Solving a hub covering location problem under capacity constraints by a hybrid algorithm

Mehrdad Mohammadi, Reza Tavakkoli-Moghadam*, Hazhir Tolouei and Maria Yousefi
College of Engineering, University of Tehran, Tehran, Iran

Abstract. The hub location problem appears in a variety of applications including airline systems, cargo delivery systems, and telecommunication network design. When we analyze an application separately, we observe that each area has its own characteristics. Hub location problems deal with finding the location of hub facilities and with the allocation of demand nodes to these located hub facilities. This paper considers the single-allocation hub covering problem under capacity constraints, namely Capacitated Single Allocation Hub Covering Location Problem (CSAHCLP) over the complete hub networks. Furthermore, this paper presents a mixed-integer programming model that aims at finding the location of hubs and the allocation of non-hub nodes to the located hub nodes such that the travel cost between any origin–destination pair is within the given cost bound, in which hubs are considered in a limited capacity. To solve such a problem, a hybrid algorithm, based on a genetic algorithm (GA) and simulated annealing (SA), is proposed. The related results obtained by three foregoing approaches are compares. After this comparison, it is concluded that the hybrid algorithm gives better results in comparison with GA and SA.

Keywords: hub covering location problem; network design; capacity constraint; genetic algorithm; simulated annealing

Introduction

Hub-and-spoke systems are prevalent in many areas of everyday life from passenger travel through an airline’s network of airports, to postal delivery, communication, and public transportation networks. These applications have led to increased research and researcher on hub location problems, which find the best design of a hub-and-spoke network by locating hubs and assigning non-hub nodes to the hub nodes. The hub location problem includes the selection of the location of hub facilities and the allocation of the demand nodes to these located hub facilities. The hub nodes have two roles: 1) they collect demands from the non-hub nodes and transport the demands to the other hubs in the network, and 2) they distribute demands received from other hub nodes in the network to the non-hub nodes that they serve. In this paper, the hub nodes are assumed to be fully interconnected as shown in Figure 1, where the squares and circles represent hub and non-hub nodes, respectively.

As for how non-hub nodes are allocated to the located hub nodes, two fundamental types of hub networks are defined in the literature, namely single and multiple allocations. In single allocation hub networks, each non-hub node is allocated to exactly one hub, while in multiple allocation networks, a non-hub node can be allocated to more than one hub. O’Kelly (1986) first introduced the hub location problem. Then, O’Kelly (1987) presented the first recognized mathematical formulation for a hub location problem by studying airline passenger networks. His formulation is referred to the single allocation p-hub median problem that minimizes the total transportation cost of the demand flow. To reflect the economies of scale in hub-to-hub connections, he introduced a constant
discount factor, \( \alpha \in [0, 1] \), for using inter-hub connections. In addition, O’Kelly (1987) introduced a data set based on the airline passenger interactions between 25 US cities in 1970 evaluated by the Civil Aeronautics Board (CAB). The rest of the literature on the hub location problem has been primarily focused on the linearization of the quadratic model proposed by O’Kelly (1987), for example, Campbell (1996), Ernst and Krishnamoorthy (1996), O’Kelly et al. (1995), and Skorin-Kapov et al. (1996). These studies introduce different mathematical formulations and solution procedures for the minimization of the total transportation cost. Campbell (1994) introduced different hub location problems (p-hub center and covering) to the literature and considered different objective functions. In particular, the hub covering problem minimizes the total cost of establishing hub networks, so that the cost (or travel time) between any origin-destination pair is within a given bound. Campbell (1994) provided quadratic and linear formulations for both single and multiple allocation variants of the problem. However, Kara and Tansel (2003) first attempted to show computational results for the single allocation hub covering problem. They also suggested various linear formulations and proved the NP-hardness of the hub covering problem. Ernst et al. (2005) proposed a better mathematical formulation for the hub covering problem using the “radius” idea. For the single allocation incapacitated hub location problem, Labbe and Yaman (2004) derived a family of valid inequalities that generalizes the facet-defining inequalities and that can be separated in polynomial time. Ernst and Krishnamoorthy (1999) presented two new formulations for the capacitated single allocation hub location problem. Their formulations are a modified version of the previous mixed-integer formulations developed for the p-hub median problem.

In facility covering problems, demand nodes are considered to be covered if they are within a specified distance of a facility in order to serve their demands. Campbell (1994) defined three coverage criteria for hubs. The origin destination pair \((i, j)\) is covered by hubs \(k\) and \(m\) if the following assumptions happen.

