Capacitated Multiple Allocation Hub Location with Service Level Constraints for Multiple Consignment Classes

Sachin Jayaswal[a,*], Navneet Vidyarthi[1]

[a] *Indian Institute of Management, Vastrapur, Ahmedabad, Gujarat 380 015, India.*
Ph: +91-79-6632-4877, Fax: +91-79-6632-6896, E-mail: sachin@iimahd.ernet.in
[b] *Department of Decision Sciences and Management Information Systems, John Molson School of Business, Concordia University, Montreal, QC, H3G 1M8, Canada.*
Ph: +1-514-848-2424-x2990, Fax: +1-514-848-2424, E-mail: navneetv@jmsb.concordia.ca

Abstract

Hub-and-spoke systems have wide applications ranging in airline transportation, freight transportation, urban traffic, postal delivery, telecommunications and distribution in supply chains. These systems are usually characterized by stochastic demand and congestion, which adversely affect the quality of service to customers. These systems are further characterized by different classes of customers who need different levels of service. In this paper, we study the problem of hub-and-spoke network design under conditions wherein customer demands are stochastic and consignments from one class are served at hubs with priority over those from the other class to maintain the different service levels required by them. We present a model for designing a capacitated multiple allocation hub location problem with a service level constraint, defined using the distribution of time spent at hubs, for each priority class. The model seeks to determine the hub-and-spoke network design at the minimum total cost, which includes the total fixed cost of equipping open hubs with sufficient processing capacity and the variable transportation costs, subject to a service level constraint for each consignment class. The network of hubs, given their locations, is thus modeled as spatially distributed preemptive priority M/M/1 queues. The problem is challenging to solve, especially in absence of any known analytical expression for the sojourn time distribution of low priority customers in a preemptive priority M/M/1 queue. To resolve this problem, we exploit the concavity of the sojourn time distribution of low priority consignments to eliminate the non-linearity in their service level functions at the expense of a large number of tangent hyperplanes, which are determined numerically using matrix geometric method. The problem is solved to optimality using a cutting plane method. Computational results based on the US Civil Aeronautics Board (CAB) data are provided. The results show that an explicit account for service level constraints at hubs may result in a significantly different network configuration. Further, it is interesting to note that increasing the fraction of consignments that receive priority in service or/and that have a lower value of the maximum threshold on sojourn time may not necessarily increase the total cost of the network design.

Keywords: Hub-and-spoke network design, service level, priority queue, cutting plane method, matrix geometric method

[*] Corresponding author

1. Introduction

Hub-and-spoke networks have wide applications in areas ranging from airline passenger transportation, freight transportation, urban public transport, postal delivery to telecommunications. Hubs are special facilities that serve as switching, transshipment or sorting points in distribution systems. They exploit the economies of scale arising from concentrating traffic at certain nodes, called hubs, from/to several origins/destinations, instead of serving each origin-destination pair directly. Flows from the same origin with different destinations in a hub-and-spoke network are, therefore, consolidated on their route at the hub where they are combined with flows that have different origins but the same destination (Alumur and Kara, 2008; Campbell and O'Kelly, 2012). In multi-hub networks, traffic concentrated at a hub is directed to a second hub, which distributes it to the final destinations, thereby exploiting the economies of scale on the inter-hub flows.

Examples of applications of hub-and-spoke networks include companies such as FedEx, UPS, DHL, and the United States Postal Service (USPS), which receive and deliver millions of packages every day. The configuration of the hub-and-spoke network plays a central role in the cost-efficient distribution of such large volumes of packages transported between many different origin-destination points. For example, FedEx has strategically located its hub at Memphis in order to provide overnight service to the US and to serve 95% of the global economy (220 countries on six continents) customers within 24-48 hours[1].

A hub-and-spoke network design involves the optimal location of hubs and allocation of non-hub origin and destination points to hubs. The criterion for optimality is generally minimization of total cost of transportation (in p-hub median problem (O'Kelly and Bryan, 1998; Yaman, 2009)) or the total fixed cost of installing facilities at hubs plus the variable transportation cost (in hub location problem with fixed costs), given an estimate of volume of traffic among various origin and demand pairs. Other categorizations of hub-and-spoke network design problem (p-hub center problem; and hub covering problem) based on different optimization criteria are discussed by Alumur and Kara (2008). Irrespective of its optimization criterion, a hub-and-spoke network design problem is NP-hard, which combines a facility location problem with network design, and the difficulty arises from the inherent interrelation between the two. The problem has, therefore, received great attention ever since it was first reported in the literature by O'Kelly (1986a). Campbell et al. (2002),

[1] http://www.memphisairport.org/notes/mem_2010_august_hub.htm

Alumur and Kara (2008), and Campbell and O'Kelly (2012) provide excellent reviews of the related literature.

In this paper, we consider hub-and-spoke network design in air freight transportation in which both the fixed cost of installing facilities at hubs as well as the variable transportation costs are important cost components. Our focus in this paper is, therefore, on hub location problem with fixed costs. O'Kelly (1992b) introduced the hub location problem with fixed costs, in which the number of hubs to open is a decision variable as opposed to a p-hub median problem in which the number of hubs to open is given. The model he proposed assumes single allocation in the sense that the traffic from an origin node cannot be split among 2 or more hubs. Similarly, the traffic to a destination node cannot be combined from two or more hubs. Following O'Kelly's work, several papers have reported either a different formulation or a different solution approach to the single allocation hub location problem with fixed costs (Campbell, 1994b; Abdinnour-Helm and Venkataramanan, 1998; Labbe and Yaman, 2004; Cunha and Silva, 2007; Alumur et al., 2009, 2012; Contreras et al., 2012). Simultaneously, there have been papers that have reported the multiple allocation version of the problem, in which the traffic from an origin node is allowed to be split among 2 or more hubs, and likewise the traffic to a destination node is allowed to be combined from two or more hubs (Campbell, 1994b; Boland et al., 2004; Hamacher et al., 2004; Marin, 2005b; Racunica and Wynter, 2005; Marin et al., 2006; Canovas et al., 2007; Alumur et al., 2012).

The hub-and-spoke network design problem, when solved purely as a cost minimization problem, however, tends to overload the resulting hubs in absence of any capacity constraints on the hubs in the model. In reality, the hubs have a finite limit on the amount of flows they can handle. An airline company, for example, may have a limit on the amount of consignments it can sort at a hub before distributing them. Thus, the solution of a pure cost minimization hub location problem, when implemented, is likely to cause traffic delays due to congestion in presence of a finite capacity at the hubs. Explicitly modeling such a limit on the amount of flows a hub can handle leads to a capacitated hub location problem. Aykin (1994), Ernst and Krishnamoorthy (1999), Labbe et al. (2005), Costa et al. (2008), and Correia et al. (2010), among others, have studied the single allocation version, while Ebery et al. (2000), Boland et al. (2004), and Marin (2005a) have dealt with the multiple allocation version of the capaciatated hub location problem.

Although flights follow a schedule, they are very often subject to delays either at the origin airports or during the flight, making their arrivals at the hubs non-deterministic. The service rates of the hubs are also variable due to vagaries of weather and other operating conditions at a facility as complex as a hub airport (Marianov and Serra, 2003). The use of explicit limits on the load at a hub in presence of such uncertainties in the arrival and the service rates is not enough to prevent congestion at a hub and the resulting traffic delays, even when the traffic at a hub is less than its capacity. Elhedhli and Hu (2005) and Elhedhli and Wu (2010) attempt to avoid such hub congestion in their solution by imposing an increasing penalty on each incremental unit of traffic flow at a hub in a single allocation model, while Camargo et al. (2009) do the same in a multiple allocation model. To the best of our knowledge, Marianov and Serra (2003) is the only work that tries to avoid congestion at hubs by explicitly imposing a probabilistic constraint (based on an $M/D/c$ queueing model of a hub) on the length of waiting flights at a hub.

In this paper, we first present a multiple allocation hub location problem with fixed costs and service level constraints, defined as the minimum probability of servicing a waiting flight at a hub within a predefined threshold time. This is a straight forward extension of Marianov and Serra (2003) based on an $M/M/1$ queueing model for the congestion at hubs if all the traffics are homogeneous. However, clearly, all flights with consignments arriving at any of the hubs of FedEx or UPS are not homogeneous, for some of them carry normal 1-week-delivery (henceforth called 'regular') consignments while others carry overnight (henceforth called 'express') consignments. The two classes of consignments need different treatments at the hubs, with the express consignments deserving a priority over the regular consignments. Our second model takes such heterogeneous customers into account by imposing a different service level constraint for each consignment class. We model this by considering hubs as preemptive priority $M/M/1$ queues. Further, we consider the service capacity to be a variable in the model. The resulting Mixed Integer Programming (MIP) problem with probabilistic constraints is challenging to solve, especially in absence of any known closed form expression for the service level constraint for low priority customers (regular consignments). To resolve this problem, we exploit the concavity of the sojourn time distribution of low priority consignments to eliminate the non-linearity in their service level function, but at the expense of a large number of tangent hyperplanes, determined numerically using matrix geometric method. The problem is solved efficiently using a cutting plane method.

The remainder of the paper is organized as follows. In Section 2, we define the modelling framework, followed by a discussion on the solution methodology in Section 3. Section 4 presents our computational study and discussion of results. The paper concludes with a summary of results and a discussion on future research in Section 5.

2. Model Formulation

Let N be the set of all nodes that exchange traffic, and also represent the set of potential hub nodes. We use i and j as indices for the origin and destination nodes, while k and m as indices for potential hub locations. Further, let F_k be the amortized cost of establishing a hub at node $k \in N$. Define λ_{ij} as the amount of traffic (number of flights) to be routed from the origin node $i \in N$ to destination $j \in N$. The transportation cost per unit of traffic from node i to node j routed via hubs k and m, in that order, is given by $C_{ijkm} = C_{ik}+\delta C_{km}+C_{mj}$, where C_{ik} is the collection cost (per unit of traffic) from origin node i to hub k; C_{mj} is the distribution cost (per unit of traffic) from hub m to destination node j; C_{km} is the inter-hub transfer cost (per unit of traffic), and $\delta \in (0,1)$ is the discount factor, reflecting economies of scale in inter-hub flows. Let the binary variable $z_k = 1$ represent location of a hub at node k; 0 otherwise. Variable x_{ijkm} represents the fraction of total traffic from node i to node j routed via hubs located at nodes k and m, in that order. The problem is to optimally decide appropriate nodes, $k, m \in N$, to locate hubs, and path(s) between all origin and destination pairs,(i, j), such that every path traverses one or more hubs to benefit from the inter-hub flow discounts. With these notations, we first present the strongest known formulation of the *Uncapacitated Multiple Allocation Hub Location Problem* (UMAHLP), proposed by Hamacher et al. (2004), since our proposed model builds onto it:
[UMAHLP]:

$$\min \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{m\in N} C_{ijkm}\lambda_{ij}x_{ijkm} + \sum_{k\in N} F_k z_k \tag{1}$$

$$\text{s.t.} \quad \sum_{k\in N}\sum_{m\in N} x_{ijkm} = 1 \qquad \forall i,j \in N \tag{2}$$

$$\sum_{m\in N} x_{ijkm} + \sum_{m\in N\setminus\{k\}} x_{ijmk} \leq z_k \qquad \forall i,j,k \in N \tag{3}$$

$$x_{ijkm} \geq 0 \qquad \forall i,j,k,m \in N \tag{4}$$

$$z_k \in \{0,1\} \qquad \forall k \in N \tag{5}$$

The objective function (1) minimizes the sum of total transportation costs between all the origin-destination node pairs and the amortized cost of establishing all the hubs. Constraint set (2) requires that the traffic demand between any pair of nodes be completely satisfied. Constraint set (3) prohibits traffic from being routed via any intermediate node that is not a hub. Constraints (4) and (5) are non-negativity and integrality requirements.

Certain applications impose further restriction of at most 2 hubs en route any path from an origin node to a destination node, including origin or destination if either of them itself is a hub. For example, it may not be desirable for a passenger aircraft to stop at more than 2 hubs for small distance flights. Similarly, postal services may require that any mail should not visit more than 2 post offices before its final destination (Marin et al., 2006). Such a restriction is implicitly taken care of by the model (1)-(5) if the transportation costs between different pairs of nodes satisfy the triangle inequality, i.e., $C_{ij} < C_{ik} + C_{kj}$. This is so because any flow routed via hubs k and m that traverses another intermediate hub between them is costlier (incurs additional transportation cost without any additional inter-hub flow discount) than the flow routed directly from hub k to hub m (Marin et al., 2006). However, the transportation costs may not always satisfy the triangle inequality, especially if they are not proportional to distances. In such cases, the restriction of at most 2 hubs on any feasible path from an origin node to a destination node needs to be explicitly imposed through the following additional constraints (Camargo et al., 2009):

$$x_{ijij} \geq z_i + z_j - 1 \qquad \forall i, j \in N \tag{6}$$

$$\sum_{m \in N \setminus \{j\}} x_{ijim} \geq z_i - z_j \qquad \forall i, j \in N \tag{7}$$

$$\sum_{k \in N \setminus \{i\}} x_{ijkj} \geq z_j - z_i \qquad \forall i, j \in N \tag{8}$$

Constraint set (6) restricts any flow from origin node i to destination node j to travel only via the path $i \to i \to j \to j$ if both i and j are hubs. Constraint sets (7) and (8) ensure that any flow from origin node i to destination node j travels only via the path $i \to i \to m \to j$ and $i \to k \to j \to j$, respectively if i or j is a hub.

Model (1)-(8) does not assume any capacity limit at the hubs. In reality, as highlighted in §1, hubs have a finite limit on the amount of flows they can handle. Camargo et al. (2009) extend the UMAHLP model of Hamacher et al. (2004) with a penalty term in the objective function for congestion at hubs. In the following subsection, we extend the capacitated multiple allocation hub location model of Camargo et al. (2009) by capturing finite capacity and the resulting congestion (due to uncertain demand and service times) at hubs using service level constraints.

