

LEARNING TABU-SEARCH FOR THE HUB LOCATION PROBLEM

Carlos C. Brochine Jr.,

Mariá C. V. Nascimento

Instituto Tecnológico de Aeronáutica (ITA)

Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, São José dos Campos - SP, 12228-9

carlosbrochinejr@gmail.com,

mariah@ita.br

RESUMO

O problema de *hub location* é um problema que busca satisfazer a demanda de movimentação entre nós de origem e destinos, definindo *hubs* para otimizar (minimizar) o número de ligações entre tais nós. Diversas áreas utilizam o conceito de rede com *hubs*, tendo como as principais: empresas de linhas aéreas e as companhias postais. A solução exata de problemas de larga escala é computacionalmente inviável, tornando os métodos heurísticos boas ferramentas para a obtenção de soluções aproximadas. Os métodos heurísticos podem ainda ser combinados a técnicas de aprendizado de máquina, para a solução ainda mais rápida, conforme demonstrado nos resultados deste estudo. Este estudo propõe uma metaheurística híbrida com aprendizado para o problema de *hub-location*. O modelo considera múltiplas alocações, uma função objetivo baseada no problema *p-hub* medianas e os *hubs* têm capacidade ilimitada. A solução foi baseada em busca tabu com baixo custo computacional, modificada para incluir aprendizado através de um algoritmo de propagação de rótulos. Para a avaliação dos resultados, foi utilizado um banco de dados dos correios australiano. A análise comparativa da busca tabu com a versão com aprendizado mostra indicativos de redução de tempo e alocação de memória quando consideradas as instâncias maiores.

PALAVRAS CHAVE. Localização de Hubs, Busca-Tabu, Aprendizado de Máquina.

Tópicos: MH - Metaheurísticas, AM - Aprendizado de Máquina, OC- Otimização Combinatória

ABSTRACT

The hub location problem is a problem that seeks to meet the demand for movement between origin and destination nodes, defining hubs to optimize (minimize) the number of connections between such nodes. Several areas use the concept of a network with hubs, including airlines and postal companies. The exact solution of large-scale problems is computationally unfeasible, making heuristic methods good tools for obtaining approximate solutions. Heuristic methods can also be combined with machine learning techniques for even faster solutions, as demonstrated in the results of this study. This study proposes a hybrid metaheuristic with learning for the hub location problem. The model considers multiple allocations, an objective function based on the *p-hub* median problem, and the hubs have unlimited capacity. The solution was based on tabu search with low computational cost, modified to include learning through a label propagation algorithm. To evaluate the results, a real database from the Australian post office was used. The comparative analysis of tabu search with the version with learning shows indications of reduced time and memory allocation when considering larger instances.

KEYWORDS. Hub-Location. Tabu-search. Machine-Learning.

Paper topics: MH - Metaheuristics, ML - Machine Learning, CO - Combinatorial Optimization.

1. Introduction

Hub location problems (HLPs) consist in a challenging class of NP-hard optimization problems whose goal is to locate a number of hub facilities in a network. Application areas for HLPs in transportation are diverse. These include express package delivery, air freight and passenger travel, postal delivery, trucking, and rapid transit systems. Hub facilities correspond to centers in which one or more nodes interact. There exist several reviews and surveys on HLPs, each of them presenting different aspects of these problems (Basallo-Triana et al. [2021], Alumur et al. [2021], Farahani et al. [2013]).

HLPs are related to facility location problems. As a result there are some analogous formulations for several classical models. The problem studied in this paper is the p -hub median, that is closely linked to the p -median problem, for which several heuristic algorithms were developed. Among them, we highlight metaheuristics (MHs), as those proposed by [Calik et al., 2009], [Tran et al., 2017], [Lin et al., 2012] and [Meyer et al., 2009].

Among the literature heuristics for the HLP, we focus on those based on the metaheuristic tabu search (TS) [Glover, 1986]. TS is a local search-based metaheuristic that uses mechanisms to prematurely escape from local optima. It works by applying a local search to a given solution until it reaches its local optimum. Then, it allows modifying the local optimum through movements that worsen the quality of the solution in order to find solutions in regions around which a new, possibly better, local optimum can be obtained after a new local search. Additionally, TS was one of the first methods to consider memory mechanisms to prevent the search from returning to previously-visited solutions by marking some movements as “tabu” (forbidden) if they were performed within a specific short-term window.

