Working Paper Series

1/2011

Reformulations and Computational Results for the Uncapacitated Single Allocation Hub Covering Problem

Andreas T. Ernst, Houyuan Jiang, Mohan Krishnamoorthy and Davaatseren Baatar

Cambridge Judge Business School Working Papers

Cambridge Judge Business School author contact details are as follows:

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

Please address enquiries about the series to:

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

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

Reformulations and Computational Results for the Uncapacitated Single Allocation Hub Covering Problem

Andreas T. Ernst[1], Houyuan Jiang[b], Mohan Krishnamoorthy[c], Davaatseren Baatar[d]

[a] *CSIRO Mathematics, Informatics and Statistics, Private Bag 33, Clayton South MDC, Clayton, VIC 3169, Australia*

[b] *Judge Business School, University of Cambridge, Trumpington Street, Cambridge CB2 1AG, UK*

[c] *IITB-Monash Research Academy, IIT Bombay, Powai, Mumbai 400076, India*

[d] *Department of Mechanical Engineering, Monash University, Clayton, VIC 3800, Australia.*

Abstract

We study the *single allocation hub covering problem*, which is a special case of the general hub location problem. The hub covering problem is also an important extension to traditional network covering problems. Hubs are located at some nodes in the network and are used to facilitate – consolidate, transfer and distribute – flows of goods or information in such networks. An important feature in hub location is that the transfer cost between hub nodes is often discounted. The hub covering problem is to locate a minimum number of hubs such that the travel cost between each o-d pair in the network does not exceed a given threshold. We propose ways to improve the best existing integer programming formulation for this problem by lifting constraints to produce facet defining inequalities. We also develop a new formulation. Our numerical results demonstrate that our new formulation performs significantly better than existing formulations in the literature even when the constraints have been lifted.

Keywords: facility planning and design, hub location, hub covering, integer programming, branch and bound

1. Introduction

Hubs form critical elements in many airline, transportation, postal and telecommunications networks. Hubs are centralized facilities in these networks whose functions are to consolidate, switch and sort flows. Flow concentration and consolidation occur on the arcs that connect hub nodes (*"hub arcs"*). By using high-capacity-low-cost methods of conducting flows along these hub arcs, economies of scale can be exploited. This allows for more efficient transportation of origin-destination (o-d) flows in terms of coordination of the flows as well as cost. Just as importantly hub location networks provide a trade-off between cost of building and using the network. Completely connected networks require a large number of links but give a direct connection between any pair of nodes. On the other hand tree networks have a minimum number of links, but the routes between points may need many hops. In standard hub location problems routes are never longer than three hops. However the total distance may still be large. Hub covering problems address this by adding a constraint on the maximum distance between any pair of points.

Typical applications of hub location include airline passenger travel [3], telecommunication systems [8] and postal networks [10, 11].

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. The hub-median problem is applicable in airline and telecommunication systems. This model of hub network design can sometimes lead to unsatisfactory results where worst-case o-d distances can, sometimes, be excessively large. This is a natural result of the fact that hub median model tend to minimise average distances rather than minimising maximum distances.

Difficulties of this kind are avoided in approaches like *hub center* models and in *hub covering* models. These models are particularly important for delivery of perishable or time-sensitive items. In the hub center problem, the main objective is one of minimizing the maximum distance or cost between o-d pairs. In the hub covering problem, the objective is to minimize the number of hubs that are selected in order to ensure that the maximum distance (or, cost) between o-d pairs does not exceed a given threshold. Campbell *et al* [7] present an extensive review of hub location theory and its applications. Many variants of the hub median problem have been well studied and several formulations and numerical approaches have been developed for these variants.

The p-hub center problem is defined in [5, 16]. Campbell [5] formulates it as a quadratic program and reformulates it as a linear program. Several new linearizations of the quadratic program are proposed by Kara and Tansel [13], who also provide numerical comparisons for the linearizations. Based on a new concept that they introduced – of the *radius* – Ernst *et al.* [9] proposed a two-index formulation for the uncapacitated single allocation p-hub center problem (USApHCP). Numerical results demonstrated that this new formulation performed much better than all other existing formulations. Ernst *et al.* [9] also develop a branch and bound method for solving the uncapacitated multiple allocation p-hub center problem (UMApHCP). This method was originally designed for solving the uncapacitated multiple allocation p-hub location problem (UMApHLP) – see [12].

The hub covering problem has been previously studied in [6, 15]. It has wide range of applications in real life problems. As a result new formulations, approaches, solution techniques and applications are emerging. Most recently, Wagner [19] introduced new formulations of the hub covering problem for quantity dependent and independent discount factor. Alamur *et al.* [1] and Calik *et al.* [4] considered the single allocation hub covering problem in incomplete hub networks. Tan *et al.* [18] considered the latest arrival hub covering problem. Of these approaches, our view is that the model and approach of Wagner [19] is the most comprehensive and hence, is considered the "state of the art" approach for hub covering problems.

