Journal of the Operations Research
Society of Japan
2003, Vol. 46, No. 4, 409-428

ON THE HUB-AND-SPOKE MODEL WITH ARC CAPACITY CONATRAINTS

Mihiro Sasaki
Nanzan University

Masao Fukushima
Kyoto University

(Received September 2, 2002; Revised April 28, 2003)

Abstract In this paper, we present a new formulation of one-stop capacitated hub-and-spoke model as a natural extension of the uncapacitated one-stop model. The model involves arc capacity constraints as well as hub capacity constraints, which enables us to incorporate some practical factors into the model. We also present a branch-and-bound based exact solution method with Lagrangian relaxation bounding strategy, and report some results of numerical experiments using real aviation data. Computational results show the practical significance of the proposed capacitated model.

Keywords: Facility planning, hub location, integer programming

1. Introduction

The hub location problem has attracted much attention since O'Kelly [18] formulated the single allocation hub-and-spoke model as a quadratic integer program. Campbell [4] classifies the p-hub location problem into four classes, and presented basic formulations and formulations with flow thresholds for spokes for each of them. The four classes are the p-hub median problem, the p-hub center problem, the uncapacitated hub location problem and the hub covering problem. Besides the basic single allocation and multiple allocation models, Campbell [4] presents models involving flow thresholds for spokes, and points out that the p-hub median problem and the uncapacitated hub location problem have mainly been studied and other classes of problems have not been considered seriously. Skorin-Kapov, Skorin-Kapov and O'Kelly [25] considered the uncapacitated p-hub median problem and developed linear programming formulations of both single and multiple allocation models that yield tight relaxations to the original problem. For the same class of problems, O'Kelly et al. [20] introduced a more compact formulation and discussed sensitivity analysis of the model with respect to various parameters. On the other hand, Aykin [1] formulated the capacitated hub-and-spoke model as a 0-1 integer programming problem and proposed two heuristics to solve the problem. Ebery et al. [11] also addressed the capacitated multiple allocation hub location problem that arises in a postal delivery system.

The above mentioned models are of two-stop type in the sense that a trip between each origin-destination (OD) pair uses at most two hubs. Therefore, the number of variables and constraints in the problem rapidly grows with the size of the model and hence we may expect to solve problems of practical size only approximately. Sasaki, Suzuki and Drezner [23, 24] considered the uncapacitated one-stop model, in which a trip between any OD pair uses only one hub. The one-stop model appears to be useful in certain situations and is simple enough to be solved exactly. In the multiple allocation two-stop model, if we assume that there is no discount between hubs and the triangle inequality holds for link travel costs, then all trips will use a one-stop service. Therefore, a one-stop model may be regarded as a special case of the multiple allocation two-stop model [19]. However, it is not always true that the triangle inequality holds unless the link travel cost is proportional to the link length. Moreover, we can afford a significant reduction in the computational time by using directly the one-stop model compared with treating it as a special case of the two-stop model [24]. This is because the formulation is developed by taking into account the special structure of the one-stop models. Thus, not only is it useful in certain situations, the one-stop model is one of the important models from a computational viewpoint. In this paper, we propose a capacitated one-stop model as a natural extension of the uncapacitated one-stop model of [23, 24] and present a branch-and-bound based exact solution method with Lagrangian relaxation bounding strategy. Specifically, we introduce capacity constraints on both arcs and hubs. Similar arc capacitated network design problem have been considered in [13, 14]. However, studies on arc capacitated hub location problems are scarce. Hub capacity may be regarded as the maximum number of passengers a node can deal with when it is selected as a hub. For example, the number of runways and the number of slots assigned to the airline company may constitute the hub capacity. On the other hand, arc capacity may represent the number of available aircrafts for the airline company on that arc. By introducing arc capacity constraints as well as hub capacity constraints, we can enrich the model so as to take into account more practical situations. Bryan [2] first incorporated arc capacities into a hub location model. The model is an extension of the model proposed by O'Kelly et al. [21], which uses a piecewise linear cost function. Assuming hub locations were given, they only focused on the allocation phase and examined how the arc capacity levels affect the total network cost.

This paper is organized as follows. In Section 2, we formulate the capacitated one-stop hub-and-spoke model. In Section 3, we present a branch-and-bound method for solving the proposed model and particularly describe Lagrangian relaxation strategy for finding good lower bounds for partial problems. In Section 4, we report some numerical experience. In Section 5, we give concluding remarks and briefly mention some future work.

2. Capacitated One-Stop Model

2.1. Model description

Jaillet et al. [15] propose an airline network design problem. They examined the resultant network structure and figured out that the one-stop policy could bring results as good as those of two-stop policy, especially in the situation with relatively high level of demands. In some countries like Japan, a majority of domestic routes are provided by one-stop services via a hub. Even in the United States, a large number of trips seem to use one-stop routes. Motivated by these observations, an uncapacitated one-stop model has been proposed in [23, 24]. In this paper, we generalize this model to construct a capacitated one-stop model and formulate it as a 0-1 mixed integer programming problem. The problem is a NP-hard problem since the uncapacitated one-stop problem, which can be transformed into the p-median problem, is a NP-hard problem [16]. We use a branch-and-bound method with Lagrangian relaxation bounding strategy to obtain an exact optimal solution.

Aykin [1] formulated the capacitated two-stop hub-and-spoke model and proposed two heuristics for the problem. In the model, capacity constraints on hubs are primarily considered. However, since airline companies may sometimes fail to provide flight service to all passengers who want to use some arc, it is also practically significant to consider capacity constraints on arcs as well as on hubs. Concerning route patterns, we suppose that

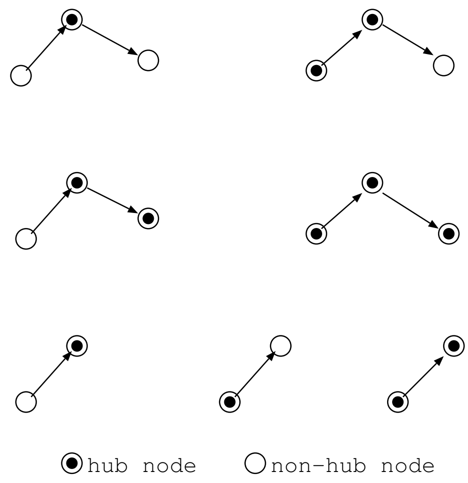


Figure 1: Route Patterns

nonstop services between non-hub nodes are not available, while passengers whose origin or destination is a hub are permitted to travel via another hub. Possible seven route patterns according to this rule are shown in Figure 1.

One of the major factors that characterize a hub location model is its allocation rule. Basically, there are two possible choices of allocation rules; the single allocation rule and the multiple allocation rule [3]. Allocating each demand node to more than one hubs is prohibited in the former rule and allowed in the latter rule. The first hub-and-spoke model presented by O'Kelly [18] adopts the single allocation rule. Recent papers often adopt the multiple allocation rule because it allows more flexible routing between OD pairs [5, 8, 20], while some others adopt single allocation rule [7, 10]. Skorin-Kapov et al. [25] and Ernst et al. [9] employ both rules in the uncapacitated p-hub median problem. These models are all two-stop models. The choice of the allocation rule makes a sensible difference in a two-stop model. However, in a one-stop model like the presented model, it is necessary to adopt the multiple allocation rule. Because, if we adopt the single allocation rule in a one-stop model, then any two demand nodes allocated to different hubs are not reachable.

The routing rule is another important factor in the model. There are also two possible choices of routing rules; the single routing rule and the multiple routing rule. Only one route service provided for each OD pair in the former rule and more than one route services are allowed in the latter rule. In other words, all passengers whose origin and destination are the same have to travel using the same route in the single routing rule. We note that the multiple routing rule is meaningful only in a capacitated problem, because in an uncapacitated problem, all passengers for each OD pair will use the least cost route even if multiple routes are available.

In this paper, we adopt the multiple routing rule. The objective of the model is to find hub locations and passenger routing for each OD pair that minimize the total transportation and location costs subject to the constraints derived from the assumed rules.

2.2. Model formulation

We now present a mathematical formulation of the hub location problem with the multiple allocation and multiple routing rules. The following notation is employed:

N : the set of demand nodes, $|N| = n$.
A : the set of arcs.
$H \subseteq N$: the set of hub candidates, $|H| = h \leq n$.

