IJE TRANSACTIONS C: Aspects Vol. 26, No. 9, (September 2013) 1043-1058

International Journal of Engineering

Journal Homepage: www.ije.ir

Comparing Three Proposed Meta-heuristics to Solve a New *p*-hub Location-allocation Problem

A. Ghodratnama[a], R. Tavakkoli-Moghaddam*[b], A. Baboli[c]

[a] *Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Karaj, Iran*
[b] *Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran,Iran*
[c] *INSA-Lyon, DISP Laboratory, 69621 Villeurbanne Cedex, France*

PAPER INFO

Paper history:
Received 28 March 2013
Received in revised form 02 May 2013
Accepted 16 May 2013

Keywords:
Hub Location-allocation
Vehicle Capacity
Plant Production Capacity
Simulated Annealing
Genetic Algorithm
Particle Swarm Optimization

ABSTRACT

This paper presents a sophisticated mathematical model, in which the location of hubs is fixed and their capacity is determined based on facilities and factories allocated to it. In order to feed the client's nodes, different types of vehicles of different capacities are considered, in which the clients are allocated to hubs, and types and numbers of vehicles are allocated to the factory's facilities. To come up with solutions, we propose to use three meta-heuristic algorithms, namely, genetic algorithm (GA), particle swarm optimization (PSO), and simulated annealing (SA). The efficiency and computational results of these algorithms are compared with one another.

doi: *10.5829/idosi.ije.2013.26.09c.11*

1. INTRODUCTION

In recent years, location and allocation issue has attracted the interests of many researchers, specifically, those who have studied the existing facts and constraints pertaining to transport problems and services provided by potential hub centers to other client nodes. There are different types of hub location problems. In a *p*-hub covering problem, the hub delivers services within a certain radius. The *p*-hub median problem minimizes the transport costs from the point of origin to the destination through the hub nodes. In fixed cost hub problem, minimizing total cost of installation hubs in such a way that operational constraints are to be satisfied, is taken into account, and the *p*-hub center problem minimizes the maximum cost or time in the communication lines.

This paper deals with a *p*-hub median serving a single allocation, which means that a client can be linked with only one hub. New characteristics, such as different capacities for hubs feeding to their clients by their plants and dedicated vehicles, are directly dependent upon the number and kind of established factories. To transport commodities from one feeder node to client nodes, different vehicles of different capacities are used. Thus, the types and number of facilities at different hubs which meet the minimum demands are also considered. We furthermore look at the type and number of vehicles used to transport commodities from the factories located at the hubs. Note that we define the total number and location of hub nodes as serving points, to manufacture raw materials originated from one specific node and thereafter sending the value added products to client nodes or customers.

The concerned objective function has three parts. The first part tries to minimize the transport costs from the origin to the destination, through two mediator hubs, or only through one hub in special cases. Part two minimizes the set-up costs of different factories at various hubs, and part three attempts to minimize the total costs of purchasing different vehicles for the transfer of the goods from the facilities to other nodes.

Taking into account the above points, and because of the NP-hard nature of the primary problems of the *p*-hub median, three meta-heuristic algorithms, namely,

* Corresponding Author Email: tavakoli@ut.ac.ir (R. Tavakkoli-Moghaddam)

genetic, particle swarm, and simulated annealing are used to solve problems. In order to compare results of the algorithm applications, we use the branch and bound method as provided in the Lingo 8 software.

The remainder of this paper is organized as follows: the literature review is provided in section 2. The proposed *p*-hub location allocation mathematical model is elaborated in section 3. The proposed three meta-heuristic algorithms are discussed in section 4. Computational experiments and respective results are reported in section 5. Finally, the concluding remarks are given in section 6.

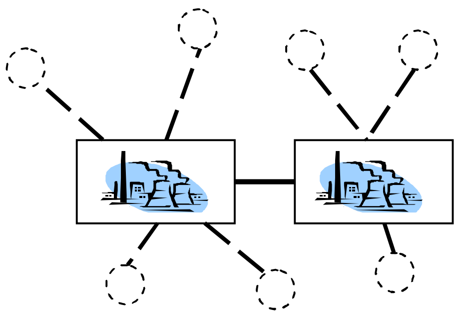

Figure 1. Hubs and their plants serving client nodes

2. LITERATURE REVIEW

The literature on hub location covers a wide variety of models where the most important goal is to minimize some globalizing function of the operation costs. There are many alternatives: non-hub nodes may be allocated to only one single-hub-allocation [1-3] or to several multiple-hubs; allocation of the use of straight links between non-hub nodes may be permitted; the location of some (or all) hubs may be predetermined; hub nodes may be permitted to be located anywhere in a continuous region-continuous hub location problem [4] or may be chosen from a discrete set of place-discrete hub location problems; there may exist a limit in fixing the number of nodes that will be chosen as the *p*-hub problem [5] or a predetermined cost for establishing a hub may be taken into account; instead of choosing nodes to locate hubs, we can select the arcs connecting the non-hub nodes as hub arc location problem [6, 7].

Capacities in hub location problems may have different aspects: there can be capacities on the hub nodes (restraining the volume of flow into the hub [8,9] or for the total flow through the hub) as well as on the flows between hubs or between hubs and non-hubs; on the other hand, a minimum flow amount required to allow for service on the link between a non-hub node and a hub may be present [10]. The first paper on capacitated single-allocation hub location problems was presented by Drezner and Hamacher [11]. Also, further modifications on his model were accomplished by Labbe et al. [12], who described the capacity of the flow that passed through each hub.

Only operation costs related to the flow were investigated, namely, a cost for the flow transported between hubs (transfer cost) and a cost for the flow transported between non-hubs and hubs (the collection and distribution costs coincide). Yaman and Carello considered a hub location problem with modular link capacities in which there was also a limit on the entire flow going through a hub [13]. Only establishment costs were taken into account in this problem, namely, establishment costs for the hubs and for the relations. Exact and heuristic methods were proposed, namely, a branch-and-cut method and a tabu search procedure. Willoughby and Uyeno developed a mixed integer programming location/allocation model that splited the bus assignments when capacity limitations were reached at a transit center [14]. To solve their problem, they proposed a heuristic procedure to assign buses to transit centers (garages) in such a way that all the buses on a particular route were assigned to a single transit center.

Costa et al. presented a bi-objective approach [15], where the model proposed by Ernst and Krishnamoorthy was enlarged with the inclusion of a second objective function to be minimized, that quantified the time to process the flow entering the hubs [9]. Contreras et al. [16] proposed a similar formulation for the same problem studied by Ernst and Krishnamoorthy [9] and used this formulation for developing a Lagrangean relaxation-based procedure. Although most papers have been devoted to minimization of the overall transportation cost (sum), in some cases other objectives have also been taken into account. Many different cost functions have moreover been studied, for example, flow-dependent cost functions [17], and latest arrival time functions [18]. For further study see the surveys by Alumnar and Kara and Campbell et al. and the references therein [19, 20].

To solve the *p*-hub location allocation problems, many methods have been developed. The first approach by O'Kelly is based on an extensive search for all possible choices of *p*-hub locations [3]. In the first allocation method HEUR1, every location is allocated to the nearest hub, while the second HEUR2 allocates every non-hub to the nearest or the second nearest hub. An exchange heuristic based on local improvement was developed by Klincewicz [21], who made an allowance for both the single and double exchange methods. His comparison elucidated the fact that these heuristics are superior to the heuristics proposed by O'Kelly [3]. Klincewicz offered a tabu search and a greedy randomized adaptive search procedure (GRASP) [22]. He considered the traffic among the hubs as well as the distance criteria and also proposed an associated clustering heuristic. A cluster represents one hub with all locations allocated to it. A simulated annealing (SA) method is described by Ernst and Krishnamoorthy [1]. It begins with a randomly generated solution and uses a simple geometric cooling rule with two types of transitions to generate neighborhood solutions. Different from former studies on continuous BAP, a capable method is proposed to deal with the problem to recognize the possible locations for next vessel in the Time-space chart. Then two versions of GRASP are proposed to search for near optimal solutions.