By trying to gather information from the generated data throughout the search process, machine learning (ML) approaches can assist MHs [Karimi-Mamaghan et al., 2022]. In particular, they can guide MHs to make wiser decisions, which increases their intelligence and considerably enhances their performance in terms of solution quality, convergence rate, and robustness. For instance, ML was used to improve TS on Niroumandrad et al. [2022] and Sun et al. [2022]. In order to enhance TS by reducing its search space, Niroumandrad et al. [2022] also provided a learning method. Sun et al. [2022] introduced a tabu search using two customized evaluation functions, and a reinforcement learning mechanism to direct the search to new beneficial areas.

According to Basallo-Triana et al. [2021], despite the good performance of literature heuristics new MHs to approach mainly large scale instances are valuable contributions for the HLP. In particular, artificial intelligence-based metaheuristics to solve this type of problem has a great potential since its clustering nature could give insightful directions to heuristic solutions.

This paper proposes a TS algorithm for the HLP. The version of the used TS is simple, in order to include a learning step through the community detection algorithm, known as Label Propagation Algorithm (LPA) producing a method with a low complexity to better deal with large-scale instances. LPA [Raghavan et al., 2007] is a fast community detection algorithm. It detects communities based solely on network structure and does not require a pre-defined objective function or prior knowledge about the communities. LPA works by spreading labels across the network and forming communities based on this process. The algorithm is based on the idea that a single label can quickly become dominant in a densely connected group of nodes, but will struggle to cross a sparsely connected region. Labels will become trapped within a densely connected group of nodes, and nodes that end up with the same label when the algorithms are completed can be considered part

of the same community. One intriguing feature of LPA is the ability to assign preliminary labels to nodes in order to narrow the range of solutions generated. This means that it can be used as a semi-supervised method of community detection in which we hand-pick some initial communities to be used in the learning process. Computational experiments present the results achieved by TS without learning with the learning TS, comparing them with the exact methods presented by Wandelt et al. [2022]. The results indicate that the machine learning step improved TS and that the algorithm proposed can be useful in larger instances, since the running times of exact methods abruptly increase with the number of hubs.

The rest of the paper is organized as follows. Section 2 provides a description of HLP, followed by a mathematical formulation used in this paper. Section 3 presents the main components of the proposed learning tabu search. Section 4 shows the results obtained from both tabu search and learning tabu search. Section 5 concludes the paper and discusses future research directions.

2. Hub Location Problem

The hub location is a problem concerned with determining which nodes in a given network are designated as hubs and non-hubs. It essentially consists of locating a set of hubs and allocating a set of non-hub nodes (also known as spokes) to the located hubs for transferring the origin-destination flows through the hubs in order to minimize total transportation costs. The main idea in this type of network is that hubs are consolidating points.

In general, HLPs are NP-hard. Their main difficulty is associated with the interrelationship between two levels of the decision problem [Laporte et al., 2015]. The first level consists in selecting hub facilities. The second level is concerned with connecting origin-destination and relevant hubs. As stated by [Laporte et al., 2015], one of the major modeling challenges in HLPs is due to the fact that knowing the hub network structure is not sufficient to evaluate the objective function. It is necessary to model the path used for routing each flow to determine the flow cost.

The HLP is an extension of classical facility location problem. The classical form of the problem is known as p -HLP with quadratic criteria [O'Kelly, 1986]. A linear model is proposed by Campbell [1991], similar to the p -median formulation named p -hub median location problem, as highlighted on [Campbell, 1994].

An exhaustive review on the HLP was presented by Farahani et al. [2013], where some definitions and classifications are proposed. This review takes into account mathematical models, solution methods and applications. Regarding the solution methods, we highlight the heuristics that, despite being more massive when considering the variants that focus on multiple allocation, received a significant amount of attention to tackle single allocation, like those proposed by Guan et al. [2018]. Wandelt et al. [2022] investigated the Benders decomposition and column-and-row generation on Euclidean data for a variety of models.

2.1. Mathematical Formulation

Campbell [1991] proposed the formulation considered to define the problem. Because every non-hub node can be allocated to one or more hub nodes, this model is called multiple allocation p -HLP. The number of hubs to be located is given as an input of the problem, denoted by p . The goal of the problem is to locate a set of p hub facilities with the objective of minimizing the total flow cost for routing the flows through the hub network.

