Working Paper Series

10/2008

Uncapacitated single and multiple allocation *p*-hub center problems

Ernst, A.T., Hamacher, H., Jiang, H., Krishnamoorthy, M. and Woeginger, G.

Forthcoming in *Computers & Operations Research*

CAMBRIDGE
Judge Business School

These papers are produced by 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.

Corresponding author contact details are as follows:

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

Please address enquiries about the series to:

Research Support Manager
Judge Business School
Trumpington Street
Cambridge CB2 1AG, UK
Tel: 01223 760546 Fax: 01223 339701
Email: research-support@jbs.cam.ac.uk

Uncapacitated Single and Multiple Allocation p-Hub Center Problems

Andreas T. Ernst[a], Horst Hamacher[b],
Houyuan Jiang[c], Mohan Krishnamoorthy[a], Gerhard Woeginger[d]

[a] CSIRO Mathematical and Information Sciences, Private Bag 10, Clayton South MDC, Clayton, VIC 3169, Australia

[b] Fachbereich Mathematik, Universitaet Kaiserslautern, Postfach 3049, D-67653 Kaiserslautern, Germany

[c] Judge Business School, University of Cambridge, Trumpington Street, Cambridge CB2 1AG, United Kingdom

[d] Faculty of Mathematical Sciences, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

The hub median problem is to locate hub facilities in a network and to allocate non-hub nodes to hub nodes such that the total transportation cost is minimized. In the hub center problem, the main objective is one of minimizing the maximum distance/cost between origin destination pairs. In this paper, we study uncapacitated hub center problems with either single or multiple allocation. Both problems are proved to be NP-hard. We even show that the problem of finding an optimal single allocation with respect to a given set of hubs is already NP-hard. We present integer programming formulations for both problems and propose a branch-and-bound approach for solving the multiple allocation case. Numerical results are reported which show that the new formulations are superior to previous ones.

Keywords: Facility planning and design, Hub center, NP-hard, heuristic.

1 Introduction

Hubs form critical elements in many airline, transportation, postal and telecommunications networks. They are centralized facilities in these networks whose functions are to consolidate, switch and sort flows. Flow concentration and consolidation on the arcs that connect hub nodes ("hub arcs") allow us to exploit transportation flow economies. It is also possible to eliminate many expensive direct connection arcs between origin destination pairs.

Typical applications of hub location include airline passenger travel [2], telecommunication systems [7] and postal networks [11]. Reviews of the hub location literature including theory and applications can be found in [1, 4]. Two major classes of objective functions are considered, median and center objectives.

The *hub median* problem is to locate hub facilities in a network and to allocate non-hub nodes to hub nodes such that the total transportation cost is minimized. It is applicable, for instance, in airline and telecommunication systems. This model of hub network design can sometimes lead to unsatisfactory results when worst-case origin-destination distances are excessively large. In order to avoid this drawback, *hub center* problems may be a better suited model. Here the main objective is to minimize the maximum distance or cost between origin-destination pairs. This objective is particularly important for the delivery of perishable or time sensitive items.

While the hub median problem has been well studied in the literature - including several problem variants like the latest arrival hub location problem by Kara and Tansel [19] - the hub center problem has attained much less attention. It was introduced in [24, 3]. Campbell [3] formulates it as a quadratic program and reformulates it as a linear program. Several linearizations of the quadratic program are proposed by Kara and Tansel [18], who also provide an NP-completeness proof for the single allocation case and numerical comparisons for the linearizations. A single-relocation algorithm with tabu search is developed for a hub center problem considering flow volumes in Pamuk and Sepil [25] where extensive numerical experiments are also carried out. Based on a previous version of our paper from 2002, Hamacher and Meyer [16] proposed a solution approach for the hub center problem which consists of an iterative solution of hub covering problems. A polyhedral analysis of the hub center polytope can be found in [17].

In this paper we study the *uncapacitated single allocation p-hub center problem (USApHCP)* and *uncapacitated multiple allocation p-hub center problem (UMApHCP)* defined on a complete, symmetric network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with node set $\mathcal{N} = \{1, 2, \ldots, N\}$ and arc set $\mathcal{E}$. Each arc $[i, j]$ has infinite capacity and cost $c_{ij} = c_{ji}$ satisfying the triangle inequality. We also assume that $c_{ij} \geq 0$ and often $c_{ii} = 0$ though this is not required. Both USApHCP and UMApHCP require the selection of a complete subnetwork of *p hubs* where each hub has infinite node capacity for flow collection, transfer and distribution. The cost of flow between hub nodes is discounted by a factor $\alpha \in [0, 1]$, such that the cost on arcs $[m, k]$ connecting two hub nodes is reduced to αc_{mk}. Note that in general there will be multiple equivalent optimal solutions since only the length of the longest path matters for the objective and hence there are often multiple choices for paths between other i and j pairs. However we will generally assume that in this case the optimal solution is chosen to keep path lengths as short, eg by not including more than two hubs even if a third or fourth hub could be included between some pairs of nodes in the optimal solution.

In USApHCP, each non-hub node is allocated to a unique hub, whereas in UMApHCP, non-hub nodes can be allocated to more than one hub. Once the hub nodes have been selected, transportation between origin-destination pairs $[i, j]$ can only take place via allocated hub nodes. Hence any transportation path has the form (i, k, l, j), where i is allocated to hub k and j is allocated to hub l.

The goal in the hub center problem is to minimize the discounted cost of the largest origin-destination path, i.e.

$$\min \max_{i,j \in \mathcal{N}} (c_{ik} + \alpha c_{km} + c_{mj}),$$

where k and m are the hub nodes allocated to i and j, respectively. This represents the situation where the performance of the network is to be optimised (eg the amount of time to move goods between any pair of points in a mail network or supply chain). The discount on hub arcs in this case represents the fact that faster links, for example faster planes or higher throughput telecommunication links, can be employed between major hubs than would be economical to use over the other edges.

Depending on the application the costs could measure travel time or monetary costs. We use cost or path length interchangeably in this paper.

Some authors have commented in the context of p-hub median problems that the use of a completely interconnected hub network, where discounts apply only on inter-hub arcs, is somewhat unrealistic. Alternative models are possible. See for example [5, 26]. The virtue of the p-hub center model presented here is that it provides a simple model that allows alternative approaches and solution methods to be investigated.

In this paper, we present integer programming formulations for USApHCP and UMApHCP in Sections 2 and 3 respectively. The former is based on the innovative concept of *hub radius* and yields a two-index formulation with a linear objective function. For UMApHCP, which had not been studied before in the literature, we give two alternative three-index formulations. In Section 4, we prove that both problems are NP-hard and that the single allocation problem with respect to a given set of hubs is already NP-hard. We describe a shortest path based branch and bound method for solving UMApHCP in Section 5. Extensive numerical tests with regard to all formulations and the new branch-and-bound algorithm are presented in Section 6. They show that the proposed solution approaches are very efficient and outperform existing approaches by an order of magnitude. The results of our paper are summarized in the concluding Section 7.

2 The uncapacitated single allocation p-hub center problem

Define a binary variable X_{ik} such that $X_{ik} = 1$ if and only if node i is allocated to hub k, and $X_{kk} = 1$ if and only if k is a hub node. Let z be the maximum transportation cost between all origin-destination pairs. USApHCP is defined as a quadratic integer program in Campbell [3]:

$$
\begin{aligned}
\min \quad & \max_{i,j,k,m \in \mathcal{N}} (c_{ik} + \alpha c_{km} + c_{mj}) X_{ik} X_{jm} \\
\text{s.t.} \quad & \sum_{k=1}^{N} X_{ik} = 1, \quad i = 1, \ldots, N \\
& X_{ik} \le X_{kk}, \quad i, k = 1, \ldots, N \\
& \sum_{k=1}^{N} X_{kk} = p \\
& X_{ik} \in \{0, 1\}, \quad i, k = 1, \ldots, N.
\end{aligned}
$$

Here, the objective is to minimize the maximum transportation cost between all origin–destination pairs, the first constraint states that each node is allocated to exactly one hub, the second says that node k must be a hub if a node i is allocated to it, the third indicates exactly p hubs to be established, and the last constraint specifies X_{ik} to be binary.

Let s_{ijkm} be a binary variable such that $s_{ijkm} = 1$ if and only if i is allocated to k and j to m. A linearization of USApHCP based on s_{ijkm} is proposed in Campbell [3] and is not reproduced here.

Kara and Tansel [18] consider a few linear programming versions of the quadratic formulation for USApHCP. They also introduce the following two-index linear programming formulation.

$$
\begin{aligned}
\min \quad & z \\
\text{s.t.} \quad & z \ge \sum_{k=1}^{N} (c_{ik} + \alpha c_{km}) X_{ik} + c_{mj} X_{jm}, \quad i, j, m = 1, \ldots, N \\
& \sum_{k=1}^{N} X_{ik} = 1, \quad i = 1, \ldots, N \\
& X_{ik} \le X_{kk}, \quad i, k = 1, \ldots, N \\
& \sum_{k=1}^{N} X_{kk} = p \\
& X_{ik} \in \{0, 1\}, \quad i, k = 1, \ldots, N.
\end{aligned}
$$

This formulation has fewer variables than other known linear programming formulations. Moreover, it is reported in Kara and Tansel [18] that this formulation is more efficient in terms of computational effort required to solve it, thus making possible the solution of relatively large sized problems.

