall congestion expressed in monetary values (Camargo et al., 2009). Then, the third formulation due to Elhedhli and Hu (2005) is stated with minor notation changes, as:

$$Min \sum_i \sum_j \sum_k \sum_m C_{ijkm} X_{ijkm} + \sum_k (F_k H_k + \Gamma_k H_k) \tag{17}$$

s.t.

$$\sum_i \sum_j \sum_m X_{ijkm} + \sum_i \sum_j \sum_{m \ne k} X_{ijkm} = g_k, \quad \forall k \tag{18}$$

$$\sum_k \sum_m X_{ijkm} = W_{ij}, \quad \forall i, j \tag{19}$$

$$\sum_m X_{ijkm} + \sum_{m \ne k} X_{ijkm} \le W_{ij} H_k, \quad \forall i, j, k \tag{20}$$

$$X_{ijij} \ge W_{ij}\left(H_i + H_j - 1\right), \quad \forall i, j \tag{21}$$

$$\sum_{k \ne j} X_{ijik} \ge W_{ij}\left(H_i - H_j\right), \quad \forall i, j \tag{22}$$

$$\sum_{k \ne i} X_{ijkj} \ge W_{ij}\left(H_j - H_i\right), \quad \forall i, j \tag{23}$$

$$X_{ijkm}, g_k \ge 0, H_k \in \{0, 1\}, \quad \forall i, j, k, m \tag{24}$$

$$C_{ijkm} = \gamma C_{ik} + \alpha C_{km} + \delta C_{mj} \tag{25}$$

$$\Gamma_k = e g_k^b \tag{26}$$

In the paper of Camargo et al. (2009), the congestion function has been calculated proportional to the flow through the hub which is considered increasing properly convex and smooth on [0, ∞). There are other different congestion functions in the literature, as mentioned previous section.

3. Revised Model Formulation with Congestion Cost

Several mixed integer linear programming models have been proposed for uncapacitated multiple allocation hub location problem (UMAHLP). Our model is an extension of the models which are summarized in previous section. This extension essentially includes additional constraints and a reconstruction of the objective function. In our model, the objective function is revised in such a way that a congestion function is added as a penalty cost. This congestion function is determined by the excess of accumulated transportation flows at hubs from capacities of hubs. This surplus of the flow at a given hub is multiplied by the average transportation cost of corresponding hub and a scalar e. The capacity constraint of the model by Ebery et al. (2000) is removed in our model. As for notations used in our model, the following parameters are defined:

- S_k represents the capacity of hub k.
- Γ_k represents the appropriate congestion cost function for surplus of the capacity.
- τ represents desired capacity level in terms of hub capacity (70%, 85%, 100%).
- O_k represents the quantity calculated by multiplying the desired capacity level and capacities of hubs.

Using these four parameters above and in the existing literature, the model for uncapacitated multiple allocation hub location problem (UMAHLP) with congestion can be stated as follows:

$$Min \sum_i \sum_j \sum_k \sum_m C_{ijkm} X_{ijkm} + \sum_k (F_k H_k + \Gamma_k H_k) \tag{27}$$

s.t.

$$\sum_i \sum_j \sum_m X_{ijkm} + \sum_i \sum_j \sum_{m \ne k} X_{ijkm} = g_k, \quad \forall k \tag{28}$$

$$\sum_k \sum_m X_{ijkm} = W_{ij}, \quad \forall i, j \tag{29}$$

$$\sum_m X_{ijkm} + \sum_{m \ne k} X_{ijkm} \le W_{ij}.H_k, \quad \forall i, j, k \tag{30}$$

$$X_{ijij} \ge W_{ij}\left(H_i + H_j - 1\right), \quad \forall i, j \tag{31}$$

$$\sum_{k \ne j} X_{ijik} \ge W_{ij}\left(H_i - H_j\right), \quad \forall i, j \tag{32}$$

$$\sum_{k \ne i} X_{ijkj} \ge W_{ij}\left(H_j - H_i\right), \quad \forall i, j \tag{33}$$

$$X_{ijkm}, g_k \ge 0, H_k \in \{0, 1\}, \quad \forall i, j, k, m \tag{34}$$

$$C_{ijkm} = \gamma C_{ik} + \alpha C_{km} + \delta C_{mj} \tag{35}$$

$$\Gamma_k = \left(\frac{e \sum_i C_{ij}}{n}\right)\left(\frac{g_k - O_k}{g_k}\right) = E\left(\frac{g_k - O_k}{g_k}\right), \tag{36}$$

$$O_k = \tau S_k \tag{37}$$

The objective function is formulated to minimize the total cost comprising the transportation, opening and congestion costs. Constraint 28 determines the total amount of flow through hub k. Constraint 29 assures that the flow for every (i, j) pair is routed via some hub. Constraint 30 guarantees that the flow through a hub only happens if that node is selected as a hub.Constraint 31 assures that the flows between hubs only goes through the corresponding hubs. Constraints 32-33 limit the number of hubs utilized to 2 when the origin or destination nodes are also hubs. Constraint 34 represents the non-negativity constraints of the variables X_{ijkm} and g_k, and the binary variables H_k to 0 or 1. Constraints 35-37 represent the equations of the parameters utilized in the model such as total transportation cost, congestion function and predetermined service level of hubs.

4. Particle Swarm Optimization (PSO)

Particle swarm optimization is a population based meta-heuristic algorithm that simulates the movements of bird flocks and fish schools in the nature. Particle swarm optimization which has been first proposed by Kennedy and Eberhart (1995) is one of the heuristic algorithms recently utilized in most of the optimization problems. Then, Shi and Eberhart (1999) have developed this evolutionary technique for engineering and computer science. Since this technique performs better with low computational cost and easy implementation, researchers focus on more this technique than existing evolutionary techniques such as genetic algorithm. The algorithm is developed based on three simple rules:

- When one bird locates a target or food, it instantaneously transmits the information to all other birds.
- All other birds gravitate to the target or food , but not directly.
- There is a component of each bird's own independent instinctual behaviour as well as its past memory (Adeh, Baroughi, & Alizadeh, 2017).

PSO is a bio-inspired algorithm that mimics the intelligent swarming behavior (information exchange and sharing) that flock of birds deploy in order to swarm in the direction of a good outcome or target (Sen & Krishnamoorthy, 2018). In the context of an optimization problem, a PSO algorithm starts with different points in the solution space to attain global optimal solution and shares the acquired information to determine good search spaces. Each particle is an initial feasible solution. Many such particles are randomly generated, and these moves around in the solution space under some well-defined guidelines (Sen & Krishnamoorthy, 2018). In context of our hub location problem, each particle is a feasible solution which is represented by two different arrays: selected hubs and allocation array. Installing hubs are randomly selected and all nodes are randomly allocated to hubs. Thus, the generation of a particle is realized and the particle has the information of selected hubs, allocation decisions of non-hub nodes and the objective function value of that network design.

PSO is initialized with random solutions assigned to the population, similarly to genetic algorithm (GA) (Jia, Zheng, Qu, & Khan, 2011). It is unlike a GA, however, in that each potential solution is also assigned a randomized velocity, and the potential solutions, called particles, are then 'flown' through the problem space (Eberhart & Shi, 2001). Each particle improves its earlier best solution in the problem space which is achieved so far through previous experiences to stochastically converge towards the best solution in the swarm. This value is called *lbest*. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location decisions obtained so far by any particle in the swarm (Eberhart & Shi, 2001). This is called *gbest*.

The particle swarm optimization concept consists of, at each time step (iteration), changing the velocity of each particle toward its *lbest* and *gbest* locations (Eberhart & Shi, 2001). Every particle improves its own earlier position through previous experience to stochastically converge towards the best position of the swarm. Whereas the particle holding the best position in the swarm is entitled as global best; the best position of the particle until the current state is defined as local best. These values are kept in the memory of algorithm. Consequently, PSO algorithm is based on a convergence where the positions of the particles in swarm are getting closer to the best position of the swarm

For an optimization problem, the decision variable $X= (x_1, x_2, \ldots, x_n)$ is n-dimensional variable, and the variables are called the particles. Assuming the solution space is n-dimensional solution space, each particle has two state descriptions, position $X_i= (x_{i1}, x_{i2}, \ldots, x_{in})$, and velocity $V_i= (v_{i1}, v_{i2}, \ldots, v_{in})$. The position X_i represents a candidate solution (feasible solution) of the problem. Particles change their positions or states with time (iterations) and navigate in a multidimensional solution space. Each particle moves toward the optimum solution based on its present velocity, its previous experience and the experience of its neighbors (Wu & Wang, 2014).

