Transportation Research Part E 108 (2017) 141–159

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

Single allocation p-hub median location and routing problem with simultaneous pick-up and delivery

Zuhal Kartal[a,*], Servet Hasgul[b], Andreas T. Ernst[c]

[a] Department of Industrial Engineering, Anadolu University, 26555 Eskisehir, Turkey
[b] Department of Industrial Engineering, Eskisehir Osmangazi University, 26480 Eskisehir, Turkey
[c] School of Mathematical Sciences, Monash University, Clayton, 3800 Victoria, Australia

ARTICLE INFO

Keywords:
Location
Hub location and routing
Hub location
Simulated annealing
Ant colony system

ABSTRACT

We introduce the single allocation *p*-hub median location and routing problem with simultaneous pick-up and delivery based on observations from real life hub networks. The aim of our problem is to minimize the cost of transferring the flow between hubs and routing the flow in the network. We propose several mixed integer programming formulations and two heuristic approaches based on multi start simulated annealing and ant colony system to solve these problems. Extensive results demonstrate that using our methods good solutions can be found despite the computationally challenging nature of the problem.

1. Introduction

In various many-to-many distribution networks, including airlines, cargo/postal delivery services, and telecommunication networks, hub facilities serve as sorting, transshipment and consolidation points. The aim of a hub location problem is to find the location of the hubs and allocation of the non-hub nodes to the hubs in order to minimize costs. It is generally assumed that there are economies of scale due to the accumulation of large amounts of flow at the hub facilities. This economy of scale is approximated using a constant discount rate for the travel costs between every hub pair.

Hub location problems can be categorized in terms of the objective functions of the mathematical models. In the literature, hub location problems with total transportation cost objectives (median), min-max type objectives (center), and covering type objectives have been well studied Alumur et al. (2012). The interested reader is referred to the reviews by Campbell (1994), Alumur and Kara (2008), and Campbell and O'Kelly (2012).

Although the basic structure of the flow movement is the same in hub location problems, each application network has its different requirements and specifications. In this study, we focus on the operational characteristics of a leading cargo company in Turkey. Through interviews with this company, it was determined that the cargo firms' main objective is to satisfy customers. To guarantee customer satisfaction in this sector, companies have started to pay more attention to service levels. Recently, cargo companies have started to offer "delivery within the same day in the city," "next day delivery," or "delivery within 24 h" guarantees to their customers in Turkey. "Next day delivery" or "delivery within 24 h" has been a target for a long time for the cargo companies operating in Turkey. For this purpose, cargo companies have investigated the costs and advantages of using airlines in their hub networks Alumur et al. (2012). Apart from that, the "delivery within the same day in the city" service offer is a fairly recent application for cargo companies. In particular, one of the leading cargo companies has started offering this service to its customers. Therefore, this company has started to investigate a cost-minimizing hub location and routing network structure in Istanbul. We note

* Corresponding author.
E-mail addresses: zkartal@eskisehir.edu.tr (Z. Kartal), shasgul@ogu.edu.tr (S. Hasgul), Andreas.Ernst@monash.edu.au (A.T. Ernst).

http://dx.doi.org/10.1016/j.tre.2017.10.004
Received 10 January 2017; Received in revised form 22 September 2017; Accepted 4 October 2017
Available online 01 November 2017

here that this service is usually preferred by companies that dispatch products to their dealers within the same day, such as banks that forward documents across their branch offices.

Motivated by the “delivery within the same day in the city” application, we extend the uncapacitated single allocation *p*-hub median problem to include vehicle routing with simultaneous pick-up and delivery. Although we explain the operating characteristics of the motivating example in the following, our aim with this paper is not to focus on this particular application and it will not consider all of the idiosyncrasies of this company. Instead we are interested in extensions of the existing model and formulations, how solution algorithms scale, and the structure of the solutions obtained when allowing vehicle routing with simultaneous pick-up and delivery.

To explain the operating structure of the service network in detail, consider that there are two types of nodes in the *same city*: branch offices (*non-hub nodes*) and operation centers (*hubs*). Initial contact with the customers is made at the branch offices, where parcels are picked-up from the customers. The branch offices also act as delivery points, where parcels are given to the consignees. Operation centers are major sorting centers where incoming parcels are collected and rerouted to their destinations. In such networks, each branch office is customarily allocated to a single operation center, mainly owing to ease of management (Serper and Alumur, 2016). In the “delivery within the same day in the city” application, the company accepts the parcels at the branch offices no later than 11 a.m. in order to distribute them by 6 p.m. on the same day. We note here that this service is available only for documents and small packages. After all the parcels are collected at the branch offices, each vehicle starts its first route from its respective operation center, travels across the branch offices while picking-up and delivering the flows at each branch office along the route, and returns back to the same operation center. The parcels that will be sent to the other branch offices start their journey from the first operation center to the final operation center. At the final operation center, each vehicle performs its second route *just* by delivering the flows at each branch office that originated from branch offices that belong to different vehicle routes, and the route ends when the vehicle returns back to the operation center. In their second route, vehicles only perform delivery because the flows that originated at each branch office were already collected in the first route.

With these considerations, the problem we consider in this paper is to design a hub location-routing network. We refer to this as the single allocation *p*-hub median location and routing problem with simultaneous pick-up and delivery, denoted by pHMLRP-SPD throughout this paper. It includes locating *p* hubs, assigning each non-hub node to exactly one hub and forming the vehicle routes. In our problem branch offices are modelled as nodes, also called demand centres, some of which may be selected as operation centers (hubs). Since a single allocation scheme is adopted, each non-hub node must be served by exactly one vehicle.

We assume that the hubs and vehicles (in the hub network and routes) are uncapacitated. Although vehicle capacity constraints can be easily incorporated into our models, because this service is only available for documents and small packages, it is assumed that the capacity of any vehicle is sufficient to transfer the traffic flows. We also assume that the number of vehicles based at each hub is fixed and limited. The fixed costs of using links and vehicles that might depend on traversed distance and driver costs, respectively, are not considered in our formulation. However, these features could easily be incorporated into our models, if required.

Traditionally in the *p*-hub median problem, it is assumed that one vehicle is available between each demand center and hub. Especially if a cargo company employs a hub location network in big cities, such an approach results in higher transportation costs as well as traffic congestion, because it forces the use of separate vehicles. By allowing vehicle routes we allow companies to determine the best network design for a fixed investment budget specified in terms of the number of hubs and vehicles to be used.

Our primary concern with this study is to propose effective models for the *p*-hub median location and routing problem based on well-known modeling ideas with three-index (Ernst and Krishnamoorthy, 1996) and four-index variables (Campbell, 1994) from the hub location literature, which can be essentially simple but give a strong basis for easy extension to other constraints; these include hub/vehicle capacity and/or time/distance constraints. Moreover, we intend this paper to contribute to the hub location and routing literature by following as closely as possible the “radial”, i.e., single/multiple allocation *p*-hub median model, to what degree ideas and solution algorithms from this extensive part of the literature can be extended to a case where the access network involves vehicle routes.

In most of the studies in the hub location and routing problem literature, the majority of the papers focus on minimizing total transportation costs (e.g. Rieck et al., 2014; Karimi and Setak, 2014). These costs generally include the total costs of establishing hubs and allocating demand centers to hubs, and the cost of flow through the network. However, the routing costs for a vehicle that starts and ends at the same hub are calculated only as a function of distance Cetiner et al. (2010) or a function of total demand plus total supply for a node Camargo et al. (2013). Even though the proposed models are realistic representations of the hub location and routing problem concerning the specific applications, none of them consider the costs as a function of distance traversed and the flow. Recognizing such a need for a real-life application, in this paper, we propose a more consistent objective function, in which the transportation costs in all parts of the network, both between hubs and between non-hub nodes, is proportional to both the distance and the amount of flow volume carried through the links.

The structure of our problem is illustrated by Fig. 1, which shows a hub location-routing solution for a network with three hubs and seven vehicles in total.

In airline companies, if separate aircraft and separate air staff are assigned to each flight, they incur high investment and operating costs. For passenger airlines, the number of legs is a significant factor as travelers prefer shorter, more direct flights. However, for star access networks traffic congestion occurs at the hub airports. Hence, many parcel delivery companies that use air transport do not connect every demand node directly to a hub (Bowen, 2012).

This problem may also appear frequently in railway and maritime transportation applications. Furthermore, similar networks are used in the military logistics sector where personnel and equipment are picked-up and delivered simultaneously at each regional center.

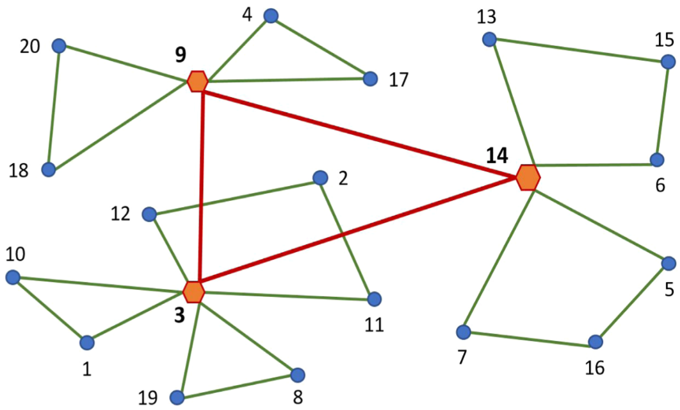


Fig. 1. Hub location-routing solution structure with three hubs and seven vehicles.

The remainder of the paper is organized as follows. In the next section, we establish the relation between our study and the existing literature. In the third section, mathematical formulations for the pHMLRP-SPD are introduced. In the fourth section, we present and explain our heuristic algorithms.

We test and compare the performance of our heuristics with the optimization solver CPLEX 12.6 on the well-known Civil Aeronautics Board (CAB) data set and on the Turkish network by using reduced instances of these two datasets. To assess how the heuristics scale to larger instances, computational results on the Turkish network of 81 nodes, on the Australia Post data set with up to 200 nodes and on the URAND data set with up to 400 nodes are also presented. The last section is devoted to concluding remarks.

2. Literature review

A hub location problem consists of determining the nodes that become hubs, i.e., nodes that consolidate and disseminate the flow in a given set of nodes with corresponding demands, and allocating the non-hub nodes to the selected hubs. Since the hub location problem was first presented by O'Kelly (1987), many researchers have extensively investigated this field. In particular, variants of hub location problems with different objective functions and additional constraints have been well studied.

Hub location problems mostly focused on the total transportation cost objective. If the aim is to locate a fixed number of hubs, the *p*-hub median problem arises with the minimization objective of the costs for routing the flow, while other hub location problems include set-up costs for the hubs in the objective function. The interested reader may refer to Farahani et al. (2013) and Contreras (2015) for recent surveys of hub location studies in the literature.

The *p*-hub median location and routing problem that we focus on is equivalent to the uncapacitated single allocation *p*-hub median problem if one vehicle is assigned between each demand center and hub. O'Kelly (1987) introduced the first mathematical formulation for the hub location problem. He proposed a quadratic integer programming formulation to minimize total transportation costs for the uncapacitated single allocation *p*-hub median problem (USApHMP). The USApHMP has been studied by many others since then (e.g. Campbell, 1994; Ernst and Krishnamoorthy, 1996; Kratica et al., 2007; Stanimirovic, 2010).

The network structure of solutions to the pHMLRP-SPD is similar to a multi-depot multiple travelling salesman problem. If an objective of minimizing fixed cost of links were to be used, then finding the optimal network structure for a fixed set of hubs would be a multi depot vehicle routing problem. However, the cost structure here is very different, as the objective is to minimize the transportation cost, which is the product of the flow volume and distance travelled. Thus, our model is closer to the traditional hub location and the network design literature than to studies of travelling salesman and vehicle routing problem variants. Further references can be found in a survey of multiple travelling salesman problem variants by Bektas (2006) and a survey of multi-depot vehicle routing problems by Montoya-Torres et al. (2015). We also refer the interested reader to Toth and Vigo (2014) for an overview of vehicle routing problems.

In standard hub location models a different, more efficient, mode of transport is assumed to operate between hubs than for the access network though more complicated intermodal networks have also been considered. For example, a capacitated intermodal hub network design problem is considered by Serper and Alumur (2016). The authors allowed alternative transportation modes and different types of vehicles in the hub networks to be designed. To solve this problem, they developed a mixed integer programming (MIP) model and a variable neighborhood search algorithm as a solution methodology. Gelareh et al. (2011) considered the simultaneous design of network and fleet deployment of a deep-sea liner service provider. The network design problem was formulated using a four-index hub location problem, which is a class of formulations known for their tightness. The problem was solved by using a primal decomposition method. A hierarchical multimodal hub location problem was studied in Alumur et al. (2012). This problem involves two types of hubs and hub links to be located, by considering ground and air transportation, while ensuring time-definite deliveries. The authors developed a mixed integer linear programming model (MILP) and proposed some variable fixing rules and valid inequalities to solve the problem. For LTL carriers, Estrada-Romeu and Robust (2015) studied the effectiveness of stopover and hub-and-spoke strategies in the long-haul routing design problem. The design problem involves some probabilistic consolidation strategies. A heuristic algorithm was presented to evaluate which consolidation strategy may reduce the costs.

