Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 5, N. 1, 2017.

Trabalho apresentado no CNMAC, Gramado - RS, 2016.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

A multi-start heuristic for the design of hub-and-spoke networks

Rodrigo de Carvalho[1]
Programa de Pós Graduação em Engenharia Elétrica, UFMG, Belo Horizonte, MG
Bruno N. Gomes[2]
Departamento de Computação, IFMG, Sabará, MG
Alexandre X. Martins[3]
Departamento de Engenharia de Produção, UFOP, João Monlevade, MG
Rodney R. Saldanha[4]
Departamento de Engenharia Elétrica, UFMG, Belo Horizonte, MG
Ricardo S. de Camargo[5]
Departamento de Engenharia de Produção, UFMG, Belo Horizonte, MG

Abstract. Design of Hub-and-spoke networks is an extension of classical facility location problem and it is very important due to its applications in cargo, passenger and telecommunication systems. The problem consists in determining the number and location of the hubs, besides define the allocation of non-hub nodes to the installed hubs, aiming to minimize the total costs. This problem is known to be NP-hard and it has been tackled by heuristic based approaches. In this paper it is proposed an efficient multi-start heuristic composed by a simple construction phase, a perturbation mechanism and an adaptive local search. Computational experiments using standard benchmark problems shows that the proposed approach is competitive when compared with the best heuristics in the literature.

Key words. hub-and-spoke networks, heuristics, combinatorial optimization

1 Introduction

The hub-and-spoke system has risen from industry's efforts to develop more efficient networks. In this system, a hub is a strategic center of the network, responsible for routing and redistribution of the demand flow and a spoke (non-hub node) is a demand points. In this type of network that the flow from different origins but addressed to the same destination can be combined at hub nodes before be transmitted to their destination, resulting in lower per unit transmission costs. This is called economies of scale and is the main advantage of hub-and-spoke networks.

Hub-and-spoke networks are present in many applications, such as cargo and passengers transportation and telecommunication systems [3, 9]. There are numerous variants of

[1]rodrigo@rclink.com.br
[2]bruno.nonato@ifmg.edu.br
[3]xmartins@decea.ufop.br
[4]rodney@cpdee.ufmg.br
[5]rcamargo@dep.ufmg.br

hub location problems, this paper focus on networks with the single allocation, where each non-hub node is allocated to a single hub, and there are no capacity constraints on hubs. Furthermore, the number of hubs is not known beforehand, and there are fixed costs for installation of hubs.

This problem is known in the literature as Uncapacitated Single Allocation Hub Location Problem(USAHLP). The USAHLP is know to be a NP-hard problem. Due to its complexity and practical applications, many researchers have developed heuristics to solve it. Topcuoglu et al. [16] proposed a Genetic Algorithm (GA). Cunha and Silva [6] develop an efficient combination of genetic algorithm and simulated annealing method. Chen [5] presents a hybrid heuristic based on the Simulated Annealing method, tabu list and improvement procedures. Silva and Cunha [15] implement three variants of a Multi-Start Tabu Search heuristic and a two-stage integrated Tabu Search heuristic. Gomes et al. [11] presented an efficient combination of Greedy Randomized Search Procedure and GA. More recently, AbyaziSan et al. [2] proposed a tabu search(TS) with new tabu rules and procedures to partial evaluations of objective function, this method outperforms the results of others methods on one standard data sets commonly used in the literature.

This paper proposes a multi-start heuristic that uses a randomized construction phase combined with the adaptative local search method. The proposed approach outperforms the heuristic for the problem proposed in [11] and has performance competitive when compared the best known heuristic for the problem [2].