The parameters of the problem are: the number of hubs to locate, denoted by p ; the total number of nodes of the network, referred to as n ; h_{ij} , which is the amount of flow between nodes i and j ; C_{ij} , which is the unit cost of transferring from nodes i and j ; and C_{ij}^{km} , which is the unit transportation cost between nodes i and j passing through hubs k and m , defined in Equation (1).

$$C_{ij}^{km} = C_{ik} + \alpha C_{km} + C_{mj} \quad (1)$$

In Equation (1), let α be the discount factor, where $0 \leq \alpha \leq 1$. The transportation cost between hubs is smaller than others.

The problem variables are: X_j , that receives 1 when a hub facility is located at node j , and 0, otherwise; and Z_{ij}^{km} , that denotes the fraction of flow from origin i to destination j that is routed via hubs from locations k to m . The mathematical formulation is defined next.

$$\min f(Z) = \sum_i \sum_j \sum_k \sum_m C_{ij}^{km} h_{ij} Z_{ij}^{km} \quad (2)$$

subject to

$$\sum_k X_k = p \quad (3)$$

$$\sum_k \sum_m Z_{ij}^{km} = 1 \forall i, j \quad (4)$$

$$Z_{ij}^{km} \leq X_m \forall i, j, k, m \quad (5)$$

$$Z_{ij}^{km} \leq X_k \forall i, j, k, m \quad (6)$$

$$Z_{ij}^{km} \geq 0 \forall i, j, k, m \quad (7)$$

$$X_k \in 0, 1 \forall k \quad (8)$$

The objective function seeks to reduce the total transportation costs. Constraint (3) ensures that p hubs are selected. Constraints (4) assign each flow (i, j) to a pair of hubs (k, m) . Constraints (5) and (6) ensure that, if Z_{ij}^{km} is 1, then k and m must be hubs. Finally, constraints (7) and (8) define the variable domains.

3. Learning TS

This section presents the introduced TS, displaying its version with and without the learning step.

3.1. Tabu Search Algorithm

TS [Glover, 1986] is a metaheuristic widely employed in the solution of combinatorial optimization problems. It is a local-search based method that from a potential solution of a problem, searches its immediate neighbors – solutions that can be obtained to neighbor movements – as an attempt to find a better neighbor solution. Different from conventional local search methods that get stuck in suboptimal regions, TS has memory and relaxation mechanisms to avoid this phenomenon: to accept movements that provides a worse quality solution and a list of prohibitive movements, the tabu list, to prevent the search from returning to previously-visited solutions. A tabu list is a list of elements that are not allowed to join the solution for a given number of iterations as a memory mechanism of the search, unless it meets the aspiration criterion.

As the HLP formulation considered in this paper is related to the classical p -median problem Campbell [1994], we base the introduced TS on the TS metaheuristic that works well in the p -median problem [Gomes e Senne, 2006]. Loranca et al. [2014] showed that the TS algorithm developed by Rolland et al. [1997] obtains good results with reduced computational time.

3.1.1. Initial Solution

The initial solution of the proposed TS is obtained by a greedy strategy that, to an empty hub set, iteratively adds the node that provides the best partial solution value to the hub set. The strategy stops when the set reaches the cardinality p . The reference solution of the TS starts as this initial solution. Then a sequence of movements is applied to the reference solution as part of the search process. These steps are better discussed in the next.

3.1.2. Stochastic Local Search

The developed TS has two types of movements to construct a neighbor solution: ADD and DROP. On the one hand, the ADD movement consists in inserting one node not in the tabu list to the set of hubs of a given solution (first parameter is the tabu list-TL, the second is the hub set). The ADD movement chooses a node that is not in the current solution to be added to its hub set. The chosen node is the one from the available nodes that yields the best (partial or complete) solution value. The aspiration criterion of the proposed TS is the one widely used in the literature: it allows elements in the tabu list to join the solution if it produces a solution that is better than the currently-known best solution. The tabu list is, therefore, a set of hubs found (added) in recent solutions during the search process. This list is not permanent and after a given number of iterations the element is removed. The tabu list created takes into account the frequency of that node is in the hub set.