- The cost from \(i\) to \(j\) via \(k\) and \(m\) does not exceed a specified value.
- The cost for each link in the path from \(i\) to \(j\) via \(k\) and \(m\) does not exceed a specified value.
- Each of the origin-hub and hub-destination links meets separate specified values.

Perhaps the most closely related study in the literature to our problem is given by Ernst and Krishnamoorthy (1999) who considered the hub location problem over a capacity constraint.

In this paper, we study the single allocation hub covering with a capacity constraint, namely CSAHCLP. This problem also includes cover constraints, which have the effect of restricting certain assignments of nodes to hubs. For example, in the facility location literature, a facility is said to be able to cover a node if the node is within a pre-specified distance from the facility. If the facility covers the node, then the node can be assigned to that facility. This implies that a feasible solution to the CSAHCLP is obtained when all the assignments of nodes to hubs satisfy the coverage requirement. In addition, we study the hub covering problem over complete hub networks. Our aim is to find the location of hubs and the allocation of non-hub nodes to the located hub nodes such that each of the origin-hub and hub-destination links meets separate specified values.

Similar to other hub location studies, we use a constant cost discount factor \( \alpha \in [0,1] \) to represent the economies of scale in hub-to-hub connections, and also use a constant cost discount factor \( \gamma \in [0,1] \) to represent the economies of scale in hub-to-node or node-to-hub connections, where \( \gamma \geq \alpha \). We do not allow direct connections between each pair of the non-hub nodes. Unlike Ernst et al. (2005), we consider the sum of the cost of all flows between each pair of nodes and the fixed cost of opening a new hub for the objective function. Furthermore, in our problem, we consider the capacity constraint for each hub. Each hub has a radius used for covering the non-hubs. Non-hub nodes can allocate to one hub when they are in the coverage area of that hub. Unlike Ernst et al. (2005), we consider
the hub covering version of the problem. We do not locate a fixed number of hub arcs and we force the hub arc network to be connected. Also, we do not impose any structure on the hub network. Our integer programming formulation involves $O(n^4)$ decision variables and $O(n^4)$ constraints. To solve realistically sized instances, we present a heuristic for this problem. In contrast to other hub location problems, constructing feasible solutions for the hub covering problem, especially with tight radius and capacity bounds, is a challenge.

Many studies in the literature have used different heuristic approaches to hub location problems; for example, the tabu search method proposed by Klincewicz (1992) and Skorin-Kapov (1994), the simulated annealing method by Ernst and Krishnamoorthy (1996), and the Lagrangean relaxation method by Pirikul and Schilling (1998) for p-hub median problems. Additional contributions include the shortest path method by Ebery et al. (2000), the genetic algorithm by Cunha and Silva (2007), a hybrid method by Chen (2007), and a dual-ascent method by Cánovas et al. (2007) for hub location problems with fixed costs. Lastly, proposals for p-hub center problems include a tabu search method and a greedy heuristic method by Pamuk and Sepil (2001) and Ernst et al. (2002), respectively. For hub location problems, the reader should note that the nearest allocation strategy (i.e., assigning a non-hub node to its nearest hub) does not necessarily give optimum solutions for the hub location problem.

In this paper, we relax the complete hub network assumption in the hub covering problem. To be able to handle real-sized problems, we propose a hybrid meta-heuristic method based on a genetic algorithm (GA) and simulated annealing (SA) for our problem. As shown in the computational studies, our proposed hybrid method behaves quite effectively. To the best of our knowledge, both the formulation and heuristics are new for the hub covering problem. The structure of this paper is as follows. We propose an integer programming formulation in Section 2. In Section 3, we present and explain our hybridization of GA with SA. Section 4 is dedicated to the computational analysis. Finally, Section 5 is devoted to concluding remarks.

Mathematical formulation

In this paper, we assume that there is a given node set $N$ with $n$ nodes that some of them (e.g., hubs) can be located. The mathematical model locates hubs, constructs the hub network, and allocates the remaining nodes in set $N$ to these hubs, such that any hub can allocate only some nodes which can cover them. Also just the limited number of nodes can be allocated to a specific hub because of considering capacity constraints for all hubs. The objective of our mathematical model is to minimize the total cost of flows between any origin-destination pair and the total cost of establishing hubs.