2 Formulation

Given a set of demand nodes N which exchange flows and a set of candidates nodes to become hubs. The binary variables z_{ik} indicates whether a node $i \in N$ is allocated to the hub $k \in N$ ($z_{ik} = 1$) or not ($z_{ik} = 0$). Further, if a k node is set as a hub, $z_{kk} = 1$, or $z_{kk} = 0$ otherwise. For all pair of nodes (i, j) with $i \neq j$, w_{ij} represents the demand flow from origin to destination, which is routed through one or two installed hubs. Thus, c_{ijkm} denotes the transportation cost per unit of flow from node i to j routed via hubs k and m. This transportation cost is the composition of three cost segments: $c_{ijkm} = c_{ik} + \alpha c_{km} + c_{mj}$, where c_{ik} and c_{mj} are the transportation cost per unit of flow from location i to hub k and from hub m to node j, and αc_{km} is the transportation cost between hubs k and m considering the discount factor. The discount factor $0 \leq \alpha \leq 1$ represents the scale economies on the inter-hub connections. Moreover, let f_k be the installation cost of a hub at node k. The USAHLP can be formulated as:

$$\min \quad \sum_{k=1}^{|N|} f_k z_{kk} + \sum_{i=1}^{|N|} \sum_{\substack{j=1:\\ j\neq i}}^{|N|} \sum_{k=1}^{|N|} \sum_{\substack{m=1:\\ m\neq k}}^{|N|} (w_{ij} c_{ijkm} + w_{ji} c_{ijmk}) z_{ik} z_{km} \tag{1}$$

Subject to:

$$\sum_{k=1}^{|N|} z_{ik} = 1 \quad \forall i \in N \tag{2}$$

$$z_{ik} \leq z_{kk} \quad \forall i, k \in N : i \neq k \tag{3}$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \in N \tag{4}$$

The objective function (1) minimizes the costs of flow transportation and hub installation. The constraints in equation (2) assure that all the nodes are allocated to only one hub. The constraints in equation (3) assure that non-hub nodes are assigned only to hubs. Finally, constraints in equation (4) are the integrality constraints. Although there are other formulations for the SAHLP [8], this is chosen because of its simplicity in presentation.

3 A Multi-Start Heuristic

In multi-start heuristics a number of different initial solutions are generated and improved by means of some local search procedure. This type of heuristic has been used in combinatorial optimization by many authors to achieve diversification in the search space, because of its simple framework for solving hard optimization problems. Different approaches to this methodology can be found in [13].

On the other hand, this method focuses the search in the full space, which can harm the intensification phase. Thus, a procedure based on Iterated Local Search (ILS) method is used in order to make the intensification phase of the heuristic more efficient. The essential idea of ILS is to focus the search not on the full space of solutions but on a smaller subspace defined by the solutions that are locally optimal for a given optimization engine [12].

In this paper, the initial solutions for the problem is randomly constructed, then an adaptive local search algorithm is applied on this initial solution in order to explore the neighborhood for better solutions. The local search is adaptive due to the fact of neighborhood structures are used in accordance with their performance. Figure 1 illustrates the proposed heuristic, where B is the set of neighborhood structures, and π is the degree of perturbation applied to the incumbent solution which is used in local search procedure. $\phi(s)$ is a function that calculates the total cost of a solution s according to equation (1), σ is the threshold for update the probability p values, β is the threshold for perturb a solution s, and p_i the probability of the $i-th$ neighborhood structure used in the local search procedure. This section, details the implemented Multi-start algorithm for the USAHLP.

3.1 Construction Procedure

The number of hubs, as well as the possibility of any node becomes hub is randomly defined in the generation of the solutions. The nearest allocation heuristic [14] is used to allocate the non-hub nodes to the installed hubs.

3.2 Local Search

When tackling the USAHLP, a great variety of neighborhood search strategies can be explored. In this paper the local search procedure uses five neighborhood structures:


```
PROCEDURE MultiStart(B, π, σ, β)
BEGIN
  p_i ← 1/|B|, ∀i ∈ B
  WHILE stop criterion is not satisfied
      s ← Construction Procedure()
      s ← Local Search(s,p,π,σ,β)
      IF φ(s) < φ(s*) THEN
          s* ← s
      END IF
  END WHILE
  Return s*
END
```

Figure 1: Multi-start algorithm.

