UNIVERSITY OF CAMBRIDGE
Judge Business School

Working Paper Series

2/2011

Exact Computational Approaches to a Stochastic Uncapacitated Single Allocation p-Hub Center Problem

Edward Hult, Houyuan Jiang and Daniel Ralph

Cambridge Judge Business School Working Papers

These papers are produced by Cambridge Judge Business School, University of Cambridge. They are circulated for discussion purposes only. Their contents should be considered preliminary and are not to be quoted without the authors' permission.

Cambridge Judge Business School author contact details are as follows:

Houyuan Jiang
Cambridge Judge Business School
University of Cambridge
h.jiang@jbs.cam.ac.uk

Danny Ralph
Cambridge Judge Business School
University of Cambridge
d.ralph@jbs.cam.ac.uk

Please address enquiries about the series to:

Research Manager
Cambridge Judge Business School
Trumpington Street
Cambridge CB2 1AG
UK

Tel: 01223 760546
Fax: 01223 339701
Email: research-support@jbs.cam.ac.uk

Exact Computational Approaches to a Stochastic Uncapacitated Single Allocation p-Hub Center Problem

Edward Hult, Houyuan Jiang, Daniel Ralph
Judge Business School, University of Cambridge, Trumpington Street, Cambridge CB2 1AG, United Kingdom,
{e.hult,h.jiang,d.ralph@jbs.cam.ac.uk}

The stochastic uncapacitated single allocation p-hub center problem is an extension of the deterministic version which aims to minimize the longest origin-destination path in a hub and spoke network. Considering the stochastic nature of travel times on links is important when designing a network to guarantee the quality of service measured by a maximum delivery time for a proportion of all deliveries. We propose an efficient reformulation for a stochastic p-hub center problem and develop exact solution approaches based on cutting planes and Benders' decomposition. We report numerical results to show effectiveness of our new reformulations and approaches by finding global solutions of small-medium sized problems.

Key words: hub location; center problem; stochastic programming; Benders' decomposition
History: Accepted by ....

---

1 Introduction

Transportation of goods and people plays a vital role in the economy. One of the key elements affecting the efficiency of shipping services is the network structure. Hub and spoke networks are often employed to balance cost of building and maintaining services between pairs of nodes and having short routes, in terms of distance or time, between pairs of nodes. The hub and spoke structure has found wide application, in air and rail transportation, postal mail systems, and telecommunication networks [1, 6].

Although traffic flows are patently uncertain, with stochastic travel time on links and stochastic volumes between nodes to name just two sources of uncertainty, the main work in the design of hub and spoke networks has focussed on deterministic hub location models. In this literature stochastic data are replaced by averages. This may be sensible in some cases but can suggest network designs that are far from optimal, on average, in a stochastic environment. This phenomenon has been recognized in the facility location literature [2, 26] and is an instance of a more general notion, the so-called flaw of averages [23] which merely says that knowing the average values of stochastic inputs to a process is not enough to determine its average output.

The benefit of deterministic network design is the relative tractability of this problem. The deterministic model underlying our work is the so-called p-center problem which, given a list of N nodes and number p less than N, is to identify p nodes as hubs in order to minimize the maximum travel time across the network. Although this is a combinatorial optimization problem, progress has been impressive. For instance the work of Meyer et al. [20] shows that combining clever integer linear programming reformulations with hybrid algorithms, in a branch and bound framework, can achieve globally optimality in reasonable time for networks of up to 400 nodes.

We consider a stochastic version of the uncapacitated single allocation p-hub center problem equivalent to the formulation proposed by Sim et al. [24]. The challenge is the computational burden of dealing with the stochastic problem since it is not obvious how to take advantage of the formulations and methods that are efficient for deterministic problems. For instance in [24], exact solution seems to be out of the question and even heuristic methods are at their limits when $N = 50$. We approach the goal of finding global solutions in reasonable time in three steps. First we formulate a stochastic model with chance constraints, second we develop a compact integer linear program (ILP) reformulation of the stochastic model under certain conditions, and thirdly we adopt and manipulate various methods and techniques such as cutting planes and Benders' decomposition to find efficient solution procedures.

Although our stochastic model is similar to that in [24], our reformulation is an ILP using $O(N^2)$ binary variables and $O(N^3)$ constraints (see Proposition 1). We therefore call it a (2,3) formulation, referring to the number of indices needed to conveniently represent the $O(N^2)$ variables and $O(N^3)$ constraints. This formulation is compact relative the (4,4) model of [24]. Nevertheless computational efficiency benefits considerably from cutting plane and Benders' decomposition approaches. Indeed the building blocks of our approach — compact ILP reformulations and hybrids of branch and bound with higher level algorithms — are standard; our contribution is in part the development of a new reformulation but mainly to demonstrate the value of exploring generic ILP tools in the stochastic case.

The paper is organised as follows. We review the hub and spoke literature in Section 1.1. In Section 2 we present the stochastic p-hub center problem (SpHCP) and establish our compact integer linear programming reformulation (SpHCP$^{\mathrm{L}}$) . In Section 3, we propose pull (cutting plane) and push (Benders' decomposition) methods for solving the ILP reformulation exactly. In Section 4, we report numerical results in which exact solution of the stochastic p-hub problem for up to $N = 50$ is routine.

1.1 Literature review

Sim et at. [24] are the first to consider the single allocation p-hub center problem with stochastic travel times on each link. They formulate a stochastic mathematical program and then reformulate the problem into an integer linear program by assuming travel time on links have independent normal distributions. Since exact solutions to this problem are hard to get the authors focus their attention on developing several heuristic methods to approximately solve network sizes up to 40 nodes.

In addition to [24] we mention three other hub and spoke papers considering stochastic properties. Marianov and Serra [19] consider stochastic number of customers at hub airports and model this as $M/D/c$ queuing system. They setup a capacity constraint on the number of airplanes needed in the queue to be less than or equal to some probability. This is based on the probability of how many customers are in the system. They develop a heuristic based on tabu search to approximately solve network sizes of 30 nodes. Yang [27] looks at an air freight hub location problem where demand and the associated discount factors on hub-hub links are stochastic. The model is separated into two stages where first stage determines the number and location of hubs taking into account stochasticity that appears in the second stage flight routing problem. [27] assumes a discrete probability distribution involving only three possible scenarios on demand. The paper includes a case study of a 10 node air freight network in Taiwan and China. Contreras et al. [7] study the uncapacitated multiple allocation hub location problem and include stochastic properties of both demand and transportation cost. For the rather special case in which a single random parameter affects the transportation cost on all links equally, they show that their stochastic model is equivalent to the deterministic problem in which all random variables are replaced by their average values, i.e., there is no flaw of averages. For independent transportation costs, however, where the uncertainty of a link cost is independent of all other links in the network, the same cannot be done. They therefore move on to approximately solve the later problem for network sizes up to 50 nodes using Monte-Carlo simulation coupled with Benders' decomposition.