The direction and the magnitude of the movement of a particle is represented as a velocity vector which is utilized to update the particles on each iteration to reflect the desired search direction. The particle update rule consists of two equations, one for an update in the velocity vector and the other for a change in the position of the particle due to the movement. The velocity and position update rules are given in following equations respectively:

$$v_{ik+1} = w_k v_{ik} + c_1 r_1 (lbest_{ik} - x_{ik}) + c_2 r_2 (gbest_{ik} - x_{ik}) \tag{38}$$

In the equation above, $w \in [w_{min}, w_{max}]$ is the inertia factor. r_1, r_2 are random numbers belonging to the range [0, 1]. c_1, c_2 specify the importance of the social moves versus cognitive moves. c_1 represents the social learning coefficient for each particle and is useful to calculate the local optimum while updating the velocity vector. In other words, c_1 helps each particle to move based on its own previous experiences. In the same manner, c_2 represents the cognitive learning coefficient which identifies the contribution of the global optimum in recalculating velocity vector. Kennedy and Eberhart (1995) and Shi and Eberhart (1999) have shown that when c_1 and c_2 parameters take the common value of 2, the algorithm performs better. Finally, the next position of the particle is calculated as follow:

$$x_{ik+1} = x_{ik} + v_{ik+1} \tag{39}$$

In order to prevent a particle from leaving far away out of the search space, the elements of the velocity vector are restricted in a predetermined minimum and maximum values. If the velocity of a particle would reach at w_{max}, then the velocity is left at w_{max} and could not exceed w_{max} . Algorithm 1 represents a pseudocode for the PSO problem.

5. Particle Swarm Optimization for congested UMAHLP

Particle swarm optimization (PSO) is a real-coded algorithm and is generally exercised to solve optimization problems in continuous space. For discrete-space problems, the framework of the algorithm would remain same, but it could still be adapted with the help of some hybrid approaches such as genetic operations or list based representation for discrete optimization problems. In a hub location problem, the network is represented as an $n \times n$ matrix to express the flows or costs between nodes. Similarly, the selected nodes to become hubs can be expressed in a binary $n \times 1$

Algorithm 1 .

Identify all parameters needed and assign values.
Initialize all particles;
while termination criteria (optimum or number of iterations) is not met
}
for each particle do
}
Calculate fitness value;
if Fitness value is < particles best-known value then
Set current value as particles best-known value (local optimum);
end if
}
end for
Choose the particle
with the best fitness value of all the particles as the global best position;
for each particle
do
}
Calculate particle velocity;
Update particle position;
Calculate fitness value;
Update the local optimum;
Update the global optimum;
}
end for

matrix where a '1' means an allocation and a '0', otherwise. So, these representations can make our discrete hub location problem solvable by PSO's designed to solve the continuous optimization problems.

In the following, we design a modified PSO algorithm by combining existing PSO algorithm with genetic operators and a local search strategy to update and improve particle positions. Several improvements have been made to enhance the performance of the new algorithm, including among others:

- A specific representation is applied to keep the feasibility of particle positions. Note that this representation does not consist of scalars such as distance of a particle to the best position, but two n dimensional arrays: *LocatingArray* and *AllocatingArray*;
- Most PSO algorithms employ two values, *lbest* and *gbest* which are determined using respective update formulas (The *lbest* and *gbest* denote the objective function values at local best and global best particle positions, respectively). The purpose of updating the particle position is to obtain a new feasible solution. In the proposed algorithm, we used the genetic operator 'mutation' in the update formula to update the particle positions;
- A local search strategy is applied to the initial population and the update process. This strategy decreases the probability of a particle's getting trapped in a local optimum to further enhance the search ability of the algorithm.

The proposed algorithm is detailed in the following subsections (*5.1, 5.2, 5.3, 5.4*).

5.1. Particle Representation

In this algorithm, a feasible solution (i.e., a particle position) is represented by two n dimensional arrays (vectors): *LocatingArray* ($\boldsymbol{x}$) and *AllocatingArray* ($\boldsymbol{h}$). The dimension of these arrays is given by the total number of nodes, n, in the network. *LocatingArray* consists of 0's and 1's, where 1 means that the node is a hub and 0 means that it is not a hub. *AllocatingArray* represents the assignments of non-hub nodes to selected hubs where the nodes take the number of its corresponding hub as a value. Figure 1 illustrates an example of *AllocatingArray* (i.e., [2, 2, 3, 2, 7, 3, 7, 3, 3, 7]) for a network consisting of 10 nodes. Additionally, each hub is evaluated as they assigned to themselves in the *AllocatingArray*. The location of hubs and allocation of non-hubs are described simply and accurately at once by these representations. On the other hand, the algorithm utilizes a cost matrix derived from unit cost matrix of nodes (C_{ij}) through location and allocation assignments of particle by evaluating alternative paths for (i, j) pairs according to minimum cost. A direct link between (i, j) is not permitted, but the algorithm tries to obtain minimum cost for each (i, j) pair by selecting different hubs to assign them.

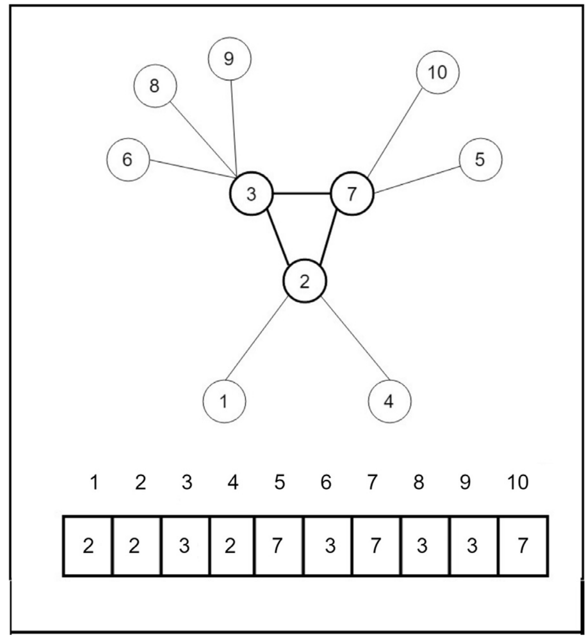

Fig. 1. A list-based network encoding.

5.2. Particle Evaluation

The generation of initial particle positions starts with creating two random arrays: *LocatingArray* and *AllocatingArray*. An $n \times n$ matrix ($\hat{X}$) is generated randomly. After its diagonal elements are sorted sequentially, appropriate number of hubs is selected to form *LocatingArray*. To establish *LocatingArray*, the number of hubs is not predetermined just limited to total number of nodes in network. Differently from earlier uncapacitated models; a predetermined number of hubs (p) to be selected does not exist. So, the number of hubs to select is also randomly determined. Generating of particles sequentially up to population size, all feasible particles, namely swarm, are produced. A fitness function is applied to evaluate the particles. Total cost of the network which includes transportation, opening and congestion costs is utilized as fitness function. As a result, particles of smaller objective values are assigned larger fitness values.

5.3. Particle Update Rule

Traditionally PSO was proposed to solve problems in continuous space, and thus presents extra challenges when it comes to discrete optimization (Bailey, Ombuki-Berman, & Asobiela, 2013). In order to overcome the difficulties of this discrete structure, the update rules of the heuristic algorithm are modified. The algorithm

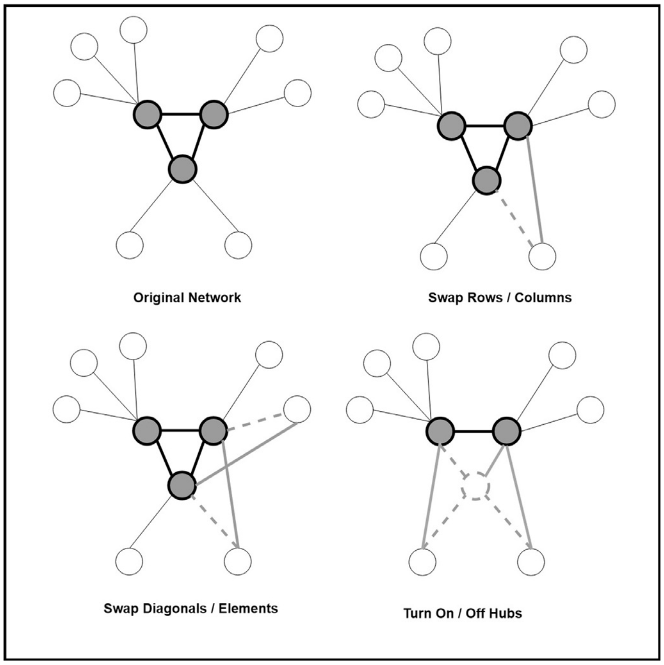

Fig. 2. Swap and change hub functions.

takes the magnitude of the change during particle update to be the difference between a particleâs current position and the best-known position in the swarm and neighborhood. So, the PSO algorithm even in discrete space aims essentially to design an update procedure considering both effect of the local best position and global best position in some manner. Sen and Krishnamoorthy (2018) have stated that this process of information exchange is done through two mechanisms in the literature such as defining an import mechanism to copy certain elements of the best-known solution and defining some genetic operators (crossover, mutation, etc.).

In our algorithm, new feasible solutions are obtained in update procedure by using genetic operator 'mutation'. This mutation operator applies to *LocatingArray* of a particle and corresponding *AllocatingArray* is modified. Genetic operator 'mutation' randomly modifies the particle positions by converting first generated $n \times n$ matrix called '$\hat{X}$'. Swapping rows, columns, elements and diagonals of this matrix or selecting a hub to close or a non-hub to open are the functions of mutation process. After that, the corresponding *AllocatingArray* is also modified. Figure 2 shows some examples of these functions.

Considering premature convergence is still the main deficiency of many PSO algorithm (Hu, Shi, & Eberhart, 2004), it is possible for the proposed algorithm to possibly get trapped in a state where all the particles will become unable to discover better solutions. Thus, the local search strategy is applied to improve particle positions.

