Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Solving uncapacitated multiple allocation p-hub center problem by Dijkstra's algorithm-based genetic algorithm and simulated annealing

Masoud Rabbani* **and Seyed Mahmood Kazemi**

School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

C H R O N I C L E

Article history:
Received September 14 2014
Received in Revised Format
January 10 2015
Accepted February 5 2015
Available online
February 8 2015

Keywords:
P-hub center problem
Dijkstra's algorithm
Genetic Algorithm
Simulated Annealing

A B S T R A C T

In the existing literature, there are a huge number of studies focused on p-hub median problems and inventing heuristic or metaheuristic algorithms for solving them. But such analogous body of literature does not exist for its counterpart problem; p-hub center problem. In fact, since p-hub center has been lately introduced and has a particular objective function, minimizing the maximum cost between origin-destination nodes, there are few studies investigating the problem and the challenges for solving it. In this study, after presenting a complete definition of the uncapacitated multiple allocation p-hub center problem (UMApHCP) two well-known metaheuristic algorithms are proposed to solve the problem for small scale and large scale standard data sets. These two algorithms are one single solution-based algorithm, Simulated Annealing (SA), and one population-based metaheuristic, Genetic Algorithm (GA). Because of the particular nature of the problem, Dijkstra's algorithm has been incorporated in the fitness function calculation part of the proposed methods. The numerical results of running the GA and SA for standard test problems show that for smaller scale test problems, single solution-based SA shows greater performance versus GA but for larger scales of data sets the GA generally yield more desirable solutions.

1. Introduction

In some networks, for example, telecommunication or transportations systems, there exist many nodes whose flows should depart from one origin node and arrive in a destination node. Since establishing a direct link between every pair of nodes needs considerable investments and is financially inefficient, some nodes are chosen as hubs, which operate as switching, transshipment and sorting centers and the flow of every node is conveyed to destination points through these nodes.

The problem of optimally locating these hubs and allocating demand nodes to them is called hub location problem. There are two basic types of hub networks: single allocation and multiple allocations. Their difference is in the way of allocating non-hub nodes to hub centers. According to Alumur and Kara (2008) "In single allocation, all the incoming and outgoing traffic of every demand center is routed through a single hub; in multiple allocation, each demand center can receive and send flow through more

* Corresponding author.
E-mail: mrabani@ut.ac.ir (M. Rabbani)

doi: 10.5267/j.ijiec.2015.2.002

than one hub.”. Fig. 1 shows the mentioned difference of single allocation and multiple allocation hub networks.

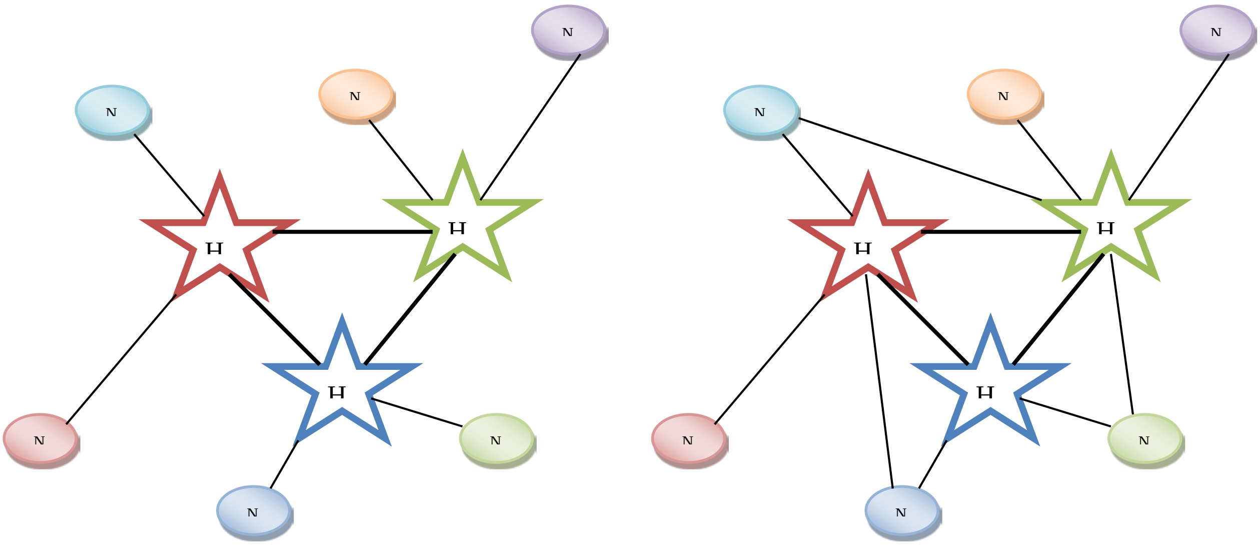

Fig. 1. A single allocation (left hand side) and multiple allocation (right hand side) hub network

The p-hub center problem is a well-known NP-hard problem (Kara & Tansel, 2000) which was firstly introduced and formulated by Campbell (1994). It is worth noting that it is a minimax type problem and is similar to the p-center problem. Campbell (1994) categorized this problem into three different types:

- The maximum cost for movement on any single link (origin-to-hub, hub-to-hub and hub-to-destination) is minimized.
- The maximum cost of movement between a hub and an origin/destination is minimized (vertex center).
- The maximum cost for any origin–destination pair is minimized.