We propose next a new two-index formulation for USApHCP based on a concept which we call the *radius of hub k*. The latter is defined as a nonnegative variable r_k representing the maximum distance (cost) between hub k and the nodes that are allocated to it. Using this concept, USApHCP can be formulated as the following mixed integer linear program:

$$\min \quad z \tag{1}$$

$$\text{s.t.} \quad \sum_{k=1}^{N} X_{ik} = 1, \qquad i = 1, \ldots, N \tag{2}$$

$$X_{ik} \leq X_{kk}, \qquad i, k = 1, \ldots, N \tag{3}$$

$$\sum_{k=1}^{N} X_{kk} = p \tag{4}$$

$$r_k \geq c_{ik}\, X_{ik}, \qquad i, k = 1, \ldots, N \tag{5}$$

$$z \geq r_k + r_m + \alpha c_{km} \quad k \leq m = 1, \ldots, N \tag{6}$$

$$X_{ik} \in \{0, 1\}, \qquad i, k = 1, \ldots, N \tag{7}$$

Here, z is a free variable to represent the objective. The objective of USApHCP is to minimize the maximum of the total unit costs between any pair of nodes i and j. Constraints (2), (3), (4) are standard. Constraint (5) states that the radius of a hub is greater than or equal to the cost of going to any node which is allocated to this hub, and that the radius of a non-hub node can be as small as zero (see below). Constraint (6) ensures that the objective is no less than the travel cost between any pair of nodes allocated to hubs k and m, since r_k and r_m give the distance to the furthest node allocated to k and m respectively.

Note that r_k is not necessarily zero in an optimal solution if k is not a hub node. However, for any optimal solution (X^*_{ik}, r^*_k, z^*), we can construct a new optimal solution (X^+_{ik}, r^+_k, z^+) such that $X^+_{ik} = X^*_{ik}$ for any i and k, $z^+ = z^*$, $r^+_k = r^*_k$ for any hub node k, and $r^+_k = 0$ for any non-hub node k. Furthermore, the constraint (6) is still valid even if either or both nodes k and m are non-hub nodes because of the triangular inequality assumption and the fact that $\alpha \leq 1$.

Our new model has $N^2 + N + 1$ variables of which N^2 are binary, and $3N^2 + N + 1$ constraints. It has N more continuous variables than the one proposed in Kara and Tansel [18], but our model has fewer constraints. This reduction in the number of constraints is computationally significant for solving large-scale problems (when N is large). This observation is confirmed by the computational results to be presented in Section 6.

Constraint (6) can be strengthened by the following constraint, for $k, l = 1, 2, \ldots, N$

$$z \geq r_k + r_l + \alpha c_{kl} + (1 - \alpha)(1 - X_{kk}) \min_{i:i \neq k} c_{ik} + (1 - \alpha)(1 - X_{ll}) \min_{i:i \neq l} c_{il} \tag{8}$$

It is obvious that constraint (8) is stronger than (6). On the other hand, we need to check that (8) does not eliminate any optimal solution of our two-index formulation. When both k and l are hub nodes, (8) and (6) are identical. When both k and l are non-hub nodes, $r_k = r_l = 0$, $X_{kk} = X_{ll} = 0$, and the left-hand side of (8) is bounded above by

$$\alpha(c_{ka(k)} + c_{a(k)a(l)} + c_{a(l)l}) + (1 - \alpha)c_{a(k)k} + (1 - \alpha)c_{a(l)l} = c_{ka(k)} + \alpha c_{a(k)a(l)} + c_{a(l)l},$$

which is clearly bounded by the optimal value z^* in the two-index formulation. Here $a(k)$ and $a(l)$ are assumed to be the hubs for k and l respectively. When exactly one of k and l is a hub node, we can similarly show that (8) does not eliminate any optimal solution of our two-index formulation.

If the hub set is given in USApHCP, then we only need to allocate all nodes to hubs. This subproblem is called the hub center single allocation problem (HCSAP). Although seemingly much simpler than USApHCP, HCSAP is proved to be NP-hard in Section 4.

3 The uncapacitated multiple allocation p-hub center problem

Since in UMApHCP a node can be allocated to several different hubs, the binary variable X_{ik} needed in USApHCP is no longer required. Subsequently, we give a four-index formulation for UMApHCP. Let y_{ijkm} be a binary variable such that $y_{ijkm} = 1$ if and only if for the flow between i and j, i is allocated to k and j to m. Let Z_k be a binary variable defined by $Z_k = 1$ if and only if node k is selected to be a hub. Then UMApHCP is to find an optimal solution of the following optimization problem:

$$\min \quad z \tag{9}$$

$$\text{s.t.} \quad \sum_{k=1}^{N} Z_k = p \tag{10}$$

$$\sum_{k=1}^{N}\sum_{m=1}^{N} y_{ijkm} = 1, \qquad i,j = 1,\ldots,N, \tag{11}$$

$$\sum_{k=1}^{N} y_{ijkm} \le Z_m, \qquad i,j,m = 1,\ldots,N \tag{12}$$

$$\sum_{m=1}^{N} y_{ijkm} \le Z_k, \qquad i,j,k = 1,\ldots,N \tag{13}$$

$$z \ge \sum_{k=1}^{N}\sum_{m=1}^{N} y_{ijkm}(c_{ik} + \alpha c_{km} + c_{mj}), \qquad i,j = 1,\ldots,N \tag{14}$$

$$Z_k,\ y_{ijkm} \in \{0,1\}, \qquad i,j,k,m = 1,\ldots,N. \tag{15}$$

Constraint (10) indicates that exactly p hubs are chosen. Constraint (11) together with (15) shows that there is a unique path between each origin-destination pair. Constraints (12) and (13) imply that a node must be selected to be a hub if another node is allocated to it. Constraint (14) defines the lower bound for the objective function z, which represents the maximum transportation cost between all origin-destination pairs. We remark that some tighter constraints than (12) and (13) have been proposed in [6, 15, 21].

In the above formulation, the $(0,1)$ property of the variables y_{ijkm} enforces a unique path between each pair of origin-destination nodes, although nodes can be allocated to several hubs for collection, transfer and distribution of flows. If we drop the integrality constraint on the

y-variables, multiple paths may be allowed for each origin-destination pair. The next result states, however, that even in this situation an integral solution can easily be obtained from the optimal solution of the mixed integer program version of (9 - 15).

Proposition 3.1 *For any optimal solution of UMApHCP formulation (9 - 15), where the integrality of the variables y_{ijkm} is relaxed to $y_{ijkm} \geq 0$, there exists an optimal solution such that y is integral.*

Proof. Any optimal solution y defines a flow of minimal cost and of value 1 between nodes i and j. Either it uses for all i, j only single shortest ij-paths, thus making y a binary vector, or it uses for some pairs (i, j) different paths (i, k, m, j) and (i, k', m', j) with $0 < y_{ijkm} < 1$ and $0 < y_{ijk'm'} < 1$. Optimality implies that in this case both paths must be shortest paths. Hence the flow can be accumulated along a single shortest path, since the capacities on all arcs are unbounded. The latter single path flow is clearly another optimal flow. In this way we get an alternative optimal solution for (9 - 15) with integer-valued y. ■

The equivalence of the pure and mixed integer formulation of (9 - 15) has the advantage of a more flexible modelling. It is worth mentioning that this equivalence should not be confused with the fact that USApHCP and UMApHCP are different problems.

Traditional four-index formulations for hub median problems similar to (9 - 15) have a poor computational performance. This is also true from our computational experiments. Some much improved four-index formulations for hub median problems with multiple allocation have been proposed in [6, 10, 20]. We next propose a more compact formulation for UMApHCP.

With respect to any origin-destination pair (i, j) let $U^{ik}_{\cdot j}$ and $V^{i\cdot}_{lj}$ be two binary variables defined by $U^{ik}_{\cdot j} = 1$ if and only if i is allocated to hub k, and $V^{i\cdot}_{lj} = 1$ if and only if j is allocated to hub l. Moreover, let $C_{\max} = \max_{i,j \in \mathcal{N}} c_{ij}$. Then a three-index formulation for UMApHCP is given by

$$\min \quad z \tag{16}$$

$$\text{s.t.} \quad \sum_{k=1}^{N} Z_k = p \tag{17}$$

$$U^{ik}_{\cdot j} \leq Z_k, \qquad i, j, k = 1, \ldots, N, \tag{18}$$

$$V^{i\cdot}_{lj} \leq Z_l, \qquad i, j, l = 1, \ldots, N, \tag{19}$$

$$\sum_{k=1}^{N} U^{ik}_{\cdot j} = 1, \qquad i, j = 1, \ldots, N, \tag{20}$$

$$\sum_{l=1}^{N} V^{i\cdot}_{lj} = 1, \qquad i, j = 1, \ldots, N, \tag{21}$$

$$z \geq \sum_{k=1}^{N} (c_{ik} + \alpha c_{kl}) U^{ik}_{\cdot j} + \sum_{n=1}^{N} c_{nj} V^{i\cdot}_{nj} - \alpha(1 - V^{i\cdot}_{lj}) C_{\max}, \qquad i \geq j, l = 1, \ldots, N, \tag{22}$$

$$Z_k, U^{ik}_{\cdot j}, V^{i\cdot}_{lj}, \in \{0, 1\}, \qquad i, j, k, l = 1, \ldots, N. \tag{23}$$

Constraints (17), (18), (19) and (23) are self-explanatory. Constraints (20) and (21) specify that for any origin-destination pair (i, j), i is allocated to exactly one hub k and j to exactly one hub l.