The parameters of the presented model are as follows. Parameter $W_{ij}$ is the flow between nodes $i$ and $j$, $C_{ij}$ is the transportation cost of one unit flow between $i$ and $j$, $F_k$ is the fixed cost of opening a hub at node $k$, and $r_k$ is the maximum collection/distribution cost between hub $k$ and nodes that are allocated to hub $k$. $Chub_k$ is the capacity of hub $k$. In addition, $\alpha \in [0,1]$ is the cost discount factor for between two hubs and $\gamma \in [0,1]$ is the same, but it is for between non-hub nodes and hubs, in which most likely $\gamma$ is bigger than $\alpha$, and it is expected to be a number close to 1.

We define the decision variables of the model as follows. $X_{ik}$ = 1 if node $i$ is allocated to hub at node $k$; 0 otherwise.

The objective function of our mathematical model is to minimize the total cost of flows between any origin-destination pair and the total cost of establishing hubs. With the previously defined parameters and decision variables, the objective function is expressed by:

$$\min \sum_{i=1}^{n} \sum_{k=1}^{n} \gamma \times C_{ij} X_{ik} O_{i} + \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha \times C_{ij} Y'_{ik} + \sum_{k=1}^{n} F_{k} X_{ik}$$

subject to

$$\sum_{i=1}^{n} X_{ik} = 1 \quad \forall i$$

$$\sum_{i=1}^{n} Y'_{ik} + \sum_{i=1}^{n} W_{ik} X_{ik} = O_{i} X_{ik} + \sum_{i=1}^{n} Y'_{ik} \quad \forall i, k$$
In the first term of the objective function (1), we sum the total cost of flows between non-hub nodes and hub nodes; in the second term, we sum the total cost of flows between each two hubs; and in the third term, we calculate the total cost of establishing hubs. Constraints (2) and (10) ensure that each node is assigned to exactly one hub. Eq. (3) is the flow balance equation (i.e., divergence equation) for commodity $i$ at node $k$ where the demand and supply at the node is determined by the allocations $X_{ik}$. Constraint (4) ensures that the allowed nominal capacity of the hub is not exceeded by the preventing cargo from entering. Constraint (5) makes sure that node $i$ can only be allocated to node $k$, if cost $C_{ik}$ between nodes $i$ and $k$ is at most the radius $r_k$ of node $k$. Constraint (6) states that a node cannot be allocated to another node unless that node is a hub node. Constraints (7) to (9) ensure that $Y_{ik}$ can be higher than zero only when nodes $k$ and $l$ are hubs. Finally, Constraint (11) ensures that the variable $Y_{ik}$ is bigger than zero, because it is the amount of flows and needs to be more than zero. To explain our mathematical model thoroughly, we present an example as depicted in Figures 2 and 3. Figure 3 represents the resulting network of a potential solution regarding the node set shown in Figure 2.

According to a typical solution $X_{11} = X_{33} = X_{77} = X_{99} = 1$, nodes 1, 3, 7 and 9 are chosen as hub nodes. Variables $X_{11} = X_{33} = X_{53} = X_{77} = X_{97} = X_{109} = 1$ represent the allocations of the non-hub nodes to the hub nodes. All other $X$ variables have zero values at this solution.

![Fig. 2. The node set $N$](image)

![Fig. 3. The resulting network of a solution](image)
Hybrid genetic algorithm and simulated annealing

It is so difficult to optimally solve most of the NP-complete problems for realistically sized instances. For the problem at hand, even finding a feasible solution is challenging. With this motivation, we decided to develop a meta-heuristic algorithm for our problem. To obtain a better solution in a short time, we hybrid the genetic algorithm (GA) with simulated annealing (SA). In this hybrid method, we inserted GA as one of the GA operators. Also, we use SA to create some of our initial solutions in the first population. This hybridization leads to shorter time and fewer gaps. In this section, according to Takano and Arai (2009), the proposed GA for the hub location problem is presented and described.

Representation of the solution

The solution of the hub location problem should represent the network by showing the location of the hubs and the allocation of the remaining nodes to a valid hub. The procedure uses integer numbers to represent a given network. The length of the vector is equal to the number of nodes in the network. Each bit or bit string expresses a node in the network, where its value explains the number of the hub or the node is allocated. Further, when the value of the string is equal to the number of the node, the node is considered a hub. For example, we can represent the following solution of the network shown in Figure 3.

\[ Ind = [1 7 3 7 3 7 1 9 9] \]

Initial population

The initialization process begins with creating the initial population. The size of the population is predefined. For the hub covering problem with single allocation under capacity constraints, the number of hubs in the network is undefined. After locating hubs in the network, the remaining spokes (i.e., nodes are not hubs) should be allocated to the located hubs. For the initial population, spokes are allocated to their nearest hub, based on cost (radius) values. Also, all hubs should cover some nodes that are not exceeded from their capacity. Designing each network that satisfies all constraints is challenging.

