An Improved Algorithm for Fixed-Hub Single Allocation Problems

Dong-Dong Ge1 · Zi-Zhuo Wang2 · Lai Wei3 · Jia-Wei Zhang4

Received: 25 October 2015 / Revised: 10 October 2016 / Accepted: 31 October 2016 / Published online: 24 December 2016
© Operations Research Society of China, Periodicals Agency of Shanghai University, Science Press, and Springer-Verlag Berlin Heidelberg 2016

Abstract This paper discusses the fixed-hub single allocation problem (FHSAP). In this problem, a network consists of hub nodes and terminal nodes. Hubs are fixed and fully connected; each terminal node is assigned to a single hub which routes all its traffic. The goal is to minimize the cost of routing the traffic in the network. In this paper, we propose a new linear programming (LP) relaxation for this problem by incorporating a set of validity constraints into the classical formulations by Ernst and Krishnamoorthy (Locat Sci 4:139–154, Ann Op Res 86:141–159). A geometric rounding algorithm is then used to obtain an integral solution from the fractional solution. We show that by incorporating the validity constraints, the strengthened LP often provides much tighter upper bounds than the previous methods with a little more

The first author was supported by the National Natural Science Foundation of China (No. 11471205).

Dong-Dong Ge
dongdong@mail.shufe.edu.cn
Zi-Zhuo Wangzwang@umn.edu
Lai Weilaiwi@umich.edu
Jia-Wei Zhangjzhang@stern.nyu.edu

1 School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China
2 Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN 55455, USA
3 Stephen M. Ross School of Business, University of Michigan, Ann Arbor, MI 48109, USA
4 Leonard N. Stern School of Business, New York University, New York, NY 10012, USA
computational effort and the solution obtained often has a much smaller gap with the optimal solution. We also formulate a robust version of the FHSAP and show that it can guard against data uncertainty with little costs.

**Keywords** Hub location · Network design · Linear programming · Worst-case analysis

**Mathematics Subject Classification** 90

1 Introduction

Hub-and-spoke networks have been widely used in transportation, logistics, and telecommunication systems. In such networks, traffic is routed from numerous nodes of origin to specific destinations through hub facilities. The use of hub facilities allows for the replacement of direct connections between all nodes with fewer, indirect connections. One main benefit is the economies of scale as a result of the consolidation of flows on relatively few arcs connecting the nodes. In the United States, hub-and-spoke routing is practically universal. Airlines adopted it after the industry was deregulated in 1978. Many logistics service providers such as UPS and FedEx also have distribution systems using hub-and-spoke structure.

Given its widespread use, it is of practical importance to design efficient hub-and-spoke networks. In the literature, such problems are often referred to as hub location problems, in which two major questions are studied (1) where hubs should be located and (2) how the traffic/flow should be routed. We refer the readers to [1] for a comprehensive review of the literature on hub location problems.

In this paper, we focus on a sub-problem which is called the fixed-hub single allocation problem (FHSAP). In the FHSAP, the locations of the hubs are fixed, and the decisions are to assign each terminal nodes to a unique hub node. Although the FHSAP is a sub-problem of the hub location problems, it is still of great interest. First, in many practical situations, the locations of the hubs are predetermined and remain unchanged in the medium to long term. In such cases, the hubs can be viewed as fixed and only the assignment of terminal nodes needs to be decided. Second, the number of nodes that can be used as hubs are usually small, which makes it possible to enumerate all possible locations of the hubs to find the optimal location. Therefore, solving the fixed-hub allocation problem efficiently would be of great help for solving the hub location problem. Moreover, even confined to fixed hubs, optimally assigning terminal nodes to hubs is still a challenging task. Indeed, it is known that FHSAP is NP-hard even for problem with three hubs [2]. Therefore, designing efficient algorithms to solve FHSAP is still of great interest, both to researchers and practitioners.

To address the FHSAP, several prior approaches have been proposed. In O’Kelly [3], the author proposes a quadratic integer program to model this problem. The formulation is non-convex and thus hard to solve. Therefore, the author proposes two heuristics to solve it. Following [3], several other heuristics are proposed, see, e.g., Klincewicz [4], Campbell [5], and Skorin-Kapov et al. [6].

One major method to solve the FHSAP is to use a linearization model for the quadratic integer program in [3]. Several such linearizations are developed [7–11].
One of the earliest such linearization model is introduced by [9], in which a natural LP relaxation of the quadratic integer program is obtained. This LP relaxation is quite attractive: Skorin-Kapor et al [6] show that this LP relaxation is very tight and outputs integral solutions automatically in 95% of the instances they test. However, the size of this LP relaxation is relatively large, and thus restricts its applications to large-scale problems. To solve this problem, Ernst and Krishnamoorthy [7,8] propose a further relaxation of the model by [9]. The idea of the further relaxation is to use combined flow variables, and the size of this LP is significantly smaller than those in [9] and [6]. However, in some situations, the relaxation has a large gap with the optimal solution.

In this paper, one of our contributions is to propose a new LP relaxation for the FHSAP. Our new LP relaxation is based on the one proposed by [7,8], but we add a set of flow validity constraints to it. We show that by adding the flow validity constraints, we can often tighten the gap between the LP relaxation of [7,8] and the optimal solution, and yield integer solutions more frequently. Moreover, it comes with reasonable computational costs. Therefore, we believe that our approach is a good balance between the LP relaxation by [9] and [7,8].