In this current paper we consider the uncapacitated single allocation p-hub covering problem (USAHCoP). We show that the current "state of the art" formulation of Wagner [19] can be tightened by lifting some of the constraints to produce facet defining inequalities. We also introduce a new formulation for USAHCoP and show that this performs even better computationally.

In the above nomenclature of the problem, we use "Co" (to indicate "covering") rather than "C" in order not to confuse the suggested naming convention with the related hub center problems: USApHCP and the UMApHCP. In these hub center problems, the letter "C" stands for "Center", whereas the use of "Co" suggests "covering".

Given a completely connected network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with the node set $\mathcal{N} = \{1, 2, \ldots, N\}$ and the arc set $\mathcal{E}$, and given a *covering threshold*, β, both the USAHCoP and the UMAHCoP require the establishment of a minimum number of hubs so that the transportation cost between each o-d pair does not exceed β. We assume that each hub has infinite capacity for flow collection, transfer and distribution. In the USAHCoP, each non-hub node is allocated to a unique hub for collection, transfer and distribution of flows, whereas in the UMAHCoP, non-hub nodes can be multiply allocated to hubs. Let d_{ik} be the distance between the nodes i and k or the length of the arc $[i,k]$. We assume that $d_{ik} = d_{ki}$. Let $\alpha \in [0,1]$ be the economic discount factor applied to flow along hub arcs. For any arc $[k,l]$, we assume that the cost of that arc is d_{kl} if only k or l is a hub; if both k and l are hubs, then $[k,l]$, is a hub arc and the cost is αd_{kl}. We assume the triangular inequality property holds on d_{ik}.

We present integer programming formulations for the USAHCoP in Section 2. In Section 3 we show that the conflict constraints in Wagner's formulation can be lifted to produce facet defining inequalities. Numerical results are presented in Section 4 showing that even with the strengthened constraints this formulation performs worse than our new formulation.

2. The uncapacitated single allocation hub covering problem

The USAHCoP problem is to select the minimum number of nodes as hubs and allocate all other nodes to a single hub in such a way that the travel time/cost between any pair of nodes does not exceed a given threshold β. This problem has been shown to be NP-hard by Kara and Tansel [15].

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 β be the allowable maximum transportation cost between all o-d pairs. The USAHCoP is defined as an integer program with quadratic constraints in Campbell [5]:

USAHCoP

$$\min \quad \sum_{k=1}^{N} X_{kk} \tag{1}$$

$$\begin{aligned}
\text{s.t.} \qquad & \sum_{k=1}^{N} X_{ik} = 1, && i = 1,\dots,N && (2)\\
& X_{ik} \le X_{kk}, && i,k = 1,\dots,N && (3)\\
& (d_{ik} + \alpha d_{kl} + d_{jl}) X_{ik} X_{jl} \le \beta, && i,j,k,l = 1,\dots,N && (4)\\
& X_{ik} \in \{0,1\}, && i,k = 1,\dots,N. && (5)
\end{aligned}$$

Here the objective (1) is to minimize the total number of hubs to be used, Constraint (2) implies that each node is assigned to exactly one hub, Constraint (3) states that each node can only be assigned to a node that is selected to be a hub, Constraint (4) ensures that the transportation cost between each o-d pair does not exceed the threshold cost β, and Constraint (5) specifies that X_{ik} is a binary variable.

We observe that USAHCoP, as defined by Constraints (1)-(5), above, is a quadratic formulation due to Constraint (4). A few linearizations of USAHCoP have been presented in the literature.

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 the USAHCoP based on s_{ijkm} is proposed in Campbell [5] and is not reproduced here.

Kara and Tansel [13] consider a few linear programming versions of the quadratic formulation for the USAHCoP. They also introduce the following two-index linear programming formulation. Replace Constraint (4) with

$$(d_{ik} + \alpha d_{kl}) X_{ik} + d_{jl} X_{jl} \le \beta, \quad i,j,k,l = 1,\dots,N. \tag{6}$$

This formulation is called HC-Lin in [15].

The formulation HC-Lin has fewer variables than other known linear programming formulations in the literature. Moreover it is reported in Kara and Tansel [13] 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.

Constraint (6) can be further replaced by the following strengthened constraint,

$$\sum_{k=1}^{N}(d_{ik}+\alpha d_{kl})X_{ik}+d_{jl}X_{jl}\le\beta,\quad i,j,l=1,\dots,N. \tag{7}$$

This strategy has been used for reformulating the hub center problem in Kara and Tansel [14], but not used for the USAHCoP in [13]. We call this new formulation HCP-KT. The strengthened formulation HCP-KT is much more compact than HCP-Lin since the HCP-KT has $\mathrm{O}(N^3)$ number of constraints while the HCP-Lin has $\mathrm{O}(N^4)$ number of constraints.