5.4. Exploitative Processes

In the literature, one of the reasons that PSO is attractive is that the algorithm converges very fast comparatively to other heuristic algorithms. This speed of the convergence becomes advantageous in the optimization, however it causes a stagnation after finding a local minimum. A local search procedure as an optimization method can become beneficial to master this difficulty and reach at better solutions. The local search procedure developed by Aarts and Lenstra (2003) consists of generating a local optimal solution by exploring the neighborhood of a given initial solution.

Table 2
Particle swarm optimization parameters.

Parameter	Value
Swarm size	150
Iterations	250
Social Learning Factor	1.5
Cognitive Learning Factor	2
Inertia Factor	0.9

However, a local optimal solution in a neighborhood structure is not necessarily a local optimal in another neighborhood structure (Yang, Liu, & Zhang, 2011). For this reason, the use of several neighborhood structures helps guiding the local search to converge less rapidly to a local optimum (Yang et al., 2011).

Our local search procedure attends generating neighborhood solutions by modifying the *AllocatingArray* matrix. It does not affect the selection of hubs, but rearranges the allocation decisions of non-hub nodes. Shifting the assignments of a non-hub node to another hub or swapping the assignments of two non-hub nodes to each other are the operations done by local search. These re-assignments are applied until *AllocatingArray* can not show more improvements.

6. Computational Experiments

In this section, we present some results for the proposed heuristic model that is the subject of this paper. We also discuss the effectiveness of the two functions of congestion: power-law and delay congestion. We present the performance evaluation of this algorithm in terms of both solution quality and computation time. We evaluate the proposed algorithm using the AP data sets. We also compare the results of the various data instances in terms of algorithm performance such as CPU time (seconds) or gap (the relative difference between obtained results (i.e. total costs) and the best result (i.e. "min. total cost") of the heuristic algorithm). The algorithm is coded in MATLAB Version R2012a. Runs were made on a laptop with Intel Core i7 CPU, 3.40 GHz processor and 8 GB RAM. Table 2 describes empirically established the parameter settings used for our heuristic algorithm. All experiments were based on 50 runs of each heuristic algorithm per problem and were terminated after 250 iterations. The learning factors are assigned as the best convergence of the proposed heuristic algorithm.

Our computational experiments were carried out using the AP data sets. The AP data set is derived from a real-world hub location problem for Australia Post and is first described in Ernst and Krishnamoorthy (1996). The AP data set, which includes the fixed costs and capacities for the nodes, can be obtained from OR-Library (Beasley, 1990a). This data set consists of 200 nodes, which are represented by postcode districts. Problems of size n=40; 50; 100; 200 can be obtained from the data set. Other significant differences between the AP and other commonly used sets are that with AP in general we have the following equalities: $\gamma \neq \delta$; $W_{ij} \neq W_{ji}$; $W_{ij} \neq 0$. The economies of scale factors for collection, transfer and distribution costs (α, γ, δ) are valued as (3, 0.75, 2).

For the AP data set, the fixed costs and capacities have already been provided by Ernst and Krishnamoorthy (1999). Two types of fixed costs, loose (L) and tight (T) are considered. In the data set with tight fixed costs, nodes with larger flows have higher fixed costs. So, in general it is harder to choose the nodes which should become hubs in such cases. Similarly, two types of capacities, loose (L) and tight (T) are considered. So, for each problem with n nodes, there are four cases to be considered: LL, LT, TL and TT.

As mentioned above, particle swarm optimization algorithm is implemented using MATLAB. Before investigating the outcomes of the proposed heuristic algorithm related to corresponding versions of HLP, the computational results of the proposed hub location problem with congestion should be compared with different solution procedures. From now on, the heuristic algorithm using MATLAB is compared with both "*Concert Technology*" package of IBM CPLEX 12.7 and its implementation as a default Benders Decomposition. CPLEX 12.7 is helpful to solve HLP with number of nodes less than 40. In larger instances CPLEX is incapable of solving the problem mainly due to the insufficient memory capacities. Benders Decomposition as a feature of CPLEX enhances the capability to solve larger instances under time constraints whereas the "optimality gap" of the results for the congested UMAHLP still remains high, which means sub-optimal solutions. Both CPLEX solvers have relatively smaller computational times while the instances are up to 40 nodes. However, with large scale data and tight constraints the proposed heuristic algorithm achieves less computational time.

As Camargo et al. (2011) stated that the larger instances make the problem unmanageable on regular computers without aid of the decomposition techniques, commercial solver procedures can not find the optimal solution in reasonable times. Previously, Camargo, Miranda, and Luna (2008) remarked that instances from sizes 40 to 200 have not yet solved to optimality for UMAHLP. Since modeling congestion makes the problem non-linear and more complex, the proposed algorithm outperforms the others. Also, Alumur et al. (2018) claimed that larger instances are not solveable within an acceptable "optimality gap" in a reasonable time. They suggested that there is an obvious need to develop tailored exact or heuristic solution methodologies for these problems in the future.

The instances in Table 3 is denoted by the generic notation nFC where n is the number of nodes, F and C are the types of fixed costs and capacities respectively. For example, 40LT means that we are presenting the solution of the respective problem in a network consisiting of 40 nodes with loose fixed costs and tight capacity levels. In Table 3 we summarize the best solutions of the heuristic algorithm for uncapacitated, capacitated and congested UMAHLP with different instances of AP data set. For the congested model, the capacity level is determined as 80%. So, the congestion function is calculated for the excess flow at hubs which is larger than 80% of the hub capacity. The congestion function multiplies the excess flow at hubs by parameters: a scalar 'e' and the average of the total transportation cost of the network '$(\frac{\sum_i C_{ij}}{n})$'. Through the analyses for the different data instances, these parameters '$E = (\frac{e \sum_i C_{ij}}{n})$' are settled as 500.

The first analysis has been made to show the variation of the total cost and selected hubs comparatively for the uncapacitated, capacitated and congested versions of the problem. The obtained results are shown in Table 3. The congested version of the problem calculates the congestion cost proportional between the excess flow and total flow through each hub. This set of experiments are made to sophisticate the problem to learn the different behaviors of the model such as selected hubs or optimum cost range. As shown in Table 3, if number of prefered hubs (p) to locate increases, total cost also increases. Although the uncapacitated model behaves independant from the capacities of hubs and prefers to locate hubs with lower opening costs, the capacitated and congested models tend to select nodes as hub with especially higher capacities and lower opening costs. The difference of the total costs between the models comes from locating more number of hubs to optimize the flow distribution and hub loads. Furthermore, it can be seen from Table 3, the capacity levels of hubs turn out to be quite different in various models. The capacitated model utilizes 100% of the capacity whereas the congested one utilizes 80%.

The hub location problem with congestion resembles the capacitated version of the problem in selecting hubs, especially for the instances with loose capacity. However in the instances of tight capacity level, the congested HLP locates more hubs to balance the flow distribution and hub loads resulting in an increase in the total cost. The results from Table 4 show that the locations of hubs change according to the data instances, congestion models and parameters. In particular, if one assigned more weight to congestion, the allocation of flows to hubs varies accordingly. The total cost of the proposed congested model is higher than in the other versions, because the scalar of the congestion function 'E' suppresses the model in such a way that the congestion cost gets smaller, partly balancing of the total cost. The model works naturally in such a way that hub loads stay strictly in their desired range. As an example, if this scalar 'E' is equal to 50, the total costs of the instances converge to other versions values and the selected hubs show more similarity.

The second analysis has been designed to analyze the hub flow balance induced by the congestion effects and the variation of the cost (transportation/opening/congestion). The instances have been solved without congestion and using distinct functions of the congestion cost. The analysis assesses how the flows through hubs concentrate to the installed hubs and the number of installed hubs varies as a function of the overall contribution of congestion cost.

As seen in Table 5, the number of selected hubs is changing according to the congestion function, even if the congestion does not exist. First power-law function which estimates all flows through hubs endeavors to balance the distribution of flows at hubs. It is not interested in preventing the overload of hubs. That is demonstrated by the transportation cost ratios seen in Table 6 which are less than the cost ratio of other functions. Second power-law function which deals only the excess flow at hubs considers both the avoidance of capacity surpluses and the homogeneity of the flow distribution. Still, there have been some occurrence of overloads of hubs which cause some minor congestion cost in the larger instances.

The proposed congestion cost strictly achieves the avoidance and the homogeneity by increasing the number of selected hubs. Thereby, it has higher opening cost percentage than other functions. The proposed function provides more reliable network in terms of the capacity usage. It is clear from Table 5 that the congestion whichever function is defined imposes the selected hubs on sharing even sum of flows through themselves. If the network has tight capacity, the ratio of congestion cost per total cost tends to be in the range 0% – 5%. Compared to a non-congested network, the flow ratios of hubs become well balanced.

The third analysis assesses how the selected hubs load their associated flows related to their capacity. A hub load is the ratio between the total flow through a hub and its capacity level. It is clearly stated in Table 7 that a hub in the non-congested networks could be loaded with an overhead flow, but in a congested network, congestion functions improve the hub loads. Anyway, the existence of some hub load ratios whose values are greater than 1 verifies that the congestion function aims to minimize the average congestion and does not imitate the capacity constraints. However, the proposed congestion function limits the hub load to a prespecified capacity usage percentage, too.