Although many studies in the literature assume a direct connection between each demand center and the hub to which it is assigned, building such hub networks will obviously increase the total investment in vehicles. The relaxation of this assumption constitutes the hub location and routing problem (HLRP). In the hub location literature, since optimal allocations are affected by hub locations and optimal hub locations are affected by allocation decisions, the location and allocation problems must be considered together in designing hub networks (Alumur and Kara, 2008). In addition, when routing decisions are incorporated into the *p*-hub median problem, optimal hub locations generally differ from the *p*-hub median problem. Therefore, the optimal allocations-route structures are affected by the hub locations and the optimal hub locations are affected by the allocations-route structures.

Thus, one could justify the need to design a network that combines location-allocation considerations and vehicle routes in various applications. In cargo delivery systems, allocating separate vehicles between each demand center and hub is quite costly in terms of total investment in vehicles. Instead, if the network is designed in such a way that vehicles from a hub follow a route and visit different demand centers along their routes, then the total investment cost decreases substantially.

Basically, HLRP determines the location of the hubs, allocation of the non-hub nodes to the hubs and forms the vehicle routes so as to minimize the total routing costs of the flow in the network. For postal and parcel transportation applications, HLRP arises when nodes do not have sufficient demand to justify direct connections with the hubs Rodriguez-Martin et al. (2014) between many origins and many destinations, and this special type of HLRP is called a many-to-many hub location and routing problem (MMHLRP). In the literature, there are some MMHLRP applications with the aim of reducing mail delivery times Cetiner et al. (2010) in postal logistics and lowering the total transportation costs in parcel or mail delivery Camargo et al. (2013) and Rieck et al. (2014). MMHLRP is also closely related to location-routing problems, which were first posed by Watson-Gandy and Dohrn (1973).

The location-routing problem (LRP) is a research area within locational analysis, with the distinguishing property of paying special attention to underlying issues of vehicle routing (Nagy and Salhi, 2007). From a hierarchical viewpoint, the aim of LRP is to merge facility location and vehicle routing into a single problem where strategic location and tactical/operational routing decisions are taken simultaneously. In LRP, location decisions are affected by vehicle routes, and vehicle route decisions are affected by the location of depots (Salhi and Rand, 1989). This interdependence between the location of facilities and vehicle routing requires the combination of such decisions, which results in LRPs (Toyoglu et al., 2012). For comprehensive surveys on location and routing problems, the reader is referred to Drexl and Schneider (2015).

Even though there are some similarities between LRP and MMHLRP, such as locating facilities for the transshipment of goods and performing vehicle tours that start and end at a facility; there are also some important differences. The most important difference is that the LRP is lacking the communication between all pairs of nodes, so that there is no transfer between facilities or hubs.

The MMHLRP was first introduced by Nagy and Salhi (1998) in which the authors combined hub location with vehicle routing. In their problem variant, capacity and distance constraints are considered with fixed costs for establishing hubs and the standard median transportation cost. Two vehicles can visit one customer, one of them for pick-up and the other for delivery of the goods. They suggested a mathematical formulation and a hierarchical heuristic to solve the problem with a neighborhood search method. They also compared their proposed heuristic with a sequential heuristic. Unlike the remainder of the hub location literature, they did not allow for a discount factor between hubs. They reported only one problem instance solution that includes 249 customers and 10 terminals.

Wasner and Zapfel (2004) handled a problem in which a network of parcel services is considered. In their problem, there is a central hub and all flow of all inter-hubs is transferred via this central hub. Additionally, the vehicles carry out both pick-ups and deliveries. They decided on the locations of hubs, allocation of non-hub nodes to depots, and the vehicle routes among all of the connections. They formulated a mixed integer nonlinear programming model (MINLP) and used a two-stage local search procedure to solve this problem.

The MMHLRP studies most relevant to our problem are those that consider MMHLRP and simultaneous pick-up and delivery of the flows together. Cetiner et al. (2010) studied an MMHLRP in the Turkish postal delivery. They have the following assumptions in their problem: Hubs and vehicles are uncapacitated, multiple allocation is allowed and there is a maximum route length for each vehicle. The authors also considered the pick-up and the delivery to happen at the same time. Although they did not provide a mathematical formulation, they used (Campbell, 1996) formulation for hub location problem, and (Miller et al., 1960) formulation for multiple travelling salesman problem respectively. In order to solve the problem, an iterative two-stage solution algorithm was developed. In our problem, we consider a single allocation version of the hub location routing problem and we also provide identical mathematical formulations that vary in size and complexity.

Camargo et al. (2013) studied a hub location and vehicle routing problem that is somewhat closer to the multi-depot vehicle routing problem, in which the cost of transport is independent of the flow volume for collection and distribution. In their formulation, they combined a single allocation hub location problem formulation proposed by Skorin-Kapov et al. (1996) with a travelling salesman problem proposed by Claus (1984). They bounded the maximum time for each tour to guarantee the service quality. The objective function also includes costs for the installation of hubs, a handling cost for demand nodes that depends on the hub to which the node is allocated, but not the choice of route by which it is connected, and a fixed cost for each vehicle used. They showed how their formulation can be split into two sub-problems, and solved it using a Benders decomposition method. Unlike Camargo et al. (2013), we propose formulations that jointly consider three-index Ernst and Krishnamoorthy (1996) and four-index Campbell (1994) multi-commodity variable-based models with the travelling salesman problem.

Rodriguez-Martin et al. (2014) proposed a similar version of the single allocation HLRP. They assumed that each hub has a single vehicle, which can only visit a limited number of nodes. Unlike other studies in hub location and routing, the number of hubs to establish was given in their model. The objective function of their problem was the same as in the classic single allocation *p*-hub median problem, with the addition of vehicle routing cost. That is, the cost of flow from a non-hub node to a hub depends on the straight line distance between the node and the hub, irrespective of the vehicle route length. At the same time, the cost for the vehicles in their model depends on the distance travelled without considering flow volumes. There is exactly one vehicle for visiting the assigned demand centers on its route. The authors proposed a branch-and-cut algorithm to solve the problem. Unlike Rodriguez-Martin et al. (2014), we assume that the flows are picked-up and delivered simultaneously at each node by considering the distance traversed by vehicles and flow volumes together. We also do not put any limit on the number of non-hub nodes that a vehicle can serve. In addition, our problem allows allocating different number of vehicles to each hub.

In the literature, there are some other works on MMHLRP that are driven by an application-oriented approach. A generalized MMHLRP was studied in Rieck et al. (2014), with multiple products in the timber industry. It is assumed that there can be direct transport between pick-up and delivery locations. In their problem, there are three layers: supply points (layer 1), potential hubs (layer 2), and delivery points (layer 3). Besides this, inter-hub transports, multiple products, and direct transports between supply and delivery points are considered. The authors present an MIP model and several types of valid inequalities to strengthen the formulation. To solve larger sized problems, the authors propose a multi-start fix-and-optimize procedure and a genetic algorithm.

Lopes et al. (2016) studied a special type of the MMHLRP that arises in transportation systems, particularly in public and urban transportation of passengers. The problem involves partitioning the set of nodes of a graph into routes including exactly one hub each, and determining an extra route interconnecting all hubs. Three heuristics are proposed to solve the problem. In the first heuristic, a biased random-key genetic algorithm is presented with a local search routine, while the others include combining a multi-start scheme with a variable neighborhood heuristic.

There are many studies that apply metaheuristic solution algorithms for solving hub location problems. These include Simulated Annealing Ernst and Krishnamoorthy (1999), Tabu Search Calik et al. (2009), Variable Neighborhood Search Davari et al. (2013), Ant Colony Optimisation Meyer et al. (2009) and a Genetic Algorithm Topcuoglu et al. (2005). In addition, various hybrid methods have been tried. For hub location problems with fixed costs, for example, simulated annealing has been hybridized with both genetic algorithms Cunha and Silva (2007) and tabu search Chen (2007).

As can be seen above, there are a variety of papers in the literature dealing with at least some aspects of combined hub location and routing. To the best of our knowledge, there is no previous study on the pHMLRP-SPD in the literature. Therefore, to solve this problem, we contribute three different mixed integer programming formulations to the literature. The first two formulations have $\mathcal{O}(n^3)$ variables, while the last formulation has $\mathcal{O}(n^4)$ variables, where n is the number of nodes. Moreover, in order to solve realistically sized instances, we present a *multi-start simulated annealing* (MSA) heuristic and an *ant colony system* (ACS) heuristic. The MSA solution is used as an initial upper bound for the MIP formulations.

3. *p*-hub median location and routing problem and mathematical formulations

In this section, we introduce three MIP models for pHMLRP-SPD, which are identical in terms of the objective functions but differ in size and complexity. While developing these formulations, our aim is to show that pHMLRP-SPD can be modelled in various formulation sizes by using well-known three- and four-index variables that were proposed for the *p*-hub median problem from the hub location literature. We also aim to decrease the complexity of the proposed models by incorporating symmetry-breaking constraints. This additional property obviously improves the computation times of the models. Besides, we show that when routing considerations are included in the *p*-hub median problem, formulations can be developed which can easily accommodate additional costs and constraints.

We start by presenting the problem definition and parameters in the first subsection. Afterwards, we introduce the proposed MILP formulations. First, we present a formulation with $\mathcal{O}(n^3)$ variables in the second subsection. This formulation is based on a three-index multi-commodity flow formulation introduced for the *p*-hub median problem by Ernst and Krishnamoorthy (1996). This model is denoted by *Model1*. We remark here that the mentioned formulation is known as the best in terms of CPU time requirements for solving the *p*-hub median problem in the literature Ernst and Krishnamoorthy (1996). In the third subsection, we present a formulation that incorporates symmetry-breaking constraints into *Model1*, which we call *Model2*. In the hub location problem literature, larger formulations often lead to tighter lower bounds, although at a considerable increase on the required CPU times Ernst and Krishnamoorthy (1999). To determine if we have a similar experience in our problem as well, we present a third formulation (*Model 3*) with $\mathcal{O}(n^4)$ variables in addition to the symmetry breaking constraints.

3.1. Problem definition

The aim of pHMLRP-SPD is to determine the locations of a predetermined number of hubs, while allocating demand centers to the hubs and designing vehicle routes to visit all the demand centers with simultaneous pick-up of **supply** and delivery of **demand** volume in such a way that the total transportation cost is minimized.

Hub location and routing problems are solved on a (usually complete) undirected graph with a demand node set N and a potential hub set H, where $H = N$ for the rest of the paper. Each pair of nodes is connected by a link i–j with a routing cost (distance, time, and\or a similar measure). Let a route mean a trip traversed by a vehicle starting and ending at one of the hub nodes and in this trip, each vehicle visits at least one non-hub node. We assume that a limited fleet of homogeneous vehicles V to operate on the routes is given. These vehicles are also capable of visiting multiple stops on each route. Each route starts from a hub node where an empty vehicle is initially loaded and continues throughout the non-hub nodes until it reaches the initial hub after forming a tour. Although the vehicles that operate between hub nodes in the model are not taken into account, for this purpose it is assumed that between each pair of hubs, a faster (cheaper) vehicle is available. This may be a truck that has no capacity restriction. These faster vehicles are not permitted to make any additional stops along their routes. There are also no capacity restrictions on hubs and vehicles. We also note that since the flow variables we used in the mathematical models act like sub-tour elimination constraints, we do not need any additional variables or constraints to prevent sub-tours.

In order to present the mathematical formulation for the pHMLRP-SPD, we first need to define some parameters:

W_{ij} = demand between nodes $i \in N$ and $j \in N$.
c_{ij} = transportation cost of a unit flow between nodes $i \in N$ and $j \in N$.
α = hub-to-hub transportation cost discount factor.
q = the number of vehicles that are assigned to each hub.
V = the set of all vehicles, with $|V| = p \times q$.

The number of available vehicles could be a function of the hub node, but for simplicity of presentation, it is assumed that all hubs have the same fleet size. Therefore, in our formulation the total fleet size $|V|$ equals $p \times q$, which is the multiplication of the number of hubs by vehicle number per hub. Also, let $O_i = \sum_j W_{ij}$ be the total amount of flow originating from node i.

3.2. Three-index multi-commodity-flow-based formulation

We first formulate pHMLRP-SPD based on a multi-commodity flow formulation introduced for the p-hub median problem in the Ernst and Krishnamoorthy (1996) study, where the flow originating from a particular demand center is represented by a single commodity.