2.1. Extension to Capacitated System with Service Level Constraints

Following the arguments presented by Marianov and Serra (2003), as highlighted in §1, we assume the arrival of traffic at a hub has considerable variability, and hence is modelled as a random variable. Similarly, due to the variability in the service rate at the hub owing to vagaries in weather or other operational reasons, the service time at a hub is also modeled as a random variable. Thus, each hub can be modeled as a queueing facility, where the mean service rate of hub k, if it is allocated a capacity level $l \in L_k$, is given by $\mu_k = \sum_{l \in L_k} \mu_{kl} z_{kl}$. Here, $z_{kl} = 1$ if node k is designed as hub with capacity level l, and μ_{kl} is the service rate at l^{th} capacity level at hub k. In order to serve each consignment within its promised delivery time, the firm sets its own internal *maximum threshold service time* (τ) for consignments at any hub and a *target service level* ($\alpha \in (0, 1)$), which is the minimum probability with which a consignment at any hub should be served within the maximum threshold service time. Failure to meet the maximum threshold service time at a hub may lead to promised delivery times to customers getting missed, which may result in penalties, either in the form of a discount, partial refund or an expedited delivery (to avoid any further delay) without additional charge to the customer. For example, FedEx offers a money-back guarantee for every U.S. shipment that is even 1 minute late compared to its guaranteed delivery time.[2] If we let W_k denote the total time spent by consignments in the system (waiting in queue + service time) at hub k, then the service level constraint can be expressed as follows:

$$S_k(\tau) = P\{W_k \leq \tau\} \geq \alpha \qquad\qquad \forall k \in N$$

The resulting MIP formulation of the *Capacitated Multiple Allocation Hub Location Problem with Service Level Constraints* (CMAHLP-SLC) is as follows:

[2] http://www.fedex.com/us/services/options/mbg.html

[CMAHLP-SLC]:

$$\min \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{m\in N} C_{ijkm}\lambda_{ij}x_{ijkm} + \sum_{k\in N}\sum_{l\in L_k} F_{kl}z_{kl} \tag{9}$$

$$\text{s.t.} \quad \sum_{k\in N}\sum_{m\in N} x_{ijkm} = 1 \qquad \forall i,j \in N \tag{10}$$

$$\sum_{m\in N} x_{ijkm} + \sum_{m\in N\setminus\{k\}} x_{ijmk} \leq \sum_{l\in L_k} z_{kl} \qquad \forall i,j,k \in N \tag{11}$$

$$\sum_{l\in L_k} z_{kl} \leq 1 \qquad \forall k \in N \tag{12}$$

$$\Lambda_k \leq \sum_{l\in L_k} \mu_{kl}z_{kl} \qquad \forall k \in N \tag{13}$$

$$P\{W_k \leq \tau\} \geq \alpha \sum_{l\in L_k} z_{kl} \qquad \forall k \in N \tag{14}$$

$$x_{ijij} \geq \sum_{l\in L_k}(z_{il} + z_{jl}) - 1 \qquad \forall i,j \in N \tag{15}$$

$$\sum_{k\in N\setminus\{j\}} x_{ijik} \geq \sum_{l\in L_k}(z_{il} - z_{jl}) \qquad \forall i,j \in N \tag{16}$$

$$\sum_{k\in N\setminus\{i\}} x^c_{ijkj} \geq \sum_{l\in L_k}(z_{jl} - z_{il}) \qquad \forall i,j \in N \tag{17}$$

$$x_{ijkm} \geq 0 \qquad \forall i,j,k,m \in N \tag{18}$$

$$z_{kl} \in \{0,1\} \qquad \forall k \in N, l \in L^k \tag{19}$$

Constraint set (11) is the counterpart of (3) in the uncapacitated setting. Constraint set (12) allows a node to be opened as a hub with only one level of capacity. Constraint set (13) is required for the stability of the queueing system at open hubs, where Λ_k is the mean arrival rate of consignments at hub k, given by:

$$\Lambda_k = \sum_i\sum_j\sum_m \lambda_{ij}x_{ijkm} \tag{20}$$

Λ_k in (20) captures only the (collection) flows entering hub k directly from the origin node. It does not capture the (transfer) flows entering hub k via another hub. (20), together with (12)-(14), thus models a capacity restriction at a hub only on the volume of consignments entering it via collection. This makes sense in situations where consignments once processed (e.g., sorted) after collection do not need further processing for distribution (Ebery et al., 2000). However, in situations where the consignments need further processing before distribution, (20) should be modified as (Marin, 2005a; Camargo et al., 2009):

$$\Lambda_k = \sum_i \sum_j \sum_m \lambda_{ij} x_{ijkm} + \sum_i \sum_j \sum_{m \neq k} \lambda_{ij} x_{ijmk} \tag{20-1}$$

Here, the second summation captures the flows entering hub k only via another hub (transfer flows). Constraint set (14) are the internal service level constraints at the hub nodes. The target service level α is set by the management as an internal performance measure.

The term $\sum_{l \in L_k} z_{kl}$ in the right hand side of the (14) ensures that the service level constraint applies only to those nodes that are designated as hubs. Constraint sets (15) - (17) are the the counterparts, in a capacitated setting, of the constraint sets (6) - (8), which model the restriction of at most 2 hubs on any feasible path from an origin node to a destination node. Unlike the uncapacitated model (UMAHLP), (15) is required in a capacitated model even in the presence of transportation costs that satisfy triangle inequalities. This is to circumvent following type of absurd solutions (Marin, 2005a). A consignment from an origin node i to destination node j, when both i and j are hubs, may be routed using any of the three different sets of variables: (a) x_{ijii} corresponding to the route $i \rightarrow i \rightarrow i \rightarrow j$; (b) x_{ijjj} corresponding to the route $i \rightarrow j \rightarrow j \rightarrow j$; and (c) x_{ijij} corresponding to the route $i \rightarrow i \rightarrow j \rightarrow j$. All these three variables represent essentially the same route $(i \rightarrow j)$. However, (a) and (b) have higher associated costs than (c) since they do not involve any inter-hub discount (δ). In absence of constraint set (15), $[CMAHLP - SLC]$ may prefer (a) if hub i has, while hub j does not have, enough spare capacity to meet the service level constraint. Or, it may prefer (b) if hub j has, while hub i does not have, enough spare capacity to meet the service level constraint. Alternatively, it may route the consignment partly via all the above three routes. Obviously, such solutions do not make sense since they associate different costs for essentially the same physical route. We, therefore, explicitly include constraint sets (15) - (17) even in the presence of transportation costs that satisfy triangle inequalities.

If we assume that the rate of flows between different origin node-destination node pairs (i, j) to be independent random variables that follow a Poisson process with mean λ_{ij}, then the aggregate flow rate through hub k, following the superposition of Poisson processes, also follows a Poisson process with a mean given by (20). The service times at the hub will depend on the hub capacity, which is a decision variable. Let L_k be the set of available capacity levels of a candidate hub at node $k \in N$. If the service times at the hub follow an exponential distribution, then each hub can be modeled as an M/M/1 queue[3], where the mean service rate of hub k, if it is allocated a capacity level $l \in L_k$, is given by $\mu_k = \sum_{l \in L_k} \mu_{kl} z_{kl}$. This service rate reflects the server capacity or essentially the units of flow a hub can serve in a given time period. For a hub k, which is modeled as an M/M/1 queue, the service level constraint (14) can be specified as (Gross and Harris, 1998):

$$\sum_{l \in L_k} \mu_{kl} z_{kl} - \Lambda_k \geq \frac{-\ln(1-\alpha)}{\tau} \sum_{l \in L_k} z_{kl} \tag{21}$$

where Λ_k is given by (20). It may be noted here that the presence of the service level constraint (21) makes the queueing system stability constraint (13) redundant, which will, therefore, be omitted in the rest of the paper.

2.2. Extension to Capacitated System with Service Level Constraints for Multiple Consignment Classes

In this section, we extend the model $[CMAHLP-SLC]$ to multiple consignment classes. For simplicity, we assume only two consignment classes, indexed by $c \in \{r, e\}$, for regular (r) and express (e), corresponding respectively to 1 week regular delivery and overnight express delivery services offered by courier companies like FedEx and UPS. Demand from consignment class c for flows between origin node i and destination node j arrives according to a Poisson process with rate λ_{ij}^c. We assume the service times at a hub follow an exponential distribution such that each hub can be modeled as an M/M/1 queue, where the mean service rate of hub k, if it is allocated a capacity level $l \in L_k$, is given by $\mu_k = \sum_{l \in L_k} \mu_{kl} z_{kl}$. Consignments within each class are served on a first-come-first-served (FCFS) basis at a hub. However, express consignments at a hub are given preemptive priority in service over regular consignments. In order to serve each consignment within its promised delivery time, the firm sets its own internal target service time (τ^c) for consignment class (c) at any hub and a *target service level* (α^c). The objective of the firm is to locate the hubs with appropriate capacities and select the routes for all origin-destination pairs via some hubs such that the total network cost is minimized, subject to a separate service level constraint for each consignment class at hubs. We refer to this problem as the *Capacitated Multiple Allocation Hub Location Problem with Multi-class Service Level Constraints* (CMAHLP-MSLC). We

[3] M/M/$\cdot$ queuing model is an abstraction employed to make the problem tractable, especially since our emphasis is more on strategic rather than on operations decisions.

first define the following notations to be used in the model.

Indices:

i, j, k, m : Nodes

k, m : Hub nodes

l : capacity level at hub

c : Consignment class; $c \in \{e, r\}$.

Parameters:

N : Set of all nodes that exchange traffic; $\{i, j, k, m \in N\}$; $N = \{0, 1, 2, ..., |N - 1|\}$.

L_k : Set of all capacity levels at hub k; $\{l \in L_k\}$; $L_k = \{1, 2, ..., |L_k|\}$.

λ_{ij}^c : Rate of flows for consignment class c from origin node $i \in N$ to destination node $j \in N$.

Λ_k^c : Rate of arrival of consignments from class c at hub k.

μ_{kl} : Capacity (processing rate) corresponding to capacity level l at hub k.

μ_k : Capacity (processing rate) installed at hub k.

δ : Inter-hub flow discount; $\delta \in (0, 1)$.

C_{ij} : Transportation cost per unit of direct flow from node $i \in N$ to node $j \in N$.

C_{ijkm} : Transportation cost per unit of flow from node $i \in N$ to node $j \in N$ routed via hubs $k, m \in N$ in that order. $C_{ijkm} = C_{ik} + \delta C_{km} + C_{mj}$.

F_{kl} : Amortized cost of locating a hub with capacity level l at hub k.

τ^c : Maximum threshold on sojourn time (in queue + in service) for consignment class c.

α^c : Target service level for consignment class c at a hub.

W_k^c : Sojourn time (in queue + in service) for consignment class c at hub k.

$S_k^c(\tau^c)$: Service level achieved for consignment class c at hub k, i.e., $P\{W_k^c \leq \tau^c\}$.

Variables:

z_{kl} : 1, if node k is opened as a hub with capacity level l; 0 otherwise.

x_{ijkm}^c : fraction of the flow for consignment class c from origin node $i \in N$ to destination node $j \in N$ that is routed via hubs located at nodes $k, m \in N$ in that order.

The resulting mixed integer programming formulation of the *Capacitated Multiple Allocation Hub Location Problem with Multi-class Service Level Constraints* (CMAHLP-MSLC) is as follows:

[CMAHLP-MSLC]:

$$\min \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{m\in N}\sum_{c\in\{e,r\}} \lambda_{ij}^{c} C_{ijkm} x_{ijkm}^{c} + \sum_{k\in N}\sum_{l\in L_k} F_{kl} z_{kl} \tag{22}$$

$$\text{s.t.} \quad \sum_{k\in N}\sum_{m\in N} x_{ijkm}^{c} = 1 \qquad \forall i,j\in N, c\in\{e,r\} \tag{23}$$

$$\sum_{m\in N} x_{ijkm}^{c} + \sum_{m\in N\setminus\{k\}} x_{ijmk}^{c} \leq \sum_{l\in L_k} z_{kl} \qquad \forall i,j,k\in N, c\in\{e,r\} \tag{24}$$

$$\sum_{l\in L_k} z_{kl} \leq 1 \qquad \forall k\in N \tag{25}$$

$$\sum_{l\in L_k} \mu_{kl} z_{kl} - \Lambda_k^{e} \geq \frac{-\ln(1-\alpha^{e})}{\tau^{e}} \sum_{l\in L_k} z_{kl} \qquad \forall k\in N \tag{26}$$

$$S_k^{r}(\tau^{r}) = P\{W_k^{r} \leq \tau^{r}\} \geq \alpha^{r} \sum_{l\in L_k} z_{kl} \qquad \forall k\in N \tag{27}$$

$$x_{ijkm}^{c} \geq \sum_{l\in L_k} (z_{il} + z_{jl}) - 1 \qquad \forall i,j\in N, c\in\{e,r\} \tag{28}$$

$$\sum_{k\in N\setminus\{j\}} x_{ijik}^{c} \geq \sum_{l\in L_k} (z_{il} - z_{jl}) \qquad \forall i,j\in N, c\in\{e,r\} \tag{29}$$

$$\sum_{k\in N\setminus\{i\}} x_{ijkj}^{c} \geq \sum_{l\in L_k} (z_{jl} - z_{il}) \qquad \forall i,j\in N, c\in\{e,r\} \tag{30}$$

$$x_{ijkm}^{c} \geq 0 \qquad \forall i,j,k,m\in N, c\in\{e,r\} \tag{31}$$

$$z_{kl} \in \{0,1\} \qquad \forall k\in N, l\in L_k \tag{32}$$

The objective function (22) is the total of average flow cost per unit time and the amortized cost of installing capacities at selected hubs. Constraint sets (23) - (25) are counterparts, in a multi-class setting, of constraint sets (10) - (12). Similarly, Constraint sets (28) - (30) are counterparts, in a multi-class setting, of the constraint sets (15) - (17). Constraint sets (26) and (27) are the service level constraints for express and regular consignments, respectively, where Λ_k^{e} and Λ_k^{r} are given by:

$$\Lambda_k^{e} = \sum_{i}\sum_{j}\sum_{m} \lambda_{ij}^{e} x_{ijkm}^{e} \tag{33}$$

$$\Lambda_k^{r} = \sum_{i}\sum_{j}\sum_{m} \lambda_{ij}^{r} x_{ijkm}^{r} \tag{34}$$

The form of service level constraints (26) for express consignments is based on the fact that the sojourn time distribution $S_k^{e}(\tau^{e}) = P\{W_k^{e} \leq \tau^{e}\}$ for high priority (express) customers in a preemptive priority queue is known to be exponential (Chang, 1965). However, such an analytical characterization of the sojourn time distribution $S_k^r(\tau^r) = P\{W_k^r \leq \tau^r\}$ for low priority (regular) customers, appearing in constraint set (27), is not known (Abate and Whitt, 1997). This makes $[CMAHLP - MSLC]$ challenging to solve. We discuss how we tackle the issue of service level constraints for regular customers (corresponding to (27)) in the next section.

