Computers and Operations Research 85 (2017) 45–57

Contents lists available at ScienceDirect

Computers and Operations Research

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

Routing and scheduling decisions in the hierarchical hub location problem

Okan Dukkanci, Bahar Y. Kara*

Department of Industrial Engineering, Bilkent University, Bilkent, Ankara, 06800, Turkey

ARTICLE INFO

Article history:
Received 15 October 2015
Revised 15 March 2017
Accepted 27 March 2017
Available online 29 March 2017

Keywords:
Transportation
Hub location
Hierarchical network design
Time-definite delivery
Hub covering

ABSTRACT

Hubs are facilities that consolidate and disseminate flow in many-to-many distribution systems. The hub location problem considers decisions that include the locations of hubs in a network and the allocations of demand (non-hub) nodes to these hubs. We propose a hierarchical multimodal hub network structure, and based on this network, we define a hub covering problem with a service time bound. The hierarchical network consists of three layers in which we consider a ring-star-star (RSS) network. This multimodal network may have different types of vehicles in each layer. For the proposed problem, we present and strengthen a mathematical model with some variable fixing rules and valid inequalities. Also, we develop a heuristic solution algorithm based on the subgradient approach to solve the problem in more reasonable times. We conduct the computational analysis over the Turkish network and the CAB data sets.

1. Introduction

Hubs function as switching, transshipment and sorting points in many-to-many distribution networks. Instead of connecting each origin-destination (o-d) pair by a direct link, hubs provide a connection between each pair by using fewer links and concentrating demand flows to allow economies of scale.

The hub location problem is to decide on the locations of hubs and the allocations of demand nodes to hubs. Versions of the hub location problem are defined as 'single allocation' and 'multiple allocation'. In a single-allocation hub network, each demand node is assigned to exactly one hub. Whereas, in a multiple-allocation hub network, demand nodes can be allocated to more than one hub. The classic hub location problem has three main assumptions. First, the hub network is assumed to have a complete structure, with a link between each hub pair. Second, there are economies of scale between hubs. Third, direct transportation between demand node pairs (without using any hubs) is not allowed (Campbell, 1994).

Main application areas of the hub location problem are cargo delivery, telecommunications network design and air transportation. In this study, we mainly focus on a cargo delivery application in Turkey with a given service time promise. A classic cargo delivery system consists of branch offices and operation centers. Branch offices collect and distribute cargoes from/to customers directly, and operation centers collect and distribute cargoes from/to branch offices or send cargoes to another operation center. Although there can be more than one branch office in a city, operation centers do not exist in every city. Thus, each branch office must be assigned to operation center(s).

According to the above explanation of a cargo delivery system, cargo delivery networks and hub location networks are very similar. Branch offices and operation centers in cargo delivery networks can be considered as demand nodes and hubs, respectively. Also, there are economies of scale due to bulk transportation between operation centers. Therefore, the cargo delivery problem can be considered as a hub location problem (Alumur and Kara, 2009; Kara and Tansel, 2001; Tan and Kara, 2007; Yaman et al., 2007; 2012 and Alumur et al., 2012b). In Kara and Tansel (2001), the authors emphasize the importance of synchronization in cargo delivery systems. Later, Yaman et al. (2012) combine the release time scheduling and hub location problems in cargo delivery applications.

The classic hub location problem proposed by O'Kelly (1986a), 1986b), 1987) only considers minimization of the total transportation cost. However, in real life, cargo companies pay similar attention to customer satisfaction. To attract more customers, cargo companies focus on service levels. Service level in cargo delivery is usually measured by delivery time Yaman et al. (2012). Reducing delivery time is generally considered to increase customer satisfaction, and thus cargo companies offer different delivery time promises. For instance, in Turkey, cargo companies aim to deliver cargoes within 24 hours (next-day delivery). However, due to

* Corresponding author.
E-mail address: bkara@bilkent.edu.tr (B.Y. Kara).

http://dx.doi.org/10.1016/j.cor.2017.03.013

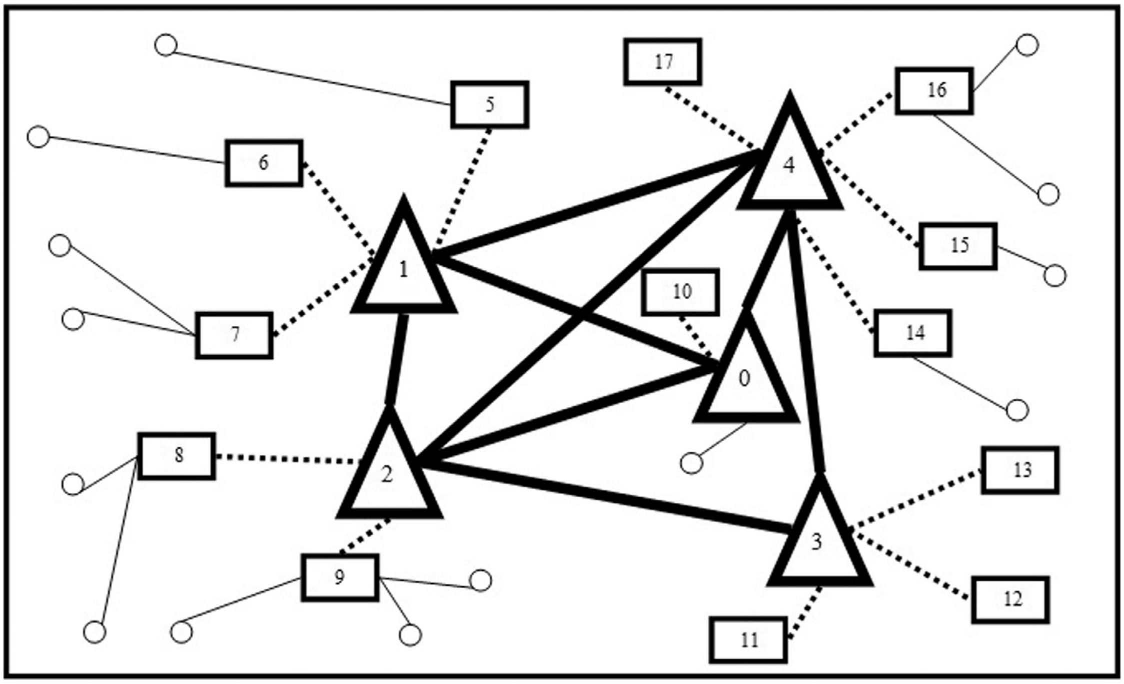

Fig. 1. Representation of a Multimodal Hierarchical Hub Network Instance.

Turkey's geographical structure, delivery within this time frame using ground transportation is almost impossible for some city pairs. Thus, in order to keep the next-day delivery promise between all city pairs, cargo companies in Turkey have begun to use airplanes in their distribution networks.

In the classic hub location problem, the hub network (the subnetwork that is induced by the hub nodes and links between them) is usually assumed to be complete with a link between each hub pair. When using airplanes, a complete hub network results in many flights, and operating a flight is very costly. Customarily, the operational cost of a flight consists of a fixed dispatch cost and a variable transportation cost, which depends on the length of the flight. In this research, we limit our study to consider only the fixed dispatch cost and to those cases where that cost is significantly more crucial than the variable transportation cost. We have two main motivations for this limitation. First, the fixed dispatch cost consists of crucial cost components of a flight such as taxiing and take off/landing costs, which are common for each flight not depending on the length of flight. Second, since we consider a real cargo delivery application in Turkey, the variable transportation cost is not as important as the fixed dispatch cost due to the geographical structure of Turkey in which the difference among distances between any two airport hub candidates can be negligible compared to other countries that has bigger land area such as USA. Based on these motivations and reasons, we set the main goal as minimizing the number of airline segments (flights). Thus, we want to provide the same worst-case level of service to all o-d pairs with minimal number of flights.

Motivated by the cargo company that uses trucks (small and large) along with airplanes in their network, we consider a hierarchical multimodal network with three layers and two types of hubs (ground and airport). Fig. 1 shows such an instance with 18 hubs; nodes 0 to 4 are airport hubs; nodes 5 to 17 are ground hubs and the small circles with no numbers represent the demand points. In this representation, airline segments are illustrated as thick lines between the airport hubs; highway segments are illustrated as thin lines between demand nodes and hubs (ground or airport), and as dashed lines between ground hubs and airport hubs.

The lowest layer of the network consists of the allocations of the demand points to the ground hubs and airport hubs (thin solid lines) as necessary to meet the single allocation. In this layer, a star structure is used to allocate the demand points. Each demand node is connected to exactly one hub (ground or airport) with a highway link. In real life, small trucks are used on these highway segments.

The middle layer includes the allocation of ground hubs to airport hubs (dashed lines), and we consider a star structure to allocate the ground hubs here as well. Each ground hub is connected to exactly one airport hub with a highway link. Large trucks, which are faster and have more capacity than small trucks, are assumed to be used on these highway segments, and thus economies of scale are considered.