Function evaluation

The evaluation function is an operation to evaluate how good the network configuration of each individual is (i.e., making the comparison between different solutions possible). The evaluation function consists of calculating the objective (i.e., fitness) function value of the network represented by each individual.

Crossover

Crossover consists of exchanging information between two selected individuals, creating new offspring with a better fitness value, as in the theory of evolution. For our GA, the tournament selection method is used to select the individual for the crossover operator. In this case, one-point crossover is chosen. Due to the individual’s structure, every new offspring created by crossover should undergo a filtering process in order to guarantee that the individual has a valid structure at the end of the process.

Simulated annealing

In this section, we propose an efficient simulated algorithm (SA) to improve each solution. Each operator has an especial percent of involving in improving the solutions. Following, we introduce some of SA algorithm’s characters and parameters. Finally, the pseudo code of the proposed SA is illustrated. Some of the important parameters are as follows:
One solution that is generated randomly (i.e., number of hubs, position of hubs and connection of spokes to hubs are random).

- **Phub**: Number of selected hubs in each iteration.
- **aa**: The neighbor solution which exchange two positions in a.
- **c**: Cost of solution related to a.
- **cc**: Cost of solution related to aa.
- **T_0**: Initial temperature that an idea is to choose $T_0$, so that several cost increasing transition are accepted at the beginning.
- **L**: Number of iteration in each temperature.
- **T_f**: Final temperature (i.e., when the temperature receives to $T_f$, the procedure will be stopped).
- **ith**: Number of iteration for determining number and position of hubs.
- **vv**: Reduction factor.
- **c_1**: Assigning cost after complete ASA, where ASA is number of iterations after determining number of hubs and hubs position.
- **c_2**: The best assigning cost obtained from SA algorithm.
- **a_11, h_12**: Parameters for the number of assignment exchange and the number of hub exchanges, respectively.

In Figure 4, we present the pseudo code for our proposed SA.

```plaintext
1. Determine L, ith, v, vv, T_0.
2. Select phub from n nodes randomly and determine hubs position randomly and assign spokes to hubs (initial solution).
3. Calculate cost of solution a (c).
4. Set answer_a1 = a and c1 = c.
5. Obtain a neighbor solution with solution a by exchange two position in a (start ASA).
6. Determine cost of solution aa (cc).
7. If cc ≤ c, then set c = cc, answer_a1 = aa, a = aa else go to Step 8.
8. (a). Generate a random number r as 0 ≤ r ≤ 1.
    (b). Determine acceptance probability ap = exp (-c1/cc)/T.
    (c). If apr ≥ 1 set a = aa else retain the current solution a, and reject the move.
9. Set a_11 = a_11 + 1; if a_11 ≤ L, go to step 5.
10. If T ≥ T_f then set T = vT, go to step 5 else go to Step 11.
11. If c_1 ≤ c_2 then set c_2 = c_1, answer_a2 = answer_a1 and go to Step 12; else, go to Step 12.
12. Set h_12 = h_12 + 1 and if h_12 ≥ ith go to Step 13; else go to Step 13 else T_0 = vvT_0 and go to Step 2.
13. Set best_a = answer_a2, OFV = c_2 and show them.
```

**Fig. 4.** Pseudo code for the proposed SA

**Mutation**

In our algorithm, mutation consists of creating the new solution. It likes the population production as mentioned above.

**Computational results**

To test our proposed hybrid algorithm, we randomly generate our data for achieving feasible solutions. Table 1 shows the related results, in which we compare both exact and heuristic solutions. According to the hub location literature, the genetic algorithm is an efficient algorithm for solving the location problem. Then for testing the efficiency of our proposed hybrid algorithm, we compare the computational results obtained by this hybrid algorithm...
with GA and SA separately as illustrated in Table 1. For exact solutions in terms of the objective function value (OFV), we use the Lingo software, as a well-known optimization solver. For comparison, we consider a number of problems with different nodes that they are 10, 11, 12, 13, 14, 15, 20, 25, 30, 35, 40, 45, 50, 60 and 70. For a larger size, Lingo takes many hours (say, more than 20 hours). Therefore for large-size problems, we just use our proposed meta-heuristic algorithms to solve the model.