Besides finding a suitable LP relaxation, another important question is how to round a fractional solution of the LP into a feasible solution to the FHSAP. In this paper, we adopt a geometric rounding algorithm introduced by Ge et al. [12]. In Ge et al. [12], the authors propose a random geometric rounding scheme for a class of assignment problem. They prove that this rounding technique, by applying to the FHSAP, leads to a constant-ratio approximation algorithm for the equilateral structure. In this paper, we show that our newly proposed LP relaxation combined with the geometric rounding algorithm can deliver good solutions to the FHSAP efficiently. It is worth noting that another dependent rounding scheme by Kleinberg and Tardos [13] can also be adopted to round the solutions.

In practical cases, the demands in FHSAP may be unknown. To tackle such situations, we propose a robust programming approach for the FHSAP when the demands are only known to be within a certain convex set. We derive a convex programming relaxation for the robust formulation which can be solved efficiently. We show in our numerical tests that by employing the decisions of the robust model, we can guard against the demand uncertainty with little cost, therefore it might be of practical interest.

The remainder of the paper is organized as follows: In Sect. 2, we introduce the model and the LP relaxation we propose for the FHSAP. Then we introduce the geometric rounding scheme in Sect. 3. In Sect. 4, we perform numerical tests to show that our proposed approach can indeed obtain better solutions to this problem. In Sect. 5, we establish a robust model for the FHSAP, and study the solution of the robust model. Then we conclude our paper in Sect. 6.

2 Model and Formulation

This section defines the FHSAP, reviews and modifies previously proposed mathematical programs. By the terminology of communication networks, the problem is to build a two-level network consisting of hubs and terminal nodes (see Fig. 1 for an
Fig. 1 An illustration of the two-level network

In the FHSAP, we assume that there are \( k \) fixed hubs denoted by \( \mathcal{H} = \{1, 2, \cdots, k\} \) (airports, routers, concentrators, etc.), which are transit nodes that are used to route traffic. There are \( n \) terminal nodes denoted by \( \mathcal{C} = \{1, 2, \cdots, n\} \) (cities, computers, etc.) which represent the origins and the destinations of the traffic. Here, all hubs are fully connected and each terminal node is connected to exactly one hub.

In this network, there is a demand \( d_{ij} \) to be routed from \( i \) to \( j \), for each pair of \((i, j)\). In order to route the demands between two terminal nodes, the original node has to deliver all its demands to which it is assigned to. Then this hub sends them to the hub the destination node is assigned to (this step is skipped if both nodes are assigned to the same hub). Finally, the destination node gets the demands from its hub. No direct routing between two terminal nodes is permitted. Two types of costs are counted during the transportation, a per unit transportation cost \( c_{is} \) to transport demand from terminal node \( i \) to hub \( s \) and a per unit transportation cost \( c_{st} \) to transport demand from hub \( s \) to hub \( t \). The problem is to assign a hub for each terminal node such that the total transportation cost is minimized. The first mathematical formulation for this problem is by O’Kelly [3], in which he formulated it as a quadratic integer program. Define \( \bar{x} = \{x_{is} : i \in \mathcal{C}, s \in \mathcal{H}\} \) to be the assignment variables. The quadratic formulation for the FHSAP is as following problem FHSAP-QP:

\[
\begin{align*}
\text{minimize} \quad & \sum_{i,j \in \mathcal{C}} d_{ij} \left( \sum_{s \in \mathcal{H}} c_{is} x_{is} + \sum_{t \in \mathcal{H}} c_{jt} x_{jt} + \sum_{s,t \in \mathcal{H}} c_{st} x_{is} x_{jt} \right) \\
\text{s.t.} \quad & \sum_{s \in \mathcal{H}} x_{is} = 1, \quad \forall i \in \mathcal{C}, \\
& x_{is} \in \{0, 1\}, \quad \forall i \in \mathcal{C}, s \in \mathcal{H}.
\end{align*}
\]

\(^{1}\) In fact, his formulation is for a more general problem, the uncapacitated single allocation \( p \)-hub median problem (USApHMP). In this paper, we only confine our discussion to the FHSAP, and thus adapt his formulation (and later formulations) to the FHSAP.
Here, we assume that all coefficients $d_{ij}$, $c_{is}$, $c_{jt}$, $c_{st}$ are non-negative and $c_{st} = c_{ts}$, $c_{ss} = 0$, for all $i, j \in C$ and $s, t \in H$. Note that the transportation cost from cities to hubs, $\sum_{i,j \in C} d_{ij} (\sum_{s \in H} c_{is} x_{is} + \sum_{t \in H} c_{jt} x_{jt})$, is linear in $x$. We later call it the linear cost of the objective function and denote it by $L(x)$. Similarly, we call the other part of the objective function the inter-hub cost or quadratic cost, and denote it by $Q(\vec{x})$.