In this paper, we consider the installation cost of facilities in hubs and the purchasing of various vehicles that have the specific capacity to transfer goods between facilities and clients. It is worth noting that there is no research on these two features. We also consider the costs of transporting the produced goods, taking economies of scale into account. We thus assume the existence of a structure that has often been considered in the literatures [9, 15, 23, 24].

Costa et al. presented a different approach to the capacitated hub location problem [15]. In their paper instead of using capacity constraints to limit the amount of flow that can be received by the hubs, they introduce a second objective function into the model (besides the traditional cost minimizing function), in order to minimize the time to process the flow entering the hubs. Correia et al. considered an extension of the capacitated single-allocation hub location problem, in which not only the capacity of the hubs was part of the decision making process, but balancing requirements was also imposed on the network [25]. The decisions to be made included: i) the selection of the hubs, ii) the allocation of the spoke nodes to the hubs, iii) the flow distribution through the sub network defined by the hubs, and iv) the capacity level at which each hub should operate.

Gelareh and Nickel proposed a 4-index formulation for the uncapacitated multiple allocation hub location problem tailored for urban transport and liner shipping network design [26]. Additionally, a very efficient greedy heuristic, proven to be efficient of obtaining high quality solutions was proposed.

Rodriguez-Martin and Salazar-Gonzalez addressed a problem consisting of determining the routes and the hubs to be used in order to send, at a minimum cost, a set of commodities from sources to destinations in a given capacitated network [27]. The capacities and costs of the arcs and hubs were specified, and the arcs linking the hubs were not assumed to generate a complete graph. These authors presented a mixed integer linear programming formulation and described two branch-and-cut algorithms based on decomposition techniques.

Correia et al. revisited a well-known formulation for the capacitated single-allocation hub location problem [28]. In their research, an example was presented showing that for some instances this formulation was incomplete. The reasons for the incompleteness were known, leading to the inclusion of an additional set of constraints. Also, in another research Correia et al. investigated single-assignment hub location problems with multiple capacity levels [29]. In our study, we develop their mathematical model with the main difference that hubs are regarded as serving nodes, in which, plants with different capacities are established to process the raw materials. Also, to transport the raw material from the origin node to hub and final product from hub to destination node, various transporters are used.

Alumnar et al. provided a uniform modeling treatment to all the single allocation variants of the existing hub location problems, under the incomplete hub network design [30]. No network structure other than connectivity was forced on the induced hub network. In this context, the incomplete hub location with predetermined costs, the incomplete hub covering, the single allocation incomplete p-hub median, and the incomplete p-hub center network design problems were introduced, as are efficient mathematical formulations for these problems with $o(n^3)$ variables. Computational results with these formulations were presented on the various instances of the CAB data set and on the Turkish network.

Yaman studied the problem of designing a three-level hub network where the top level consisted of a complete network connecting the so-called central hubs, and the second and third levels were unions of star networks connecting the remaining hubs to central hubs and the demand centers to hubs and central hubs, respectively [31].

Karimi and Bashiri accounted for hub covering location problems with different coverage types [32]. Fazel Zarandi et al. investigated the Q-coverage multiple allocation hub covering problem with mandatory dispersion [33].

Mohammadi et al. [34] presented a new model for the capacitated single-allocation hub-covering location problem. As an alternative to using the capacity constraint to limit the amount of the flows received by the hubs, the second objective function was introduced to minimize service times in the hubs. The service time in the hubs incorporated the waiting time of the received flows in a queue and the time to obtain services. Due to the NP-hardness of the problem, these authors proposed and designed a new weight-based multi-objective imperialist competitive algorithm (MOICA) to find near-optimal solutions.

3. MATHEMATICAL MODEL

The main assumptions, considered in the model formulation are as follows:

- Positions of client nodes are predefined.
- Positions and numbers of hub nodes are predefined.
- Each client node is allocated to only one hub.
- Flows between nodes are predefined.

- Costs of transportation of goods between nodes are calculated in relation to the specific equation.
- Types and capacities of plants producing goods are predefined.
- Cost of installing each plant is predefined.
- Types and capacities of transporters conveying goods are predefined.
- Cost of purchasing each type of transporter is predefined.

The following notations are also used to formulate the problem mathematically.

3. 1. Indices

N={1,…, n}	Set of nodes
i,j	Nodes $i,j \in N$
Q	Number of types of plants
q	Type of plant, q={1,…,Q}
T	Number of types of transporter
t	Type of transporter, t = {1,…,T}

3. 2. Parameters

f_q	Production capacity of type q plant
CP_q	Cost of installing type q plant
g_t	Transportation capacity of type t transporter.
CT_t	Cost of type t transporter
w_{ij}	Flow to be sent from node i to node j
d_{ij}	Distance between nodes i and j ($i,j \in N$)
χ	Cost per unit of flow and per unit of distance between a non-hub node and a hub. This value is usually known as the collection cost.
δ	Cost per unit of flow and per unit of distance between a hub and a non-hub node. This value is usually known as the distribution cost.
α	Cost per unit of flow and per unit of distance between hubs. This value is usually known as the transfer cost and it is often assumed that $0 \le \alpha < 1$. It is assumed that α is lower than the collection and distribution costs.
C_{ijkl}	Total cost for sending one unit of flow from node i to node j through hubs k and l. This means that the flow follows the path $i \rightarrow k \rightarrow l \rightarrow j$ and $C_{ijkl} = C_1 d_{ik} + C_2 d_{kl} + C_3 d_{lj} i, j, k, l \in N$).
X_{kk}	In case, the assigned value is equal to one. It indicates that node k is a hub $(k \in N)$

$O_i = \sum_{j \in N} W_{ij}$: Total flow originated at node i

$D_i = \sum_{j \in N} W_{ij}$: Total flow distained to node i

3. 3. Decision Variables

Y_{ijkl}	Fraction of the flow originated at i destined to j that is routed via hubs k and l in this order ($i, j, k, l \in$N.
X_{ik}	$\begin{cases} 1 & \text{if node } i \text{ is assigned to hub } k \\ 0 & \text{otherwise} \end{cases}$
P_{qk}	Number of type q plants in hub k
Tr_{tqk}	Number of type t transporters for type q plant in hub k

Using the above notations, the proposed mathematical programming model for the concerned p-hub median location allocation problem is as follows:

$$\min \sum_{i \in N} \sum_{i \in N} \sum_{k \in N} \sum_{i \in N} w_{ij} c_{ijkl} y_{ijkl} + \sum_{k \in N} \sum_{q \in Q} CP_q P_{qk} + \sum_{k \in N} \sum_{q \in Q} \sum_{t \in T} C_t Tr_{tqk} \tag{1}$$

s.t.

$$\sum_{k \in N} \sum_{l \in N} y_{ijkl} = 1 \qquad i,l \in N \tag{2}$$

$$\sum_{j \in N} \sum_{l \in N} (w_{ij} y_{ijkl} + w_{ji} y_{jilk}) = (O_i + D_i) x_{ik} \qquad i,k \in N \tag{3}$$

$$\sum_{i \in N} O_i x_{ik} \le \sum_{q \in Q} f_q P_{qk} \qquad k \in N \tag{4}$$

$$\sum_{t \in T} Tr_{tqk} \le \sum_{q \in Q} f_q p_{qk} \qquad k \in N, q \in Q \tag{5}$$

$$x_{ik} \le x_{kk} \qquad i,k \in N \tag{6}$$

$$x_{ik} \in \{0,1\} \qquad i,k \in N \tag{7}$$

$$y_{ijkl} \ge 0 \qquad i,j,k,l \in N \tag{8}$$

$$P_{qk} \ge 0 \qquad q \in Q, k \in N \tag{9}$$

$$Tr_{tqk} \ge 0 \qquad t \in T, q \in Q, k \tag{10}$$

The objective function (1) minimizes the overall cost which is divided into the cost of transportation activities, the total costs of installing plants in hubs, and the total cost of purchasing vehicles dedicated to plants. Constraint (2) ensures that the entire flow is delivered. Constraint (3) ensures that if a node is assigned to a hub then the entire flow which originated from or is destined to the node should go through the hub. Constraint (4) ensures that the total demand of client nodes dedicated to a particular hub node is less than the total capacities of plants installed in related hub nodes. Constraint (5) ensures that the total capacities of various types of transporters dedicated to a particular plant in related hubs are more than the total capacities of a specific type of plant. Constraint (6) ensures that every client node is allocated only to specified hubs. Finally, constraints (7) to (10) are domain constraints. Note that for constraints (9)-(10), the integer variables are more realistic than continuous ones. But for simplicity and knowing that the fractional values does not affect significantly on the other variables and the related objective function as well, we consider them as continuous ones.