3. Solution Methodology

The absence of an analytical characterization of the service level constraint (27) for regular customers makes $[CMAHLP - MSLC]$ challenging to solve. While the Laplace transform of the sojourn time distribution $S_k^r(\tau^r)$, appearing in (27), and its first few moments are well known (Stephan, 1958), the distribution itself is somewhat complicated and requires numerical computation of the inverse Laplace transform, thereby preventing its analytical characterization (Jayaswal et al., 2011). There are approximations proposed in the literature for the sojourn time distribution. However, they are very complex and often not sufficiently accurate (Abate and Whitt, 1997). Moreover, the choice of appropriate approximation to be used depends on Λ_k^e and Λ_k^r, which can only be determined endogenously, and are not known in advance in our model.

Although the exact form of $S_k^r(\tau^r)$ in constraint (27) is unknown, we exploit its special structure, determined numerically using the matrix geometric method. Plots of $S_k^r(\tau^r)$ vs. $(\Lambda_k^e,\ \Lambda_k^r)$, $S_k^r(\tau^r)$ vs. $(\Lambda_k^e,\ \mu_k)$ and $S_k^r(\tau^r)$ vs. $(\Lambda_k^r,\ \mu_k)$ are shown in Figure 1. These plots suggest that $S_k^r(\tau^r)$ is jointly concave in $(\Lambda_k^e,\ \Lambda_k^r)$, in $(\Lambda_k^e,\ \mu_k)$, and also in $(\Lambda_k^r,\ \mu_k)$. However, this does not necessarily show the joint concavity of $S_k^r(\tau^r)$ in $(\Lambda_k^e,\ \Lambda_k^r,\ \mu_k)$. We will, therefore, integrate into our solution method a mechanism to ensure that the concavity assumption is not violated.

Assuming $S_k^r(\tau^r)$ is concave, it can be approximated by a set of tangent hyperplanes at various points $((\Lambda_k^e)^p,\ (\Lambda_k^r)^p,\ (\mu_k)^p)$, $\forall\ p \in P$:

$$S_k^r(\tau^r) = \min_{p \in P} \left\{ (S_k^r(\tau^r))^p + (\Lambda_k^e - (\Lambda_k^e)^p) \left(\frac{\partial (S_k^r(\tau^r))}{\partial \Lambda_k^e} \right)^p + (\Lambda_k^r - (\Lambda_k^r))^p \left(\frac{\partial (S_k^r(\tau^r)}{\partial \Lambda_k^r} \right) + (\mu_k - (\mu_k)^p) \left(\frac{\partial (S_k^r(\tau^r))}{\partial \mu_k} \right)^p \right\},$$

where $(S_k^r(\tau^r))^p$ denotes the value of $S_k^r(\tau^r)$ at a fixed point $((\Lambda_k^e)^p,\ (\Lambda_k^r)^p,\ (\mu_k)^p)$, and $\left(\frac{\partial(S_k^r(\tau^r))}{\partial \Lambda_k^e}\right)^p$, $\left(\frac{\partial(S_k^r(\tau^r))}{\partial \Lambda_k^r}\right)^p$, and $\left(\frac{\partial(S_k^r(\tau^r))}{\partial \mu_k}\right)^p$ are the partial gradients of $S_k^r(\tau^r)$ at $((\Lambda_k^e)^p,\ (\Lambda_k^r)^p,\ (\mu_k)^p)$. Constraint (27) can thus be replaced by the following set of linear constraints:

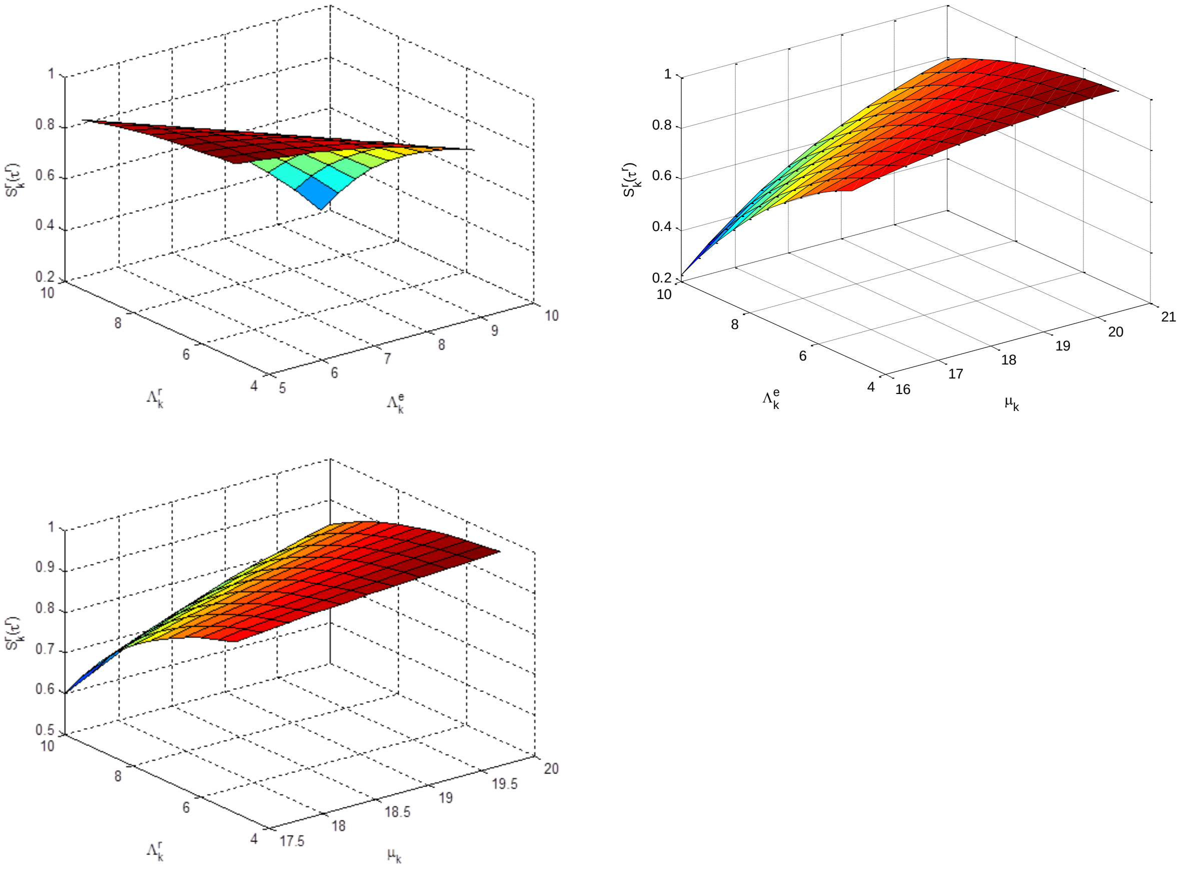

Figure 1: Service Level for Regular Consignments at Hub k vs. Demands for Regular and Express Consignments and Hub Capacity

$$(S_k^r(\tau^r))^p + (\Lambda_k^e - (\Lambda_k^e)^p)\left(\frac{\partial(S_k^r(\tau^r))}{\partial\Lambda_k^e}\right)^p + (\Lambda_k^r - (\Lambda_k^r))^p\left(\frac{\partial(S_k^r(\tau^r)}{\partial\Lambda_k^r}\right) + (\mu_k - (\mu_k)^p)\left(\frac{\partial(S_k^r(\tau^r))}{\partial\mu_k}\right)^p \geq \alpha \quad \forall p \in P \qquad (35)$$

Replacing (27) by the above set of constraints results in a finite but a large number of constraints, which is amenable to cutting plane method.

We use the matrix geometric method to numerically evaluate $(S_k^r(\tau^r))^p$ at a given point $((\Lambda_k^e)^p, (\Lambda_k^r)^p, (\mu_k)^p)$. We refer the readers to Neuts (1981) for details of the matrix geometric method. The use of the matrix geometric method yields explicit recursive formulas for the joint stationary probabilities, which can provide significant computational improvements over the transform techniques (Miller, 1981). Moreover, it gives exact solutions, in contrast to simulation, which is another alternative method to evaluate $S_k^r(\tau^r)$ that at best gives point estimates. The matrix geometric method is also computationally efficient compared to simulation. This is important in solving $[CMAHLP - MSLC]$, which requires repeated evaluation of $(S_k^r(\tau^r))^p$ for various open hubs k at various solutions points p. Once $S_k^r(\tau^r)$ is evaluated at a point $((\Lambda_k^e)^p, (\Lambda_k^r)^p, (\mu_k)^p)$, its gradients are obtained using the *finite difference method* (described in Section 3.2). The gradients are used to generate cuts of the form (35), which are added iteratively in the cutting plane algorithm. The details of the cutting plane algorithm along with its computational performance are presented in Section 3.3.

3.1. The Matrix Geometric Method

3.1.1. The Joint Stationary Queue Length Distribution at Hub k

If we define $N_k^e(t)$ and $N_k^r(t)$ as state variables representing the number of express (high priority) and regular (low priority) consignments at hub k at time t, then $\{\mathbf{N_k}(t)\} := \{N_k^r(t), N_k^e(t), t \geq 0\}$ is a continuous-time two-dimensional Markov chain with state space $\{\mathbf{n_k} = (n_k^r, n_k^e)\}$. The key idea we employ here is that $\{\mathbf{N_k}(t)\}$ is a *quasi-birth-and-death* (QBD) process, which allows us to develop a matrix geometric solution for the joint distribution of the number of consignments of each class at hub k. A simple implementation of the matrix geometric method, however, requires the number of states in the QBD process to be finite. For this, we treat the queue length of express consignments (including the one in service) to be of finite size M, but of size large enough for the desired accuracy of our results. Since express consignments are always served in priority over regular consignments, it is reasonable to assume that its queue size will always be bounded by some large number.

In the Markov process $\{\mathbf{N_k}(t)\}$, a transition can occur only if a consignment of either class arrives or served at hub k. The possible transitions are:

From	To	Rate	Condition
(n_k^r, n_k^e)	$(n_k^r, n_k^e + 1)$	Λ_k^e	for $n_k^r \geq 0$, $n_k^e \geq 0$
(n_k^r, n_k^e)	$(n_k^r + 1, n_k^e)$	Λ_k^r	for $n_k^r \geq 0$, $n_k^e \geq 0$
(n_k^r, n_k^e)	$(n_k^r, n_k^e - 1)$	μ_k	for $n_k^r \geq 0$, $n_k^e > 0$
(n_k^r, n_k^e)	$(n_k^r - 1, n_k^e)$	μ_k	for $n_k^r > 0$, $n_k^e = 0$

The infinitesimal generator Q associated with our system description is thus block-tridiagonal:

$$Q = \begin{pmatrix} B_0 & A_0 & & & \\ A_2 & A_1 & A_0 & & \\ & A_2 & A_1 & A_0 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$$

where B_0, A_0, A_1, A_2 are square matrices of order $M + 1$. These matrices can be easily constructed using the transition rates described above.

$$A_0 = \begin{pmatrix} \Lambda_k^r & & & & \\ & \Lambda_k^r & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \Lambda_k^r \end{pmatrix} ; \quad A_2 = \begin{pmatrix} \mu_k & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix} ; \quad B_0 = \begin{pmatrix} * & \Lambda_k^e & & & \\ \mu_k & * & \Lambda_k^e & & \\ & \mu_k & * & \Lambda_k^e & \\ & & \ddots & \ddots & \ddots \\ & & & \mu_k & * \end{pmatrix}$$

where $*$ is such that $A_0\mathbf{e} + B_0\mathbf{e} = \mathbf{0}$. $A_1 = B_0 - A_2$.

We denote $\mathbf{x}$ as the stationary probability vector of $\{\mathbf{N_k}(t)\}$:

$$\mathbf{x} = [x_{00}, x_{01}, \ldots, x_{0M}, x_{10}, x_{11}, \ldots, x_{1M}, \ldots, \ldots, x_{i0}, x_{i1}, \ldots, x_{iM}, \ldots, \ldots]$$

The vector $\mathbf{x}$ can be partitioned by levels into sub vectors $\mathbf{x}_i$, $i \geq 0$, where $\mathbf{x}_i = [x_{i0}$, x_{i1}, ..., $x_{iM}]$ is the stationary probability of states in level i ($n_k^r = i$). Thus, $\mathbf{x} = [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \ldots]$. $\mathbf{x}$ can be obtained using a set of balance equations, given in matrix form by the following standard relations (Latouche and Ramaswami, 1999; Neuts, 1981):

$$\mathbf{x}Q = \mathbf{0}; \quad \mathbf{x}_{i+1} = \mathbf{x}_i R$$

where R is the minimal non-negative solution to the matrix quadratic equation:

$$A_0 + RA_1 + R^2 A_2 = \mathbf{0}$$

The matrix R can be computed using well known methods (Latouche and Ramaswami, 1999). A simple iterative procedure often used is:

$$R(0) = 0 \; ; \quad R(n+1) = -\left[A_0 + R^2(n)A_2\right] A_1^{-1}$$

The probabilities $\mathbf{x}_0$ are determined from:

$$\mathbf{x}_0(B_0 + RA_2) = \mathbf{0}$$

subject to the normalization equation:

$$\sum_{i=0}^{\infty} \mathbf{x}_i \mathbf{e} = \mathbf{x}_0 (I - R)^{-1} \mathbf{e} = 1$$

where $\mathbf{e}$ is a column vector of ones of size $M+1$.

3.1.2. Estimation of $S_k^r(\tau^r)$

The sojourn time W_k^r of a regular consignment at hub k is the time between its arrival to hub k till it completes service at that hub. It may be preempted by one or more express consignments for service. So it is difficult to characterize the distribution $S_k^r(\cdot)$. Ramaswami and (1985) present an efficient algorithm based on *uniformization* to derive the complimentary distribution of waiting times in phase-type and QBD processes. Jayaswal et al. (2011) adapt their algorithm to derive $S_k^r(\cdot)$, the distribution of the waiting time plus the time in service of low priority (regular) customers, which we adopt in this paper.