Campbell [9] linearized O’Kelly’s model by formulating a mixed integer linear program (MILP) as following problem MILP1:

\[
\begin{align*}
\text{minimize} & \sum_{i,j \in C} \sum_{s,t \in H} d_{ij} (c_{is} + c_{st} + c_{jt}) X_{ijst} \\
\text{s.t.} & \sum_{s,t \in H} X_{ijst} = 1, \quad \forall i, j \in C, \\
& \sum_{t \in H} X_{ijst} = x_{is}, \quad \forall i \in C, s \in H, \\
& \sum_{s \in H} X_{ijst} = x_{jt}, \quad \forall j \in C, t \in H, \\
& X_{ijst} \geq 0, \quad \forall i \in C, j \in C, s \in H, t \in H, \\
& x_{is}, x_{jt} \in \{0, 1\}, \quad \forall i \in C, s \in H.
\end{align*}
\]

Here, $X_{ijst}$ is the portion of the flow from city $i$ to city $j$ via hub $s$ and $t$ sequentially. The formulation involves $O(n^2k^2)$ non-negative variables and $O(n^2k)$ constraints. This formulation enables us to obtain an LP relaxation for the FHSAP by replacing the zero-one constraints with non-negative constraints. In the following, we refer this LP relaxation as LP1. As shown in [6], LP1 is usually very tight and often produces integer solutions. However, the size of LP1 is relative large, which restricts its applications to large-scale problems.

In order to reduce the solution complexity, Ernst and Krishnamoorthy [7,8] propose a flow formulation to obtain a further relaxation of this problem. In this formulation, one does not need to specify the route for a pair of terminal nodes $i$ and $j$, i.e., one does not need the decision variable $X_{ijst}$. Instead, one defines $\vec{Y} = \{Y_{is}^i : i \in C, s \in H, s \neq t\}$ where $Y_{is}^i$ is the total amount of the flow originated from city $i$ and routed from hub $s$ to a different hub $t$. Then the FHSAP can be bounded from below by:

Problem MILP2

\[
\begin{align*}
\text{minimize} & \sum_{i \in C} \sum_{s \in H} c_{is} (O_i + D_i) x_{is} + \sum_{i \in C} \sum_{s,t \in H, s \neq t} c_{st} Y_{st}^i \\
\text{s.t.} & \sum_{s \in H} x_{is} = 1, \quad \forall i \in C, \\
& \sum_{t \in H, i \neq s} Y_{st}^i - \sum_{t \in H, i \neq s} Y_{is}^i = O_i x_{is} - \sum_{j \in C} d_{ij} x_{js}, \quad \forall i \in C, s \in H, \\
& x_{is} \in \{0, 1\}, \quad \forall i \in C, s \in H, \\
& Y_{st}^i \geq 0, \quad i \in C, s, t \in H, s \neq t
\end{align*}
\]
where $O_i = \sum_{j \in C} d_{ij}$ and $D_i = \sum_{j \in C} d_{ji}$ denote the total demands from and to $i$, respectively. Note that this modified formulation involves only $O(nk^2)$ non-negative variables and $O(nk)$ linear constraints, which decreases from that of MILP by a factor of $n$. We can then obtain an LP relaxation from MILP2, which we denote by LP2.

To see that MILP2 is indeed a further relaxation of the problem, note that any feasible assignment $\bar{x}$ to the FHSAP with the flow vector $\bar{Y}$ is always a feasible solution to MILP2 with the objective value equal to the transportation cost. Since MILP2 reduces the formulation size by $n$, its linear relaxation is usually very easy to solve. However, despite that it is proved that MILP2 is an exact formulation when all the costs in the system are equal, in general, there might be a positive gap between the optimal value of MILP2 and the true optimal solution. And in our numerical tests, we find that the gap sometimes is quite large. Therefore, it is useful to find an improved formulation of MILP2 without adding too much complexity.

In the following, we propose a stronger formulation than MILP2. The main idea is to add a set of validity constraints based on the following observation:

**Lemma 2.1** Let $\bar{x}$ and $\bar{Y}$ be defined as in MILP2. For any $i \in C$ and $s \in H$, we have
\[
\sum_{t \in H : t \neq s} Y_{st}^i + \sum_{t \in H : t \neq s} Y_{ts}^i = \sum_{j \in C} d_{ij} |x_{is} - x_{js}|. \tag{2.2}
\]

**Proof** We verify Eq. (2.2) in two cases.

1. If $x_{is} = 0$, then

\[
\sum_{t \in H : t \neq s} Y_{st}^i + \sum_{t \in H : t \neq s} Y_{ts}^i = \sum_{t \in H : t \neq s} Y_{ts}^i = \sum_{j \in C} d_{ij} x_{js} = \sum_{j \in C} d_{ij} |x_{is} - x_{js}|.
\]

2. If $x_{is} = 1$, then

\[
\sum_{t \in H : t \neq s} Y_{st}^i + \sum_{t \in H : t \neq s} Y_{ts}^i = \sum_{t \in H : t \neq s} Y_{st}^i = \sum_{j \in C : x_{js} = 0} d_{ij} = \sum_{j \in C} d_{ij}(1 - x_{js})
\]

\[
= \sum_{j \in C} d_{ij} |x_{is} - x_{js}|.
\]

Therefore, Eq. (2.2) holds in both cases.