Finally it is easy to verify that Constraints (22) and (16) define the objective function value of UMApHCP. If $l = n$, the right-hand side of (22) is equal to $c_{ik} + \alpha c_{kl} + c_{lj}$ which shows that z is at least as large as the maximum of the transportation costs between all origin-destination pairs. If $l \neq n$, then $V_{lj}^{i\cdot} = 0$ and the right-hand side of (22) is equal to $c_{ik} + \alpha c_{kl} + c_{nj} - \alpha C_{\max}$ and thus $c_{ik} + \alpha c_{kl} + c_{nj} - \alpha C_{\max} \leq c_{ik} + c_{nj} \leq c_{ik} + \alpha c_{kn} + c_{nj}$. This shows that the minimum value of z satisfying (22) does not exceed the maximum of the transportation costs between all origin-destination pairs.

Constraints (18) and (19) can be replaced by

$$\sum_{i=1}^{N} \sum_{j=1}^{N} U_{\cdot j}^{ik} \leq N^2 Z_k, \qquad k = 1, \ldots, N,$$

and

$$\sum_{i=1}^{N} \sum_{j=1}^{N} V_{lj}^{i\cdot} \leq N^2 Z_l, \qquad l = 1, \ldots, N,$$

respectively. This results in a smaller number of constraints, but - as our experiments showed - not in smaller solution times.

In a situation where a hub set is already given in UMApHCP, we only need to allocate all nodes to hubs for all origin-destination pairs. This subproblem is called the hub center multiple allocation problem (HCMAP). Unlike the analogous problem HCSAP in the single allocation case, HCMAP can be solved in polynomial time by solving N^2 shortest path problems in the hub network, one for each origin-destination pair.

4 Computational complexity

4.1 Complexity of p-Hub center problems

In the last two sections, we have formulated USApHCP and UMApHCP as (mixed) integer programs. In this section we show that both, USApHCP and UMApHCP are NP-hard. The first result is known from Kara and Tansel [18] who use a transformation from the dominating set problem. We present in the following a very simple proof which also shows the relation between hub center and general (non-hub) other problems and which can be generalized to a second version of hub problems.

Proposition 4.1 *USApHCP and UMApHCP are NP-hard, even if $\alpha = 0$.*

Proof: The optimal objective value of USApHCP for $\alpha = 0$ is twice the distance of the furthest node from any hub (representing the path from that node to itself via a hub). Hence

USApHCP is equivalent to the vertex p-center network problem which is known to be NP-hard [9].

In order to prove the NP-hardness of UMApHCP we show that for $\alpha = 0$ UMApHCP and USApHCP are equivalent in the sense that any optimal solution for USApHCP implies an optimal solution for UMApHCP and vice versa. Clearly any feasible solution for USApHCP is also feasible for UMApHCP and has the same cost. Now suppose $\mathcal{H}$ is a feasible set of hubs for UMApHCP. Because $\alpha = 0$, the transportation cost from node i to j is given by $c_{ik} + c_{lj}$ for some hubs $k, l \in \mathcal{H}$. For the optimal solution k and l must be the closest hubs to i and j respectively as otherwise a cheaper allocation would exist. Since this holds for any i and j there exists an optimal solution in which all non-hub nodes are allocated to the closest hub only. Hence (at least one of) the optimal solution(s) of UMApHCP is feasible for USApHCP. And this optimal solution is also optimal for the corresponding USApHCP problem as UMApHCP is a relaxation of USApHCP. ■

In some applications, like passenger transportation hubs, it may be of interest to consider alternative hub center models where the travel time (cost) of going from a node to itself is always zero, rather than requiring a round trip via a hub. We denote the corresponding single and multiple allocation variants by USApHCP' and UMApHCP', respectively. For the problem considered in this paper, the path length between nodes i and j is defined by $d_{ij} = c_{ia(i)} + \alpha c_{a(i)a(j)} + c_{a(j)j}$, where $a(i)$ and $a(j)$ are hub nodes for i and j respectively. However, for this new variant, the path length $d_{ii} = 0$ although the formula for calculating d_{ij} remains unchanged when $i \neq j$. While we do not treat this problem variant subsequently, we can use the previous result to establish that also these problem variants are NP-hard.

Proposition 4.2 *The modified hub center variants, USApHCP' and UMApHCP', are NP-hard, even if* $\alpha = 0$.

Proof: We show that any polynomial time algorithm for USApHCP' can be used to solve the corresponding USApHCP in polynomial time. Consider HCP on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with costs c_{ij}. We construct a new problem graph $\mathcal{G}' = (\mathcal{N}', \mathcal{E}')$ by duplicating each node and adding identical length arcs so that for $i \neq j$

$$c_{ij} = c_{i'j'} = c_{i'j} = c_{ij'},$$

where i' and j' are the duplicates of nodes i and j in the original graph respectively. Furthermore we set $c_{ii'} = 0$ for all nodes i and their duplicates i' in $\mathcal{G}'$. There exists at least one optimal solution to USApHCP' in $\mathcal{G}'$ where the i and i' have the same allocation as they have identical distances to all other nodes. Also the optimal solution will only ever have one of k and k' selected as hub (as $c_{kk'} = 0$ and $p \leq |\mathcal{N}'|/2$). Hence the optimal solution of USApHCP' corresponds to a feasible solution of USApHCP in graph $\mathcal{G}$. However, the set of paths between all pairs of non-identical nodes on $\mathcal{G}'$ includes the path between i and i' corresponding to a non-zero length path from a non-hub node to itself. Hence the cost of the optimal solution of USApHCP' in $\mathcal{G}'$ is identical to the corresponding solution of USApHCP in $\mathcal{G}$. Furthermore any feasible solution for the original problem can be mapped to a corresponding feasible solution of USApHCP' in $\mathcal{G}'$. This shows that an optimal solution of USApHCP' yields an optimal solution of the original problem.

Obviously we can replace in all previous arguments "single" by "multiple" allocation such that the NP-hardness of UMApHCP' follows in the same way. ∎

4.2 Complexity of the single allocation problem

As a special case we now consider hub center problems for which the location of the hubs is fixed, that is only the allocation of non-hub nodes to hubs has to be decided. These allocation problems with respect to given hub sets are denoted HCMAP and HCSAP for the multiple and single allocation case, respectively.

HCMAP can easily be solved in polynomial time by finding for each pair i, j of origin-destination nodes a shortest path in the hub network. In contrast, HCSAP turns out to be NP-hard. In order to prove the NP-hardness of HCSAP, we first define the concepts of *independent sets*, *independent transversals* and the *3-colouring problem*.

Definition 4.1 *Given a graph $G = (V, E)$, a set $I \subseteq V$ is called an* independent set *(or* stable set*), if and only if there are no arcs $(i, j) \in E$ such that $i \in V$ and $j \in V$.*

Definition 4.2 *Given a graph $G_1 = (V_1, E_1)$ and subsets $W_1, \ldots, W_k$ of V_1 (these sets are not necessarily pairwise disjoint), the* independent transversal problem *can be defined as finding an independent set $I \subseteq V_1$ that intersects every set W_i for $i = 1, \ldots, k$.*

Definition 4.3 *Given a graph $G_2 = (V_2, E_2)$, the* 3-colouring problem *is to partition V_2 into three independent sets (one for each colour).*

The 3-colouring problem is known to be NP-hard (see, for instance, [8]). We now prove that HCSAP is NP-hard, see the appendix for a proof.

Proposition 4.3 *HCSAP is NP-hard.*

Corollary 4.1 *HCSAP does not possess a polynomial time approximation algorithm with worst case guarantee strictly better than $\frac{4}{3}$.*

Proof. Referring to the previous proof, such an approximation algorithm could distinguish between max cost at most 3 and max cost at least 4. ∎

5 A branch and bound approach for UMApHCP

There is some similarity between the uncapacitated multiple allocation p-median problem (UMApHMP) and UMApHCP. Both are NP-hard problems, and their allocation problems can be solved in polynomial time by solving a series of the shortest path problems (see the last paragraph of Section 3). Ernst and Krishnamoorthy [13] propose an efficient branch and bound algorithm for solving UMApHMP by implicitly exploring all possible hub combinations in the node set $\mathcal{N}$. When the number of hubs is small, the method only needs to solve a small number of allocation problems. In this section, we give an outline of a similar branch and bound algorithm for solving UMApHCP. More details can be found in [13].