Wagner [19] introduced a compact model for single allocation hub covering model, refered as SAQI–W2, which improved the formulation HCP-Lin further. First, some assignments are ruled out in advance by considering its expected maximum travel distance. The variables for valid assignments (VA) are defined as

$$VA=\{(i,k)\,|\,2d_{ik}\le\beta\ \text{ and }\alpha\min_j d_{kj}\le\beta\} \tag{8}$$

Then, some threshold constraints (6) are removed if $d_{ik}+\alpha d_{km}+d_{mj}\le\beta$, since they are obsolete, and the remaining constraints are replaced by tighter constraints

$$X_{ik}+X_{jm}\le 1\quad\forall(i,k,j,m)\in IA \tag{9}$$

where

$$IA=\{(i,k,j,m)\,|\,(i,k),(j,m)\in VA, i<j,\ \text{ and }d_{ik}+\alpha d_{km}+d_{mj}>\beta\}. \tag{10}$$

Furthermore, the constraints (9) are tightened by aggregating the constraints. For each pair (i,j), $i<j$, all constraints (9) replaced by aggregated constraints. Wagner [19] proposed the heuristic Algorithm 2.1 to aggregate the constraints. The resulting formulation consisting of (1)–(3) plus the aggregate constraints produced by Algorithm 2.1 are referred to by Wagner as **SAQI-W2**.

Algorithm 2.1 Heuristic for Aggregating Constraints

Let $M=\{(k,m)\mid(i,k,j,m)\in IA\}$
while $M\neq\emptyset$ **do**
  $\overline{k}=\arg\max_k\ \big|\{(k,m)\mid(k,m)\in M\}\big|$
  $\overline{m}=\arg\max_m\ \big|\{(k,m)\mid(k,m)\in M\}\big|$
  if $\big|\{m\mid(\overline{k},m)\in M\}\big|<\big|\{k\mid(k,\overline{m})\in M\}\big|$ **then**
    add constraints $\sum_{k:(k,\overline{m})\in M}X_{ik}+X_{j\overline{m}}\le 1$
    set $M=M\setminus\{(k,\overline{m})\mid(k,\overline{m})\in M\}$
  else
    add constraints $X_{i\overline{k}}+\sum_{m:(\overline{k},m)\in M}X_{jm}\le 1$
    set $M=M\setminus\{(\overline{k},m)\mid(\overline{k},m)\in M\}$
  end if
end while

The formulation for USAHCoP by Wagner [19] is, in our view, the current "state of the art". It has fewer variables and constraints than HCP-Lin. Moreover, the constraints are tighter. Lastly, the number of variables and constraints depend on the threshold value of β.

We now propose a new two-index formulation for the USAHCoP based on a concept which we define as the *radius of hubs*. This concept is introduced in [9] for reformulating the uncapacitated single allocation hub center problem.

Let r_k be a non-negative variable representing the radius of hub k. That is, r_k is the maximum collection/distribution cost between hub k and the nodes that are allocated to hub k. The USAHCoP can be formulated as a mixed integer linear program based on the radius:

USAHCoP-r

$$\min \quad \sum_{k=1}^{N} X_{kk} \tag{11}$$

$$\text{s.t.} \qquad \sum_{k=1}^{N} X_{ik} = 1, \tag{12}$$

$$X_{ik} \le X_{kk}, \qquad i,k = 1,\dots,N \tag{13}$$

$$r_k \ge d_{ik}\, X_{ik}, \quad i,k = 1,\dots,N \tag{14}$$

$$r_k + r_m + \alpha d_{km} \le \beta, \qquad k,m = 1,\dots,N \tag{15}$$

$$X_{ik} \in \{0,1\}, \qquad i,k = 1,\dots,N \tag{16}$$

Constraints (12) and (13) are standard. Constraint (14) indicates that the radius of a hub is greater than or equal to any unit collection or distribution cost of the 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 (15) plays the same role as Constraint (6).

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), we can construct a new optimal solution (X^+_{ik}, r^+_k) such that $X^+_{ik} = X^*_{ik}$ for any i and k, $r^+_k = r^*_k$ for any hub node k, and $r^+_k = 0$ for any non-hub node k. Furthermore, Constraint (15) 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 $\alpha \le 1$. This argument is also valid for the USApHCP. See Ernst *et al.* [9] for a proof of this.

3. Conflict graph

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the nodes represent the assignment variables X_{ik}; and two nodes X_{ik} and X_{jm} are connected by an edge $e = (X_{ik}, X_{jm})$ if and only if those two assignments can not be realized at the same time. In other words, each constraints in (9) is represented by an edge in this conflict graph. We refer to this graph as a *"conflict graph"*. Note that $\mathcal{E}$ is a larger set than IA defined in (10):

$$\mathcal{E} \;=\; \big\{(X_{ik}, X_{jm}) \mid (i,k,j,m) \in IA\big\} \tag{17a}$$

$$\cup \;\big\{(X_{ik}, X_{im}) \mid k \ne m\big\} \tag{17b}$$

$$\cup \;\big\{(X_{ij}, X_{jm}) \mid j \ne m\big\} \;\cup\; \big\{(X_{ik}, X_{ji}) \mid i \ne j\big\} \tag{17c}$$

Here (17a) is derived from the maximum distance constraint (6), (17b) from the single allocation constraint (2) and the last sets are because a node can only be allocated to a hub (3).