The variable f^i_{jk} is the amount of flow that originates at the initial node $i \in N$ and travels, passing through hub $j \in H$ to hub $k \in H$. We also define y^i_{jk} to be the flow that originates at node i and travels from node j to node k, at least one of j and k must be a demand node and the other may be a hub.

Our formulation uses the following binary variables:

z_i^v is one if hub node i is allocated to vehicle v and zero otherwise.
h_i is one if node $i \in H$ is a hub and zero otherwise.
x_{ij}^v is one if vehicle v traverses arc (i,j) and zero otherwise.

We refer to the first mathematical programming formulation of our problem as *Model1*. *Model1* is given as follows:
Model1:

$$\min \sum_i \sum_j \sum_k d_{jk}\, y^i_{jk} + \sum_i \sum_j \sum_k \alpha\, d_{jk}\, f^i_{jk} \tag{1}$$

$$\text{s. t.} \sum_{k \neq i \in N} (y^i_{ik} + f^i_{ik}) = \mathcal{O}_i, \quad \forall\ i \in N \tag{2}$$

$$\sum_{k \in N} (y^i_{kj} + f^i_{kj}) = \sum_{k \in N} (y^i_{jk} + f^i_{jk}) + W_{ij}, \quad \forall\ i, j \neq i \in N \tag{3}$$

$$y^i_{kj} \leqslant \mathcal{O}_i \sum_{v \in V} x^v_{kj}, \quad \forall\ i,k \neq j,j \in N \tag{4}$$

$$\sum_{j \neq i \in N} x^v_{ij} - \sum_{j \neq i \in N} x^v_{ji} = 0, \quad \forall\ i \in N, v \in V \tag{5}$$

$$\sum_{i \in N} h_i = p, \tag{6}$$

$$\sum_{j \in N} (f^i_{kj} + f^i_{jk}) = \mathcal{O}_i h_k, \quad \forall\ i,k \in N \tag{7}$$

$$z_i^v \leqslant \sum_{j \neq i \in N} x^v_{ji}, \quad \forall\ i \in N, v \in V \tag{8}$$

$$\sum_{v \in V} z_i^v = q\, h_i, \quad \forall\ i \in N \tag{9}$$

$$\sum_{i \in N} z_i^v = 1, \quad \forall\ v \in V \tag{10}$$

$$\sum_{v \in V} \sum_{j \neq i \in N} x^v_{ij} = 1 + (q-1) \times h_i, \quad \forall\ i \in N \tag{11}$$

$$z_i^v \in \{0,1\}, \quad \forall \ i \in N, v \in V \tag{12}$$

$$h_i \in \{0,1\}, \quad \forall \ i \in N \tag{13}$$

$$x_{ij}^v \in \{0,1\}, \quad \forall \ i,j \in N, v \in V \tag{14}$$

$$y_{kj}^i, f_{kj}^i \geqslant 0, \quad \forall \ i,j,k \in N \tag{15}$$

The objective function (1) minimizes total transportation costs on the network. The total transportation cost is associated only with the operational cost. This cost includes the sum of two parts: the first part calculates the transportation costs for the routes and the second part calculates the costs between hubs. It would be trivial to include fixed costs for hubs ($F_k h_k$), fixed running costs for vehicles ($R_{ij} x_{ij}^v$) or double costs for a separate morning and evening run of the vehicle routes. The form presented and used here corresponds most closely to the standard single allocation p-hub median problem.

Constraint (2) ensures that the total flow out any node i is equal to the supply of node i. Constraint (3) provides for the conservation of i-flow at any other node j with demand W_{ij}. By constraint (4), the i-flow between nodes k and j can only occur if there is a vehicle allocated to this connection ($\mathcal{O}_i$ acts as a "Big M" constant that can be tightened slightly in practice or replaced by a hub capacity if required). Constraint (5) is known as a degree constraint. Constraint (5) guarantees that every node has as many vehicles arriving as leaving. Constraint (6) requires exactly p hubs to be selected. Constraint (7) prevents inter-hub flow passing through nodes that are not hubs. If node i is assigned to a vehicle v, the arc between node i and node j is served by the same vehicle; this is enforced by constraint (8).

If node i is a hub, the total number of assigned vehicles for this hub is q, which is ensured by constraint (9). Since we use the single allocation case, constraint (10) makes sure that each node is only assigned to one vehicle. Constraint (11) states that if node i is a hub, the number of arcs that leave from node i is equal to the number of assigned vehicles. Conversely, if node i is not a hub, then each i–j non-hub link is visited only once by a vehicle. Constraints (12)–(14) force the decision variables to be binary. Finally, by constraint (15), the y and f flow variables are restricted to be non-negative.

The mathematical model for pHMLRP-SPD is a mixed integer program that has ($n^2v+nv + n$) binary variables, ($2n^3$) positive variables and ($n^3+2n^2+2nv + 3n + v$) constraints. Thus, there are $\mathcal{O}(n^3)$ variables and constraints in total.

3.3. Three-index multi-commodity-flow-based formulation with symmetry-breaking constraints

Model1 includes a lot of symmetry since there are identical vehicles. In the solution of the problem, any (identical) vehicle can be assigned to any route such that renumbering the vehicles leads to equivalent (symmetric) solutions. Thus, any solution to pHMLRP-SPD has $|V|!$ ($|V| = p \times q$: total number of vehicles) equivalent solutions permuting the vehicle indices. This symmetry leads to excessive duplication in the branch-and-bound tree when the alternative solutions have to be investigated. Thus, this approach can dramatically slow down the effectiveness of the model. To address this, we develop new mathematical models that reduce symmetry.

For example, suppose we have a problem in which there are 10 nodes, three hubs and two vehicles for each hub. The solution of the problem might be for vehicles 1, 2, 3, 4, 5, and 6 (v_1, v_2, v_3, v_4, v_5, v_6); v_1: 6–4–6; v_2: 6–5–6; v_3: 3–9–10–3; v_4: 1–8–1; v_5: 3–7–3 and v_6: 1–2–1, respectively (where nodes 1, 3 and 6 are hubs). The basic idea behind our next formulation, which we refer to as *Model2*, is to ensure that if node i is assigned to vehicle $\overline{v}$, then vehicle $\overline{v}-1$ goes to at least one of the nodes $1,\ldots,i-1$. Additionally, in this new formulation, only one vehicle index is used per hub instead of per route; therefore, there are much fewer vehicle-related variables in this formulation.

Thus, this new formulation has a fleet size $\overline{V}$ equal to p: the total number of hubs. By using this formulation, our solution has the following unique routing structure for $\overline{v}_1$: 1–2–1 and 1–8–1; for $\overline{v}_2$: 3–7–3 and 3–10–9–3; for $\overline{v}_3$: 6–4–6 and 6–5–6, respectively. This change significantly reduces both the problem size and symmetry (as we now have only three vehicle indices).

We define this new per hub vehicle set as $\overline{V} = \{1,\ldots,p\}$, with $p \leqslant |V|$ (and usually much smaller). This new mathematical model requires additional decision variables $hz_i^{\overline{v}}$. These are constrained so that $hz_i^{\overline{v}} = h_i \times z_i^{\overline{v}}$ in the optimal solution; that is, $hz_i^{\overline{v}}$ is one iff vehicle $\overline{v}$ is allocated to hub node i.

The mathematical formulation of *Model2* can now be written as:

Model2:

$$\mathbf{min}(1)$$

$$\mathbf{s.\,t.} \quad (2)\text{–}(7),(10),(12)\text{–}(15)$$

$$\sum_{j\neq i\in N} x_{ij}^{\overline{v}} = z_i^{\overline{v}} + (q-1)\times hz_i^{\overline{v}}, \quad \forall \ i\in N, \overline{v} \in \overline{V} \tag{16}$$

$$\sum_{\overline{v}\in\overline{V}} hz_i^{\overline{v}} = h_i, \quad \forall \ i \in N \tag{17}$$

$$\sum_{i\in N} hz_i^{\overline{v}} = 1, \quad \forall \ \overline{v} \in \overline{V} \tag{18}$$

$$hz_i^{\overline{v}} \leqslant z_i^{\overline{v}}, \quad \forall \ i\in N, \overline{v} \in \overline{V} \tag{19}$$

$$hz_i^{\bar{v}} \leqslant \sum_{j \leqslant i \in N} hz_j^{\bar{v}-1}, \quad \forall \ i \in N, \bar{v} \in \bar{V} \backslash \{1\} \tag{20}$$

$$hz_i^{\bar{v}} \geqslant 0, \quad \forall \ i \in N, \bar{v} \in \bar{V} \tag{21}$$

Constraint (16) is like Constraint (11), which shows if node i is a hub, the number of arcs that leave from node i matches the number of assigned vehicles. Otherwise i is a non-hub node that is visited only once by a vehicle. Constraints (17)–(19) ensure that the $hz_i^{\bar{v}}$ variables take on the correct values for any solution of the h_i and $z_i^{\bar{v}}$ variables.

Constraint (20) breaks the symmetry (where the vehicles are identical). If vehicle $\bar{v}$ starts its route from hub node i, then vehicle $\bar{v}-1$ must according to this arbitrary choice begin its route from a hub node with an index smaller than i. Finally, constraint (21) defines the decision variables $hz_i^{\bar{v}}$ as positive variables – they will be binary for any integer feasible solution of the h_i and $z_i^{\bar{v}}$ variables. *Model2* is a mixed integer program that has $(n^2p + np + n)$ binary variables, $(2n^3 + np)$ positive variables and $(n^3 + n^2 + 5np + 2n + 2p)$ constraints. Thus, there are $\mathcal{O}(n^3)$ variables and constraints in total.

3.4. Four-index multi-commodity-flow-based formulation with symmetry-breaking constraints

It is known in the literature that larger formulations can lead to tighter lower bounds, which may reduce the size of the branch-and-bound tree at the expense of more computational time required to solve each LP relaxation. In an attempt to obtain such tighter LP bounds, we consider a larger, four-indexed formulation.

Here we use variables f_{kl}^{ij}, defined as the fraction of total flow from node i to node j via hubs located at nodes k and l in that order (see Campbell, 1994). To restrict the flow only along routes served by a vehicle, we need additional variables y_{kl}^{ij}. These define the fraction of total flow from origin i to destination j transported from k to l (k,l may be hubs or non-hub nodes). To reduce the number of non-zero coefficients we define a two-index binary variable xh_{ij} that keeps the route information in this model. This also allows branching on the route decision without committing to a vehicle. The new variables are introduced as the sum of the $x_{ij}^{\bar{v}}$. Thus, xh_{ij} takes on the value 1 if arc (i,j) is traversed by any vehicle and 0 otherwise.

The formulation of this new mathematical programming, *Model3*, is as follows:

Model3:

$$\min \sum_i \sum_j \sum_k \sum_l d_{kl} W_{ij} y_{kl}^{ij} + \sum_i \sum_j \sum_k \sum_l \alpha \ d_{kl} W_{ij} f_{kl}^{ij} \tag{22}$$

$$\textbf{s. t.} \qquad (6),(10),(12)-(14),(16)-(21)$$

$$f_{kl}^{ij} = f_{lk}^{ji}, \quad \forall \ i \neq j \neq k \neq l \in N \tag{23}$$

$$\sum_{k \neq i \in N} (y_{ik}^{ij} + f_{ik}^{ij}) = 1, \quad \forall \ i, j \neq i \in N \tag{24}$$

$$\sum_{l \neq k \in N, l \neq i \in N} (y_{lk}^{ij} + f_{lk}^{ij}) - \sum_{l \neq k \in N, l \neq i \in N} (y_{kl}^{ij} + f_{kl}^{ij}) = 0, \quad \forall \ i \neq j, k \neq i \in N \tag{25}$$

$$y_{kl}^{ij} \leqslant xh_{kl}, \quad \forall \ i \neq j \neq k \neq l \in N \tag{26}$$

$$\sum_{l \in N} (f_{kl}^{ij} + f_{lk}^{ij}) \leqslant h_k, \quad \forall \ i,j,k \in N \tag{27}$$

$$\sum_{j \neq i \in N} xh_{ij} = 1 + (q-1) \times h_i, \quad \forall \ i \in N \tag{28}$$

$$\sum_{j \neq i \in N} xh_{ji} = 1 + (q-1) \times h_i, \quad \forall \ i \in N \tag{29}$$

$$\sum_{\bar{v} \in \bar{V}} x_{ij}^{\bar{v}} = xh_{ij}, \quad \forall \ i,j \neq i \in N \tag{30}$$

$$xh_{ij} \in \{0,1\}, \quad \forall \ i,j \in N \tag{31}$$

$$x_{kl}^{\bar{v}} \geqslant 0, \quad \forall \ k,l, \in N, \bar{v} \in \bar{V} \tag{32}$$

$$f_{kl}^{ij} \geqslant 0, \quad \forall \ i,j,k,l, \in N \tag{33}$$

$$y_{kl}^{ij} \geqslant 0, \quad \forall \ i,j,k,l, \in N \tag{34}$$

The objective function (22) minimizes total transportation cost. The first term in the objective function calculates the transportation cost for the routes. The second term calculates the transportation cost between hubs. Constraint (23) reduces the size of the formulation based on symmetry of travel costs. Constraint (24) guarantees that total flow from node i to j is transferred. Constraint (25) is the flow conservation constraint. Constraint (26) defines the amount of vehicle flow originated at node i destined to j using the link $k{-}l$. Constraint (27) ensures that the total flow is transferred via hub k and hub l. Constraints (28) and (29) are equivalent to (11). Constraint (30) guarantees that there is at most one vehicle on any $i{-}j$ link. Constraint (31) defines the decision variable xh_{ij} as binary. The rest of the constraints of the model (32)–(34) represent non-negativity requirements for the flow variables. We remark here that since xh_{ij} keeps the path followed by vehicles, $x_{ij}^{\overline{y}}$ can be defined as a positive variable in the formulation.