4. PROPOSED META-HEURISTIC ALGORITHMS

In this section, three meta-heuristic algorithms are considered to solve the proposed mathematical model. Each subsection introduces the initial researches implemented on the corresponding algorithm and the way of applying each algorithm on the proposed mathematical model.

4. 1. Particle Swarm Optimization Algorithm The particle swarm optimization (PSO) algorithm is an evolutionary computation technique developed by Eberhart and Kennedy [35]. Two years later, Salerno used PSO on a number of neural model architectures solving the XOR problem and then applied that to a real problem [36]. Eberhart and Shi worked on the developments, applications and resources related to PSO in the area of engineering and computer science [37].

As described by Eberhart and Kennedy [35], the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor: a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions. Particle swarm has two main operators: velocity update and position update. During each generation each particle is accelerated into the particles' earlier best position and the global best position. For each iteration, a new velocity value for each particle is considered, based on its current velocity, the distance from its previous best position, and the distance from the global best position. After that the new velocity value is used to calculate the next position of the particle in the solution space. This process is then iterated for a set number of times or until a minimum error is achieved.

4. 2. Simulated Annealing Algorithm Krikpatrick et al. presented the concept of the SA algorithm [38]. This algorithm is a method to solve large combinatorial optimization problems, which is similar to the physical annealing process of solids. Solutions in a combinatorial problem are equivalent to situations of a physical system, and the cost of a solution is equal to the energy of a situation. In this search procedure, SA accepts not only better but also worse adjacent solutions with a definite probability. This means that the SA algorithm has the capability to escape from local minima. It can therefore find high-quality solutions that do not resolutely depend upon the selection of the initial solution compared to local search algorithms. In the other words, this algorithm is efficient and robust. Besides, it has been proven that the processing time of SA has a polynomial upper bound. The SA method consists of four basic segments [39, 40]:

Procedure PSO
Repeat
For i = 1 to a number of individuals, **Do**
If $G(\vec{P}_j) > G(\vec{P}_g)$ **Then** → G() Evaluates goodness
For D = 1 to dimensions, **Do** → P_{id} Is the best state found so far
$P_{id} = X_{id}$.
End For
End If
$g = i$ → Arbitrary
For j = Indices of neighbors, **Do**
If $G(\vec{P}_j) > G(\vec{P}_g)$ **Then**
$g = j$ → G is the index of the best performer in the neighborhood
End If
End For
For d = 1 to number of dimensions, **Do**
$V_{id}(T) = F(X_{id}(T-1), V_{id}(T-1), P_{id}, P_{gd})$ ⊳Update velocity
$V_{id} \in (-Vmax, +Vmax)$
$X_{id}(T) = F(V_{id}(T), X_{id}(T-1))$ ⊳Update position
End For
End For
Until the stopping criterion is met
End Procedure

- The configuration presents all of the possible solutions for the combinatorial problem.
- The move set presents the set of allowable changes. These changes must be able to reach all of the configurations.
- The cost function defines a criterion of how good any given configuration is.
- The cooling plan defines the annealing of the problem from a random solution to a good, frozen solution. It is worth noting that the cooling plan specifies the initial temperature, the rule for decreasing the value of temperature, the number of iterations for searching for better configurations at each temperature, and the time at which annealing should be terminated.

As a whole, one can use the following SA procedure to obtain the solution.

Procedure SA

Initialization

- Initial configuration *s*.
- Get an initial temperature $t > 0$.

While the stop criterion is not met:

Performe the following loop *l* times:

Pick a random neighbor s' of *s*.

Let Δ = Cost(s') - Cost (s)

If $\Delta \le 0$ 0, **Then** set $s = s'$.

If $\Delta \ge 0$, **Then** set $s=s'$ with probability .

$T_k = r \times T_{k-1}; k = 1,2,..., n$

where *r* is a control parameter, small but close to 1.

Return *s*.

End Procedure

In this paper, the generation mechanism of solutions focuses on using a hyperbolic tangent function so that the random number matches the related function attribute. This random value mentioned below.

4. 2. 1. Initial Temperature In physical comparison, the initial temperature should be great enough to heat up the solid until all particles are randomly arranged in the liquid phase. This means that at the beginning, the initial temperature should be high enough to heat up the solid until all particles are randomly arranged in the liquid phase. The temperature of the annealing procedure must therefore be high enough to confirm that the system can be transferred to all possible situations. By this attribute, the algorithm can find a solution that does not robustly depend upon the initial configuration. Since the probability of accepting a worse solution is P_0, the initial temperature T_0 can be specified by means of the cost-increasing transitions, which would be accepted at the beginning of the annealing procedure with a probability *P*. Pilot runs are executed, and the means of cost increasing $\ddot{\Delta}$ transitions is determined. T_0 is calculated by:

$$T_0 = \frac{\dddot{\Delta}}{ln(p_0^{-1})}$$

4. 2. 2. Number of Iterations at each Temperature (l) The annealing procedure transfers from one configuration to one of its neighbors with a definite probability. This is equivalent to the Markov chain. It is therefore essential to set the upper bound of the Markov chain length or the number of iterations at each temperature. The upper bound can have the characteristics of the size of the neighborhood.

4. 2. 3. Rules for Decreasing the Temperature For a specific temperature value, the temperature is reduced when the number of transitions reaches the upper bound of the Markov chain length. The control parameter (i.e., the reduction ratio of temperature) is generally chosen for small temperature changes. The Markov chain facilitates an easier move to an equilibrium state if the temperature change is small. Hence, we use the decrement rule as follows:

$$T_k = r \times T_{k-1}; k = 1,2,..., n$$

The control parameter *r* is small and close to 1.

4. 2. 4. Stopping Condition The annealing process is ended when the system is frozen (i.e. the value of the objective function of the feasible solution does not get better after a definite number of successive Markov chains). In this paper, the process is terminated if the current best configuration remains unchanged for $\ln|\theta|$ number of temperature reduction steps. Aarts and Korst [41] have shown that the upper bound of the total number of temperature reduction steps (i.e., the number of Markov chains) is proportional to $\ln|\theta|$; θ is the solution space that denotes the finite set of all possible solutions. In this paper, θ is equivalent to the factorial of *N*, namely number of client nodes. Most of the elements in θ, however, are infeasible solutions because there are too many zoning constraints, so we use $\ln|\theta|$ as the upper bound of the number of Markov chains.

4. 3. Genetic Algorithm Concepts of genetic algorithm (GA) were first devised by Holland [42] who subsequently developed this algorithm extensively [43-45]. To adapt his concepts to specific fields, other authors also processed them to a large extent [46].

The basic structure of a GA can be summarized as follows. First, a mechanism is provided whereby each problem solution is converted into a chromosome. A set of chromosomes, which is in fact a series of solutions, is then produced as an initial population. The size of the problem is flexible and determined by the user and is often created randomly. After this stage, the main challenge will be to establish new chromosomes called children, using genetic operations. This operation can be divided into two major species, i.e. crossover and mutation. Crossover rate and mutation rate are two factors that have many applications in selecting chromosomes that play parent roles. These factors are determined by the user before starting to apply the algorithm. After producing a new series of chromosomes or children, the process of selecting the fittest chromosome by evolutionary operations gets underway. In this process, the fittest parents and children are chosen so that the final number of chromosomes is equal to the initial population size. The process of selection is based on the fitness value of each string. In fact, the evaluation process is the most pivotal issue in the selection process. Up to this stage, one repeat or generation of the algorithm is done. The algorithm gradually converges towards the optimal solution after several generations. The termination criteria of the problem are the navigation of a certain number of iterations determined by the user before the start of the algorithm. The basic structure of a GA can be shown in pseudo-code as follows.