When a hub system involves perishable or time sensitive items and because of that time is really important, we can replace cost with time and in such cases the first type of p-hub center problem becomes important. An example of the second type of p-hub is the vehicle drivers that are subject to a time limit on continuous service. Similar examples to the second type can be given for the third type problem, considering that hub-to-hub links may have some particular characteristics. Campbell (1994) presented formulations for both single and multiple allocation versions of the all three types of p-hub center problem mentioned above.

Based on the results of the study of Alumur and Kara (2008), the total number of papers on the p-hub center problem is very few compared with other hub models. The main reason is that these problems are proposed in 1994 and remain untouched until 2000. However, in the last decade this indicator is growing and the research community is giving more and more attention to the problem.

Kara and Tansel (2000) studied the computational aspects of the single-assignment *p*-hub center problem on the basis of Campbell’s first type model and their proposed model. As reported by the authors, the new model outperformed different linearizations of the basic model regarding CPU times.

Ernst et al. (2009) developed a new formulation for the single allocation p-hub center problem. A new variable r_k was defined to model the maximum collection/distribution cost between hub *k* and the nodes allocated to hub *k*. Their model for single allocation was as follows:

$$\text{Min} \quad Z \tag{1}$$

subject to

$$\sum_k X_{ik} = 1 \qquad \forall i \tag{2}$$

$$\sum_k X_{kk} = p \qquad \forall k \tag{3}$$

$$r_k \geq C_{ik} X_{ik} \qquad \forall i,k \tag{4}$$

$$Z \geq r_k + r_m + \alpha C_{km} \qquad \forall k,m \tag{5}$$

$$X_{ik} \leq X_{kk} \qquad \forall i,k \tag{6}$$

$$X_{ik} \in \{0,1\} \qquad \forall i,k \tag{7}$$

$$r \geq 0 \qquad \forall k \tag{8}$$

Baumgartner (2003) proposed a branch-and-cut algorithm by investigating the polyhedral properties of a formulation proposed by Ernst in an unpublished report and identified some facet-defining inequalities and defined separation procedures. Pamuk and Sepil (2001) addressed the p-hub center problem. They proposed the first heuristic for the single allocation p-hub center problem as a means of generating location-allocation strategies in a reasonable amount of time, and superimposed tabu search on the underlying algorithm, so as to decrease the possibility of being trapped by local optima. Kratica and Stanimirovic (2006) proposed a genetic algorithm with binary coding for the uncapacitated multiple allocation p-hub center problems. They constructed and implemented problem-specific genetic operators in their genetic algorithm. Yaman et al. (2007) analyzed the latest arrival hub location problem with stopovers and included the transient times, the time spent for unloading, sorting and loading at hubs in their model. Campbell et al. (2007) presented complexity results and IP formulations for several versions of the p-hub center allocation problem including both capacitated and uncapacitated cases and established that some special uncapacitated cases are solvable in polynomial times.