Model3 is a mixed integer program that has $(n^2 + np + n)$ binary variables, $(n^2p + 2n^4 + np)$ positive variables and $(2n^4 + 2n^3 + 2n^2 + 3np + 3n + 2p)$ constraints. There are $\mathcal{O}(n^4)$ variables and constraints in total. Thus, for greater values of n, the model involves a large number of variables and constraints. This makes it extremely hard to solve realistic sized instances optimally. Therefore, we developed two metaheuristics algorithms for our problem, which are based on MSA and ACS. In the next two sections we present our MSA and ACS algorithms.

4. A multi-start simulated annealing heuristic

It is difficult to solve NP-hard problems to optimality for realistically sized instances. For the problems at hand, even finding an optimum solution for small sized problems is challenging as can be seen in the computational results below. With this motivation, we decided to develop heuristic algorithms for our problems. We used ideas from the simulated annealing heuristic methodology. Simulated Annealing (SA) is a well-known metaheuristic algorithm that generally produces satisfactory solutions in reasonable amount of time.

We propose a multi-start simulated annealing heuristic that combines the advantages of simulated annealing and multi-start hill climbing strategies for solving the pHMLRP-SPD (Yu and Lin, 2014). The strategy consists in performing several searches simultaneously on the entire search space, starting from different initial solutions, and selecting at the end the best among the solutions obtained by all searches.

First, Kirkpatrick et al. (1983) developed SA algorithm to solve combinatorial optimization problems. Beginning from that date, SA has been accepted broadly to solve many large scale problems. Basically, SA is a stochastic search algorithm which contains a sequence of iterations. At each iteration, the current solution is changed to a new solution by applying a neighborhood move randomly. If the new solution is better than the current one, it replaces the current solution and the search process continues from this new current solution. There is a small probability that a worse solution is accepted as the new current solution.

The proposed heuristic consists of two phases: repeatedly constructing an initial solution, and improving the solution using a SA procedure with six neighborhood search mechanisms. The following subsections discuss the components of the proposed simulated annealing algorithm, including generation of initial solution, neighborhood structure and the complete multi-start simulated annealing procedure.

4.1. Initial solution generation and neighbourhood search

To work effectively, our method needs to be able to generate a diverse set of initial solutions and then traverse the solution space using efficient neighbourhood moves. For the initial solution generation we use the same randomized construction heuristic used in the ACS method described below (see Section 5). However, we do not use any “learning” mechanism while generating the initial solution. We also use the same uniform pheromone matrices with equal values ($\tau_j^s = 1$) throughout the MSA method.

The proposed MSA heuristic uses six local neighborhood search mechanisms: Intra-Move, Inter-Move, Intra-Swap, Inter-Swap, Edge-opt and Hub-Move. Each new neighborhood solution is generated by one of these six mechanisms.

Intra-Move: Remove a randomly chosen non-hub node from the route of a vehicle, and insert into the same route, next to a randomly chosen node.
Inter-Move: Insert a randomly chosen non-hub node into a different and randomly chosen route, next to a randomly chosen node.
Intra-Swap: Swap the position of two randomly chosen non-hub nodes within the same route.
Inter-Swap: Swap two randomly chosen non-hub nodes between two randomly chosen different routes.
Edge-opt: Swap two consecutive non-hub nodes between two randomly chosen different routes.
Hub-Move: Change the location of the hub within a (randomly chosen) route to a different (randomly chosen) node in the same route.

We remark here that moves are only applied when there are sufficient routes or nodes per route to allow a non-trivial change to the solution. In order to obtain a new solution, the neighbors of the current solution are re-generated using one of the above move operators.

For any SA implementation to be effective, it is necessary that all of the neighborhood moves can be performed very quickly Ernst and Krishnamoorthy (1999). In our algorithm, performing any swap move, edge-opt and hub-move requires $\mathcal{O}(n^2)$ operations. Our main contribution here is to improve a procedure that computes the differences between the objective function values of neighboring solutions for a move operator. Whenever a node is moved within or between routes attached to the same hub, the inter-hub transfer flows remain unchanged. If the routes belong to different hubs, the inter-hub flows can also be changed by updating transfer flows between these two hubs. Hence to compute the new cost only requires updating the collection/distribution costs along the routes associated with old and new location of the node to be moved. This can be done by traversing each route once, subtracting off the costs associated with distribution and collection at the old location and adding on the costs associated with the new location. Hence this is an $\mathcal{O}(n)$ operations - and usually much faster than that, since the typical route length is roughly $\frac{n}{pq} \ll n$.

4.2. The MSA algorithm

The MSA heuristic first generates initial solutions by using different random number seeds as start points. Let *SPsize* denotes the number of starting points in the MSA. For each iteration, the minimum objective function solution is chosen as the iteration best amongst *SPsize* solutions. I_{iter} denotes the total number of iterations that the search proceeds at a particular temperature. T_0 represents the initial temperature.

The computation of the initial temperature, is based on White (1984), who proposed that the starting temperature should be the one at which the expected cost is just within one standard deviation of the mean cost. In practice, this amounts to computing the standard deviation of the cost for a fixed number of transitions, all of which are accepted. The initial temperature is then set to this value (Ernst and Krishnamoorthy, 1999). K is the Boltzmann constant used in the probability function to determine whether to accept a worse solution or not. Finally, θ denotes the coefficient controlling the cooling schedule. We also use a reheat mechanism (Osman, 1993). To avoid local minima, reheating is applied in situations where $Max_{non-improving}$ consecutive temperature decreases do not give the desired decrease change in the cost function.

At the beginning of the proposed MSA heuristic, a random initial solution S is generated. Let $N(S)$ denote the set of solutions neighboring S, which is generated by one of the six local search operations. S_{best} records the best solution found as the algorithm progresses and the best objective function value obtained so far is set to be $obj(S_{best})$. At each iteration, the next solution S' is generated from $N(S)$ and the objective function value is set to be $obj(S')$. Let Δ denote the difference between $obj(S)$ and $obj(S')$, that is $\Delta = obj(S')\text{-}obj(S)$. The probability of replacing S with S', given that $\Delta > 0$, is $exp(\text{-}\Delta/KT)$. This is accomplished by generating a random number $r \in (0,1)$ and replacing the solution S with S' if $r < exp(-\Delta/KT)$. Meanwhile, if $\Delta \leqslant 0$, the probability of replacing S with S' is 1.0. Whenever the algorithm finds a new best solution, this solution is set to be S_{best} and the MSA is continued. The current temperature T is decreased after running I_{iter} iterations, according to the formula $T = \theta \times T$. If a better solution is found, a deterministic local search procedure which sequentially performs hub-move local search, move local search and swap local search is used to improve the current best solution. After performing all these steps for each starting point, the fittest solution amongst the *SPsize* solutions is chosen.

In our local search strategy, we use the best improvement approach in order to get better results from our algorithm. In this approach, we test all of the possibilities for each operator and retain the one that produces the most improved cost. For instance, while applying move local search, firstly the Intra-Move operator is considered; a non-hub node in a route is reassigned to all different positions in the same route. Then, we implement Inter-Move operator; the same particular node is reassigned to all different positions in other vehicle routes, in an attempt to achieve the maximum reduction in the objective function after evaluating all possible moves.

5. Ant colony system algorithm

Ant Colony Optimization (ACO) is a population-based metaheuristic that can be used to find good solutions to difficult optimization problems Dorigo (2007). ACO is designed based on the foraging behavior of real ants searching for food sources in a colony. Ants initially wander in different directions. Once a food source is found, ants lay down scented chemicals called pheromones, on their trace back to the colony. As more ants travel on a path, the pheromone amount on that path will rise and this helps other ants of the colony to select this path with high probability. On the other hand, the pheromone evaporates over time and thus, loses its attractiveness. The longer the path between the colony and the nest, the more likely it is that the pheromone evaporates. The pheromone level varies depending on the distance and the quality of the food source. Therefore, shorter paths obviously keep higher pheromone levels. This natural behavior is simulated in ACO to find good solutions for a combinatorial optimization problem by using artificial ants. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph using a stochastic construction process Dorigo (2007).

Our proposed algorithm is based on the ACS of Dorigo and Gambardella (1997). ACS was first introduced to improve the performance of ACO in solving the travelling salesman problem. An elitist strategy is used to update pheromone trails, where only the global best ant is allowed to produce pheromone during the iterations of the algorithm. In addition, ACS uses local pheromone updating and a pseudo-random proportional rule in selecting the next node to visit in an attempt to increase the importance of exploitation versus exploration.

The ACS algorithm that we propose for solving pHMLRP-SPD relies on a hierarchical solution structure. Hierarchical approaches might give better solutions for LRP-type problems because the problem at hand is basically a facility location problem that also considers the routing decision. Thus, a hierarchy arises between locating and routing problems. The interested reader may refer to Nagy and Salhi (2007) for an excellent review on the methods of LRP.

According to the classification presented in Nagy and Salhi (2007), our proposed algorithm falls under the category of hierarchical methods with an implementation of ACS. In our algorithm, we apply a solution method in which the location of hubs is the main problem and vehicle routing (including allocation of non-hub nodes to hubs) is the subordinate one. Therefore, we have a three-stage ACS to build a solution with different transition rules. This is very similar to the idea of using a hierarchical structure with three different transition rules for solving an LRP, for example in Ting and Chen (2013). Their algorithm has location selection, customer assignment, and vehicle routing stages. Ting and Chen (2013) also use multiple (different) ant colonies for the aforementioned stages of the algorithm. By contrast we use the *same* ant colony for each stage. Indeed, we carried out many experiments with different colonies for different stages; however, we observed that the solution quality deteriorated with those settings in our problem.

An individual solution is represented by an “ant” in ACS methods. In our algorithm, we are repeatedly *choosing hub nodes*, *allocating non-hub nodes to the hubs* and *designing vehicle routes*. The construction rules for these stages are explained in the next three subsections. The coordination between the three stages is achieved through local and global pheromone updating rules. We give the details regarding both local and global pheromone updates in the fourth subsection. The local search procedure is explained in the last subsection.

5.1. Choosing hub nodes

In this phase of the algorithm, the hubs are chosen based on the location selection rule. A node j is chosen as a hub location in each iteration s, according to Eqs. (35) and (36).

$$j = \begin{cases} \arg\max\limits_{j \in B_s^h} [\tau_j^s], q_0 \leqslant q. \\ J, \text{otherwise.} \end{cases} \tag{35}$$

$$J\text{: } P_j^h = \frac{\tau_j^s}{\sum\limits_{j \in B_s^h} [\tau_j^s]}, \tag{36}$$

Here B_s^h is the set of nodes which are not selected yet at iteration s by ant h. τ_j^s shows the pheromone level of node j and q is uniformly distributed within [0,1]. With a pre-specified frequency q_0, the chosen hub is the location that gives the largest pheromone value among the node set of candidate hub locations that are not yet selected ($q_0 \leqslant q$). Otherwise, it is taken from the probability distribution induced in Eq. (36), where P_j^h is the probability of an ant h choosing a node j as a hub.

5.2. Allocating non-hub nodes to the hub nodes

Each non-hub node i is assigned to a selected hub j based on the allocation rule in this phase. In Eqs. (37) and (38) we give the construction rule for allocation of non-hub node i to hub j.

$$i = \begin{cases} \arg\max\limits_{j \in \zeta_s^h} [\tau_{ij}^s \eta_{ij}], q_0' \leqslant q'. \\ i, \text{otherwise.} \end{cases} \tag{37}$$

$$I\text{: } P_{ij}^h = \frac{\tau_{ij}^s \eta_{ij}}{\sum\limits_{j \in \zeta_s^h} [\tau_{ij}^s \eta_{ij}]}, \text{ where } \eta_{ij} = \frac{\mathscr{O}_i + \mathscr{D}_i}{d_{ij}} \tag{38}$$

where ζ_s^h shows the selected hubs of ant h at the s_{th} iteration, τ_{ij}^s is defined as the pheromone level between non-hub node i and hub j at iteration s, and η_{ij} is the heuristic information. η_{ij} is defined as the ratio of the total flow (supply plus demand) of the non-hub node i to the distance from this node i to hub node j. In (38), $\mathscr{O}_i$ denotes the total supply ($\sum_j Wij$) and $\mathscr{D}_i$ denotes the total demand ($\sum_j Wji$) of node i. q' is a random variable drawn from a uniform distribution U[0,1] and q_0' ($0 \leqslant q_0' \leqslant 1$) is a parameter that determines the relative importance of exploitation (37) versus exploration (38).