5.1 Regions and scenarios

A subset of the node set $\mathcal{N}$ is called a *region* or cluster. Let $\{C_1, \ldots, C_r\}$ be a partition of $\mathcal{N}$ such that C_i is a region for all i. Let h_i be the number of hubs contained in region C_i. Note that h_i can be zero. h_i is called the *hub-content* of region C_i. Then $\sum_{i=1}^{r} h_i = p$ must hold for partition $\{C_1, \ldots, C_r\}$.

Define $S = \{(C_i, h_i), i = 1, \ldots, r\}$ to be a scenario for a given partition $\{C_1, \ldots, C_r\}$ and the corresponding hub-contents $\{h_1, \ldots, h_r\}$. Scenarios provide a way of indicating the approximate position of hubs. For a given scenario the position of hubs is limited to the regions with positive hub content without specifying exactly which nodes are to be hubs, though of course if for all regions either $h_i = |C_i|$ or $h_i = 0$ then we have exactly specified the hub locations in that scenario.

For any scenario S a lower bound can be calculated on the UMApHCP cost of any solution that matches the scenario (ie has as many hubs in each region as the hub-content specifies). The lower bound is given by

$$\max_{i \in N, j \in N} \min_{\substack{k \in C_s : h_s \geq 1 \\ l \in C_t : h_t \geq 2 \text{or } t \neq s}} d_{ik} + \alpha\, d_{kl} + d_{lj} \tag{24}$$

That is for each node pair i, j we calculate the shortest path between i and j assuming that any node in a region with positive hub content can be a hub (except that a path can use only one hub in regions with $h_t = 1$). The longest such path determines the objective bound. Note that by using an all pairs shortest path approach this lower bound can be calculated in $\mathcal{O}(np_S^2)$ operations, where p_S is the total number of nodes in regions with positive hub content.

Based on the shortest path solution there are now three possibilities for scenario S:

(a) a feasible solution to UMApHCP is found ($p_S = p$);

(b) no better solution can be found from this scenario than the best known feasible solution to UMApHCP (ie the lower bound exceeds the upper bound); or

(c) scenario S needs to be further explored.

Below we describe how to branch in order to further narrow down the possible locations of the hubs.

5.2 Branching strategy

When branching we select the region with the most number of nodes per hub (ie selecting region $i = \arg\max_k |C_k|/h_k$). For the region C_i two child regions are created as follows: choose two nodes s and t such that the cost between s and t is the diameter of the parent region C_i. Then two regions C_i^s and C_i^t are formed by assigning each node in the parent region to the closer node among s and t.

New scenarios are now created for all possible ways of allocating the h_i hubs to the two regions C_i^s and C_i^t. This creates $h_i + 1$ child scenarios each with C_i replaced by C_i^s and C_i^t and hub contents $h_i^s = 0, \ldots, h_i$, $h_i^t = h_i - h_i^s$.

Unlike traditional branch and bound methods, the described method does not start with a single root node, but with a set of root nodes. According to the numerical experiences reported in [13], this strategy has significantly improved computational efficiency for UMApHMP. Of course we could simply start with a scenario $S = \{(N, p)\}$ (that is a single region with all of the hub nodes) but clearly this would give a very weak bound. Hence it is better to start at some lower point in this conceptual branch and bound tree, so that we have a branch and bound forest. The root nodes of this forest cover all possible options for location of the hubs.

The root nodes of the branch and bound forest are created as follows. Define the diameter of a region as the maximum distance between any two nodes in the region. Suppose that the number of regions to be generated at the root level is r. The original r regions are generated as follows.

(i) Let each node be a region by itself.

(ii) Combine any two regions such that the diameter of the new combined region is no larger than that obtained by joining any other two current regions.

(iii) If the number of regions is r, terminate. Otherwise, go to (ii).

In our implementation we use $r = N/4$. Another possible value for r is p as suggested in [13]. For the generated r regions, we can construct various scenarios by appropriately assigning hub-contents h_i to each region.

Each node of the branch and bound forest is fully specified by a scenario S. We solve the subproblem associated with scenario S. If a feasible solution to UMApHCP is found, or if no better solution can be found from this scenario than the best known feasible solution to UMApHCP, backtrack of the search tree for S is undertaken. Otherwise, perform further branching in a depth first manner. This process continues until no further backtrack or no further branching is possible and all nodes in the forest are explored.

5.3 Obtaining an upper bound

In order to reduce the size of the branch and bound forest, it is desirable to find a good feasible solution to UMApHCP before starting to explore the forest.

A simple heuristic method for a feasible solution to UMApHMP is implemented in [12]. This method is slightly modified below to find a feasible solution to UMApHCP.

The heuristic method proceeds as follows.

(i) Randomly select a hub set $\mathcal{H}$ containing p nodes.

(ii) For any given hub, optimal solutions to HCMAP can be found using an all pairs shortest path algorithm. If a better solution to UMApHCP is found, record it as the current best solution. Otherwise, go to (iii).

(iii) Generate a different $\mathcal{H}'$ by replacing one member in $\mathcal{H}$ by a node from $\mathcal{N} \setminus \mathcal{H}$. If no such $\mathcal{H}'$ can be generated, terminate. Otherwise, go to (ii).

This heuristic can be repeated several times by selecting a different random hub set $\mathcal{H}$ in (i). In the tests reported in Section 6, the number of repetitions used in the above heuristic was equal to the greatest integer not more than $(N \times p)/20$.

5.4 Alternative approaches for obtaining upper bounds

There are many ways to find approximate solutions to UMApHCP, which can be used to obtain an upper bound for the branch and bound method discussed in this section. It is interesting to investigate their performance guarantee. We start from a random heuristic. As a matter of fact, we can prove the same result for both USApHCP and UMApHCP.

Proposition 5.1 *(i) Let y and y_{opt} be the objective function values of a feasible solution and an optimal solution of USApHCP respectively. Then $y \leq (1 + \frac{2}{\alpha})y_{opt}$. Furthermore, this bound is tight.*

(ii) Let z and z_{opt} be the objective function values of a feasible solution and an optimal solution of UMApHCP respectively. Then $z \leq (1 + \frac{2}{\alpha})z_{opt}$. Furthermore, this bound is tight.

Proof. (i) Let $i \to k \to m \to j$ be a transportation path for the origin-destination pair (i, j) in a feasible solution for USApHCP. Then by analysing different scenarios on how a transportation path could be formed in the network, it can be shown that the transportation cost on this path must be equal to one of the following:

$$2c_{ik}, \quad c_{ik} + \alpha c_{km} + c_{mj}, \quad c_{ik} + \alpha c_{km}, \quad \alpha c_{km} + c_{mj}, \quad \alpha c_{km}, \quad c_{mj}, \quad c_{ik} + c_{kj}, \quad c_{ik}.$$

For any nodes or hubs i and j, let $i \to k^* \to m^* \to j$ be the transportation path between i and j in an optimal solution for USApHCP. It is obvious that the transportation cost on this path is not greater than y_{opt}. Then it follows from the triangle inequality property that

$$c_{ij} \leq c_{ik^*} + c_{k^*m^*} + c_{m^*j} \leq \frac{1}{\alpha} y_{opt}.$$

It is easy to see that $(1 + \frac{2}{\alpha})y_{opt}$ is an upper bound for any of the following numbers

$$2c_{ik}, \quad c_{ik} + \alpha c_{km} + c_{mj}, \quad c_{ik} + \alpha c_{km}, \quad \alpha c_{km} + c_{mj}, \quad \alpha c_{km}, \quad c_{mj}, \quad c_{ik} + c_{kj}, \quad c_{ik}.$$

Therefore, the desired inequality holds.

The network example displayed in Figure 1 shows that the above bound is tight. The network comprises eight nodes and 4 hubs are to be located. The network is fully connected although some connections are omitted in the graph. The lengths of some edges are shown in the graph where A and B are constants, ε and δ are positive parameters, and ε, δ are sufficiently small. A feasible solution is shown in Figure 1(a). Assume that four hubs are allocated to the nodes $1, 2, 3, 4$. Further assume that 5 is allocated to 3, 6 to 4, 7 to 2 and 8 to 1. Then the objective function value of this 4-hub center problem is the total cost between 7 and 8 which is equal to

$$y := 2(A + \varepsilon) + \sqrt{B^2 + \delta^2} + \alpha(2A + B) + 2(A + \varepsilon) + \sqrt{B^2 + \delta^2}.$$

On the other hand, in an optimal solution as shown in Figure 1(b), the four hubs are located at the nodes $3, 4, 5, 6$, and 1 is allocated to 3, 2 to 4, 7 to 5, and 8 to 6. The optimal objective function value is

$$y_{opt} := A + \varepsilon + \alpha\sqrt{B^2 + \delta^2} + A + \varepsilon.$$

It is easy to see that when $\varepsilon \to 0$, $A \to 0$ and $\delta \to 0$, then

$$\frac{y}{y_{opt}} \to 1 + \frac{2}{\alpha}.$$

Therefore the upper bound is tight.