Gavriliouk (2009) considered aggregated heuristic procedures for the hub location problems and calculated bounds for errors from such heuristics. Meyer et al. (2009) presented an exact 2-phase algorithm. In the first phase, a set of potential optimal hub locations was computed with a shortest path based B&B algorithm and in the second phase, allocation phase, the optimal allocations were computed accordingly. They also developed an ant colony optimization heuristic for the upper bound needed for the B&B. Sim et al. (2009) presented the stochastic p-hub center problem with chance constraints. To solve the problem, a two stage heuristic approach was also developed. Contreras et al. (2011) considered stochastic uncapacitated hub location problems with uncertain demands and transportation costs. They showed that if the uncertainty was associated with demands, the stochastic problem was equivalent to its expected value problem. On the contrary, if independent transportation costs were uncertain the corresponding stochastic problem would not be equivalent to its expected value problem and some other solution approaches were needed to be developed. Mohammadi et al. (2011a) considered a network with a central mine and a number of factories as customers, and tried to design and schedule transportation of raw material from the mine to its customers using a single allocation hub covering location problem. Afterward the problem was formulated as a mixed-integer programming formulation and two metaheuristics namely, genetic algorithm and shuffled frog leaping algorithm were developed. In another study, Mohammadi et al. (2011b) proposed a new model for capacitated single allocation hub covering location problem and developed a multi-objective imperialist competitive algorithm to solve the problem. Yaman and Elloumi (2012) introduced p-hub center and p-hub median problem with bounded path length. They proposed two integer programming formulations for the star p-hub center problem and three formulations for the star p-hub median problem. Furthermore, they strengthened the last formulation via specific clique inequalities and solved the considered instances to optimality within half an hour. After a while, Liang (2013) analyzed the hardness of the star p-hub center problem. Bashiri et al. (2013) considered a number of qualitative parameters for the hub location problem and proposed a GA based heuristic to solve this problem under capacitated constraints. In order to deal with uncertainty and qualitative parameters, fuzzy systems were utilized. Yang et al. (2013b) proposed a hybrid particle swarm optimization algorithm for fuzzy p-hub center problem. In their problem travel times were considered to be uncertain and modeled by normal fuzzy vectors. In a similar study, Yang et al. (2013a) considered a fuzzy p-hub center problem with fuzzy travel times and developed a genetic algorithm with local search to deal with the problem. Hult et al. (2014) proposed a reformulation for the p-hub center problem when the uncertainty of travel times was considered. Then a number of exact solution approaches based on variable reduction were developed to solve small-medium sized problems to optimality. The literature is rich enough with high-quality reviews in the context of hub location problem. In 2012, Campbell and O'Kelly (2012) provided a comprehensive review on hub location problems and discussed the present status of the literature. They also explored the shortcomings of the literature and suggested future research directions. A short after, Farahani et al. (2013) provided a latest review of models, classification, solution techniques and applications of hub location problems. To provide a concise overview of the literature review, the paramount feature of each studies mentioned above are encapsulated in Table 1.

Table 1
p-Hub center literature

Year	Authors	Notes
1994	Campbell (1994)	Different types of p-hub center formulations
2000	(Kara & Tansel, 2000)	Various linear formulations for single allocation
2001	(Pamuk & Sepil, 2001)	Heuristic for the single allocation problem
2003	(Baumgartner, 2003)	Polyhedral properties, valid inequalities and branch-and-cut algorithm
2007	(Yaman et al., 2007)	Latest arrival hub location problem with stopovers
2006	(Kratica & Stanimirovic, 2006)	GA for the uncapacitated multiple allocation problem
2007	(Campbell et al., 2007)	Complexity results and formulations for the allocation subproblem
2009	(Gavriliouk, 2009)	Heuristic procedures based on aggregation
2009	(Meyer et al., 2009)	Optimal hub locations using a shortest path based B&B and optimal allocations based on reduced size allocation; and ant colony optimization heuristic
2009	(Sim et al., 2009)	Stochastic p-hub center problems with chance constraints
2009	(Ernst et al., 2009)	Proposing integer programming formulations for both single and multiple uncapacitated hub center problems and developing a branch-and-bound approach for the multiple allocation case
2011	(Contreras et al., 2011)	Stochastic uncapacitated hub location problem by considering uncertain demands and transportation costs
2011	(Mohammadi et al., 2011a)	A two-stage capacitated single allocation hub covering location problem is considered and two metaheuristics namely, GA and Shuffled Frog leaping algorithm are proposed
2011	(Mohammadi et al., 2011b)	A new model for the capacitated single allocation hub covering location problem is proposed and a multi-objective imperialist competitive algorithm is applied
2012	(Campbell & O'Kelly, 2012)	A review of the hub location research is provided and the current status of the literature is discussed.
2012	(Yaman & Elloumi, 2012)	star p-hub center and star p-hub median problems with bounded path lengths are introduced.
2013	(Bashiri et al., 2013)	A fuzzy capacitated p-hub center problem is introduced and a genetic algorithm solution is presented.
2013	(Farahani et al., 2013)	A comprehensive survey of hub location problems is presented.
2013	(Liang, 2013)	The hardness of star p-hub center is analyzed.
2013	(Yang et al., 2013b)	A hybrid particle swarm optimization problem for p-hub center problem is developed.
2013	(Yang et al., 2013a)	Fuzzy p-hub center problem is considered and a hybrid metaheuristic consisting of a local search incorporated into a GA is proposed.
2014	(Hult et al., 2014)	For a stochastic uncapacitated single allocation p-hub center problem a number exact computational approaches are developed.
2014	(Yang et al., 2014)	Optimization of a fuzzy p-hub center problem with generalized value-at-risk criterion is considered.

In this study, our focus is on the p-hub center problem which is a minimax type problem and has the number of hubs as an exogenous parameter and previously determined p. The problem considered here is of multiple allocation type problems. Since the problem could be modeled as a graph, where the nodes represents the origin/destination or prospective location and edges represent the linkages between the nodes with weights denoting distances (or costs), Dijkstra's algorithm could be successfully used in order to find the shortest path from each node to any other node. Having the information obtained from running Dijkstra's algorithm once for each node, one can minimize the maximum length between any origin/destination nodes by incorporating this algorithm into a heuristic or metaheuristic method. Achieving this aim is exactly equivalent with solving a UMApHCP.