Although the literature on stochastic hub and spoke networks is small, there is a relatively larger pool of research done on deterministic problems. Hub location became a recognised field of study in the late 1980s. The state-of-the-art for the research in hub location can be found in two review papers [1, 6]. Attention has focussed on two themes: proposing mathematical models such as median, center and covering hub problems to approximate the real-world business challenges, and developing efficient numerical approaches for solving these proposed mathematical models.

Computational methods dominate the hub location literature. Unsurprisingly, as the number of nodes in the network grows, exact methods are more prone to computational difficulties — such as excessive time or memory requirements — than are heuristics.

Campbell [5] is the first to introduce and study the p-hub center problem in the hub literature. This includes the deterministic version of the problem we study (Section 2) which is motivated by a hub system involving perishable or time sensitive items in which cost refers to time.

Campbell [5] defines the uncapacitated single allocation p-hub center problem (USApHCP) as a quadratic programming problem and proposes a linear programming reformulation using $O(N^4)$ variables where, as above, N is the number of nodes in the network. The same problem has been studied by several different authors aiming to speed up the computational efficiency. Kara and Tansel [17] present an equivalent reformulation using $O(N^2)$ variables and $O(N^3)$ constraints. An additional improvement on the computational performance of the USApHCP was found by Ernst et al. [10] by reformulating it using what they call the radius approach, which requires only $O(N^2)$ variables and $O(N^2)$ constraints. The radius approach can solve test problems that have up to 200 nodes in the network. Recently a new 2-phase method combines a branch and bound algorithm and the radius approach is introduced by Meyer et al. [20] and can solve test examples that have up to 400 nodes. Jutte et al. [16] give a polyhedral analysis on the hub center problem based on the radius approach.

The other two standard problems in the hub literature are the p-hub median problem and the hub covering problem which both follow the same trend with different researchers trying to either improve on the original formulation computationally or to study extensions and/or variations to the problem. The p-hub median problem, which is to minimize the total cost of all flows between all OD pairs over the network, has received more attention in the literature. Transportation costs between hub nodes are usually discounted by a factor to reflect economies of scale. O'Kelly [21] is the first to formulate a hub location problem mathematically. Campbell [4, 5] later produces the first integer linear programming formulations for single and multiple allocation p-hub median problems. Skorin-Kapov et al. [25] reformulate the linear formulation to produce exact solutions in quicker times. Ernst and Krishnamoorthy [11] present a very efficient (3,3) reformulation. Recent advances for the hub median problems can be found in [8, 22].

The third standard problem is the hub covering problem, first mathematically formulated by Campbell [5]. The motivation is to find a set of hubs so that the cost of opening hubs or the number of hubs to use is minimized subject to delivering a certain service. We refer the interested reader to Alumur and Kara [1] for additional details on hub covering.

2 The Stochastic Hub Center Problem and a Reformulation

Here we present the stochastic p-hub center problem (SpHCP for short) after Sim et al. [24] and, in the deterministic case, Campbell [5]. This is a quadratic program which has $O(N^2)$ binary variables and $O(N^4)$ constraints. In Proposition 1 we show it has a (2,3) representation as an integer linear program.

Consider a network consisting of N nodes, denoted by $\mathcal{N}$ and options to link any pair of nodes. The designer wants to select p hubs in $\mathcal{N}$ and to assign each node in $\mathcal{N}$ to exactly one hub. If k and m are hub nodes and nodes i and j are allocated to k and m, respectively, then the path for delivering the goods from node i to node j is $i \to k \to m \to j$. Assume that D_{ij} represents random travel time between nodes i and j with an average travel time of d_{ij} and a standard deviation of σ_{ij}. Assume $\alpha \in (0,1)$ is the discount factor for travel times over hub arcs (links between hub nodes). The total travel time along the path $i \to k \to m \to j$ is $T_{ijkm} = D_{ik} + \alpha D_{km} + D_{mj}$.

Let β be the nominal maximum travel times between all OD pairs in the network. Assume $\gamma \in [0,1]$ is the service level. Then the service level guarantee means that for any nodes i and j, with probability of γ, the travel time for the goods delivered from i to j will not exceed β. For instance if $\beta = 24$ hours and $\gamma = 0.95$, then the service level guarantee requires that the travel time from i to j will not exceed 24 hours for 95% of journeys. This type of service level guarantee is often employed in call centers and other telecommunication services [14, 15]. Clearly, the designer prefers a smaller value for β and a larger value for γ. However, these two parameters usually act against each other. For example, a large value of γ often results in a relatively large value of β.

Define X_{ik} to be a binary variable such that $X_{ik} = 1$ if and only if k is a hub node and node i is assigned to k. SpHCP is defined by the following nonlinear stochastic program:

$$\text{(SpHCP)} \quad \min \quad \beta \tag{1}$$

$$\text{s.t.} \quad \gamma \le Pr(\beta \ge (D_{ik} + \alpha D_{km} + D_{mj}) X_{ik} X_{jm}), \quad \forall i,j,k,m \in \mathcal{N} \tag{2}$$

$$\sum_{k \in \mathcal{N}} X_{ik} = 1, \quad \forall i \in \mathcal{N} \tag{3}$$

$$X_{ik} \le X_{kk}, \quad \forall i,k \in \mathcal{N} \tag{4}$$

$$\sum_{k \in \mathcal{N}} X_{kk} = p, \tag{5}$$

$$X_{ik} \in \{0,1\}, \quad \forall i,k \in \mathcal{N}. \tag{6}$$

In the above formulation, the objective is to minimize the nominal maximum travel time between all OD pairs in the network. Constraint (3) indicates that each node is allocated to exactly one hub.