Procedure GA
  Initialization
    Parameter setting
      - P_c
      - P_m
      - Stop criteria
      - Pop size
      - Selection strategy
      - Crossover operator
      - Mutation operator
      - Perform scalability
  Initialize population
    - Randomly
    Fitness evaluation
      - Repeat
      - New generation
    Individual selection for mating pool
    - Size of mating pool is equal to pop size
      For each consecutive pair apply crossover (for each consecutive pair apply crossover with probability P_c
      Mutate children
        - Mutation with probability P_m
      Replace the current population by the resulting mating pool
      Fitness evaluation
      Until stopping criteria are met
      End For
End

4. 3. 1. Roulette Wheel Base Selection Method In this method, first the fitness value is calculated for each chromosome of the population. The total fitness is computed, and then a random number between zero and the total fitness is selected. Better chromosomes have higher chances of selection and the selection chance of each chromosome is commensurate with its fitness value. With this method the possibility of selection is directly proportional to the fitness value.

4. 3. 2. Changing Methods Making changes to chromosomes can be approached in two basic ways. The first and simplest is called mutation. Like mutations in organisms that involve changing one gene into another, GA mutation makes a small change to one point of the attributes' code.

4. 3. 2. 1. Gene Mutation Before the chromosomes are transmitted to the next generation, they are likely to undergo abrupt changes or mutations. A mutation is a sudden change in a gene. The degree of mutation indicates the possibility of mutation in a gene.

The mutation approach used in this paper is as follows. One gene is randomly selected, and after selection a random number that falls in the specified range of variables is generated, producing the new chromosome with a mutated gene. The fitness value of the chromosome prior to and after the swap is compared to ensure efficiency. If the fitness value is unchanged the chromosome is restored to its original state. Figure 2 indicates this procedure.

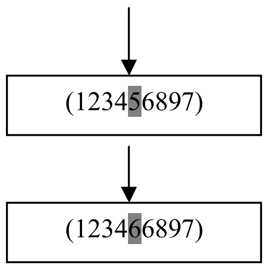

Figure 2. Mutation operator

4. 3. 2. 2. Crossover The second method is called crossover, in which two chromosomes are selected for the purpose of changing their code segments. This process simulates the combining of chromosomes during reproduction in living organisms. Most common crossover methods involve a single-point crossover in which the change point is positioned randomly among the chromosomes. The first section is before the first point and the second section continues after it. In this method every part is from a different parent and has an equal probability of being selected.

The two chromosomes each give some of their genes to create the next generation. If they remain unchanged they will be transferred to the next generation. The degree of crossover indicates that the chromosomes do change at times. It is approximately 65 to 85 percent.

One-point Crossover

First, a random number in the (1, length-1) range is produced. The two chromosomes are then broken and combined at a point. In this paper, one-point crossover is also investigated, as shown in Figure 3.

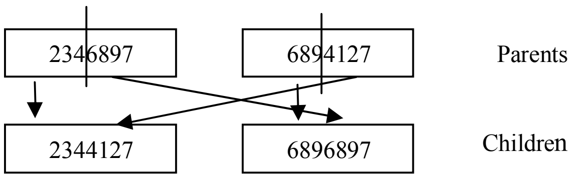

Figure 3. One-point crossover

4. 3. 3. Selection of a New Generation The selection process for the purpose of generating a new generation is similar to parent selection, with the difference that the frequency of its occurrence is equal to the number of first-generation chromosomes.

4. 3. 4. Stopping Criterion In this paper, the stopping criterion is determined by the number of generations.

5. COMPUTATIONAL RESULTS

The presented model is solved by the Lingo 8 software and the proposed algorithms. The model is run on a computer with a capacity of 2.25 GHz and 3.00 GB capability. The proposed algorithms are coded in MATLAB in a Windows XP environment. The related results are compared in a pair wise manner. A number of test problems are generated randomly and then solved. The following parameters are considered in this paper.

- Positions fall in the range of [0,100]
- Weights fall in the range of [10, 80].
- C_1 falls in the range of [80,100].
- C_3 falls in the range of [$0.8\times C_1$, C1].
- C_2 falls in the range of [$0.4\times(\frac{C_1+C_3}{2})$, $0.6\times(\frac{C_1+C_3}{2})$]
- A transporter capacity falls in the range of [0.25×Average weight, 0.5×Average weight].
- Plant capacity falls in the range of [0.5×Average capacity, Average capacity].
- A transporter cost falls in the range of [0.75×Average cost, 1.25×Average cost].
- A plant cost falls in the range of [0.75×Average cost, 1.25×Average cost].

5. 1. Tuning the Structural Parameters of the Algorithms To find the optimum or best values of the structural parameters of each algorithm, we apply the Taguchi method in which the orthogonal arrays are used widely. The main aim of this method is to carry out the factorial analysis on a small scale. To do this, we choose four of the most important parameters of each algorithm and dedicate three levels to each of them. An orthogonal array relating to this specific plan that has 27 combinations of the related levels of chosen parameters is used. The Minitab software is used to design the experiment (DOE). A test problem with forty nodes, ten hubs, five types of facilities and four types of transporters is considered for running the three algorithms.

5. 1. 1. PSO Algorithm For the proposed PSO, four parameters, namely, size of population, maximum iteration, personal and global learning factors, and inertia factor are investigated. Note that for each parameter, three levels are considered. Table 1 shows these factors and related levels. After 27 runs, each with specified parameter levels, and analyzed with Minitab software, the related results are obtained as depicted in Figure 4. These results show that the larger population, maximum iteration, and personal and global learning factors generate the less OVF and the higher inertia factor results the more objective function value. To obtain the results within a reasonable time and to come close to DOE results, we set the number of populations at 50, maximum iteration at 300, personal and global learning factors at 1.5, and inertia factor at 0.72.

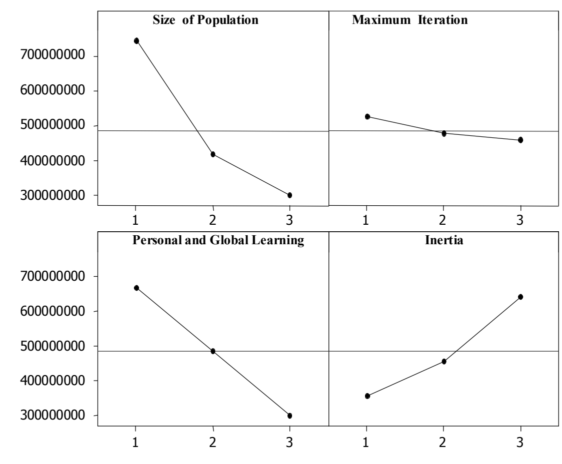

Figure 4. Outputs of the DOE analysis for the proposed PSO

TABLE 1. PSO parameters and related levels

Factor		Level		
		1	**2**	**3**
1	Size of population	25	50	75
2	Maximum iteration	100	200	300
3	Personal and global learning factors	0.5	1.5	2.5
4	Inertial factor	0.5	0.75	1

TABLE 2. SA parameters and related levels

Factor		Level		
		1	**2**	**3**
1	Length of Markov chain	2	3	4
2	Temperature decreasing steps	100	200	300
3	Moves to neighbors	3	5	7
4	Decreasing multiplier	0.4	0.8	0.95

TABLE 3. GA parameters and related levels

Factor		Level		
		1	**2**	**3**
1	Size of population	10	20	30
2	Maximum iteration	100	200	300
3	Mutation rate	0.4	0.7	1
4	Crossover rate	0.4	0.7	1

5. 1. 2. SA Algorithm For the proposed SA, four parameters, namely, length of Markov chain, maximum number of temperature decreasing steps, number of moves to neighbors, and decreasing temperature multiplier, are taken into account. Note that for each of these parameters, three levels are considered. Table 2 illustrates these parameters and related levels. After 27 runs, each with specified parameter levels, and analyzed with Minitab software, the associated results are obtained as depicted in Figure 5. The outputs show that the longer Marko chain, temperature decreasing steps and moves to neighbors result in the less objective function. For a decreasing multiplier there is no significant difference between the three levels. To obtain the results within a reasonable time and to come close to the DOE outputs, we set the Markov chain length at 3, temperature decreasing steps at 500, moves to neighbors at 5, and decreasing multiplier at 0.95.