Consider a tagged regular consignment entering the system. The time spent by the tagged consignment depends on the number of consignment of either class already present in the system ahead of it, and also on the number of subsequent express arrivals before it completes its service. All subsequent regular arrivals, however, have no influence on its time spent in the system. The tagged consignment's time in the system is, therefore, simply the time until absorption in a modified Markov process $\{\tilde{\mathbf{N}}_{\mathbf{k}}(t)\}$, obtained by setting $\Lambda_k^r = 0$. Consequently, matrix $\tilde{A}_0$, representing transitions to a higher level, becomes a zero matrix. We define an *absorbing* state, call it state $0^{'}$, as the state in which the tagged consignment has finished its service. The infinitesimal generator for this process can be represented as:

$$\tilde{Q} = \left(\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & \cdots \\ \hline b_0 & \tilde{B}_0 & 0 & & & \\ 0 & A_2 & \tilde{A}_1 & 0 & & \\ 0 & & A_2 & \tilde{A}_1 & 0 & \\ \vdots & & & \ddots & \ddots & \ddots \end{array}\right)$$

where, $\tilde{B}_0 = B_0 + A_0$; $\tilde{A}_1 = A_1 + A_0$; and $b_0 = [\mu_k \quad 0 \quad \cdots \quad 0]_{M+1}^T$. The first row and column in $\tilde{Q}$ corresponds to the absorbing state $\acute{0}$. The time spent in system by the tagged consignment, which is the time until absorption in the modified Markov process with rate matrix $\tilde{Q}$, depends on the the arrival rates Λ_k^e and Λ_k^r and the capacity μ_k at hub k. For a given point p (corresponding to arrival rates $(\Lambda_k^e)^p$, $(\Lambda_k^r)^p$ and capacity $(\mu_k)^p$ at hub k) in the solution space, the distribution of the time spent by a regular consignment at hub k is $(S_k^r(y))^p = 1 - \overline{(S_k^r(y))^p}$, where $\overline{(S_k^r(y))^p}$ is the stationary probability that a regular consignment spends more than y units of time at hub k. Further, let $\overline{(S_{ki}^r(y))^p}$ denote the conditional probability that a tagged consignment, which finds i regular consignments ahead of it, spends a time exceeding y at hub k. The probability that a tagged consignment finds i regular consignments is given, using the PASTA property, by $\mathbf{x}_i = \mathbf{x}_0 R^i$. $\overline{S_k^r}(y)$ can be expressed as:

$$\overline{(S_k^r(y))^p} = \sum_{i=0}^{\infty} \mathbf{x}_i \overline{(S_{ki}^r(y))^p}\mathbf{e} \tag{36}$$

$\overline{(S_{ki}^r(y))^p}$ can be computed more conveniently by uniformizing the Markov process $\{\tilde{\mathbf{N}}_\mathbf{k}(t)\}$ with a Poisson process with rate γ, where

$$\gamma = \max_{0\le i\le M} (-\tilde{A}_1)_{ii} = \max_{0\le i\le M} - (A_0 + A_1)_{ii}$$

so that the rate matrix $\tilde{Q}$ is transformed into the discrete-time probability matrix:

$$\hat{Q} = \frac{1}{\gamma}\tilde{Q} + I = \left(\begin{array}{c|cccccc} 1 & 0 & 0 & 0 & 0 & \cdots \\ \hline \hat{b}_0 & \hat{B}_0 & 0 & & & \\ 0 & \hat{A}_2 & \hat{A}_1 & 0 & & \\ 0 & & \hat{A}_2 & \hat{A}_1 & 0 & \\ \vdots & & & \ddots & \ddots & \ddots \end{array}\right)$$

where $\hat{A}_2 = \frac{A_2}{\gamma}$, $\hat{A}_1 = \frac{\tilde{A}_1}{\gamma} + I$, $\hat{b}_0 = \frac{b_0}{\gamma}$. In this uniformized process, points of a Poisson process are generated with a rate γ, and transitions occur at these epochs only. The probability that n Poisson events are generated in time y equals $e^{-\gamma y}\frac{(\gamma y)^n}{n!}$. Suppose the tagged consignment finds i regular consignments ahead of it. Then, for its time at hub k to exceed y, at most i of the n Poisson points may correspond to transitions to lower levels (i.e., service completions of regular consignments). Therefore,

$$\overline{(S_{ki}^r(y))^p} = \sum_{n=0}^{\infty} e^{-\gamma y}\frac{(\gamma y)^n}{n!} \sum_{v=0}^{i} G_v^{(n)}\mathbf{e}, \qquad i \ge 0 \tag{37}$$

where, $G_v^{(n)}$ is a matrix such that its entries are the conditional probabilities, given that the system has made n transitions in the discrete-time Markov process with rate matrix $\hat{Q}$, that v of those transitions correspond to lower levels (i.e., service completions of regular consignments). Substituting the expression for $\overline{(S^r_{ki}(y))^p}$ from (37) into (36), we obtain:

$$\overline{(S^r_k(y))^p} = \sum_{n=0}^{\infty} d_n e^{-\gamma y} \frac{(\gamma y)^n}{n!} \tag{38}$$

where, d_n is given by:

$$d_n = \sum_{i=0}^{\infty} \mathbf{x}_0 R^i \sum_{v=0}^{i} G^{(n)}_v \mathbf{e}, \qquad\qquad n \geq 0 \tag{39}$$

Now,

$$\begin{aligned}
&\sum_{i=0}^{\infty} R^i \sum_{v=0}^{i} G^{(n)}_v \mathbf{e} \\
&= \sum_{i=0}^{n+1} R^i \sum_{v=0}^{i} G^{(n)}_v \mathbf{e} + \sum_{i=n+2}^{\infty} R^i \sum_{v=0}^{n} G^{(n)}_v \mathbf{e} && \big(\text{since} \quad G^{(n)}_v = 0 \quad \text{for} \quad v > n\big) \\
&= \sum_{v=0}^{n+1} \sum_{i=v}^{n+1} R^i G^{(n)}_v \mathbf{e} + (I-R)^{-1} R^{n+2} \mathbf{e} && \left(\text{since} \quad \sum_{v=0}^{n} G^{(n)}_v \mathbf{e} = \mathbf{e}\right) \\
&= \sum_{v=0}^{n+1} (I-R)^{-1} (R^v - R^{n+2}) G^{(n)}_v \mathbf{e} + (I-R)^{-1} R^{n+2} \mathbf{e} \\
&= \sum_{v=0}^{n} (I-R)^{-1} R^v G^{(n)}_v \mathbf{e} + (I-R)^{-1} R^{n+1} G^{(n)}_{n+1} \mathbf{e} && \left(\text{since} \quad \sum_{v=0}^{n+1} G^{(n)}_v \mathbf{e} = \mathbf{e}\right) \\
&= \sum_{v=0}^{n} (I-R)^{-1} R^v G^{(n)}_v \mathbf{e} && \big(\text{since} \quad G^{(n)}_v = 0 \quad \text{for} \quad v > n\big) \\
&= (I-R)^{-1} H_n \mathbf{e} && n \geq 0
\end{aligned}$$

where, $H_n = \sum_{v=0}^{n} R^v G^{(n)}_v$. Therefore,

$$(S^r_k(\tau^r))^p = 1 - \overline{(S^r_k(\tau^r))^p} = \sum_{n=0}^{\infty} e^{-\gamma L_l} \frac{(\gamma L_l)^n}{n!} \mathbf{x}_0 (I-R)^{-1} H_n \mathbf{e} \tag{40}$$

H_n can be computed recursively as:

$$H_{n+1} = H_n \hat{A}_1 + R H_n \hat{A}_2; \quad H_0 = I$$

Therefore, for given arrival rates $((\Lambda^e_k)^p, (\Lambda^r_k)^p)$ and capacity $((\mu_k)^p)$ at hub k, $S^r_k(\tau^r)$ in (16) can be computed using (40).

3.2. Estimation of the Gradient of $S_k^r(\tau^r)$

There are several methods available in the literature to compute the gradients of $S_k^r(\tau^r)$. We use a *finite difference* method as it is probably the simplest and most intuitive, and can be easily explained. Using the finite difference method, the gradients can be computed as:

$$\left(\frac{\partial(S_k^r(\tau^r))}{\partial\Lambda_k^e}\right)^p = \frac{(S_k^r(\tau^r))^{((\Lambda_k^e)^p+d\Lambda_k^e,(\Lambda_k^r)^p,(\mu_k)^p)} - (S_k^r(\tau^r))^{((\Lambda_k^e)^p-d\Lambda_k^e,(\Lambda_k^r)^p,(\mu_k)^p)}}{2d\Lambda_k^e}$$

$$\left(\frac{\partial(S_k^r(\tau^r))}{\partial\Lambda_k^r}\right)^p = \frac{(S_k^r(\tau^r))^{((\Lambda_k^e)^p,(\Lambda_k^r)^p+d\Lambda_k^r,(\mu_k)^p)} - (S_k^r(\tau^r))^{((\Lambda_k^e)^p,(\Lambda_k^r)^p-d\Lambda_k^r,(\mu_k)^p)}}{2d\Lambda_k^r}$$

$$\left(\frac{\partial(S_k^r(\tau^r))^p}{\partial\mu_k}\right)^p = \frac{(S_k^r(\tau^r))^{((\Lambda_k^e)^p,(\Lambda_k^r)^p,(\mu_k)^p+d\mu_k)} - (S_k^r(\tau^r))^{((\Lambda_k^e)^p,(\Lambda_k^r)^p,(\mu_k)^p-d\mu_k)}}{2d\mu_k}$$

where $d\Lambda_k^e$, $d\Lambda_k^r$ and $d\mu_k$ (referred to as step sizes) are infinitesimal changes in the respective variables.

3.3. The Cutting Plane Algorithm

The cutting plane algorithm to solve $[CMAHLP-MSLC]$ is given below. The algorithm differs from the traditional description in that we use the matrix geometric method to generate the cuts and evaluate the function values instead of having an algebraic form for the function and using analytically determined gradients to generate the cuts.

Algorithm 1 Cutting Plane Algorithm

```
$P \leftarrow \Phi$.
repeat
    Solve $[CMAHLP-MSLC(P)]$ to obtain $x_{ijkm}^c\ \forall c \in \{e,r\}$ and $z_{kl}\ \forall k \in N,\ l \in L_k$.
    Obtain $\Lambda_k^e$ and $\Lambda_k^r$ using (33) and (34) and $\mu_k = \sum_{l\in L_k}\mu_{kl}z_{kl}\ \forall k \in \{N : \sum_{l\in L_k} z_{kl} = 1\}$. $p \leftarrow \{(\Lambda_k^e, \Lambda_k^r, \mu_k)\}_{k\in N:\sum_{l\in L_k} z_{kl}=1}$
    Obtain $S_k^r(\tau^r)$ using (40) $\forall k \in \{N : \sum_{l\in L_k} z_{kl} = 1\}$.
    if $S_k^r(\tau^r) \geq \alpha^r\ \forall k \in \{N : \sum_{l\in L_k} z_{kl} = 1\}$ then
        Stop.
    else
        Obtain cuts of the form (35) $\forall k \in \{N : \sum_{l\in L_k} z_{kl} = 1\}$.
        $P \leftarrow P \cup \{p\}$.
    end if
until $S_k^r(\tau^r) < \alpha^r$ for any $k \in \{N : \sum_{l\in L_k} z_{kl} = 1\}$.
```

The success of the cutting plane algorithm relies on the concavity of $S_k^r(\tau^r)$. We have demonstrated, using computational results obtained by the matrix geometric method, that

$S_k^r(\tau^r)$ is concave in (Λ_k^e, Λ_k^r) and separately concave in μ_k. However, it is difficult to establish the joint concavity of $S_k^r(\tau^r)$ in (Λ_k^e, Λ_k^r, μ_k). If the concavity assumption is violated, then the algorithm may cut off parts of the feasible region and terminate with a solution that is suboptimal. We conduct a test to ensure the concavity assumption is not violated. This is done by ensuring that a new point, visited by the cutting plane algorithm after each iteration, lies below all the previously defined cuts, and that all previous points lie below the newly added cut. The test, however, cannot ensure that $S_k^r(\tau^r)$ is concave unless it examines all the points in the feasible region. Still, it does help ensure that the concavity assumption is not violated at least in the region visited by the algorithm. We used this test in our numerical experiments, which did ensure that the concavity assumption was not violated for all the cases studied, at least in the region visited by the algorithm. Details of the test can be found in Atlason et al. (2004).

4. Computational Study

We report our computational experience with the solution method for problem instances based on the US Civil Aeronautics Board (CAB) data. CAB data set contains problem instances of sizes $|N| = 10, 15, 20, 25$. However, the data set does not contain hub capacities (μ_{kl}) and the associated fixed costs (F_{kl}), required for our problem. So, we generate these additional data using the data generation scheme described below.

Flows between various node pairs provided in the CAB data set are scaled such that $TF = 1$, where TF is the total flow in the network. We set 3 potential capacity levels for any hub $k \in N$, expressed as $l \times 0.4 \times TF$, where $l \in L_k = \{1, 2, 3\}$. Fixed cost of opening a hub with capacity μ_{kl} is generated using the function: $F_{kl} = 200(\mu_{kl})^a$, where a represents the economy of scale in installing capacity at a hub. We assume $a = 0.80$ in all our experiments. Inter-hub flow discount factor δ is selected from the set $\{0.2, 0.4, 0.6.0.8\}$. Composition of express (e) and regular (r) consignments is represented as: (e_f, r_f), where e_f and r_f are the fractions of express and regular consignments between any pair of nodes. Consignment composition in our experiments is varied as: (0, 1); (0.2, 0.8); (0.4, 0.6); (0.6, 0.4); (0.8, 0.2); (1, 0).

Results of our computational study for various network sizes (N), inter-hub flow discount factors (δ), and compositions of consignments (e_f, r_f) are presented in Table 1 and Table 2 corresponding to "without Service Level Constraints" and "with Service Level Constraints". For these experiments, we set the values of $\tau^e = 6$ and $\tau^r = 10$ as the threshold on the maximum sojourn time at a hub for express and regular consignment classes respectively. The target service levels $S_e^k(\tau^e = 6)$ and $S_r^k(\tau^r = 10)$ as 0.98. In these two tables, $(e_f, r_f) =$ (0, 1) corresponds to the case with only one consignment class, for which the threshold on the maximum sojourn time at a hub is $\tau^r = 10$. Similarly, $(e_f, r_f) =$ (1, 0) corresponds to the case with only one consignment class, for which the threshold on the maximum sojourn time that at a hub is $\tau^e = 6$.

The results in Table 1 show, as expected, that the service levels provided to regular and express consignments at their hubs deteriorate with an increasing proportion of express consignments in the system. It also shows that increasing discount (decreasing the value of δ) on inter-hub flows results in opening of more hubs to exploit the inter-hub flow discounts. Furthermore, in absence of any explicit service level constraints, the open hubs in the resulting solution generally provide poor service levels. For example, for $N = 10, \delta = 0.2, e_f = 0.4, e_r = 0.6$, the service level provided by the hub located at node 5 for regular consignments is as low as 0.4611.