Π : the set of OD pairs $\pi = \langle i, j \rangle, i \in N, j \in N, i \neq j$. $|\Pi| = n(n-1)$.
$A_i \subseteq \Pi$: the set of OD pairs whose origin is $i \in N$.
$B_j \subseteq \Pi$: the set of OD pairs whose destination is $j \in N$.
p : the number of hubs to be selected.
d_π : the trip demand (the number of passengers) for OD pair $\pi \in \Pi$.
$c_{\pi k}$: the travel cost per passenger between OD pair $\pi \in \Pi$ via hub $k \in H$.
f_k : the fixed cost incurred by selecting hub k.
a_k : capacity of hub $k \in H$.
b_{ij}: capacity of arc $(i, j) \in A$.

Decision variables in the model are as follows:

$x_{\pi k}$: the number of passengers who travel between OD pair $\pi \in \Pi$ via hub $k \in H$.
y_k : binary variable such that $y_k = 1$ if node k is selected as a hub, and 0 otherwise.

We note that $\langle i, j \rangle$ denotes the OD pair whose origin is node i and destination is node j, while (i, j) denotes the arc connecting node i and node j. For an OD pair $\pi = \langle i, j \rangle$, if $i \in H$, then $x_{\pi i}$ is the number of passengers traveling directly from hub candidate i to node j. Similarly, if $j \in H$, then $x_{\pi j}$ is the number of passengers traveling directly from node i to hub candidate j. Moreover, if $i \in H$ and $j \in H$ hold simultaneously, then both $x_{\pi i}$ and $x_{\pi j}$ are the number of passengers from i to j. In this case, the total number of passengers on the route consisting of the single arc (i, j) is given by $x_{\pi i} + x_{\pi j}$.

Now we are ready to formulate the model as the following mixed 0–1 integer programming problem.

$$
\begin{aligned}
P_0 : \quad & \text{minimize} && \sum_{\pi \in \Pi} \sum_{k \in H} c_{\pi k} x_{\pi k} + \sum_{k \in H} f_k y_k && && (1)\\
& \text{subject to} && \sum_{k \in H} x_{\pi k} = d_\pi, && \pi \in \Pi, && (2)\\
& && \sum_{\pi \in A_i} x_{\pi j} + \sum_{\pi \in B_j} x_{\pi i} \leq b_{ij}, && i, j \in H, i \neq j, && (3)\\
& && \sum_{\pi \in A_i} x_{\pi j} \leq b_{ij}, && i \notin H, j \in H, i \neq j, && (4)\\
& && \sum_{\pi \in B_j} x_{\pi i} \leq b_{ij}, && i \in H, j \notin H, i \neq j, && (5)\\
& && \sum_{k \in H} y_k = p, && && (6)\\
& && \sum_{\pi \in \Pi} x_{\pi k} \leq a_k y_k, && k \in H, && (7)\\
& && x_{\pi k} \geq 0, && \pi \in \Pi, k \in H, &&\\
& && y_k \in \{0, 1\}, && k \in H. &&
\end{aligned}
$$

The objective function (1) is the sum of the total travel costs and fixed costs. Constraints (2) imply that each passenger between OD pair π travels via one of the hubs. Constraints (3)–(5) represent arc capacity constraints; (3) applies to the case where both end points of arc (i, j) belong to the hub candidate set, while (4) and (5) apply to the case where one of the end points belongs to the hub candidate set. Constraint (6) requires that exactly p hubs have to be selected. Constraints (7) represent capacity constraints for hubs, implying that, when hub k is selected, the number of passengers who use hub k cannot exceed the hub capacity a_k. Constraints (7) also assure that any node cannot be connected to non-hub nodes. Namely, $x_{\pi k}$ for all $\pi \in \Pi$ are forced to vanish unless $y_k = 1$.

3. Branch-and-Bound Method

3.1. Lagrangian relaxation

We note that P_0 can naturally be decomposed into two problems by fixing an arbitrary y_k at 0 or 1. We call such variable y_k and the generated problems a branching variable and partial problems, respectively. A partial problem P_ν may have one of the following two properties:

1. An optimal solution of P_ν is obtained.
2. It is detected that P_ν cannot produce an optimal solution of P_0.

In either case, we can fathom P_ν, because it is not necessary to decompose P_ν further. To check whether or not P_ν has property 2, a lower bound test is applied to P_ν. The branch-and-bound method repeatedly applies decomposition and lower bound tests until there remain no partial problems that are active, i.e., generated but neither fathomed nor decomposed.

To describe a partial problem P_ν, we partition the hub candidate set H as $H = H_\nu^F \cup H_\nu^1 \cup H_\nu^0$, where H_ν^F, H_ν^1 and H_ν^0 are specified as follows.

H_ν^F : the set of indices k such that y_k is a free 0-1 variable.
H_ν^1 : the set of indices k such that y_k is fixed as $y_k = 1$.
H_ν^0 : the set of indices k such that y_k is fixed as $y_k = 0$.

For the partial problem P_ν, either $|H_\nu^1| = p$ or $|H_\nu^1| < p$ holds. If $|H_\nu^1| = p$, then P_ν becomes a linear programming problem by fixing all free variables to be zero. We can then obtain an upper bound of the optimal value of P_0 by solving this problem, since its optimal solution is feasible to P_0. If this upper bound is smaller than an incumbent value, we let it be a new incumbent value of P_0.

If $|H_\nu^1| < p$, then we attempt to obtain a good lower bound for the optimal value of P_ν. Let us denote $H_\nu^+ = H_\nu^F \cup H_\nu^1$. Note that we eliminate variables $x_{\pi k}, y_k, k \in H_\nu^0$ from P_ν because it immediately implies that $x_{\pi k} = 0$ for all $\pi \in \Pi$ if $k \in H_\nu^0$. Then partial problem P_ν can be written as follows.

$$
\begin{array}{llll}
P_\nu : & \text{minimize} & \displaystyle\sum_{\pi\in\Pi}\sum_{k\in H_\nu^+} c_{\pi k}x_{\pi k} + \sum_{k\in H_\nu^1} f_k + \sum_{k\in H_\nu^F} f_k y_k & \\
& \text{subject to} & \displaystyle\sum_{k\in H_\nu^+} x_{\pi k} = d_\pi, & \pi \in \Pi, \\
& & \displaystyle\sum_{\pi\in A_i} x_{\pi j} + \sum_{\pi\in B_j} x_{\pi i} \le b_{ij}, & i,j \in H_\nu^+, i \ne j, \quad (8) \\
& & \displaystyle\sum_{\pi\in A_i} x_{\pi j} \le b_{ij}, & i \notin H_\nu^+, j \in H_\nu^+, \quad (9) \\
& & \displaystyle\sum_{\pi\in B_j} x_{\pi i} \le b_{ij}, & i \in H_\nu^+, j \notin H_\nu^+, \quad (10) \\
& & \displaystyle\sum_{k\in H_\nu^F} y_k = p - |H_\nu^1|, & \\
& & \displaystyle\sum_{\pi\in\Pi} x_{\pi k} - a_k y_k \le 0, & k \in H_\nu^F, \quad (11) \\
& & \displaystyle\sum_{\pi\in\Pi} x_{\pi k} \le a_k, & k \in H_\nu^1, \\
& & x_{\pi k} \ge 0, & \pi \in \Pi, k \in H_\nu^+, \\
& & y_k \in \{0,1\}, & k \in H_\nu^F.
\end{array}
$$

We consider relaxing all the arc capacity constraints and the hub capacity constraints (11) by bringing them into the objective function. To achieve this, we introduce Lagrange multipliers $\lambda = (\lambda_{ij})$ and $\mu = (\mu_k)$ corresponding to the arc capacity constraints (8)–(10) and the hub capacity constraints (11), respectively. Then we obtain the following Lagrangian relaxation problem $RP_\nu[\lambda, \mu]$ for partial problem P_ν.