Fig. 1 depicts a mesh structure in the first (top) layer (thick lines), where airport hubs are connected with each other via an airline segment. However, to accomplish the fundamental goal, that is, to decrease the number of flights, we propose using a ring structure in the top layer instead of a mesh structure, which can cause more flights (Fig. 2). We call this type of network a ring-star-star (RSS). We assume each ring will be served by separate airplanes. To cope with synchronization issues, routing and scheduling decisions must be considered together.

With the ring structure in the top layer, the airplane route is decided while covering all o-d pairs within a given time bound. In this study, motivated by the cargo company's application, we adopt a "pick up, then deliver" type of service, which means there are two separate tours; a pick-up tour and a delivery tour. In the pick-up tour, all demands are collected from their origins and sent to a specific airport hub. After all demands arrive at this airport hub, they are sent to their final destinations in the delivery tour. We need a specific airport hub to collect all the demands at one point, and we call this the central airport hub. In Fig. 2, we denote the central airport hub with a big circle. If one airplane in a ring is not enough to cover all o-d pairs within the time limitations, there can be more than one ring, as shown in Fig. 2. In each ring, exactly one airplane can travel because there is no capacity restriction.

Pick-ups from the origins to the central airport hub and deliveries from the central airport hub to the destinations are assumed

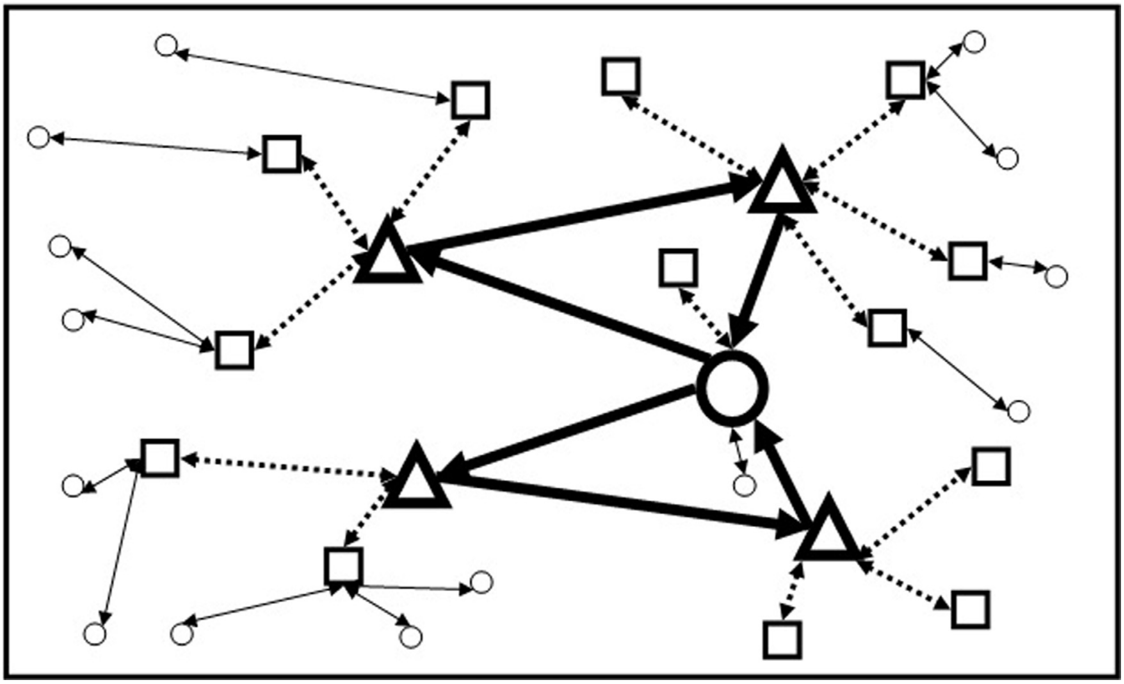

Fig. 2. Representation of a ring-star-star network instance.

to be symmetric. Therefore, the route of the delivery tour is the reverse route of the pick-up tour.

"Pick up, then deliver" service is common to most cargo companies. The main issue is to reach the consignee within the promised delivery time. In the pick-up tour, the airplane must complete the flights in half the time bound so it can complete the delivery tour in the other half.

Based on the proposed hierarchical multimodal network, the problem can be defined given a set of demand nodes, a set of possible locations for ground and airport hubs, the location of the central airport hub, the number of hubs to be located, the time bound and travel time parameters. Our proposed problem determines the location of ground hubs and airport hubs, the allocation of demand nodes to hubs (ground or airport), the allocation of ground hubs to airport hubs and the location of airline segments, all while ensuring that all o-d pairs can be served within the given time bound. In the objective function, the number of total flights (airline segments) is minimized.

In Section 2, we review the related literature, and Section 3 presents the mathematical model and some valid inequalities. In Section 4, we develop a heuristic solution algorithm based on the subgradient approach. In Section 5, we conduct a detailed computational analysis over two data sets, one from Turkey (TR) and one from the US Civil Aeronautics Board (CAB). Concluding remarks and future research directions are given in Section 6.

2. Related literature

The hub location problem was first introduced by O'Kelly (1986a); 1986b); 1987). In these studies, the author defines the problem and proposes the first mathematical model, which happens to be quadratic. Campbell (1994) categorizes the hub location problem into four problems based on the objective function: the p-hub median problem, the hub location problem with fixed costs, the p-hub center problem and the hub covering problem. For each problem, he presents linear formulations. After these pioneering studies, different versions of the hub location problems are studied by relaxing or changing the main assumptions of the hub location problem, which are highlighted in the previous section. Next, we analyze studies in the literature that relax these assumptions, namely; the hub location problem with ring structures, the multimodal hub location problems and the hierarchical hub location problems.

Relaxing the assumption which does not allow any direct links between two demand points, the ring structure concept is considered between demand nodes. The ring structure, as part of the hub location problem, is first presented by Nagy and Salhi (1998), in a many-to-many hub location routing problem. The authors state that the many-to-many location routing problem (LRP) can be reduced to the classic hub location problem when the routing problem is not considered. They present a mixed-integer programming formulation and propose some solution techniques. They also present a hierarchical heuristic, in which hub location is considered as a master problem and routing problems are considered as sub-problems. Routing problems are solved via the neighborhood search heuristic proposed by Nagy and Salhi (1996). Liu et al. (2003) present a mixed-truck delivery system that allows hub-and-spoke shipments and direct shipments. A heuristic is developed to determine the mode of delivery (hub-and-spoke or direct) and to perform vehicle routing in both delivery modes. Wasner and Zäpfel (2004) present a multi-depot hub-location vehicle-routing model for a network design of parcel services in Austria. This model can be considered as LRP, determining the location of hubs and depots, the routes between hubs/depots and their allocated demand points. The hub location part of this problem differs from the classic hub location problem in two aspects. First, there can be a direct shipment between two demand points. Second, the transportation cost between two hubs depends on the number of transports between those two hubs. The authors present a mixed-integer optimization model; however, due to its complexity, they develop a heuristic based on a local search procedure. For Turkey's postal delivery system, Çetiner et al. (2010) propose a combined hub location routing problem, which includes hub location decisions and routing decisions between demand points. In that study, multiple-allocation is allowed and it is assumed that the hubs and the vehicles are uncapacitated. The authors develop an iterative two-stage heuristic to solve this problem. De Camargo et al. (2013) present a new formulation for the many-to-many hub-location routing problem, considering single-allocation and uncapacitated hubs and vehicles. The completion of a tour is bounded by a service level and each customer is visited exactly once. Using Bender's decomposition, they solve the problem for up to 100 nodes. In addition to the ring structure, a tree topology on the sub-network induced by the hubs has been also studied by several researchers (Contreras et al., 2009; 2010; de Sá et al., 2013).

In the hub location problem, another important assumption is that between hubs and between hubs and demand nodes, only one transportation mode is used. Different transportation modes are not widely studied. In the literature, some researchers extend their studies by considering transportation mode decisions in addition to location and allocation decisions, and also by increasing the number of transportation modes. In multimodal transportation, transport modes have different cost structures. The first study of the hub location problem that includes a choice of transportation mode is proposed by O'Kelly and Lao (1991), where there are two fixed hub locations, called a mini hub and a master hub. This problem is solved by addressing two sub-problems. The first problem is the decision of transportation mode (air or truck) while satisfying given time limitations. The second problem is the allocation decision of cities to the mini hub. The multimodal hub location and hub network design problem is first introduced by Alumur et al. (2012a), who, in addition to the decisions of the classic hub location problem, consider the decision of transportation mode. The authors present a linear mixed-integer programming model and consider different variants of this problem.

Additionally, in the standard hub location problem the network consists of two layers; one between hubs and demand nodes and the other among hubs. However, real-life networks require more than two levels due to their complexity. This type of structures is called a hierarchical hub network. Smilowitz and Daganzo (2007) focus on the design of integrated package distribution systems for multiple transportation modes and a multiple service-level delivery network. They consider separate networks for each mode; for ground and air transportation modes, they propose ring-ring-complete and ring-ring-tree networks, respectively. They use a continuum-approximation approach to minimize cost. Yaman (2009) proposes a three-level hub network, which consists of a complete network on the top level and star networks on the second and third levels. Based on its objective, this problem can be considered as a hierarchical hub median problem. Yaman (2009) also studies a different version of this problem by considering service-level quality, proposing a mixed-integer programming model. Sahraeian and Korani (2010) consider the same three-level hub network structure under a maximal covering objective. Finally, Alumur et al. (2012b) present a hierarchical multimodal hub location problem with time-definite deliveries. They consider a star-incomplete-star network with air and ground transportation modes. They propose mixed-integer programming and a set of valid inequalities. In the mathematical model, they minimize the total transportation and operational costs.