On the other, the DROP movement removes one node from the set of hubs of a given solution. The DROP procedure selects from the hub set a node that, when removed from the hub set, results in the best (partial) solution. It is important to notice that the solution obtained from these movements does not necessarily produce a viable solution once the number of hubs can be different from p . The search occurs in order to infeasible solutions to escape local optima. In this case, after a certain number of iterations without improvement, the algorithm becomes more permissive. Another strategy of this algorithm is diversification, which is implemented by randomly resetting the tabu list when a solution is stuck into a local optimum. The process is presented in Algorithm 1.

The input of Algorithm 1 is data regarding the hub allocation. The output is the hub set, the best solution value and the allocation matrix (Z). The introduced tabu-search is called p -hub tabu-search (PHTBS) and it calls in line 13 tabu search (ZTBS), described in Algorithm 2. Step 2 checks if the problem state is feasible (number of hubs is equal p). If so and the new solution is better than the best-known solution, we update the best solution. In Step 3 the strategic oscillation (*slack*) parameter is increased if there is no improvement in solution quality in the last *oscillation* $\times$ *stable* iterations. In Step 4, the tabu time is randomly reset if no improvement has been made in the last *diversification* $\times$ *stable* iterations. Step 5 causes termination of the algorithm if a feasible solution has been obtained but no improvement in the best feasible solution has been found in the last *stable* moves. The steps are repeated until a maximum number of iterations is reached. The random nature of the tabu time was inspired by [Rolland et al., 1997], that stated that random time achieved a better efficiency than the fixed and dynamic methods. The tabu list evaluation is based on Equation (9), where $Add_{time}(hub)$ is the iteration when the hub was added to the solution, k is a parameter for diversification, $Freq(hub)$ is the number of times hub joined the tabu list and $tabu_{time}$ is the number of iterations that hub remains a tabu movement.

$$iteration > Add_{time}(hub) + Freq(hub) \times k + tabu_{time} \quad (9)$$

ZTBS initially allocates each non-hub to its closest hub. Then, ZTBS attempts to create new connections to reduce the objective value. When a connection is created, the non-hub connected

Algorithm 1 PHTBS: p -hubs tabu-search

Input: : $C, h, p, stable_{iterations}, oscilation, diversification$

Output: : $bestsolution, hubs, Z$

```

1: Initialize the tabu list  $TL$  as empty;
2: Build an initial hub set, represented by  $hubs$ , through the greedy strategy;
3: Assign to  $bestsolution$  the solution value of  $hubs$ ;
4:  $slack \leftarrow 0$ ;                                ▷ oscillation parameter;
5:  $slpp \leftarrow 0$ ;                                ▷ counter to increase Slack;
6:  $tbpp \leftarrow 0$ ;                                ▷ counter to increase tabu time
7: while maximum number of iterations has not been reached do
8:   if  $|hubs| \leq (p + slack)$  then                ▷ Step 1: Choose if movement is ADD or DROP
9:      $ADD(TL, hubs)$ 
10:  else if  $|hubs| \geq (p + slack)$  then
11:     $DROP(hubs)$ 
12:  end if
13:   $Z \leftarrow ZTBS(hubs, nonhubs, C, h)$ 
14:  if  $|hubs| = p$  AND  $f(Z) \leq bestsolution$  then    ▷ Step 2: Evaluate Solution
15:     $bestsolution \leftarrow f(Z)$ ;
16:     $slack \leftarrow 0$ ;
17:     $improve \leftarrow iteration$ ;
18:    Randomly select an integer value between 1 to  $p$  and assign it to  $tabu_{time}$ ;
19:  else
20:     $slpp \leftarrow slpp + 1$ ;                        ▷ counter to increase Slack
21:     $tbpp \leftarrow tbpp + 1$ ;
22:  end if
23:  if  $slpp \geq stable_{iterations} \times oscilation$  then ▷ Step 3: If no changes for long time improve
    oscillation
24:     $slack \leftarrow slack + 1$ ;
25:     $slpp \leftarrow 0$ ;
26:  end if
27:  if  $tbpp \geq stable_{iterations} \times diversification$  then ▷ Step 4: If no changes for long time
    improve diversification
28:    Randomly select an integer value between  $p$  to  $p \times 2$  and assign it to  $tabu_{time}$ ;
29:     $tbpp \leftarrow 0$ ;
30:  end if
31:  if  $(iteration - improve) = stable_{iterations}$  then    ▷ Step 5: Stop if solution is stable
32:     $iteration \leftarrow max_{iterations}$ ;
33:  end if
34:   $Update(TL)$ 
35: end while

```

is added to the tabu list. The process is presented in Algorithm 2.