5. 1. 3. GA Algorithm For the proposed GA, four parameters, namely, size of population, maximum iterations, mutation rate and crossover rate are taken into account. For each of these factors, three levels are investigated. Table 3 elucidates these parameters and related levels. After 27 runs, each with the specified parameter levels, and analyzed with Minitab software, the related results are obtained as shown in Figure 6. The outputs show that the larger population, maximum iteration, mutation rate and crossover rate result in the less objective function. To obtain the results within a reasonable time and to come close to the DOE results, we set the number of populations at 25, maximum iteration at 500, crossover rate at 0.8 and mutation rate at 0.3.

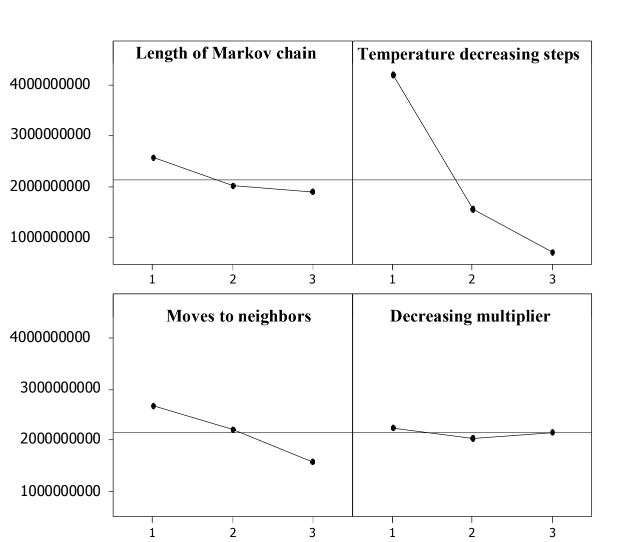

Figure 5. Outputs of the DOE analysis for the proposed SA

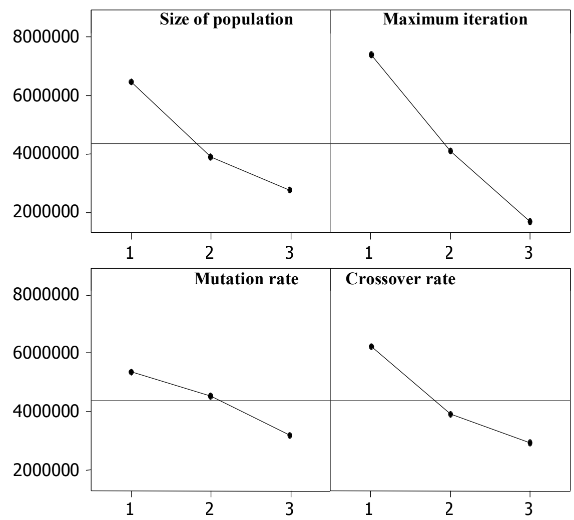

Figure 6. Outputs of DOE analysis for the proposed GA

5. 2. Sensitivity Analysis To make a sensitivity analysis, the following five important and efficient factors are evaluated:

- Problem dimensions
- Number of vehicle types
- Number of plant types
- Coefficient of weights
- Coefficient of distances

5.2.1. Analysis Based on Problem Dimensions In view of the above, the effect of the problem dimension is evaluated for a variety, ranging between three clients and 40 clients. Table 4 and Figures 7 and 8 show the computational results in detail. Note that the objective function values reported by Lingo 8 software and the other three meta-heuristic algorithms are very close. Obviously, the Lingo software cannot solve the problems involving more than seven clients. This importance arises from to be NP_hard of the related problem. Additionally, the results show that when the problem size increases, the objective function values for all algorithms grow exponentially.

Furthermore, the computational time for all algorithms tends to increase. Statistically, the ANOVA hypothesis test shows that in 0.95 significant level there is no significant difference between these algorithms regarding the objective function value (OFV) measure.

However, when it comes to the computation time, a significant difference does exist. As a whole, the proposed SA algorithm introduces the best results simultaneously in terms of the OFV and the computational time. The second suitable algorithm is the proposed GA. However, it is worth mentioning that all the proposed algorithms have acceptable outputs. As a whole, the proposed SA algorithm introduces the best results simultaneously in terms of the OFV and the computational time. The second suitable algorithm is the proposed GA. However, it is worth mentioning that all the proposed algorithms have acceptable outputs.

5. 2. 2. Analysis Based on the Number of Vehicles For more in-depth analysis, a problem with 40 clients is considered. Clearly, the Lingo 8 software is incapable of solving the models but proposed algorithms are used to survey the impact of the number of vehicle types. Table 5 and Figures 9 and 10 show that the impact of this feature is not important. Statistically, the ANOVA hypothesis test shows that in 0.95 significant level, there is a significant difference between these algorithms regarding the OVF and the computational time measures all together. As a whole, the SA algorithm is the most effective of the three algorithms with respect to the processing time and objective function values criteria put together. As the second and third ranks, GA and PSO algorithms are considered consecutively.

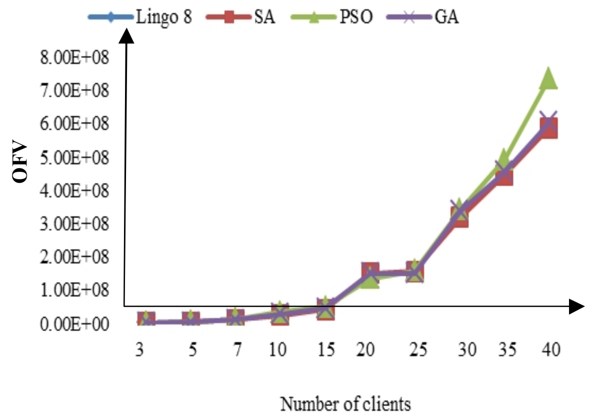

Figure 7. Comparison of the outputs based on the number of clients vs. the OFV

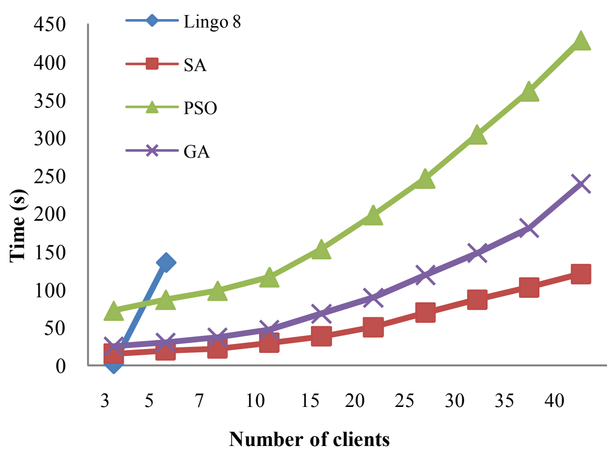

Figure 8. Comparison of the outputs based on the number of clients vs. the computational time

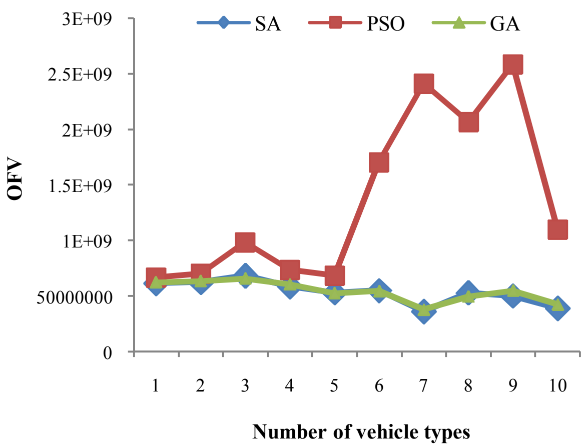

Figure 9. Comparison of the outputs in terms of the number of vehicle types vs. the OFV