(ii) The bound can be established in a similar way to (i) though the feature of multiple allocations must be taken into account. We omit details. Let us consider USApHCP for the same network. It was proved for (i) that for any positive ε, we can find a counterexample of USApHCP such that

$$y \geq (1 + \frac{2}{\alpha})y_{opt} - \varepsilon$$

where y and y_{opt} are the objectives of a feasible solution and an optimal solution to this counterexample. Let z_{opt} be the optimal objective of the same counterexample in the version of UMApHCP. It is trivial to observe that the optimal value for UMApHCP provides a lower bound for the optimal value for USApHCP, i.e., $y_{opt} \geq z_{opt}$. This shows that

$$y \geq (1 + \frac{2}{\alpha})z_{opt} - \varepsilon$$

Note that y is also a feasible solution for the same counterexample in the version of UMApHCP. This implies that the above bound is also tight for UMApHCP. ∎

The worst bounds presented in Proposition 5.1 can be improved if some smarter searches are introduced. The following is such an example.

Heuristic 1

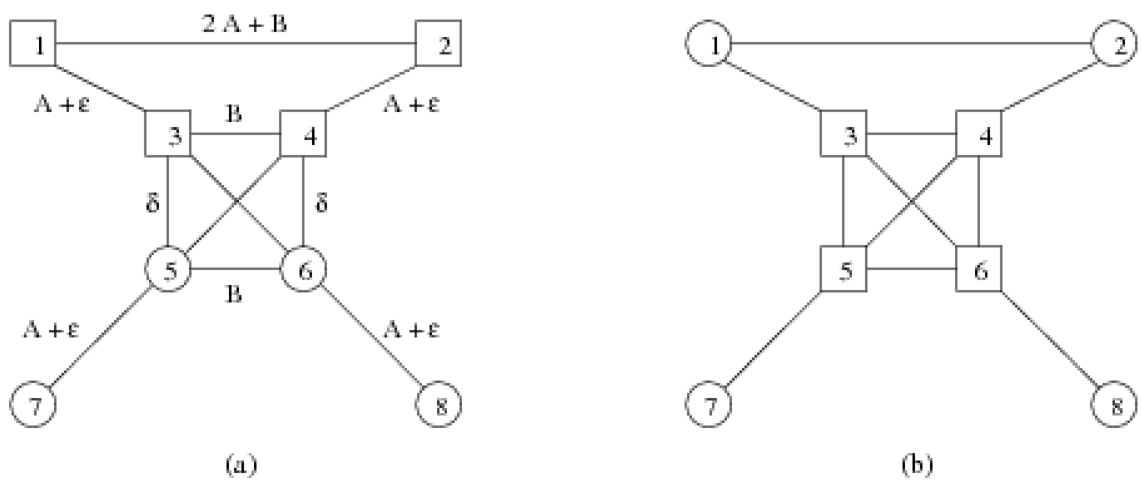

Figure 1: A feasible (a) and optimal (b) hub locations for a network example that demonstrate the USApHCP bound in Proposition 5.1 is tight.

Step 1. Solve UMApHCP optimally with $p = 1$.

Step 2. Assume there is a feasible solution for UMApHCP with q hubs ($q < p$).

- Select the longest path in the existing solution. Choose a non-hub node, say k, in this path and insert it into the existing solution.
- Solve HCMAP optimally using the given $q + 1$ hubs.

Step 3. Go to Step 2 if $q < p$. Otherwise, terminate.

Below we state a performance guarantee result for Heuristic 1 without a proof because the proof is relatively easy and similar to that for Proposition 5.1.

Proposition 5.2 *Suppose z and z_{opt} are the objective function values of the solution obtained from Heuristic 1 and the optimal objective of UMApHCP respectively. Then $z \leq \frac{2}{\alpha} z_{opt}$.*

We remark that a method similar to Heuristic 1 for USApHCP can be proposed and a similar result to Proposition 5.2 can be obtained. The detail is omitted.

6 Numerical experiments

We tested our algorithms for both USApHCP and UMApHCP with the CAB data set [14, 23] and with the AP data set [12]. The CAB data set is generated from the Civil Aeronautics Board Survey of 1970 passenger data in the United States. The AP data set is derived from the real-world application of a postal delivery network.

In test problem $a.b.c$ of CAB, there are a nodes and b hubs, and c represents the economic discount factor for the cost of transfer of flow between hub nodes ranging from 0.2 to 1.0. In test problem $a.b$ of AP, there are a nodes and b hubs. The economic discount factor for the cost of flow transfer between hub nodes was always 0.75 for all problems in the AP data set (and we ignore the non-unit collection and distribution cost factors traditionally used for the AP data).

All of the numerical experiments were carried out on a ***DEC Alpha machine*** and algorithms were coded in C/C++. All integer programs were solved using Ilog CPLEX Version 9.1[1] using the default parameter settings except that the optimality gap tolerance was set to $2 \times 10^{-6}\%$.

We tested the formulation of Kara and Tansel and the new radius formulation for USApHCP presented in (1)–(7). Numerical results for USApHCP are shown in Tables 1 and 2. For each test problem, we report:

Prob for the problem name;

Obj for the optimal objective;

Gap the gap between the relaxed LP solution and the optimal solution (except for the heuristic gap which is the percentage difference between the heuristic upper bound and the optimum);

Nodes the number of branch and bound nodes; and

CPU the CPU time in seconds required for solving the formulation with CPLEX.

Prob	Optimal Obj.	Heur. Gap	Radius Formulation Gap	Radius Formulation Nodes	Radius Formulation CPU	KT Formulation Gap	KT Formulation Nodes	KT Formulation CPU
10.2.2	1425.58	31.47%	75.24%	24	0.34	29.96%	7	0.56
10.2.4	1627.52	35.74%	56.63%	28	0.12	29.92%	17	0.75
10.2.6	1759.13	41.20%	39.81%	22	0.09	21.97%	13	0.62
10.2.8	1759.13	43.51%	19.74%	0	0.06	9.68%	3	0.33
10.2.1	1839.65	54.66%	4.07%	5	0.06	2.87%	0	0.12
10.3.2	1119.54	62.11%	68.47%	0	0.29	36.13%	14	0.56
10.3.4	1185.07	47.59%	40.43%	1	0.07	24.27%	27	0.58
10.3.6	1387.00	50.20%	23.66%	6	0.08	16.19%	24	0.54
10.3.8	1588.94	50.23%	11.15%	9	0.04	8.02%	17	0.46
10.3.1	1790.55	44.09%	1.44%	9	0.04	1.43%	7	0.39
10.4.2	830.25	67.47%	57.49%	20	0.09	32.75%	13	0.51
10.4.4	968.20	49.75%	27.09%	0	0.04	20.15%	3	0.44
10.4.6	1146.19	60.22%	7.62%	10	0.03	7.52%	0	0.33
10.4.8	1454.44	52.27%	2.93%	16	0.04	2.93%	2	0.24
10.4.1	1764.79	40.40%	0.00%	0	0.01	0.00%	0	0.08
15.2.2	2005.02	49.70%	74.06%	25	0.77	28.23%	41	4.92
15.2.4	2160.75	38.91%	51.87%	20	0.35	26.71%	57	5.54
15.2.6	2214.09	35.57%	29.54%	18	0.32	13.89%	70	5.19
15.2.8	2423.80	23.84%	14.18%	23	0.29	6.80%	11	3.60

[1]Brief testing of the formulation on a Linux machine with a 3GHz Intel Xeon processor and CPLEX 11.0 indicated that very little improvements in run time could be achieved with the newer version of the solver though it seemed to generate fewer branch and bound nodes and more cuts.