Conflict graphs can be used effectively to generate valid strong cuts and facet defining inequalities.

Proposition 3.1. *Let $\mathcal{S} \subseteq \mathcal{V}$ be a clique of $\mathcal{G}$. Then,*

$$\sum_{X_{ik} \in \mathcal{S}} X_{ik} \leq 1 \tag{18}$$

is a valid inequality for the USAHCoP.

This follows directly from the definition of $\mathcal{E}$. If one variable in $\mathcal{S}$ is set to one, all others must be zero to satisfy the constraints that were used to define $\mathcal{E}$.

We refer to the constraint defined by a clique as a *clique* constraint. Using the Proposition 3.1 one can easily show that the constraints obtained by the Algorithm 2.1 are indeed clique constraints. Let us demonstrate that on a example. Let $i < j$ and $\mathcal{G}^{'}$ be a subgraph induces from the graph $\mathcal{G}$ by considering all nodes X_{pq}, where either $p = i$ or $p = j$. Figure 1 depicts an instance of the subgraph $\mathcal{G}^{'}$. Nodes in $\mathcal{G}^{'}$ can be partitioned into two subsets $\mathcal{S}^i$ and $\mathcal{S}^j$ where $p = i$ and $p = j$, respectively. Then, $\mathcal{S}^i$ and $\mathcal{S}^j$

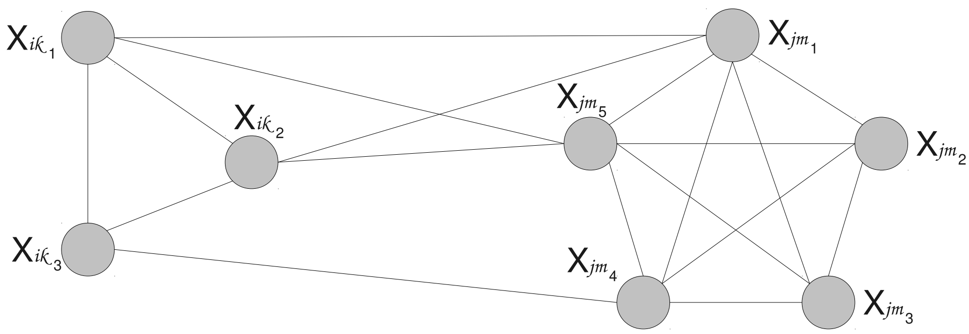

Figure 1: Example of Algorithm 2.1

are cliques since we have a single allocation problem. Clique constraints associated with these cliques are given by the single allocation constraint (2) and (12). There are more cliques that can be extracted from the graph, namely cliques that contains nodes from both sets $\mathcal{S}^i$ and $\mathcal{S}^j$. To do so, for each node the algorithm considers the number of edges that are connecting the node to the nodes in the other partition. Then the algorithm extracts cliques starting from the node with largest number of edges. As soon as the clique is extracted the edges of the clique that are connecting the sets $\mathcal{S}^i$ and $\mathcal{S}^j$ are removed.

In this example we have 4 candidates X_{ik_1}, X_{ik_2}, X_{jm_1} and X_{jm_5}, each has 2 edges, so we can pick any one of them, say X_{ik_1}. Then with respect to the clique set $\{X_{ik_1}, X_{jm_1}, X_{jm_5}\}$ we get the clique constraint

$$X_{ik_1} + X_{jm_1} + X_{jm_5} \leq 1 \tag{19}$$

While repeating the procedure we get the remaining clique constraints

$$X_{ik_2} + X_{jm_1} + X_{jm_5} \leq 1 \tag{20}$$

$$X_{ik_3} + X_{jm_4} \leq 1 \tag{21}$$

corresponding to cliques $\{X_{ik_1}, X_{jm_1}, X_{jm_5}\}$ and $\{X_{ik_3}, X_{jm_{41}}\}$, respectively.

However, we could get a much tighter constraint if we have considered a clique set $\{X_{ik_1}, X_{ik_2}, X_{jm_1}, X_{jm_5}\}$ instead of the cliques $\{X_{ik_1}, X_{jm_1}, X_{jm_5}\}$ and $\{X_{ik_1}, X_{jm_1}, X_{jm_5}\}$, which replace constraints (19) and (20) by a single constraint:

$$X_{ik_1} + X_{ik_2} + X_{jm_1} + X_{jm_5} \leq 1.$$

We can formalise this observation and show that maximum clique constraints provide facet defining inequalities. We give a brief summary of the theory below though similar results have been shown for set packing and clique polytopes previously (see [2, 17]). We say a clique is a *maximal clique* if it can not be extended any further. Clearly maximum clique constraints are the strongest of this category of cuts and in fact are facet defining. In order to make the exposition easier we will consider the integer polytope

$$\begin{aligned} \mathcal{P} = \Big\{ X \quad | \quad & \sum_{k=1}^{N} X_{ik} \leq 1 \;\forall\; i = 1, \ldots, N; \\ & X_{ik} \leq X_{kk} \quad \forall\; i, k = 1, \ldots, N; \\ & d_{ik}\, X_{ik} + \alpha\, d_{kl}\, X_{kl} + d_{lj}\, X_{jl} \leq \beta \;\; \forall\; i, j, k, l = 1, \ldots, N \\ & X_{ik} \in \{0, 1\} \Big\} \end{aligned} \tag{22}$$