Hence, in essence, the objective of this study considering the literature provided above, is to propose well-qualified metaheuristic methods for solving the UMApHCP, improve the existing GA in the literature, develop SA for the first time for solving the problem and compare the results of the two algorithms for both small-scale and large-scale data sets. The rest of the paper is organized as follows: In section 2, the problem formulation is presented. In section 3, the proposed algorithms are explained in detail. The numerical experiments are illustrated in section 4. Finally, the concluding remarks are drawn in section 5.

2. Problem definition

The p-hub center multiple allocation problems is to allocate each non-hub node to one or more hubs such that the maximum travel time between any o–d pair is minimized. It is clear that the multiple allocation problems will have an objective function value no larger than that of the single allocation problem since the unique allocation constraints are relaxed in the multiple allocation problems. Thus, the solution to a multiple allocation problem can be used as a lower bound for solving a single allocation problem. Here, the model developed by Ernst et al. (2009) is presented.

$$\text{Min} \quad z \tag{9}$$

$$\sum_{k=1}^{n} z_k = p \tag{10}$$

$$\sum_{k=1}^{n}\sum_{m=1}^{n} y_{ijkm} = 1, \qquad i,j = 1,\dots,n \tag{11}$$

$$\sum_{k=1}^{n} y_{ijkm} \le Z_m \qquad i,j,m = 1,\dots,n \tag{12}$$

$$\sum_{m=1}^{n} y_{ijkm} \le Z_k \qquad i,j,k = 1,\dots,n \tag{13}$$

$$z \ge \sum_{k=1}^{n}\sum_{m=1}^{n} y_{ijkm}\,(c_{ik} + \alpha c_{km} + c_{mj}), \qquad i,j = 1,\dots,n \tag{14}$$

$$Z_k, y_{ijkm} \in \{0,1\} \qquad i,j,k,m = 1,\dots,n \tag{15}$$

Constraint (10) indicates that exactly p hubs are chosen. Constraint (11) together with (15) shows that there is a unique path between each origin–destination pair. Constraints (12) and (13) imply that a node must be selected to be a hub if another node is allocated to it. Constraint (14) defines the lower bound for the objective function ***z***, which represents the maximum transportation cost between all origin–destination pairs.

3. Proposed methods

Based on the results of the study of Alumur and Kara (2008), the total number of papers on the p-hub center problem is very few compared with other hub models. The main reason is that these problems are proposed in 1994 and remain untouched until 2000. These problems are a fairly new research area and there is still a lot of ground to cover; there is a need to develop more exact solution procedures and heuristic algorithms for these problems. To the best knowledge of authors, application of metaheuristics in p-hub center problems is very limited and has not well been studied. Therefore, in this section we propose two metaheuristics for solving UMApHCP; simulated annealing and genetic algorithm.

Because of the nature of the problem and its objective function, which makes it different from p-hub median problems, a special procedure is needed to calculate the fitness function. This procedure is based on Dijkstra's algorithm to find the maximum distance between every origin-destination pairs that is the amount of objective function value. Since the problem could be modeled as a graph, where the nodes represents the origin/destination or prospective hub location and edges represent the linkages between the nodes with weights denoting distances (or costs), Dijkstra's algorithm could be successfully used in order to find the shortest path from each node to any other node. Having the information obtained from running Dijkstra's algorithm once for each node, one can minimize the maximum length between any origin/destination nodes which is exactly equivalent with solving an UMApHCP.

3.1. Genetic Algorithm

Before explaining the main steps of the proposed GA, it is worth mentioning that it is not the first GA implemented for solving UMApHCP. Kratica and Stanimirovic (2006) proposed an efficient GA for the problem and reported the numerical results. The aim of this paper, in addition to improve the performance of their proposed GA, is comparing the performance of a population-based algorithm (GA) and a single solution-based algorithm (SA) while the fitness function calculation procedure operates based on Dijkstra's algorithm.

The pseudo code of the Dijkstra's algorithm is illustrated in Fig. 2 to provide the respected reader with a general view of the algorithm. The obtained results of the SA and GA will guide the respected reader to suitably choose between these two different categories of metaheuristics for coping with different real size problems of UMApHCP. Pseudo code for the proposed GA is presented in the following:

```
STEP 1: Generate Npop random solutions.

STEP 2: Calculate fitness function for each solution

        In this step a Dijkstra's algorithm is implemented to calculate the fitness of each solution.
        (The Dijkstra's pseudo code is shown in Fig. 2). For each solution, Dijkstra's algorithm
        is run by number of genes times. For example, if a chromosome has n genes (number of
        nodes) Dijkstra's algorithm will be run n times for that solution in order to calculate the
        minimum distance of every node from other nodes. The maximum of these distances is
        regarded as the fitness function of that solution and the goal of the GA is minimizing it.

STEP 3: Sort the solutions based on their fitness values.

STEP 4: Move the elitist solution to the new generation
```

STEP 5: Select good solutions by Tournament selection method and crossover a fraction of them and move to the new generation

STEP 5: Mutate a number of solutions and transfer them to the new generation.