When $q_0' \leqslant q'$, non-hub node i is allocated to hub j according to the largest pheromone value ($\arg\max_{j \in \zeta_s^h}[\tau_{ij}^s \eta_{ij}]$). Otherwise, the non-hub node i is assigned to hub j using the probability distribution induced in (38), where P_{ij}^h is the probability of an ant h allocating a non-hub node i to hub j.

5.3. Designing vehicle routes

In this phase of the construction algorithm, the vehicles for each hub are routed by considering a multiple travelling salesman problem (MTSP) type ACS approach. In the MTSP algorithm, vehicle routes are formed according to the allocation and hub selection decisions. The number of ants for the MTSP algorithm is the same as in the previous two phases. While forming the routes, an ant h chooses node k after visiting node i in a route by the following state transition rule:

$$k = \begin{cases} \arg\max\limits_{k \in \omega_{is}^h} [\tau_{ik}^s \eta_{ik}], q_0'' \leqslant q''. \\ K, \text{otherwise.} \end{cases} \tag{39}$$

$$K\text{: } P_{ik}^h = \frac{\tau_{ik}^s \eta_{ik}}{\sum_{k \in \omega_{is}^h} [\tau_{ik}^s \eta_{ik}]}, \text{ where } \eta_{ik} = \frac{1}{d_{ik}} \tag{40}$$

Here ω_{is}^h is the set of nodes which are not visited yet by ant h at iteration s at node i, τ_{ik}^s is the pheromone level of arc i–k at iteration s, η_{ik} is the heuristic information and defined as the inverse proportion of a distance which is calculated as the distance from node i to another node k. q'' is a random variable drawn from a uniform distribution U[0,1] and q_0'' ($0 \leqslant q_0'' \leq 1$) is a parameter that determines the relative importance of exploitation (39) versus exploration (40). When $q_0'' \geqslant q''$, the expression of visiting the next node k from the node i is calculated induced in (40), where P_{ik}^h is the probability of an ant h visiting the arc k–j in that order. Alternatively, when $q_0'' \leqslant q''$, ($\arg\max_{k \in \omega_{is}^h} [\tau_{ik}^s \eta_{ik}]$) formula is used for determining the next node to be visited (39).

In summary, in the first stage of the algorithm, the ant colony determines which node is to be selected as a hub. The ant colony decides what nodes are allocated to the hubs in the second stage, and in the last stage, it solves the MTSP sub-problem. While solving the MTSP, the algorithm forces any ant to return to the initial hub where the route begins and thus, the vehicle route is assumed to be completed. This process continues until each node is visited. Until the stopping criterion has met, these three stages of ACS are applied iteratively.

5.4. Pheromone updating rules

After each ant constructs a solution, the amount of pheromone level associated with the solution are decreased, so that the chosen hub location, assignment and routing decisions become less attractive. The aim here is to avoid repeatedly choosing the same nodes as hubs, assigning the same non-hub nodes to the same hubs and forming the same route structures. Global updating increases the pheromone trail level corresponding to the best solution to direct the search in the neighbourhood of the solution found. In the proposed ACS algorithm the local pheromone updating rules are:

$$\tau_j^{s+1} = (1-\rho)\tau_j^s + \rho/n \quad \text{if } j \in F_h \tag{41}$$

$$\tau_{ij}^{s+1} = (1-\rho')\tau_{ij}^s + \rho'/n \quad \text{if } (i,j) \in F_h \tag{42}$$

$$\tau_{ik}^{s+1} = (1-\rho'')\tau_{ik}^s + \rho''/n \quad \text{if } (i,k) \in F_h \tag{43}$$

Here F_h is the solution constructed by ant h; ρ, ρ' and ρ'', $(0 \leqslant \rho,\rho',\rho'' \leqslant 1)$ are the pheromone evaporation parameters. Locating of the hubs, allocation and routing phases are connected through the global pheromone updates. The global best solution is denoted by F_{gb} and the iteration best solution is denoted by F_{ib}. The pheromone for determining the hubs, assignment decisions for non-hub nodes to the hubs and routing phases are reinforced using the rules:

$$\tau_j^{s+1} = (1-\rho)\tau_j^s + \rho\Delta\tau_j^s \quad \text{where} \quad \Delta\tau_j^s = Q \times (G_b \backslash I_b) \quad \text{if } j \in F_{ib} \text{ or if } j \in F_{gb} \tag{44}$$

$$\tau_{ij}^{s+1} = (1-\rho')\tau_{ij}^s + \rho'\Delta\tau_{ij}^s \quad \text{where} \quad \Delta\tau_{ij}^s = Q \times (G_b \backslash I_b) \quad \text{if } (i,j) \in F_{ib} \text{ or if } (i,j) \in F_{gb} \tag{45}$$

$$\tau_{ik}^{s+1} = (1-\rho'')\tau_{ik}^s + \rho''\Delta\tau_{ik}^s \quad \text{where} \quad \Delta\tau_{ik}^s = Q \times (G_b \backslash I_b) \quad \text{if } (i,k) \in F_{ib} \text{ or if } (i,k) \in F_{gb} \tag{46}$$

Here G_b and I_b are the objective of F_{gb} and F_{ib} respectively, and Q is a scaling parameter. The update increases the probability of producing solutions similar to the global and iteration best. Our aim is to increase the pheromone values corresponding to the global best and iteration best solution at each iteration.

5.5. Local search

Every time all ants have generated their routes, three local search procedures are performed to improve the best solution found by all the ants in the current iteration in the same way as in the MSA. If a new global best solution is found, hub-move local search, move local search and swap local search operators are applied to this solution in this order. Again this is a deterministic search with all possible moves tried in a fixed order.

6. Computational results

The proposed MSA and ACS heuristic algorithms were coded in C♯. We solved three mathematical programming formulations using CPLEX 12.6. Some preliminary experiments were carried out to determine good parameter settings for CPLEX. The only settings changed from the defaults were to use the dual-simplex algorithm at the root node and to turned off all cut generation. For these problems, cut generation did not improve solution times. All computations were executed on a linux server with a 2.50 GHz Intel Xeon processor and 16 GB RAM. We test the performance of our methods with four different data sets. The CAB data O'Kelly (1987) is derived from airline passenger data between 25 US cities. The Turkish network (TR) instances Tan and Kara (2007) include 81 nodes corresponding to administrative districts in Turkey and demand data from a cargo application. The Australia Post (AP) data (Ernst and Krishnamoorthy, 1996) is based on 200 postcodes and mail volumes in an Australian city. Finally random data (URAND) (Meyer et al., 2009; Ilic et al., 2010) is used to test performance of our heuristic on instances with 400 nodes.[1]

The MSA heuristic that we propose in this study uses multiple parameters including $SPsize$,I_{iter}, T, K, $Max_{non-improving}$ and θ. After some initial testing we settled on the following parameter values: $SPsize = 10$, $I_{iter} = n$ (number of nodes), $K = 1.0$, $\theta = 0.95$ and $Max_{non-improving} = 200$. For the MSA algorithm, the probabilities for the intra-move, inter-move, intra-swap, inter-swap, edge-opt, and hub-move operators are set at 0.2, 0.2, 0.2, 0.2, 0.1, and 0.1, respectively.

For the ACS algorithm, after some experimentation, we found the following parameter values to generate the best results: $\rho = \rho' = \rho'' = 0.02$; $q = q' = q'' = 0.1$; $Q = 0.05$, and the number of ants equals the number of nodes ($h = n$).

[1] Available from http://turing.mi.sanu.ac.rs/nenad/phub/.

Table 1
Results on the CAB data set and the Turkish network for $n = 10$. M1, M2, M3 refer to *Model1* to *Model3*. M1 & M2 have the same LP gap. The best CPU time requirement highlighted in bold for each instance.

Problem	*Optimal*	*Hubs*	*% LP Gap*		*Branch & Bound Nodes*			*CPU time (sec)*		
			M1/2	M3	M1	M2	M3	M1	M2	M3
CAB10.2.1	1.35135e + 09	4,9	54.23	33.15	445248	360388	4359	593.5	482.46	**331.87**
CAB10.2.2	1.00885e + 09	4,9	38.7	20.44	178412	93241	411	312.39	88.05	**27.34**
CAB10.3.1	1.03994e + 09	4,5,9	40.53	23.47	175203	183551	1335	282.46	259.73	**94.85**
CAB10.3.2	8.23251e + 08	4,7,9	24.88	12.95	58077	14764	92	138.81	22.19	**14.05**
CAB10.4.1	8.35553e + 08	4,6,7,9	25.98	13.10	25350	40690	308	38.81	58.68	**32.94**
TR10.2.1	2.30815e + 09	1,6	48.79	26.14	95135	76755	1801	116.42	**92.1**	152.93
TR10.2.2	1.79698e + 09	3,6	34.29	14.62	60001	34244	119	89.88	33.76	**14.69**
TR10.3.1	1.77433e + 09	1,3,6	33.46	13.79	35101	68585	185	45.18	90.56	**24.71**
TR10.3.2	1.43641e + 09	1,3,6	17.8	5.46	11064	6043	31	43.1	**6.39**	8.13
TR10.4.1	1.47962e + 09	1,3,6,7	20.21	5.39	22115	17467	33	37.78	30.95	**11.49**

6.1. Effectiveness of the formulations

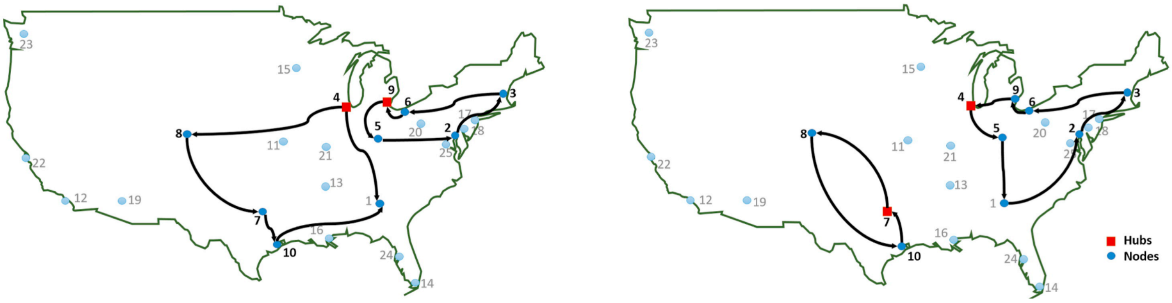
In this section we compared the effectiveness of our mathematical model formulations in Table 1. To compare the mathematical models, we first focused on the cases with $n = 10$ which could be solved optimally in a reasonable amount of time. We varied the problem instances by using different values of p and q.

In the first column of the Table 1 we give the characteristics of the test problem as $n.\ p.\ q$, with n nodes, p hubs and q vehicles per hub. The next two columns give the optimal objective and hub locations. We also provide the LP relaxation gaps, number of branch and bound nodes, and the required CPU times. Here the LP gap is calculated as (1 − LP solution/optimal solution) × 100.

For the 10 node instances, we could solve the three mathematical models to optimality in less than 10 min. *Model3* always has the tightest LP relaxation over all the problem instances. This demonstrates empirically that *Model3* is the tightest formulation as expected. This large and strong formulation with tighter lower bounds from LP relaxation significantly reduced the number of tree nodes that have to be searched, while requiring much more time to solve the LP for each node. The net effect is that *Model3* requires almost 40% less CPU time on average than *Model2*. This is a surprising result as the three-indexed formulations perform much better for single allocation p-hub median problems Ernst and Krishnamoorthy (1996). Based on these results we will only consider *Model3* below.

Table 2
Effect of discount factor on the CAB data set for $n = 10$; solution times for *Model3*; the increase in cost is with respect to the p-hub median objective for the same node-to-hub allocation decisions.