Based on Lemma 2.1, we obtain a strengthened formulation with additional constraints
\[
\sum_{t \in H : t \neq s} Y_{st}^i + \sum_{t \in H : t \neq s} Y_{ts}^i = \sum_{j \in C} d_{ij} y_{ijs},
\]
\[
x_{is} - x_{js} \leq y_{ijs},
\]
\[
x_{js} - x_{is} \leq y_{ijs}.
\]

We call this MILP2’ problem and LP2’ relaxation. Note that LP2’ problem has both $O(n^2k + nk^2)$ variables and constraints. We further reduce the number of additional
constraints by summing up the validity constraints. We get our final formulation as following problem MILP3:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in C} \sum_{s \in \mathcal{H}} c_{is}(O_i + D_i)x_{is} + \sum_{i \in \mathcal{C}} \sum_{s,t \in \mathcal{H}; s \neq t} c_{st}Y_{st}^i \\
\text{s. t.} & \quad \sum_{s \in \mathcal{H}} x_{is} = 1, \quad \forall i \in \mathcal{C}, \\
& \quad \sum_{t \in \mathcal{H}; t \neq s} Y_{st}^i - \sum_{t \in \mathcal{H}; t \neq s} Y_{ts}^i = O_is - \sum_{j \in \mathcal{C}} d_{ij}x_{js}, \quad \forall i \in \mathcal{C}, s \in \mathcal{H}, \\
& \quad 2\sum_{i \in \mathcal{C}} \sum_{s,t \in \mathcal{H}; s \neq t} Y_{st}^i = \sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{H}} d_{is}y_{is}, \\
& \quad x_{is} - x_{js} \leq y_{ijs}, \quad \forall i, j \in \mathcal{C}, s \in \mathcal{H}, \\
& \quad x_{js} - x_{is} \leq y_{ijs}, \quad \forall i, j \in \mathcal{C}, s \in \mathcal{H}, \\
& \quad x_{is} \in \{0, 1\}, \\
& \quad Y_{st}^i, x_{is}, y_{ijs} \geq 0, \quad i \in \mathcal{C}, s, t \in \mathcal{H}, s \neq t.
\end{align*}
\]

We call the LP relaxation of MILP3 by LP3. The number of variables and constraints in MILP3 and LP3 are both \(O(n^2 k + nk^2)\). Although it does not reduce the size of LP2 significantly, computational results indicate that LP3 can be solved much more efficiently, yet the results are usually very good.

There are two implications of our results. First, it provides a tighter lower bound for the FHSAP than LP2. Second, it provides a new way to solve the FHSAP using LP relaxations. In the next section, we detail on how to obtain an integer solution from the fraction solution solved from the linear relaxations. In Sect. 4, we perform numerical tests to show the performance of our proposed approach.

### 3 Rounding Procedure: a Geometric Rounding Algorithm

Note that in the above formulations, a solution to the FHSAP can be completely defined by the assignment variables \(\{x_{is}\}\). Therefore, after solving an LP relaxation (LP1, LP2 or LP3), we only need to focus on rounding the fractional assignment variables to binary integers. We adopt the geometric rounding algorithm in [12]. In this section, we present a brief review of this algorithm. Note that in the three relaxations presented above (LP1, LP2 or LP3), we all have the constraints that \(\sum_{s \in \mathcal{H}} x_{is} = 1\). Therefore, for a terminal node \(i\), any optimal solution \(x_i = (x_{i1}, \ldots, x_{ik})\) of the LP relaxation must fall on the standard \((k - 1)\)-dimensional simplex \(\Delta_k = \{w \in \mathbb{R}^k \mid w \geq 0, \sum_{i=1}^k w_i = 1\}\). A fractional assignment vector on node \(i\) corresponds to a non-vertex point of \(\Delta_k\). Our goal is to round any fractional solution to a vertex point of \(\Delta_k\), which is of the form \(\{w \in \mathbb{R}^k \mid w_i \in \{0, 1\}, \sum_{i=1}^k w_i = 1\}\). It is clear that \(\Delta_k\) has exactly \(k\) vertices. In the following, we denote the vertices of \(\Delta_k\) by \(v_1, v_2, \ldots, v_k\), where the \(i\)th coordinate of \(v_i\) is \(1\).

Before presenting the rounding procedure, we define some notations of the geometry of the problem. For a point \(x \in \Delta_k\), connect \(x\) with all vertices \(v_1, \ldots, v_k\) of \(\Delta_k\). Denote the polyhedra which exactly has vertices \(\{x, v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k\}\) by \(A_{x,i}\). Thus, the simplex \(\Delta_k\) can be partitioned into \(k\) polyhedrons \(A_{x,1}, \ldots, A_{x,k}\), and
the interiors of any distinct pair of these \( k \) polyhedrons do not intersect. The geometric rounding algorithm is given as following geometric rounding algorithm (GRA):

1. Solve an LP relaxation of the FHSAP (LP1, LP2 or LP3). Denote the optimal solution by \( \bar{x}^* \).
2. Generate a random vector \( u \), which follows a uniform distribution on \( \Delta_k \).
3. For each \( x_i^* = (x_{i1}^*, \ldots, x_{ik}^*) \), if \( u \) falls into \( A_{x_i^*,s} \), let \( \hat{x}_{is} = 1 \); other components \( \hat{x}_{it} = 0 \).
4. Output \( \hat{x} \).

4 Computational Results

In this section, we implement our algorithm (FHSAP-GRA) and report its performance. We test the algorithm on both randomly generated instances (Table 1) and a benchmark problem dataset (Table 2). All linear programs in the experiments are solved by CPLEX version 9.0 at a workstation with 3 GHz CPU and 8 GB memory.