$$
\begin{aligned}
RP_\nu[\lambda,\mu]: \quad \text{minimize} \quad & \sum_{\pi\in\Pi}\sum_{k\in H_\nu^+} c_{\pi k}x_{\pi k} + \sum_{k\in H_\nu^F} f_k y_k \\
& + \sum_{i\in H_\nu^+}\sum_{j\in H_\nu^+, j\neq i} \lambda_{ij}\Big(\sum_{\pi\in A_i} x_{\pi j} + \sum_{\pi\in B_j} x_{\pi i}\Big) \\
& + \sum_{i\notin H_\nu^+}\sum_{j\in H_\nu^+} \lambda_{ij}\sum_{\pi\in A_i} x_{\pi j} + \sum_{i\in H_\nu^+}\sum_{j\notin H_\nu^+} \lambda_{ij}\sum_{\pi\in B_j} x_{\pi i} \\
& + \sum_{k\in H_\nu^F} \mu_k\Big(\sum_{\pi\in\Pi} x_{\pi k} - a_k y_k\Big) + L_{\nu 0}(\lambda) \\
\text{subject to} \quad & \sum_{k\in H_\nu^+} x_{\pi k} = d_\pi, && \pi\in\Pi, \\
& \sum_{k\in H_\nu^F} y_k = p - |H_\nu^1|, \\
& \sum_{\pi\in\Pi} x_{\pi k} \le a_k, && k\in H_\nu^1, \\
& x_{\pi k} \ge 0, && \pi\in\Pi, k\in H_\nu^+, \\
& y_k \in \{0,1\}, && k\in H_\nu^F,
\end{aligned}
$$

where

$$L_{\nu 0}(\lambda) = \sum_{k\in H_\nu^1} f_k - \sum_{i\notin H_\nu^0}\sum_{j\notin H_\nu^0, j\neq i} \lambda_{ij}b_{ij}.$$

It is well-known that the optimal value of $RP_\nu[\lambda, \mu]$ provides a lower bound of the optimal value of P_ν for arbitrary $\lambda \ge 0$ and $\mu \ge 0$. We can obtain an optimal solution of $RP_\nu[\lambda, \mu]$ easily by decomposing it into some simple problems. First, note that the objective function of $RP_\nu[\lambda, \mu]$ can be rewritten as

$$
\begin{aligned}
& \sum_{\pi\in\Pi}\sum_{k\in H_\nu^+} c_{\pi k}x_{\pi k} + \sum_{\pi\in\Pi}\sum_{k\in H_\nu^F} \mu_k x_{\pi k} + \sum_{i\in H}\sum_{j\in H_\nu^+, j\neq i} \lambda_{ij}\sum_{\pi\in A_i} x_{\pi j} + \sum_{i\in H_\nu^+}\sum_{j\in H, j\neq i} \lambda_{ij}\sum_{\pi\in B_j} x_{\pi i} \\
& + \sum_{k\in H_\nu^F} (f_k - \mu_k a_k) y_k + L_{\nu 0}(\lambda). \qquad (12)
\end{aligned}
$$

Let α_π and β_π denote the origin and the destination of OD pair π, respectively. Then the third term of (12) can be rewritten as

$$\sum_{\pi\in\Pi}\sum_{j\in H_\nu^+, j\neq\alpha_\pi} \lambda_{\alpha_\pi j}x_{\pi j}. \qquad (13)$$

Similarly, the fourth term of (12) can be rewritten as

$$\sum_{\pi\in\Pi}\sum_{i\in H_\nu^+, i\neq\beta_\pi} \lambda_{i\beta_\pi}x_{\pi i}. \qquad (14)$$

By substituting (13) and (14) into (12), we may rewrite $RP_\nu[\lambda,\mu]$ as follows.

$RP_\nu[\lambda,\mu]$:

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{\pi\in\Pi}\left(\sum_{k\in H_\nu^+} c_{\pi k}x_{\pi k} + \sum_{k\in H_\nu^+,k\neq\alpha_\pi}\lambda_{\alpha_\pi k}x_{\pi k} + \sum_{k\in H_\nu^+,k\neq\beta_\pi}\lambda_{k\beta_\pi}x_{\pi k} + \sum_{k\in H_\nu^F}\mu_k x_{\pi k}\right) & \\
& \displaystyle + \sum_{k\in H_\nu^F}(f_k-\mu_k a_k)y_k + L_{\nu 0}(\lambda) & \\
\text{subject to} & \displaystyle\sum_{k\in H_\nu^+} x_{\pi k} = d_\pi, & \pi\in\Pi, \\
& \displaystyle\sum_{\pi\in\Pi} x_{\pi k} \le a_k, & k\in H_\nu^1, \\
& \displaystyle\sum_{k\in H_\nu^F} y_k = p - |H_\nu^1|, & \\
& x_{\pi k}\ge 0, & \pi\in\Pi, k\in H_\nu^+, \\
& y_k\in\{0,1\}, & k\in H_\nu^F.
\end{array}$$

Let us define $C_{\pi k}$ to represent the aggregate coefficient of $x_{\pi k}$:

$$C_{\pi k} = \begin{cases}
c_{\pi k}+\lambda_{\alpha_\pi k}+\lambda_{k\beta_\pi}+\mu_k, & \text{if } k\in H_\nu^F, k\neq\alpha_\pi, k\neq\beta_\pi \\
c_{\pi k}+\lambda_{\alpha_\pi k}+\mu_k, & \text{if } k\in H_\nu^F, k=\beta_\pi \\
c_{\pi k}+\lambda_{k\beta_\pi}+\mu_k, & \text{if } k\in H_\nu^F, k=\alpha_\pi \\
c_{\pi k}+\lambda_{\alpha_\pi k}, & \text{if } k\in H_\nu^1, k=\beta_\pi \\
c_{\pi k}+\lambda_{k\beta_\pi}, & \text{if } k\in H_\nu^1, k=\alpha_\pi.
\end{cases}$$

Then, problem $RP_\nu[\lambda,\mu]$ can be decomposed into the linear programming problem:

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{\pi\in\Pi}\sum_{k\in H_\nu^+} C_{\pi k}x_{\pi k} & \\
\text{subject to} & \displaystyle\sum_{k\in H_\nu^+} x_{\pi k} = d_\pi, & \pi\in\Pi, \\
& \displaystyle\sum_{\pi\in\Pi} x_{\pi k}\le a_k, & k\in H_\nu^1, \\
& x_{\pi k}\ge 0, & k\in H_\nu^+,
\end{array} \tag{15}$$

and the 0–1 integer programming problem:

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{k\in H_\nu^F}(f_k-\mu_k a_k)y_k & \\
\text{subject to} & \displaystyle\sum_{k\in H_\nu^F} y_k = p-|H_\nu^1|, & \\
& y_k\in\{0,1\}, & k\in H_\nu^F.
\end{array} \tag{16}$$

Note that the linear programming problem (15) reduces to a transportation problem, if we regard the nodes $k\in H_\nu^+$ as source nodes and OD pairs $\pi\in\Pi$ as sink nodes. Moreover, an optimal solution $\bar{y}_k, k \in H_\nu^F$, of the 0–1 integer programming problem (16) is obtained in the following simple way: Order the indices $k \in H_\nu^F$ as $k_1, k_2, \cdots, k_{|H_\nu^F|}$ in the increasing order of the coefficients $f_k - \mu_k a_k$, and let

$$\bar{y}_k = \begin{cases} 1, & \text{if } k \in K, \\ 0, & \text{if } k \notin K, \end{cases} \tag{17}$$

where $K = \{k_1, k_2, \cdots, k_{p-|H_\nu^1|}\}$.

3.2. Subgradient method

It is practically important to use appropriate values of Lagrange multipliers so as to obtain a good lower bound for the optimal value of the partial problem P_ν. The optimal Lagrange multipliers can be obtained by solving the following Lagrangian dual problem for P_ν.

$$\begin{aligned} LD_\nu: \quad & \text{maximize} \quad G_\nu(\lambda, \mu) \\ & \text{subject to} \quad \lambda, \mu \geq 0, \end{aligned}$$

where $G_\nu(\lambda, \mu)$ denotes the optimal value of $RP_\nu[\lambda, \mu]$. This is a problem of maximizing a nondifferentiable concave function. Since it is expensive to obtain an optimal solution of LD_ν exactly, we use the subgradient method, which is practically useful because it can find an approximate solution of LD_ν conveniently.