STEP 6: Check if the termination criteria are met. If so, terminate the algorithm otherwise, go to STEP 2.

```
Function Dijkstra (Graph, Source):

        for each vertex v in Graph:
            dist[v] := infinity;
            previous pv[ := undefined;
        end for;

        dist[Source] :=0;
        Q := the set of all nodes in graph;

        While Q is not empty:
              u := vertex in Q with smallest distance in dist [];
              if dist [u] = infinity:
                     break;
              endif;
              remove u from Q;
              for each neighbor v of u:
                    alt := dist [u] + dist_between(u, v);
                    if alt < dist [v]:
                          dist[v] := alt;
                          previous [v] := u;
                          decrease-key v in Q;
                    end if;
              end for;
        end while;

    return dist[];
  end Dijkstra.
```

Fig. 2. Dijkstra's algorithm pseudo code (Taken from contributors (2014))

3.1.1. Solution representation

In this GA implementation the binary encoding of individuals is used. Each solution is represented by the binary string of length *n*. Gene 1 in the genetic code denotes that particular hub is established, while 0 shows it is not. Since users can be assigned only to open hub facilities, only array $z_k$is obtained from the genetic code.

3.1.2. Selection

The selection method used in this paper is Tournament selection procedure. In tournament selection, a number *Tour* of individuals is chosen randomly from the population, and the best individual from this group is selected as parent. This process is repeated as often as individuals to choose.

3.1.3. Crossover operator

The Crossover operator employed in the proposed GA is just like the operator employed by Kratica and Stanimirovic (2006). Here, the operator is elaborated in detail. Generally, crossover operator executes a swap between two random parts of a selected pair of parents producing two offsprings. However, because of the specific structure of the solution representation, where number of ones stands for number of hubs, this action will yield infeasible solution. To cope with the problem, Kratica and Stanimirovic (2006) proposed a modified version of the classical crossover operator. In this version, the operator is simultaneously tracing the genetic codes of the parents from right to left searching the position *i* on which the first parent has 1 and second 0. The individuals exchange genes on the found position (that is identified as crossover point), and similar process is performed starting from the left side of genetic codes. Operator seeks for the position *j* where the first parent is 0 and the other one is 1. Genes of the *j*th position are swapped while the total number of located hubs remains unaffected. The process is repeated until $j \geq i$.

3.1.4. Mutation Operator

Each offspring produced by crossover operator is mutated only one time by probability of *MutateRate*. If a random number chosen uniformly between [0,1] is less than *MutateRate*, two genes, one with value 1 and the other with value 0, are randomly selected from the chromosome and their values are exchanged. This Mutation procedure keeps the feasibility characteristics of the solutions.

3.2. Simulated annealing

Simulated annealing whose name come from annealing in metallurgy is a suitable probabilistic algorithm for finding the optimum value of a cost function that may have several local minima. This is a single solution based method that starts with an initial solution and searches the neighbors of that solution in each iteration by a local search mechanism. In fact, SA emulates the physical process of slowly cooling of a material to increase the size of its crystals and reduce their defects. One important advantage of using SA is its simple implementation. The pseudo code is really straightforward and the idea behind the algorithm could be easily understood. The pseudo code of the algorithm used in this paper is briefly presented below:

```
$r = 0, T = T_0, BestSol = \emptyset$
$Generate\ Initial\ solution\ X_0$
$BestSol = X_0$
$\{While\ r < R\ do\ (Outer\ loop)$
    $n = 0$
    $\langle While\ n < N\ do(Inner\ loop)$
        $Select\ a\ move\ randomly\ and\ run\ over\ X_n\ as\ \Delta C = Cost(X_{new}) - Cost(BestSol)$
        $if\ \Delta C < 0\ then$
            $BestSol = X_{new}\ and\ X_n = X_{new}$
        $Else$
            $Generate\ a\ uniform\ random\ number\ from\ [0,1]\ and\ call\ it\ 'y'$
            $Set\ Z = \exp(-\frac{\Delta C}{T_r})$
            $if\ y < Z\ then\ X_n = X_{new}$
    $n = n + 1\rangle$
$r = r + 1$
$T_r = Alpha \times T_{r-1}\}$
$Return\ BestSol\ as\ the\ optimum$
```

Remark. $Cost(X)$ denotes the objective function value of solution X. Thus, a calculation similar to that one mentioned in STEP 2 of the proposed GA is required.