Table 1 shows that our proposed hybrid algorithm takes fewer minutes than a simple GA and SA. The related gap is less than gaps of GA and SA. In 10, 11 and 15 nodes, our hybrid algorithm does not have any gap. However in 12, 13, 14 and 20 nodes, we have gaps of 0.7, 0.36, 0.87 and 0.4, respectively. To solve large-sized problems, it is better to use heuristic or meta-heuristic algorithms. It is worth to note that our proposed hybrid algorithm has a fewer gaps compared with GA and SA. Thus, it is reasonable to use the hybrid algorithm instead of a simple GA and SA separately.

Table 1. Computational results of Lingo and the meta-heuristic algorithms for n-nodes networks.

<table>
<thead>
<tr>
<th>No. of nodes</th>
<th>No. of hubs</th>
<th>Lingo</th>
<th>Proposed method</th>
<th>GA</th>
<th>SA</th>
<th>Proposed method</th>
<th>GA</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>1425.20</td>
<td>1425.20</td>
<td>1425.20</td>
<td>0.00</td>
<td>1425.20</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>1245.65</td>
<td>1245.65</td>
<td>1245.65</td>
<td>0.00</td>
<td>1270.30</td>
<td>0.00</td>
<td>2.00</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>1398.50</td>
<td>1408.00</td>
<td>1446.40</td>
<td>0.70</td>
<td>1472.10</td>
<td>3.30</td>
<td>5.00</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>1647.20</td>
<td>1653.00</td>
<td>1678.40</td>
<td>0.36</td>
<td>1743.20</td>
<td>1.85</td>
<td>5.50</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>1933.00</td>
<td>1950.12</td>
<td>2016.00</td>
<td>0.87</td>
<td>2065.68</td>
<td>4.12</td>
<td>6.40</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>2078.64</td>
<td>2078.64</td>
<td>2124.35</td>
<td>0.00</td>
<td>2246.50</td>
<td>2.00</td>
<td>7.50</td>
</tr>
<tr>
<td>20</td>
<td>11</td>
<td>2562.00</td>
<td>2574.30</td>
<td>2658.28</td>
<td>0.40</td>
<td>2824.70</td>
<td>3.60</td>
<td>9.30</td>
</tr>
<tr>
<td>25</td>
<td>12</td>
<td>-</td>
<td>4954.67</td>
<td>5232.54</td>
<td>-</td>
<td>5573.34</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>-</td>
<td>6569.78</td>
<td>6742.42</td>
<td>-</td>
<td>6928.20</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>35</td>
<td>16</td>
<td>-</td>
<td>10713.49</td>
<td>11480.12</td>
<td>-</td>
<td>12394.65</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>18</td>
<td>-</td>
<td>16017.92</td>
<td>17314.38</td>
<td>-</td>
<td>22462.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>45</td>
<td>23</td>
<td>-</td>
<td>25634.36</td>
<td>27239.88</td>
<td>-</td>
<td>31792.16</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>50</td>
<td>27</td>
<td>-</td>
<td>35761.68</td>
<td>37351.53</td>
<td>-</td>
<td>41996.40</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>60</td>
<td>31</td>
<td>-</td>
<td>52105.52</td>
<td>56475.29</td>
<td>-</td>
<td>65782.35</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>70</td>
<td>39</td>
<td>-</td>
<td>83419.37</td>
<td>89103.98</td>
<td>-</td>
<td>98284.60</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.33</td>
<td>2.12</td>
<td>5.10</td>
</tr>
</tbody>
</table>

Conclusion

In this paper, we have studied the single allocation hub covering location problem under a capacity constraint. We have also presented an integer programming formulation of this problem. To solve realistically sized instances, we have proposed a hybrid algorithm based on a genetic algorithm (GA) and simulated annealing (SA). In contrast to other hub location problems, constructing feasible solutions for the hub covering problem, especially with capacity constraints, is challenging. We have tested our proposed hybrid algorithm with the data generated at random. Finally, we have compared the performance of our algorithm with the Lingo software for 7 small-sized problems. We have found that there is a little gap between the solutions obtained by our proposed hybrid algorithm and Lingo. It is concluded that our proposed hybrid algorithm is able to find good quality solutions in comparison with GA or SA. The average gaps of the solutions reported by Lingo with our proposed hybrid algorithm, GA and SA are 0.33, 2.13 and 5.10, respectively.
References