Algorithm 2 ZTBS: Z_{ij}^{km} tabu-search

Input: : $hubs, nonhubs, C, h$

Output: : Z

```

1: for  $el \in nonhubs$  do                                ▷ Allocation of each non-hub to the closest hub
2:    $closest[el] \leftarrow \arg \min_{k \in nonhub} C_{el,k};$ 
3: end for
4:  $\forall i, j, k, m \ Z_{ij}^{km} = 1$ , if  $closest[i] = k$  and  $closest[j] = m$ , and  $Z_{ij}^{km} = 0$  otherwise;
5:  $TL \leftarrow \emptyset;$ 
6: while the number of iterations is lower than  $iter_{max}$  do
7:   Randomly select  $node$  from  $nonhub \setminus TL$ ;
8:   Find the closest element from  $hub \setminus c[node]$  and add it to  $c[node]$ ;
9:   Update  $Z$  according to the new node allocation since it provides a better solution and add it
      to  $TL$ ;
10: end while

```

3.2. Label Propagation Algorithm

A modified version of the label propagation algorithm proposed by Raghavan et al. [2007] was employed to optimize the tabu search presented in the previous section. The modified version was based on the one introduced by Zhur e Ghahramanirh [2002], where a semi-supervised clustering algorithm propagates labels to estimate the class probability of unlabeled nodes.

The main idea when using this clustering algorithm was to propagate labels to nodes not assigned as hubs. Therefore, we considered the hubs as cluster and propagated this classification to non-hub nodes. Therefore, the tabu search can investigate the path of best probabilities. This algorithm is described in Algorithm 3.

Algorithm 3 LPA:label propagation algorithm

Input: : $hubs, C$

Output: : Y

```

1:  $\sigma \leftarrow 2/3 \times median(C);$ 
2:  $w_{ij} \leftarrow exp(-C_{ij}^2/\sigma^2), \forall i, j;$                                 ▷ Weighted Euclidean Distance
3:  $Y \leftarrow labels;$                                                     ▷  $Y$  is a label node matrix
4:  $\forall i, j, T_{ij} \leftarrow w_{ij} / \sum_j w_{ij}$                                 ▷  $T_{ij}$  is the probability to jump from node  $j$  to  $i$ 
5: while  $error \leq tolerance$  do
6:    $Y_{old} \leftarrow Y;$ 
7:    $Y \leftarrow T \times Y;$                                                     ▷ Step 1: propagate labels
8:    $Y_{ij} \leftarrow Y_{ij} / \sum_j Y_{ij};$                                     ▷ Step 2: row normalize  $Y$ 
9:   if  $i \in hubs$  then                                                    ▷ Step 3: recover original labels
10:     $Y(i) = Y_{old}(i);$ 
11:   end if
12:    $error \leftarrow ||Y - Y_{old}||$ 
13: end while

```

The input of Algorithm 3 are the hub set and the matrix costs C , used to calculate the distance between two nodes and create a higher probability to closer nodes. Y is a matrix that

receives 1 in the diagonal whose indexes are hubs, and 0 for others. T is a matrix that represents the probabilities of node i to get label j . The rows of matrix Y that refer to hubs, referred to as $Y(i)$ where $i \in hubs$, cannot change. Therefore they return to the initial state ($Y_{old}(i)$). This is done recursively until the probabilities no longer change.

This label propagation is used to achieve a matrix of probabilities that is used to assign the non-hub nodes to the most probable hubs. Algorithm 4 presents the procedure of the learning version of ZTBS.

Algorithm 4 ZTBS-LPA: Z_{ij}^{km} tabu-search with LPA

Input: : $hubs, nonhubs, C, h$

Output: : Z

```

1: for  $el \in nonhubs$  do                                ▷ Allocation of each non-hub to the closest hub
2:    $closest[el] \leftarrow \arg \min_{k \in nonhub} C_{el,k}$ ;
3: end for
4:  $\forall i, j, k, m$   $Z_{ij}^{km} = 1$ , if  $closest[i] = k$  and  $closest[j] = m$ , and  $Z_{ij}^{km} = 0$  otherwise;
5:  $TL \leftarrow \emptyset$ ;
6:  $Y \leftarrow LPA(hubs, C)$ ;
7: while the number of iterations is lower than  $iter_{max}$  do
8:   Randomly select  $node$  from  $nonhub \setminus TL$ ;
9:   Find the element from  $i \in hub \setminus c[node]$  with the highest  $Y_{node,i}$  and add it to  $c[node]$ ;
10:  Update  $Z$  according to the new node allocation since it provides a better solution and add  $i$ 
    to  $TL$ ;
11: end while