Constraint (4) shows that if node i is allocated to hub k, then node k must be a hub. Constraint (5) specifies that exactly p hubs are established. Constraint (6) states that X_{ik} is a binary variable. The most complicated is constraint (2), which defines the service level: The probability such that the travel time on each OD path is less than or equal to β is greater than or equal to γ. Note that when either $X_{ik} = 0$ or $X_{jm} = 0$, $i \to k \to m \to j$ is not a valid travel path between nodes i and j. In this case, constraint (2) still holds because $Pr(\beta \geq 0) = 1$ and this constraint becomes redundant.

A few remarks are made in order. Firstly, if the travel times on all links are deterministic, then constraint (2) is equivalent to the following constraint:

$$\beta \geq (d_{ik} + \alpha d_{km} + d_{mj}) X_{ik} X_{jm},$$

and SpHCP is equivalent to the p-hub center problem studied by Campbell [5]. This shows that SpHCP is an extension of the traditional deterministic p-hub center problem. Secondly, SpHCP is equivalent to the stochastic p-hub center problem investigated by Sim et al. [24], where four-index binary variables y_{ijkm} is used and $y_{ijkm} = 1$ is implied by $X_{ik} = X_{jm} = 1$.

It is easy to define and understand SpHCP. Also, up to this this point, we have not needed specific assumptions on either the probability distribution of link travel times D_{ik}, or on the relationship between these random travel times for different links. However, this formulation involves both integer variables and the stochastic constraint (2). These features make the above formulation computationally intractable. In the sequel, we aim to convert the above formulation into a computationally tractable deterministic integer linear program based on two techniques: linearizing quadratic terms, and replacing probabilistic expressions by equivalent deterministic counterparts. To this end, we make one more assumption which states that, first, D_{ij} is normally distributed and, second, is mutually and stochastically independent of $D_{k\ell}$ for any nodes i, j, k, ℓ with $(i, j) \neq (k, \ell)$. Based on this assumption, the total travel time $T_{ijkm} = D_{ik} + \alpha D_{km} + D_{mj}$ along the path $i \to k \to m \to j$ is also normally distributed with a mean of $t_{ijkm} = d_{ik} + \alpha d_{km} + d_{mj}$ and a standard deviation of $\delta_{ijkm} = \sqrt{(\sigma_{ik})^2 + \alpha^2(\sigma_{km})^2 + (\sigma_{mj})^2}$.

The fact that T_{ijkm} has a normal distribution shows that the stochastic constraint (2) is equivalent to a deterministic and quadratic constraint:

$$\beta \geq (t_{ijkm} + z_\gamma \delta_{ijkm}) X_{ik} X_{jm}, \quad \forall i, j, k, m \in \mathcal{N},$$

where z_γ is the z-value for a standard normal distribution and satisfies equation $Pr(Z \leq z_\gamma) = \gamma$ and Z is a standard normally distributed random variable. Now we can apply some standard tricks for reformulating a quadratic inequality as a linear inequality.

Evidently, the optimal value for β is nonnegative. Because X_{ik} is a binary variable, it is easy to verify that constraints $\beta \geq (t_{ijkm} + z_\gamma \delta_{ijkm}) X_{ik} X_{jm}$ for all $i, j, k, m \in \mathcal{N}$ can be collectively replaced by constraints:

$$\beta \geq (t_{ijkm} + z_\gamma \delta_{ijkm})(X_{ik} + X_{jm} - 1), \quad \forall i, j, k, m \in \mathcal{N}.$$

Furthermore, constraint $\beta \geq (t_{ijkm} + z_\gamma \delta_{ijkm}) X_{ik} X_{jm}$ or constraint $\beta \geq (t_{ijkm} + z_\gamma \delta_{ijkm})(X_{ik} + X_{jm} - 1)$ is redundant if either $X_{ik} = 0$ or $X_{jm} = 0$. In view of constraint (3), for any fixed i, j, m, constraints $\beta \geq (t_{ijkm} + z_\gamma \delta_{ijkm})(X_{ik} + X_{jm} - 1)$ for all k are equivalent to a single constraint

$$\beta \geq \sum_{k \in \mathcal{N}} (t_{ijkm} + z_\gamma \delta_{ijkm})(X_{ik} + X_{jm} - 1), \quad \forall i, j, m \in \mathcal{N}. \tag{7}$$

By the above two reformulation techniques, we have obtained an equivalent deterministic integer linear program for the S*p*HCP.

Proposition 1 *Assume that for all $i, j \in \mathcal{N}$, D_{ij} are mutually and stochastically independent and normally distributed. Then the stochastic optimization problem SpHCP defined by (1)-(6) is equivalent to the following integer linear program:*

$$\begin{array}{lll} (\mathrm{SpHCP^L}) & \min & \beta \\ & \text{s.t.} & (3), (4), (5), (6), (7). \end{array}$$

Having introducing S*p*HCP and its deterministic counterpart S*p*HCP^L, we make three remarks in addition to the two made earlier in this section. First, Sim et al. [24] derive a deterministic counterpart from their stochastic optimization problem and the deterministic counterpart is also an integer linear program. However, the deterministic formulation in Sim et al. [24] has $N^4 + N^2 + N + 1$ constraints and $N^4 + N^2$ binary variables, whereas our S*p*HCP^L has only $N^3 + N^2 + N + 1$ constraints and N^2 binary variables. The (2,3) representation S*p*HCP^L seems computationally more promising in terms of both computer memory and CPU times. Second, the appendix of [24] gives an alternative (2,4) ILP representation that requires under an extra assumption on the stochastic variables, a kind of stochastic triangle inequality. Third, when randomness of travel times is removed, S*p*HCP^L is similar to the integer linear programming formulation studied in Kara and Tansel [17]. The only difference is in the constraint (7) for defining the lower bound β on the objective function, which in [17] is given by $\beta \geq \sum_{k \in \mathcal{N}} (d_{ik} + \alpha d_{km}) X_{ik} + d_{mj} X_{jm}$, for each i, j and m in $\mathcal{N}$.

It is important to note that the same approach can be applied for any distribution of the travel time between nodes provided that the percentiles T^D_{ijkm} are known in advance, where T^D_{ijkm} satisfies the equation $Pr(D_{ij} + \alpha D_{km} + D_{mj} \leq T^D_{ijkm}) = \gamma$. If, unlike the normal distribution, these percentiles cannot be computed directly, then they can be found by sampling or simulation techniques which can be carried out independently from and prior to solving S*p*HCP^L.