This is the set of solutions for a slight relaxation of the original problem where nodes may not be allocated anywhere. This is convenient as it leads to a full dimensional polytope. It is also equivalent to the original problem, in fact if a suitably large constant is subtracted from the cost of any of the variables then the optimal solution of USApHCoP is also optimal for the problem with the relaxed feasible set $\mathcal{P}$ and vice versa any optimal solution to $\mathcal{P}$ will be feasible for USApHCoP (provided a feasible solution exists).

Theorem 3.1. *If $\mathcal{S}$ is a maximal clique of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ then the corresponding clique constraint (18) is a facet defining inequality for the polytope $\mathcal{P}$.*

Proof: First note that $0 \in \mathcal{P}$. Let $x(i, k)$ denote the vector x such that $x_{ik} = 1$ and $x_{kk} = 1$ (or $i = k$) but all other elements are zero. We will assume w.l.o.g. that $x(i, k) \in \mathcal{P}$ for all $(i, k) \in \mathcal{V}$, as otherwise (i, k) and perhaps (k, k) can be removed from $\mathcal{V}$ to give an equivalent problem. Hence the number of dimensions of the set of convex combinations of points in $\mathcal{P}$ is equal to the number of variables, i.e. $\dim(\mathrm{conv}(\mathcal{P})) = |\mathcal{V}| \leq N^2$

We will now show that there are $|\mathcal{V}|$ linearly independent solutions that satisfy (18) with equality. Clearly $x(i, k)$ lies on the constraint for $(i, k) \in \mathcal{S}$. Now

$$\forall\; (j, l) \in \mathcal{V} \setminus \mathcal{S},\; \exists (p, q) \in \mathcal{S} : x(j, l) + x(p, q) \in \mathcal{P}.$$

This is easy to see since if no such (p, q) were to exist then (p, q) could be added to the clique, but $\mathcal{S}$ is already maximal. Obviously the set of all such $x(i, k)$ and $x(j, l) + x(p, q)$ are linearly independent thus proving the result. ■

Note that generating all such maximal cliques is in general too computationally expensive and leads to an excessive number of constraints. For our computational experiments we will restrict ourselves to extending the constraints proposed by Wagner into maximal clique constraints, though a branch and cut approach would also be possible (though separating maximum clique cuts is non-trivial).

This gives rise to the following formulation:

SAQI-Cliq

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{N} X_{kk} && && (23)\\
\text{s.t.} \quad & \sum_{k=1}^{N} X_{ik} = 1, && i = 1,\dots,N && (24)\\
& \sum_{(i,k)\in C} X_{ik} \le 1 && C \in \mathcal{C} && (25)\\
& X_{ik} \le X_{kk}, && i,k = 1,\dots,N && (26)\\
& X_{ik} \in \{0,1\}, && i,k = 1,\dots,N. && (27)
\end{aligned}
$$

Here the set of cliques $\mathcal{C}$ is generated by starting with Algorithm 2.1 and then testing all other (i,k) pairs in a random order to see if they can be added to the clique in order to generate a maximum clique. An (i,k) can be added if for all other (j,l) already in C at least one of the following conditions holds:

- $d_{ik} + \alpha\, d_{kl} + d_{jl} > \beta$ (exceeds threshold);
- $i == j$ and $k \neq l$ (multiple assignment);
- $k == j$ and $j \neq l$ (i allocated to non-hub node j);
- $l == i$ and $i \neq k$ (i must be a hub for j to allocate to it).

Typically multiple additional variables can be lifted into the constraints in this way.

4. Numerical experiments

We tested our algorithms for both the USAHCoP with the AP data set [11][1] and Turkish highway data [18][2]. The AP data set is derived from the real-world application of a postal delivery network in Australia, though in our tests here we have discarded the collection and distribution costs and only use the distance matrix given in the data set. The Turkish highway data is real data where demand centers correspond to 81 cities in Turkey.

We tested the models with different threshold values of β. After solving a problem the value of β is chosen for the next problem in the following way

$$
\beta = \begin{cases} 0.95 * \beta & \text{if the problem is feasible} \\ 1.25 * \beta & \text{otherwise} \end{cases}
$$

In this way we were able to test the problem for different loose or tight threshold values. As an initial value of β we used the average of the maximum distances of the nodes. If the threshold is too tight the problem is infeasible and both approaches detected it quickly. Therefore, we did not include infeasible problem instances in the following tables.