“Reallocation”, “Hub Exchange”, “Remove Hub”, “Add Hub” and “Hub Exchange with Reallocation”. The first four were explored using two different strategies known as best-improvement and first-improvement. In the first-improvement, the solutions moves to the first neighbor which cost function value is smaller than that of the current solution. In the other case, all neighbors are investigated and the current solutions is replaced by the best neighbor. The hub exchange reallocation neighborhood structure is explored only using the first-improvement strategy due to its high cardinality.

The neighborhood structures are described as follow: the **Reallocation** tries all reallocation of a non-hub node to the installed hubs, the number and the installed hubs are not changed; in the **Hub Exchange** a hub is selected to become a non-hub node and a non-hub node is chosen to become a hub, then the non-hub nodes are reallocated to the nearest hub; in the **Remove Hub** a selected hub is removed, then non-hub nodes are reallocated to the nearest hub; in the **Add Hub** a chosen non-hub node become a hub, then a non-hub nodes are reallocated to the nearest hub; in the **Hub Exchange with Reallocation** for each movement involving the **Hub Exchange** structure, it is applied a local search using the **Reallocation** movement.

The neighborhood structure is selected in adaptive manner, i.e., the probability associated with each neighborhood is adjusted according to their efficiency. The neighborhood structure that produces better solutions are more likely to be chosen. The probabilities are adjusted according to the equations (5-6). Equation (5) is used at each σ iterations, where t_i is the number of times the new solution is better than the current solution for a neighborhood i. Equation (6) only is used when the algorithm gets stuck in a minimum local, the aim it is to allow the use of all neighborhood structures even the less efficient ones. This is important in this situation because the solution is exposed to a perturbation procedure. Empirical tests have been shown that the probability of new solutions may be improved by these neighborhoods.

$$p_i = t_i / \sum_{j \in B} t_j \quad \forall i \in B \tag{5}$$

$$p_i = (p_i + 1/|B|)/2 \quad \forall i \in B \tag{6}$$

The perturbation procedure implemented in this work consists in changing the function of nodes, i.e., a hub is transformed into a non-hub node and a non-hub node becomes a hub. Then a non-hub nodes are reallocated to the nearest hub. The percentage of nodes changed is given by the parameter π.

Figure 2 illustrates the local search procedure implemented, where g is a number of iterations and the stop criterion is h iterations without improvement.

```
PROCEDURE LocalSearch(s, p, π, σ, β)
BEGIN
  g, h ← 0
  s* ← s
  WHILE stop criterion is not satisfied
      choose i ∈ B with probability p_i, i = {1, 2, ..., |B|}
      g ← g + 1
      s' ← LocalSearch(B_i,s)
      IF φ(s') < φ(s) THEN
          s ← s'
          t_i ← t_i + 1
          IF φ(s) < φ(s*) THEN
              s* ← s
              t_i ← t_i + 1
          END IF
      ELSE
          h ← h + 1
      END IF
      IF g ≥ σ THEN
          update p    (use equation (5))
          g ← 0
      END IF
      IF h ≥ β THEN
          update p    (use equation (6))
          perturbation(s*,π)
          h ← 0
      END IF
  END WHILE
  Return s*, p
END
```

Figure 2: Local search procedure.

4 Computational experiments

The performance of the Multi-start heuristic was compared to [11] on AP data sets [7], considering instances with sizes $|N|$={10,20,30,...,100,130,150,170,200}. The scale economies were set to α={0.2,0.4,0.6,0.8}. The instances are named by $APN - \alpha$, where N is the number of nodes on the network and α is the discount factor used. The instances have fixed costs for the first 50 nodes, thus, the costs were randomly generated for all nodes as in [4]. Each algorithm was run 10 times to all instances using distinct random seeds, but utilizing the same seed in each run.

All computational tests were carried out on a computer with a Intel i7 processor at 2.0 GHz and 6 GB of RAM, running the Ubuntu operating system. Further, all algorithms were implemented in C++. The generator of random numbers is the one embedded in the C++ language. The parameter values of the heuristic were set to: $\pi = 10\%$, $\sigma = 20$, and $\beta = 5$. The stopping criterion for both methods was set to 0.6 $\times|N|$ seconds, the GA and TS parameters was maintained as in original paper [2, 11].