As shown in Table 7, average hub loads are less than 80%; because the proposed model tries to suppress the hub load less than the capacity usage used in the congestion function to set the minimum congestion cost. However, the model congested by power-law function does not force the model to utilize hubs below the prespecified level. Thereby, the model provides lower total cost as mentioned before. As can be seen from Table 7, for 50TT, 100TT and 200TT instances, congested model with power-

Table 3
Solutions of the uncapacitated capacitated-congested UMAHLP for the AP data sets for Γ=0.8, α=0.75,$E = 500$[a].

	Uncapacitated						Capacitated			Congested		
	'Optimal' Set of Hubs			'Min.' Total Cost			'Optimal' Set of Hubs	# of Hubs	'Min.' Total Cost	'Optimal' Set of Hubs	# of Hubs	'Min.' Total Cost
Instances	p=2	p=3	p=4	p=2	p=3	p=4						
40LL	14-28	11-12-28	6-11-22-28	229,978.7	233,979.4	246,153.9	14-29	2	230,642.3	14-29	2	230,642.3
40TL	14-19	1-14-19	1-14-19-22	284,900.2	302,541.1	321,123.0	14-19	2	284,900.2	14-19	2	287,223.4
40LT	14-28	11-22-28	6-11-22-28	229,978.7	233,980.7	247,345.4	14-26-30	3	258,871.5	14-19-26-30	4	283,111.9
40TT	14-19	14-19-22	1-14-19-22	284,900.2	303,414.1	321,252.6	14-25-40	3	344,864.3	14-19-25-38	4	369,192.3
50LL	15-36	3-27-35	6-22-27-35	228,173.6	234,528.3	247,248.2	15-35	2	229,126.9	15-35	2	229,305.1
50TL	17-48	17-41-48	3-27-41-48	288,757.3	308,189.7	329,536.4	3–24	2	304,902.4	3–24	2	310,497.3
50LT	15-36	3-27-35	3-22-27-35	228,173.6	234,528.3	245,984.0	14-26-35-48	4	272,709.5	14-25-32-46	4	291,300.3
50TT	17-48	24-27-48	3-27-41-48	288,757.3	289,273.7	329,536.4	12-26-41-48	4	389,715.7	6-25-26-41	4	454,483.1
100LL	29-73	29-52-70	29-39-46-73	231,194.7	240,180.2	255,681.1	30-73	2	239,957.0	30-75	2	245,827.1
100TL	5-52	5-19-52	5-19-41-52	298,916.3	320,409.9	348,255.8	44-52	2	343,075.4	44-52	2	345,837.3
100LT	29-73	29-52-70	29-39-46-73	231,194.7	240,180.2	255,681.1	29-72-76	3	265,035.0	26-29-68-69	4	289,697.0
100TT	5-52	5-19-52	5-19-41-52	298,916.3	318,659.2	343,441.8	5-34-41-76	4	482,479.7	5-18-34-41-95	5	547,864.0
	p=5	5-12-19 41-52	376,449.5									
200LL	43-148	40-195-148	30-61-120-148	228,221.2	238,115.4	260,167.6	43-139	2	239,092.7	43-159	2	249,480.6
200TL	50-120	50-90-186	51-54-122-168	265,837.4	277,764.9	286,923.6	95-186	2	281,684.0	73-133-141-158	4	329,721.0
200LT	43-148	40-195-148	30-61-120-148	228,221.2	238,115.4	260,167.6	80-133-170	3	288,161.9	96-111-133-171	4	326,262.8
200TT	50-120	50-90-186	51-54-122-168	265,837.4	277,764.9	286,923.6	52-53-168-186	4	328,402.6	47-54-57-96-186	5	395,954.7
	p=5	25-52-57 113-186	313,676.6									

Table 4
Solutions of different congested UMAHLP for AP data sets for Γ=0.8, α=0.75, E=500[a].

	Congested by $e(g_k)^b$			Congested by $e(g_k - O_k)^b$			Congested by $(\frac{e\sum_i C_{ij}}{n})(g_k - O_k)/g_k$		
Instances[b]	'Optimal' Set of Hubs	# of Hubs	'Min.' Total Cost	'Optimal' Set of Hubs	# of Hubs	'Min.' Total Cost	'Optimal' Set of Hubs	# of Hubs	'Min.' Total Cost
40LL	11-22-28	3	292673.49	14-29	2	230642.33	14-29	2	230642.33
40TL	14-19-22	3	360,427.31	14-19	2	286,714.54	14-19	2	287,223.43
40LT	11-22-28	3	292,476.24	14-26-30	3	272,063.51	14-19-26-30	4	283,111.95
40TT	14-19-22	3	360,749.43	6-14-19-38	4	351,377.04	14-19-25-38	4	369,192.38
50LL	15-27-35	3	295,656.57	15-35	2	229,214.48	15-35	2	229,305.15
50TL	17-48	2	370,061.13	3–24	2	309,337.97	3–24	2	310,497.33
50LT	2-27-35	3	295,637.30	6-26-32-35	4	269,247.98	14-25-32-46	4	291,300.37
50TT	17-48	2	370,061.13	6-26-41-48	4	388,721.77	6-25-26-41	4	454,483.11
100LL	29-52-71	3	306,500.69	30-75	2	245,700.63	30-75	2	245,827.12
100TL	5–52	2	395,220.73	44-52	2	344,548.32	44-52	2	345,837.31
100LT	26-52-70	3	307,500.13	29-72-76	3	270,174.31	26-29-68-69	4	289,697.03
100TT	5–52	2	395,302.57	5-41-52-95	4	450,980.55	5-18-34-41-95	5	547,864.09
200LL	57-95-148	3	316,219.25	43-159	2	244,263.93	43-159	2	249,480.68
200TL	50-120-126	3	348,833.17	95-186	2	291,320.52	73-133-141-158	4	329,721.01
200LT	50-120-126	3	312,385.22	48-133-171	3	300,085.79	96-111-133-171	4	326,262.80
200TT	50-120-126	3	349,662.55	57-113-168-186	4	329,655.07	47-54-57-96-186	5	395,954.70

[a] $(\frac{e\sum_i C_{ij}}{n}) = E$ [b]See Section 6 for the abbrevations

Table 5
Comparison of flow distribution ratios for UMAHLP with non-congested case and three different congestion cases.

	Flow Ratio of Hubs													
	Non Congested		Congested by $e(g_k)^b$			Congested by $e(g_k - O_k)^b$				Congested by $(\frac{e\sum_i C_{ij}}{n})(g_k - O_k)/g_k$				
	Hub no.		Hub no.			Hub no.				Hub no.				
	1	2	1	2	3	1	2	3	4	1	2	3	4	5
40LL	0.38	0.62	0.30	0.28	0.41	0.36	0.64	-	-	0.36	0.64	-	-	-
40TL	1	-	0.30	0.40	0.29	0.41	0.59	-	-	0.43	0.57	-	-	-
40LT	0.38	0.62	0.29	0.30	0.41	0.32	0.34	0.34	-	0.22	0.22	0.29	0.27	-
40TT	1	-	0.30	0.40	0.29	0.20	0.27	0.25	0.28	0.26	0.21	0.29	0.24	-
50LL	0.32	0.68	0.27	0.33	0.41	0.37	0.63	-	-	0.38	0.62	-	-	-
50TL	1	-	0.43	0.57	-	0.34	0.66	-	-	0.36	0.64	-	-	-
50LT	0.32	0.68	0.26	0.31	0.43	0.22	0.26	0.28	0.24	0.18	0.27	0.31	0.24	-
50TT	1	-	0.43	0.57	-	0.23	0.34	0.18	0.25	0.22	0.30	0.29	0.19	-
100LL	0.35	0.65	0.27	0.29	0.44	0.36	0.64	-	-	0.36	0.64	-	-	-
100TL	1	-	0.21	0.79	-	0.57	0.43	-	-	0.58	0.42	-	-	-
100LT	0.35	0.65	0.31	0.31	0.38	0.33	0.34	0.33	-	0.23	0.26	0.25	0.26	-
100TT	1	-	0.38	0.62	-	0.20	0.30	0.17	0.33	0.15	0.15	0.22	0.24	0.25
200LL	0.35	0.65	0.31	0.28	0.41	0.36	0.64	-	-	0.60	0.40	-	-	-
200TL	1	-	0.29	0.44	0.27	0.54	0.46	-	-	0.19	0.28	0.26	0.26	-
200LT	0.35	0.65	0.28	0.34	0.38	0.33	0.35	0.32	-	0.22	0.25	0.28	0.25	-
200TT	1	-	0.24	0.34	0.42	0.26	0.22	0.24	0.28	0.17	0.16	0.18	0.24	0.25

Table 6
Comparison of cost ratios for UMAHLP with different congestion functions.