The subgradient method for solving LD_ν consists of the following iterative process: Choose initial Lagrange multipliers $\lambda^{(0)} \geq 0, \mu^{(0)} \geq 0$, and then successively update $(\lambda^{(l)}, \mu^{(l)})$ by

$$\begin{aligned} \lambda^{(l+1)} &= \max\{0, \lambda^{(l)} + \theta^{(l)} g_\lambda^{(l)}\}, \\ \mu^{(l+1)} &= \max\{0, \mu^{(l)} + \theta^{(l)} g_\mu^{(l)}\}, \end{aligned} \tag{18}$$

where $\theta^{(l)} > 0$ is a stepsize and $(g_\lambda^{(l)}, g_\mu^{(l)})$ is a subgradient of G_ν at $(\lambda^{(l)}, \mu^{(l)})$. Using an optimal solution $(\bar{x}^{(l)}, \bar{y}^{(l)})$ of $RP_\nu[\lambda^{(l)}, \mu^{(l)}]$, where $\bar{x}^{(l)}$ is an optimal solution of the linear programming problem (15) with $(\lambda, \mu) = (\lambda^{(l)}, \mu^{(l)})$ and $y^{(l)}$ is determined by (17), we may compute each element of the subgradient $(g_\lambda^{(l)}, g_\mu^{(l)})$ as

$$\left[g_\lambda^{(l)}\right]_{ij} = \begin{cases} \displaystyle\sum_{\pi \in A_i} \bar{x}_{\pi j}^{(l)} + \sum_{\pi \in B_j} \bar{x}_{\pi i}^{(l)} - b_{ij}, & \text{if } i \in H_\nu^+ \text{ and } j \in H_\nu^+, \\ \displaystyle\sum_{\pi \in A_i} \bar{x}_{\pi j}^{(l)} - b_{ij}, & \text{if } i \notin H_\nu^+ \text{ and } j \in H_\nu^+, \\ \displaystyle\sum_{\pi \in B_j} \bar{x}_{\pi i}^{(l)} - b_{ij}, & \text{if } i \in H_\nu^+ \text{ and } j \notin H_\nu^+, \end{cases}$$

$$\left[g_\mu^{(l)}\right]_k = \sum_{\pi \in \Pi} \bar{x}_{\pi k}^{(l)} - a_k \bar{y}_k^{(l)}, \quad k \in H_\nu^F.$$

Stepsize $\theta^{(l)}$ is determined according to the rule

$$\theta^{(l)} = \frac{\xi(\hat{G}_\nu - G_\nu(\lambda^{(l)}, \mu^{(l)}))}{\|(g_\lambda^{(l)}, g_\mu^{(l)})\|^2}, \tag{19}$$

where $\hat{G}_\nu$ is an estimate of the optimal value of LD_ν, and ξ is a parameter satisfying $\epsilon \le \xi \le 2 - \epsilon$ for some $\epsilon > 0$ [22]. If we use the exact optimal value G_ν^* of LD_ν instead of $\hat{G}_\nu$, the sequence generated by (18) converges to an optimal solution of LD_ν. However, the optimal value of LD_ν is usually unknown. So we iterate (18) with (19) using an appropriate estimate $\hat{G}_\nu$ of the optimal value G_ν^*. By using an appropriate adjusting strategy, we may expect that $\hat{G}_\nu$ approaches G_ν^* and $(\lambda^{(l)}, \mu^{(l)})$ converges to an optimal solution of LD_ν.

As a matter of fact, the convergence rate of subgradient method significantly depends on the strategy for updating an estimate of the optimal value. For simplicity, we take $\xi = 1$ in the following discussion. Suppose that for all $\hat{G}_\nu > G_\nu^*$ and $0 < \zeta < 1$ there is a positive integer S such that

$$l \ge S \implies G_\nu^{\max} - (2G_\nu^* - \hat{G}_\nu) \ge \zeta \left[G_\nu(\lambda^{(0)}, \mu^{(0)}) - (2G_\nu^* - \hat{G}_\nu) \right], \tag{20}$$

where $G_\nu^{\max}$ is the maximum objective value of LD_ν obtained up to the l-th iteration with the initial point $(\lambda^{(0)}, \mu^{(0)})$. Then a possible procedure to update the estimated value $\hat{G}_\nu$ is described as follows [17]:

[Updating $\hat{G}_\nu$]

Input: $\zeta \in (0, 1), S \in \{1, 2, \cdots\}$.

Step 0: Let $\hat{G}_\nu$ be an arbitrary upper bound of G_ν^*. Select an initial point $(\lambda^{(0)}, \mu^{(0)})$. Set $l := 0$.

Step 1: Set $G_\nu^{\max} := -\infty$.

Step 2: If $\mathrm{mod}(l + 1, S) \neq 0$ then go to Step 3. Otherwise, go to Step 4.

Step 3: If $G_\nu(\lambda^{(l)}, \mu^{(l)}) > G_\nu^{\max}$, then update $G_\nu^{\max}$ as $G_\nu^{\max} := G_\nu(\lambda^{(l)}, \mu^{(l)})$, and let $\lambda^{\max} := \lambda^{(l)}$ and $\mu^{\max} := \mu^{(l)}$. Compute $\lambda^{(l+1)}, \mu^{(l+1)}$ according to (18)–(19). Set $l := l + 1$ and go to Step 2.

Step 4: Update the estimated value $\hat{G}_\nu$ as $\hat{G}_\nu := \dfrac{G_\nu^{\max} + (1-\zeta)\hat{G}_\nu - \zeta G_\nu(\lambda^{(0)}, \mu^{(0)})}{2 - 2\zeta}$. Set $\lambda^{(l+1)} := \lambda^{\max}$, $\mu^{(l+1)} := \mu^{\max}$, $l := l + 1$ and go to Step 1.

With this strategy, $\hat{G}_\nu > G_\nu^*$ always holds because of (20) and hence $\hat{G}_\nu$ is expected to decrease steadily and converge to G_ν^*. If $G_\nu^{\max} - \hat{G}_\nu$ becomes smaller than a predetermined tolerance, then we stop the iterations. In general, it is impossible to obtain the exact values of ζ and S. However, it is seen that S gets smaller as ζ gets larger because of (20). From some preliminary computations, we have decided to set $\zeta = 0.5$ and $S = 5$ in our computational experiments.

Although we may use an arbitrary upper bound of G_ν^* as an initial estimate of the optimal value of LD_ν in Step 0, it is not always easy to obtain it. In the above procedure, we use an arbitrary upper bound for P_ν rather than G_ν^*, because any upper bound for P_ν is greater than or equal to G_ν^*. We note that the uncapacitated one-stop hub location problem can be regarded as a shortest path problem if any p hubs are fixed. Therefore, for an uncapacitated problem, we can easily obtain an upper bound by solving the shortest path problem. For a hub capacitated problem, we can also find a feasible flow by solving a shortest path problem, if we select p hubs so that the total hub capacity is greater than the total demand. Thus, we can obtain an upper bound of a hub capacitated problem by allocating flows appropriately to each arc. However, it is not necessarily easy to obtain an upper bound for P_ν in the case where both hubs and arcs are capacitated. The reason is that an arc capacitated problem often becomes infeasible even if we select p hubs so as to satisfy the above-mentioned total capacity condition. In general, any feasible solution of P_ν is also feasible to P_0, though the converse does not always hold. However, if a feasible solution of P_0 is still feasible to P_ν, its objective value also provides an upper bound of P_ν. Therefore we keep a list of any feasible solutions to P_0 obtained during the branch-and-bound computation, and try to find a feasible solution of P_ν from the list so as to use it as an initial estimated value of LD_ν. If there are more than one feasible solutions to P_ν, then we may choose the one with smallest objective value. On the other hand, if the list contains no such feasible solution for P_ν, then we try to obtain a new upper bound by solving a new shortest path problem that is constructed from P_ν by adding $p-|H_\nu^1|$ hubs arbitrarily chosen form H_ν^F. If the new shortest path problem is feasible, we use its objective value as an initial estimated value and keep the solution for the subsequent computation. Otherwise, we let $\hat{G}_\nu$ be a tentative estimated value, which is chosen to be greater than the current incumbent value z^* of P_0. We note that there is no guarantee that such a tentative estimated value is an upper bound of G_ν^*. If $\hat{G}_\nu < G_\nu^*$, then $G_\nu(\lambda^{(l)}, \mu^{(l)})$ will converge to a smaller value than G_ν^* [17]. Nevertheless, we may fathom the partial problem P_ν if we can find a $(\lambda^{(l)}, \mu^{(l)})$ such that $G_\nu(\lambda^{(l)}, \mu^{(l)})$ is greater than the incumbent value z^*. After fathoming a partial problem and backtracking to another partial problem, some feasible solutions of P_0 in the list may become invalid in the sense that they can never be feasible to any partial problem generated in the subsequent computation. By removing such invalid data, we can reduce the area of searching for feasible solutions of a partial problem.