3 Solution Methods

Looking ahead to Section 4, our preliminary computational experience in applying CPLEX to SpHCP^L indicates that this combination of model and software is adequate for problems up of to 25 nodes (Table 1). However $N = 25$ is small for the standard dataset of test problems, the Australian Post (AP) [12], where deterministic problems of hundreds of nodes can be solved exactly [20], reasonably quickly. Therefore, in the following two subsections, we investigate two computational approaches for solving SpHCP^L more efficiently. These are called pull and push approaches, respectively.

In the pull approach, we aim to remove redundant constraints related to (7) and to add some valid cuts. The push approach is a particular application of the well-known Benders' decomposition [3] in which we start with no or a few constraints of type (7) and we gradually add more such constraints if required. It is worthwhile to mention that Camargo et al. [9] solve the uncapacitated multiple allocation problem using the Benders' decomposition approach, which allows them to solve test examples with a network size of up to 200 nodes.

3.1 SpHCP-Pull, cutting planes

Suppose that β^U and β^L are upper and lower bounds for the optimal objective function value for SpHCP^L or equivalently SpHCP. We give a lemma that uses these bounds to identify valid cuts and redundant constraints for SpHCP^L; its proof is elementary and appears in the Appendix. Then we suggest a way to generate valid upper bounds.

Lemma 1

(a) If for given j, m, $2d_{jm} + z_\gamma \sqrt{2}\sigma_{jm} > \beta^U$, then $X_{jm} = 0$ is a valid cut for SpHCP^L.

(b) If for given i, j, m, $\min_{k=1}^N (t_{ijkm} + z_\gamma \delta_{ijkm}) > \beta^U$, then $X_{jm} = 0$ is a valid cut for SpHCP^L.

(c) If for given i, j, m, $\max_{k=1}^N (t_{ijkm} + z_\gamma \delta_{ijkm}) < \beta^L$, then constraint (7) is redundant for the given i, j, m and can be removed.

(d) Assume the triangular inequality property holds over d_{ij}, i.e. $d_{ij} + d_{im} \geq d_{im}$ and the travel time is symmetric: i.e., D_{ij} and D_{ji} have the same probability distribution. If for given j, m, $d_{jm} + z_\gamma \sigma_{jm} + \alpha \max_{\ell=1}^N d_{m\ell} > \beta^U$, then $X_{jm} = 0$ is a valid cut.

(e) If $X_{jm} = 0$ is a valid cut, then for any i and the corresponding j, m, constraint (7) is redundant for SpHCP^L.

Based on Lemma 1, if an upper bound and/or a lower bound for SpHCP^L are available, then we can obtain a better integer linear program than SpHCP^L by adding some additional cuts and by removing some redundant constraint. We call this modified integer linear programming formulation SpHCP^L-Pull. The exact format of SpHCP^L-Pull depends on the test example and available upper and lower bounds.

Next we design a heuristic method for deriving an upper bound.

Define $\tilde{d}_{ik} = d_{ik} + z_\gamma \sigma_{ik}$. It is easy to verify that $Pr(D_{ik} \leq \tilde{d}_{ik}) = \gamma$, which represents the service level on the link between i and k. Based on $\tilde{d}_{ik}$, we construct a new network whose network structure is identical to the existing network, but the direct stochastic travel time D_{ik} between i and k is replaced by the deterministic travel time $\tilde{d}_{ik}$. For this new network, we follow Ernst et al. [10] to propose the radius-based formulation:

$$\begin{array}{rrcll} \text{(Heuristic)} & \min & & \beta & \\ \text{s.t.} & r_k & \geq & \tilde{d}_{ik} X_{ik}, \forall i, k \in \mathcal{N} & \\ & \beta & \geq & r_k + r_m + \alpha \tilde{d}_{km}, & \forall k, m \in \mathcal{N} \\ & & & (3), (4), (5), (6). & \end{array}$$

Here r_k represents the radius of hub node k. When k is not a hub node, $r_k = 0$ holds automatically. If for all i, k, D_{ik} is a random variable with a single pulse, then $D_{ik} = \tilde{d}_{ik} = d_{ik}$ and the above formulation reduces to the radius-based formulation proposed in Ernst et al. [10]. In terms of computational times, the radius-based formulation is the state-of-the-art integer linear programming formulation for the single allocation p-hub center problem. The optimal solution for (Heuristic) gives a feasible solution for SpHCP, which in turn generates an upper bound for SpHCP.

3.2 SpHCP-Push, Benders' decomposition

In this subsection we focus on the so-called push approach SpHCP^L-Push, which is an application of Benders' decomposition [3]. SpHCP^L-Push consists of two separate components: so-called restrictive master problems and, to check their optimality, subproblems. A restrictive master problem is a modification of SpHCP^L by ignoring some constraints of type (7). A subproblem is to check whether or not the most recent restrictive master problem indeed gives an optimal solution to SpHCP^L or SpHCP. This can be done by checking whether or not any constraint of type (7) violates if β in (7) is replaced by the optimal objective function value of the most recent restrictive master problem. If not, an optimal solution for SpHCP^L is obtained. Otherwise, such violating constraints are added to the most recent master problem to form an updated restrictive master problem.

Due to the special structure of SpHCP^L, we only need to check constraint violations for some selected ones of type (7). Suppose X^*_{ik} and β^* are the optimal solution and the optimal objective function value for the most recent restrictive master problem, respectively. If $X^*_{jm} = 0$, then for any i and the corresponding j, m, $\{X^*_{ik}\}$ satisfies constraint $\beta^* \geq \sum_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm})(X^*_{ik} + X^*_{jm} - 1)$ automatically, which implies that there is no need to check constraint violations for any i and the corresponding j, m. If $X^*_{jm} = 1$ and constraint $\beta^* \geq \sum_{k \in \mathcal{N}} T^D_{ijkm} X^*_{ik}$ is violated, then for any i and the corresponding j, m, constraint (7) is added to the most recent restrictive master problem. Note that for any j, there is exactly one m such that $X^*_{jm} = 1$. Therefore, we add at most N^2 constraints of type (7) to the new restrictive master problem. The above argument also shows that the subproblem can be solved in polynomial time: checking if $X_{jm} = 0$, and checking if $\beta^* \geq \sum_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm}) X^*_{ik}$ is violated when $X_{jm} = 1$.