	Non-Congested				Congested by $e(g_k)^b$				Congested by $e(g_k - O_k)^b$				Congested by $(\frac{e\sum_i C_{ij}}{n})(g_k - O_k)/g_k$			
	No. of Hubs	Cost Ratio			No. of Hubs	Cost Ratio			No. of Hubs	Cost Ratio			No. of Hubs	Cost Ratio		
		TC	FC	CC		TC	FC	CC		TC	FC	CC		TC	FC	CC
40LL	2	0.76	0.24	0	3	0.54	0.27	0.19	2	0.77	0.23	0	2	0.77	0.23	0
40TL	1	0.86	0.14	0	3	0.59	0.26	0.15	2	0.78	0.22	0	2	0.78	0.22	0
40LT	2	0.76	0.24	0	3	0.54	0.27	0.19	3	0.63	0.32	0.05	4	0.61	0.38	0
40TT	1	0.86	0.14	0	3	0.59	0.26	0.26	4	0.55	0.44	0.01	4	0.55	0.45	0
50LL	2	0.75	0.25	0	3	0.54	0.27	0.27	2	0.74	0.26	0	2	0.74	0.26	0
50TL	1	0.82	0.18	0	2	0.52	0.26	0.22	2	0.70	0.30	0	2	0.70	0.30	0
50LT	2	0.75	0.25	0	3	0.55	0.26	0.19	4	0.56	0.43	0.01	4	0.56	0.44	0
50TT	1	0.82	0.18	0	2	0.52	0.26	0.22	4	0.49	0.47	0.04	4	0.51	0.47	0.02
100LL	2	0.77	0.23	0	3	0.56	0.26	0.18	2	0.75	0.25	0	2	0.75	0.25	0
100TL	1	0.89	0.11	0	2	0.63	0.15	0.22	2	0.68	0.32	0	2	0.68	0.32	0
100LT	2	0.77	0.23	0	3	0.57	0.26	0.17	3	0.68	0.29	0.02	4	0.64	0.36	0
100TT	1	0.89	0.11	0	2	0.63	0.15	0.21	4	0.54	0.40	0.06	5	0.53	0.47	0
200LL	2	0.80	0.20	0	3	0.60	0.23	0.17	2	0.76	0.24	0	2	0.76	0.24	0
200TL	1	0.86	0.14	0	3	0.61	0.23	0.16	2	0.82	0.17	0	4	0.64	0.36	0
200LT	2	0.80	0.20	0	3	0.59	0.23	0.17	3	0.64	0.30	0.06	4	0.66	0.34	0
200TT	1	0.86	0.14	0	3	0.62	0.22	0.16	4	0.66	0.32	0.02	5	0.63	0.37	0

Table 7
Comparison of hub loads for UMAHLP with different congestion functions.

	Non-Congested				Congested by $e(g_k - O_k)^b$				Congested by $(\frac{e\sum_i C_{ij}}{n})(g_k - O_k)/g_k$			
	'Optimal' Set of Hubs	Hub Loads			'Optimal' Set of Hubs	Hub Loads			'Optimal' Set of Hubs	Hub Loads		
		MIN	AVG	MAX		MIN	AVG	MAX		MIN	AVG	MAX
40LL	14-28	0.63	3.33	6.03	14-29	0.60	0.67	0.74	14-29	0.60	0.67	0.74
40TL	19	1.38	1.38	1.38	14-19	0.69	0.75	0.81	14-19	0.71	0.75	0.79
40LT	14-28	1.12	2.35	3.57	14-26-30	0.93	0.96	0.98	14-19-26-30	0.66	0.76	0.80
40TT	19	3.70	3.70	3.70	6-14-19-38	0.73	0.85	0.95	14-19-25-38	0.76	0.78	0.80
50LL	15-36	0.65	4.03	7.41	15-35	0.76	0.78	0.80	15-35	0.77	0.78	0.79
50TL	24	1.25	1.25	1.25	3–24	0.45	0.64	0.82	3–24	0.48	0.64	0.80
50LT	15-36	19.6	31.3	43.0	6-26-32-35	0.70	0.80	0.96	14-25-32-46	0.56	0.71	0.78
50TT	24	1.88	1.88	1.88	6-26-41-48	0.75	0.93	1.16	6-25-26-41	0.80	0.80	0.81
100LL	30-73	0.47	0.83	1.19	30-75	0.48	0.63	0.78	30-75	0.48	0.63	0.78
100TL	52	1.88	1.88	1.88	44-52	0.73	0.77	0.81	44-52	0.74	0.77	0.79
100LT	30-73	3.85	4.65	5.45	29-72-76	0.88	0.90	0.92	26-29-68-69	0.69	0.76	0.79
100TT	52	9.45	9.45	9.45	5-41-52-95	0.87	1.15	1.58	5-18-34-41-95	0.43	0.70	0.80
200LL	90-148	1.59	1.63	1.67	43-159	0.58	0.70	0.82	43-159	0.64	0.71	0.77
200TL	122	1.84	1.84	1.84	95-186	0.79	0.80	0.82	73-133-141-158	0.79	0.79	0.80
200LT	90-148	2.44	2.56	2.67	48-133-171	0.97	0.98	0.98	96-111-133-171	0.66	0.74	0.78
200TT	122	4.68	4.68	4.68	57-113-168-186	0.87	0.93	1.01	47-54-57-96-186	0.56	0.69	0.78

Table 8
Network statistics of UMAHLP versus congestion functionâs scalar 'E'.

$(E = \frac{e\sum_i C_{ij}}{n})$	Total Hub Flow Ratio				Imbalance Ratio	Hub Loads				Average Hub Load
0	0.32	0.68			2.16	19.69	43.04			31.36
25	0.21	0.79			3.74	0.78	3.20			1.99
50	0.19	0.28	0.52		2.69	0.72	0.79	2.12		1.21
100	0.17	0.27	0.32	0.24	1.88	0.63	0.75	0.79	0.80	0.74
125	0.18	0.29	0.31	0.23	1.78	0.66	0.80	0.79	0.70	0.73
250	0.20	0.30	0.27	0.23	1.74	0.74	0.79	0.76	0.76	0.76
500	0.18	0.27	0.31	0.24	1.69	0.56	0.72	0.77	0.78	0.71
1000	0.26	0.28	0.28	0.19	1.49	0.79	0.78	0.69	0.61	0.72
5000	0.18	0.29	0.32	0.22	1.48	0.66	0.80	0.77	0.79	0.76

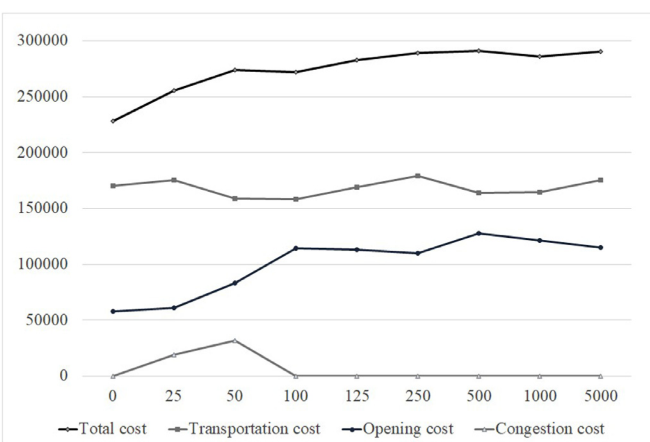

Fig. 3. Variation of cost components versus congestion function scalar 'E'.

law causes some overloads at hubs superior than desired level such as 1.58, 1.16 and 1.15.

The remaining analyses investigate the variations of the model proposed according to the parameters such as capacity level (70%, 85%, 100%), econo-mies of scale (0.2, 0.4, 0.6, 0.8), congestion function scalar 'E' (0, 25, 50, 100, 125, 250, 500, 1000, 5000). These experiments are made with the AP50LT instances. In Table 8, 9, 10 and Figure 3, the sensitivity of the model versus different parameters are summarized.

Figure 3 shows that after the scalar 'E' takes a value larger than 100, the congestion cost becomes zero. Thus, the capacity limitation through hubs is ensured. While the transportation cost stays in the range from 150000 to 200000, the opening costs may vary too much, which may correspondingly increase the total cost.

Furthermore, Table 8 and Table 9 represent the behavior of the model with respect to balance of the hub flows and hub loads. The imbalance ratio of the hub flows decreases inversely proportional to the scalar 'E'. Likewise, an increase of the value of the scalar 'E' pushes the hub loads under the prespecified capacity usage percentage.

Under different capacity usage levels, the model tends to select more hubs. The total cost decreases proportionally to the opening cost ratio. This decrease indicates that the model restricts the selection of hubs with lower opening cost to extend permitted by the capacity usage level. The hubs' loads do not exceed their capacity level, yet the more capacity level increases, the higher average hub load become as shown in Table 9. As mentioned before, the formulations are tested on the AP data set and the data set have the following inequalities: $W_{ij} \neq W_{ji}$; $W_{ij} \neq 0$. Thus, the AP data set is observed as asymmetrical. To observe the sensitivity of economies of scale factors, it is better to modify the data set as symmetrical. Therefore, the values of the scaling factors for collection, transfer and distribution costs (α, γ, δ) are determined as (1, 0.75, 1).

In addition, the tests are made under different values of economies of scale factor such as (0.2, 0.4, 0.6, 0.8) on AP50LT instance. Table 10 demonstrates that the model tends to utilize the hubs less than the other values for the greater economies of scale factor.

The final analysis aims to express the performance of the algorithm relatively to the different versions of congestion (non-congested, power-law and proposed congested) and to evaluate how the particle swarm optimization parameters influence the solution procedure. The performance of the algorithm is examined with the factors such as computational times (CPU), number of it-

Table 9
Network statistics of UMAHLP versus different capacity usages.