In Table 1, we consider three setups of the size of the problem: \( n = 50, k = 5 \); \( n = 100, k = 10 \), and \( n = 200, k = 10 \). In each of the setup, demands between any two cities are generated from uniform distributions \( U[0, 100] \) and hub-to-city costs are generated from \( U[1, 11] \). Then, we choose different distributions for the inter-hub costs to conduct our tests, which are shown in the second column. We try to solve all the three LP relaxation problems (LP1, LP2 and LP3) introduced in Sect. 2 and apply the geometric rounding algorithm to the solution we obtain. The results are shown in the three columns GRA-LP1, GRA-LP2, and GRA-LP3. Within each of the three sets of experiments, the CPU columns show the time (seconds) our program takes to solve each LP relaxation (we find that the time to perform the rounding procedure is negligible, therefore we only report the time to solve the LPs in our test results). Gap-LP1 columns show the gap between the cost of our obtained integer solution (for each fractional solution we obtain, we do the random rounding 5 000 times and pick the best results) and the optimal value of LP1. To be more precise, if we denote the optimal value of LPi by \( v_i \) and the value of an integral solution by algorithm GRA-LPi by \( w_i \), then GAP-LP1 is \( (w_i/v_1 - 1) \times 100\% \). Similarly, Gap-LP3 columns show the gap between the cost of our obtained integer solution and the optimal value of LP3. That is, GAP-LP3 is \( (w_i/v_3 - 1) \times 100\% \). Since all the LP relaxations are lower bounds of the actual optimal for the original problem, the gaps presented is an upper bound of the performance gap between the obtained solution and the true optimal allocation. Note that when the problem size is large, e.g., \( n = 200 \) and \( k = 10 \), we are not able to solve LP1, we use N/A to denote such cases.

In Table 1, we can see that there are several features using our proposed algorithm (GRA-LP3). First, although solving LP3 is not as efficient as solving LP2, it is still mostly tractable while the solution time to LP1 increases very fast and soon becomes intractable. On the other hand, the solution provided by GRA-LP3 could provide significant improvement over the solution that is obtained using LP2. In the 15 tests we presented, there are nine cases in which GRA-LP3 could produce the exact optimal solution, while there are only three if one uses GRA-LP2. Therefore, we can conclude
Table 1  Computational results

| n and k | c_{st} | \( n = 50 \) | \( n = 100 \) | \( n = 200 \) |
|---------|-------|--------------|--------------|--------------|
|         |       | GRA-LP1 |       | GRA-LP2 |       | GRA-LP3 |       | GRA-LP3 |       |
|         |       | CPU     | Gap-LP1/\% | CPU     | Gap-LP1/\% | Gap-LP3/\% | CPU     | Gap-LP1/\% | Gap-LP3/\% |
|         |       |         |           |         |           |           |         |           |           |
|         |       | 3.30    | 0.00      | 0.04    | 4.24      | 12.19     | 3.5     | 3.55      | 11.45     |
|         |       | 3.08    | 0.00      | 0.04    | 1.83      | 1.83      | 1.58    | 0.00      | 0.00      |
|         |       | 2.55    | 0.00      | 0.04    | 4.47      | 4.47      | 2.2     | 0.00      | 0.00      |
|         |       | 3.1     | 0.00      | 0.04    | 9.25      | 9.25      | 1.36    | 0.00      | 0.00      |
|         |       | 2.04    | 0.00      | 0.04    | 0.00      | 0.00      | 2.14    | 0.00      | 0.00      |
|         |       | 15.249  | 0.00      | 0.85    | 10.95     | 51.17     | 148     | 10.95     | 51.17     |
|         |       | 16.851  | 0.00      | 3.12    | 2.76      | 15.07     | 329     | 2.30      | 14.55     |
|         |       | 15.439  | 0.00      | 3.22    | 5.86      | 7.47      | 322     | 0.92      | 2.45      |
|         |       | 10.103  | 0.00      | 1.08    | 9.25      | 9.25      | 230     | 0.00      | 0.00      |
|         |       | 13.780  | 0.00      | 4.07    | 0.00      | 0.00      | 310     | 0.00      | 0.00      |
|         |       | N/A     | N/A       | 22.5    | N/A       | 33.11     | 2549    | N/A       | 33.11     |
|         |       | N/A     | N/A       | 23.1    | N/A       | 11.88     | 1750    | N/A       | 12.80     |
|         |       | N/A     | N/A       | 27.3    | N/A       | 0.72      | 3311    | N/A       | 0.00      |
|         |       | N/A     | N/A       | 20.2    | N/A       | 5.04      | 1981    | N/A       | 0.00      |
|         |       | N/A     | N/A       | 32.7    | N/A       | 0.00      | 3278    | N/A       | 0.00      |
that GRA-LP3 delivers high-quality solutions for medium-large sized problems in a reasonable amount of time.

Next, we test our algorithm using a benchmark problem set Australia post (AP), which was collected from a real postal delivery network in Australia, see [7]. The dataset contains 200 terminal nodes, the demand between each of them are specified. In [7] and [8], Ernst and Krishnamoorthy solve the $p$-hub location problems for AP dataset, and we test our algorithms on hubs their solutions specified. In particular, some of the hub-to-city cost coefficients are non-symmetric in the AP dataset. In our experiment, we make adjustment to it accordingly by specifying in-flow and out-flow coefficients separately for each $x_{is}$. The results of our tests are shown in Table 2.