The SpHCP^L-Push approach cycles iteratively between restrictive master problems and subproblems and terminates when an optimal solution for SpHCP^L is found. Because there are exactly N^3 constraints of type (7) and at least one new and different constraint of type (7) is added to the restrictive master problem, an optimal solution for SpHCP^L can be found in at most N^3 iterations between the restrictive master problem and the subproblem. The SpHCP^L-Push approach as well as some related results is outlined below.

SpHCP^L-Push:

Step 1 Generate an initial restrictive master problem: Modifying SpHCP^L by removing all the constraints of type (7).

Step 2 Solve the most recent restrictive master problem. Let X^*_{ik} and β^* be its optimal solution and optimal objective function value, respectively.

Step 3 Solve the subproblem by checking constraint violations for (7) based on the procedure described above. If there is no constraint violation, terminate and an optimal solution for SpHCP^L is obtained. Otherwise, go to Step 4.

Step 4 Add newly violating constraints to the most recent restrictive master problem to form a new restrictive master problem. Go to Step 2.

Proposition 2

(a) *The subproblem in SpHCP^L-Push is polynomially solvable.*

(b) An optimal solution for SpHCP^L can be obtained after at most N^3 iterations between the restrictive master problem and the subproblem.

4 Numerical Results

In this section, we test the computational performance of the approaches proposed in the previous sections using well-known test examples from two libraries of test problems, (CAB) [13, 21] and (AP) [12]. Our code is written in C++, integer linear programs are solved using ILOG CPLEX Version 12.1, and all of the numerical experiments are carried out on a HP Pavilion laptop with an Intel(R)Core(TM)2 CPU processor with 3.00 GB RAM.

The standard CAB and AP test examples are designed for the deterministic p-median hub location problem. Recall that D_{ik} is assumed to be normally distributed with a mean of d_{ik} and a standard deviation of σ_{ik}. In our test examples, we assume that d_{ik} takes the value given in the (deterministic) test libraries and $\sigma_{ik} = \nu d_{ik}$, for a constant ν called the coefficient of variation. In our runs we set $\nu = 1$ and $\gamma = 0.95$ for all test examples and later check the sensitivity of the CPU times with respect to both γ and ν.

We test three approaches: SpHCP^L, SpHCP^L-Pull, and SpHCP^L-Push and let each run for a maximum of 1 CPU hour (3600 CPU seconds). Note that the problem (Heuristic) introduced in section 3.1 is used for finding an upper bound for SpHCP^L-Pull. Numerical results are shown in the tables below. For each test problem, we report:

`Prob` problem name (for CAB examples, a name has a format of $N.p.q$ such that $q = 10 * \alpha$, with the exception of $q = 1$ than $q = \alpha$, and for AP examples, a name has a format of $N.p$ and $\alpha = 0.75$),

`Obj` optimal objective function value,

`Upper` objective function value of the feasible solution generated from (Heuristic) which provides an upper bound to `Obj`,

SpHCP^L CPU time of running SpHCP^L,

SpHCP^L-Pull CPU time of running SpHCP^L-Pull,

SpHCP^L-Push CPU time of running SpHCP^L-Push, and

`Iter` number of iterations for SpHCP^L-Push.

For network sizes of 25 nodes or fewer (Table 1), all methods perform relatively well. In comparison to the test runs in Sim et al. [24] — where the computational demands of the (4,4) formulation of [24] meant that only heuristic approaches could be usefully employed — our standard SpHCP^L formulation appears to be very promising because CPLEX produces globally optimal solution for all problems within a few minutes. Moreover, the difference between SpHCP^L-Pull and SpHCP^L-Push is very small for network sizes of 25 nodes or fewer as can be seen in Tables 1 and 2.

For network sizes of 40 nodes or more (Table 2), however, we can no longer solve the test problems using SpHCP^L with one exception, 40.2, or solve networks of sizes 50 nodes using SpHCP-Pull within 1 CPU hour. What is interesting is that SpHCP^L-Push performs very well for the test examples with 40 or 50 nodes.

Prob	γ	ν	Obj	SpHCP^L	SpHCP^L-Pull	SpHCP^L-Push	Upper	Iter
5.3.1	0.95	1	2078.44	0.08	0.02	0.02	2276.84	2
5.3.6	0.95	1	1736.03	0.08	0.02	0.04	1788.54	2
10.2.1	0.95	1	3937.62	0.61	0.13	0.04	3986.21	2
10.2.2	0.95	1	2896.63	0.59	0.08	0.05	2896.63	2
10.2.4	0.95	1	3207.72	0.6	0.17	0.21	3207.72	3
10.2.6	0.95	1	3576.5	0.76	0.16	0.14	3811.55	3
10.2.8	0.95	1	3811.55	0.92	0.07	0.06	3811.55	2
10.3.1	0.95	1	3829.18	0.66	0.06	0.05	3881.41	2
10.3.2	0.95	1	2425.73	0.44	0.05	0.04	2425.73	2
10.3.4	0.95	1	2425.73	0.47	0.05	0.04	2425.73	2
10.3.6	0.95	1	2807.9	0.37	0.05	0.04	2807.9	2
10.3.8	0.95	1	3361.34	0.49	0.05	0.04	3386.14	2
10.4.1	0.95	1	3829.18	0.4	0.21	0.25	4676.7	4
10.4.2	0.95	1	1707.95	0.45	0.06	0.04	1707.95	2
10.4.4	0.95	1	1921.89	0.41	0.03	0.03	1921.89	2
10.4.6	0.95	1	2800.5	0.4	0.04	0.06	2806.02	3
10.4.8	0.95	1	3361.34	0.38	0.06	0.09	3741.36	3
15.2.1	0.95	1	5420.99	3.31	0.69	0.35	5655.94	4
15.2.2	0.95	1	4126.04	2.07	0.54	0.57	4126.04	5
15.2.4	0.95	1	4430.18	2.45	0.66	0.53	4681.75	5
15.2.6	0.95	1	4785.7	2.11	0.45	0.24	4798.08	3
15.2.8	0.95	1	5184.13	2.3	0.45	0.16	5257.06	3
15.3.1	0.95	1	5092.54	2.49	1.44	1.58	6890.21	7
15.3.2	0.95	1	3647.14	1.98	0.26	0.18	3789.68	3
15.3.4	0.95	1	3813.75	2.54	0.53	0.36	3813.75	4
15.3.6	0.95	1	4009.01	2.35	0.14	0.08	4009.01	2
15.3.8	0.95	1	4702.49	2.3	1.1	0.28	5512.17	4
15.4.1	0.95	1	5092.54	1.72	1.39	1.6	6890.21	7
15.4.2	0.95	1	2782.98	1.72	0.25	0.08	2782.98	2
15.4.4	0.95	1	2949.82	2.71	0.13	0.06	2949.82	2
15.4.6	0.95	1	4009.01	3.28	0.41	0.2	4134.12	3
15.4.8	0.95	1	4702.49	2.72	0.69	0.31	5512.17	4
20.2.1	0.95	1	5420.99	13.92	2.55	0.84	5655.94	5
20.2.2	0.95	1	4101.59	9.57	1.14	0.14	4101.59	2
20.2.4	0.95	1	4284.84	9.86	2.25	1.29	4681.75	5
20.2.6	0.95	1	4785.7	9.62	2.07	0.74	4928.59	4