Table 2 reports the cost of service quality (CoSQ), which is the additional cost of network design to guarantee a target service level ($\alpha = 0.98$) to both the consignment classes. It is computed as the difference between the total cost of network design with and without service level constraints. Figure 2 shows that the change in CoSQ with an increase in the fraction of express consignments (e_f) is not necessarily monotonic. An increase in the fraction of express consignments (e_f), who have a lower value of the maximum threshold on sojourn time, should ideally increase the capacity required to meet their target service level. However, an increase in e_f is accompanied by a corresponding decrease in the fraction of regular consignments (r_f), who receive a less preferential treatment at hubs in presence of priority in service, thereby decreasing the capacity required to meet their target service level. Hence, in presence of priority in service, two opposite forces come into play, the net result of which may be either an increase or a decrease in the capacity required, and hence a corresponding increase or a decrease in CoSQ. For example, as observed from Table 2, CoSQ, in general, increases with an increase in e_f. However, for $N = 10, \delta = 0.4$, CoSQ decreases from 365.1 to 357.6 corresponding to an increase in e_f from 0.8 to 1.0. This is an interesting observation as it suggests that increasing the fraction of consignments that receive priority in service or/and that have a lower value of the maximum threshold on sojourn time may not necessarily increase the total cost of the network design.

A comparison of results between Table 1 and Table 2 shows that the optimal hub-and-spoke network configuration without any service level constraint may differ significantly from the one in presence of such service level constraints. This is amply highlighted, for example, in the case $N = 10, \delta = 0.2, e_f = 0.2, r_f = 0.8$, which results in the following hub (capacity) configuration in absence of any explicit consideration of service levels: $2(1), 3(1), 5(1), 6(1)$. However, in presence of explicit service level constraints ($S_k^e(\tau^e) = 0.98, S_k^r(\tau^r) = 0.98$), the optimal hub (capacity) configuration is: $6(2), 8(3)$. We note here that the economy of scale (a) in hub capacity also plays an important role in the optimal hub location and capacity selection. In absence, of any economy of scale ($a = 1$), an explicit consideration of service level constraints should generally result in more hubs being opened. However, we notice in the above example that the number of open hubs have decreased in presence of such service level constraints, although at higher capacities so as to exploit the economies of scale ($a = 0.80$) in hub capacities.

In Table 3, we show the effect of varying τ^e and τ^r on the network configuration for $N = 15$. For this, we fix τ^e at 8, and vary τ^r between 8 and 128. It can be observed from the results that an increase in τ^e, implying a less stringent service level constraint, generally results in either fewer hubs being opened or the same number of hubs with smaller capacities. For example, for $\delta = 0.2, e_f = 0.5, r_f = 0.5, \tau^e = 8$, an increase in τ^r from 8 to 16 results in a decrease in the number of hubs being opened from 5 to 3. On the other hand, for $\delta = 0.8, e_f = 0.5, r_f = 0.5, \tau^e = 8$, an increase in τ^r from 16 to 32 does not result in any change in the hub locations, but the capacities of both the opened hubs (3 and 6) reduce from level 3 to level 2. Further, the portions in the extreme right side of the plots in Figure 3 shows that a substantial decrease in the maximum threshold on sojourn time (τ^r) can be achieved with only minimal increase in total cost of network design. However, after a certain point, the total cost increases exponentially even with a small decrease in the maximum threshold on sojourn time for regular consignments.

5. Conclusions

In this paper, we studied the hub location and network design problem, characterized by stochastic demand and congestion, with an explicit consideration for customer heterogeneity. Customers were thus assumed to belong to two different priority classes, express and regular, with express customers always receiving priority in service at hubs. To account for the heterogeneous customer requirements, we used a different service level constraint, defined as a lower limit on the probability of a consignment waiting for more than a given threshold

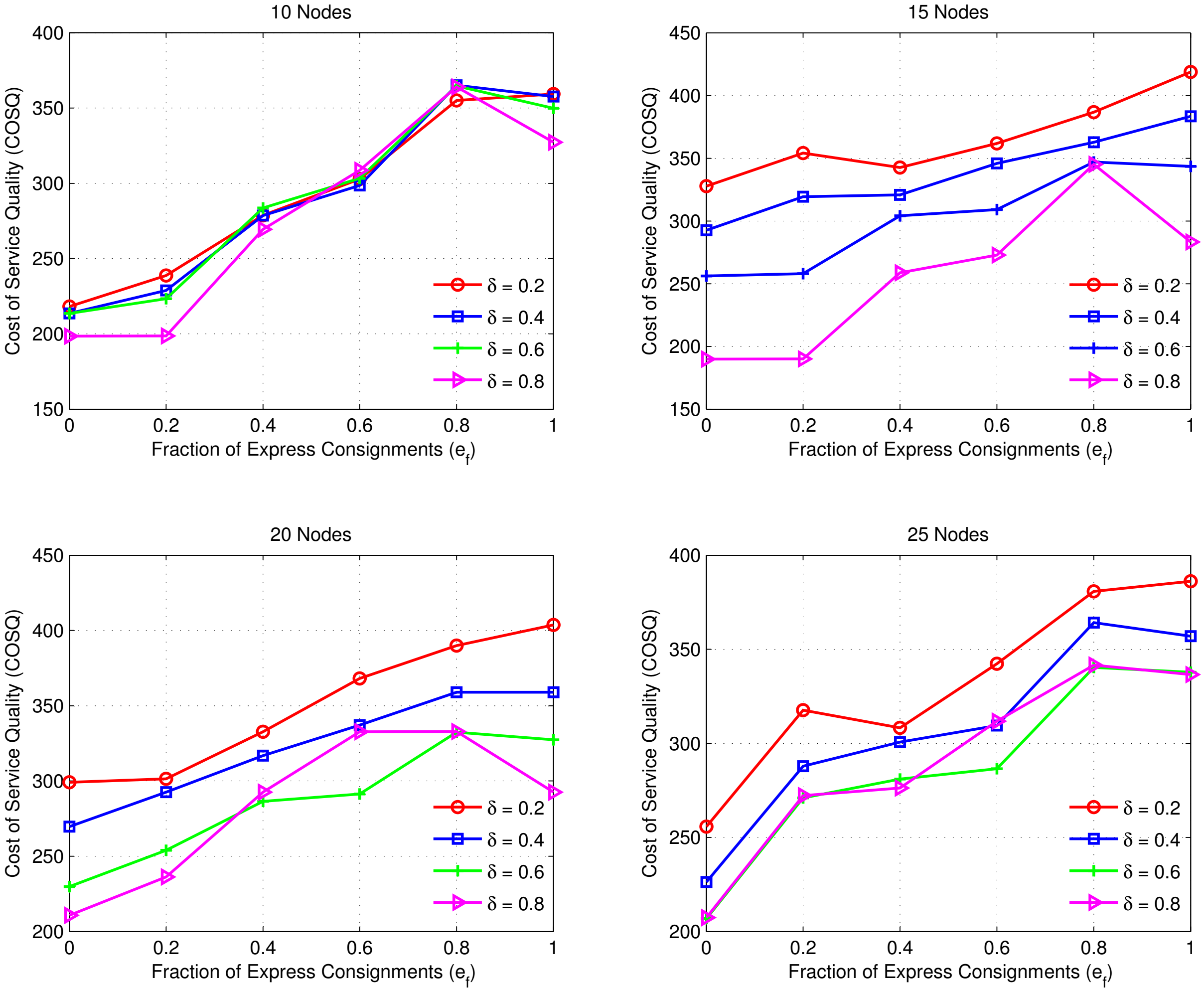

Figure 2: Cost of Service Quality (CoSQ) vs. Fraction of Express Consignments (e_f

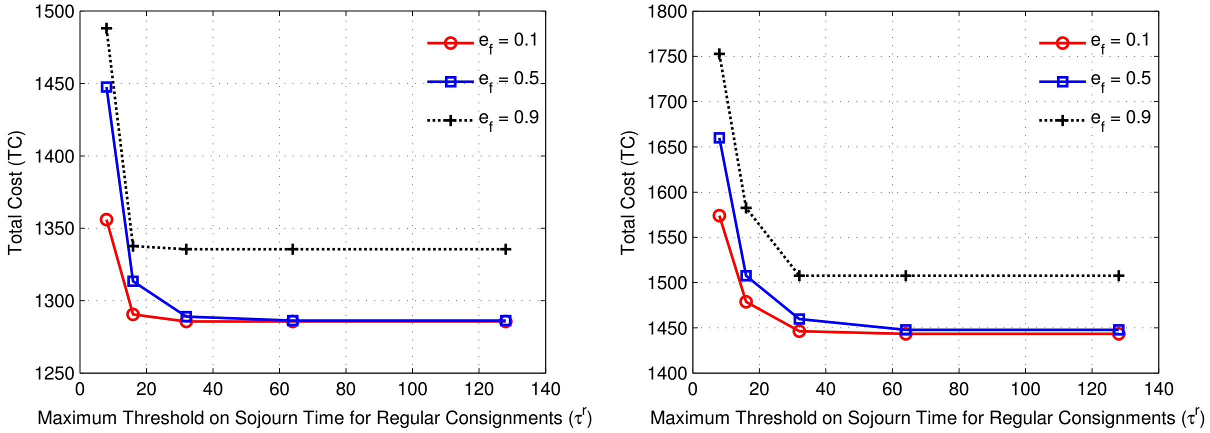

Figure 3: Total Cost (TC) vs. Maximum Threshold Sojourn Time for Regular Consignments (τ_r)

at a hub, for each customer class. The network of hubs, given their locations, was thus modeled as spatially distributed preemptive priority M/M/1 queues. The model sought to determine the hub-and-spoke network design at the minimum total cost, which included the total fixed cost of equipping open hubs with sufficient processing capacity and the variable transportation costs, subject to a service level constraint for each consignment class. The problem proved to be challenging, especially in absence of any known analytical expression for the sojourn time distribution of low priority customers in a preemptive priority M/M/1 queue. To this end, we developed a solution technique that uses the matrix geometric method in a cutting plane framework. Based on an extensive computational study, we demonstrated that the optimal network configuration that accounts for different service levels demanded by heterogeneous customers classes may differ significantly from the one that does not consider service level constraints. Further, we observed that increasing the fraction of consignments that receive priority in service or/and that have a lower value of the maximum threshold on sojourn time may not necessarily increase the total cost of the network design.

This work reported in this paper can be extended in a number of ways. Our study is based on the assumption that each hub behaves like a preemptive priority $M/M/1$ queue. An immediate extension of the current work will be to consider a non-preeemptive priority discipline at hubs. Another possible extension would be a more generalized queuing model, like a priority $M/G/1$ queue model, of the hubs, although the resulting model will be extremely challenging to solve.

Acknowledgements

This research was supported by the Research & Publication Grant, Indian Institute of Management Ahmedabad to the first author, and by the National Science and Engineering Research Council of Canada (NSERC) grant to the second author.

References

Abate, J., Whitt, W., 1997. Asymptotics for M/G/1 low-priority waiting-time tail probabilities. Queueing Systems 25, 173–233.

Abdinnour-Helm, S., Venkataramanan, M.A., 1998. Solution approaches to hub location problems. Annals of Operations Research 78, 31–50.

Alumur, S., Kara, B., 2008. Network hub location problems: The state-of-the-art. European Journal of Operational Research 190, 1–21.

Alumur, S., Kara, B., Karasan, O., 2009. The design of single allocation incomplete hub-networks. Transportation Research B 43, 936–951.

Alumur, S., Nickel, S., da Gama, F.S., 2012. Hub location under uncertainty. Transportation Research: Part B 46, 529–543.

Atlason, J., Epelman, M., Henderson, S., 2004. Call center staffing with simulation and cutting plane methods. Annals of Operations Research 127, 333–358.

Aykin, T., 1994. Lagrangean relaxation based appraoches to capacitated hub-and-spoke network design problem. European Journal of Operational Research 79, 501–523.

Boland, N., Krishnamoorthy, M., Ernst, A.T., Ebery, J., 2004. Preprocessing and cutting for multiple allocation hub location problems. European Journal of Operational Research 155, 638–653.

Camargo, R., Jr., G.M., Ferreira, R., Luna, H., 2009. Multiple allocation hub-and-spoke network design under hub congestion. Computers and Operations Research 36, 3097–3106.

Campbell, J., 1994b. Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72, 387–405.

Campbell, J., Ernst, A., Krishnamoorthy, M., 2002. Hub location problems, in: Drezner, Z., Hamacher, H. (Eds.), Location Analysis: Theory and Applications, Springer, Berlin. pp. 373–408.

Campbell, J., O'Kelly, M., 2012. Twenty-five years of hub location research. Transportation Science 46, 153–169.

Canovas, L., Garcia, S., Marin, A., 2007. Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. European Journal of Operations Research 179, 990–1007.

Chang, W., 1965. Preemptive priority queues. Operations research 13, 820–827.

Contreras, I., Cordeau, J.F., Laporte, G., 2012. Benders decomposition for large-scale uncapacitated hub location problem. Operations Research 59, 1477–1490.

Correia, I., Nickel, S., Saldanha-da-Gama, F., 2010. Single-assignment hub location probems with multiple capacity levels. Transportation Research: Part B 44, 1047–1066.

Costa, M., Captivo, M.E., Climaco, J., 2008. Capacitated single allocation hub location problem - A bi-criteria approach. Computers and Operations Research 35, 3671–3695.

Cunha, C., Silva, M., 2007. A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. European Journal of Operational Research 179, 747–758.

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

Elhedhli, S., Hu, F.X., 2005. Hub-and-spoke network design with congestion. Computers and Operations Research 32, 1615–1632.

Elhedhli, S., Wu, H., 2010. A Lagrangean heuristic for hub-and-spoke system design with capacity selection and congestion. INFORMS Journal of Computing 22, 282–296.

Ernst, A., Krishnamoorthy, M., 1999. Solution algorithms for the capacitated single allocation hub location problem. Annals of Operations Research 86, 141–159.

Gross, D., Harris, C., 1998. Fundamentals of Queueing Theory. 3 ed., John Wiley and Sons, New York.

Hamacher, H., Labbe, M., Nickel, S., Sonneborn, T., 2004. Adapting polyhedral properties from facility to hub location problems. Discrete Applied Mathematics 145, 104–116.

Jayaswal, S., Jewkes, E., Ray, S., 2011. Product differentiation and operations strategy in a capacitated environment. European Journal of Operational Research 210, 716 – 728.

Labbe, M., Yaman, H., 2004. Projecting the flow variables for hub location problems. Network 44, 84–93.

Labbe, M., Yaman, H., Gourdin, E., 2005. A branch and cut algorithm for hub location problems with single assignment. Mathematical Programming 102, 371–405.

Latouche, G., Ramaswami, V., 1999. Introduction to Matrix Analytic Methods in Stochastic Modeling. Society for Industrial and Applied Mathematics, Philadelphia, USA.