For comprehensive surveys on hub location, we refer the reader to Alumur and Kara (2008); Campbell et al. (2002); Kara and Taner (2011), and Campbell and O'Kelly (2012); and to Nagy and Salhi (2007) for a location-routing survey.

3. Problem definition and formulation

The problem is defined on a complete directed graph $G = (N, A)$ where $N = \{0, 1, \ldots, n\}$ denotes the set of nodes and $A = \{(i, j) : i, j \in N \;\&\; i \neq j\}$ is the set of arcs. Demand point set is N. The possible hub and airport hub location sets are denoted by H and AH ($H \subseteq N \;\&\; AH \subseteq H$), respectively and 0 be the central airport hub ($0 \in AH$). A fleet of trucks and airplanes serves the customers from p ground or airport hub(s) (that need to be opened) within time bound T for each origin-destination pair. The travel time from node i to node j by small truck and airplane is denoted by t_{ij} and t_{ij}^{air}, respectively. The total loading-unloading time at airport l is denoted by m_l. α represents the discount factor of time for large trucks relative to small trucks. Maximum travel time between a ground hub and an airport hub plus loading-unloading time at that airport is denoted by M.

For this problem, we first propose a linear mixed-integer mathematical model.

The decision variables are defined as follows: x_{ij} equals to 1 if demand point $i \in N$ is allocated to hub $j \in H$, and 0 otherwise. x_{jj} equals to 1 if a hub is opened at node $j \in H$, and 0 otherwise. If hub $j \in H$ is allocated to airport hub $l \in AH$, then y_{jl} equals to 1, and 0 otherwise. y_{ll} equals to 1 if an airport hub is opened at node $l \in AH$, and 0 otherwise. u_{kl} equals to 1 if there is a direct link between airport hub $k \in AH$ and airport hub $l \in AH$, and 0 otherwise. The earliest time that all small trucks arrive at hub $j \in H$ is denoted by r_j and $r_l^{airplane}$ represents the earliest time that the airplane departs from airport hub $l \in AH$ to other airport hubs.

It is assumed that travel time data is symmetric and satisfies triangular inequality. Also, the loading and unloading time at airports is assumed to be independent of the load size. The mixed-integer programming formulation of the proposed problem is as follows:

$$\text{Minimize} \sum_{k \in AH} \sum_{l \in AH \setminus \{k\}} u_{kl} \quad (1)$$

subject to

$$\sum_{k \in H} x_{ik} = 1 \qquad \forall i \in N \quad (2)$$

$$\sum_{k \in H} x_{kk} = p \quad (3)$$

$$x_{ik} \leq x_{kk} \qquad \forall i \in N, k \in H \quad (4)$$

$$\sum_{l \in AH} y_{jl} = x_{jj} \qquad \forall j \in H \quad (5)$$

$$y_{jl} \leq y_{ll} \qquad \forall j \in H, l \in AH \quad (6)$$

$$y_{00} = 1 \quad (7)$$

$$\sum_{l \in AH \setminus \{k\}} u_{kl} = y_{kk} \qquad \forall k \in AH : k \neq 0 \quad (8)$$

$$\sum_{l \in AH \setminus \{k\}} u_{lk} = y_{kk} \qquad \forall k \in AH : k \neq 0 \quad (9)$$

$$\sum_{l \in AH \setminus \{0\}} u_{0l} = \sum_{l \in AH \setminus \{0\}} u_{l0} \quad (10)$$

$$r_j \geq t_{ij} \cdot x_{ij} \qquad \forall i \in N, j \in H \quad (11)$$

$$r_l^{airplane} \geq r_j + (\alpha \cdot t_{jl} + m_l) \cdot y_{jl} - M \cdot (1 - y_{jl})$$
$$\forall j \in H, l \in AH : l \neq j \quad (12)$$

$$r_k^{airplane} - r_l^{airplane} + T \cdot u_{kl} \leq T - (t_{kl}^{air} + m_l) \cdot u_{kl}$$
$$\forall k \in AH : k \neq 0, l \in AH : l \neq k \quad (13)$$

$$r_k^{airplane} \geq (t_{0k}^{air} + m_k) \cdot u_{0k} \qquad \forall k \in AH : k \neq 0 \quad (14)$$

$$2 \cdot r_0^{airplane} \leq T \quad (15)$$

$$x_{ik} \in \{0, 1\} \qquad \forall i \in N, k \in H \quad (16)$$

$$y_{jl} \in \{0, 1\} \qquad \forall j \in H, l \in AH \quad (17)$$

$$u_{kl} \in \{0, 1\} \qquad \forall k \in AH, l \in AH : l \neq k \quad (18)$$

$$r_j \geq 0 \qquad \forall j \in H \quad (19)$$

$$r_l^{airplane} \geq 0 \qquad \forall l \in AH \quad (20)$$

The objective function (1) minimizes the number of airline links between airport hubs. Constraint (2) ensures that every demand node is allocated to exactly one hub. By Constraint (3), the number of hubs to be located is p. With Constraint (4), we guarantee that no demand node is allocated to a non-hub node. By Constraint (5), a hub is allocated to exactly one airport hub. Also, with Constraint (5), $y_{kk} = 1$ implies that $x_{kk} = 1$. Constraint (6) guarantees that no hub is allocated to a non-airport hub. Constraint (7) establishes the central airport hub. Constraints (8) and (9) construct the ring structure for the airport hubs (top layer of the hierarchical network). Constraint (10) allows a structure with more than one ring, which means there can be more than one airplane, if necessary. Also, with Constraint (10), if no airport hub is opened, then the central airport hub is used as a central ground hub.

Constraint (11) calculates the earliest time that small trucks arrive at their allocated hub. Constraint (12) guarantees that an airplane cannot leave an airport hub before all the vehicles (small and large trucks) from the other nodes (demand point or ground hub) allocated to that airport hub have arrived. With Constraint (13), we ensure that the earliest time an airplane departs from any airport hub is within a predetermined time bound. Constraint (13) is an adaptation of the well known Miller-Tucker-Zemlin (MTZ) constraint proposed by Miller et al. (1960). Constraint (14) calculates the earliest time an airplane departs from the airport hub with a direct link to the central airport hub. We compute this time separately to complete the ring structure for the top layer. By Constraint (15), we guarantee that all o-d pairs are covered within a given time bound. Because we assume symmetrical travel time data, we consider pick up and delivery the same, so we multiply by 2. Finally, Constraints (16)-(20) are the domain constraints.

This mathematical model is a mixed binary programming model with $O(n^2)$ binary variables, $O(n)$ non-negative variables and $O(n^2)$ constraints where n is the number of nodes.

3.1. Pre-processing and valid inequalities

We now propose some variable fixing rules and valid inequalities. First, we present two variable fixing rules, which are prespecified as parameters of the model:

Variable Fixing Rule 1: For $i \in N$ and $j \in H\backslash\{i\}$, if $t_{ij} > T/2$, then demand node i cannot be allocated to the hub (ground or airport) j, because travel time between city i and city j exceeds half of the time bound T. With Variable Fixing Rule 1, if $t_{ij} > T/2$, then x_{ij} will be equal to 0.

Variable Fixing Rule 2: For $j \in H$ and $l \in A\backslash\{j\}$, if $\alpha \cdot t_{jl} + m_l > T/2$, then hub j cannot be allocated to the airport hub l, because the reduced travel time between city j and city l and the loading/unloading time at city l exceeds half of the time bound T. With Variable Fixing Rule 2, if $\alpha \cdot t_{jl} + m_l > T/2$, then y_{jl} will be equal to 0.

We now propose two valid inequalities:

For $i \in N$, $j \in H\backslash\{i\}$ and $l \in A\backslash\{i, j\}$, if $t_{ij} + \alpha \cdot t_{jl} + m_l > T/2$, then demand node i cannot be allocated to ground hub j, and ground hub j cannot be allocated to airport hub l at the same time. Therefore, the inequality

$$x_{ij} + y_{jl} \leq 1 \;\; \forall i \in N, j \in H\backslash\{i\}, l \in A\backslash\{i, j\}, \text{if } t_{ij} + \alpha \cdot t_{jl} + m_l > T/2 \quad (21)$$

is valid.

For $i \in N$ and $j \in H\backslash\{i\}$, if $t_{ij} + \alpha \cdot t_{jl} + m_l > T/2$ for airport hub l and there is a hub at city j, then demand node i cannot be allocated to ground hub j, and ground hub j cannot be allocated to airport hub l at the same time. Therefore, the inequality

$$x_{ij} + \sum_{l \in A\backslash\{i,j\}:\; t_{ij}+\alpha \cdot t_{jl}+m_l > T/2} y_{jl} \leq x_{jj} \;\; \forall i \in N, j \in H\backslash\{i\} \quad (22)$$

is valid. Note here that valid inequality (22) is the stronger version of valid inequality (21).