Capacity Usage (%)	'Optimal' Set of Hubs	'Min.' Total Cost	Cost Ratio			Flow Ratio					Hub Loads					Average Hub Load
75	6-25-26 32-48	303776.8	0.56	0.44	0	0.19	0.22	0.22	0.23	0.14	0.69	0.60	0.62	0.57	0.64	0.63
80	14-25 32-46	291300.3	0.56	0.44	0	0.18	0.27	0.31	0.24		0.56	0.72	0.77	0.78		0.71
85	14-26 32-38	284024.1	0.57	0.43	0	0.27	0.16	0.33	0.24		0.84	0.44	0.83	0.84		0.73
90	6-26 32-38	275493.1	0.58	0.42	0	0.19	0.22	0.35	0.24		0.71	0.62	0.87	0.84		0.76
100	6-26 32-38	266053.3	0.61	0.39	0	0.21	0.35	0.23	0.21		0.78	0.98	0.58	0.98		0.83

Table 10
Detailed model results versus economies of scale factor for AP50LT instance.

Economies of Scale Factor (α)	'Optimal' Set of Hubs	'Min.' Total Cost	Cost Ratio			Flow Distribution Ratio	Hub Loads			Average Hub Load		
0.2	6-26-32	155157.20	0.48	0.52	0	0.26	0.35	0.39	0.97	0.96	0.96	0.97
0.4	6-26-32	159388.23	0.49	0.51	0	0.27	0.36	0.37	0.88	0.99	0.93	0.93
0.6	12-26-32	163372.48	0.51	0.49	0	0.25	0.35	0.40	0.82	0.99	0.98	0.93
0.8	12-26-32	167556.25	0.52	0.48	0	0.26	0.35	0.39	0.84	0.99	0.96	0.93

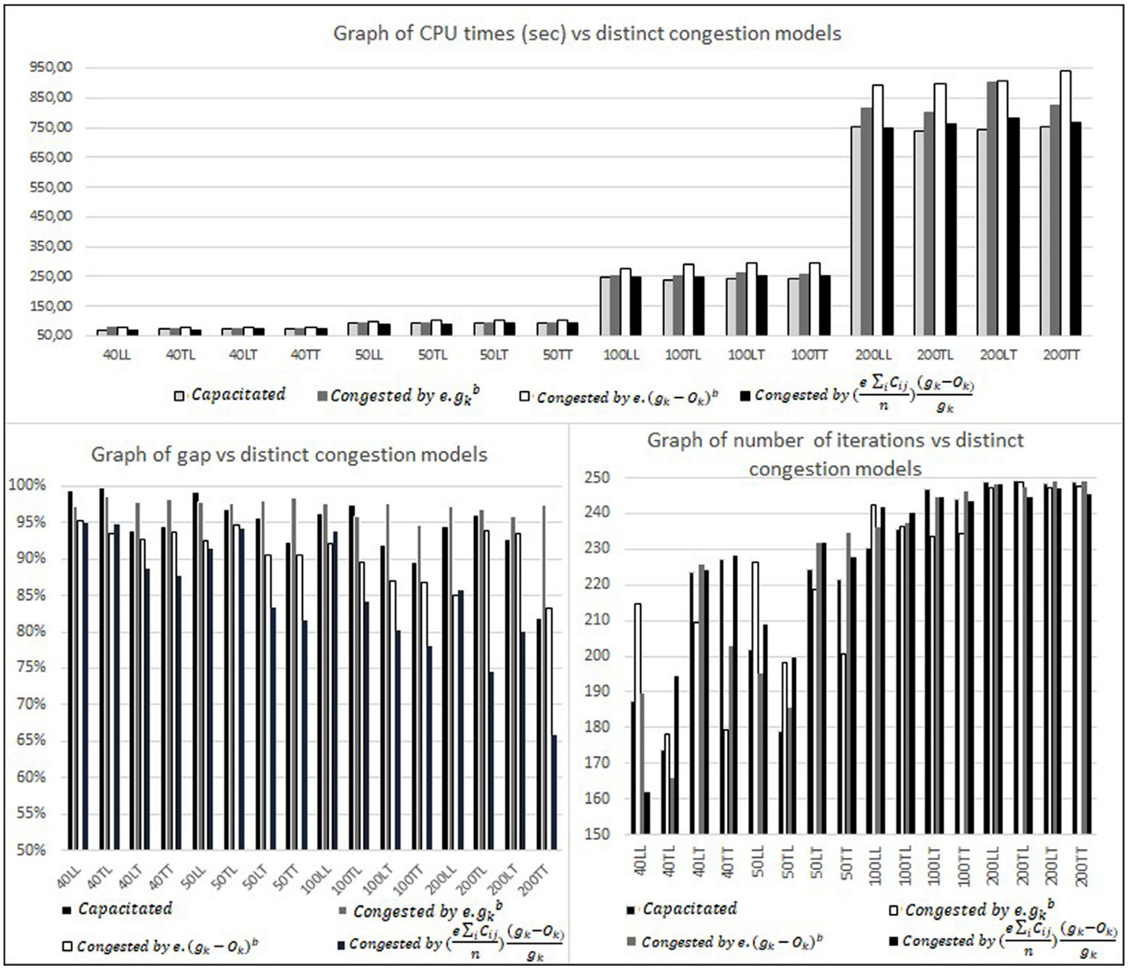

Fig. 4. Performance of the heuristic algorithm.

erations and gap which is defined as the percentage of the relative deviation of each solution of heuristic run from the best solution. The computational times required by MATLAB to solve the problem vary not only with the number of nodes, but also with the values of costs or capacities, the allocation patterns and the congestion factor.

Generally, the problems with tight fixed costs, capacities and with a high congestion factor require higher computational time. Furthermore, the power-law congestion model becomes more time consuming than the proposed model. The computational times increase much more if the power-law function is applied to all flow through hubs. In Table 11 and Figure 4, the outputs of the algorithm for different spectrum of the problems (number of nodes, congestion factor, different set of cost and capacities) can be visualized.

In Table 11, the iteration columns represent the average number of iterations for each heuristic run where the best solution of the algorithm is fixed and achieved. While the number of iterations are closer to 250, the heuristic algorithm might need more iterations to optimize the allocation of the non-hub nodes and the flow distributions as can be seen for large data instances with tight factors. The instances with loose capacities or costs can be optimized elaborately, thus the gap results lie between 90% and 95%. Consequently, the earlier formulations are not as effective as the proposed formulation in consideration of the results.

The particle swarm optimization can find the optimal solution or the best possible solution for most of the problems. There are some irregularities that the heuristic algorithm turned out to get worse results. Generally, increasing number of nodes may cause ineffectiveness for the heuristic algorithms. In case most of these results could be classified as more complicated problems with tight capacities and fixed costs. One of the significant case is AP data instance 200TT with a gap of 75%. These can be explained as our heuristic solution is caught at a locally optimal neighborhood. The enrichment of the particle update and the local search procedure with more genetic operators can improve the results. Although the heuristic algorithm could take more time to compute, increasing the number of iterations in each run may become another improvement. The average computation times are 70, 90, 245, 765 seconds with gaps of 91%, 88%, 84%, 80% for networks with 40, 50, 100, 200 nodes, respectively.

Additionally, the maximum iteration number, swarm size, learning coefficients are varied to observe the heuristics' performance. The experiments show that the more iteration, the less gap achieved. The swarm size is an efficient factor to achieve the optimum, but it has a sharp negative effect on computational time. The learning coefficients may cause the problem to trap in a local optimum.

Figure 5 shows the sensitivity analysis of the heuristic algorithm for the chosen values of the selected parameters. Iteration numbers larger than 200 give better result for all cases, but the computational times show an excessive increase. According to the gap results, it is advantageous to determine the iteration number as 250. The population size behaviors can be explained similarly. The best results are obtained in the range of 150-250 of population.

Table 11
The heuristics' efficiency for different congestion models.

	Capacitated			Congested by $e(g_k)^b$			Congested by $e(g_k - O_k)^b$			Congested by $(\frac{e\sum_i C_{ij}}{n})(g_k - O_k)/g_k$		
	Avg. no. of Iterations	CPU Times (sec)	Gap (%)	Avg. no. of Iterations	CPU Times (sec)	Gap	Avg. no. of Iterations	CPU Times (sec)	Gap	Avg. no. of Iterations	CPU Times (sec)	Gap
40LL	187.35	70.33	99.20	214.60	77.06	97.25	189.70	77.94	95.29	162	69.85	94.90
40TL	173.65	71.47	99.63	178.25	71.86	98.63	166	78.22	93.49	194.40	70.01	94.67
40LT	223.40	72.86	93.83	209.35	73.25	97.83	225.65	79.10	92.60	224	72.02	88.61
40TT	227.24	71.92	94.35	179.45	71.91	98.24	202.65	78.02	93.76	228.30	71.71	87.76
50LL	201.85	92.84	99.16	226.25	93.69	97.87	195.10	100.12	92.42	208.90	89.47	91.38
50TL	178.95	93.11	96.67	198.40	92.98	97.65	185.55	100.29	94.66	199.75	89.82	94.22
50LT	224.40	93.56	95.49	218.60	93.77	98.05	231.70	102.12	90.44	231.75	92.06	83.32
50TT	221.38	93.58	92.21	200.65	93.35	98.35	234.55	102.05	90.45	227.70	91.89	81.56
100LL	230.50	248.05	96.06	242.55	253.56	97.55	236.15	276.31	92.08	241.95	245.29	93.85
100TL	235.45	238.53	97.40	236.55	250.86	95.80	237.45	292.76	89.48	240	244.70	84.10
100LT	246.75	242.28	91.88	233.70	261.58	97.59	244.75	295.64	87.03	244.40	249.69	80.18
100TT	244.15	243.87	89.51	234.35	257.54	94.63	246.05	295.93	86.73	243.25	249.98	78.12
200LL	248.80	751.22	94.27	247.10	816.66	97.20	248.05	892.70	84.95	248.20	748.92	85.65
200TL	249.40	735.89	95.88	248.90	802.38	96.91	247.55	898.88	93.77	244.55	764.53	76.52
200LT	248.50	743.80	92.69	247.14	900.09	95.87	248.85	908.88	93.57	246.80	780.48	79.95
200TT	249	750.26	81.86	247.78	825.84	97.48	249.10	942.69	83.20	245.40	768.60	75.84