CAB Problem	α	Hubs	Optimal value	Routes	Nodes	CPU (s)	% Incr. in costs
10.2.1	1	4,9	1.35135 + E9	4–8–7–10-1–4; 9–5–2–3–6–9	4359	331.87	54.54
	0.8	4,9	1.32758 + E9	4–8–7–10-1–4; 9–5–2–3–6–9	5573	410.96	55.94
	0.6	4,9	1.30381 + E9	4–8–7–10-1–4; 9–5–2–3–6–9	5541	459.5	57.43
	0.4	4,9	1.28004 + E9	4–8–7–10–1–4; 9–5–2–3–6–9	6314	491.82	59.10
	0.2	4,7	1.24030 + E9	4–5–1–2–3-6–9-4; 7–8–10–7	6658	921.71	87.10
10.2.2	1	4,9	1.00885 + E9	4–7-10-4; 4–8-4; 9–3–2–6–9; 9–5–1–9	411	27.34	14.61
	0.8	4,9	9.84382 + E8	4–7–10–4; 4–8–4; 9–3–2–6–9; 9–5–1–9	530	37.99	15.02
	0.6	4,9	9.59913 + E8	4–7–10–4; 4–8–4; 9–3–2–6–9; 9–5–1–9	550	39.79	15.47
	0.4	4,7	9.28417 + E8	4–2–3–4; 4–5–6–9–4; 7–8–7; 7–10–1–7	823	44.46	24.26
	0.2	6,7	8.71064 + E8	6–3–2–6; 6–9–4–5-6; 7–8–7; 7–10–1–7	750	41.75	32.01
10.3.1	1	4,5,9	1.03993 + E9	4–8-4; 5–7–10–1–5; 9–3–2–6–9	1335	94.85	43.85
	0.8	4,7,9	9.97949 + E8	4–1–5–4; 7–8–10–7; 9–3–2–6–9	1353	112.69	38.48
	0.6	4,7,9	9.28959 + E8	4–1–5–4; 7–8–10–7; 9–6–2–3–9	832	76.98	42.56
	0.4	4,7,9	8.59969 + E8	4–5–1–4; 7–8–10–7; 9–6–2–3–9	934	82.04	47.60
	0.2	4,7,9	7.90979 + E8	4–5–1–4; 7–8–10–7; 9–6–2–3–9	854	66.05	53.99
10.3.2	1	4,7,9	8.23251 + E8	4–1–4; 4–5–4; 7–8–7; 7–10–7; 9–6–9; 9–3–2–9	92	14.05	4.26
	0.8	4,7,9	7.54261 + E8	4–1–4; 4–5–4; 7–8–7; 7–10–7; 9–6–9; 9–3–2–9	47	14.18	4.67
	0.6	4,6,7	6.8430 + E8	4–1–4; 4–5–4; 6–2–3–6; 6–9–6; 7–8–7; 7–10–7	30	12.08	5.28
	0.4	4,6,7	6.0912 + E8	4–1–4; 4–5–4; 6–2–3–6; 6–9–6; 7–8–7; 7–10–7	27	10.72	5.98
	0.2	4,6,7	5.3394 + E8	4–1–4; 4–5–4; 6–2–3–6; 6–9–6; 7–8–7; 7–10–7	33	9.49	6.88
10.4.1	1	4,6,7,9	8.35553 + E8	4–8–4; 6–3–2–6; 7–10–7; 9–5–1–9	308	32.94	9.05
	0.8	4,6,7,9	7.69195 + E8	4–8–4; 6–3–2–6; 7–10–7; 9–5–1–9	155	21.48	9.90
	0.6	4,6,7,9	7.02838 + E8	4–8–4; 6–3–2–6; 7–10–7; 9–5–1–9	184	27.37	10.94
	0.4	4,6,7,9	6.36480 + E8	4–8–4; 6–3–2–6; 7–10–7; 9–5–1–9	386	32.31	12.22
	0.2	1,3,4,7	5.41605 + E8	1–8–1; 3–2–3; 4–5–6–9–4; 7–10–7	201	22.88	10.87



(a) CAB10.2.1, α=1.0;0.8;0.6;0.4 (b) CAB10.2.1, α=0.2

Fig. 2. CAB10.2.1 problem solutions with respect to different discount factors.

The next question we investigated was the effect of the discount factor on the solution of the hub location-routing design network. For that purpose we varied α from 0.2 to 1.0 on the CAB network examples with 10 nodes. The results are summarized in Table 2 and Figs. 2 and 3.

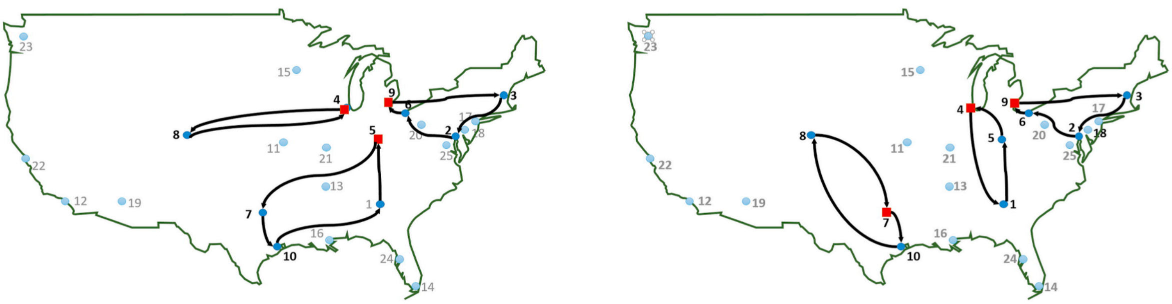
For each instance, Table 2 reports the inter-hub discount α, the hub nodes selected, solution cost, vehicle routes, branch & bound nodes and the required CPU time in seconds. The last column shows the percentage increase in transportation cost when the same node to hub allocation decisions are used in single allocation *p*-hub median problem.

The results show a large range of CPU times (9.49 to 921.71 seconds). The main factor affecting the problem difficulty appears to be the average number of nodes visited per vehicle which varies from 4 (10.2.1) to 1.5 (10.4.1). A less pronounced effect is that for instances with smaller values of α, CPLEX tended to require more CPU time; however, this effect is inconsistent and most pronounced for the instance CAB10.2.1. It may well be that this is in part again because for small α the allocation (vehicle routing) becomes more important and vehicle routing is a difficult problem in its own right.

Observe from Table 2 that the location of the hubs and the related route structure is changed at least once as the discount factor α is varied. For CAB10.2.2 the five different discount values give rise to three different solutions. Unsurprisingly, since the discount affects the cost of flow between hubs, the vehicle routes only change when a different set of hubs is optimal.

Table 2 shows that the percentage increase in transportation costs compared to the direct (single) allocation cost is up to 87.10%. The reason for this increase stems from the structure of our objective function, because the flows are cumulatively transferred by visiting all stopovers along a route while performing simultaneous pick-up and delivery. By contrast, the *p*-hub median problem assumes a separate vehicle services each non-hub node, allowing a direct connection. In the extreme case, where each vehicle visits a single node, the costs are the same as the single allocation *p*-hub median costs, hence the small difference for the CAB10.3.2 instances. While the discount factor appears to affect the percentage in Table 2, this is only because it affects the overall cost of the solution. The absolute difference in cost due to simultaneous pick-up and delivery for a given route is unaffected by this.

Figs. 2 and 3 illustrate the network structure of the optimal solutions from the 10.2.1 and 10.3.1 CAB instances. Hub locations tend to vary with different discount factors, but depend partly on the total demand. For instance Chicago (4) has the highest flow requirement in the data set and was chosen as a hub in every 10 node instance. For CAB10.2.1 when $\alpha = 0.2$ (Fig. 2b), the hubs are located 790 kms apart in Chicago (4) and Dallas (7). However for larger α the discount is insufficiently so Detroit (9) is chosen as second hub even though it is only 237 kms from Chicago (4) (Fig. 2a). Similarly for CAB10.3.1, when $\alpha = 1.0$, Chicago (4), Cincinnati (5) and Detroit (9) are located as hub nodes (Fig. 3a), and when $\alpha \leq 0.8$ the hubs are Chicago (4), Dallas (7) and Detroit (9) (Fig. 3b). It is also interesting that the same routes are used for Dallas in Figs. 2b and 3b indicating that good routes are sometimes stable when changes happen in another part of the network. We conclude that, as the cost for inter-hub transfer decreases, the benefit from the inter-hub discounts outweighs the gains made through simultaneous pick-up and delivery. Hence, the hubs are further spread and the optimal vehicle routes become shorter.



(a) CAB10.3.1, α=1.0 (b) CAB10.3.1, α=0.8;0.6;0.4;0.2

Fig. 3. CAB10.3.1 problem solutions with respect to different discount factors.

Table 3
Results for MSA and ACS on the CAB data set; Heuristics run times are 500 s for 10 node and 2000 s for 15 node instances in total; 2 h limit on CPLEX (∗∗ indicates that the time limit was reached by CPLEX).

Problem	Hubs	CPLEX	MSA % Gap		ACS % Gap	
		Best Cost	Min	Avg	Min	Avg
CAB10.2.1	4,9	1.35135 + E9	0.00	0.00	0.00	0.00
CAB10.2.2	4,9	1.00885 + E9	0.00	0.00	0.00	0.00
CAB10.3.1	4,5,9	1.03993 + E9	0.00	0.001	0.00	0.00
CAB10.3.2	4,7,9	8.2325 + E8	0.00	0.00	0.00	0.00
CAB10.4.1	4,6,7,9	8.3555 + E8	0.00	0.00	0.00	0.01
CAB15.2.1**	4,9	5.57793 + E9	0.00	0.00	0.00	0.04
CAB15.2.2**	4,9	4.02115 + E9	0.00	0.00	0.01	0.04
CAB15.3.1**	4,11,15	4.34583 + E9	0.00	0.01	0.00	0.03
CAB15.3.2	4,5,11	3.22922 + E9	0.00	0.006	0.02	0.06
CAB15.4.1**	4,5,9,15	3.5329 + E9	0.00	0.009	0.01	0.06
CAB15.4.2	4,5,11,13	2.89078 + E9	0.00	0.007	0.00	0.06
CAB15.5.1**	1,4,8,9,11	3.01715 + E9	0.00	0.004	0.00	0.08
CAB15.6.1**	1,4,6,7,8,9	2.73778 + E9	0.00	0.002	0.00	0.06
		Avg	0.00	0.003	0.003	0.034

6.2. Performance of the heuristics

In this section, we describe the performance tests of the MSA and ACS heuristics on the CAB data set and the Turkish network. In Tables 3 and 4, we give the performance measures based on 10 repeated runs with different random number seeds. The inter-hub discount factor α is set to 1.0 for all the experiments reported in these two tables. For the MSA and ACS, the upper bound gap is calculated as (heuristic solution/optimal solution − 1) × 100. The best known solution value was used where CPLEX could not solve instances to optimality within two hours.

Observe from Table 3 that the MSA algorithm was able to find the best known solution in at least one of the ten runs. The ACS algorithm performed somewhat worse on the CAB data set, requiring 10 runs to achieve the same average performance as MSA achieves in a single run.

Table 4 summarizes results for the Turkish network. Again the MSA algorithm performed better on average, producing solutions that are at most 0.01% away from the best known solution.

A closer analysis of the ACS algorithm results showed that the optimality gaps usually stem from a poor choice of hubs leading to higher cost allocations and route structures. We conjecture that this issue might be overcome by a more sophisticated heuristic function to find the hubs in the hub selection phase of ACS, but initial experiments with alternative heuristics did not result in an improvement. A similar analysis of the MSA algorithm did not find any particular pattern with respect to the resulting location-allocation or routing structure. The neighborhood search is sufficiently powerful to escape local optima in most cases. Overall both of our heuristic algorithms produce good quality solutions using a reasonable amount of CPU time in their current forms.

As noted earlier, we develop a procedure for a move operator which requires $\mathcal{O}(n)$ operations. To evaluate the importance of this we tested MSA_{wm} and ACS_{wm}, the algorithms MSA and ACS without the move operator. The quality of solutions produced by MSA_{wm}

Table 4
Results for MSA and ACS; MSA_{wm} and ACS_{wm} on the Turkish network; Heuristics run time are 500 s for 10 node; 2000 s for 15 node instances in total; 2 h limit on CPLEX (∗∗ indicates CPLEX reached the time limit).