We analyze the performances of these two valid inequalities in detail in Section 5. Based on these analysis, we include valid inequality (22) into the mathematical model.

4. Lagrangian relaxation based solution approach

Since getting the optimal solution takes too much time despite of all variable fixing rules and valid inequalities, we propose an alternative solution approach for the problem under consideration. In order to find the optimal or near optimal solutions in a reasonable time, a different solution method based on the subgradient algorithm is developed to solve those problem instances, which cannot be solved in a few seconds. So, we propose this algorithm for the problem instances with tighter time bounds.

Subgradient algorithm is one of the well-known solution approaches for the combinatorial optimization problems based on the Lagrangian relaxation. The algorithm consists of two main components. First, the problem is relaxed by removing some sets of constraints from the formulation and adding them to the objective function after multiplying them with Lagrange multipliers. Solving the relaxed problem, in case of minimization, provides a lower bound for the original problem. Second, by utilizing the solution of the relaxed problem, a feasible solution is obtained. This feasible solution gives an upper bound for the original problem. By using these two components, the subgradient algorithm tries to strengthen the lower and upper bounds in order to fill the gap between them and reach the optimal solution.

Now, we explain the proposed subgradient algorithm in three parts. Initially, we describe the relaxed problem. Secondly, we give a detailed explanation on how to find a feasible solution by using the solution obtained for the relaxed problem. Finally, we define the subgradient algorithm itself.

4.1. Relaxed problem

In order to apply this Lagrangian relaxation approach, Constraints (13) are relaxed from the original formulation. Constraints (13) are versions of the Miller-Tucker-Zemlin subtour elimination constraint that keeps the departure times of the airplanes in a tour.

It is a big-M type constraint. The formulation of the relaxed problem with Lagrange multipliers π_{kl} is as follows:

$$\text{Minimize} \sum_{k\in A}\sum_{l\in A\setminus\{k\}} u_{kl}$$

$$+ \sum_{k\in A\setminus\{0\}}\sum_{l\in A\setminus\{k\}} \pi_{kl}\cdot(r_k^{airplane} - r_l^{airplane} + (T + t_{kl}^{air} + m_l)\cdot u_{kl} - T)$$

subject to

(2-12), (14-20), (22)

Solving the above formulation will give a lower bound for the original problem.

4.2. Finding a feasible solution

In order to apply a subgradient algorithm, we need to find an upper bound on the original problem. One approach is to change the solution obtained from the relaxed problem to make it feasible for the relaxed constraint. In our problem, it is hard to change the solution to make it feasible because we need to adjust the departure time of the airplanes and also we need to eliminate subtours if there are any. Thus, we propose to solve another problem which is a restricted version of the original problem. After solving the relaxed problem, we give the values of the decision variables associated with the location of hubs (x_{jj}) to the original problem and then we solve the resulting new problem in order to find an upper bound. The formulation of the new restricted problem is as follows:

$$\text{Minimize} \sum_{k\in A}\sum_{l\in A\setminus\{k\}} u_{kl} \tag{1}$$

subject to

(2-20), (22)

$$x_{jj} = x'_{jj} \qquad \forall j\in J : |J| = p \ \ \& \ \ J\subseteq H \tag{23}$$

where x'_{jj} is obtained from the relaxed problem.

However, by using Constraint (23) we cannot ensure the feasibility of the restricted problem. Due to the time bound constraint, it is possible that the opened facilities cannot cover all o-d pairs within the given time limit. Therefore, if the problem becomes infeasible because of Constraint (23), we need to decrease the number of fixed locations obtained from the relaxed problem. While decreasing the number of fixed locations, we randomly select the fixed location to remove. The reduction on the number of fixed locations continues until one fixed location that is the given central airport hub remains. So, we can adopt Constraint (23) as follows:

$$x_{jj} = x'_{jj} \qquad \forall j\in J : |J| \le p \ \ \& \ \ J\subseteq H \quad (23^*)$$

where set $|J|$ depends on the feasibility of the problem.

By solving this restricted problem, we can obtain an upper bound on the original problem.

4.3. Subgradient algorithm

By using lower and upper bounds obtained from the solution techniques explained previously, we can apply a subgradient algorithm to the proposed problem. The algorithm is constructed as follows:

Step 0: Choose an initial Lagrange multiplier π_{kl}^0 and set $t = 0$.

Step 1: Let $\pi_{kl} = \pi_{kl}^t$ and solve the relaxed problem with the optimal value $z(\pi_{kl}^t)$ and update the lower bound as follows:

$$LB \leftarrow max\{LB, z(\pi_{kl}^t)\}$$

Step 2: Given the location of hubs from the relaxed problem, solve the restricted problem with $z(x_{jj}^t)$. If the restricted problem yield an infeasible solution, the number of fixed locations obtained from the relaxed problem is decreased until the restricted problem provides a feasible solution. Then, the upper bound is updated as follows:

$$UB \leftarrow min\{UB, z(x_{jj}^t)\}$$

Step 3: Update the Lagrange multipliers as follows:

$$\pi_{kl}^{t+1} \leftarrow max\{\pi^t + \mu_t\cdot(r_k^{airplane} - r_l^{airplane} + (T + t_{kl}^{air} + m_l)\cdot u_{kl} - T), 0\}$$

where $\mu_t = f\cdot\frac{UB - z(\pi_{kl}^t)}{||r_k^{airplane} - r_l^{airplane} + (T + t_{kl}^{air} + m_l)\cdot u_{kl} - T||^2}$ and being f a number taken between 0 and 2 that is decreased after a certain number of iterations without improvement.

Step 4: $t \leftarrow t + 1$ and

Step 5: If t reaches the maximum number of iterations, then Stop. Otherwise, go to Step 1.

In the proposed subgradient algorithm, initially, an initial Lagrange multiplier (π_{kl}^0) is chosen. Then, based on the chosen Lagrange multiplier, the relaxed problem is solved and if the objective function value of the relaxed solution is greater than the lower bound, the lower bound is updated accordingly. After that, based on the location of hubs obtained from the relaxed solution, the restricted problem is solved to find a feasible solution and if the objective function value of this feasible solution is less than the upper bound, then the upper bound is updated accordingly. Next, the Lagrange multipliers are updated based on the given formula in Step 3. After updating the Lagrange multipliers, the relaxed problem is solved again with the new Lagrange multipliers and the algorithm continues until the maximum number of iterations is reached. The values of the different parameters of the subgradient algorithm are given in the computational analysis section.

5. Computational study

5.1. Data sets

In the computational studies, since we consider a cargo delivery application in Turkey, we use Turkish network (TR) data set. In 2007, Tan and Kara (2007) introduced the TR data set to the literature, and it consists of 81 cities in Turkey.

These cities are illustrated in Fig. 3; the numbers represent Turkey's vehicle license plate numbers, which are unique to each city. Each city is considered as a demand point, so there are 81 demand points ($|N|$ =81). There are 22 potential hub nodes ($|H| = 22$), represented as red circles in the map (Fig. 3). Since Afyon (3), Aksaray (68) and Duzce (81) do not have airports, 19 of these potential hub nodes are considered as potential airport hubs ($|AH| = 19$). Ankara (6) is considered the central airport hub due to its geographical and geopolitical advantages: it is near the center of Turkey, it has the country's second biggest amount of flow and it is the capital city.

The time discount factor α is taken as 0.9. Distance data is taken from Tan and Kara (2007). Travel times are calculated by assuming that the trucks travel at a speed of 70 km/hr and that the airplanes travel at a speed of 700 km/hr. The loading/unloading time at an airport is taken as 30 minutes.

In addition to the TR data set, we also consider the CAB data set, which is based on airline passenger interactions between 25 US cities in 1970, and was introduced to the literature by O'Kelly (1987). The CAB data set is illustrated in Fig. 4. Twenty-five nodes represent the demand nodes and the potential ground and airport hubs ($|N| = |H| = |AH| = 25$). The central airport hub is assigned to New York (17) because it is the biggest city in the US in terms of population. The distance data is taken from O'Kelly (1987). The other settings are the same as in the Turkish network.

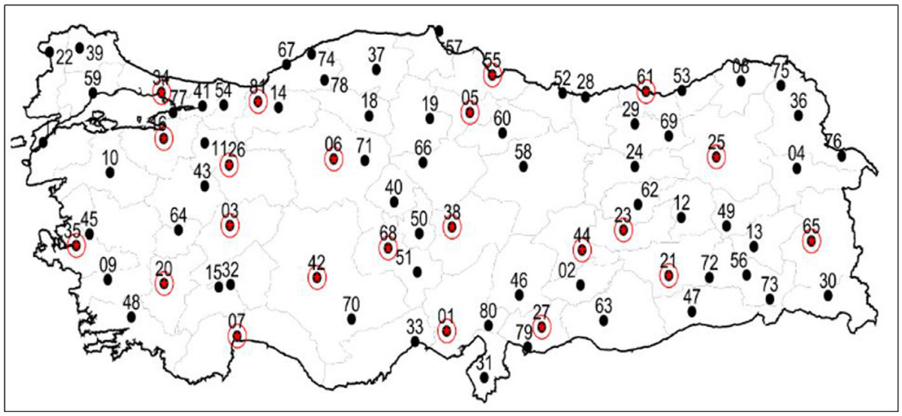

Fig. 3. Map of Turkey with cities and potential hub sets.