20.2.8	0.95	1	5089.27	13.36	2.6	0.81	5420.99	5
20.3.1	0.95	1	5092.54	14.74	9.07	4.04	6890.21	8
20.3.2	0.95	1	3075.01	6.11	0.85	0.14	3075.01	2
20.3.4	0.95	1	3688.54	8.04	2.32	0.8	3813.75	4
20.3.6	0.95	1	4004.81	21.09	0.49	0.11	4032.51	2
20.3.8	0.95	1	4671.23	28.1	6.46	0.69	5171.19	4
20.4.1	0.95	1	5092.54	14.65	6.46	6.77	6890.21	8
20.4.2	0.95	1	2830.12	16.17	1.3	0.44	2922.82	3
20.4.4	0.95	1	3028.72	12.47	0.77	0.2	3028.72	2
20.4.6	0.95	1	4001.28	19.58	0.77	0.33	4060.07	3
20.4.8	0.95	1	4671.23	16.15	5.06	1.26	5406.61	5
25.2.1	0.95	1	5629.69	100.35	1.52	0.19	5646.17	2
25.2.2	0.95	1	4517.81	33.71	6.95	1.61	4617.72	5
25.2.4	0.95	1	4814.33	26.98	10.32	6.18	4895.53	7
25.2.6	0.95	1	5117.46	32.42	2.43	0.63	5117.46	4
25.2.8	0.95	1	5362.94	110.05	1.98	0.21	5362.94	2
25.3.1	0.95	1	5455.94	148.55	3.21	1.22	5982.31	5
25.3.2	0.95	1	3880.84	34.35	5.67	0.89	4105.16	3
25.3.4	0.95	1	4407.47	60.15	15.43	5.66	4439.91	6
25.3.6	0.95	1	4624.12	101.85	4.57	0.36	4624.12	2
25.3.8	0.95	1	5048.59	92.58	0.84	0.14	5060.09	2
25.4.1	0.95	1	5455.94	119.88	18.55	5.96	6911.02	7
25.4.2	0.95	1	3216.94	56.39	2.46	0.71	3216.94	3
25.4.4	0.95	1	3813.75	96.97	2.34	0.5	3813.75	3
25.4.6	0.95	1	4449.14	192.99	14.59	3.3	4617.72	6
25.4.8	0.95	1	5048.59	109.1	5.16	0.87	5342.62	4

Table 1: Numerical results for CAB examples.

To visualize the relative performance among all three numerical methods, we plot the ratio of the CPU time for each method against the best CPU time among the three methods. This relative performance of CPU times is given in Figure 1, which shows that SpHCP^L-Push is a clear winner. We remark that the graphs are truncated at the performance ratios of 20 and 80 for the CAB and AP datasets, respectively.

Some additional observations can be made. First, additional tests in which we applied our proposed methods to the deterministic single-allocation p-hub center problem showed that they are more efficient than the methods of [18] but less efficient than the radius method of [10] that itself is dominated by the more recent advance of [20] in which problems of hundreds of nodes can be routinely solved. This gives an idea of the gap between tractability of stochastic and deterministic hub center problems. Second, it is clear from constraint (7) in SpHCP^L that the objective value is increasing in the discount factor α, the coefficient of variation ν, and the service level parameter γ since increasing any of these will most often increase the right hand side of (7) for all $i, j, m \in \mathcal{N}$. The exceptions being if β is bound by a path where $k = m$ or by a path with a single pulse, than respectively increasing α or ν and/or γ no more so as to keep β bound by the same path,

Table 2: Numerical results for AP examples.

Prob	γ	ν	Obj	SpHCP^L	SpHCP^L-Pull	SpHCP^L-Push	Upper	Iter
5.2	0.95	1	60402	0.08	0.02	0.05	61525.2	3
5.3	0.95	1	60402	0.07	0.03	0.06	65123.5	3
10.2	0.95	1	87498.2	0.69	0.11	0.04	87498.2	2
10.3	0.95	1	73634.5	0.69	0.1	0.09	75413.2	3
10.4	0.95	1	70902.5	0.53	0.04	0.04	70902.5	2
10.5	0.95	1	70902.5	0.33	0.09	0.15	75086.6	3
20.2	0.95	1	99570.1	8.3	3.13	0.24	99570.1	3
20.3	0.95	1	93111.2	17.91	0.97	0.25	94036.9	3
20.4	0.95	1	90871.9	13.23	2.65	1.75	100351	5
20.5	0.95	1	90871.9	16.5	2.54	1.23	100351	5
20.10	0.95	1	90871.9	4.74	2.47	1.47	100351	5
25.2	0.95	1	114205	51.99	6.2	0.63	115289	3
25.3	0.95	1	109781	53.8	30.53	7.44	120714	6
25.4	0.95	1	109781	30.97	9.17	6.08	120714	7
25.5	0.95	1	109781	37.74	12.36	6.75	120714	7
25.10	0.95	1	109781	16.27	8.86	7.42	120714	14
40.2	0.95	1	131626	1904.91	58.54	2.41	133652	3
40.3	0.95	1	121203	*	278.7	19.19	126088	6
40.4	0.95	1	114571	*	901.98	107.36	131814	7
40.5	0.95	1	114571	*	85.27	125.44	131814	9
40.10	0.95	1	114571	*	150.62	253.41	131814	9
50.2	0.95	1	133722	*	*	196.88	141971	6
50.3	0.95	1	120783	*	*	70.16	124513	5
50.4	0.95	1	117921	*	*	2651.61	131557	10
50.5	0.95	1	117921	*	*	1192.33	134376	11
50.10	0.95	1	117921	*	*	739.38	134376	19