TABLE 4. Comparison of outputs in terms of the computational time and objective function

Test problem	No. of clients	No. of hubs	Computational time (s)				Objective function value			
			Lingo 8	SA	PSO	GA	Lingo 8	SA	PSO	GA
1	3	1	2	15	72	25	734506	734697	1442898	1099052
2	5	2	135	19	86	30	3296251	3422290	4979382	4203082
3	7	3		22	98	37		10189154	12657520	10269177
4	10	4		29	116	47		24982801	33974360	28754274
5	15	5		38	153	68		41079108	49215348	45273432
6	20	6		50	198	89		150376959	134785664	150970639
7	25	7		69	246	119		156931884	157616712	150110712
8	30	8		86	304	148		318704156	341858535	337313852
9	35	9		102	361	181		446974225	491566270	456608088
10	40	10		120	428	239		587772228	737720874	605894013

TABLE 5. Comparison of the outputs based on the number of vehicle types in terms of the OVF and the computational time

Test problem	No. of vehicle types	Computational time (s)			Objective function value		
		SA	PSO	GA	SA	PSO	GA
1	1	125	432	218	617109970	669232593	625555152
2	2	126	427	221	625841892	703453206	637149889
3	3	120	431	221	685570082	984294174	660161172
4	4	120	428	239	587772228	737720874	605894013
5	5	121	444	221	531846155	685270821	527844048
6	6	122	429	221	551560402	1705744660	547084118
7	7	133	434	220	363806126	2413057857	381485987
8	8	123	434	221	531801558	2069838090	497897759
9	9	125	438	229	503779317	2585031776	551754320
10	10	125	430	222	392125462	1098836844	426563913

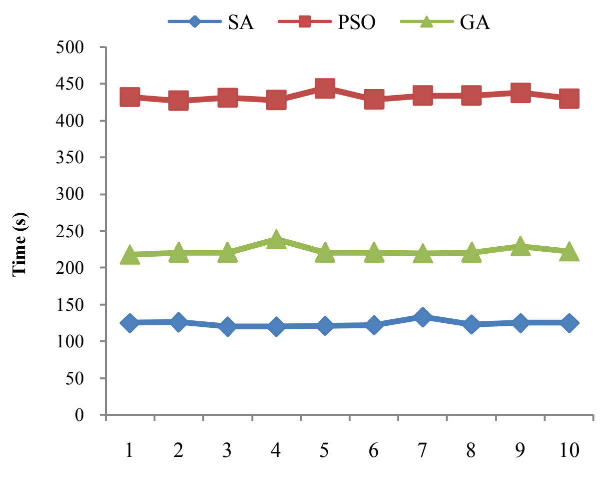

Figure 10. Comparison of the outputs based on the number of vehicle types vs. the computational time

5. 2. 3. Analysis Based on the Number of Plant Types Related to this, the results show that when the number of plant types increases, the cost function decreases considerably, especially if the number of plants increases from one to three. The reason behind this can be the trade-off between the number of plants and the other features, which has a tremendous impact. Table 6 and Figures 11 and 12 show the related results. Statistically, the ANOVA hypothesis test shows that in 0.95 significant level, there is a significant difference among these algorithms regarding the computational time measure. However, a big difference between algorithms related to the objective function measure is not reported. As a whole, three proposed algorithms have the same behavior, but as before the proposed SA algorithm outperforms the other two regarding the processing time mostly, and even more so the objective function criteria. The next best algorithm considering these two measures is GA and the last one is PSO. Note that the Lingo software is incapable of solving the considered problem up to 40 clients.

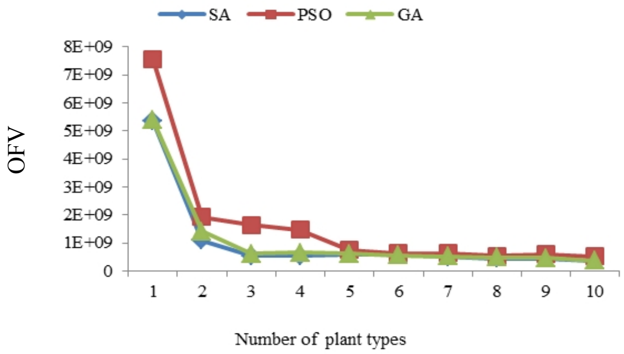

Figure 11. Comparison of the outputs in terms of the number of plant types vs. the OFV

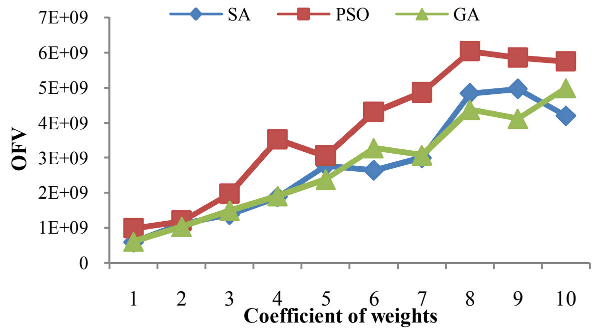

Figure 13. Comparison of outputs based on the coefficient of weights vs. the OFV

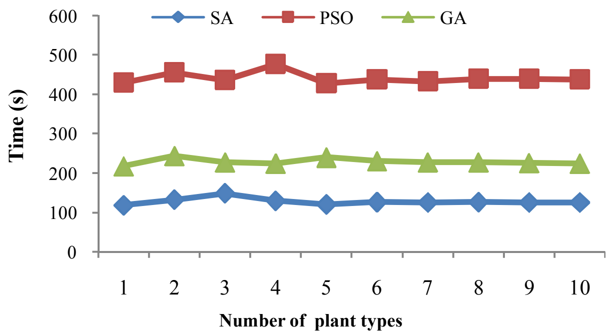

Figure 12. Comparison of outputs based on the number of plant types vs. the computational time

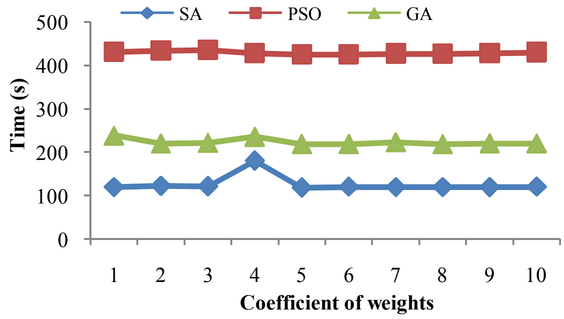

Figure 14. Comparison of outputs based on the coefficient of weights vs. the computational time

TABLE 6. Comparison of the outputs based on the number of plant types

Test problem	No. of plant types	Computational time (s)			Objective function value		
		SA	PSO	GA	SA	PSO	GA
1	1	118	430	217	5338781180	7538457315	5387144390
2	2	132	456	243	1094460462	1930764781	1408173207
3	3	148	436	227	536608643	1635927266	618161468
4	4	129	477	224	530186458	1467187013	643649201
5	5	120	428	239	587772228	737720874	605894013
6	6	126	438	230	570977266	636959030	550704464
7	7	125	433	227	497141929	634554907	518650348
8	8	126	439	227	448173476	539289647	479058744
9	9	125	439	225	438273942	595854991	459099023
10	10	125	437	224	368072926	514038186	365776736

TABLE 7. Comparison of the outputs based on the coefficient of weight in terms of the OVF and computational time

Test problem	Coefficient of weight	Time (s)			Objective function		
		SA	PSO	GA	SA	PSO	GA
1	1	120	431	239	587772228	984294174	605894013
2	2	123	434	220	1091717628	1192889409	1020727626
3	3	122	435	222	1368515195	1976269600	1486194571
4	4	181	428	236	1873030012	3513765423	1900757542
5	5	119	425	219	2764401845	3054726895	2369530604
6	6	121	425	219	2633583524	4310486813	3272230597
7	7	120	427	223	2990812987	4861727193	3060946719
8	8	120	426	219	4826933280	6037262112	4369947366
9	9	120	428	221	4952833901	5858538820	4103637804
10	10	121	430	221	4183719596	5745199660	4985728487