3.3. Upper bound

Upper bounds for the optimal value of problem P_0 play an important role in developing a branch-and-bound method. An upper bound can be obtained by finding any feasible solution of P_0, which is not always easy as described in the previous subsection. However, since we cannot expect to fathom any partial problem without knowing an upper bound for P_0, we need to obtain it before getting into a lower bound computation. In addition, whether we can obtain a good upper bound in the early stage of iterations has a significant effect on the performance of the branch-and-bound method.

First we note that any partial problem P_ν as well as P_0 reduces to an arc capacitated network flow problem if we select p hubs. In general, it is more likely to be able to find a feasible flow of the arc capacitated network flow problem by selecting hub candidates with larger capacities. However, selecting such hubs is generally costly because hubs with large capacities usually incur a high fixed cost. On the other hand, to obtain a good lower bound, it seems effective to select those hub candidates which are expected to be selected at optimal solutions. Consequently, we use the following strategy to obtain an upper bound. First we select p hubs according to an appropriate criterion, then solve the resultant arc capacitated network flow problem. If this problem is feasible, the obtained objective value is an upper bound. Here we introduce the following criterion as a likelihood of each hub candidate to be selected as a hub at an optimal solution:

$$h(k) = \gamma \frac{f_k}{a_k} + \sum_{\pi \in \Pi} c_{\pi k}, \qquad k \in H,$$

where $\gamma > 0$ is a parameter. The first term represents the fixed cost per unit capacity multiplied by a positive parameter γ and the second term represents the total travel cost when all passengers use the hub k. Note that we do not consider capacity constraints here. By introducing this criterion, a hub candidate associated with small $h(k)$ is likely to be selected as a hub at an optimal solution. We sort $h(k)$ in increasing order and express the ordered indices of $h(k)$ as $\kappa_1, \kappa_2, \cdots, \kappa_{|H|}$. If the arc capacitated network flow problem in which hubs $\{\kappa_1, \kappa_2, \cdots, \kappa_p\}$ are selected is feasible, we use the objective value as an initial upper bound. Otherwise, we select p hubs with p largest a_k's and solve the resultant network flow problem. If this problem is also infeasible, we proceed to branch-and-bound iterations without knowing any upper bound at this stage.

3.4. Branching strategies

The computational efficiency of branch-and-bound method significantly depends on a branching strategy to select a branching variable in each stage to decompose the current partial problem into new partial problems. Selecting a variable whose value is likely to be 1 at an optimal solution is one of the practical strategies. However, when we proceed to branch-and-bound iterations without finding an upper bound for P_0, we need to obtain at least one feasible solution of P_0 as fast as possible. In such a case, we therefore employ a different strategy.

When we do not have any feasible solution of P_0, we select the index

$$\tilde{k} = \arg \max_{k \in H_\nu^F} a_k \tag{21}$$

as the next branching variable with the hope to obtain a feasible solution. In this case, we just decompose the current partial problem P_ν by fixing $y_{\tilde{k}}$ to 0 and 1, and generate two new partial problems. We do not compute a lower bound because an upper bound is unknown. If $|H_\nu^1| = p$, this partial problem P_ν becomes an arc capacitated network flow problem and we solve it exactly.

Once we find a feasible solution of P_0, we select the index

$$\tilde{k} = \arg \max_{k \notin K} \sum_{\pi \in \Pi} \bar{x}_{\pi k} \tag{22}$$

as the next branching variable. Note that when we solve a Lagrangian relaxation problem $RR_\nu[\lambda, \mu]$ of P_ν, we obtain as a byproduct an exact solution $\bar{y}_k$ as shown in (17). If some $k \notin K$ violate the capacity constraints (11) of P_ν, selecting y_k that corresponds to one of such hubs k as a branching variable may yield a good feasible solution to P_ν. By (22), we actually select y_k such that hub $k \notin K$ violates the capacity constraints (11) of P_0 most seriously.

Now we summarize the branch-and-bound method for solving P_0, which uses the depth-first search rule. We let $\mathcal{A}$ denote the set of active partial problems, i.e., partial problems that have been generated but not tested yet.

[Branch-and-Bound Method]

Step 0: (Initialize) Find an optimal solution (x^*, y^*) of the arc capacitated network flow problem, in which candidates $\kappa_1, \kappa_2, \cdots, \kappa_p$ are selected as hubs. Let the optimal value of this problem be the incumbent value z^*. If this problem is infeasible, then find an optimal solution (x^*, y^*) of another arc capacitated network flow problem, in which candidates with p largest a_k's are selected as hubs. Let the optimal value of this problem be the incumbent value z^*. If this problem is infeasible, set $z^* = \infty$. Set $\nu := 0, H_\nu^F :=$

$H, H_\nu^1 := \emptyset, H_\nu^0 := \emptyset$ and $\mathcal{A} := \{P_0\}$. Select an integer parameter S, and real parameters $0 < \zeta < 1, \eta > 0$ and $\epsilon > 0$.

Step 1: (Search) If $\mathcal{A} = \emptyset$, then go to Step 7. Otherwise, select an active partial problem $P_{\nu+1} \in \mathcal{A}$ according to the depth-first search strategy. Set $\nu := \nu + 1$. If $H_\nu^1 = p$ or $H_\nu^0 = n - p$, then go to Step 2. If a feasible solution of P_0 is unknown, then go to Step 4. If $H_\nu^1 < p$, then go to Step 3.

Step 2: (Compute upper bound) If P_ν is infeasible, then go to Step 5. Find an optimal solution (x_ν^*, y_ν^*) of P_ν with the objective value z_ν^*. If $z_\nu^* < z^*$, then let $z^* := z_\nu^*, x^* := x_\nu^*, y^* := y_\nu^*$ and go to Step 5.

Step 3: (Subgradient method)

(a) Set $\lambda^{(0)} := 0$, $\mu^{(0)} := 0$, and $l := 0$. Let $\hat{G}_\nu$ be upper bound of P_ν. If an upper bound is not known, set $\hat{G}_\nu := \eta z^*$.

(b) Set $G_\nu^{\max} := -\infty$.

(c) Solve $RP_\nu[\lambda^{(l)}, \mu^{(l)}]$ to obtain an optimal solution $(\bar{x}, \bar{y})$, and optimal value G_ν^*. If $G_\nu^* > G_\nu^{\max}$, then update $G_\nu^{\max} := G_\nu^*$, $\lambda^{\max} := \lambda^{(l)}$ and $\mu^{\max} := \mu^{(l)}$. If $G_\nu^* > z^*$, then go to Step 5.

(d) If $\mathrm{mod}(l+1, S) = 0$, then go to Step 3(e). Otherwise, compute Lagrange multipliers $\lambda^{(l+1)}$ and $\mu^{(l+1)}$ according to (18)–(19). Set $l := l + 1$ and go to Step 3(c).

(e) Update $\hat{G}_\nu$ as $\hat{G}_\nu := \dfrac{G_\nu^{\max} + (1-\zeta)\hat{G}_\nu - \zeta G_\nu(\lambda^{(l-S+1)}, \mu^{(l-S+1)})}{2 - 2\zeta}$. If $\hat{G}_\nu < z^*$ or $|(\hat{G}_\nu - G_\nu^{\max})/\hat{G}_\nu| < \epsilon$, then go to Step 5. Otherwise, set $\lambda^{(l+1)} := \lambda^{\max}$, $\mu^{(l+1)} := \mu^{\max}, l := l + 1$ and go to Step 3(b).

Step 4: (Branching) If a feasible solution of P_0 is unknown, select the next branching variable $\tilde{k}$ according to (21). Otherwise, select the next branching variable $\tilde{k}$ according to (22).
Set $\mathcal{A} := \mathcal{A} \cup \{(H_\nu^1 \cup \{\tilde{k}\}, H_\nu^0, H_\nu^F \backslash \{\tilde{k}\}), (H_\nu^1, H_\nu^0 \cup \{\tilde{k}\}, H_\nu^F \backslash \{\tilde{k}\})\}$ and go to Step 1.

Step 5: (Backtrack) Set $\mathcal{A} := \mathcal{A} \backslash \{P_\nu\}$ and go to Step 1.

Step 6: (Terminate) If $z^* = \infty$, then P_0 is infeasible. Else, (x^*, y^*) is an optimal solution of P_0, and z^* is the optimal value.

4. Numerical Experiments

In this section, we report some computational results. All programs were coded in MATLAB R12.1 (version6.1) with optimization toolbox [6]. They were run on a DELL DIMENSION 4400 computer with Pentium4 2.0AGHz processor operated under Windows XP professional with 512 Mb DDR-SDRAM memory. We used the function linprog included in the optimization toolbox, which solves a linear programming problem using the interior point method.