In Table 2, we test 15 AP benchmark problems. Since solving FHSAP-LP1 already produced optimal integral assignments for all 15 problems in less than 120 s, we omitted it in the table. GRA-LP3 obtained optimal assignments on 14 out of 15 problems, and only 0.004% higher than the optimal cost on the remaining one, with much less time than GRA-LP1. Therefore, again, we can see that our approach is quite reliable and efficient in solving real problems.

| $n$ | $k$ | Optimal $nk$ | Optimal GRA-LP3 |
|-----|-----|--------------|-----------------|
|     |     | LP3          | GRA3            | CPU Gap 1/3%   |
| 50  | 5   | 132 367      | 132 122         | 132 372        | 6.94 0.004 |
| 50  | 4   | 143 378      | 143 200         | 143 378        | 4.04 0.000 |
| 50  | 3   | 158 570      | 158 473         | 158 570        | 1.92 0.000 |
| 40  | 5   | 134 265      | 133 938         | 134 265        | 2.17 0.000 |
| 40  | 4   | 143 969      | 143 924         | 143 969        | 1.16 0.000 |
| 40  | 3   | 158 831      | 158 831         | 158 831        | 0.60 0.000 |
| 25  | 5   | 123 574      | 123 574         | 123 574        | 0.23 0.000 |
| 25  | 4   | 139 197      | 138 727         | 139 197        | 0.17 0.000 |
| 25  | 3   | 155 256      | 155 139         | 155 256        | 0.09 0.000 |
| 20  | 5   | 123 130      | 122 333         | 123 130        | 0.11 0.000 |
| 20  | 4   | 135 625      | 134 833         | 135 625        | 0.08 0.000 |
| 20  | 3   | 151 533      | 151 515         | 151 533        | 0.05 0.000 |
| 10  | 5   | 91 105       | 89 962          | 91 105         | 0.02 0.000 |
| 10  | 4   | 112 396      | 111 605         | 112 396        | 0.01 0.000 |
| 10  | 3   | 136 008      | 135 938         | 136 008        | 0.01 0.000 |

5 Robust FHSAP

In previous sections, we studied the fixed-hub single allocation problem with deterministic demand. In practice, demand may not be exactly known to the decision makers when the allocation of the hubs has to be done. In such cases, it is of great interest for the decision maker to have a “robust” policy, which protects him from any realizations of demand. In this section, we study a robust formulation for the FHSAP and propose an efficient algorithm for it.

We adopt the notations used in Sect. 2. However, instead of knowing the pairwise demand $\vec{d} = \{d_{ij}\}$ exactly, we only know that they are in the following set:
Here, $\tilde{u} = \{u_{ij}\}$ is the nominal demand, $\Sigma = \text{diag}\{\sigma_{ij}\}$ is a weight matrix and $|| \cdot ||_p$ is the $p$-norm ($p \geq 1$) of a vector defined by

$$||x||_p = \left( \sum_{i=1}^{n} x_i^p \right)^{1/p}. \quad (5.2)$$

The right-hand side $Q$ in (5.1) is the “budget” of robustness, indicating one’s uncertainty level about the input data. Such an uncertainty set is quite common in the robust optimization literature with most common choices of $p$ to be 1, 2, or $\infty$. For a comprehensive review of the robust optimization literature, we refer the readers to Ben-Tal et al. [14].

Now we consider the robust formulation for FHSAP. We start from FHSAP-QP. In the robust formulation, we aim to minimize the worst-case cost for any demand realization that is in set $\mathcal{D}$. Therefore, the robust formulation for FHSAP-QP can be written as

$$\text{minimize}_{x_{ij} \in \{0, 1\}} \text{maximize}_{d_{ij}} \sum_{i,j \in \mathcal{C}} d_{ij} \left[ \sum_{s \in \mathcal{H}} c_{is} x_{is} + \sum_{t \in \mathcal{H}} c_{jt} x_{jt} + \sum_{s,t \in \mathcal{H}} c_{st} x_{is} x_{jt} \right]$$

$$\text{s.t.} \quad ||\Sigma^{-1}(\tilde{d} - \tilde{u})||_p \leq Q. \quad (5.3)$$

One nice feature of this robust formulation is that given a set of $\tilde{x}$, the inside maximization problem has an explicit optimal solution. Define

$$f_{ij} = \sum_{s \in \mathcal{H}} c_{is} x_{is} + \sum_{t \in \mathcal{H}} c_{jt} x_{jt} + \sum_{s,t \in \mathcal{H}} c_{st} x_{is} x_{jt},$$

the inside problem can be written as

$$\text{maximize}_{\tilde{d}} \tilde{f}^T \tilde{d}$$

$$\text{s.t.} \quad ||\Sigma^{-1}(\tilde{d} - \tilde{u})||_p \leq Q. \quad (5.4)$$