TABLE 8. Comparison of the outputs based on coefficient of distance in terms of the OVF and computational time

Test problem	Coefficient of distance	Computational time (s)			Objective function value		
		SA	PSO	GA	SA	PSO	GA
1	1	120	431	239	587772228	984294174	605894013
2	2	124	430	223	826021138	1243934910	890522866
3	3	121	435	224	1346259157	2958138131	1439928464
4	4	124	432	222	2208312539	2512563394	2552027357
5	5	122	440	223	2119487386	2683599132	2761001100
6	6	123	435	225	3968028787	4642243889	4764989635
7	7	123	442	226	2837065965	4193329660	3672363027
8	8	124	441	229	3999894256	7869075742	4363899935
9	9	125	437	225	4339911433	6658532669	4670921822
10	10	124	445	235	3585669956	4225698901	4355522704

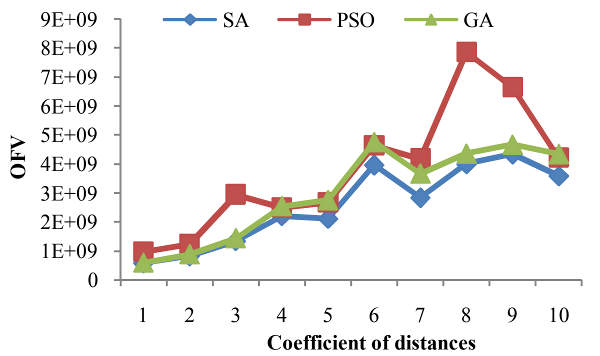

Figure 15. Comparison of the outputs based on the coefficient of distances vs. the OFV

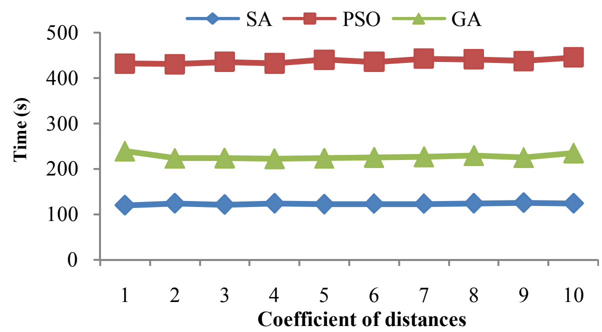

Figure 16. Comparison of the outputs based on the coefficient of distances vs. the computational time

5. 2. 4. Analysis Based on the Coefficient of Weights Considering this feature, when the related coefficient increases, the cost function rises steeply. The reason is clearly that the large flows among clients result in high transportation costs. Statistically, the ANOVA hypothesis test shows that in 0.95 significant level there is a significant difference between these algorithms regarding the computational time measure. However, there is no difference between the proposed algorithms regarding the objective function measure. Generally, as before, the capability order of these algorithms can be the GA first, SA second, and PSO last, regarding the objective function measure. But when it comes to the computational time criteria, the SA algorithm is first and GA and PSO algorithms are ranked second and third, respectively. Note that the Lingo 8 software is incapable of solving this problem with up to forty clients. Table 7 and Figures 13 and 14 show the computational results regarding this feature.

5. 2. 5. Analysis Based on the Coefficient of Distances This feature corresponds to the coefficient of weight features, meaning that greater distances result in higher costs. Rationally this impact is expected, because greater distances generate the higher transportation costs. Statistically, the ANOVA hypothesis test shows that in 0.95 significant level there are no large differences between algorithms regarding the objective function criteria. Likewise, regarding the computational time criteria, no difference between the algorithms is reported. As a whole, three algorithms have the same manners, but the SA algorithm strongly outperforms the other two concerning the processing time and, to a lesser extent, the objective function criteria. The next best algorithm, considering these two measures, is GA and the last one is PSO. Note that the aggrandized behavior of GA and SA algorithms has been emerged in PSO pattern. Also, the Lingo software is unable to solve the considered problem up to 40 clients. Table 8 and Figures 15 and 16 show the computational results for this feature

6. CONCLUSIONS

This paper has presented a *p*-hub median problem with two new features, namely plants and their corresponding transporters, to meet the demand of client nodes. Since this type of location allocation problem belongs to a NP-hard class, obtaining an optimal solution by common software packages is almost impossible, or at best it is very time consuming. Thus, three meta-heuristic algorithms based on simulated annealing (S A), particle swarm optimization (PSO) and genetic algorithms (GA) have been proposed to solve this type of hard problem. To ensure that each algorithm is applied under the optimum conditions, the design of experiment (DOE) method has been used to obtain the best values of the structural parameters of these algorithms. The results obtained from the proposed algorithms are compared with those results reported by the Lingo 8 software using the branch-and-bound method for small problems, in order to demonstrate the efficiency and capability of our proposed algorithms. A wide sensitivity analysis has also been performed to demonstrate the impact of the number of plants, number of vehicle types, coefficients of weights and distances. The results show that the largest number of plant types generates the lowest costs. But, by increasing the number of vehicle types, coefficients of weights and distances, the related cost increases steeply. All in all, the proposed algorithms act equally and introduce solutions of high quality. In a myopic analysis, however, the proposed SA algorithm outperforms the other two, regarding both cost and computational time criteria.