We prepared the demand data based on the well-known U.S. 25 cities data evaluated in 1970 by CAB (Civil Aeronautics Board). The data set, however, does not contain node capacities and fixed costs, so we generated them based on the data reported by FAA (Federal Aviation Administration) [12]. They report the passenger enplanements in the fiscal year of 2000 and the passenger forecast in the fiscal year of 2012. In Table 1, these data are shown in the column labeled "(a) FY2000 Enplanements" and the column labeled "(b) FY2012 Forecast", respectively, and the growth forecast of passenger enplanements from 2000 to 2012 is shown in the column labeled "(b)−(a)". Moreover, $\tilde{f}_k$ denotes a scaled fixed cost obtained by setting the fixed cost of New Orleans to be 1.0. Similarly, $\tilde{a_k}$ denotes a scaled capacity obtained by setting the capacity of New Orleans to be 1.0.

Table 1: Scaled fixed cost $\tilde{f}_k$ and capacity $\tilde{a}_k$

hub#	City	(a)FY2000 Enplanements $\times 10^3$	(b)FY2012 Forecast $\times 10^3$	(b)−(a) $\times 10^3$	Fixed Cost $\tilde{f}_k$	Capacity $\tilde{a}_k$
1	Atlanta	39,375	59,353	19,978	11.69	8.98
2	Baltimore	10,618	16,497	5,879	3.44	2.50
3	Boston	13,816	18,695	4,879	2.86	2.83
4	Chicago	34,153	46,178	12,025	7.04	6.99
5	Cincinnati	9,186	18,749	9,563	5.60	2.84
6	Cleveland	6,746	10,935	4,189	2.45	1.65
7	Dallas-FW	28,066	41,759	13,693	8.01	6.32
8	Denver	18,884	27,541	8,657	5.07	4.17
9	Detroit	17,874	28,287	10,413	6.09	4.28
10	Houston	16,564	26,510	9,944	5.82	4.01
11	Kansas City	5,889	7,978	2,089	1.22	1.21
12	Los Angeles	32,332	47,294	14,962	8.76	7.16
13	Memphis	6,234	8,488	2,254	1.32	1.28
14	Miami	16,716	24,156	7,440	4.35	3.65
15	Minneapolis	17,203	25,914	8,711	5.10	3.92
16	New Orleans	4,900	6,609	1,709	1.00	1.00
17	New York	16,081	22,991	6,910	4.04	3.48
18	Philadelphia	13,023	20,748	7,725	4.52	3.14
19	Phoenix	18,652	31,252	12,600	7.37	4.73
20	Pittsburgh	10,521	14,319	3,798	2.22	2.17
21	St. Louis	1,4553	21,631	7,078	4.14	3.27
22	San Francisco	18,499	25,634	7,135	4.18	3.88
23	Seattle	14,225	20,854	6,629	3.88	3.16
24	Tampa	8,200	11,384	3,184	1.86	1.72
25	WashingtonDC	8,502	15,146	6,644	3.89	2.29

Since the data of node capacities are not available in the published documents, we determined the node capacity a_k for each candidate by

$$a_k = \frac{\sum_{\pi \in \Pi} \tilde{d}_\pi}{p \cdot \min_{k \in H} a_k} \tilde{a}_k, \quad \forall k \in H, \tag{23}$$

where $\tilde{d}_\pi$ is a scaled trip demand between OD pair π, that is $\tilde{d}_\pi = d_\pi / \min_{\pi \in \Pi} d_\pi$. In a similar way, we determined arc capacities as follows. First we calculate the scaled arc capacities by assigning node capacities a_k equally to all arcs connected to it. In the computational experiments, we assume that $H = N$, i.e., all nodes are candidate hubs. So the number of arcs connected to each hub candidate is $2(n-1)$. Consequently, $\tilde{b}_{ij}$ are calculated as

$$\tilde{b}_{ij} = \frac{a_i + a_j}{2(n-1)}, \qquad (i,j) \in A,$$

from which we determined arc capacities b_{ij} as

$$b_{ij} = \delta \tilde{b}_{ij}, \qquad (i,j) \in A,$$

where $\delta \geq 1$ is a parameter. It is clear that a problem with small δ has tight arc capacity constraints. Similarly, we determined f_k, the fixed cost of hub candidate k, as

$$f_k = \rho \tilde{f}_k, \qquad k \in H,$$

where $\rho > 0$ is a parameter. To see how the arc capacity constraints and fixed costs affect the total optimal cost and hub locations, we solved the problem with different values of δ and ρ.

The computational results are given in Tables 2 and 3. Table 2 shows the results with $\rho = 1000$, where the ratio of fixed cost to the total cost is slightly less than 5%. Table 3 shows the results with $\rho = 10000$, where the ratio of fixed cost to the total cost is approximately 10%. The column labeled "optimal hubs" and "optimal cost" show the set of optimal hubs and the optimal total cost, respectively. The column labeled "# problems solved" shows the number of partial problems P_ν solved exactly, i.e., the partial problems with $|H_\nu| = p$. The column labeled "B&B CPU time" shows the total CPU time required to solve the problem by the branch-and-bound method. The column labeled "% cost reduced" shows how the optimal cost reduces as δ is increased from $\delta = 1$.

A problem with small δ often contains many infeasible partial problems because of its tight arc capacity constraints. In such a case, it is generally difficult to obtain a good upper bound in the early stage of branch-and-bound iterations, and hence it takes much computation time. In fact, Tables 2 and 3 indicate that the computational time and the number of partial problems solved increase as δ decreases. The optimal cost decreases monotonically as δ increases, because arc capacity constraints become loose as δ increases. However, when $\rho = 1000$, the decrease of the optimal cost is less significant. We can observe that the difference of the optimal cost between problems with $\delta = 1$ and $\delta = 3$ is less than 2% regardless of the value of p. In particular, when $p = 4$, the difference is much smaller than that in the cases of $p = 2$ and $p = 3$. The reason for this phenomenon may be resorted to the fact that p is relatively large and the ratio of fixed cost to total cost is relatively small. Regardless of the value of δ, Kansas City (#11) is always selected as a hub when $p = 2$, and St. Louis (#21) and San Francisco (#22) are always selected when $p = 3$. Another selected hub changes in the order of New York (#17), Philadelphia (#18), Baltimore (#2) as δ increases. Note that these three cities are located closely to each other. These results indicate that the optimal hub location does not change substantially when $\rho = 1000$.

On the other hand, when $\rho = 10000$, hub locations are significantly affected by the value of δ. Table 3 shows that the maximum difference of the optimal cost between the two cases $\delta = 1$ and $\delta = 4$ is nearly 5%. Although the difference of 5% seems small, it brings a considerable amount of cost reduction when we deal with a large network. Figures 2∼4 display the results for problems with $n = 25$ and $p = 2, 3, 4$. The thickness of each arc represents flow congestion on that arc. These figures show that the optimal hub location varies with the value of δ. In particular, since the ratio of fixed costs to the total cost is relatively large compared with the case of $\rho = 1000$, a candidate with lower fixed cost is likely to be selected. As δ increases, the hub location changes from New York (#17) to Pittsburgh (#20), from Memphis (#13) to New Orleans (#16) and from New York (#17) to Boston (#3). These changes occur in such a way that a candidate is replaced by another candidate with lower fixed cost. However, when $p = 4$, the selected hub location is drastically