Problem	Hubs	CPLEX	MSA % Gap		ACS % Gap		MSA_{wm} % Gap		ACS_{wm} % Gap	
		Best Cost	Min	Avg	Min	Avg	Min	Avg	Min	Avg
TR10.2.1	1,6	2.30815 + E9	0.00	0.00	0.00	0.00	0.00	3.06	0.00	0.00
TR10.2.2	3,6	1.79698 + E9	0.00	0.00	0.00	0.00	0.00	4.42	0.00	0.00
TR10.3.1	1,3,6	1.77432 + E9	0.00	0.00	0.00	0.00	0.00	3.34	0.00	0.00
TR10.3.2	1,3,6	1.43641 + E9	0.00	0.0002	0.00	0.00	0.00	4.14	0.00	0.00
TR10.4.1	1,3,6,7	1.47962 + E9	0.00	0.001	0.00	0.01	0.00	5.92	0.00	0.00
TR15.2.1**	1,6	3.29315 + E9	0.00	0.00	0.00	0.06	0.00	6.96	0.00	0.00
TR15.2.2**	1,6	2.70514 + E9	0.00	0.002	0.00	0.03	0.00	3.23	0.03	0.03
TR15.3.1**	1,3,6	2.69003 + E9	0.00	0.003	0.00	0.01	0.00	4.42	0.00	0.5
TR15.3.2	1,3,6	2.10818 + E9	0.00	0.01	0.00	0.06	0.00	7.89	0.00	0.21
TR15.4.1**	1,3,5,6	2.28263 + E9	0.00	0.002	0.00	0.02	0.00	4.58	0.00	0.38
TR15.4.2	1,3,5,6	1.90908 + E9	0.00	0.002	0.00	0.02	0.00	6.26	0.00	0.28
TR15.5.1	1,2,3,6,7	1.99179 + E9	0.00	0.003	0.00	0.04	0.00	7.33	0.00	0.00
TR15.6.1	1,3,5,6,7,12	1.85613 + E9	0.00	0.007	0.01	2.03	0.00	5.09	0.00	0.00
		Avg	0.00	0.002	0.00	0.223	0.00	5.126	0.002	0.099

and ACS_{wm} are reported in the last four columns of Table 4. The ACS_{wm} simply has the move local search operator removed, whereas in the MSA_{wm} the probabilities for the intra-swap, inter-swap, edge-opt, and hub-move operators are set at 0.25. There is also no move local search operator in the MSA_{wm}. The time limits for the heuristics were 50 and 200 s for the 10 node and 15 node instances, respectively. This includes the "warm-up" time for MSA to find the initial temperature.

Observe from Table 4 that the average gaps with the move operator (MSA,ACS) are significantly better than without (MSA_{wm}, ACS_{wm}). Although the MSA_{wm} obtained all of the best solutions, the average gap increased from 0.002% to 5.126%. This coincides with our expectations as without the move operator MSA_{wm} has a smaller neighbourhood and can only evaluate fewer neighbours. The ACS_{wm} has a similar performance without the move operator, indicating that the pheromone based search is more important to the ACS performance. It is obvious that for larger instances, the chance of finding better solutions by means of our move operator that operates in $\mathcal{O}(n)$ time increases substantially, because it allows the exploration of more of the solution space in a shorter time in the MSA algorithm.

Tables 3 and 4 also indicate that the CPU time requirements for CPLEX increase significantly as the problem size is increased to 15 nodes with 10 out of 16 instances not solved to optimality in two hours. To try to improve this performance we will look at using heuristics and a change in branching strategy.

6.3. Variable ordering and heuristic upper bound analysis with Model3

In this section, we analyze the effects of using a priority order to enforce branching on hub variables before other variables, and using the MSA solution as an initial upper bound in the CPLEX branching process. We observed in the previous section that the location of the hubs affects the solution quality considerably, thus we changed the variable priority to ensure CPLEX branches on the h_i variables before selecting any other variables. Since our MSA algorithm yields high-quality solutions and could find all the best solutions for the small sized CAB and the Turkish network data sets, we use the solutions obtained from our MSA algorithm as upper bounds to try to reduce the size of the branch and bound tree.

For this analysis, we again concentrate on 10 node instances because larger node problem instances were still too difficult to solve in a reasonable amount of time. We compared the default settings of CPLEX with the option of using both the variable ordering and heuristic bounds ($O{-}H$). Table 5 shows the number of nodes in the branch and bound tree, LP % gaps and the CPU requirements reported by CPLEX for the CAB data set and Turkish network.

The results show that the $O{-}H$ settings always reduced the CPU time requirement by an average of 28%, and also reduced the number of branch and bound nodes for every instance except CAB10.2.1 and TR10.4.1. In general, the branch and bound numbers increase with problem complexity, which is mainly determined by the average number of nodes visited per vehicle.

We also remark here that the LP relaxations are improved substantially with $O{-}H$ settings. In the instances where LP gaps are the same for both settings, we observed that initial incumbent solutions were always larger with default settings resulting in a larger tree to be searched.

In Table 6, we present results for 15 node instances on the CAB data set and the Turkish network. The best solution was always found with the $O{-}H$ settings and the root node LP gap is the same for both approaches. The final gap is calculated as (ub-lb)/ub× 100, with the final upper (ub) and lower bound (lb). The gap for the default settings is the percentage difference in best solution found by this setting and the $O{-}H$ setting.

Observe from Table 6 that both versions could solve six out of 16 problem instances within two hours, but the $O{-}H$ settings required significantly less CPU time on average. The CPU time requirements by CPLEX are considerably higher to solve the models to optimality on the CAB data set compared to the Turkish network. This might stem from the spatial distribution of the demand centers in the CAB data set, where nodes more evenly distributed in the Turkish network. Notice also that the LP gaps are much larger for the CAB data set compared to the Turkish network and CPLEX needed longer time to close those gaps. It is obvious from the table that $O{-}H$ settings are superior to default settings in terms of computational time and solution quality.

We also remark here that we tried to solve 15 node instances with our $\mathcal{O}(n^3)$ mathematical models, but could not solve any to optimality within two hours. Even with the help of better parameter settings by using the fastest mixed integer programming model

Table 5
Results on the CAB data set and the Turkish network for $n = 10$ with $O{-}H$ and Default settings.

Problem	Nodes		LP% Gap		CPU (s)	
	O−H	*Default*	*O−H*	*Default*	*O−H*	*Default*
CAB10.2.1	4741	4359	33.15	33.15	292.87	331.87
CAB10.2.2	305	411	20.44	20.44	17.67	27.34
CAB10.3.1	825	1335	23.47	23.47	56.95	94.85
CAB10.3.2	53	92	12.95	12.95	10.13	14.05
CAB10.4.1	152	308	8.84	13.1	16.92	32.94
TR10.2.1	1577	1801	12.93	26.14	121.19	152.93
TR10.2.2	117	119	6.39	14.62	9.64	14.69
TR10.3.1	146	185	6.11	13.79	14.75	24.71
TR10.3.2	11	31	1.29	5.46	3.95	8.13
TR10.4.1	41	33	1.82	5.39	7.49	11.49

Table 6
Results on the CAB and the Turkish network data sets with $O\text{–}H$ and Default settings; all gaps are relative to best cost obtained with the $O\text{–}H$ approach. (** Indicates CPLEX could not prove optimality in 2 h.)

Problem	CPLEX		$O\text{–}H$			Default		
	Optimal/Best	LP gap	Final Gap	Nodes	CPU (s)	Gap	Nodes	CPU (s)
CAB15.2.1**	5.57793 + E9	42.8	38.9	798	7200	15.65	726	7200
CAB15.2.2**	4.02115 + E9	27.4	12.6	1146	7200	4.56	1021	7200
CAB15.3.1**	4.34583 + E9	32	27	630	7200	11.75	842	7200
CAB15.3.2	3.22922 + E9	17.3	0.00	1708	5896.46	0.00	1769	6058.83
CAB15.4.1**	3.53290 + E9	23	13.5	921	7200	23.98	968	7200
CAB15.4.2	2.89078 + E9	13	0.00	467	1462.2	0.00	455	2579.6
CAB15.5.1**	3.01715 + E9	14.7	4.5	615	7200	5.68	772	7200
CAB15.6.1**	2.73778 + E9	9.3	2.1	1386	7200	0.55	1343	7200
TR15.2.1**	3.29315 + E9	32.3	19.5	1166	7200	11.68	875	7200
TR15.2.2**	2.70514 + E9	23.6	5.7	2134	7200	1.37	1792	7200
TR15.3.1**	2.69003 + E9	23.6	13.1	1811	7200	1.9	1326	7200
TR15.3.2	2.10818 + E9	0.5	0.00	841	943.29	0.00	990	1774.33
TR15.4.1**	2.28263 + E9	15	4.00	1614	7200	0.77	2130	7200
TR15.4.2	1.90908 + E9	6.7	0.00	272	535.59	0.00	328	1114.86
TR15.5.1	1.99179 + E9	0.5	0.00	1023	2807.45	0.00	1235	3553.33
TR15.6.1	1.85613 + E9	0.3	0.00	506	1402.9	0.00	358	1243.62

we developed, most of the 15 node instances could not be solved exactly by CPLEX. Therefore, for larger problem instances, we consider only the MSA and ACS solution methods. In the next subsection, we present larger sized instances with the 81 node Turkish network, 200 node AP data set and 400 node URAND data set.

6.4. Larger sized problem instances

Real world problems are, unfortunately, significantly larger than any solved exactly above. With our proposed heuristic algorithms, we are now able to solve larger problems on the Turkish network, AP and URAND data sets. In Table 7, we give the results for problems with 81, 200 and 400 node instances, which are taken from the Turkish network, AP and URAND data sets. Note here that these are the largest hub network data sets in the literature, and are obviously too large to be solved exactly using any of the

Table 7
URAND data set, Largest AP data set and Turkish network instances; 3000 s time limit for the Turkish network, 7200 s time limit for the AP instances; 10,800 s time limit for the URAND instances for each run.

Problem	Least cost	Hubs	MSA % Gap		ACS % Gap	
			Min	Avg	Min	Avg
TR81.4.2	1.00182E + 11	1, 3, 6,68	0.00	5.61	10.37	12.86
TR81.4.3	9.85959E + 10	6, 43, 54, 71	0.00	15.70	9.97	12.65
TR81.4.4	7.57444E + 10	6, 26, 38, 42	0.00	8.41	10.53	14.43
TR81.4.5	6.64072E + 10	6, 26, 38, 68	0.00	9.55	1.78	5.62
TR81.5.2	8.04442E + 10	6, 19, 26, 40, 54	0.00	8.07	8.19	10.95
TR81.5.3	7.88019E + 10	6, 26, 38, 42, 66	0.00	5.49	9.09	10.86
TR81.5.4	6.78240E + 10	6, 19, 38, 42, 54	0.00	6.14	11.13	13.26
TR81.5.5	7.10617E + 10	3, 6, 19, 38, 58	0.00	3.30	9.41	11.24
TR81.8.2	7.11370E + 10	1, 6, 26, 38, 42, 44, 54, 60	0.00	3.21	5.92	7.53
TR81.8.3	5.90961E + 10	3, 6, 38, 54, 55, 60, 66, 68	0.00	10.35	11.6	13.26
TR81.8.4	6.02347E + 10	1, 3, 6, 16, 19, 38, 41, 43	0.00	3.94	10.4	11.09
TR81.8.5	5.56973E + 10	3, 6, 19, 38, 54, 58, 66, 68	0.00	4.55	7.52	9.95
AP200.5.2	2.43804E + 8	147, 151, 157, 159, 161	0.00	4.07	11.44	15.21
AP200.5.3	1.90887E + 8	140, 150, 157, 159, 161	0.00	4.79	14.49	16.00
AP200.5.4	1.66870E + 8	149, 150, 158, 159, 161	0.00	2.99	8.43	11.71
AP200.5.5	1.46385E + 8	42, 148, 149, 158, 159	0.00	3.83	10.07	12.03
AP200.8.2	1.79846E + 8	101, 141, 147, 149, 151, 157, 159, 161	0.00	1.96	13.64	15.70
AP200.8.3	1.47678E + 8	92, 111, 121, 147, 149, 151, 158, 159	0.00	2.42	12.02	13.64
AP200.8.4	1.30552E + 8	65, 80, 81, 122, 140, 151, 159, 161	0.00	2.04	12.03	14.10
AP200.8.5	1.18018E + 8	42, 92, 112, 120, 151, 157, 159, 161	0.00	1.58	8.99	12.84
URAND400.8.2	1.45543E + 10	209, 309, 332, 296, 273, 134, 54, 361	0.00	5.10	9.09	11.45
URAND400.8.3	1.03353E + 10	200, 7, 266, 198, 34, 49, 317, 386	0.00	3.84	13.47	15.17
URAND400.8.4	9.41229E + 9	51, 7, 49, 34, 317, 326, 375, 266	0.00	3.37	11.19	14.48
URAND400.8.5	8.73828E + 9	284, 357, 34, 335, 112, 188, 81, 200	0.00	1.54	2.14	3.99
		Avg	0.00	5.07	9.70	12.08

mathematical models proposed in this study. However, we have included the results to demonstrate the ability of the heuristic algorithms to produce results for large problems in a reasonable amount of computing time.

For the Turkish network, AP and URAND data sets, we left all the parameters unchanged in the MSA and ACS algorithms while solving the test problems. We ran the problems 10 times starting with 10 different random number seeds. We limited the run times to 3000, 7200 and 10,800 s respectively, for the 81 node, 200 node and 400 node instances. The best objective value found in one of these 10 runs of each algorithm are reported in Table 7, together with the minimum and average percentage deviation above this least cost value.