DOI: 10.5540/03.2017.005.01.0471

The following metrics were computed: BestValue, DevMed, DevMin, #Best and %Executions. BestValue is the best solution attained among the algorithms considered for a given instance. For each method, DevMin and DevMed represents, respectively, the average minimum and the average mean of the deviation between the best solution attained by the algorithm and the BestValue of each instance. Thus, lower values for DevMed and DevMin imply in better algorithm. Further, #Best represents the number of instances in which the method return the BestValue. Finally, %Executions is the percentage os times that the method obtained the BestValue considering all running for each algorithm.

Table 1 shows the values of all computed metrics. Although these results provide a good performance indicator, they cannot be used to derive more general conclusions. Then, the performance evaluation was performed using Friedman's statistical test [10] with a significance level of 5% (95% of confidence level). It was found that there is significant difference among them, however it is not possible to state that one heuristic is the best. Multi-Start and Tabu Search heuristics outperforms GA, but there is no statistically significant difference between those two.

Table 1: GA x multi-start x Tabu Search results.

Metric	GA	Multi-start	Tabu Search
DesvMin	0.0016	0.0002	0.0001
DesvMed	0.0018	0.0004	0.0001
#Best	34	48	49
%Executions	61.7	78.4	87.5

5 Conclusions

In this work was presented a Multi-Start heuristic to the USAHLP, simpler than the others of literature . This method implements an efficient combination of intensification and diversification phases, allowing a good exploration of the search space. In the intensification phase was used an adaptive local search procedure, significantly improving its performance. The proposed approach generated competitive results for the benchmark instances.

References

[1] S. Abdinnour-Helm. A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operation Research*, 2:489–499, 1998.

[2] R. Abyazi-Sani and R. Ghanbari: An efficient tabu search for solving the uncapacitated single allocation hub location problem. Computers & Industrial Engineering **93**, 99 – 109 (2016).

[3] S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operation Research*, 190:1–21, 2008.

[4] R. Carmargo, G. Miranda, and H. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers and Operations Research*, 35(1047-1064), 2008.

[5] J. F. Chen. A hybrid heuristic for the uncapacitated single allocation hub location problem. *Omega*, 36:211–220, 2007.

[6] C. B. Cunha and M. R. Silva. A genetic algorithm for the problem of configuring a hub-and-spoke network for a ltl trucking company in brazil. *European Journal of Operation Research*, 179(3):747–758, 2007.

[7] A. T. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4(3):139–154, 1996.

[8] Ebery, J., Krishnamoorthy, M., Ernst, A., Boland, N.: The capacitated multiple allocation hub location problema: Formulations and algorithms. European Journal of Operational Research, 120:614–631,2000.

[9] R. Z. Farahani, M. Hermarfar, A. B. Arabani, and E. Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(3):1096–1108, 2013.

[10] M. Friedman. The use of ranks to avoid the assumption o f normallity in the analysus of variance. Journal of the American Statistical Association, **32**(200): 675–701, 1937.

[11] B. N. Gomes, A. X. Martins, R. S. Camargo, and J. A. Ramirez. An efficient genetic algorithm for the design of hub-and-spoke networks. *IEEE Communications Letters*, 17:293–296, 2013.

[12] H. R. Lourenco, O. C. Martin, and T. Stutzle. Iterated local search: Framework and applications. *Handbook of Metaheuristics*, 146:363–397, 2010.

[13] R. Marti. Multi-start methods. *Handbook of Metaheuristics*, 57:355–368, 2003.

[14] M. E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operation Research*, 18(4):343–356, 1987.

[15] M. R. Silva and C. B. Cunha. New simple and efficient heuristics for the uncapacitated single allocation hub location problem. *Computers & Operations Research*, 36(12):3152–4165, 2009.

[16] H. Topcuoglu, F. Corut, M. Ermis, and G. Yilmaz. Solving the uncapacitated hub location problem using genetic algorithms. *Computer & Operations Research*, 32:967–984, 2005.

DOI: 10.5540/03.2017.005.01.0471