Table 2: Computational Results for $\rho = 1000$

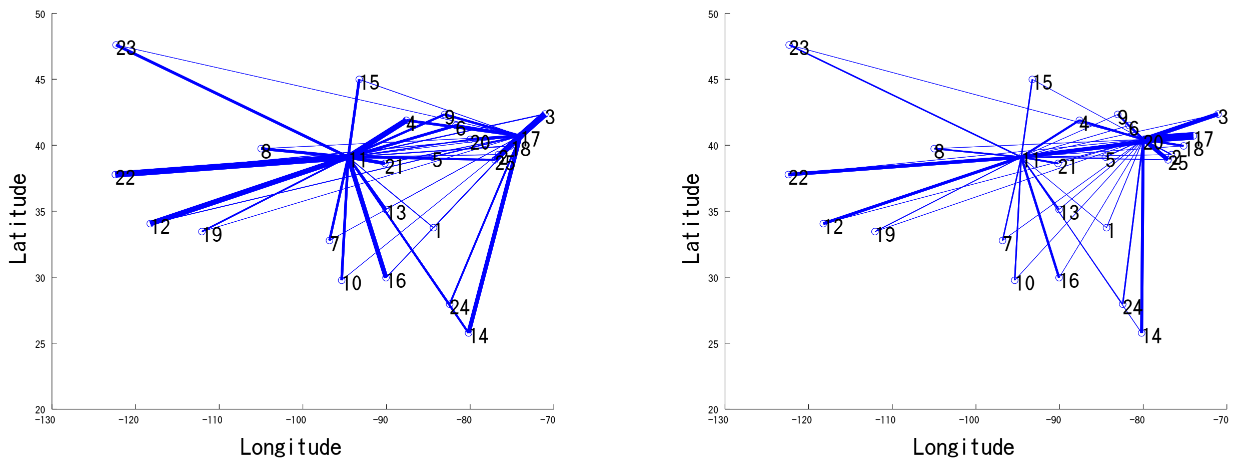
n	p	δ	Optimal hubs	Optimal cost($\times 10^5$)	#Problems solved	B & B CPU time	% Cost reduced
25	2	1	11,17	3.3289	171	102.31	–
		2	11,18	3.3044	174	62.59	0.74
		3	2,11	3.2746	172	61.28	1.63
25	3	1	17,21,22	2.9741	890	848.16	–
		2	17,21,22	2.9541	904	573.50	0.67
		3	18,21,22	2.9448	890	574.88	0.98
25	4	1	6,12,13,17	2.8762	3287	4433.56	–
		2	6,12,13,17	2.8571	3463	3400.27	0.66
		3	6,12,13,17	2.8566	3272	3248.89	0.68

Table 3: Computational Results for $\rho = 10000$

n	p	δ	Optimal hubs	Optimal cost($\times 10^5$)	#Problems solved	B & B CPU time	% Cost reduced
25	2	1	11,17	3.8023	174	108.64	–
		2	11,20	3.6991	166	59.86	2.72
		3	11,20	3.6374	160	57.88	4.34
		4	11,20	3.6373	157	57.45	4.34
25	3	1	11,13,17	3.8237	1047	1041.38	–
		2	11,16,17	3.7882	867	550.58	0.93
		3	11,16,20	3.6916	796	501.55	3.45
		4	11,16,20	3.6535	772	489.38	4.45
25	4	1	11,16,17,20	3.8790	2272	3630.61	–
		2	3,11,16,20	3.8569	2027	2148.84	0.57
		3	3,11,16,20	3.8286	1630	1714.69	1.30
		4	11,16,20,22	3.7695	1593	1648.42	2.82

changed from Boston (#3) to San Francisco (#22) in spite of the expensive fixed cost of San Francisco. Note that, when $p \leq 3$, no candidate located in the western area is selected. It becomes possible to select such a candidate when $\delta = 4$, because of capacity relaxation. By selecting a candidate located in the western area, we can reduce the cost of travel which originates from and destines for the western area. This travel cost reduction is large enough to compensate for the increase of fixed cost; thereby it induces the total cost reduction. Figure 4 shows that, when $p \leq 3$, the trips originating from Seattle, San Francisco and Los Angeles are long and the links emanating from these cities are congested.

From the above observation, we see that the total cost is affected by arc capacities, especially for problems in which the ratio of the fixed cost to the total cost is large, i.e., ρ is large. Taking into account the fact that the arc capacities are likely to be changed frequently by various surroundings, it is important to select a robust or stable candidate. To select such hubs, airlines may expect to have a constant revenue. When $\rho = 10000$, Kansas City is always selected regardless of the values of p and δ. Its good location in the central area and the cheap fixed cost may be the main reason why it is selected constantly. On the other hand, when $\rho = 1000$, there is no such stable hub candidate which is chosen independently of the value of p and δ. In this case, almost all candidates except for some very expensive ones are likely to become hubs, because the ratio of the fixed cost to the total cost is small. When $p = 2$, Kansas City (#11) is always selected regardless of the value of δ. Similarly, when $p = 3$, St. Louis (#21) is always selected. Moreover, when $p = 4$, selected hubs remain the same completely. We may regard such hubs as stable for each p.



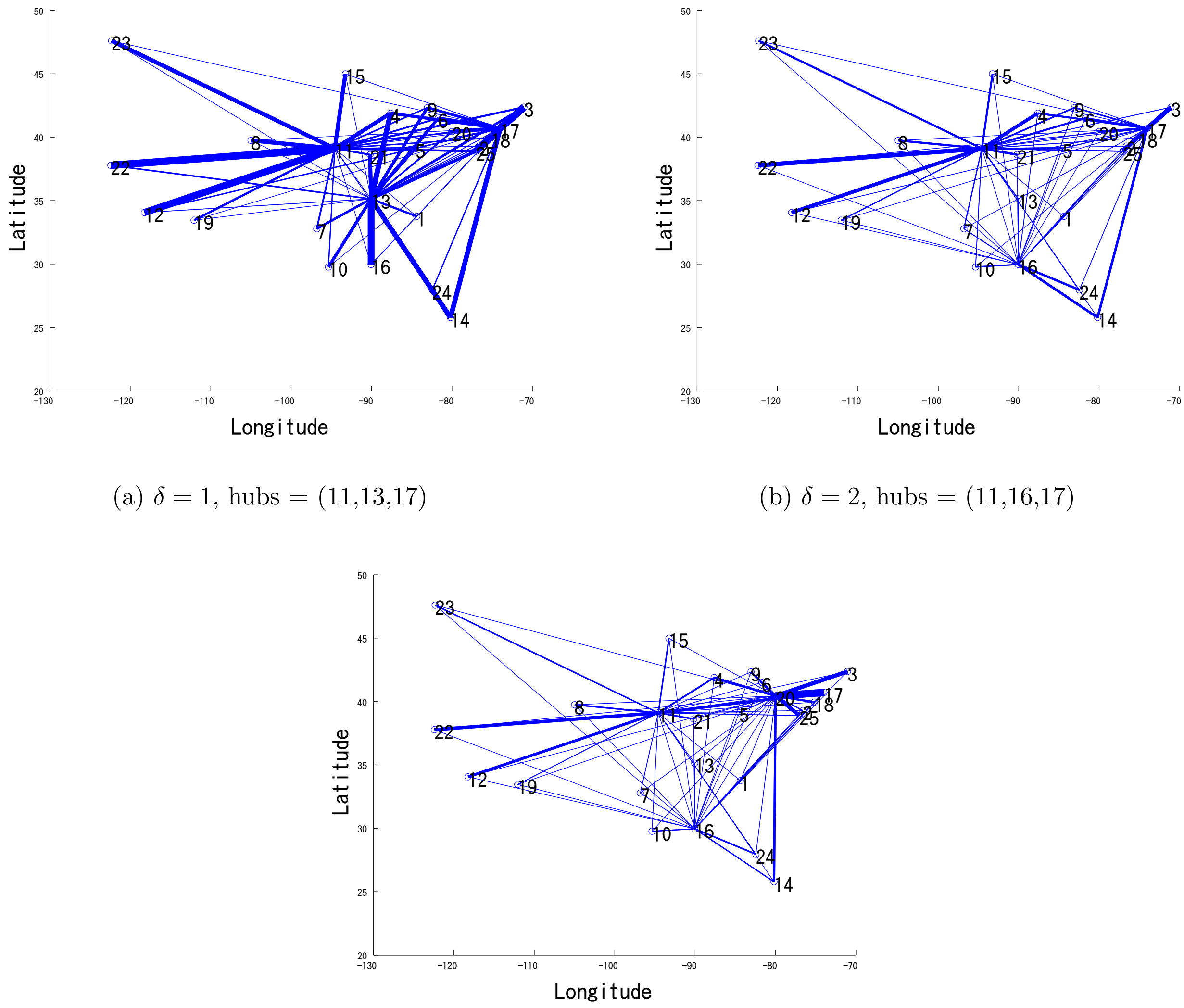
(a) $\delta = 1$, hubs = (11,17)

(b) $\delta = 2$, hubs = (11,20)

Figure 2: Results for $n = 25, p = 2, \rho = 10000$

5. Conclusion

In this paper, we have considered a one-stop capacitated hub location problem in which both hubs and arcs have capacity constraints. We formulated the problem as a mixed 0–1 integer programming problem and solve the problem using branch-and-bound method with Lagrangian relaxation bounding strategy. We also made some computational experiments using actual aviation data based on the well-known CAB data set and the future enplanements forecast reported by FAA. From our computational results, we see that the total cost is affected by arc capacities, especially for problems in which the ratio of the fixed cost to the