We note here that we also aim to analyze the resulting hub location and routing networks with the generated instances on the Turkish network. When we look at the locations of the hubs in the 81 node Turkish network, our solutions with four and five hubs usually establish hubs at the center of Turkey. However, we found out that our solutions for the instances with eight hubs located between four and six of them in the seven geographical districts of Turkey. The resulting hub locations coincide with the actual locations of the hubs for cargo firms in terms of geographical districts. This shows that the way we have modelled our objective function and constraints is a reasonable approximation to the way cargo firms make their network design decisions. This is a reasonable conclusion because our objective function considers both the quantity of flow and the distance.

We also investigated the impact of the number of hubs and vehicles on the cost and structure of the solutions. Table 7 shows that the cost of the solutions tends to increase with decreasing number of hubs and vehicles operating on the routes. This is a trade-off against the fixed costs which are not explicitly included in our model. The only outlier that did not fit this observation is TR81.5.5 which has an increased total transportation cost. Here, the optimal solution is to establish all hubs at the center of Turkey and very close to each other. In particular, the hubs located in 19 (Corum) and 58 (Tokat) are established next to each other on the map, with 6 (Ankara) and 38 (Kayseri) also nearby. While generally more vehicles provide more flexibility and hence lower cost, it is possible to get an increase in costs when forcing the use of additional vehicles. The reason is that the simultaneous pick-up and delivery mode can provide a more direct route between two nodes, than using a separate vehicle to travel via the hub. The other instances of the 81 node Turkish network, 200 node AP and 400 node URAND data set showed the same tendency of decreasing costs with increasing number of hubs and vehicles. Therefore, we believe that decision makers for cargo delivery companies could use our methodology to determine the best trade-off in these operational cost savings against the increased cost and complexity of establishing hub location-routing networks with additional hubs or vehicles. It can be seen from Table 7 that the MSA performs significantly better than ACS for larger-sized problems. The only exceptions to this are the instances TR81.4.3 and TR81.4.5, where ACS produced a slightly better result on average. We observed that ACS algorithm generally selects different hub nodes from the MSA algorithm, which results in a different route structure and hence generally worse solution quality. However MSA is based on neighborhood search, thus the quality of solutions improves when using diverse neighborhood operators in the algorithm. It is clear from Table 7 that the MSA method scales better with increasing problem size than the other methods tested and can be expected to be the best performing method for even larger sizes.

Obviously, there is a trade-off between CPU time and solution quality. The above results may be improved if we use different parameter settings from those used to obtain results for the smaller problems. However, for the sake of consistency, the MSA and ACS heuristics were applied with the same parameters as before.

7. Conclusions

In this paper, we introduced a new hub location and routing problem motivated by the “delivery within the same day in the city” service offer of a cargo delivery company. The aim of our problem is to design a minimum cost hub location and routing network with simultaneous pick-up and delivery. The contributions of this paper are many-fold. First, we developed three mixed integer programming formulations of the problem, ranging in size from $O(n^3)$ variables to $O(n^4)$ variables. To the best of our knowledge, the mathematical models are the first in the literature that are proposed for the single allocation p-hub median location and routing problem with simultaneous pick-up and delivery. We also compared the solutions of our formulations with the solutions of the single allocation p-hub median problem on some instances. Although it is observed that generally high increases in costs occur when considering vehicle routes in the p-hub median problem, the investment and operational costs of vehicles are not considered in these models. When these costs are considered, it is expected that this new approach will reduce overall costs. This should make these models of great interest to cargo and postal delivery companies, allowing the number of vehicles that operate in the network and the cost of moving the commodity to be reduced substantially.

Comprehensive computational experiments are also presented with different formulations on the well-known CAB data set and on the Turkish network to test the efficiency of the models. As our numerical results demonstrate, our problem is significantly harder than the standard single allocation p-hub median problem. We showed that, unlike the experience in the literature for the single allocation variant, the formulation *Model3* with $\mathcal{O}(n^4)$ variables (where n is the number of nodes) is superior to the other formulations of our problem in terms of computational time and solution quality.

To solve realistically sized instances, it is clearly necessary to use heuristic approaches. It is not obvious what the best type of heuristic approach is for these challenging problems that involve decisions on location, allocation and routing. We adapted a state of the art ACS approach using a hierarchical pheromone structure. While this performs well compared to CPLEX, we are able to show that a local search approach based on MSA performs even better, particularly as the problem size increases. This is in part because we are able to perform very efficient evaluation of neighbourhood moves in $\mathcal{O}(n)$ time while the ACS approach has complexity $\mathcal{O}(n^2)$ for constructing and evaluating each solution.

We also conducted some analyses with the 81 node Turkish network, 200 node AP data set and 400 node URAND data set.

Managerial insights can be gained from the reduction in costs that can be achieved by investing in more hubs or vehicles. Our results show that the mathematical models and heuristic algorithms proved to be valuable tools for decision makers in the design of hub location and routing networks.

The models could readily be extended to include fixed costs on hubs and vehicle travel, as well as capacities. Another extension to the single allocation *p*-hub median location and routing problem with simultaneous pick-up and delivery is to include time constraints derived from the need of cargo companies to delivery within the same day. Such extensions are also possible but add additional scheduling considerations and hence further complexity into the models. The computational results indicate that there is a need for more sophisticated exact algorithms to allow medium or large size instances to be solved.

Acknowledgement

The authors thank three anonymous referees and editor, whose feedbacks helped us in improving the paper. The first author is supported from the Scientific and Technological Research Council of Turkey (TÜBİTAK).

References

Alumur, S., Kara, B.Y., 2008. Network hub location problems: the state of the art. Euro. J. Oper. Res. 190 (1), 1–21.
Alumur, S.A., Kara, B.Y., Karasan, O.E., 2012. Multimodal hub location and hub network design. Omega 40 (6), 927–939.
Alumur, S.A., Yaman, H., Kara, B.Y., 2012. Hierarchical multimodal hub location problem with time-definite deliveries. Transport. Res. Part E: Logist. Transport. Rev. 48 (6), 1107–1120.
Bektas, T., 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34 (3), 209–219.
Bowen, J.T., 2012. A spatial analysis of fedex and ups: hubs, spokes, and network structure. J. Transport Geogr. 24, 419–431.
Calik, H., Alumur, S.A., Kara, B.Y., Karasan, O.E., 2009. A tabu-search based heuristic for the hub covering problem over incomplete hub networks. Comput. Oper. Res. 36 (12), 3088–3096.
Camargo, R.S.D., Miranda, G.D., Lokketangen, A., 2013. A new formulation and an exact approach for the many-to-many hub location-routing problem. Appl. Math. Model. 37 (1213), 7465–7480.
Campbell, J.F., 1994. Integer programming formulations of discrete hub location problems. Euro. J. Oper. Res. 72 (2), 387–405.
Campbell, J.F., 1996. Hub location and the p-hub median problem. Oper. Res. 44 (6), 923–935.
Campbell, J.F., O'Kelly, M.E., 2012. Twenty-five years of hub location research. Transport. Sci. 46 (2), 153–169.
Cetiner, S., Sepil, C., Sural, H., 2010. Hubbing and routing in postal delivery systems. Ann. Oper. Res. 181 (1), 109–124.
Chen, J.-F., 2007. A hybrid heuristic for the uncapacitated single allocation hub location problem. Omega 35 (2), 211–220.
Claus, A., 1984. A new formulation for the travelling salesman problem. SIAM J. Algebr. Discrete Meth. 5 (1), 21–25.
Contreras, I., 2015. Hub location problems. In: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. Springer International Publishing, pp. 311–344.
Cunha, C.B., Silva, M.R., 2007. A genetic algorithm for the problem of configuring a hub and spoke network for a ltl trucking company in brazil. Euro. J. Oper. Res. 179 (3), 747–758.
Davari, S., Zarandi, M.F., Turksen, I., 2013. The incomplete hub-covering location problem considering imprecise location of demands. Sci. Iran. 20 (3), 983–991.
Dorigo, M., 2007. Ant colony optimization. Scholarpedia 2 (3), 1461.
Dorigo, M., Gambardella, L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. 1 (1), 53–66.
Drexl, M., Schneider, M., 2015. A survey of variants and extensions of the location-routing problem. Euro. J. Oper. Res. 241 (2), 283–308.
Ernst, A.T., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. Location Sci. 4 (3), 139–154.
Ernst, A.T., Krishnamoorthy, M., 1999. Solution algorithms for the capacitated single allocation hub location problem. Ann. Oper. Res. 86 (0), 141–159.
Estrada-Romeu, M., Robust, F., 2015. Stopover and hub-and-spoke shipment strategies in less-than-truckload carriers. Transport. Res. Part E: Logist. Transport. Rev. 76, 108–121.
Farahani, R.Z., Hekmatfar, M., Arabani, A.B., Nikbakhsh, E., 2013. Survey: hub location problems: a review of models, classification, solution techniques, and applications. Comput. Ind. Eng. 64 (4), 1096–1109.
Gelareh, S., Pisinger, D., deployment, Fleet, 2011. network design and hub location of liner shipping companies. Transport. Res. Part E: Logist. Transport. Rev. 47 (6), 947–964.
Ilic, A., Urosevic, D., Brimberg, J., Mladenovic, N., 2010. A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. Euro. J. Oper. Res. 206 (2), 289–300.
Karimi, H., Setak, M., 2014. Proprietor and customer costs in the incomplete hub location-routing network topology. Appl. Math. Model. 38 (3), 1011–1023.
Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.
Kratica, J., Stanimirovic, Z., Tosic, D., Filipovic, V., 2007. Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. Euro. J. Oper. Res. 182 (1), 15–28.
Lopes, M.C., de Andrade, C.E., de Queiroz, T.A., Resende, M.G.C., Miyazawa, F.K., 2016. Heuristics for a hub location-routing problem. Networks 68 (1), 54–90.
Meyer, T., Ernst, A.T., Krishnamoorthy, M., 2009. A 2-phase algorithm for solving the single allocation p-hub center problem. Comput. Oper. Res. 36 (12), 3143–3151.
Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. J. ACM 7 (4), 326–329.
Montoya-Torres, J.R., Franco, J.L., Isaza, S.N., Jimnez, H.F., Herazo-Padilla, N., 2015. A literature review on the vehicle routing problem with multiple depots. Comput. Ind. Eng. 79, 115–129.
Nagy, G., Salhi, S., 1998. The many-to-many location-routing problem. Top 6 (2), 261–275.
Nagy, G., Salhi, S., 2007. Location-routing: issues, models and methods. Euro. J. Oper. Res. 177 (2), 649–672.
O'Kelly, M.E., 1987. A quadratic integer program for the location of interacting hub facilities. Euro. J. Oper. Res. 32 (3), 393–404.
Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann. Oper. Res. 41 (4), 421–451.
Rieck, J., Ehrenberg, C., Zimmermann, J., 2014. Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. Euro. J. Oper. Res. 236 (3), 863–878.
Rodriguez-Martin, I., Salazar-Gonzalez, J.-J., Yaman, H., 2014. A branch-and-cut algorithm for the hub location and routing problem. Comput. Oper. Res. 50 (0), 161–174.
Salhi, S., Rand, G.K., 1989. The effect of ignoring routes when locating depots. Euro. J. Oper. Res. 39 (2), 150–156.
Serper, E.Z., Alumur, S.A., 2016. The design of capacitated intermodal hub networks with different vehicle types. Transport. Res. Part B: Methodol. 86, 51–65.
Skorin-Kapov, D., Skorin-Kapov, J., O'Kelly, M., 1996. Tight linear programming relaxations of uncapacitated p-hub median problems. Euro. J. Oper. Res. 94 (3), 582–593.
Stanimirovic, Z., 2010. A genetic algorithm approach for the capacitated single allocation p-hub median problem. Comput. Inform. 29 (1), 117–132.
Tan, P.Z., Kara, B.Y., 2007. A hub covering model for cargo delivery systems. Networks 49 (1), 28–39.
Ting, C.-J., Chen, C.-H., 2013. A multiple ant colony optimization algorithm for the capacitated location routing problem. Int. J. Prod. Econ. 141 (1), 34–44.
Topcuoglu, H., Corut, F., Ermis, M., Yilmaz, G., 2005. Solving the uncapacitated hub location problem using genetic algorithms. Comput. Oper. Res. 32 (4), 967–984.
Toth, P., Vigo, D., 2014. Vehicle Routing. Society for Industrial and Applied Mathematics, Philadelphia, PA.
Toyoglu, H., Karasan, O.E., Kara, B.Y., 2012. A new formulation approach for location-routing problems. Networks Spat. Econ. 12 (4), 635–659.
Wasner, M., Zapfel, G., 2004. An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. Int. J. Prod. Econ. 90 (3), 403–419.
Watson-Gandy, C., Dohrn, P., 1973. Depot location with van salesmen – a practical approach. Omega 1 (3), 321–329.
White, S.R., 1984. Concepts of scale in simulated annealing. AIP Conf. Proc. 122 (1), 261–270.
Yu, V.F., Lin, S.-W., 2014. Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery. Appl. Soft Comput. 24, 284–290.