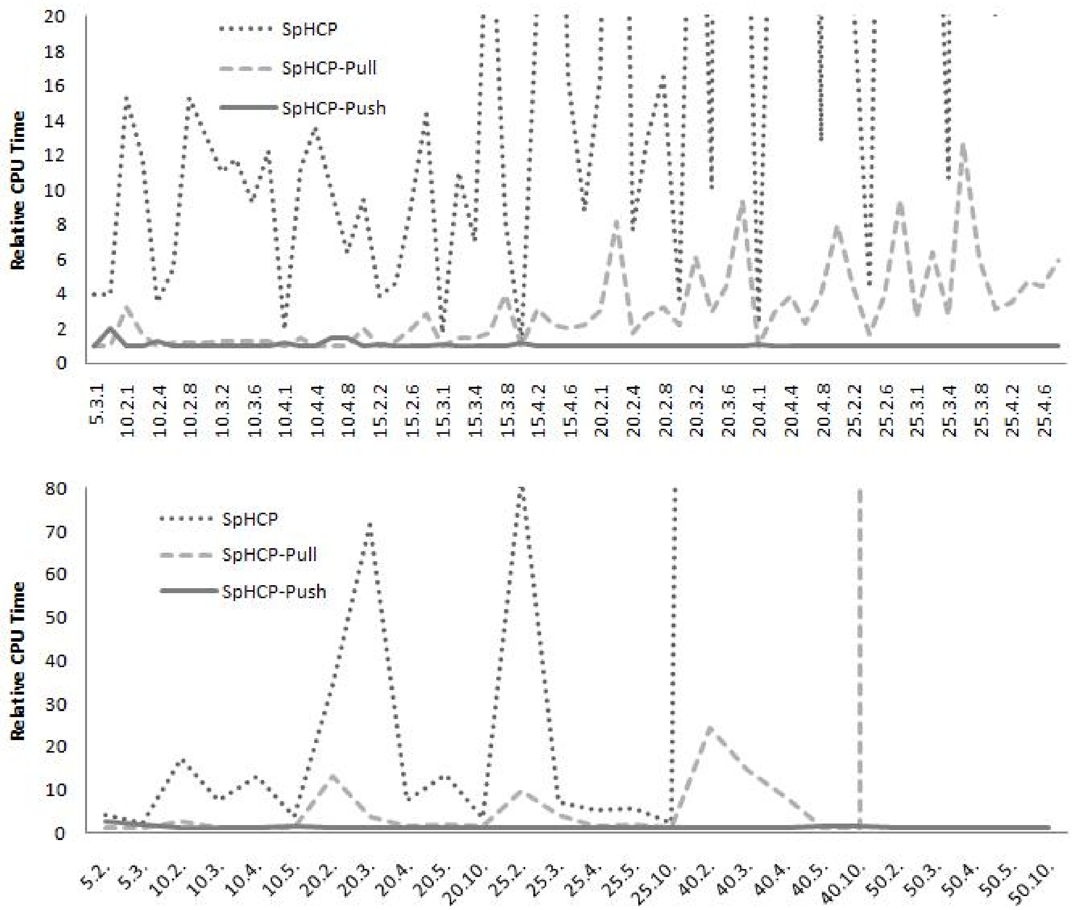

Figure 1: Performance of CPU times for SpHCP^L, SpHCP^L-Pull, and SpHCP^L-Push: The top panel for the CAB dataset and the bottom panel for the AP dataset.

will not affect the objective value. However it is not as clear how this may affect the CPU times. We therefore conduct a sensitivity analysis on a single test problem, AP 25.2, for a fixed value of $\gamma = 0.95$ and $\nu = 1$ respectively. We check sensitivity of the CPU times of the three methods as functions of ν (γ, resp.) and observe that the CPU times are not very sensitive to either service level γ or coefficient of variation ν as can be seen in figure 2.

5 Conclusion

In this paper we introduce a new formulation (Proposition 1) for the stochastic uncapacitated single allocation p-hub center problem together with two solution procedures for finding globally optimal solutions based on a cutting plane approach and Benders' decomposition.

The combination of modelling and optimization techniques allows us to solve small to medium-sized problems in reasonable time. This is in contrast to the original formulation of the problem in

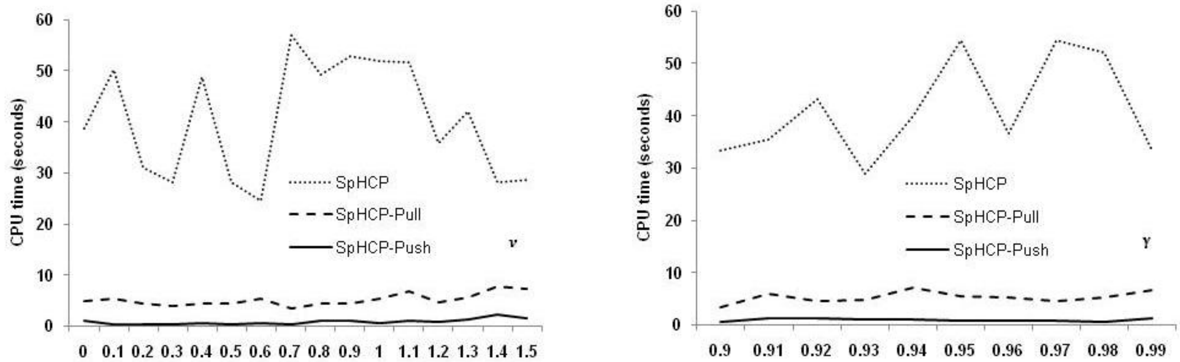

Figure 2: Sensitivity results of the CPU times in coefficient of variation ν and service level parameter γ for test example AP 25.2.

Sim et al. [24] where computational difficulties led to the use of heuristics for even small problems. Of note is the Benders' decomposition approach where the subproblem is polynomially solvable, requiring at most N^3 iterations. The combination of the new model formulation and Benders' decomposition outperforms the cutting plane method and is able to solve test examples up to 50 nodes in size.

Appendix

Proof of Lemma 1. (a) Suppose that $X_{jm} = 1$. Let $j = i$ and $m = k$. Then constraint (7) reduces to $\beta \geq t_{jjmm} + z_\gamma \delta_{jjmm} = 2d_{jm} + z_\gamma \sqrt{2}\sigma_{jm}$, which implies that a feasible solution with $X_{jm} = 1$ gives a worse objective function value than the current available upper bound β^U. Hence, $X_{jm} = 0$ is a valid cut for SpHCP^L.