Marianov, V., Serra, D., 2003. Location models for airline hubs behaving as M/D/c queues. Computer and Operations Research 30, 983–1003.

Marin, A., 2005a. Formulating and solving splittable capacitated multiple allocation hub location problems. Computer and Operations Research 32, 3093–3109.

Marin, A., 2005b. Uncapacitated Euclidean hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. Journal of Global Optimization 33, 393–422.

Marin, A., Canovas, L., Landete, M., 2006. New formulations for the uncapacitated multiple allocation hub location problem. European Journal of Operations Research 172, 274–292.

Miller, D., 1981. Computation of steady-state probabilities for m/m/1 priority queues. Operations Research 29, pp. 945–958.

Neuts, M., 1981. Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach. Dover Publications, Mineola, USA.

O'Kelly, M., 1986a. The location of interacting hub facilities. Transportation Science 20, 92–106.

O'Kelly, M., 1992b. Hub facility location with fixed costs. Papers in Regional Science 71, 293–306.

O'Kelly, M., Bryan, D., 1998. Hub location with flow economies of scale. Transportation Research Part B 32, 605–616.

Racunica, I., Wynter, L., 2005. Optimal location of intermodal freight hubs. Transportation Research Part B: Methodological 39, 435–477.

Ramaswami, V., , D.L., 1985. Stationary waiting time distribution in queues with phase type service and in quasi-birth-and-death processes. Communications in Statistics. Stochastic Models 1, 125–136.

Stephan, F., 1958. Two queues under preemptive priority with poisson arrival and service rates. Operations Research 6, pp. 399–418.

Yaman, H., 2009. The hierarchical hub median problem with single assignment. Transportation Research Part B 43, 643–658.

Table 1: Configuration of the Hub-and-Spoke System without Service Level Constraints ($\tau^e = 6$, $\tau^r = 10$)

N	δ	e_f	r_f	Hub (Capacity)	$(S_k^e(\tau^e), S_k^r(\tau^r))$	TC	CPU
10	0.2	0	1	2(1), 3(1), 5(1), 6(1)	(–, 0.9382), (–, 0.6834), (–, 0.5595), (–, 0.7121)	773.4	0.5
		0.2	0.8	2(1), 3(1), 5(1), 6(1)	(0.8950, 0.9231), (0.8722, 0.6343), (0.8671, 0.5094), (0.8737, 0.6642)	773.4	1.4
		0.4	0.6	2(1), 3(1), 5(1), 6(1)	(0.8785, 0.9069), (0.8202, 0.5854), (0.8053, 0.4611), (0.8242, 0.6163)	773.4	1.4
		0.6	0.4	2(1), 3(1), 5(1), 6(1)	(0.8595, 0.8897), (0.7469, 0.5379), (0.7149, 0.4154), (0.7554, 0.5692)	773.4	1.3
		0.8	0.2	2(1), 3(1), 5(1), 6(1)	(0.8374, 0.8717), (0.6437, 0.4925), (0.5825, 0.3731), (0.6596, 0.5239)	773.4	1.3
		1	0	2(1), 3(1), 5(1), 6(1)	(0.8119, –), (0.4984, –), (0.3885, –), (0.5262, –)	773.4	0.5
	0.4	0	1	3(1), 5(1), 6(1)	(–, 0.2475), (-, 5.5e-016), (–, 0.8201)	832.7	0.5
		0.2	0.8	3(1), 5(1), 6(1)	(0.8648, 0.2202), (0.8462, 0.0011), (0.8806, 0.7816)	832.7	7.6
		0.4	0.6	3(1), 5(1), 6(1)	(0.7879, 0.1918), (0.7527, 0.0011), (0.8430, 0.7417)	832.7	6.4
		0.6	0.4	3(1), 5(1), 6(1)	(0.6392, 0.1622), (0.6331, 0.0011), (0.7935, 0.7012)	832.7	5.4
		0.8	0.2	3(1), 5(1), 6(1)	(0.4339, 0.1419), (0.4098,0.0031), (0.7284, 0.6608)	832.7	3.6
		1	0	3(1), 5(1), 6(1)	(0.1568, –), (3.3e-016, –), (0.6427, –)	832.7	0.5
	0.6	0	1	3(1), 5(1), 6(1)	(–, 0.1036), (–, 0.1604), (–, 0.8201)	874.7	0.5
		0.2	0.8	3(1), 5(1), 6(1)	(0.8553, 0.0896), (0.8564, 0.1395), (0.8806, 0.7816)	874.7	2.5
		0.4	0.6	3(1), 5(1), 6(1)	(0.7692, 0.0771), (0.7728,0.1206), (0.8430, 0.7417)	874.7	2.1
		0.6	0.4	3(1), 5(1), 6(1)	(0.6319, 0.0661), (0.6404, 0.1039), (0.7935, 0.7012)	874.7	2.0
		0.8	0.2	3(1), 5(1), 6(1)	(0.4128, 0.0589), (0.4310, 0.0911), (0.7284, 0.6608)	874.7	1.7
		1	0	3(1), 5(1), 6(1)	(0.0635, –), (0.0996, –), (0.6427, –)	874.7	0.5
	0.8	0	1	3(1), 5(1), 6(1)	(–, 0.0490), (–, 0.2474), (–, 0.8108)	913.7	0.5
		0.2	0.8	3(1), 5(1), 6(1)	(0.8542, 0.0422), (0.8583, 0.2170), (0.8799, 0.7712)	913.7	2.7
		0.4	0.6	3(1), 5(1), 6(1)	(0.7659, 0.0361), (0.7786, 0.1892), (0.8411, 0.7302)	913.7	2.5
		0.6	0.4	3(1), 5(1), 6(1)	(0.6239, 0.0308), (0.6543, 0.1643), (0.7897, 0.6888)	913.7	2.3
		0.8	0.2	3(1), 5(1), 6(1)	(0.3959, 0.0294), (0.4601, 0.1435), (0.7217, 0.6476)	913.7	2.0
		1	0	3(1), 5(1), 6(1)	(0.0297, –), (0.1568,–), (0.6318, –)	913.7	0.6
15	0.2	0	1	3(1),5(1),6(1),11(1),13(1)	(–, 0.5645), (–, 0.7362), (–, 0.8707), (–, 0.9330), (–, 0.9543)	999.06	4.8
		0.2	0.8	3(1),5(1),6(1),11(1),13(1)	(0.8673, 0.5144), (0.8750, 0.6898), (0.8853, 0.8400), (0.8940, 0.9164), (0.8987, 0.9442)	999.06	11.8
		0.4	0.6	3(1),5(1),6(1),11(1),13(1)	(0.8059, 0.4659), (0.8279, 0.6429), (0.8550, 0.8075), (0.8761, 0.8986), (0.8870, 0.9334)	999.06	11.3
		0.6	0.4	3(1),5(1),6(1),11(1),13(1)	(0.7161, 0.4201), (0.7630, 0.5966), (0.8166, 0.7740), (0.8553, 0.8798), (0.8739, 0.9219)	999.06	11.1
		0.8	0.2	3(1),5(1),6(1),11(1),13(1)	(0.5848, 0.3776), (0.6736, 0.5517), (0.7682, 0.7398), (0.8310, 0.8600), (0.8593, 0.9099)	999.06	11.0
		1	0	3(1),5(1),6(1),11(1),13(1)	(0.3927, –), (0.5504, –), (0.7070, –), (0.8025, –), (0.8430, –)	999.06	4.8
	0.4	0	1	3(1),5(1),6(1),11(1),13(1)	(–, 0.4695), (–, 0.7362), (–, 0.8938), (–, 0.9330), (–, 0.9543)	1127.8	3.6
		0.2	0.8	3(1),5(1),6(1),11(1),13(1)	(0.8641, 0.4224), (0.8750, 0.6898), (0.8879, 0.8677), (0.8940, 0.9164), (0.8987, 0.9442)	1127.8	8.3
		0.4	0.6	3(1),5(1),6(1),11(1),13(1)	(0.7965, 0.3777), (0.8279, 0.6429), (0.8617, 0.8398), (0.8761, 0.8986), (0.8870, 0.9334)	1127.8	7.9
		0.6	0.4	3(1),5(1),6(1),11(1),13(1)	(0.6952, 0.3362), (0.7630, 0.5966), (0.8292, 0.8107), (0.8553, 0.8798), (0.8739, 0.9219)	1127.8	7.6
		0.8	0.2	3(1),5(1),6(1),11(1),13(1)	(0.5435, 0.2986), (0.6736, 0.5517), (0.7891, 0.7808), (0.8310, 0.8600), (0.8593, 0.9099)	1127.8	8.1
		1	0	3(1),5(1),6(1),11(1),13(1)	(0.3164, –), (0.5504, –), (0.7397, –), (0.8025, –), (0.8430, –)	1127.8	3.6
	0.6	0	1	3(1),5(1),6(1),11(1)	(–, 0.0802), (–, 0.6881), (–, 0.8708), (–, 0.9330)	1236.9	4.3
		0.2	0.8	3(1),5(1),6(1),11(1)	(0.8548, 0.0692), (0.8725, 0.6392), (0.8853, 0.8401), (0.8940, 0.9164)	1236.9	11.8
		0.4	0.6	3(1),5(1),6(1),11(1)	(0.7677, 0.0594), (0.8208, 0.5905), (0.8550, 0.8077), (0.8761, 0.8986)	1236.9	11.2
		0.6	0.4	3(1),5(1),6(1),11(1)	(0.6284, 0.0508), (0.7482, 0.5430), (0.8167, 0.7742), (0.8553, 0.8798)	1236.9	11.9
		0.8	0.2	3(1),5(1),6(1),11(1)	(0.4055, 0.0461), (0.6463, 0.4976), (0.7683, 0.7400), (0.8310, 0.8600)	1236.9	12.9
		1	0	3(1),5(1),6(1),11(1)	(0.0489, –), (0.5030, –), (0.7071, –), (0.8025, –)	1236.9	4.3
	0.8	0	1	3(1),5(1),6(1),11(1)	(–, 0.1521), (–, 0.7022), (–, 0.8724), (–, 0.9230)	1317.7	11.8
		0.2	0.8	3(1),5(1),6(1),11(1)	(0.8562, 0.1321), (0.8732, 0.6539), (0.8855, 0.8420), (0.8922, 0.9037)	1317.7	36.1
		0.4	0.6	3(1),5(1),6(1),11(1)	(0.7722, 0.1142), (0.8228, 0.6056), (0.8554, 0.8098), (0.8719, 0.8830)	1317.7	36.3
		0.6	0.4	3(1),5(1),6(1),11(1)	(0.6391, 0.0982), (0.7524, 0.5583), (0.8175, 0.7765), (0.8479, 0.8612)	1317.7	36.7
		0.8	0.2	3(1),5(1),6(1),11(1)	(0.4283, 0.0862), (0.6540, 0.5130), (0.7696, 0.7426), (0.8193, 0.8384)	1317.7	39.2
		1	0	3(1),5(1),6(1),11(1)	(0.0942, –), (0.5165, –), (0.7092, –), (0.7853, –)	1317.7	14.6

TC = Total Cost; CPU = Computation Time (seconds); $(e_f, r_f) = (0, 1)$ refers to a single consignment class with the maximum threshold on sojourn time = 10, whereas $e_f, r_f = (1, 0)$ refers to a single consignment class with the maximum threshold on sojourn time = 6.

Table 1 Continued: Configuration of the Hub-and-Spoke System without Service Level Constraints ($\tau^e = 6$, $\tau^r = 10$)