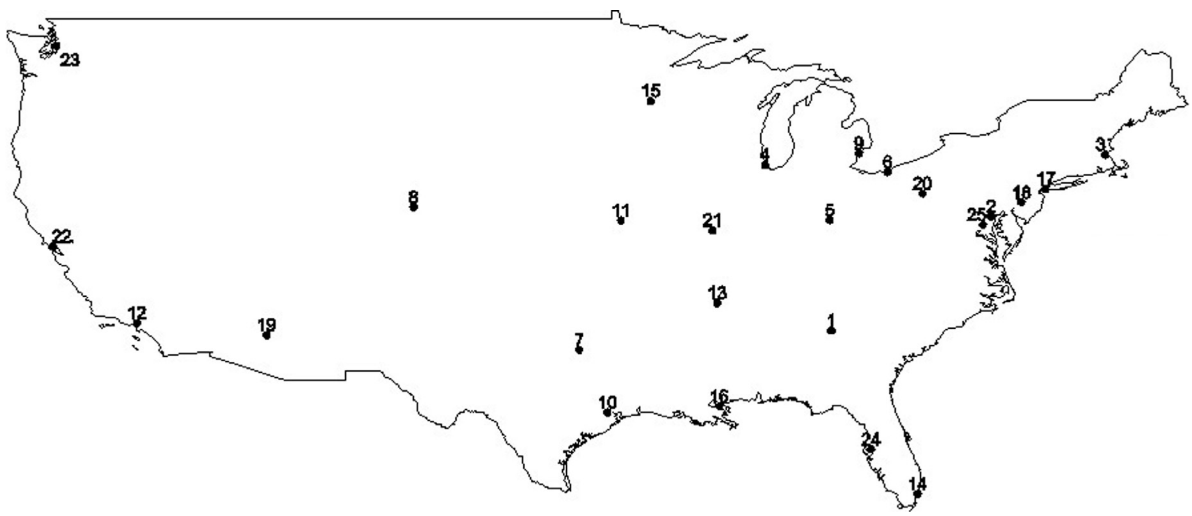

Fig. 4. Map of the US with 25 Cities.

For more information on the CAB and TR data sets, we refer the reader to Beasley (2012).

The subgradient algorithm related parameters are taken as the same for both data sets. The initial Lagrange multipliers (π_{kl}^{0}) are set to 0. f is taken as 1 and it is halved after 5 iterations without improvement. The initial lower and upper bounds of the algorithm are 0 and the highest number of airline segments possible, respectively. While applying the subgradient algorithm, we use different maximum number of iterations such as 1, 3, 5, 10, 20, 50 & 100.

Computational studies were carried out on a server with 4 AMD Opteron Interlagos 6282 SEs and 96 GB of RAM. The proposed formulation and the proposed subgradient algorithm were coded in Java via NetBeans IDE 8.0.2. As solver, we used Gurobi optimization software, version 6.0.3. The time limit was set as 6 hours.

5.2. Performance of the valid inequalities

We first tested the performance the two valid inequalities proposed in Section 3 for the mathematical model on the Turkish network data set. We did not consider Valid Inequality (21) and Valid Inequality (22) together because Valid Inequality (22) is the stronger version of Valid Inequality (21). The results are depicted in Table 1. The first two columns of Table 1 represent the parameters: the time bound (T) in hours and the number of hubs to be opened (p). The “No Valid Inequalities”, “Valid Inequality (21)” and “Valid Inequality (22)” columns represent the solutions of the appropriate models. The columns indicated by “LP Gap” and “CPU” show the gap in linear programming relaxation from the optimal value in percentages and the CPU time requirement in seconds, respectively. Finally, the column with label “Nodes” presents the number of nodes that were evaluated in a branch-and-bound tree.

As evident from Table 1, including only Valid Inequality (21) and only Valid Inequality (22) generally decreases CPU time. When we compare the results in terms of CPU time, the highest improvement was observed in the Valid Inequality (21) column for three instances, in the Valid Inequality (22) column for remaining seven instances. Based on these results, and because Valid Inequality (22) is the stronger version of Valid Inequality (21), we included only Valid Inequality (22) into the mathematical model.

5.3. Computational analysis of the mathematical model

We next observed the effect of T (time bound) and p (number of hubs to be opened) on the optimal network configuration by varying them. We varied the time bound and number of hubs to be established on both data sets. For the TR data set, we evaluated T between 24 and 14 hours, and determined that the RSS model is infeasible when the time bound is 12 hours or less. Because the US is much larger than Turkey in terms of area, the service time level must be higher for that case. Therefore, we varied T from 60 to 32 hours, and determined that the RSS model is infeasible when the time bound is 22 hours or less. For each T bound, we used three different values of p starting from the first feasible p value for the corresponding T value as long as the CPU time allowed it

Table 1
Performance of Valid Inequalities, TR Data Set.

T	p	No Valid Inequalities			Valid Inequality (21)			Valid Inequality (22)		
		Lp Gap	CPU	Nodes	Lp Gap	CPU	Nodes	Lp Gap	CPU	Nodes
18	6	95.83	321.27	202700	82.92	85.41	127476	82.92	**59.65**	145235
18	7	95.83	888.57	1910129	82.92	**32.5**	166974	82.92	123.52	190014
18	8	95.83	398.09	1001490	82.92	103.96	296374	82.92	**85.16**	177697
17	6	92.86	24.08	98732	78.57	25.51	95806	78.57	**23.83**	81900
17	7	92.86	641.98	1584395	78.57	303.15	1289257	78.57	**172.77**	873851
17	8	92.86	1021.43	4340362	78.57	899.56	8071544	78.57	**494.23**	2339322
16	6	94.44	60.4	288113	78.89	**16.92**	83396	78.89	74.22	226549
16	7	94.44	885.92	4408706	78.89	**450.47**	2890848	78.89	822.47	4992912
16	8	93.75	6864.49	19089335	76.25	2404.16	10474880	76.25	**1731.65**	8296941
15	8	95	7765.05	39483982	70	1577.5	10597225	70	**1210.64**	8305331

Table 2
Results on TR data set.

T	p	# Flights	# Airplanes	A. Hubs	G. Hubs (allocated A. Hubs)	CPU
24	2	2	1	6,23	–	0.09
24	3	2	1	6,23	21(23)	0.26
24	4	2	1	6,25	21(25), 65(25)	0.41
23	2	2	1	6,23	–	0.09
23	3	2	1	6,21	61(21)	0.28
23	4	2	1	6,25	21(25), 61(25)	0.29
22	2	2	1	6,23	–	0.09
22	3	2	1	6,23	21(23)	0.3
22	4	2	1	6,23	20(6), 61(23)	0.26
21	3	3	1	6,21,25	–	0.41
21	4	2	1	6,25	1(6), 65(25)	1.04
21	5	2	1	6,25	1(6), 21(25), 65(25)	0.81
20	4	3	1	6,21,25	34(6)	0.9
20	5	3	1	6,23,65	34(6), 61(23)	1.27
20	6	3	1	6,21,25	23(21), 65(21), 81(6)	1.36
19	5	5	2	6,21,25,26	27(21)	2.08
19	6	4	1	6, 21, 25, 34	16(6), 27(21)	5.73
19	7	4	1	6, 21, 25, 34	16(6), 27(21), 42(6)	1.3
18	6	6	2	6,27,34,61,65	20(6)	59.65
18	7	6	2	6,25,34,44,65	20(6), 55(6)	123.52
18	8	6	2	6,26,44,61,65	23(61), 34(26), 81(6)	85.16
17	6	7	2	6,25,27,34,35,65	–	23.83
17	7	7	2	6,1,16,20,25,65	27(1)	172.77
17	8	7	2	6,16,20,25,27,65	21(65), 23(25)	494.23
16	6	9	4	6,20,34,44,61,65	–	74.22
16	7	9	4	6,16,20,44,61,65	21(44)	822.47
16	8	8	3	6,7,16,21,25,27,65	55(6)	1731.65
15	8	10	3	6,20,21,25,27,34,61,65	–	1210.64
15	9	10	3	6,1,20,21,25,34,61,65	68(6)	14987.67
14	8	11	4	6,1,20,21,25,34,61,65	–	990.34
14	9	10	3	6,1,20,21,25,34,61,65	16(34)	18659.45

(the CPU time requirement of the model increases exponentially with p). Thus, for some T values, we only report results with two different p values. The results can be seen in Tables 2 and 3.