Prob	Optimal	Heur.	Radius Formulation			KT Formulation		
15.2.1	2609.18	16.18%	0.35%	17	0.21	0.05%	0	0.71
15.3.2	1749.04	28.62%	70.27%	16	0.40	37.89%	134	8.19
15.3.4	1760.15	27.81%	40.91%	33	0.35	25.49%	54	4.46
15.3.6	1844.92	21.93%	15.44%	20	0.22	10.32%	8	5.07
15.3.8	2166.54	8.43%	3.99%	11	0.21	3.17%	2	2.23
15.3.1	2600.08	2.13%	0.00%	1	0.11	0.00%	0	0.71
15.4.2	1340.96	56.03%	61.22%	76	0.43	35.54%	64	4.28
15.4.4	1434.38	45.86%	27.49%	38	0.25	19.64%	166	5.88
15.4.6	1754.51	19.25%	11.08%	36	0.16	10.50%	25	3.53
15.4.8	2080.06	9.16%	0.00%	17	0.17	0.00%	0	0.80
15.4.1	2166.54	8.43%	3.99%	11	0.21	3.17%	2	2.23
20.2.2	1892.99	26.35%	72.53%	38	0.94	24.00%	77	30.23
20.2.4	2160.75	20.47%	51.87%	42	1.29	26.76%	162	47.62
20.2.6	2274.67	29.04%	31.42%	44	0.81	15.80%	131	26.48
20.2.8	2501.93	34.31%	16.86%	18	0.65	9.27%	121	38.93
20.2.1	2609.18	45.08%	0.35%	16	0.48	0.05%	0	5.84
20.3.2	1551.25	38.60%	66.48%	113	1.11	31.45%	209	30.64
20.3.4	1760.15	27.84%	40.91%	153	1.10	25.50%	195	39.21
20.3.6	1997.79	35.38%	21.91%	66	0.85	15.34%	186	32.11
20.3.8	2263.54	29.48%	8.11%	73	0.74	6.04%	23	16.81
20.3.1	2600.08	30.12%	0.00%	6	0.32	0.00%	0	5.88
20.4.2	1355.41	49.79%	61.63%	124	0.99	36.53%	379	39.58
20.4.4	1472.71	44.58%	29.38%	165	1.09	19.88%	143	32.47
20.4.6	1834.83	30.23%	14.98%	89	0.75	12.33%	60	23.94
20.4.8	2153.00	27.75%	3.39%	162	0.79	3.27%	43	15.68
20.4.1	2600.08	16.12%	0.00%	4	0.24	0.00%	15	35.69
25.2.2	2131.20	24.23%	74.42%	23	2.64	28.78%	176	151.43
25.2.4	2402.55	30.80%	54.62%	45	1.88	26.66%	549	236.51
25.2.6	2558.74	61.09%	36.08%	141	2.35	17.58%	146	137.67
25.2.8	2714.93	71.05%	19.68%	70	1.68	9.74%	131	85.97
25.2.1	2827.16	68.09%	3.59%	42	1.52	2.60%	10	62.56
25.3.2	1923.12	11.49%	71.65%	94	2.08	36.59%	706	219.86
25.3.4	2100.47	22.83%	48.09%	94	2.77	28.25%	2050	705.10
25.3.6	2340.25	55.05%	30.12%	377	3.23	18.84%	248	292.12
25.3.8	2554.13	79.22%	14.62%	239	2.29	8.93%	260	116.30
25.3.1	2758.39	71.08%	1.18%	56	1.58	0.95%	13	216.50
25.4.2	1619.48	9.48%	66.34%	36	1.49	38.39%	2929	446.36
25.4.4	1884.84	26.60%	42.16%	204	2.59	27.80%	516	296.83
25.4.6	2182.49	31.73%	25.07%	214	3.03	17.72%	332	235.23
25.4.8	2454.35	49.32%	11.15%	195	2.54	7.96%	335	172.09
25.4.1	2726.28	51.22%	0.02%	24	1.32	0.02%	231	266.98

Prob	Optimal	Heur.	Radius Formulation	KT Formulation

Table 1: Numerical results of USApHCP for CAB problems.

The heuristic gaps are generally very large, however they take almost no time to compute (no more than 0.01 seconds for all of the CAB problems), so we have passed them as initial upper bounds to CPLEX. Generally these do not have a significant impact on the solution times of the MILPs. From the results in Table 1 one can see that the new formulation is not necessarily tighter than that of Kara and Tansel. In fact for most of the problems it starts with a larger gap. However the LP relaxations are much faster to solve and on average it requires less nodes than the Kara and Tansel formulation leading to improvements of an order of magnitude for the small problems and up to two orders of magnitude for the larger problems.

The performance of both formulations for the AP problems (see Table 2) was similar to that for the CAB problems. Clearly, the new formulation is superior to Kara and Tansel's formulation. Kara and Tansel's formulation failed to solve all large problems with $n \geq 40$ within 1000 seconds CPU time - in fact for many of these larger problems CPLEX could not even solve the root node within 1000 seconds. Evidently it becomes much more difficult to find optimal solutions using Kara and Tansel's formulation when the size of test problems becomes large because the number of constraints in Kara and Tansel's formulation grows rapidly. Optimal solutions were found by the new formulation within the CPU time limit (1000 seconds) for all AP problems except some of the large 100 node AP problems. In order to test the formulations more rigorously on these large problems we tried solving the 100 node instances without CPU time limit but found that the KT Formulation ran out of memory before solving them, while our formulation took a very large amount of CPU time particularly for 100.3 as reported in Table 2.

Prob.	Optimal Obj.	Radius Formulation			KT Formulation		
		Gap	Nodes	CPU	Gap	Nodes	CPU
10.2	40382.7	28.96%	0	0.07	14.31%	2	0.50
10.3	34772.4	17.50%	1	0.04	9.97%	2	0.51
10.4	32574.2	11.94%	0	0.04	9.35%	14	0.61
10.5	32531.2	11.82%	40	0.06	11.82%	18	0.65
20.2	45954.2	17.60%	16	0.44	8.07%	41	21.91
20.3	43400.4	12.75%	58	0.68	10.52%	21	24.98
20.4	38607.3	1.91%	55	0.71	1.91%	96	40.46
20.5	37868.1	0.00%	7	0.39	0.00%	170	34.13
20.10	37868.1	0.00%	0	0.08	0.00%	0	4.98
25.2	53207.5	14.39%	2	0.96	6.97%	49	96.62
25.3	46608.3	2.27%	93	1.58	2.27%	2	38.37
25.4	45552.5	0.00%	180	1.96	0.00%	0	44.35
25.5	45552.5	0.00%	0	0.24	0.00%	70	108.65
25.10	37868.1	0.00%	7	0.39	0.00%	170	34.13

Prob.	Optimal	Radius Formulation			KT Formulation		
40.2	61682.5	18.36%	16	8.93	11.14%	*	>1000.00
40.3	58192.8	14.52%	118	14.05	18.46%	32	1663.36
40.4	52265.3	4.83%	61	9.20	22.92%	*	>1000.00
40.5	49741.2	0.00%	26	10.35	*	*	>1000.00
40.10	49741.2	0.00%	0	0.44	0.00%	20	1822.11
50.2	65523.4	22.61%	97	52.71	*	*	>1000.00
50.3	60132.1	15.67%	775	98.91	*	*	>1000.00
50.4	52905.8	4.15%	210	48.48	*	*	>1000.00
50.5	50707.9	0.00%	79	34.90	*	*	>1000.00
50.10	50707.9	0.00%	38	3.33	2.22%	*	>1000.00
100.2	65914.8	21.32%	945	610.67	***	***	***
100.3	60658.9	14.51%	32840	15672.00	***	***	***
100.5	54243.5	4.39%	979	1259.79	***	***	***
100.10	51860.0	0.00%	33	68.36	***	***	***

Table 2: Numerical results of USApHCP for AP problems. *** Indicates that the forumlation could not be solved with CPLEX due to insufficient memory.

We tested the 3- and 4-index formulations for UMApHCP. Initial experiments showed that both formulations are inefficient even for small problems. Numerical results for UMApHCP for a few small CAB test problems are shown in Table 3 with the same performance measures as for the other formulations.

The three index formulation is clearly not useful for solving even small UMApHMP problems. However the four index formulation, due to its greater tightness can at least solve small problems in a reasonable amount of time, though it is clearly not competitive with our branch and bound method. For larger CAB problems with 15 nodes the four index formulation still solves but run times increase to over 100 seconds on average.

Heuristic 1 did not perform well in terms of solution quality although its CPU time requirement is negligible. On average, the Heuristic 1 upper bound is about 30% larger than the optimal objective value. The corresponding heuristic for USApHCP also performed poorly. Therefore, we do not publish numerical performance of these heuristics.

We tested the shortest path based branch and bound method for UMApHCP. Numerical results for UMApHCP are shown in Tables 4 and 5. For each test problem, we report the heuristic gap $\frac{\text{Obj}'-\text{Obj}}{\text{Obj}} \times 100.0$ where Obj' represents the objective of the solution obtained from the simple euristic described in Section 5.3. Also the **Total CPU** is the runtime in seconds for both the heuristic and the branch and bound.

Problem	Optimal Obj.	Heur. Gap	Heur. CPU	Branch & Bound Nodes	Branch & Bound Total CPU
10.2.2	1421.88	0.00%	0.00	9	0.00