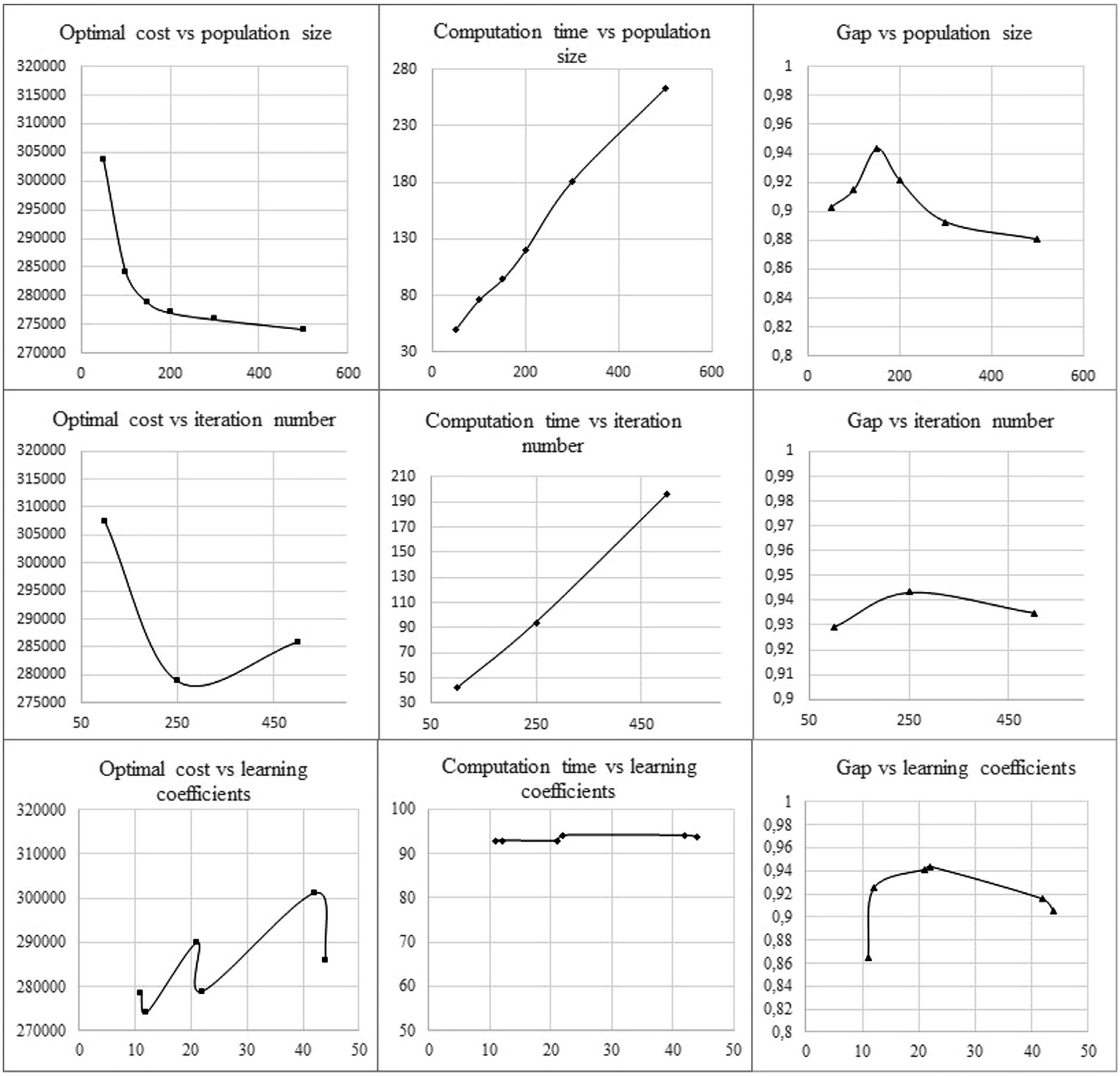

Fig. 5. Sensitivity of the heuristics' performance.

Incorporating of the learning coefficients for the particles can improve the effectiveness of the solution. The computation time is not affected by the change of the learning coefficients. In order to achieve better gap which are closer to optimum, the social learning coefficients should be set in the range from 1 to 2 and the cognitive learning coefficients should be equal to 2. As it can be observed in Figure 5, higher ranges of the learning coefficients produce worse results in terms of gap which are more distant from optimum, because the heuristic algorithm converges fast and the particles are caught by their own local optimum.

Moreover, in order to show the contribution of the genetic operator, mutation used in the update process and in the local search of the particle swarm optimization, the algorithm is applied without mutation and with some of the operations of mutation separately. As mentioned before, swapping rows, columns, elements and diagonals of allocation matrix or selecting a hub to close or a non-hub node to open are the functions of mutation process. These functions are applied to the update process one by one insequence (i.e. not randomly) to see their corresponding improvement effects.

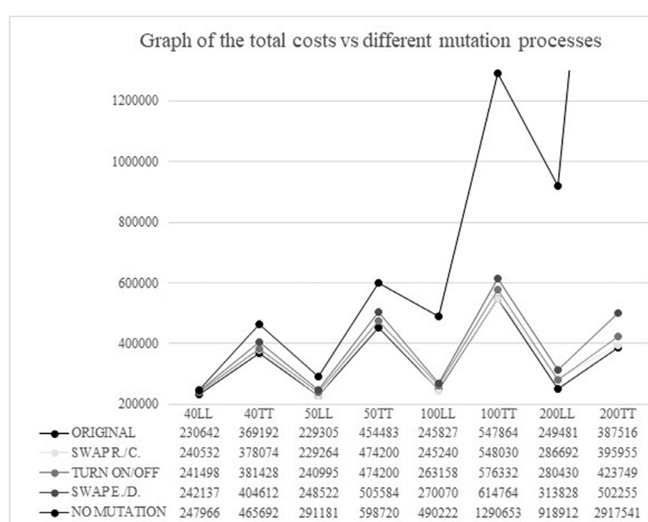

Fig. 6. Comparison of the different mutation processes for pso algortihm.

Figure 6 shows the changes in the solutions of the cases which are update process with swapping rows/columns, swapping elements/diagonals of the allocation matrix, turning on/off hubs and without mutation operator.The results show that smaller total costs are achieved when the mutation operator is applied. The most useful operation of the mutation is swapping rows and columns, since this operation improves simultaneously both hub locations and allocation of non-hubs. On the other hand, the worst performing operation is swapping elements and diagonals of the allocation matrix. This is because this operation focuses only on the allocation decision of the network. In addition, it is important to remark that the computational times decrease by approximately 10%, while the mutation is not utilized in the update process and in the local search.

7. Conclusions

This paper has considered the multiple allocation *p*-hub location problem under congestion. A congestion cost function has been introduced as a convex function which makes the problem a nonlinear mixed integer programming problem. This revised congestion function is defined as a ratio of the excess flow on each hub to the total flow through this hub as mentioned in Equation 36 in Section 3. This function assists in ensuring the capacity limitation of each hub without an explicit determination. Because of this, the proposed non-linear model needs a relatively more effortful solution procedure. A heuristic algorithm, particle swarm optimization, has been applied to solve the problem effectively for the large instances. The particle swarm optimization algorithm makes the solution of the problem easier, especially in dealing with the non-linearity of the model. The algorithm has shown great effectiveness being able to solve large instances under reasonable computational times. However, the exact solution methodologies do not show comparable success to solve the NP-hard problems with some of the instances examined in this paper.

The well-known AP data set is utilized to test the models and the heuristic algorithm. 16 distinct data instances were generated by classifying in number of nodes, tight-loose capacity and fixed costs. The analyses under different scenarios including the modification of congestion function or the variation in congestion factor, capacity level, economies of scale were made. Besides the sensitivity analyses to compare the heuristic efficiency under different pso parameters such as population size, learning coefficients and number of iterations were examined.