Remark. In the inner loop of the proposed SA, a local search is performed on the current solution. A non-hub node of the current solution is randomly selected and changed to a hub and an arbitrarily chosen hub is changed to a non-hub node. Thus, at each inner loop iteration only two digits of the solution are changed.

The respected reader should keep in mind that the solution representation of the SA is just like the one employed in genetic algorithm. (A string of digits where 1 denotes a hub node and 0 denotes a non-hub one).

4. Results and discussions

In this section, the performance of the proposed algorithms above is assessed and compared using two sets of modified ORLIB instances. The proposed algorithms were programmed in MATLAB (R2009a) software and were run on a PC with 2.2 GHz CPU and 2.0 GB RAM. The first set of test problem instances belong to Civil Aeronautics Board (CAB) data set based on airline passenger flow between cities of the United States. The instances we used for analysis have 25 nodes, 2, 3 or 4 hubs and α values from 0.2 to 1. The other data set is of Australian Post (AP) that is drawn from study of postal delivery system. The instances used in this paper for assessment have 200 nodes, a maximum of 25 hubs and $\alpha = 0.75$.

4.1. Parameter tuning

As is known, the quality of genetic algorithm and simulated annealing solutions is significantly influenced by its parameter. Thus, considering the objective function values of these algorithms as a response of input parameters, response surface methodology (RSM) can be perfectly applied to optimize the parameter values. In most RSM applications, the form of the relationship between the response and the input variables is unknown. Therefore, finding an appropriate approximation for the true functional relationship between y (response) and the set of independent variables is the first step in RSM.

Frequently, a first-order model for beginning is sufficient because the initial parameter values are usually far from the optimum region. Then, RSM moves from the initial values using steepest descent method (if the objective is to minimize a measure) to a better area with regard to the GA response. Afterward, an ANOVA is applied to check the adequacy of a new first-order model with the new parameter values. This approach is repeated until no suitable first-order model is found; this situation usually happens in the areas near the optimum region. In such circumstances a polynomial of higher degree must be used, such as the second-order model shown in Eq. (16).

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{i<j}^{k} \beta_{ij} x_i x_j + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \varepsilon \tag{16}$$

By analyzing the second order model the optimum parameter values will be determined. The convergence of RSM is negatively affected by an increase in the number of independent variables. Thus, it is encouraging to keep the number of input variables at a minimum. To do so, we first set $N = 40, Alpha = 0.9$ for SA and Number of generations = 100 for GA using a trial-and-error approach. Afterward, we let RSM choose the optimum values of T_0, R and Elitist probability, Tour and Mutation rate for both SA and GA and for both CAB and AP data sets. Table 2 shows the initial values we considered for SA and GA parameters.

Table 2
Initial parameter values of SA and GA for RSM

SA initial parameters	T_0	30000
	R	150
GA initial parameters	Elitist probability	0.2
	Tour	4
	Mutation rate	0.1

In order to fit the second-order models, central composite design (CCD) which is a very effective design is employed. Commonly, CCD consists of a 2^kfactorial (or fractional factorial of resolution V) with n_F runs, $2k$ axial runs and $n_C$center runs. Our CCD created by MATLAB software for 3 input variables, contains $n_F = 8$ factorial runs, 6 axial runs and $n_C = 10$ center runs. Considering rotatability property, parameter α (distance of axial runs from the design center) is set to 1.6818 $(= (n_F)^{1/4} = \sqrt[4]{8})$. Rotatability is needed to have a consistent variance of the predicted response at different set of parameter values. The optimum parameter values obtained by RSM methodology for SA and GA based on the experiments designed above are shown in Table 3.

Table 3
Optimal parameter values obtained by RSM for SA and GA

SA optimal parameters for CAB data set	T_0	21760
	R	188
SA optimal parameters for AP data set	T_0	53389
	R	112
GA optimal parameters for both CAB and AP data sets	Elitist probability	0.12
	Tour	3.8
	Mutation rate	0.17

4.2. Discussions

Table 4 shows the results of the SA for different CAB data set instances. Likewise, the outputs of the GA for the same test problem instances are provided in Table 5.

Note. For each instance, each of the algorithms were run 10 times and OF average column denotes the average of obtained objective function values, OF Minimum column shows the minimum amount, the average processing time for each instance has been shown in the fifth column, the best solution already found in the literature is shown in the sixth column and in the last column our algorithm is compared against the best solution in the literature.