Problem	Obj.	Gap	CPU	Nodes	Total CPU
10.2.4	1548.37	0.00%	0.00	9	0.00
10.2.6	1749.04	0.13%	0.00	21	0.00
10.2.8	1749.04	0.00%	0.00	11	0.00
10.2.1	1764.79	0.00%	0.00	9	0.00
10.3.2	1119.54	0.00%	0.00	19	0.00
10.3.4	1181.37	0.00%	0.00	22	0.00
10.3.6	1308.85	0.00%	0.00	29	0.00
10.3.8	1502.14	4.60%	0.00	35	0.00
10.3.1	1764.79	0.00%	0.00	13	0.00
10.4.2	809.36	37.41%	0.00	23	0.00
10.4.4	968.20	0.00%	0.00	23	0.00
10.4.6	1146.19	0.00%	0.00	28	0.00
10.4.8	1411.83	3.02%	0.00	37	0.00
10.4.1	1764.79	0.00%	0.00	13	0.00
15.2.2	2005.02	0.00%	0.00	15	0.00
15.2.4	2027.69	0.00%	0.00	18	0.00
15.2.6	2081.04	0.00%	0.00	13	0.00
15.2.8	2335.82	0.00%	0.00	12	0.00
15.2.1	2600.08	0.00%	0.00	8	0.00
15.3.2	1716.14	1.29%	0.00	116	0.00
15.3.4	1738.32	0.00%	0.00	26	0.00
15.3.6	1823.10	0.00%	0.00	63	0.00
15.3.8	2141.83	0.00%	0.00	19	0.00
15.3.1	2600.08	0.00%	0.00	11	0.00
15.4.2	1287.78	0.00%	0.00	59	0.00
15.4.4	1395.88	0.00%	0.00	47	0.00
15.4.6	1751.45	0.00%	0.00	43	0.01
15.4.8	2080.06	0.00%	0.00	12	0.00
15.4.1	2600.08	0.00%	0.00	12	0.00
20.2.2	1892.99	0.00%	0.00	20	0.00
20.2.4	2027.69	0.00%	0.00	21	0.00
20.2.6	2248.13	0.00%	0.00	27	0.00
20.2.8	2335.99	8.32%	0.00	65	0.00
20.2.1	2600.08	0.00%	0.00	17	0.00
20.3.2	1551.25	10.65%	0.00	116	0.01
20.3.4	1738.32	0.00%	0.00	40	0.01
20.3.6	1916.16	0.00%	0.00	225	0.01
20.3.8	2195.22	0.00%	0.00	159	0.01
20.3.1	2600.08	0.00%	0.00	14	0.01
20.4.2	1287.78	0.00%	0.01	105	0.01
20.4.4	1472.71	0.00%	0.01	230	0.02

Problem	Obj.	Gap	CPU	Nodes	Total CPU
20.4.6	1808.70	0.80%	0.01	1906	0.06
20.4.8	2128.11	1.17%	0.01	291	0.02
20.4.1	2600.08	0.00%	0.01	17	0.01
25.2.2	2049.48	0.00%	0.00	26	0.01
25.2.4	2402.55	0.00%	0.00	52	0.01
25.2.6	2558.74	0.00%	0.00	51	0.01
25.2.8	2714.93	0.00%	0.00	75	0.01
25.2.1	2739.22	0.00%	0.00	54	0.01
25.3.2	1911.60	0.17%	0.01	217	0.02
25.3.4	2064.67	1.28%	0.01	174	0.02
25.3.6	2243.77	0.00%	0.01	181	0.02
25.3.8	2515.58	0.00%	0.01	184	0.02
25.3.1	2725.79	0.00%	0.01	17	0.01
25.4.2	1619.48	7.05%	0.02	446	0.04
25.4.4	1774.45	0.00%	0.02	523	0.05
25.4.6	2127.13	0.00%	0.02	1204	0.08
25.4.8	2437.71	0.37%	0.02	1267	0.08
25.4.1	2725.79	0.00%	0.02	23	0.02

Table 4: Numerical results of UMApHCP for CAB problems.

	Optimal	Heur.		Branch & Bound	
Problem	Obj.	Gap	CPU	Nodes	Total CPU
10.2	39922.11	1.15%	0.00	32	0.00
10.3	32713.94	0.00%	0.00	52	0.00
10.4	31577.96	0.00%	0.00	79	0.00
10.5	30371.32	0.00%	0.00	100	0.00
20.2	45954.15	0.00%	0.00	16	0.00
20.3	40909.59	6.09%	0.00	51	0.01
20.4	38320.25	0.00%	0.01	105	0.01
20.5	37868.15	0.00%	0.01	12	0.01
20.10	37868.15	0.00%	0.05	67	0.05
25.2	51533.30	0.00%	0.00	15	0.00
25.3	45552.50	8.67%	0.01	24	0.01
25.4	45552.50	0.00%	0.02	31	0.02

		3 index forumlation			4 index formulation		
Prob.	Obj.	Gap	Nodes	CPU	Gap	Nodes	CPU
10.2.2	1421.88	35.55	114	5.22	26.39	12	6.45
10.2.4	1548.37	39.14	488	12.71	22.26	10	3.77
10.2.6	1749.04	44.29	7224	110.43	21.17	21	4.13
10.2.8	1749.04	45.45	144348	921.38	12.38	18	3.54
10.2.1	1764.79	47.96	12110	314.95	0.00	12	16.96
10.3.2	1119.54	40.59	3304	65.87	32.31	19	3.88
10.3.4	1181.37	34.36	770	21.58	19.97	20	4.91
10.3.6	1308.85	35.45	2070	36.25	11.35	20	3.30
10.3.8	1502.14	40.88	2292	62.50	6.01	13	5.99
10.3.1	1764.79	48.29	263	26.65	0.00	0	4.41
10.4.2	809.36	39.19	144	13.85	30.56	13	2.23
10.4.4	968.20	28.64	2475	45.05	19.47	19	2.49
10.4.6	1146.19	27.03	1835	35.64	7.62	4	3.63
10.4.8	1411.83	37.04	1990	37.31	0.00	5	4.08
10.4.1	1764.79	48.29	1646	69.28	0.00	0	0.88

Table 3: Numerical results of UMApHCP MILP formulations for a few small CAB problems

Problem	Obj.	Gap	CPU	Nodes	Total CPU
25.5	45552.50	0.00%	0.03	40	0.03
25.10	45552.50	0.00%	0.13	193	0.14
40.2	61140.80	0.00%	0.02	45	0.03
40.3	56309.88	0.37%	0.05	186	0.08
40.4	51279.14	5.06%	0.11	170	0.14
40.5	49741.20	0.00%	0.17	12	0.18
40.10	49741.20	0.00%	0.98	51	1.00
50.2	61179.03	0.00%	0.04	46	0.05
50.3	56729.94	0.00%	0.11	185	0.16
50.4	52905.77	0.00%	0.20	327	0.28
50.5	50707.87	0.00%	0.50	26	0.52
50.10	50707.87	0.00%	2.15	27	2.17
100.2	63197.10	0.95%	0.44	66	0.56
100.3	57925.66	0.00%	1.48	99	1.67
100.5	53949.33	0.76%	5.91	17131	22.52
100.10	51860.03	0.00%	38.07	213	39.04

Problem	Obj.	Gap	CPU	Nodes	Total CPU
200.3	62945.55	0.00%	25.77	1252	35.37

Table 5: Numerical results of UMApHCP for AP problems.

Even though the heuristic described in Section 5.3 does not have any theoretical performance guarantee it performed much better (in terms of solution quality achieved in small amounts of CPU time) than the heuristic method for UMApHCP described in Section 5.4. From Tables 4 and 5 we can see that optimal solutions of all test problems are found. For all but a few large problems with larger p, the CPU time was less than one second, and the number of branch and bound nodes explored is not large at all. It is notable however that the CPU times for AP50.10, AP100.5, AP100.10 and AP200.3 are very large compared with other test problems and clearly grow exponentially with problem size. This is mostly due to large values of p. We remark that in most practical applications, the benefits of hub network configurations are largest when p is small.

Compared with the shortest path based branch and bound method, performance of both the three and four-index formulations (see Table 3 above) was poor. Even when we used in CPLEX the upper bound derived from the heuristic method (see 5.3), the CPU times for both the three and four-index formulations were improved by at most half. In conclusion, the three and four-index formulations are not competitive.

As proved theoretically in Section 4, for the same test problem, the computational results show that the optimal objective value for UMApHCP is no greater than that for USApHCP. It is interesting to note that these two values do coincide for quite a few test problems (for example, Problem 10.3.2). It may be possible to exploit this feature for developing more efficient algorithms for USApHCP, see [22] for one approach.

7 Conclusions

In this paper we have studied USApHCP and UMApHCP. We have developed a new mixed integer programming formulation for USApHCP and two integer programming formulations for UMApHCP. Both problems are proved to be NP-hard even when the economic discount factor is zero. We also showed that the allocation sub-problem of USApHCP is NP-hard. A shortest path based branch and bound method is proposed similar to that developed in [13] for UMApHMP.

We have carried out numerical experiments using well-known test datasets in the literature for both USApHCP and UMApHCP. The numerical results showed that the new formulation for USApHCP is clearly superior to the best known formulation from Kara and Tansel [18] in terms of computational time by 1-2 orders of magnitude. The numerical experiments demonstrated that the shortest path based branch and bound method is extremely efficient for solving UMApHCP. By contrast the four index formulation only offers viable computational performance for small problems and even then is orders of magnitude slower than our branch and bound method. The three index formulation - while smaller than the four index formulation - performs very badly computationally.

Acknowledgement. The authors are grateful to Mark Horn for carefully proof reading an earlier draft of this paper. We are thankful to two referees for their constructive comments which have helped to improve the presentation of the paper significantly.

References

[1] S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21, 2008.

[2] D. L. Bryan and M. E. O'Kelly. Hub-and-spoke networks in air transportation: An analytical review. *Journal of Regional Science*, 39(2):275–295, 1999.

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

[4] 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.

[5] J. F. Campbell, A. T. Ernst, and M. Krishnamoorthy. Hub arc location problems: Part i - introduction and results. *Management Science*, 51(10):1540–1555, 2005.