N	δ	e_f	r_f	Hub (Capacity)	$(S_k^e(\tau^e), S_k^r(\tau^r))$	TC	CPU
20	0.2	0	1	3(1), 6(1), 11(1), 13(1), 16(1)	(–, 0.5990), (–, 0.9297), (–, 0.9482), (–, 0.9609), (–, 0.2015)	938.48	4.3
		0.2	0.8	3(1), 6(1), 11(1), 13(1), 16(1)	(0.8686, 0.5486), (0.8934, 0.9122), (0.8972, 0.9362), (0.9006, 0.9531), (0.8573, 0.1759)	938.48	10.0
		0.4	0.6	3(1), 6(1), 11(1), 13(1), 16(1)	(0.8097, 0.4995), (0.8747, 0.8934), (0.8836, 0.9233), (0.8912, 0.9448), (0.7755, 0.1527)	938.48	9.9
		0.6	0.4	3(1), 6(1), 11(1), 13(1), 16(1)	(0.7244, 0.4526), (0.8528, 0.8735), (0.8681, 0.9096), (0.8808, 0.9360), (0.6469, 0.1321)	938.48	9.8
		0.8	0.2	3(1), 6(1), 11(1), 13(1), 16(1)	(0.6009, 0.4088), (0.8270, 0.8527), (0.8507, 0.8952), (0.8695, 0.9267), (0.4445, 0.1153)	938.48	9.4
		1	0	3(1), 6(1), 11(1), 13(1), 16(1)	(0.4221, –), (0.7967, –), (0.8309, –), (0.8571, –), (0.1263, –)	938.48	4.4
	0.4	0	1	3(1), 6(1), 11(1), 13(1), 16(1)	(–, 0.6344), (–, 0.9395), (–, 0.9483), (–, 0.9603), (–, -1.7e-015)	1075	4.6
		0.2	0.8	3(1), 6(1), 11(1), 13(1), 16(1)	(0.8708, 0.5851), (0.8953, 0.9247), (0.8972, 0.9362), (0.9005, 0.9522), (0.8524, 0.0012)	1075	16.4
		0.4	0.6	3(1), 6(1), 11(1), 13(1), 16(1)	(0.8115, 0.5331), (0.8791, 0.9089), (0.8836, 0.9233), (0.8908, 0.9436), (0.7660, 0.0011)	1075	15.2
		0.6	0.4	3(1), 6(1), 11(1), 13(1), 16(1)	(0.7301, 0.4858), (0.8605, 0.8921), (0.8682, 0.9096), (0.8802, 0.9346), (0.6217, 0.0011)	1075	13.9
		0.8	0.2	3(1), 6(1), 11(1), 13(1), 16(1)	(0.6143, 0.4413), (0.8389, 0.8745), (0.8507, 0.8952), (0.8685, 0.9250), (0.3875, 0.0039)	1075	11.9
		1	0	3(1), 6(1), 11(1), 13(1), 16(1)	(0.4532, –), (–, 0.8141), (–, 0.8309), (–, 0.8557), (-1.1e-015)	1075	4.8
	0.6	0	1	3(1), 6(1), 11(1), 16(1)	(–, 0.2619), (–, 0.9274), (–, 0.9536), (–, 1.6e-015)	1182.4	28.0
		0.2	0.8	3(1), 6(1), 11(1), 16(1)	(0.8577, 0.2297), (0.8929, 0.9093), (0.8985, 0.9433), (0.8542, 0.0012)	1182.4	82.2
		0.4	0.6	3(1), 6(1), 11(1), 16(1)	(0.7789, 0.2007), (0.8737, 0.8899), (0.8866, 0.9323), (0.7639, 0.0012)	1182.4	93.9
		0.6	0.4	3(1), 6(1), 11(1), 16(1)	(0.6580, 0.1750), (0.8511, 0.8693), (0.8733, 0.9206), (0.6156, 0.0012)	1182.4	83.7
		0.8	0.2	3(1), 6(1), 11(1), 16(1)	(0.4684, 0.1528), (0.8244, 0.8478), (0.8583, 0.9083), (0.3774, 0.0045)	1182.4	87.8
		1	0	3(1), 6(1), 11(1), 16(1)	(0.1666, –), (0.7928, –), (0.8417, –), (169.9e-016, –)	1182.4	27.8
	0.8	0	1	3(1), 6(1), 16(1)	(–, 3.3e-015), (–, 0.8646), (–, -1.2e-014)	1245.8	18.2
		0.2	0.8	3(1), 6(1), 16(1)	(0.8507, 0.0011), (0.8869, 0.8355), (0.8531, 0.0012)	1245.8	67.2
		0.4	0.6	3(1), 6(1), 16(1)	(0.7625, 0.0011), (0.8542, 0.8002), (0.7622, 0.0012)	1245.8	84.8
		0.6	0.4	3(1), 6(1), 16(1)	(0.6180, 0.0011), (0.8125, 0.7638), (0.6184, 0.0011)	1245.8	69.1
		0.8	0.2	3(1), 6(1), 16(1)	(0.3923, 0.0037), (0.7582, 0.7266), (0.3823, 0.0042)	1245.8	76.8
		1	0	3(1), 6(1), 16(1)	(1.9e-015, –), (0.6988, –), (-7.3e-015, –)	1245.8	18.7
25	0.2	0	1	3(1), 11(1), 16(1), 23(1)	(–, 0.550295), (–, 0.904348), (–, 0.248776), (–, 0.923292)	1002.8	11.8
		0.2	0.8	3(1), 11(1), 16(1), 23(1)	(0.8667, 0.5004), (0.8893, 0.8804), (0.8583, 0.2182), (0.8922, 0.9040)	1002.8	23.9
		0.4	0.6	3(1), 11(1), 16(1), 23(1)	(0.8044, 0.4523), (0.8651, 0.8549), (0.7787, 0.1903), (0.8720, 0.8834)	1002.8	23.9
		0.6	0.4	3(1), 11(1), 16(1), 23(1)	(0.7128, 0.4070), (0.8355, 0.8282), (0.6545, 0.1653), (0.8480, 0.8616)	1002.8	23.4
		0.8	0.2	3(1), 11(1), 16(1), 23(1)	(0.5783, 0.3651), (0.7994, 0.8006), (0.4606, 0.1443), (0.8195, 0.8389)	1002.8	24.6
		1	0	3(1), 11(1), 16(1), 23(1)	(0.3809, –), (0.7554, –), (0.1577, –), (0.7857, –)	1002.8	14.7
	0.4	0	1	3(1), 11(1), 16(1), 23(1)	(–, 0.6195), (–, 0.8946), (–, 0.0253), (–, 0.9365)	1138.9	10.6
		0.2	0.8	3(1), 11(1), 16(1), 23(1)	(0.8694, 0.5691), (0.8880, 0.8686), (0.8538, 0.0218), (0.8946, 0.9209)	1138.9	25.1
		0.4	0.6	3(1), 11(1), 16(1), 23(1)	(0.8121, 0.5198), (0.8619, 0.8409), (0.7645, 0.0186), (0.8777, 0.9042)	1138.9	25.0
		0.6	0.4	3(1), 11(1), 16(1), 23(1)	(0.7296, 0.4726), (0.8296, 0.8119), (0.6206, 0.0159), (0.8581, 0.8864)	1138.9	26.4
		0.8	0.2	3(1), 11(1), 16(1), 23(1)	(0.61087, 0.4281), (0.78989, 0.7822), (0.38879, 0.0170), (0.8353, 0.8679)	1138.9	23.4
		1	0	3(1), 11(1), 16(1), 23(1)	(0.4399, –), (0.7408, –), (0.0152, –), (0.8088, –)	1138.9	11.2
	0.6	0	1	3(1), 11(1), 16(1)	(–, 1.6e-015), (–, 0.8646), (–, -1.1e-015)	1247.1	254.4
		0.2	0.8	3(1), 11(1), 16(1)	(0.8537, 0.0011), (0.8846, 0.8328), (0.8530, 0.0012)	1247.1	841.1
		0.4	0.6	3(1), 11(1), 16(1)	(0.7640, 0.0011), (0.8533, 0.7993), (0.7620, 0.0011)	1247.1	684.8
		0.6	0.4	3(1), 11(1), 16(1)	(0.6154, 0.0011), (0.8136, 0.7648), (0.6187, 0.0011)	1247.1	811.0
		0.8	0.2	3(1), 11(1), 16(1)	(0.3798, 0.0043), (0.7630, 0.7296), (0.3825, 0.0042)	1247.1	611.4
		1	0	3(1), 11(1), 16(1)	(9.9e-016, –), (0.6988, –), (6.64e-016, –)	1247.1	256.4
	0.8	0	1	3(1), 11(1), 16(1)	(–, 0), (–, 0.8646), (–, 1.7e-015)	1316.4	247.0
		0.2	0.8	3(1), 11(1), 16(1)	(0.8538, 0.0012), (0.8848, 0.8330), (0.8527, 0.0015)	1316.4	831.7
		0.4	0.6	3(1), 11(1), 16(1)	(0.7628, 0.0011), (0.8537, 0.7997), (0.7627, 0.0012)	1316.4	852.2
		0.6	0.4	3(1), 11(1), 16(1)	(0.6175, 0.0012), (0.8131, 0.7644), (0.6176, 0.0012)	1316.4	794.8
		0.8	0.2	3(1), 11(1), 16(1)	(0.3791, 0.0044), (0.7627, 0.7294), (0.3839, 0.0042)	1316.4	755.8
		1	0	3(1), 11(1), 16(1)	(0, –), (0.6988, –), (1.1e-015, –)	1316.4	281.3

TC = Total Cost; CPU = Computation Time (seconds); $(e_f, r_f) = (0, 1)$ refers to a single consignment class with the maximum threshold on sojourn time = 10, whereas $e_f, r_f = (1, 0)$ refers to a single consignment class with the maximum threshold on sojourn time = 6.

Table 2: Configuration of the Hub-and-Spoke System with Service Level Constraints ($\tau^e = 6$, $\tau^r = 10$, $\alpha^e = 0.98$, $\alpha^r = 0.98$)

N	δ	e_f	r_f	Hub (Capacity)	($S_k^e(\tau^e)$, $S_k^r(\tau^r)$)	TC	CoSQ	CPU
10	0.2	0	1	3(2), 5(2), 6(2)	(–, 0.9921), (–, 0.98), (–,0.9947)	991.5	218.1	2.4
		0.2	0.8	6(2), 8(3)	(0.9882, 0.9892), (0.9982, 0.9823)	1012.2	238.8	22.9
		0.4	0.6	3(2), 5(3), 6(2)	(0.9836, 0.9816), (0.9978, 0.9964), (0.9840, 0.9832)	1053.2	278.8	12.8
		0.6	0.4	3(2), 6(2), 8(3)	(0.98, 0.98), (0.98, 0.9799), (0.9953, 0.9871)	1076.3	302.9	28.5
		0.8	0.2	3(3), 5(3), 6(2)	(0.9954, 0.9928), (0.9938,0.9861), (0.98,0.9844)	1128.5	355.1	21.1
		1	0	2(2), 3(3), 6(3)	(0.98, –), (0.98, –), (0.9953, –)	1132.8	359.4	14.0
	0.4	0	1	3(2), 5(2), 6(2)	(–, 0.9873), (–, 0.98), (–, 0.9967)	1046.2	213.5	5.7
		0.2	0.8	6(2), 8(3)	(0.9882, 0.9892), (0.9982, 0.9823)	1061.6	228.8	19.1
		0.4	0.6	3(2), 5(3), 6(2)	(0.9838, 0.9799), (0.9978, 0.9963), (0.9840, 0.9851)	1111.3	278.6	15.0
		0.6	0.4	3(2), 6(2), 8(3)	(0.98, 0.9799), (0.98, 0.9799), (0.9953, 0.9872)	1131.3	298.6	11.5
		0.8	0.2	3(3), 6(2), 8(3)	(0.9955, 0.9926), (0.98, 0.9851), (0.9938, 0.9861)	1197.8	365.1	15.5
		1	0	6(3), 8(3)	(0.9887, –), (0.98, –)	1190.3	357.6	4.2
	0.6	0	1	3(2), 5(2), 6(2)	(–, 0.9835), (–, 0.9846), (–, 0.9967)	1088.4	213.6	2.8
		0.2	0.8	6(2), 8(3)	(0.9879, 0.9889), (0.9983, 0.9825)	1098.1	223.3	15.5
		0.4	0.6	3(2), 5(3), 6(2)	(0.9841,0.9799), (0.9975, 0.9951), (0.9855, 0.9890)	1158.4	283.6	18.8
		0.6	0.4	3(2), 6(2), 8(3)	(0.9801, 0.9799), (0.98, 0.981), (0.9953, 0.9866)	1178.0	303.3	15.8
		0.8	0.2	3(3), 6(2), 8(3)	(0.9954, 0.9925), (0.98, 0.9851), (0.9938, 0.9861)	1239.4	364.6	18.4
		1	0	6(3), 8(3)	(0.9887, –), (0.98, –)	1224.6	349.9	4.8
	0.8	0	1	6(2), 8(3)	(–, 0.9949), (–, 0.9909)	1112.3	198.5	2.2
		0.2	0.8	6(2), 8(3)	(0.9884, 0.9895), (0.9982, 0.9820)	1112.4	198.6	13.5
		0.4	0.6	6(3), 8(3)	(0.9982, 0.9983), (0.9965, 0.9799)	1183.3	269.5	16.6
		0.6	0.4	3(2), 6(2), 8(3)	(0.9805, 0.9799), (0.98, 0.9810), (0.9952, 0.9866)	1222.3	308.6	21.2
		0.8	0.2	3(3), 6(2), 8(3)	(0.9958,0.9936), (0.98, 0.9850), (0.9933, 0.9841)	1277.8	364.0	15.6
		1	0	6(3), 8(3)	(–,0.9887), (-,0.98)	1241.0	327.2	3.9
15	0.2	0	1	0(2), 3(2), 6(2), 11(2)	(–, 0.9958), (–, 0.98), (-, 0.9972), (–, 0.9987)	1326.8	327.7	84.0
		0.2	0.8	3(3), 11(2), 12(2)	(0.9985, 0.9957), (0.9903, 0.9979), (0.9879, 0.9854)	1353.3	354.2	42.1
		0.4	0.6	3(2), 5(2), 6(2), 11(2)	(0.9834, 0.9799), (0.9844, 0.9799), (0.9824, 0.9819), (0.9887, 0.9967)	1341.7	342.6	176.1
		0.6	0.4	3(3), 6(2), 11(2), 13(2)	(0.9945, 0.9799), (0.9817, 0.9853), (0.9868, 0.9952), (0.9884, 0.9972)	1360.9	361.8	103.8
		0.8	0.2	0(2), 3(3), 6(2), 11(2)	(0.98, 0.9848), (0.9922, 0.9799), (0.98, 0.9853), (0.9837, 0.9921)	1385.8	386.8	200.9
		1	0	3(3), 11(2), 12(3)	(–, 0.98), (–, 0.9820), (–, 0.9948)	1417.7	418.7	87.1
	0.4	0	1	3(2), 11(2), 12(2)	(0.98, –), (0.9979, –), (0.98, –)	1420.3	292.6	67.6
		0.2	0.8	3(3), 11(2), 12(2)	(0.9985, 0.9963), (0.9903, 0.9979), (0.9877, 0.9827)	1447.1	319.3	55.7
		0.4	0.6	3(3), 11(2), 12(2)	(0.9969, 0.9884), (0.9887, 0.9967), (0.9839, 0.9799)	1448.6	320.8	118.0
		0.6	0.4	3(3), 6(2), 11(2), 13(2)	(0.9945, 0.9799), (0.9823, 0.9869), (0.9868, 0.9952), (0.9880, 0.9967)	1473.7	345.9	95.9
		0.8	0.2	0(2), 3(3), 6(2), 11(2)	(0.98, 0.9844), (0.9921, 0.9799), (0.98, 0.9857), (0.9838, 0.9922)	1490.5	362.7	229.5
		1	0	3(3), 11(2), 12(3)	(0.9812, –), (0.9820, –), (0.9945, –)	1511.2	383.4	55.7
	0.6	0	1	0(2), 3(2), 7(2)	(–, 0.9879), (–, 0.98), (–, 0.9965)	1493.1	256.2	49.5
		0.2	0.8	3(3), 6(2)	(0.9983, 0.9877), (0.9878, 0.9840)	1495.0	258	59.9
		0.4	0.6	0(2), 3(3), 7(2)	(0.9851, 0.9875), (0.9973, 0.9918), (0.9860, 0.9904)	1541.1	304.2	209.8
		0.6	0.4	3(3), 6(2), 11(2)	(0.9946, 0.9799), (0.98, 0.9799), (0.9826, 0.9888)	1546.1	309.1	110.4
		0.8	0.2	0(2), 3(3), 6(2), 11(2)	(0.98, 0.9847), (0.9922, 0.9799), (0.98, 0.9859), (0.9836, 0.9918)	1583.9	347	450.0
		1	0	3(3), 6(3)	(0.98, –), (0.9887, –)	1580.4	343.5	25.1
	0.8	0	1	3(3), 6(2)	(–, 0.9906), (–, 0.9951)	1507.5	189.8	29.6
		0.2	0.8	3(3), 6(2)	(0.9982, 0.9819), (0.9884, 0.9896)	1507.7	190	21.9
		0.4	0.6	3(3), 6(3)	(0.9967, 0.9799), (0.9981, 0.9982)	1576.3	258.7	141.2
		0.6	0.4	0(2), 3(3), 6(2)	(0.9830, 0.9881), (0.9945, 0.98), (0.9801, 0.9813)	1590.5	272.8	90.3
		0.8	0.2	0(2), 3(3), 6(3)	(0.98, 0.9852), (0.9925, 0.9799), (0.9962, 0.9951)	1662.8	345.1	238.1
		1	0	3(3), 6(3)	(0.98, –), (0.9887, –)	1601.0	283.3	24.1

TC = Total Cost; CPU = Computation Time (seconds); CoSQ= Cost of Service Quality; ; $(e_f, r_f) = (0, 1)$ refers to a single consignment class with the maximum threshold on sojourn time = 10, whereas $e_f, r_f = (1, 0)$ refers to a single consignment class with the maximum threshold on sojourn time = 6.