We used CPLEX 12.1 embedded in C++ to test the formulations. First we compared USAHCoP-r against SAQI-W2. In our initial tests we followed the suggestions given in Wagner [19] and modified the model slightly by replacing the objective function coefficients for X_{kk} by $1 + k/10000$ and set the MIP

[1] Available from the OR Library http://people.brunel.ac.uk/~mastjjb/jeb/info.html
[2] Available from http://www.bilkent.edu.tr/~bkara/hubloc.htm, last accessed December 2010.

gap tolerance to $1 - n(n-1)/10000$. However, with CPLEX 12.1 it did not improve the CPU times as was reported. Therefore, in our tests we used the original formulation.

Also some preliminary analysis showed that CPLEX's cut generation at the root, while effective in raising the lower bound, generally made the overall solution times worse. Hence we turned off all cut generation. This also allows us to report a more accurate root lower bound value. The CPLEX solver was invoked twice, initially with the number of branch and bound nodes limited to zero to only solve the root node. It appears that this initial root solve includes not only pre-processing but also some post processing and at least one heuristic. The second call to the solver than starts with the root node and carries out the branch and bound, up to the time limit of 1800 seconds (half hour) of CPU time.

All tests where carried out on a Linux computer with four quad-core Intel Xeon processors running at 2.93GHz and 64Gb of RAM. Although CPLEX 12.1 can uses multiple threads during the branch and bound, all CPU times reported below are for the total processing time across all processors.

Numerical results are shown in Tables 1–6. The labels of the columns are as follows:

β - threshold value

p^* - optimal number of hubs

LB - the lower bound on the number of hubs at the root node.

t_{root} - CPLEX time to solve the root node.

t_{all} - total CPU time including model formulation (finding cliques etc), t_{root} and the branch and bound.

B&B - the number of nodes in the branch and bound tree.

NZ - number of non-zero coefficients in ths model. This appears to be a better indicator of the time taken to solve the linear programming formulation than the number of rows or columns

4.1. Comparison of SAQI formulation variants

		USAHCoP-r				SAQI-W2				SAQI-Cliq			
β	p*	LB	t_{all}	t_{root}	NZ	LB	t_{all}	t_{root}	NZ	LB	t_{all}	t_{root}	NZ
50414.5	5	4.13	0.12	0.03	11,000	5.0	0.08	0.07	7,102	5.0	0.32	0.2	17,175
50546.6	5	4.13	0.11	0.03	11,000	5.0	0.09	0.08	7,094	5.0	0.27	0.15	17,003
51328.6	5	4.11	0.12	0.03	11,000	5.0	0.06	0.05	7,780	5.0	0.33	0.2	18,555
52259.4	5	4.0	0.15	0.04	11,000	4.5	0.12	0.08	8,208	4.5	0.32	0.14	18,869
53067.9	4	4.0	0.11	0.03	11,000	4.0	0.06	0.05	8,518	4.0	0.27	0.11	18,995
53207.0	4	4.0	0.11	0.04	11,000	4.0	0.08	0.07	9,876	4.0	0.34	0.15	24,994
54030.2	4	4.0	0.14	0.05	11,000	4.0	0.22	0.19	16,277	4.0	1.62	1.26	47,267
55009.9	4	4.0	0.21	0.06	11,000	4.0	0.27	0.25	18,175	4.0	1.73	1.32	49,759
55860.9	4	4.0	0.11	0.06	11,000	4.0	0.26	0.23	20,046	4.0	3.18	2.75	52,841
56007.4	4	4.0	0.11	0.07	11,000	4.0	0.36	0.32	20,496	4.0	3.15	2.69	53,577
56873.8	4	3.0	0.23	0.06	11,000	3.33	0.5	0.37	24,887	3.33	2.84	2.14	65,150
57905.1	4	3.0	0.28	0.06	11,000	3.2	0.91	0.79	29,108	3.25	5.46	4.67	81,115
58801.0	3	3.0	0.17	0.07	11,000	3.0	1.12	1.07	32,349	3.0	3.46	2.79	85,182
58955.1	3	3.0	0.17	0.08	11,000	3.0	0.94	0.9	31,878	3.0	3.44	2.79	83,251
59867.2	3	3.0	0.16	0.06	11,000	3.0	2.69	2.6	36,581	3.0	9.31	8.48	94,687
60952.8	3	2.0	0.29	0.06	11,000	2.5	1.72	1.47	42,786	2.5	10.44	9.01	115,690
62058.0	2	2.0	0.13	0.06	11,000	2.0	1.24	1.2	48,391	2.0	17.61	16.62	130,038

Table 1: AP data with 40 nodes, $\alpha = 0.75$

The SAQI-W2 model has fewer variables than USAHCoP-r and in some cases fewer non-zeros. Table 1 shows that the solution time for the root node and hence overall solution times appear to be slightly faster for the SAQI-W2 model where the number of non-zeros is less than for the radius formulation USAHCoP-r. However as soon as the number of non-zero coefficient increases, the solution times slow down significantly. This is also true when going from SAQI-W2 to SAQI-Cliq which have identical number of constraints and variables but different number of coefficients due to the lifting in of additional variables into the clique constraints. Note that CPLEX solved all of the problems in Table 1 without branching for the SAQI formulations. For the USAHCoP-r formulation many were solved at the root node though six of them required a limited amount of branching (a maximum of 29 nodes for $\beta = 60952.8$).