[6] L. Cánovas, S. García, and A. Marín. Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. *European Journal of Operational Research*, 179(3):990–1007, 2007.

[7] S.-H. Chung, Y.-S. Myung, and D.-W. Tcha. Optimal design of a distributed network with a two-level hierarchical structure. *European Journal of Operational Research*, 62:105–115, 1992.

[8] P. Crescenzi, V. Kann, M. Halldorsson, M. Karpinski, and G. Woeginger. A compendium of NP optimization problems. http://www.nada.kth.se/ṽiggo/problemlist/compendium.html, 2001.

[9] M. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley and Sons Inc, New York, 1995.

[10] R. S. de Camargo, G. Miranda, and H. P. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research*, 35(4):1047–1064, 2008.

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

[12] 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(1):100–112, 1998.

[13] A. T. Ernst and M. Krishnamoorthy. An exact solution approach based on shortest-paths for p-hub median problems. *INFORMS Journal on Computing*, 10(2):149–162, 1998.

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

[15] H. Hamacher, M. Labbé, S. Nickel, and T. Sonneborn. Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics*, 145(1):104–116, 2004.

[16] H. Hamacher and T. Meyer. Hub coverand hub center problems. Technical Report 98, FB Mathematik, TU Kaiserslautern, 2006.

[17] 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.

[18] B. Y. Kara and B. Tansel. On the single assignment p-hub center problem. *European Journal of Operational Research*, 125(3):648–655, 2000.

[19] B. Y. Kara and B. Tansel. The latest arrival hub location problem. *Management Science*, 47(10):1408–1420, 2001.

[20] A. Marín. Uncapacitated euclidean hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. *Journal of Global Optimization*, 33(3):393–422, 2005.

[21] A. Marín, L. Cánovas, and M. Landete. New formulations for the uncapacitated multiple allocation hub location problem. *European Journal of Operational Research*, 172(1):274–292, 2006.

[22] T. Meyer, A. T. Ernst, and M. Krishnamoorthy. A 2-phase algorithm for solving the single allocation p-hub center problem. *Computers and Operations Research (submitted)*, 2008.

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

[24] M. E. O'Kelly and H. J. Miller. Solution strategies for the single facility minimax hub location problem. *Papers in Regional Science: The Journal of the RSAI*, 70:367–380, 1991.

[25] P. Pamuk and C. Sepil. A solution to the hub center problem via a single-relocation algorithm with tabu search. *IEE Transactions*, 33:399–411, 2001.

[26] H. Podnar, J. Skorin-Kapov, and D. Skorin-Kapov. Network cost minimization using threshold-based discounting. *European Journal of Operational Research*, 137(2):371–386, 2002.

Proof of Proposition 4.3.

The proof will be in two stages. In the first stage, we will show that 3-colouring is polynomially reducable to the independent transversal problem, thus showing that the independent transversal problem is NP-complete. In the second stage we reduce independent transversal to HCSAP thus establishing that HCSAP is NP-hard.

Given an instance (V_2, E_2) of 3-colouring, we construct an instance (V_1, E_1) of independent transversal as follows (see Figure 2 for an example).

- For every vertex $v \in V_2$, create three new vertices $(v, 1), (v, 2), (v, 3) \in V_1$. These three vertices form the subset W_v.
- For every $v \in V_2$, connect $(v, 1), (v, 2), (v, 3)$ by a triangle in E_1.
- If there is an edge $[u, v] \in E_2$, then connect (u, i) and (v, i) by an edge in E_1 for $i = 1, 2, 3$.

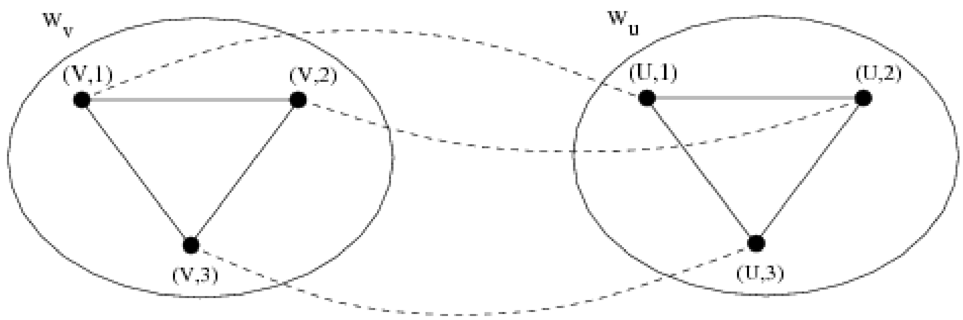

Figure 2: Example illustrating the construction of an independent transversal problem corresponding to a simple graph with two vertices joined by a single edge in $(V_2, E_2) = (\{v, u\}, \{(v, u)\})$.

Since a vertex (v, i) in the independent set for (V_2, E_2) corresponds to colouring the vertex v in V_1 by colour i, the graph (V_2, E_2) is 3-colourable if and only if there exists an independent set in the newly constructed graph that intersects all the sets W_v.

Next we prove that the independent transversal problem polynomially reduces to HCSAP, establishing that the latter problem is, indeed, NP-complete.

Consider an instance (V_1, E_1) with subsets $W_1, \ldots, W_k$ of independent transversal. We now create a corresponding hub center problem (V_3, E_3) as follows (see Figure 3):

- For every v in V_1 create a hub h_v. If u and v are connected by an edge in E_1, then the cost between hubs h_u and h_v is 2. Otherwise, the cost between h_u and h_v is 1. Then, a subset of the hubs is independent if and only if they can reach each other by distances of length 1.
- For every subset W_i create a non-hub f_{W_i}. The cost between non-hubs f_{W_i} and hubs h_v are as follows: If v is contained in W_i, then the cost is 1. If v is not contained in W_i, then the cost is 2.

Since all costs are 1 or 2, they automatically satisfy the triangle inequality. In the rest of the

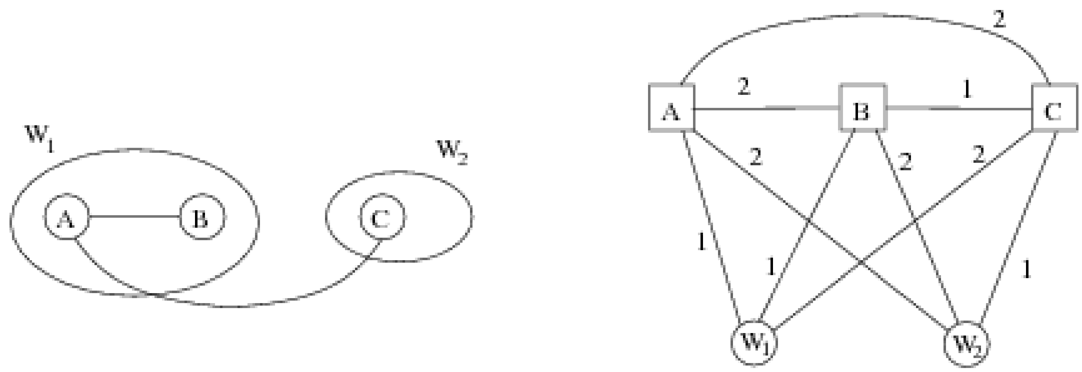

Figure 3: Example of a simple transversal problem and the corresponding hub center network construction from the proof of Proposition 4.3

proof, we show that there exists an assignment of HCSAP for (V_3, E_3) with the optimal value at most 3 if and only if (V_1, E_1) has an independent transversal with respect to $W_1, W_2, \ldots, W_t$.

Let $w_1, w_2, \ldots, w_t$ be an independent transversal for (V_1, E_1) with respect to $W_1, W_2, \ldots, W_t$ such that $w_i \in W_i$. Then $\{w_1, w_2, \ldots, w_t\}$ is an independent set of (V_1, E_1). For every W_i, assign f_{W_i} to hub h_{w_i} with $w_i \in W_i$. Then both the collection and distribution cost between a non-hub node and a hub node is 1. Because $\{w_1, w_2, \ldots, w_t\}$ is an independent set of (V_1, E_1), the transportation cost between non-hub nodes f_{W_i} and f_{W_j} is 3 (or just 2 in the special case where all the W_i intersect in a single node). Note that the cost of any other transportation path is not more than 3. Hence we have obtained a solution for HCSAP with the optimal value at most 3.

Conversely, suppose HCSAP of (V_3, E_3) has a solution with a cost of at most 3. Then every non-hub node f_{W_i} must be assigned to a hub node h_v such that $v \in W_i$. Otherwise, the transportation cost from f_{W_i} would be 4, which is greater than 3. Let w_i be the hub to which f_{W_i} is assigned. Then $\{w_1, w_2, \ldots, w_t\}$ is an independent transversal of (V_1, E_1) with respect to $W_1, W_2, \ldots, W_t$. Firstly, $w_i \in W_i$ by the construction of (V_3, E_3). Secondly, for any i, j with $w_i \neq w_j$, w_i and w_j are not connected in (V_3, E_3). Otherwise, the transportation cost of HCSAP between f_{W_i} and f_{W_j} would be $1 + 2 + 1 = 4 > 3$, a contradiction. ■