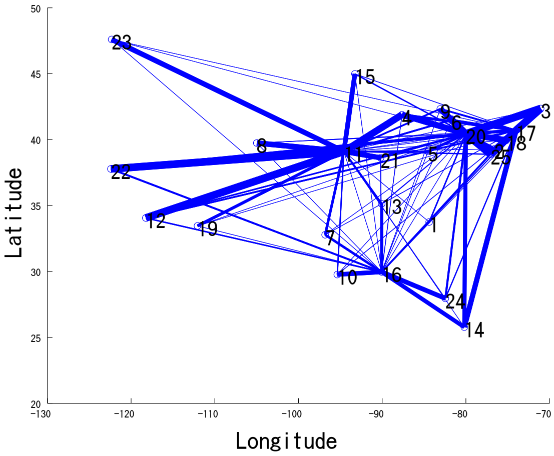


(a) $\delta = 1$, hubs = (11,13,17)

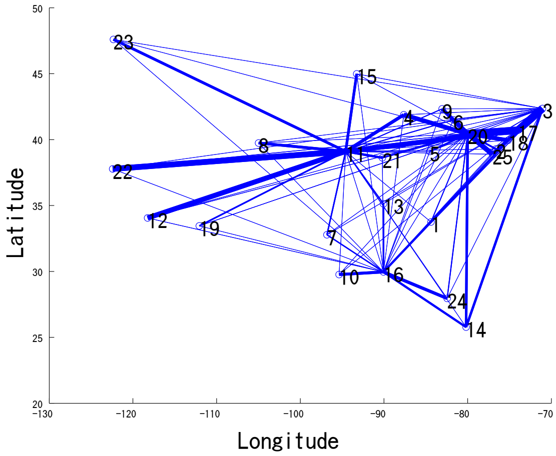
(b) $\delta = 2$, hubs = (11,16,17)

(c) $\delta = 3$, hubs = (11,16,20)

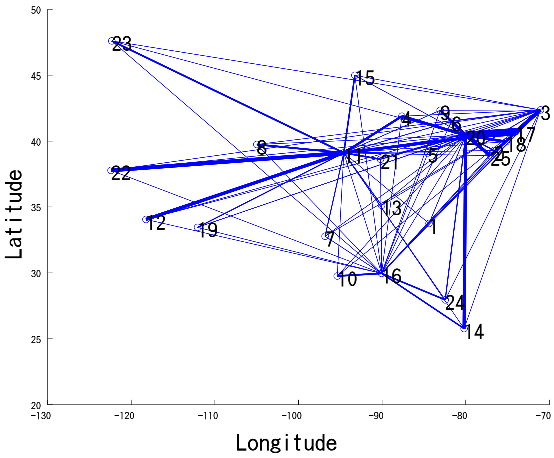
Figure 3: Results for $n = 25, p = 3, \rho = 10000$



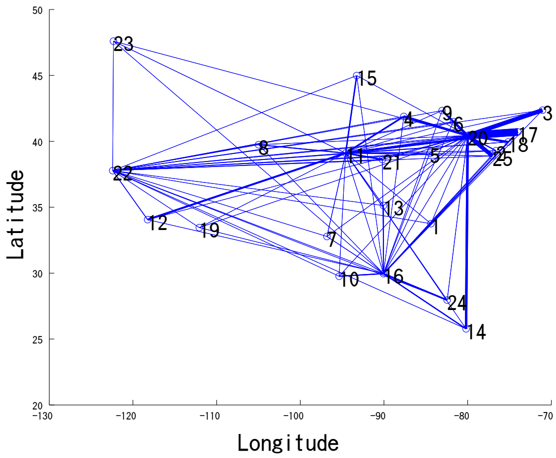
(a) $\delta = 1$, hubs =(11,16,17,20)



(b) $\delta = 2$, hubs=(3,11,16,20)



(c) $\delta = 3$, hubs =(3,11,16,20)



(d) $\delta = 4$, hubs =(11,16,20,22)

Figure 4: Results for $n = 25, p = 4, \rho = 10000$

total cost is large. We also observe that the maximum difference in the optimal cost between the case of loose capacity constraints and that of tight capacity constraints is nearly 5 %. These results indicate that it is important to incorporate arc capacities into the model when we deal with a large network. Taking into account the fact that the arc capacities are likely to be changed frequently by various surroundings, it is also important to select hubs that are insensitive to those changes. In our computational experiments, some hub candidates are always chosen independently of the number of selected hubs and arc capacities. It is an interesting future research to develop an effective method of selecting such a stable hub set by using sensitivity analysis.

Acknowledgments

The authors are grateful to the anonymous referees for their helpful comments. The work was partially supported by Grant for Young Scientists (B), Japan Society for the Promotion of Science, and Nanzan University Pache Subsidy I-A.

References

[1] T. Aykin: Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, **79** (1994) 501–523.

[2] D. Bryan: Extensions to the hub location problem. *Geographical Analysis*, **30** (1998) 315–330.

[3] J. F. Campbell: A survey of network hub location. *Studies in Locational Analysis Issue*, **6** (1994) 31–49.

[4] J. F. Campbell: Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, **72** (1994) 387–405.

[5] J. F. Campbell: Hub location and the p-hub median problem. *Operations Research*, **44** (1996) 923–935.

[6] T. F. Coleman, M. A. Branch and A. Grace: *Optimization Toolbox User's Guide* (The Math Works, Inc., U.S.A., 1999).

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

[8] A. T. Ernst and M. Krishnamoorthy: Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, **104** (1998) 100–112.

[9] A. T. Ernst and M. Krishnamoorthy: An exact solution approach based on shortest-paths for p-hub median problems. *INFORMS Journal on Computing*, **10** (1998) 149–162.

[10] A. T. Ernst and M. Krishnamoorthy: Solution algorithms for the capacitated single allocation hub location problem. *Annals of Operations Research*, **86** (1999) 141–159.

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

[12] FAA: 2001 Aviation Capacity Enhancement Plan (Federal Aviation Administration, Washington DC, 2001).

[13] J. W. Herrmann, G. Ioannou, I. Minis and J. M. Proth: A dual ascent approach to the fixed-charge capacitated network design problem. *European Journal of Operational*

Research, **95** (1996) 476–490.

[14] K. Holmberg and D. Yuan: Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, **48** (2000) 461–481.

[15] P. Jaillet, G. Song and G. Yu: Airline network design and hub location problems. *Location Science*, **4** (1996) 195–212.

[16] O. Kariv and S. L. Hakimi: An algorithmic approach to network location problems. II: The p-medians. *SIAM Journal on Applied Mathematics*, **37** (1979) 539–560.

[17] M. Minoux: *Mathematical Programming: Theory and Algorithms*, (John Wiley & Sons, Chichester, 1986).

[18] M. E. O'Kelly: A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, **32** (1987) 393–404.

[19] M. E. O'Kelly, D. Skorin-Kapov and J. Skorin-Kapov: Lower bounds for the hub location problem. *Management Science*, **41** (1995) 713–721.

[20] M. E. O'Kelly, D. Bryan, D. Skorin-Kapov and J. Skorin-Kapov: Hub network design with single and Multiple Allocation: A computational study. *Location Science*, **4** (1996) 125–138.

[21] M. E. O'Kelly and D. Bryan: Hub location with flow economies of scale. *Transportation Research*, **32B** (1998) 605–616.

[22] B. T. Polyak: Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, **9** (1969) 14–29.

[23] M. Sasaki, A. Suzuki and Z. Drezner: On the selection of relay points in a logistics system. *Asia-Pacific Journal of Operational Research*, **14** (1997) 39–54.

[24] M. Sasaki, A. Suzuki and Z. Drezner: On the selection of hub airports on the airline hub-and-spoke system. *Computers and Operations Research*, **26** (1999) 1411–1422.

[25] D. Skorin-Kapov, J. Skorin-Kapov and M. O'Kelly: Tight linear programming relaxations of uncapacitated P-hub median problems. *European Journal of Operational Research*, **94** (1996) 582–593.

Mihiro Sasaki
Department of Mathematical Sciences
Nanzan University
27 Seirei, Seto, Aichi, 489-0863, Japan
E-mail: `mihiro@ms.nanzan-u.ac.jp`