The first two columns in Table 2 represent the two parameters: T and p. The third and fourth columns in the table present the optimal objective function value (the number of airline links) and the number of airplanes (number of rings), respectively. We can deduce the number of airplanes from the solution by observing for how many i nodes, the value of u_{0i} equals to 1. The fifth column lists the location of airport hubs. The sixth column presents the location of ground hubs and their allocated airport hubs in parenthesis. Finally, the last column indicates the CPU time in seconds to solve the instance to optimality.

As evident from Table 2, two airline segments and two airport hubs with one airplane are enough to cover all o-d pairs in Turkey within 24, 23 or 22 hours. When we decrease T to 21 hours, three airport hubs with three airline segments are required. Then, if we increase p from 3 to 4, two airport hubs with two airline segments are enough to cover the country because of two additional ground hubs (which shows the importance of ground hubs). When $T = 20$ and $p = 3$, the problem becomes infeasible. If we increase p from 3 to 4, there can be a solution with three airport hubs and one ground hub rather than with four airport hubs. When we further decrease the time bound to 19 hours with five hubs are opened, two airplanes are required to cover all cities in Turkey with five flights among four airport hubs. If we continue to decrease the time bound to 16, 15 and 14 hours (which are very tight time bounds for the Turkish network), the number of airline segments increases to eight, nine, 10 and 11, and the number of airplanes increases to three and four.

On the other hand, one airplane and two airport hubs are enough to cover the US when the time bound is between 60 and 56 hours and where the central airport hub is New York (17) (Table 3). If the time bound is between 52 and 36 hours, then generally one airplane and more than two airport hubs are required. When we reduce T to 32 hours or fewer, more than one airplane is needed.

Also, as evident from Table 3, when we increase p for each time bound level, generally the additional hub is opened as a ground hub because each additional airport hub can lead to an increase

Table 3
Results on CAB Data Set.

T	p	# Flights	# Airplanes	A. Hubs	G. Hubs (allocated A. Hubs)	CPU
60	2	2	1	17,8	–	0.06
60	3	2	1	17,8	18(17)	0.23
60	4	2	1	17,22	7(17), 10(17)	0.63
56	3	2	1	17,8	23(8)	0.16
56	4	2	1	17,8	3(17), 23(8)	0.41
56	5	2	1	17,8	10(8), 12(8), 23(8)	0.73
52	3	3	1	17,10,22	–	0.16
52	4	3	1	17,8,22	16(17)	1.76
52	5	2	1	17,8	1(17), 22(8), 23(8)	0.58
48	3	3	1	17,13,22	–	0.13
48	4	3	1	17,13,22	1(13)	3.46
48	5	3	1	17,13,22	11(13), 14(17)	2.71
44	5	4	1	17,13,19,23	14(13)	10.05
44	6	4	1	17,12,13,23	8(13), 24(13)	3.4
44	7	4	1	17,12,13,23	8(13), 16(13), 24(13)	6.64
40	5	5	1	17,1,11,12,23	–	2.33
40	6	5	1	17,1,12,15,23	8(15)	8.81
40	7	5	1	17,1,12,15,23	8(15),13(1)	11.73
36	6	5	1	17,1,11,12,23	10(1)	7.51
36	7	5	1	17,11,12,16,23	8(11), 14(16)	7.67
36	8	5	1	17,1,11,12,23	8(11), 10(1), 16(1)	43.54
32	7	7	2	17,11,12,13,14,23	8(11)	197.15
32	8	7	2	17,10,11,12,23,24	8(11), 15(11)	493.92
32	9	7	2	17,11,12,13,14,23	8(11), 18(17), 22(12)	224.24

Table 4
Results of different central airport hubs, TR Data Set.

T	p	Ankara (6)			Istanbul (34)			Izmir (35)			Kayseri (38)			Elazig (23)		
		# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU
24	3	2	1	0.26	2	1	0.36	3	1	0.39	3	1	0.15	2	1	0.55
24	4	2	1	0.41	2	1	0.43	3	1	4.54	3	1	0.75	2	1	0.62
24	5	2	1	0.43	2	1	0.66	3	1	4.11	3	1	0.96	2	1	1.29
21	5	2	1	0.81	5	2	13.39	4	1	37	4	1	2.99	3	1	2
21	6	2	1	0.49	4	1	19.67	4	1	8.62	4	1	2.98	3	1	2.04
21	7	2	1	0.68	4	1	21.12	4	1	224.08	4	1	2.54	3	1	1.82
18	6	6	2	59.65	7	2	454.41	7	2	445.31	6	2	66.11	6	2	68.88
18	7	6	2	123.52	7	2	2662.24	7	3	2361.68	6	2	104.17	6	2	78.09
18	8	6	2	85.16	7	2	2122.34	7	3	2985.12	6	2	96.1	6	2	158.18

in the number of airline segments as the objective function of the mathematical model minimizes it.

Decreasing the time bound and increasing the number of hubs to be opened generally increases the CPU time. For the TR data set, if the time bound is between 24 and 19 hours, the model is solved within a few seconds for all instances. If it is between 18 and 14 hours, the CPU time requirement is within two hours except in two instances, $T = 15$ and $p = 9$ and $T = 14$ and $p = 9$. When the time bound is between 60 and 48 hours in the CAB data set, the problem is solved within a few seconds. If it is between 44 and 32 hours, all instances are solved within 10 minutes.

We also analyzed the effect of a different central airport hub location on the results of the RSS model for both data sets (Table 4). In Turkey, we chose Istanbul (34; northwestern Turkey) and Izmir (35; western Turkey) as central airport hubs due to the high amount of demand; Istanbul is the country's largest city and Izmir is the third largest. We also selected Kayseri (38) and Elazig (23) because of their locations (central and eastern Turkey, respectively). We varied the time bounds and the number of hubs to be located for each possible central airport hub location. As expected, the cities' geographical positions directly affects the results. For example, for some p values, the problem becomes infeasible when the central airport hub is not Ankara. When $T = 24$ and $p = 2$, if the central airport hub is Ankara, Istanbul or Elazig, there exists a solution; if it is Izmir or Kayseri, the problem is infeasible because the time bound cannot be satisfied with two hubs. Because Izmir is located in the far west of Turkey, cities in the east cannot be covered with only two airport hubs. Interestingly, although Kayseri is in the center of Turkey, two airport hubs are also not enough to cover all o-d pairs in Turkey because if a second airport hub is opened in the west, then cities in the east cannot be reached on time, and if it in the east, then cities in the west cannot be covered on time. To satisfy the time bound for Kayseri, at least three hubs are required; therefore, having the central airport hub in the center of the country may not be as efficient as it might seem.

When we compare the results, we see that the objective function values (the number of airline links) for the Istanbul, Izmir and Kayseri cases are generally higher than for Ankara and Elazig. This finding indicates that if the central airport hub is located on one side of the country (such as in Istanbul or Izmir) or in the center of the country (such as in Kayseri), more flights are necessary to ensure coverage of the whole country. On the other hand, cities located near the center, but not exactly in the center (such as Ankara and Elazig) are more advantageous for being a central airport hub in terms of the number of flights.

We also explored different options for the central airport hub for the CAB data set (Table 5). We first chose Los Angeles (12), which is the second-biggest city in the US in terms of population. We also chose Kansas City (11), because it is very near the center of the US. Additionally, we considered Memphis (13) and Cincinnati (5) as the central airport hub because several cargo companies are headquartered there.

Table 5
Results for different central airport hubs, CAB data set.

T	p	New York (17)			Los Angeles (12)			Kansas City (11)			Memphis (13)			Cincinnati (5)		
		# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU	# Flights	# Airplanes	CPU
60	2	2	1	0.06	2	1	0.1	2	1	0.05	2	1	0.05	2	1	0.06
60	3	2	1	0.23	2	1	0.78	2	1	0.21	2	1	0.2	2	1	0.23
60	4	2	1	0.63	2	1	5.9	2	1	0.13	2	1	0.16	2	1	0.2
48	4	3	1	3.46	3	1	8.73	3	1	1.74	3	1	0.85	2	1	0.18
48	5	3	1	2.71	3	1	7	3	1	2.28	3	1	2.76	2	1	0.17
48	6	3	1	2.23	3	1	7.01	3	1	1	3	1	0.9	2	1	0.2
36	6	5	1	7.51	6	2	148.56	5	1	8.04	6	1	17.4	5	1	2.07
36	7	5	1	7.67	6	2	159.36	5	1	9.02	6	2	47.46	5	1	3.84
36	8	5	1	43.54	5	1	87.18	5	1	7.07	6	2	17.3	5	1	3.71

Table 6
Coverage Percentages of O-D Pairs in TR Data Set for T = 19, p = 6.

	Ankara	Istanbul	Izmir	Kayseri	Elazig
# Flights	4 flights	6 flights	6 flights	5 flights	4 flights
# Airplanes	1 airplane	2 airplanes	2 airplanes	2 airplanes	1 airplane
Service Time	Percentage of Coverage				
18	98.3	98.09	**99.78**	96.45	97.84
16	88.03	84.29	**91.88**	80.46	81.48
14	67.22	61.94	**67.35**	58.24	58.55
12	**41.42**	38.77	37.96	35.19	33.83
10	**20.43**	18.55	17.41	17.13	15.74

We varied the time bound and the number of hubs to be opened for these five central airport hub locations, as we did in the New York case. As evident from the table, the results differ markedly from each other, especially for the tight time bounds, because of their locations in different regions. New York is located in the northeastern US, Los Angeles is in the southwest and Kansas City in the center. Memphis and Cincinnati are located in the central eastern portion of the US.