7. REFERENCES

1. Ernst, A. T. and Krishnamoorthy, M., "Efficient algorithms for the uncapacitated single allocation p-hub median problem", ***Location Science***, Vol. 4, No. 3, (1996), 139-154.
2. Ernst, A. T. and Krishnamoorthy, M., "An exact solution approach based on shortest-paths for p-hub median problems", ***INFORMS Journal on Computing***, Vol. 10, No. 2, (1998), 149-162.
3. O'kelly, M. E., "A quadratic integer program for the location of interacting hub facilities", ***European Journal of Operational Research***, Vol. 32, No. 3, (1987), 393-404.
4. O'Kelly, M. E. and Miller, H. J., "Solution strategies for the single facility minimax hub location problem", ***Papers in Regional Science***, Vol. 70, No. 4, (1991), 367-380.
5. Skorin-Kapov, D., Skorin-Kapov, J. and O'Kelly, M., "Tight linear programming relaxations of uncapacitated p-hub median problems", ***European Journal of Operational Research***, Vol. 94, No. 3, (1996), 582-593.
6. Campbell, J. F., Ernst, A. T. and Krishnamoorthy, M., "Hub arc location problems: Part I—introduction and results", ***Management Science***, Vol. 51, No. 10, (2005), 1540-1555.
7. Campbell, J. F., Ernst, A. T. and Krishnamoorthy, M., "Hub arc location problems: Part II—formulations and optimal algorithms", ***Management Science***, Vol. 51, No. 10, (2005), 1556-1571.
8. Ebery, J., Krishnamoorthy, M., Ernst, A. and Boland, N., "The capacitated multiple allocation hub location problem: Formulations and algorithms", ***European Journal of Operational Research***, Vol. 120, No. 3, (2000), 614-631.
9. Ernst, A. T. and Krishnamoorthy, M., "Solution algorithms for the capacitated single allocation hub location problem", ***Annals of Operations Research***, Vol. 86, No., (1999), 141-159.
10. Lee, H., Shi, Y., Nazem, S. M., Yeol Kang, S., Ho Park, T., and Ho Sohn, M., "Multicriteria hub decision making for rural area telecommunication networks", ***European Journal of Operational Research***, Vol. 133, No. 3, (2001), 483-495.
11. Drezner, Z. and Hamacher, H. W., "Facility location: Applications and theory", Springer, (2004).
12. Labbé, M., Yaman, H. and Gourdin, E., "A branch and cut algorithm for hub location problems with single assignment", ***Mathematical programming***, Vol. 102, No. 2, (2005), 371-405.
13. Yaman, H. and Carello, G., "Solving the hub location problem with modular link capacities", ***Computers & Operations Research***, Vol. 32, No. 12, (2005), 3227-3245.
14. Willoughby, K. A. and Uyeno, D. H., "Resolving splits in location/allocation modeling: A heuristic procedure for transit center decisions", ***Transportation Research Part E: Logistics and Transportation Review***, Vol. 37, No. 1, (2001), 71-83.
15. da Graça Costa, M., Captivo, M. E. and Clímaco, J., "Capacitated single allocation hub location problem—a bi-criteria approach", ***Computers & Operations Research***, Vol. 35, No. 11, (2008), 3671-3695.
16. Contreras, I., Díaz, J. A. and Fernández, E., "Lagrangean relaxation for the capacitated hub location problem with single assignment", ***OR Spectrum***, Vol. 31, No. 3, (2009), 483-505.
17. O'Kelly, M. E. and Bryan, D., "Interfacility interaction in models of hub and spoke networks", ***Journal of Regional Science***, Vol. 42, No. 1, (2002), 145-164.
18. Kara, B. Y. and Tansel, B. Ç., "The latest arrival hub location problem", ***Management Science***, Vol. 47, No. 10, (2001), 1408-1420.
19. Alumur, S. and Kara, B. Y., "Network hub location problems: The state of the art", ***European Journal of Operational Research***, Vol. 190, No. 1, (2008), 1-21.
20. Campbell, J. F., Ernst, A. T. and Krishnamoorthy, M., "Hub location problems", ***Facility location: applications and theory***, Vol. 1, (2002), 373-407.
21. Klincewicz, J. G., "Heuristics for the p-hub location problem", ***European Journal of Operational Research***, Vol. 53, No. 1, (1991), 25-37.
22. Klincewicz, J. G., "Avoiding local optima in thep-hub location problem using tabu search and grasp", ***Annals of Operations Research***, Vol. 40, No. 1, (1992), 283-302.
23. Contreras, I., Fernandez, E. and Marin, A., "Tight bounds from a path based formul for the tree of hub location problem", ***Computers & Operations Research***, Vol. 36, No. 12, (2009), 3117-3127.
24. Contreras, I., Fernández, E. and Marín, A., "The tree of hubs location problem", ***European Journal of Operational Research***, Vol. 202, No. 2, (2010), 390-400.
25. Correia, I., Nickel, S. and Saldanha-da-Gama, F., "Hub and spoke network design with single-assignment, capacity decisions and balancing requirements", ***Applied Mathematical Modelling***, Vol. 35, No. 10, (2011), 4841-4851.
26. Gelareh, S. and Nickel, S., "Hub location problems in transportation networks", ***Transportation Research Part E: Logistics and Transportation Review***, Vol. 47, No. 6, (2011), 1092-1111.
27. Rodriguez-Martin, I. and Salazar-Gonzalez, J. J., "Solving a capacitated hub location problem", ***European Journal of Operational Research***, Vol. 184, No. 2, (2008), 468-479.
28. Correia, I., Nickel, S. and Saldanha-da-Gama, F., "The capacitated single-allocation hub location problem revisited: A note on a classical formulation", ***European Journal of Operational Research***, Vol. 207, No. 1, (2010), 92-96.
29. Correia, I., Nickel, S. and Saldanha-da-Gama, F., "Single-assignment hub location problems with multiple capacity

levels", ***Transportation Research Part B: Methodological***, Vol. 44, No. 8, (2010), 1047-1066.

30. Alumur, S. A., Kara, B. Y. and Karasan, O. E., "The design of single allocation incomplete hub networks", ***Transportation Research Part B: Methodological***, Vol. 43, No. 10, (2009), 936-951.

31. Yaman, H., "The hierarchical hub median problem with single assignment", ***Transportation Research Part B: Methodological***, Vol. 43, No. 6, (2009), 643-658.

32. Karimi, H. and Bashiri, M., "Hub covering location problems with different coverage types", ***Scientia Iranica***, Vol. 18, No. 6, (2011), 1571-1578.

33. Fazel Zarandi, M., Davari, S. and Haddad Sisakht, S., "The q-coverage multiple allocation hub covering problem with mandatory dispersion", ***Scientia Iranica***, Vol. 19, No. 3, (2012), 902-911.

34. Mohammadi, M., Tavakkoli-Moghaddam, R. and Rostami, R., "A multi-objective imperialist competitive algorithm for a capacitated hub covering location problem", ***International Journal of Industrial Engineering Computations***, Vol. 2, No. 3, (2011).

35. Eberhart, R. and Kennedy, J., "A new optimizer using particle swarm theory", in Micro Machine and Human Science, MHS'95., Proceedings of the Sixth International Symposium on, IEEE. (1995), 39-43.

36. Salerno, J., "Using the particle swarm optimization technique to train a recurrent neural model", in Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on, IEEE., (1997), 45-49.

37. Shi, Y., "Particle swarm optimization: Developments, applications and resources", in Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, IEEE. Vol. 1, (2001), 81-86.

38. Brooks, S. and Morgan, B., "Optimization using simulated annealing", ***The Statistician***, (1995), 241-257.

39. Wang, T.-Y., Wu, K.-B. and Liu, Y., "A simulated annealing algorithm for facility layout problems under variable demand in cellular manufacturing systems", ***Computers in Industry***, Vol. 46, No. 2, (2001), 181-188.

40. Rutenbar, R. A., "Simulated annealing algorithms: An overview", ***Circuits and Devices Magazine, IEEE***, Vol. 5, No. 1, (1989), 19-26.

41. Aarts, E. and Korst, J., "Simulated annealing and boltzmann machines", (1988).

42. Holland, J. H., "Iterative circuit computers", in Papers presented at the May 3-5, 1960, western joint IRE-AIEE-ACM computer conference, ACM. (1960), 259-265.

43. Holland, J. H., "Outline for a logical theory of adaptive systems", ***Journal of the ACM (JACM)***, Vol. 9, No. 3, (1962), 297-314.

44. Holland, J. H., Hierarchical descriptions, universal spaces and adaptive systems., DTIC Document. (1968)

45. John, H., "Adaptation in natural and artificial systems", Ann Arbor: University of Michigan Press, (1975).

46. Melanie, M., "An introduction to genetic algorithms", Cambridge, Massachusetts London, England, Fifth printing, Vol. 3, (1999).

Comparing Three Proposed Meta-heuristics to Solve a New *p*-hub Location-allocation Problem

A. Ghodratnama[a], R. Tavakkoli-Moghaddam[b], A. Baboli[c]

[a] *Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Karaj, Iran*
[b] *Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran,Iran*
[c] *INSA-Lyon, DISP Laboratory, 69621 Villeurbanne Cedex, France*

PAPER INFO

Paper history:
Received 28 March 2013
Received in revised form 02 May 2013
Accepted 16 May 2013

Keywords:
Hub Location-allocation
Vehicle Capacity
Plant Production Capacity
Simulated Annealing
Genetic Algorithm
Particle Swarm Optimization

چکیده

در این مقاله یک مدل ریاضی پیچیده ارائه می گردد. در این مدل مکان هاب‌ها مشخص و ظرفیت- شان بر اساس تسهیلات و کارخانجاتی که به آن اختصاص می یابند تعیین می گردد. بمنظور برآورده ساختن نیازهای گره های مشتری انواع مختلفی از وسائل نقلیه با ظرفیت های مختلف در نظر گرفته می شود. گره های مشتری به هاب ها تخصیص یافته و وسائط نقلیه به کارخانجات و تسهیلات مستقر در هاب ها اختصاص می یابند. بمنظور بدست آوردن جواب ها از سه الگوریتم فراابتکاری با نام های الگوریتم ژنتیک، بهینه سازی انبوه ذرات و شبیه سازی تبرید استفاده شده است. کارایی و نتایج محاسباتی این الگوریتم ها با یکدیگر نیز مقایسه گردیده است

doi: *10.5829/idosi.ije.2013.26.09c.11*