Table 2 Continued: Configuration of the Hub-and-Spoke System with Service Level Constraints ($\tau^e = 6$, $\tau^r = 10$, $\alpha^e = 0.98$, $\alpha^r = 0.98$)

N	δ	e_f	r_f	Hub (Capacity)	($S^e_k(\tau^e)$, $S^r_k(\tau^r)$)	TC	CoSQ	CPU
20	0.2	0	1	3(2), 11(2), 15(2), 16(2)	(–, 0.9930), (–, 0.9990), (–, 0.9971), (–, 0.9850)	1237.7	299.2	992.8
		0.2	0.8	3(2), 11(2), 15(2), 16(2)	(0.9874, 0.9825), (0.9906, 0.9984), (0.9893, 0.9947), (0.9877, 0.9799)	1239.9	301.4	193.7
		0.4	0.6	0(2), 3(2), 11(2), 16(2)	(0.9815, 0.98), (0.9844, 0.9799), (0.9887, 0.9970), (0.9843, 0.9799)	1271.2	332.7	442.7
		0.6	0.4	3(2), 11(2), 15(2), 16(3)	(0.9805, 0.9799), (0.9880, 0.9967), (0.9816, 0.9848), (0.9964, 0.9934)	1306.6	368.1	660.3
		0.8	0.2	3(2), 11(2), 15(2), 16(3)	(0.98, 0.9836), (0.9842, 0.9929), (0.98, 0.9853), (0.9919, 0.9799)	1328.4	389.9	455.8
		1	0	3(3), 11(2), 16(3)	(0.9869, –), (0.9846, –), (0.9907, –)	1342.2	403.7	69.4
	0.4	0	1	3(2), 16(2), 18(2)	(–, 0.98), (–, 0.98), (–, 0.9979)	1344.6	269.6	718.7
		0.2	0.8	3(2), 6(2), 16(2)	(0.9875, 0.9799), (0.9870, 0.9867), (0.9885, 0.9799)	1367.5	292.5	321.8
		0.4	0.6	0(2), 3(2), 11(2), 16(2)	(0.9816, 0.9799), (0.9839, 0.9799), (0.9889, 0.9971), (0.9844, 0.9799)	1391.8	316.8	1169.0
		0.6	0.4	3(2), 6(2), 16(3)	(0.98025, 0.9799), (0.98, 0.9799), (0.9953, 0.9872)	1412.3	337.2	808.6
		0.8	0.2	3(3), 11(2), 16(3)	(0.9940, 0.9867), (0.9858, 0.9948), (0.9934, 0.9843)	1433.9	358.9	291.3
		1	0	3(3), 11(2), 16(3)	(0.9899, –), (0.9838, –), (0.9886, –)	1433.9	358.9	52.6
	0.6	0	1	3(2), 6(2), 16(2)	(–, 0.9876), (–, 0.9966), (–, 0.98)	1412.2	229.8	716.0
		0.2	0.8	3(2), 6(2), 16(2)	(0.9872, 0.9799), (0.9873, 0.9868), (0.9885, 0.9799)	1436.4	254.0	516.1
		0.4	0.6	3(2), 6(2), 16(3)	(0.9835, 0.9799), (0.9854, 0.9890), (0.9976, 0.9951)	1468.8	286.4	725.1
		0.6	0.4	3(2), 6(2), 16(3)	(0.9803, 0.98), (0.98, 0.9804), (0.9953, 0.9869)	1473.7	291.3	944.2
		0.8	0.2	3(3), 11(2), 16(3)	(0.9945, 0.9887), (0.9855, 0.9944), (0.993, 0.9825)	1514.8	332.4	581.2
		1	0	3(3), 16(3)	(0.9816, –), (0.9877, –)	1509.7	327.3	101.1
	0.8	0	1	3(2), 6(2), 16(2)	(–, 0.98), (–, 0.9979), (–, 0.98)	1456.6	210.8	468.5
		0.2	0.8	10(2), 16(3)	(0.9879, 0.9799), (0.9983, 0.9901)	1482.1	236.3	227.9
		0.4	0.6	3(3), 16(3)	(0.9974, 0.9925), (0.9976, 0.9950)	1538.3	292.5	493.7
		0.6	0.4	6(2), 16(2), 19(3)	(0.9805, 0.9799), (0.9801, 0.9799), (0.9952, 0.9871)	1578.5	332.7	636.0
		0.8	0.2	3(3), 6(2), 16(3)	(0.9957, 0.9931), (0.9811, 0.9876), (0.9931, 0.9831)	1578.5	332.8	725.2
		1	0	3(3), 16(3)	(0.9827, –), (0.9869, –)	1538.3	292.5	65.5
25	0.2	0	1	3(2), 11(2), 16(2)	(–, 0.98), (–, 0.9979), (–, 0.98)	1258.5	255.7	1952.5
		0.2	0.8	13(2), 16(2), 20(2), 23(2)	(0.9899, 0.9968), (0.9880, 0.9799), (0.9872, 0.9799), (0.9900, 0.9976)	1320.6	317.7	1715.1
		0.4	0.6	0 (2), 3(2), 11(2), 16(2)	(0.9840, 0.9871), (0.9831, 0.9799), (0.9877, 0.9948), (0.9846, 0.9799)	1311.0	308.2	4764.6
		0.6	0.4	11(2), 17(3), 20(2)	(0.9831, 0.9891), (0.9940, 0.9799), (0.9816, 0.9799)	1345.1	342.3	3255.2
		0.8	0.2	13(2), 11(2), 15(2), 17(3)	(0.98, 0.9844), (0.9802, 0.9864), (0.98, 0.985), (0.9936, 0.9858)	1383.6	380.8	2778.1
		1	0	11(2), 16(3), 20(3)	(0.98, –), (0.9893, –), (0.9913, –)	1389.0	386.1	423.2
	0.4	0	1	3(2), 11(2), 16(2)	(–, 0.98), (–, 0.9979), (–, 0.98)	1365.1	226.3	2257.9
		0.2	0.8	3(2), 11(2), 16(2), 23(2)	(0.9874, 0.9813), (0.9898, 0.9965), (0.9876, 0.9799), (0.9903, 0.9977)	1426.7	287.8	1631.3
		0.4	0.6	3(2), 11(2), 16(3)	(0.9846, 0.9799), (0.9871, 0.9937), (0.9971, 0.9924)	1439.6	300.7	2411.5
		0.6	0.4	3(2), 11(2), 17(3)	(0.9809, 0.9799), (0.9833, 0.9891), (0.9941, 0.9799)	1448.4	309.5	2523.9
		0.8	0.2	1(3), 3(3), 11(2)	(0.9924, 0.98), (0.9961, 0.9945), (0.9809, 0.9874)	1503.1	364.2	2943.5
		1	0	3(3), 11(2), 16(3)	(0.9911, –), (0.98, –), (0.9895, –)	1495.7	356.9	415.0
	0.6	0	1	3(2), 11(2), 16(2)	(–, 0.98) (–, 0.9979), (–, 0.98)	1454.0	206.9	1289.9
		0.2	0.8	3(2), 11(2), 16(3)	(0.9877, 0.9799), (0.9897, 0.9964), (0.9986, 0.9979)	1518.1	271.0	1074.8
		0.4	0.6	3(2), 11(2), 16(3)	(0.9845, 0.9799), (0.9869, 0.9933), (0.9972, 0.9927)	1528.1	281.0	2204.9
		0.6	0.4	11(), 17(3), 20(2)	(0.9832, 0.9891), (0.9941, 0.9799), (0.9809, 0.9799)	1533.6	286.5	2939.7
		0.8	0.2	1(3), 3(3), 11(2)	(0.9924, 0.9799), (0.9960, 0.9943), (0.9812, 0.9880)	1587.5	340.4	2904.2
		1	0	3(3), 11(2), 16(3)	(0.99068, –), (0.98, –), (0.9901, –)	1584.9	337.8	507.8
	0.8	0	1	3(2), 11(2), 16(2)	(–, 0.98), (–, 0.9979), (–, 0.98)	1523.8	207.4	1128.9
		0.2	0.8	3(2), 11(2), 16(3)	(0.9878, 0.9799), (0.9897, 0.9963), (0.9986, 0.9979)	1588.6	272.2	1768.5
		0.4	0.6	3(2), 11(2), 17(3)	(0.9836, 0.9799), (0.9873, 0.9938), (0.9973, 0.9924)	1592.7	276.3	2659.9
		0.6	0.4	11(2), 19(3), 20(2)	(0.9831, 0.9890), (0.9944, 0.9799), (0.9802, 0.9799)	1628.2	311.8	2388.2
		0.8	0.2	1(3), 3(3), 11(2)	(0.9924, 0.9799), (0.9961, 0.9946), (0.9807, 0.9870)	1658.0	341.6	3383.8
		1	0	3(3), 11(2), 16(3)	(0.9909, –), (0.98, –), (0.9897, –)	1653.0	336.6	832.2

TC = Total Cost; CPU = Computation time (seconds); CoSQ= Cost of Service Quality; $(e_f, r_f) = (0, 1)$ refers to a single consignment class with the maximum threshold on sojourn time = 10, whereas $e_f, r_f = (1, 0)$ refers to a single consignment class with the maximum threshold on sojourn time = 6.

Table 3: Effect of Varying τ^e and τ^r on the Configuration of the Hub-and-Spoke System ($N = 15$, $\alpha^e = 0.98$, $\alpha^r = 0.98$)

δ	e_f	r_f	τ^e	τ^r	Hubs (Capacity)	$(S_k^e(\tau^e), S_k^r(\tau^r))$	TC	CPU
0.2	0.1	0.9	8	8	3(3), 11(2), 12(2)	(0.9998, 0.9892), (0.9981, 0.9943), (0.9980, 0.9799)	1356.0	38.5
				16	3(2), 11(2), 12(2)	(0.9975, 0.9799), (0.9981, 0.9999), (0.9977, 0.9988)	1290.5	51.4
				32	3(2), 6(2), 11(2)	(0.9971, 0.9848, (0.9980, 1), (0.9981, 1)	1285.7	3.4
				64	3(2), 6(2), 11(2)	(0.9971, 0.9998), (0.9980, 1) (0.9981, 1)	1285.7	3.4
				128	3(2), 6(2), 11(2)	(0.9971, 1), (1, 0.9980), (0.9981, 1)	1285.7	3.4
0.2	0.5	0.5	8	8	0(2), 3(3), 6(2), 5(2), 11(2)	(0.9967, 0.9847), (0.9997, 0.9941), (0.9967, 0.9799), (0.9965, 0.9814), (0.9972, 0.9891)	1447.5	69.2
				16	3(2), 11(2), 12(2)	(0.9921, 0.9799), (0.9967, 0.9995), (0.9902, 0.9799)	1313.4	80.7
				32	3(2), 6(2), 11(2)	(0.9844, 0.9799), (0.9942, 0.9999), (0.9972, 1)	1289.0	34.6
				64	3(2), 6(2), 11(2)	(0.98, 0.9999), (0.9955, 1), (0.9972, 1)	1286.2	4.8
				128	3(2), 6(2), 11(2)	(0.98, 1), (0.995512, 1), (0.99721, 1)	1286.2	4.8
0.2	0.9	0.1	8	8	3(3), 5(3), 6(2), 11(2), 13(2)	(0.9990, 0.9799), (0.9995, 0.9916), (0.9955, 0.9799), (0.9957, 0.9821), (0.9967, 0.9892)	1488.1	123.6
				16	3(2), 5(2), 6(2), 11(2)	(0.9839, 0.9799), (0.9840, 0.9799), (0.9905, 0.9932), (0.9957, 0.9992)	1337.7	92.8
				32	3(2), 5(2), 6(2), 11(2)	(0.9839, 0.9987), (0.9807, 0.9977), (0.9921, 0.9999), (0.9957, 0.9999)	1335.5	73.1
				64	3(2), 5(2), 6(2), 11(2)	(0.9839, 0.9999), (0.9807, 0.9999), (0.9921, 1), (0.9957, 1)	1335.5	72.9
				128	3(2), 5(2), 6(2), 11(2)	(0.9839, 1), (0.9807, 1), (0.9921, 1), (0.9957, 1)	1335.5	74.0
0.8	0.1	0.9	8	8	3(3), 6(3)	(0.9998, 0.9799), (0.9999, 0.9984)	1573.9	57.5
				16	3(2), 6(2)	(0.9976, 0.9799), (0.9974, 0.9921)	1478.8	60.3
				32	3(2), 6(2)	(0.9971, 0.9799), (0.9978, 0.9999)	1446.2	24.0
				64	3(2), 6(2)	(0.9970, 0.9913), (0.9979, 1)	1443.4	3.3
				128	3(2), 6(2)	(0.9970, 1), (0.9979, 1)	1443.4	3.2
0.8	0.5	0.5	8	8	0(2), 3(3), 6(3)	(0.9964, 0.98), (0.9995, 0.9799), (0.9997, 0.9950)	1659.9	113.8
				16	3(3), 6(3)	(0.9987, 0.9881), (0.9951, 0.9973)	1507.5	22.9
				32	3(2), 6(2)	(0.9834, 0.9799), (0.9908, 0.9992)	1459.7	38.5
				64	3(2), 6(2)	(0.98, 0.9997), (0.9924, 1)	1447.8	4.3
				128	3(2), 6(2)	(0.98, 1), (0.9924, 1)	1447.8	4.2
0.8	0.9	0.1	8	8	0(3), 3(3), 6(3)	(0.9995, 0.9906), (0.9990, 0.9797), (0.9991, 0.9832)	1752.7	84.2
				16	3(3), 6(2), 11(2)	(0.9947, 0.9799), (0.9897, 0.9917), (0.9953, 0.9983)	1582.5	85.6
				32	3(3), 6(2)	(0.9867, 0.9977), (0.9885, 0.9996)	1507.5	20.4
				64	3(3), 6(2)	(0.9867, 1), (0.9885, 1)	1507.5	20.4
				128	3(3), 6(2)	(0.9867, 1), (0.9885, 1)	1507.5	21.1

TC = Total Cost; CPU = Computation Time (seconds)