```

4. Computational Experiments

Computational experiments were carried out to attest the efficiency of learning tabu search (PHTBS-LPA) over the same method without learning and (PHTBS) by reporting the gaps to the exact solutions and running times. The exact solutions provided by Wandelt et al. [2022] were used as reference for the result analysis. The exact methods considered in this experiment are Benders (BD1) and CPLEX (CPn4). We report the average results of 3 independent runs.

4.1. Parameter setting

The parameter setting relied on the values presented by [Gomes e Senne, 2006], presented in Table 1.

Parameter	Value
$max_{iterations}$	100
$stable_{iterations}$	30
$oscillation$	2.5
$diversification$	1.6

Table 1: Parameters set according the literature.

Table 2 reports the values of the parameters set after preliminary experiments.

Parameter	Value
k (Equation (9))	14
$iter_{max}$	100
$ TL $	$0.5 \times n$

Table 2: Parameters set according the preliminary tests.

4.2. Instances

A well-known dataset is used as a case study to evaluate the performance of the proposed algorithm. The AP (Australia Post) data set includes 200 postcode districts in Australia, along with their locations and pairwise travel demands, and was presented by Ernst e Krishnamoorthy [1996]. A total of 64 instances was used in the experiments.

4.3. Results

Table 1 shows the results of the experiment per instance. The first column indicates the id of the instances, followed by their number of nodes n , number of hubs p and the discount factor α values. The average gaps of the heuristic solutions PHTBS-LPA and PHTBS to the optimal solutions and computational times of the heuristics and exact methods are reported in the following columns.

It is possible to notice that PHTBS-LPA and PHTBS achieved average gaps of 5% and 6%, respectively. As the higher values and dispersion were reduced, the learning tabu-search produced better results in 67% of the cases.

Regarding the solution times, PHTBS-LPA was faster in all cases. Moreover, as expected, the exact methods do not scale well.

One issue with the used formulation, as highlighted by Farahani et al. [2013], is the memory usage. Memory consumption during testing was not relevant (it was always below 100 MB).

5. Final Remarks and Future Work Directions

In this paper, we introduced a heuristic for solving the p -Hub Median location Problem. The heuristic is based on a tabu search proposed by Rolland et al. [1997] for the travelling salesman problem. As computational effort and solution quality are crucial considerations in the design of any heuristic, we considered a clustering algorithm to introduce a refined version for the problem under investigation.

It is possible to observe how machine learning can improve the proposed algorithm. Computational experiments indicate that the tabu search version considering learning in contrast to the same algorithm without learning improved either in objective function values or the computational times. The learning tabu search was used to solve 64 test problems ranging in size from 10 to 25 nodes. We compared it results with exact solutions obtained in Wandelt et al. [2022]. The learning tabu search was only capable of achieving faster times in 10% of tested instances. The clustering strategy in the tabu search allowed an optimized path to the algorithm. It reduced the computational time and achieved near-optimal solutions.

There is an indication that the learning tabu search algorithm may be useful for larger problems. We infer that because in solving the largest tested instances the learning tabu search showed a less abrupt increas in computational time is regarding the exact methods. Another evidence is that memory usage does not grow significantly with the problem size.