For a loose time bound such as 60 or 48 hours, the results are generally the same for each case. If we tighten the time bound to 36 hours, again for New York, Kansas City and Cincinnati, the results are the same, but for Los Angeles, more airplanes and flights are required because of how far west it is located and because most of the cities in the CAB data set are located in the east. The Memphis case also requires more airplanes and flights, for a similar reason as the Kayseri case in the TR data set.

The results of the TR and CAB data sets indicate that if the central airport hub is in the east or west of the country, the airplane first travels to an airport hub located on the other side of the country, which is generally farthest from the central one. If the central airport hub is in the central part of the country, the airplane first travels to one side (west or east) of the country, then to the other side and then returns to the central airport hub. The results for both data sets also show that when the central airport hub is more centrally located, CPU time generally decreases.

While analyzing the outputs for both data sets, we observed that although all o-d pairs are covered within the fixed time bound T, some are covered within a bound far shorter than the time bound T over the proposed network. We now provide a different analysis where we compare the resulting networks based on their "percentage of coverage" performance, which we define as the percentage of the whole demand served within different (smaller than T) time bounds. In the TR data set, we analyzed one instance ($T = 19$, $p = 6$) for five central airport hub locations. For this analysis, after calculating the service time for every o-d pair, we computed the percentage of coverage based on service time (Table 6).

The analysis indicates that actually more than half of the o-d pairs are covered within 14 hours, regardless of where the central airport hub is located. When we compare the results, we see that Izmir has the highest service level percentage for $T = 18$, 16 and 14. When the time bound is equal to 12 and 10 hours, we see that Ankara has the highest coverage. Generally, the coverage percentages for Ankara, Izmir and Istanbul are more than for Kayseri and Elazig. However, we should note that the Istanbul, Izmir and Kayseri cases have two airplanes and the Ankara and Elazig cases just one. Having two airplanes directly increases service level percentages; therefore, the Istanbul, Izmir and Kayseri cases have more advantages compared to the Ankara and Elazig cases. Nevertheless, Ankara has the second-highest service level percentage when the time bound is between 18 and 14 hours, and the highest when T is between 12 and 10 hours. Therefore, in terms of service level percentage, Ankara is the most advantageous city ($T = 19$, $p = 6$).

We also compare the service level percentages of the five central airport hub locations in the CAB data set for $T = 48$ and $p = 4$ (see Table 7).

Based on the results shown in Table 7, nearly 70% of all o-d pairs are covered within 32 hours for each case. When the service level is between 44 and 36 hours, the Cincinnati and Kansas City cases have the highest coverage percentages. If the service level is tightened to fewer than 36 hours, Cincinnati and New York have the highest percentages. Although New York is located on the eastern edge of the country, when we decrease the time bound, its service level percentage is generally higher than the other cases percentages; this finding is related to the high number of cities close to it. This situation is also valid for Cincinnati, but not for Kansas City, Memphis or (and especially) Los Angeles. The Cincinnati case generally gives the highest coverage percentages for this instance ($T = 48$, $p = 4$).

5.4. Computational analysis of the subgradient heuristic algorithm

In this section, the solution obtained from the subgradient based heuristic algorithm will be compared with the optimal solution in order to evaluate the quality of the proposed solution approaches in terms of the optimality gap and CPU time.

When we apply the proposed subgradient algorithm for both TR and CAB data set, for most of the instances, the gap between lower and upper bounds is really high. The reason for this can be

Table 7
Coverage percentages of O-D Pairs in CAB Data Set for T = 48, p = 4.

	New York	Los Angeles	Kansas City	Memphis	Cincinnati
# Flights	3 flights	3 flights	3 flights	3 flights	2 flights
# Airplanes	1 airplane	1 airplane	1 airplane	1 airplane	1 airplane
Service Time	Percentage of Coverage				
44	93.67	98	98	97.33	**99**
40	89	89.33	**95**	90	94
36	80.67	78	82.33	77	**87.33**
32	69	63	67.67	62	**76.33**
28	54.33	44	49	45.67	**59**
24	40	29.33	31	30	**41.67**
20	**28**	17	18	18	27.67

Table 8
Comparison between Optimal and Heuristic Solutions (1 iteration) on the TR Data Set.

T	p	# Fl	Optimal Solution		Heuristic Solution		Gap	Time Imp.
			Obj. Func.	CPU	Obj. Func.	CPU		
20	4	2	3	0.9	3	0.49	0.00	45.56
20	5	5	3	1.27	3	0.36	0.00	71.65
20	6	6	3	1.36	3	0.76	0.00	44.12
19	5	4	5	2.08	7	1.14	40.00	45.19
19	6	6	4	5.73	6	0.53	50.00	90.75
19	7	6	4	1.3	6	1.89	50.00	−45.38
18	6	5	6	59.65	6	1.42	0.00	97.62
18	7	6	6	123.52	7	2.8	16.67	97.73
18	8	5	6	85.16	6	35.27	0.00	58.58
17	6	1	7	23.83	7	22.92	0.00	3.82
17	7	5	7	172.77	7	7.72	0.00	95.53
17	8	6	7	494.23	7	22.1	0.00	95.53
16	6	1	9	74.22	9	69.2	0.00	6.76
16	7	2	9	822.47	9	297.2	0.00	63.86
16	8	4	8	1731.65	9	844.79	12.50	51.21
15	8	1	10	1210.64	10	1459.24	0.00	−20.53
15	9	3	10	14987.67	11	5270	10.00	64.84
14	8	2	11	990.34	11	541.36	0.00	45.34
14	9	6	10	18659.45	12	641.79	20.00	96.56
Average							10.48	53.09

related to the objective function. However, since the gaps are really high, the optimal solution cannot be found by using the subgradient algorithm. Instead, we can consider the solution of the restricted problem as a heuristic solution and the upper bounds that we obtained from the restricted problem as the objective function value of the heuristic solution.

First, we show the results of TR data set by considering different T (time bound) and p (number of hubs to be opened) pairs. In Table 8, we compared the optimal solutions with the heuristics solutions obtained from the proposed subgradient based heuristic algorithm with only 1 iteration. The first three columns of the table indicate the parameters: the time bound, the number of hubs to be opened and the number of fixed locations. The next four columns show the objective function values and CPU times for both the optimal solution and heuristic solution, respectively. Finally, the last two columns show the optimality gap and time improvement in percentages, respectively.

Except 7 out of all 19 instances, the upper bounds found at the first iteration of the heuristic algorithm give the optimal solution and the average optimality gap and the worst optimality gap for these 13 instances are 10.48% and 50%, respectively. This indicates that the heuristic method can be considered as a good solution approach for this problem in terms of solution quality. Also, it performs even better when we consider CPU times. As evident from Table 8, except 2 out of 19 instances, the CPU times improve very well. The average time improvement is equal to 53.09%.

In Table 9, we compare heuristic algorithm with different maximum number of iterations; 1, 3, 5, 10, 20, 50 and 100 iterations. In the table, the first column indicates the number of iterations and in the next two columns, we report the average optimality gap and the average solution time improvement in percentages, respectively.

Table 9
The heuristic algorithm results with different number of iterations (TR Data Set).

# Iterations	Average	
	Gap	Time Imp.
1	10.48	53.09
3	8.90	−29.79
5	8.25	−85.99
10	8.25	−178.39
20	8.25	−406.47
50	8.25	−942.11
100	8.25	−1770.92

As it can be seen from Table 9, if the number of iterations is increased to three, the average optimality gap is reduced to 8.90%. However, with three iterations, the heuristic algorithm becomes slower than solving the formulation. When the number of iterations is equal to five or more, the average optimality gap is 8.25%, but the heuristic algorithm performs much worse than solving the formulation in terms of solution time. We can conclude that when we increase the number of iterations, the reduction on the average optimality gap is very small compared to the increase on the solution time. Thus, we propose to use the heuristic algorithm with one iteration.

Table 10
Comparison between Optimal and Heuristic Solutions (1 iteration) on the CAB Data Set.

T	p	# Fl	Optimal Solution		Heuristic Solution		Gap	Time Imp.
			Obj. Func.	CPU	Obj. Func.	CPU		
40	5	1	5	2.33	5	3.62	0.00	−55.36
40	6	2	5	8.81	5	10.53	0.00	−19.52
40	7	5	5	11.73	6	2.16	20.00	81.59
38	5	1	5	6.24	5	2.69	0.00	56.89
38	6	2	5	12.15	5	9.92	0.00	18.35
38	7	4	5	43.87	5	2.55	0.00	94.19
36	6	2	5	7.51	5	7.22	0.00	3.86
36	7	4	5	7.67	6	13.07	20.00	−70.40
36	8	6	5	43.54	7	5.07	40.00	88.36
34	6	3	6	39.76	6	4.37	0.00	89.01
34	7	4	6	63.54	6	6.3	0.00	90.08
34	8	6	6	179.51	7	5.64	16.67	96.86
32	7	5	7	197.15	7	1.90	0.00	99.04
32	8	5	7	493.92	8	48.57	14.29	90.17
32	9	6	7	224.24	7	19.21	0.00	91.43
30	7	1	8	99.7	8	32.71	0.00	67.19
30	8	4	7	842.04	8	11.38	14.29	98.65
30	9	6	7	471.27	7	13.3	0.00	97.18
Average							6.96	56.53

Table 11
The heuristic algorithm results with different number of iterations (CAB Data Set).