		USAHCoP-r				SAQI-W2					SAQI-Cliq				
β	p*	LB	B&B	t_{all}	t_{root}	LB	B&B	t_{all}	t_{root}	NZ	LB	B&B	t_{all}	t_{root}	NZ
51174.7	5	5.0	0	0.09	0.06	5.0	0	0.1	0.08	11,994	5.0	0	0.46	0.17	30,210
51308.8	5	5.0	0	0.09	0.06	5.0	0	0.12	0.1	14,431	5.0	0	0.88	0.53	38,324
52102.6	5	5.0	0	0.09	0.06	5.0	0	0.13	0.11	18,404	5.0	0	2.21	1.66	50,545
53047.4	4	4.0	0	0.12	0.07	4.0	0	0.19	0.17	21,751	4.0	0	2.7	2.08	63,707
53868.1	4	4.0	0	0.1	0.07	4.0	0	0.32	0.3	28,131	4.0	0	3.41	2.6	82,562
54009.3	4	4.0	0	0.19	0.09	4.0	0	0.36	0.33	28,866	4.0	0	4.1	3.28	85,306
54844.9	4	4.0	0	0.18	0.08	4.0	0	0.61	0.52	35,624	4.0	0	12.14	11.03	108,091
55839.4	4	4.0	0	0.18	0.12	4.0	0	0.95	0.86	42,006	4.0	0	7.48	6.13	129,006
56703.3	4	3.12	14	0.43	0.07	3.67	0	1.61	1.09	48,174	3.67	7	24.53	15.5	146,322
56851.9	4	3.0	4	0.32	0.08	3.67	0	2.25	1.91	49,090	3.6	0	33.78	18.61	146,169
57731.5	4	3.0	11	0.31	0.1	3.5	6	10.46	1.61	56,318	3.44	0	31.16	26.69	162,456
58778.3	4	3.0	43	0.61	0.12	3.0	0	3.26	2.11	66,335	3.19	0	56.24	52.12	195,083
59687.7	4	3.0	40	0.57	0.14	3.0	3	6.69	2.8	75,105	3.08	0	61.04	56.82	208,171
59844.1	4	3.0	32	0.65	0.15	3.0	3	11.91	5.0	75,362	3.07	0	57.21	52.28	210,614
60770.0	3	3.0	0	0.31	0.13	3.0	0	4.01	3.33	89,592	3.0	0	37.73	30.88	266,315
61871.9	3	3.0	0	0.23	0.13	3.0	0	4.66	4.0	102,501	3.0	0	144.74	138.95	300,028
62993.8	3	2.0	70	2.62	0.14	2.5	0	4.69	2.32	111,834	2.5	0	159.97	140.23	313,608

Table 2: AP data: 50 nodes, $\alpha = 0.75$. The USAHCoP-r formulation always had 17,250 NZ coefficients.

Table 2 the results for the slightly larger problems with 50 nodes are presented. Again the same pattern can be observed. The increasing tightness of SAQI-W2 and SAQI-Cliq comes at a significant price in terms of the time to solve the root node relaxation leading to worse overall run times except for the more trivial cases with very tight β values. The lifted constraints of SAQI-Cliq only add fairly little to the lower bound while imposing a significant additional computational cost. In fact the increase in computation costs are rather surprising. Comparing SAQI-W2 and SAQI-Cliq which have the same formulation except for the additional coefficients in the clique constraints, we see a three fold increase in the number of coefficients leading to an increase in the root node solve time by a factor of 10 or more.

There are also some peculiar instances in Table 2 such as for $\beta = 56851.9$ and $\beta = 57731.5$, where the lower bound at the root nodes is slightly larger for SAQI-W2 than for the tighter formulation SAQI-Cliq. While the number of cuts to be generated by CPLEX has been limited to zero, it still carries out preprocessing. There are significant differences in how effectively the two formulations are preprocessed and this may have some impact on the initial lower bound. For example for $\beta = 57731.5$ the number of rows is reduced to $9,198$ for SAQI-W2 but only to $16,517$ for SAQI-Cliq even though both start off with $17,437$ constraints initially. Interestingly CPLEX reports that after pre-processing SAQI-W2's number of non-zero coefficients had increased by about 15% while for SAQI-Cliq it had reduced by about 17%. We conjecture that these differences in preprocessing are the cause for the exceptions in the lower bound and may partly be the reason why SAQI-Cliq performs so much worse than the original SAQI-W2 formulation.