(b) Suppose that $X_{jm} = 1$. Then constraint (7) reduces to

$$\sum_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm})X_{ik} \geq \min_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm}) \sum_{k=1}^{N} X_{ik} = \min_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm}),$$

which is greater than β^U according to the assumption. This shows that any feasible solution with $X_{jm} = 1$ gives a worse objective function value than β^U. Therefore, $X_{jm} = 0$ is a valid cut for SpHCP^L.

(c) Suppose that $X_{jm} = 1$. Then constraint (7) reduces to

$$\beta \geq \sum_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm})X_{ik} \leq \max_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm}) \sum_{k=1}^{N} X_{ik} = \max_{k=1}^{N}(t_{ijkm} + z_\gamma \delta_{ijkm}) < \beta^L,$$

which is redundant for the given i, j, m.

(d) Suppose that $X_{jm} = 1$ and assume $d_{mi} = \max_{\ell=1}^{N} d_{m\ell}$ and $X_{in} = 1$. It follows from constraint (7) and the triangle inequality property that

$$\begin{array}{rcl} \beta & \geq & (t_{ijnm} + z_\gamma \delta_{ijnm}) \\ & = & d_{in} + \alpha d_{nm} + d_{mj} + z_\gamma \sqrt{(\sigma_{in})^2 + \alpha^2(\sigma_{nm})^2 + (\sigma_{mj})^2} \\ & \geq & \alpha d_{im} + d_{mj} + z_\gamma \sigma_{mj} \\ & = & d_{mj} + \alpha \max_{\ell=1}^{N} d_{m\ell} + z_\gamma \sigma_{mj} \\ & > & \beta^U. \end{array}$$

This shows that any feasible solution with $X_{jm} = 1$ gives a worse objective function value than β^U. Hence $X_{jm} = 0$ is a valid cut.

(e) When $X_{jm} = 0$, the right-hand side of constraint (7) for any i and the corresponding j, m is non-positive. Clearly, this constraint is redundant for SpHCP^L.

References

[1] S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *Eur. J. of Oper. Res.*, 190(1):1–21, 2008.

[2] O. Baron, O. Berman, and D. Krass. Facility location with stochastic demand and constraints on waiting time. *Manufaturing & Service Oper. Management*, 10:484–505, 2008.

[3] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[4] J. F. Campbell. Location and allocation for distribution systems with transshipments and transportation economies of scale. *Ann.of Oper. Res.*, 40:77–99, 1992.

[5] J. F. Campbell. Integer programming formulations of discrete hub location problems. *Eur. J. of Oper. Res.*, 72:387–405, 1994.

[6] J. F. Campbell, A. Ernst, and M. Krishnamoorthy. Hub location problems. In H. Hamacher and Z. Drezner, editors, *Location Theory: Applications and Theory*, pages 373–406. Springer-Verlag, 2001.

[7] I. Contreras, J.-F. Cordeau, and G. Laporte. Stochastic uncapacitated hub location. HEC Montreal and Interuniversity Res. Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal, Canada, 2010.

[8] I. Contreras, J. A. Diaz, and E. Fernandez. Branch-and-price for large-scale capacitated hub location problems with single assignment. *INFORMS J. on Comput.*, to appear.

[9] R. S. de Camargo, G. Miranda, and H. P. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Comput. and Oper. Res.*, 35(4):1047–1064, 2008.

[10] A. T. Ernst, H. Hamacher, H. Jiang, M. Krishnamoorthy, and G. Woeginger. Uncapacitated single and multiple allocation p-hub center problems. *Comput. and Oper. Res.*, 36:2230–2241, 2009.

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

[12] A. T. Ernst and M. Krishnamoorthy. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *Eur. J. of Oper. Res.*, 104(1):100–112, 1998.

[13] A. S. Fotheringham. A new set of spatial interaction models: The theory of competing destinations. *Environment and Planning A*, pages 15–36, 1983.

[14] O. Garnett, A. Mandelbaum, and M. Reiman. Designing a call center with impatient customers. *Manufacturing and Service Oper. Management*, 4(3):208–227, 2002.

[15] I. Gurvich, M. Armony, and A. Mandelbaum. Service-level differentiation in call centers with fully flexible servers. *Management Sci.*, 54(2):279–294, 2008.

[16] S. Juette, O. Gavriliouk, and H. Hamacher. Polyhedral analysis of uncapacitated single allocation p-hub center problems. Technical Report 109, FB Mathematik, TU Kaiserslautern, 2007.

[17] B. Y. Kara and B. Tansel. On the single assignment p-hub center problem. *Eur. J. of Oper. Res.*, 125(3):648–655, 2000.

[18] B. Y. Kara and B. C. Tansel. On the single-assignment p-hub center problem. *Eur. J. of Oper. Res.*, 125(3):648–655, 2000.

[19] V. Marianov and D. Serra. Location models for airline hubs behaving as m/d/c queues. *Comput. & Oper. Res.*, 30:983–1003, 2003.

[20] T. Meyer, A. T. Ernst, and M. Krishnamoorthy. A 2-phase algorithm for solving the single allocation p-hub center problem. *Comput. and Oper. Res.*, 36:3143–3151, 2009.

[21] M. E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *Eur. J. of Oper. Res.*, 32:393–404, 1987.

[22] H. W. Samir Elhedhli. A Lagrangean heuristic for hub-and-spoke system design with capacity selection and congestion. *INFORMS J. on Comput.*, 22:282–296, 2010.

[23] S. Savage. *Decision Making with Insight.* Duxbury Press, 2003.

[24] T. Sim, T. Lowe, and B. Thomas. The stochastic p-hub center problem with service-level constraints. *Comput. and Oper. Res.*, 36:3166–3177, 2009.

[25] D. Skorin-Kapov, J. Skorin-Kapov, and M. E. O'Kelly. Tight linear programming relaxations of uncapacitated p-hub median problems. *Eur. J. of Oper. Res.*, 94:582–593, 1996.

[26] J. Wang. The β-reliable median on a network with descrete probabilistic demand weights. *Oper. Res.*, 55:966–975, 2007.

[27] T.-H. Yang. Stochastic air freight hub location and flight routes planning. *Appl. Math. Model.*, 33:4424–4430, 2009.