# Iterations	Average	
	Gap	Time Imp.
1	6.96	56.53
3	6.16	−0.25
5	6.16	−60.39
10	6.16	−139.67
20	6.16	−303.21
50	6.16	−783.28
100	6.16	−1580.40

Next, we considered CAB data set with the same approach as we did for the TR data set. Initially, we applied the subgradient algorithm with only 1 iteration in order to observe its performance. Again, due to the high gaps obtained by the subgradient algorithm, we consider the solution of the restricted problem (upper bound) as a heuristic solution as we did for the TR data set. The optimal and heuristic solutions with only 1 iteration are compared in Table 10.

For 12 out of 18 instances, the heuristic method gives the optimal solution. In terms of CPU time, the heuristic method performs well. For all instances expect 3, CPU times improve and the average time improvement is 56.53%, which is higher than the one obtained by the TR data set.

For the CAB data set, we also compare the heuristic algorithm with different number of iterations. The average results are depicted in Table 11.

The results on Table 11 indicate that with 3 iterations, the average optimality gap is reduced to 6.16%, whereas the average solution time is very close to the one of solving the formulation. When the number of iterations is increased even further, the average optimality gap does not decrease, while the average solution time increases rapidly. Similar to the results obtained for the TR data set, the heuristic algorithm with only 1 iteration is the only solution technique that has a solid improvement on the CPU time. Therefore, it is the best version of the heuristic algorithm.

In general, for both data sets, this heuristic algorithm with only 1 iteration performs very well in terms of solution quality and CPU time. It yields very low optimality gap despite the structure of the objective function and it solves the problem very fast compared to the mixed integer programming model.

6. Conclusions and future work

In this study, we introduce the hierarchical multimodal hub covering problem over a service network configuration. We propose to configure the upper level as a ring structure mainly because upper levels usually require more sophisticated transportation vessels leading to more costs. In the top layer, airplanes pick up and deliver cargoes with tours between airport hubs. Therefore, in addition to routing decisions, scheduling decisions are also included in this problem.

We develop a mixed-integer programming formulation and propose some valid inequalities to strengthen the model. Also, an alternative solution approach based on Lagrangian relaxation is developed in order to solve the problem instances with tighter time bounds in a reasonable time. The proposed approach is directly related to a subgradient algorithm. Initially, the problem is attempted to solve by using a subgradient algorithm, but due to high gaps between lower bounds obtained from the relaxed problem and upper bounds obtained from the restricted problem, we propose a heuristics method based on the subgradient approach, in which the upper bounds obtained from the restricted problems in the subgradient algorithm are considered as a heuristic solution to the original problem.

We conduct comprehensive computational studies for the proposed mathematical model on both CAB and Turkish network data sets. First of all, we solve the proposed problem to optimality and present some computational analysis to observe the effects of some key parameters such as the time bound, the number of hubs to open and the central airport hub. After, we conduct another computational study over TR and CAB data sets in order to evaluate the performance of the subgradient based heuristic method with the proposed formulation. Based on the computational results, we conclude that the heuristic approach with only 1 iteration works very well in terms of the solution quality (the optimality gap) and the CPU time for both data sets.

As stated earlier, we aim to cover all o-d pairs in a given time bound (T). The computational analysis show that most o-d pairs are covered within a bound far lower than T. Further, when we compare different locations for the central airport hub for the

RSS model, we find that cities located near the center of the country (Ankara for TR and Cincinnati for CAB), as distinct from cities on the edges or exactly in the center, are more favorable in terms of the number of airplanes, airline segments and coverage percentages.

In future research, flow and variable transportation cost issues could be included in the proposed problem settings to observe their effects on the solutions. Vehicle capacity could also be added into the problem if flow is considered. With a capacity constraint, more vehicles might be required to deliver the cargoes within the time bound. Cargo delivery with "pick up and deliver together" service could also be considered, which would decrease package travel time. Finally, to solve the proposed problem, more sophisticated solution techniques such as branch and cut approach can be developed.

Acknowledgements

The authors thank two anonymous reviewers and the Editor for their valuable comments on an earlier version of this paper that resulted in improved content and exposition.

References

Alumur, S., Kara, B.Y., 2008. Network hub location problems: the state of the art. Eur. J. Oper. Res. 190 (1), 1–21.
Alumur, S., Kara, B.Y., 2009. A hub covering network design problem for cargo applications in turkey. J. Oper. Res. Soc. 60 (10), 1349–1359.
Alumur, S.A., Kara, B.Y., Karasan, O.E., 2012a. Multimodal hub location and hub network design. Omega (Westport) 40 (6), 927–939.
Alumur, S.A., Yaman, H., Kara, B.Y., 2012b. Hierarchical multimodal hub location problem with time-definite deliveries. Transp. Res. Part E: Log. Transp. Rev. 48 (6), 1107–1120.
Beasley, J. E., 2012. Or-library: Hub location. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/phubinfo.html. Accessed: 2012-09-17.
Campbell, J.F., 1994. Integer programming formulations of discrete hub location problems. Eur. J. Oper. Res. 72 (2), 387–405.
Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., 2002. Hub location problems. Facility Location: Applic. Theory 1, 373–407.
Campbell, J.F., O'Kelly, M.E., 2012. Twenty-five years of hub location research. Transp. Sci. 46 (2), 153–169.
Çetiner, S., Sepil, C., Süral, H., 2010. Hubbing and routing in postal delivery systems. Ann. Oper. Res. 181 (1), 109–124.
Contreras, I., Fernández, E., Marín, A., 2009. Tight bounds from a path based formulation for the tree of hub location problem. Comput. Oper. Res 36 (12), 3117–3127.
Contreras, I., Fernández, E., Marín, A., 2010. The tree of hubs location problem. Eur. J. Oper. Res. 202 (2), 390–400.
De Camargo, R.S., de Miranda, G., Løkketangen, A., 2013. A new formulation and an exact approach for the many-to-many hub location-routing problem. Appl. Math. Model. 37 (12), 7465–7480.
Kara, B.Y., Taner, M.R., 2011. Hub location problems: The location of interacting facilities. Springer, New York, NY, pp. 273–288.
Kara, B.Y., Tansel, B.Ç., 2001. The latest arrival hub location problem. Manag. Sci. 47 (10), 1408–1420.
Liu, J., Li, C.-L., Chan, C.-Y., 2003. Mixed truck delivery systems with both hub-and-spoke and direct shipment. Transp. Res. Part E: Log. Transp. Rev. 39 (4), 325–339.
Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. J. ACM (JACM) 7 (4), 326–329.
Nagy, G., Salhi, S., 1996. Nested heuristic methods for the location-routeing problem. J. Oper. Res. Soc. 1166–1174.
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. Eur. J. Oper. Res. 177 (2), 649–672.
O'Kelly, M.E., 1986a. Activity levels at hub facilities in interacting networks. Geogr. Anal. 18 (4), 343–356.
O'Kelly, M.E., 1986b. The location of interacting hub facilities. Transp. Sci. 20 (2), 92–106.
O'Kelly, M.E., 1987. A quadratic integer program for the location of interacting hub facilities. Eur. J. Oper. Res. 32 (3), 393–404.
O'Kelly, M.E., Lao, Y., 1991. Mode choice in a hub-and-spoke network: a zero-one linear programming approach. Geogr. Anal.
de Sá, E.M., de Camargo, R.S., de Miranda, G., 2013. An improved benders decomposition algorithm for the tree of hubs location problem. Eur. J. Oper. Res. 226 (2), 185–202.
Sahraeian, R., Korani, E., 2010. The hierarchical hub maximal covering problem with determinate cover radiuses. In: Proceedings of the 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, pp. 1329–1333.
Smilowitz, K.R., Daganzo, C.F., 2007. Continuum approximation techniques for the design of integrated package distribution systems. Networks 50 (3), 183–196.
Tan, P.Z., Kara, B.Y., 2007. A hub covering model for cargo delivery systems. Networks 49 (1), 28–39.
Wasner, M., Zäpfel, 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.
Yaman, H., 2009. The hierarchical hub median problem with single assignment. Transp. Res. Part B: Methodol. 43 (6), 643–658.
Yaman, H., Kara, B.Y., Tansel, B.Ç., 2007. The latest arrival hub location problem for cargo delivery systems with stopovers. Transp. Res. Part B: Methodol. 41 (8), 906–919.
Yaman, H., Karasan, O.E., Kara, B.Y., 2012. Release time scheduling and hub location for next-day delivery. Oper. Res. 60 (4), 906–917.