The results revealed that a congested network causes higher total cost than a capacitated one for tight capacities of fixed cost. The number of selected hubs becomes the basic factor of that increase. To obtain a more homogeneous network, the congested problem tends to select more hubs and tolerate higher opening costs. Among the different models of congestion, total costs share similarity except from the congestion model which is a convex function of all flow through a hub and ends up with an exorbitant cost. The total cost and selected hubs differs slightly. Only, the proposed model provides more strict capacity usage causing slight increase of the total cost. However, the congested models show some significant differences in terms of the flow ratios, the cost ratio and the hub loads. Firstly, the transportation cost of a non-congested network is higher than a congested network. All congested models do not imitate the capacity constraint as mentioned before, so a minor congestion cost ratio exists in the solution except the proposed model. Subsequently, the flows between non-hub and hub nodes distributed evenly to avoid an overhead loading at hubs. Though the effect of the proposed model on the flow ratio of each hubs is more uniform than the power-law congestion model. Eventually, the hub loads at a non-congested network turn out to be more severe overload such as 7 times more than the capacities of hubs. Whereas the congested networks try to fix up the loading of flows to hubs according to their capacities. Instead of the presence of some hub loads exceed its own capacity at the power-law congestion model, the proposed model provides the hub loads less than a prespecified capacity level.

The analyses of the proposed model under different parameters such as congestion scalar, capacity level and discount factor presented the behaviors of the network under different conditions. It is important to highlight whether the congestion scalar affects excessively, the imbalance ratio between flows on hubs approaches to value of 1. Furthermore, the capacity level becomes dominant on the number of hubs selected. As high as the capacity is utilized, the number of hubs at a network is decreases. Finally, while the discount factor is greater, the hubs selected are utilized under their capacity level. Apart from that, both version of the discount factor tends to design network similarly.

In conclusion all of the previous results show that the congested networks behave unlike from the previous versions of the problem and become a distinctive aspect of the problem which will be investigated deeply as a further study. Congestion could not be thought only a control factor on the capacity of selected hubs, it is an additional cost, which affect the network design to ensure sustainability, reliability and foresight of the networks requirement. Thus, considering both time, capacity and cost components of the congestion becomes a visionary approach. Besides, congestion could be defined in different mathematical and logical formulations. The flow lower than capacity and the flow larger than capacity could be identified as distinct congestion formulations such as cost or delay. For future research, presenting newer functions to characterize the congestion such as linear, exponentially incremental, piecewise linear function opens new prospects to the problem. Thereby, the structure of both network and mathematical model is evolving.

This paper has presented a particle swarm optimization heuristic algorithm which is relatively a new solution procedure used in hub location problem. The analyses have demonstrated that the proposed heuristic algorithm may become an advantageous engineering tool to solve large instances in reasonable times. On the other hand, the proposed heuristic algorithm can be developed to get good upper bounds and get lower gap results for hub location problems. Future research may try to attach more improvement procedures and some genetic operators to update and exploitative processes. Consequently, future researchs can be done to improve the model and proposed algorithm to reflect the real-life requirements of an HLP network. Future works may study other appropriate mathematical models and relevant solution procedures on HLP problems to achieve better computational performances and solutions.

Acknowledgements

The authors wish to thank Professor Temel Öncan of Galatasaray University, İstanbul, for his valuable suggestions and advice at different stages of this work. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors

References

Aarts, E. H., & Lenstra, J. K. (2003). *Local Search in Combinatorial Optimization*. New York, 1997: Wiley.

Adeh, I. M., Baroughi, F., & Alizadeh, B. (2017). A modified particle swarm optimization algorithm for general inverse ordered p-median location problem on network. *Facta Universitatis, Series: Mathematics and Informatics*, 447–468.

Alumur, S., & Kara, B. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research, 190*, 1–21.

Alumur, S. A., Nickel, S., Rohrbeck, B., & Saldanha-da gama, F. (2018). Modeling congestion and service time in hub location problems. *Applied Mathematical Modeling, 55*, 13–32.

Bailey, A., Ombuki-Berman, B., & Asobiela, S. (2013). Discrete PSO for the uncapacitated single allocation hub location problem. In *Computational intelligence in production and logistics systems (CIPLS), 2013 IEEE workshop* (pp. 92–98).

Beasley, J. E. (1990a). OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society, 41*, 1069–1072.

Bryan, D. (1998). Extensions to the hub location problem: Formulations and numerical examples. *Geographical Analysis, 30*(4), 315–330.

Camargo, R. S. d., & Miranda, G. (2012). Single allocation hub location problem under congestion: Network owner and user perspectives. *Expert Systems with Applications, 39*, 3385–3391.

Camargo, R. S. d., Miranda, G., & Ferreira, R. (2009). Multiple allocation hub and spoke network design under hub congestion. *Computers & Operations Research, 36*, 3097–3106.

Camargo, R. S. d., Miranda, G., & Ferreira, R. P. (2011). A hybrid outer approximation/benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters, 39*, 329–337.

Camargo, R. S. d., Miranda, G. D., & Luna, H. P. (2008). Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research, 35*(4), 1047–1064.

Campbell, J. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research, 72*(2), 387–405.

Campbell, J. (1996). Hub location and the p-hub median problem. *Operations Research, 44*, 923–935.

Campbell, J., & O'Kelly, M. (2012). Twenty-five years hub location research. *Transportation Science, 46*(2), 153–169.

Campbell, J. F., Ernst, A. T., & Krishnamoorthy, M. (2002). Hub location problems. In Z. Drezner, & H. W. Hamacher (Eds.), *Facility location: Applications and theory 1* (pp. 373–407). Berlin: Springer.

Contreras, I., Laporte, G., Nickel, S., & Gama, F. S. (2015). Hub location problems. In *Location science* (pp. 311–344). Berlin: Springer.

Costa, M. d., Captivo, M. E., & Climaco, J. (2008). Capacitated single allocation hub location problem-a bi-criteria approach. *Computers & Operations Research, 35*, 3671–3695.

Eberhart, R., & Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. In *Evolutionary computation, 2001. IEEE. proceedings of the 2001 congress: vol. 1* (pp. 81–86).

Ebery, J., Krishnamoorthy, M., Ernst, A., & Boland, N. (2000). The capacitated multiple allocation hub location problem: Formulations and algorithms. *European Journal of Operational Research, 120*(3), 614–631.

Elhedhli, S., & Hu, F. (2005). Hub-and-spoke network design with congestion. *Computers & Operations Research, 32*, 1615–1632.

Elhedhli, S., & Wu, H. (2010). A lagrangean heuristic for hub-and-spoke system design with capacity selection and congestion. *INFORMS Journal on Computing, 22*(2), 282–296.

Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science, 4*(3), 139–154.

Ernst, A. T., & Krishnamoorthy, M. (1998). Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research, 104*(1), 100–112.

Federal Highway Administration (2002). Freight analysis framework (FAF). In *Freight management and operations. u.s. department of transportation. federal publication*. Available via: http://www.ops.fhwa.dot.gov/freight/freight_analysis/faf/(accessed March 25, 2018)

Gillen, D., & Levinson, D. (1999). The full cost of air travel in the california corridor. *Transportation Research Record: Journal of the Transportation Research Board,, 1662*, 1–9.

Grove, P. G., & O'Kelly, M. E. (1986). Hub networks and simulated schedule delay. *Regional Science, 59*(1), 103–119.

Hu, X., Shi, Y., & Eberhart, R. (2004). Recent advances in particle swarm. *Evolutionary Computation, 2004. CEC2004. Congress, 1*, 90–97.

Ishfaq, R., & Sox, C. (2012). Design of intermodal logistics networks with hub delays. *European Journal of Operational Research, 220*, 629–641.

Jia, D. L., Zheng, G. X., Qu, B. Y., & Khan, M. K. (2011). A hybrid particle swarm optimization algorithm for high dimensional problems. *Computers & Industrial Engineering, 61*(4), 521–537.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ*, 1942–1948.

Marianov, V., & Serra, D. (2003). Location models for airline hubs behaving as m/d/c queues. *Computers & Operations Research, 30*(7), 983–1003.

O'Kelly, M. E., & Miller, H. J. (1994). The hub network design problem: a review and synthesis. *Journal of Transport Geography, 2*(1), 31–40.

de Palma, A., & Marshall, F. (1998). From w. vickrey to large-scale dynamic traffic models. In *Proceedings of the 1998 European Transport Conference, Loughborough University, UK*.

Rahimi, Y., Tavakkoli-Moghaddam, R., Mohammadi, M., & Sadeghi, M. (2016). Multi-objective hub network design under uncertainty considering congestion: An m/m/c/k queue system. *Applied Mathematical Modelling, 40*(5-6), 4179–4198.

Rodriguez, V., Alvarez, M., & Barcos, L. (2007). Hub location under capacity constraint. *Transportation Research Part E, 43*, 495–505.

Rodriguez-Martin, I., & Salazar-Gonzalez, J. (2008). Solving a capacitated hub location problem. *European Journal of Operational Research, 1884*, 468–479.

Sen, G., & Krishnamoorthy, M. (2018). Discrete particle swarm optimization algorithms for two variants of the static data segment location problem. *Applied Intelligence, 48*(3), 771–790.

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress, 3*, 1945–1950.

Sohn, J., & Park, S. (2000). The single allocation problem in the interacting three-hub network. *Networks, 35*(1), 17–25.

Wu, W., & Wang, H. (2014). Chaotic particle swarm optimization algorithm for hub and spoke systems with congestion. *Open Automation and Control Systems Journal, 6*, 609–615.

Yang, K., Liu, Y., & Zhang, X. (2011). Stochastic p-hub center problem with discrete time distributions. *Lecture Notes in Computer Science, 6676*(2), 182–191.