id	Problem			gap		Runtime			
	n	p	α	PHTBS-LPA	PHTBS	Heuristic		Exact Methods	
						PHTBS	PHTBS-LPA	BD1	CPn4
1	10	2	0.2	0.01	0.00	0.57	4.32	0.77	0.57
2	10	2	0.3	0.01	0.01	0.47	3.58	0.61	0.57
3	10	2	0.4	0.02	0.02	0.46	2.87	0.50	0.62
4	10	2	0.5	0.02	0.03	0.31	3.33	0.50	0.43
5	10	2	0.6	0.03	0.04	0.47	3.61	0.47	0.41
6	10	2	0.7	0.04	0.04	0.43	3.57	0.51	0.47
7	10	2	0.8	0.05	0.05	0.41	3.62	0.48	0.56
8	10	2	0.9	0.06	0.07	0.35	4.62	0.21	0.56
9	15	2	0.2	0.01	0.01	2.23	11.55	0.27	1.47
10	15	2	0.3	0.01	0.01	2.22	14.92	0.17	1.77
11	15	2	0.4	0.02	0.02	2.36	19.09	0.31	1.80
12	15	2	0.5	0.02	0.03	2.41	14.29	0.15	1.55
13	15	2	0.6	0.03	0.03	2.32	11.51	0.14	1.99
14	15	2	0.7	0.04	0.04	3.96	14.88	0.21	1.66
15	15	2	0.8	0.04	0.05	2.35	15.57	0.18	1.62
16	15	2	0.9	0.04	0.05	2.15	13.86	0.21	2.09
17	15	4	0.2	0.03	0.02	4.59	46.47	11.46	2.33
18	15	4	0.3	0.03	0.03	7.33	46.13	7.82	1.89
19	15	4	0.4	0.07	0.05	6.98	32.37	5.99	2.02
20	15	4	0.5	0.05	0.07	9.67	54.31	5.72	1.66
21	15	4	0.6	0.06	0.08	9.15	63.72	4.73	1.37
22	15	4	0.7	0.07	0.10	7.12	51.19	3.87	1.79
23	15	4	0.8	0.08	0.12	6.38	50.70	3.04	1.81
24	15	4	0.9	0.12	0.15	3.94	51.35	2.75	1.81
25	20	2	0.2	0.07	0.04	7.64	26.94	0.36	12.44
26	20	2	0.3	0.05	0.05	5.18	41.85	0.33	13.09
27	20	2	0.4	0.05	0.05	4.87	38.00	0.32	13.85
28	20	2	0.5	0.08	0.06	9.16	29.73	0.32	13.74
29	20	2	0.6	0.06	0.09	8.96	27.08	0.32	12.96
30	20	2	0.7	0.09	0.07	5.46	30.52	0.32	11.80
31	20	2	0.8	0.08	0.08	9.60	49.75	0.17	15.18
32	20	2	0.9	0.06	0.07	6.93	36.25	0.27	11.00
33	20	4	0.2	0.06	0.06	24.18	120.88	13.99	13.04
34	20	4	0.3	0.07	0.06	21.61	100.41	14.83	12.63
35	20	4	0.4	0.06	0.06	36.56	118.05	20.84	11.39
36	20	4	0.5	0.06	0.07	25.26	121.21	21.65	10.55
37	20	4	0.6	0.07	0.09	20.40	78.45	19.69	8.57
38	20	4	0.7	0.07	0.09	22.57	78.88	17.36	13.50
39	20	4	0.8	0.08	0.11	22.29	141.06	15.94	15.03
40	20	4	0.9	0.11	0.13	21.38	127.81	12.60	13.85
41	25	2	0.2	0.01	0.01	16.61	87.13	0.61	57.71
42	25	2	0.3	0.02	0.02	22.34	105.46	0.53	65.81
43	25	2	0.4	0.02	0.02	18.64	83.75	0.47	36.31
44	25	2	0.5	0.03	0.03	22.30	100.18	0.39	45.09
45	25	2	0.6	0.04	0.04	21.67	100.97	0.36	68.72
46	25	2	0.7	0.04	0.05	26.32	65.24	0.34	43.25
47	25	2	0.8	0.04	0.05	24.15	54.82	0.47	66.42
48	25	2	0.9	0.04	0.06	26.93	51.88	0.32	48.48
49	25	4	0.2	0.01	0.01	60.98	262.64	25.44	53.15
50	25	4	0.3	0.02	0.02	62.12	304.59	28.78	48.14
51	25	4	0.4	0.04	0.04	62.73	243.30	25.64	69.35
52	25	4	0.5	0.06	0.06	37.91	275.39	14.20	65.26
53	25	4	0.6	0.07	0.08	76.26	291.19	16.16	69.12
54	25	4	0.7	0.07	0.09	56.86	250.61	13.48	52.12
55	25	4	0.8	0.09	0.11	51.63	263.81	12.64	71.97
56	25	4	0.9	0.10	0.15	61.42	200.89	10.29	70.70
57	25	8	0.2	0.01	0.02	105.39	972.10	10654.32	59.02
58	25	8	0.3	0.02	0.02	163.35	959.06	12049.26	74.56
59	25	8	0.4	0.03	0.03	101.85	576.95	15460.13	53.73
60	25	8	0.5	0.04	0.04	170.25	618.26	21925.94	63.97
61	25	8	0.6	0.05	0.06	157.47	901.31	25637.96	84.71
62	25	8	0.7	0.07	0.07	201.98	613.55	26720.38	82.17
63	25	8	0.8	0.10	0.09	231.13	1007.88	21865.02	880.36
64	25	8	0.9	0.12	0.12	174.40	948.88	12188.61	126.42
avg				0.05	0.06	35.71	172.63	2294.41	40.50