Table 4
Results of the SA for different CAB data set instances

α	p	OF average	OF Minimum	Avg. Time(s)	Kratica and Stanimirovic (2006) best solution	Decrease in the best solution
0.2	2	2.0495e+003	2.0495e+003	4.3746	2049.48	0
0.4	2	2.4025e+003	2.4025e+003	4.4543	2402.55	0
0.6	2	2.5587e+003	2.5587e+003	4.4305	2558.74	0
0.8	2	2.7149e+003	2.7149e+003	4.4813	2714.93	0
1	2	2.7392e+003	2.7392e+003	4.4631	2739.22	0
0.2	3	1.7729e+003	1.7651e+003	17.0272	1911.60	7.66%
0.4	3	2.0673e+003	2.0647e+003	17.5873	2064.67	0
0.6	3	2.2438e+003	2.2438e+003	18.1846	2243.77	0
0.8	3	2.5156e+003	2.5156e+003	18.9688	2515.58	0
1	3	2.7258e+003	2.7258e+003	24.5331	2725.79	0
0.2	4	1.6585e+003	1.6195e+003	28.7445	1619.48	0
0.4	4	1.8111e+003	1.7744e+003	25.4180	1774.45	0
0.6	4	2.1293e+003	2.1271e+003	28.0046	2127.13	0
0.8	4	2.4377e+003	2.4377e+003	29.1487	2437.71	0
1	4	2.7258e+003	2.7258e+003	76.9415	2725.79	0

Table 5
Results of the GA for different CAB data set instances

α	p	OF average	OF Minimum	Avg. Time(s)	Kratica and Stanimirovic (2006) best solution	Decrease in the best solution
0.2	2	2.0495e+003	2.0495e+003	6.6355	2049.48	0
0.4	2	2.4025e+003	2.4025e+003	6.6036	2402.55	0
0.6	2	2.5587e+003	2.5587e+003	6.5451	2558.74	0
0.8	2	2.7149e+003	2.7149e+003	6.6089	2714.93	0
1	2	2.7392e+003	2.7392e+003	7.2774	2739.22	0
0.2	3	1.7866e+003	1.7651e+003	7.8678	1911.60	7.66%
0.4	3	2.0885e+003	2.0647e+003	8.0270	2064.67	0
0.6	3	2.2438e+003	2.2438e+003	7.6414	2243.77	0
0.8	3	2.5156e+003	2.5156e+003	7.7504	2515.58	0
1	3	2.7258e+003	2.7258e+003	9.2883	2725.79	0
0.2	4	1.6674e+003	1.6195e+003	9.6749	1619.48	0
0.4	4	1.8669e+003	1.7744e+003	8.9404	1774.45	0
0.6	4	2.1674e+003	2.1271e+003	8.8509	2127.13	0
0.8	4	2.4499e+003	2.4377e+003	8.6370	2437.71	0
1	4	2.7258e+003	2.7258e+003	15.1055	2725.79	0

Observing the results of the proposed algorithms in Tables 4 and 5, the following conclusions can be drawn for CAB data set:

1. For α=0.2 and *p*=3 the proposed algorithms (both SA and GA) were able to find a value 7.66% better than the best value already found in the literature. Also, for all of the other instances both of the proposed algorithms reach to the optimum value. These experimental results imply that the proposed algorithms outperform the already proposed GA in the literature for solving UMApHCP.

2. The proposed GA is computationally more efficient than the SA because needs less time to solve larger CAB instances although for the smaller ones the SA performs better.

3. The average of obtained objective function values for the SA are less than or equal to those of the GA and hence, from the quality of solutions view point the SA shows greater performance.

For AP data set, both SA and GA were run for five times and the best obtained results are presented in Tables 6 and 7.

Table 6
Results of the GA for different AP data set instances

p	Objective function value	Time (s)	Kratica and Stanimirovic (2006) best solution	Decrease in the best solution
2	6.7083e+004	428.5157	67083.28	0
3	6.2946e+004	514.2701	62945.55	0
5	5.7419e+004	650.0875	57419.32	0
10	5.5959e+004	1.0358e+003	55958.75	0
15	5.5959e+004	1.4727e+003	55958.75	0
20	5.5959e+004	2.0195e+003	55958.75	0
25	5.5959e+004	2.2261e+003	55958.75	0

Table 7
Results of the SA for different AP data set instances

p	Objective function value	Time (s)	Kratica and Stanimirovic (2006) best solution	Decrease in the best solution
2	6.7083e+004	691.7911	67083.28	0
3	6.2550e+004	1.2726e+003	62945.55	0.63%
5	5.7419e+004	1.9504e+003	57419.32	0
10	5.5959e+004	1.6374e+003	55958.75	0
15	5.5959e+004	2.1720e+003	55958.75	0
20	5.6003e+004	3.1328e+003	55958.75	0
25	5.5959e+004	3.6111e+003	55958.75	0

As can be seen in Table 7, for AP data set, for *p*=3 the proposed simulated annealing was able to find a value 0.63% better than the best value already found in the literature. Comparing the obtained results for large scale instances of AP test problem, one can see that in all cases both algorithms have reached the best solution reported in the literature. Moreover, in one case the SA algorithm was able to find a solution better than the best solution reported for that specific instance in the literature. These results together with the results obtained for CAB data set imply that the both proposed algorithms perform well and are suitable for solving UMApHCP problem even for large scale instances. Also, it could be concluded that for this problem the proposed GA is computationally more efficient that could be a result of being a population-based algorithm.