By using standard Lagrangian method, one can obtain the optimal value to (5.4) as

$$\tilde{f}^T \tilde{u} + Q ||\Sigma \tilde{f}||_q,$$

where $q = p/(p - 1)$. Therefore, the robust counterpart of (5.4) can be written as:

$$\text{minimize}_{x_{is}} \sum_{i,j \in \mathcal{C}} u_{ij} \left[ \sum_{s \in \mathcal{H}} c_{is} x_{is} + \sum_{t \in \mathcal{H}} c_{jt} x_{jt} + \sum_{s,t \in \mathcal{H}} c_{st} x_{is} x_{jt} \right]$$

$$+ Q \left( \sum_{i,j} (\sigma_{ij} \sum_{s \in \mathcal{H}} c_{is} x_{is} + \sum_{t \in \mathcal{H}} c_{jt} x_{jt} + \sum_{s,t \in \mathcal{H}} c_{st} x_{is} x_{jt})^q \right)^{1/q}$$

$$\text{s.t.} \quad x_{is} \in \{0, 1\}, \quad \forall i, s.$$
Now if the inter-hub costs are the same (w.l.o.g., all equals to one), we can write \( \sum_{s,t \in \mathcal{H}} c_{st} x_{is} x_{jt} \) as \( \sum_{r \in \mathcal{H}} |x_{ir} - x_{jr}| \). Further relaxing the binary constraints on \( x_{is} \), we obtain a convex optimization problem:

\[
\begin{align*}
\text{minimize} & \quad x_{is}, Z_{ij} \quad \sum_{i,j} u_{ij} Z_{ij} + Q t \\
\text{s.t.} & \quad Z_{ij} \geq \sum_{s \in \mathcal{H}} c_{is} x_{is} + \sum_{t \in \mathcal{H}} c_{jt} x_{jt} + \sum_{r \in \mathcal{H}} |x_{ir} - x_{jr}| \\
& \quad t \geq \| \sum \Sigma Z \|_q, \\
& \quad \sum_{s \in \mathcal{H}} x_{is} = 1, \quad \forall i, \\
& \quad 0 \leq x_{is} \leq 1, \quad \forall i, s.
\end{align*}
\tag{5.5}
\]

In general, the optimal solution in (5.5) is fractional. However, as we will show in the simulation results, in most of our tested case, an integer solution is automatically obtained. And for the cases in which a fraction solution exists, we can again apply the geometric rounding technique introduced in Sect. 3 to obtain an integer solution.

### 5.1 Numerical Experiments

In this section, we perform numerical tests using the robust approach we proposed above. We show that the robust approach can indeed guard against data uncertainty without compromising much the average performance (both computational efficiency and solution quality).

In this section, we use the robust approach (5.3) with \( p = 2 \). In such cases, the robust counterpart (5.5) will be a second-order cone program. In the tests, we study different experimental setups. For each setup, we consider the following two approaches:

1. Simply use the nominal demand to obtain the solution. We use the LP3-GRA approach discussed in Sect. 3. The integer solution obtained (after rounding) is denoted by \( \tilde{x} \).
2. We use the robust approach with a predetermined budget of robustness \( Q \). We then solve the program (5.5) to obtain the optimal solution (combined with the same geometric rounding procedure if the solution is fractional). We denote this solution by \( \hat{x} \).

To evaluate the solutions, we use two performance measures. First, we evaluate the solutions at the nominal demand. We write \( \hat{F}(x) \) to denote the cost of using \( x \) at the nominal demand and define Gap 1 as \( \frac{\hat{F}(\tilde{x}) - \hat{F}(\hat{x})}{\hat{F}(\tilde{x})} \times 100\% \) to measure the percentage gap between the costs in nominal cases. Second, we compute the worst-case costs of our solutions for the demand in the robust set we defined. That is, we compute the objective values of (5.5). We use \( \hat{F}(x) \) to denote the worst-case cost of using \( x \) and define Gap 2 as \( \frac{\hat{F}(\tilde{x}) - \hat{F}(\hat{x})}{\hat{F}(\tilde{x})} \times 100\% \).

In our experiment, we test the 15 AP benchmark problems. We make two adjustments in both the nominal demand test and the robust approach test. First, we adjust the city-to-hub cost symmetric as in Table 2. Second, as the program (5.5) assumes equal inter-hub costs, we also adjust the inter-hub costs by replacing them with the maximal...
inter-hub cost. When testing the robust approach, we use the nominal demand of AP dataset as the parameter $u$ which could be interpreted as the mean of uncertain demand in robust cases. The parameter $\sigma$ is generated from a standard lognormal distribution, and then multiplied by 100. The parameter $Q$ varies with the magnitude of demands in different benchmark cases. The results are shown in Table 3.

In Table 3, Time 1 is the computational time of the regular approach and Time 2 is the computational time of the robust approach. First we can observe that although Time 2 is larger than Time 1, it is still tractable. The computational cost of robust tests is affordable in these cases. Regarding the performances of the two approaches, we can see that Gap 1 is smaller than 1% in 11 cases out of 15 and is less than Gap 2 in 12 cases while Gap 2 is larger than 1% in 9 cases in 15 and is almost 15% in the 20-city-2-hub case. Therefore, the additional cost for the solution $\hat{x}$ that one has to pay at the nominal demand is much smaller than the potential additional costs one need to pay if one use $\tilde{x}$ but the demand turns out to be adverse. Or in other words, the benefit of the robust approach in this case significantly outweighs the cost.