Table 3: Results table.

References

- Alumur, S. A., Campbell, J. F., Contreras, I., Kara, B. Y., Marianov, V., e O’Kelly, M. E. (2021). Perspectives on modeling hub location problems. *European Journal of Operational Research*, 291(1):1–17. ISSN 0377-2217. URL <https://www.sciencedirect.com/science/article/pii/S0377221720308432>.
- Basallo-Triana, M. J., Vidal-Holguín, C. J., e Bravo-Bastidas, J. J. (2021). Planning and design of intermodal hub networks: A literature review. *Computers & Operations Research*, 136:105469.
- Calık, H., Alumur, S. A., Kara, B. Y., e Karasan, O. E. (2009). A tabu-search based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research*, 36(12):3088–3096.
- Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405.
- Campbell, J. (1991). Hub location problems and the p-hub median problem. *Center for Business and Industrial Studies, University of Missouri, St. Louis, MO*.
- Ernst, A. T. e Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, 4(3):139–154.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., e Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & industrial engineering*, 64(4):1096–1109.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549.
- Gomes, N. e Senne, E. (2006). Um algoritmo de busca tabu para solução de problemas de localização de p-medianas. *XXVIII Encontro Nacional de Engenharia de Produção*.
- Guan, J., Lin, G., e Feng, H.-B. (2018). A learning-based probabilistic tabu search for the uncapacitated single allocation hub location problem. *Computers & operations research*, 98:1–12.
- Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., e Talbi, E.-G. (2022). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, 296(2):393–422.
- Laporte, G., Nickel, S., e Saldanha da Gama, F., editors (2015). *Location Science*. Springer International Publishing AG.
- Lin, C.-C., Lin, J.-Y., e Chen, Y.-C. (2012). The capacitated p-hub median problem with integral constraints: An application to a chinese air cargo network. *Applied Mathematical Modelling*, 36(6):2777–2787. ISSN 0307-904X. URL <https://www.sciencedirect.com/science/article/pii/S0307904X11006226>.
- Loranca, M. B. B., Ruiz-Vanoye, J. A., Velazquez, R. G., Flores, M. A. R., e Analco, M. E. (2014). P-median: A performance analysis. *Res. Comput. Sci.*, 88:19–30.

- Meyer, T., Ernst, A., e Krishnamoorthy, M. (2009). A 2-phase algorithm for solving the single allocation p-hub center problem. *Computers Operations Research*, 36(12):3143–3151.
- Niroumandrad, N., Lahrichi, N., e Lodi, A. (2022). Learning tabu search algorithms: A scheduling application.
- O’kelly, M. E. (1986). The location of interacting hub facilities. *Transportation science*, 20(2): 92–106.
- Raghavan, U. N., Albert, R., e Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106.
- Rolland, E., Schilling, D. A., e Current, J. R. (1997). An efficient tabu search procedure for the p-median problem. *European Journal of Operational Research*, 96(2):329–342.
- Sun, Z., Benlic, U., Li, M., e Wu, Q. (2022). Reinforcement learning based tabu search for the minimum load coloring problem. *Computers & Operations Research*, 143:105745.
- Tran, T. H., O’Hanley, J. R., e Scaparra, M. P. (2017). Reliable hub network design: Formulation and solution techniques. *Transportation Science*, 51(1):358–375.
- Wandelt, S., Dai, W., Zhang, J., e Sun, X. (2022). Toward a reference experimental benchmark for solving hub location problems. *Transportation Science*, 56(2):543–564.
- Zhur, X. e Ghahramanirh, Z. (2002). Learning from labeled and unlabeled data with label propagation.