5. Conclusion

Since p-hub center has been lately introduced and has a specific objective function (minimizing the maximum cost between origin-destination nodes), there are few studies investigating the problem and the challenges for solving it. Because of the specific structure of the problem Dijkstra's algorithm could be successfully used in order to find the shortest path from each node to any other node. Using the information obtained from running Dijkstra's algorithm once for each node, one can minimize the maximum length between any origin/destination nodes which is exactly equivalent with solving a UMApHCP. In this paper, two well-known metaheuristics were adapted to suit to solving the NP-hard problem. The computational results of applying the proposed algorithms (SA and GA) shows that for smaller scale test problems, single solution-based SA shows greater performance versus GA but for larger scales of data sets the GA generally yield more desirable solutions.

References

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

Bashiri, M., Mirzaei, M., & Randall, M. (2013). Modeling fuzzy capacitated p-hub center problem and a genetic algorithm solution. *Applied Mathematical Modelling, 37*(5), 3513-3525.

Baumgartner, S. (2003). *Polyhedral analysis of hub center problems* (Doctoral dissertation, Technische Universität Kaiserslautern).

Campbell, A. M., Lowe, T. J., & Zhang, L. (2007). The p-hub center allocation problem. *European Journal of Operational Research*, *176*(2), 819-835.

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

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

Contreras, I., Cordeau, J. F., & Laporte, G. (2011). Stochastic uncapacitated hub location. *European Journal of Operational Research*, *212*(3), 518-528.

Contributors, W. (2014). Dijkstra's algorithm. Wikipedia, The Free Encyclopedia. .

Ernst, A. T., Hamacher, H., Jiang, H., Krishnamoorthy, M., & Woeginger, G. (2009). Uncapacitated single and multiple allocation p-hub center problems.*Computers & Operations Research*, *36*(7), 2230-2241.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, *64*(4), 1096-1109.

Gavriliouk, E. O. (2009). Aggregation in hub location problems. *Computers & Operations Research*, *36*(12), 3136-3142.

Hult, E., Jiang, H., & Ralph, D. (2014). Exact computational approaches to a stochastic uncapacitated single allocation p-hub center problem. *Computational Optimization and Applications*, *59*(1-2), 185-200.

Kara, B. Y., & Tansel, B. C. (2000). On the single-assignment p-hub center problem. *European Journal of Operational Research*, *125*(3), 648-655.

Kratica, J., & Stanimirović, Z. (2006). Solving the uncapacitated multiple allocation p-hub center problem by genetic algorithm. *Asia-Pacific Journal of Operational Research*, *23*(04), 425-437.

Liang, H. (2013). The hardness and approximation of the star -hub center problem. *Operations Research Letters*, 41, 138-141.

Meyer, T., Ernst, A. T., & Krishnamoorthy, M. (2009). A 2-phase algorithm for solving the single allocation p-hub center problem. *Computers & Operations Research*, *36*(12), 3143-3151.

Mohammadi, M., Tavakkoli-Moghaddam, R., Ghodratnama, A., & Rostami, H. (2011). Genetic and Improved Shuffled Frog Leaping Algorithms for a 2-Stage Model of a Hub Covering Location Network.

Mohammadi, M., Tavakkoli-Moghaddam, R., & Rostami, H. (2011b). A multi-objective imperialist competitive algorithm for a capacitated hub covering location problem. *International Journal of Industrial Engineering Computations*, 2, 671-688.

Pamuk, F. S., & Sepil, C. (2001). A solution to the hub center problem via a single-relocation algorithm with tabu search. *IIE Transactions*, *33*(5), 399-411.

Sim, T., Lowe, T.J., & Thomas, B.W. (2009). The stochastic -hub center problem with service-level constraints. *Computers & Operations Research* 36(12), 3166-3177.

Yaman, H., & Elloumi, S. (2012). Star p-hub center problem and star p-hub median problem with bounded path lengths. *Computers & Operations Research*, 39, 2725-2732.

Yaman, H., Kara, B.Y., & Tansel, B.Ç. (2007). The latest arrival hub location problem for cargo delivery systems with stopovers. *Transportation Research Part B: Methodological,* 41, 906-919.

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

Yang, K., Liu, Y., &Yang, G. (2013b). An improved hybrid particle swarm optimization algorithm for fuzzy p-hub center problem. *Computers & Industrial Engineering,* 64, 133-142.

Yang, K., Liu, Y., & Yang, G. (2014). Optimizing fuzzy p-hub center problem with generalized value-at-risk criterion. *Applied Mathematical Modelling*, 38, 3987-4005.