6 Conclusion

In this paper, we studied the fixed-hub single allocation problem. We made two contributions. First, we proposed a new solution approach for this problem by establishing a new LP relaxation formulation. This new LP relaxation lies in between two known relaxations in the literature, and by showing numerical results of this relaxation, we show that it seems to have a good balance between computational complexity and the solution quality. Second, we propose a robust version of the FHSAP problem. The robust problem aims to minimize the worst-case cost when the demand is known to be in a certain set. We propose an algorithm to solve the robust FHSAP problem and show that indeed it can guard against data uncertainty with relatively little costs. We believe

Table 3 Robust approach test

| n  | k  | Q  | Time 1 | Time 2 | $\tilde{F}(\tilde{x})$ | $\tilde{F}(\hat{x})$ | Gap 1/% | $\tilde{F}(\tilde{x})$ | $\tilde{F}(\hat{x})$ | Gap 2/% |
|----|----|----|--------|--------|---------------------|---------------------|--------|---------------------|---------------------|--------|
| 50 | 5  | 100| 9.51   | 1378.08| 164806  | 165833  | 0.62   | 5951208            | 5741017            | 3.66   |
| 50 | 4  | 100| 4.87   | 810.92 | 153588  | 153802  | 0.14   | 5388655            | 5373987            | 0.27   |
| 50 | 3  | 100| 2.81   | 750.52 | 163094  | 164633  | 0.94   | 5393514            | 5364942            | 0.53   |
| 40 | 5  | 100| 4.52   | 317.26 | 159470  | 159470  | 0.00   | 4394564            | 4394587            | 0.00   |
| 40 | 4  | 100| 3.06   | 288.05 | 155004  | 155004  | 0.00   | 4053568            | 4053573            | 0.00   |
| 40 | 3  | 100| 1.97   | 297.74 | 164604  | 165257  | 0.40   | 5037397            | 4997983            | 0.79   |
| 25 | 5  | 200| 1.72   | 50.18  | 154884  | 155293  | 0.26   | 4028791            | 3958828            | 1.77   |
| 25 | 4  | 200| 1.42   | 41.53  | 160143  | 161396  | 0.78   | 3795295            | 3711129            | 2.27   |
| 25 | 3  | 200| 1.63   | 59.45  | 159383  | 160119  | 0.46   | 4011292            | 3964292            | 1.19   |
| 20 | 5  | 400| 1.31   | 18.43  | 151508  | 154375  | 1.89   | 4242145            | 4230509            | 0.28   |
| 20 | 4  | 400| 0.91   | 15.02  | 155042  | 159317  | 2.76   | 4314851            | 4232389            | 1.95   |
| 20 | 3  | 400| 1.28   | 12.06  | 153353  | 155955  | 1.46   | 7627887            | 6676250            | 14.25  |
| 10 | 5  | 600| 2.31   | 4.51   | 122539  | 126367  | 3.12   | 1669913            | 1652178            | 1.07   |
| 10 | 4  | 600| 1.50   | 4.35   | 121391  | 121524  | 0.11   | 1805759            | 1718371            | 5.09   |
| 10 | 3  | 600| 0.77   | 3.35   | 137723  | 139034  | 0.95   | 2685820            | 2481755            | 8.22   |
that both of our contributions may help people to find the desired model/approach when facing such problems in practice.

Acknowledgements The authors would like to thank anonymous referees for their helpful comments.

References

[1] Campbell, J., Ernst, A., Krishnamoorthy, M.: Hub location problems. In: Drezner, Z., Hamcher, H.W. (eds.) Facility Location: Application and Theory. Springer, Berlin (2002)
[2] Sohn, J., Park, S.: The single-allocation problem in the interacting three-hub network. Networks 35(1), 17–25 (2000)
[3] O’Kelly, M.: A quadratic integer program for the location of interacting hub facilities. Eur. J. Oper. Res. 33, 393–402 (1987)
[4] Klincewicz, J.: Heuristics for the $p$-hub location problem. Eur. J. Oper. Res. 53, 25–37 (1991)
[5] Campbell, J.: Hub location and the $p$-hub median problem. Oper. Res. 44, 925–935 (1996)
[6] Skorin-Kapov, D., Skorin-Kapov, J., O’Kelly, M.: Tight linear programming relaxations of uncapacitated $p$-hub median problems. Eur. J. Oper. Res. 94, 582–593 (1996)
[7] Ernst, A., Krishnamoorthy, M.: Efficient algorithms for the uncapacitated single allocation $p$-hub median problem. Locat. Sci. 4(3), 139–154 (1996)
[8] Ernst, A., Krishnamoorthy, M.: Solution algorithms for the capacitated single allocation hub location problem. Ann. Oper. Res. 86, 141–159 (1999)
[9] Campbell, J.: Integer programming formulation of discrete hub location problems. Eur. J. Oper. Res. 72, 387–405 (1994)
[10] O’Kelly, M., Bryan, D., Skorin-Kapov, D., Skorin-Kapov, J.: Hub network design with single and multiple allocation: a computational study. Locat. Sci. 4, 125–138 (1996)
[11] O’Kelly, M., Skorin-Kapov, D., Skorin-Kapov, J.: Lower bounds for the hub location problem. Manag. Sci. 41, 713–721 (1995)
[12] Ge, D., He, S., Ye, Y., Zhang, J.: Geometric rounding: dependent randomized rounding scheme. J. Comb. Optim. 22(4), 699–725 (2011)
[13] Kleinberg, J., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. J. ACM 49(5), 616–639 (2002)
[14] Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust Optimization. Princeton University Press, Princeton (2009)