4.2. Reduced SAQI Formulation

The better performance of SAQI-W2 compared to SAQI-Cliq suggests the possibility of improving the performance by taking the opposite approach. Carry out "inverse-lifting" to reduce the number of coefficients and minimise the size of the formulation at the expense of tightness. The formulation SAQI-Min uses constraints constructed as follows:

1. Generate constraints using Algorithm 2.1 as for SAQI-W2;
2. Extend these constraints into maximal cliques in a randomised way as for SAQI-Cliq but only add variables if they violate the threshold with at least one existing variable;
3. Go through all constraints in order of decreasing clique size and remove any variables that are not required for logical correctness. That is we remove a variable X_{ik} if for every other X_{jl} in the constraint the distance threshold (4) holds or the same pair X_{ik}, X_{jl} already appears in an earlier constraint.

		SAQI-W2						SAQI-Min					
n	β	p*	LB	B&B	t_{all}	t_{root}	NZ	p*	LB	B&B	t_{all}	t_{root}	NZ
50	56703.3	4	3.67	0	1.55	1.09	48,174	4	3.67	0	3.71	1.59	32,073
50	56851.9	4	3.67	0	2.21	1.91	49,090	4	3.67	0	1.99	1.67	32,784
50	57731.5	4	3.5	6	10.4	1.61	56,318	4	3.33	0	2.81	2.2	36,989
50	58778.3	4	3.0	0	3.19	2.11	66,335	4	3.11	0	2.36	1.88	42,916
50	59687.7	4	3.0	3	6.61	2.8	75,105	4	3.0	0	4.57	2.74	48,635
50	59844.1	4	3.0	3	11.83	5.0	75,362	4	3.0	0	10.53	2.7	48,861
50	60770.0	3	3.0	0	3.93	3.33	89,592	3	3.0	0	4.51	3.88	56,938
50	61871.9	3	3.0	0	4.57	4.0	102,501	3	3.0	0	22.57	5.62	64,164
50	62993.8	3	2.5	0	4.59	2.32	111,834	3	2.5	0	9.96	5.63	70,015
100	52376.0	7	7.0	0	4.72	4.51	158,901	7	7.0	0	4.84	4.65	102,934
100	55132.6	5	4.43	5	1,711.32	12.23	323,942	5	4.33	0	221.42	121.94	209,514
100	58034.4	13†	3.38	0	1,800.62	120.78	581,868	4	3.5	0	672.35	421.54	363,459
100	61088.8	4†	3.0	0	1,800.43	628.36	920,721	11†	3.0	0	1,802.15	1,047.7	572,353

Table 3: AP data: $\alpha = 0.75$. Comparisons of the SAQI-Min formulation for AP data sets with 50 and 100 nodes. Solutions marked with † represent the best heuristic value found within the 1800 second time limit.

In Table 3 shows that while the reduced formulation size can have a positive effect on solution times this is not always the case. In these results there appears to be very little correlation between formulation size and the time taken to solve the root node. For example the instance with $n = 50$, $\beta = 59844.1$ where the number of coefficients has been reduced by 35% and the root node run time nearly halved. On the other hand, $n = 50$, $\beta = 62993.8$ where the number of coefficients was reduced by a slightly larger percentage the root node solve time more than doubled. This is despite the fact that even after preprocessing the SAQI-Min MILP was much smaller (12,482 rows, 1,834 columns and 80,703 non-zeros) than the SAQI-W2 MILP (22,063 rows, 1,834 columns and 127,210 non-zeros). Even where there are significant differences in the CPU time used by CPLEX, these appear to be more due to "luck", that is how quickly CPLEX managed to find a good heuristic solution with no obvious connection to properties of the two formulations.

4.3. Results for plain branch and bound

The curious anomalies mentioned above in the relationship between the types of constraints used, the root lower bound and run time deserve further investigation. To eliminate the effects of some of the sophisticated preprocessing, cutting and heuristic solution methods that are included in CPLEX, we decided to turn off all of these[3]. In addition we stopped CPLEX from choosing the root node solution algorithm automatically by fixing this to the primal simplex method.

		SAQI-W2				SAQI-Cliq				SAQI-Min			
β	p*	LB	B&B	t_{all}	t_{root}	LB	B&B	t_{all}	t_{root}	LB	B&B	t_{all}	t_{root}
51174.7	5	5.0	2	1.88	0.12	5.0	0	0.68	0.36	5.0	0	0.3	0.1
51308.8	5	5.0	0	0.1	0.08	5.0	0	0.59	0.19	5.0	0	0.5	0.26
52102.6	5	5.0	0	0.25	0.22	5.0	0	0.75	0.2	5.0	0	0.61	0.32
53047.4	4	4.0	0	0.27	0.22	4.0	0	1.0	0.31	4.0	2	2.68	0.52
53868.1	4	4.0	0	0.27	0.24	4.0	0	1.46	0.56	4.0	0	0.96	0.42
54009.3	4	4.0	0	0.42	0.35	4.0	0	1.33	0.4	4.0	0	1.74	1.17
54844.9	4	4.0	0	0.56	0.48	4.0	1	3.46	1.5	4.0	0	0.89	0.21
55839.4	4	4.0	0	0.59	0.5	4.0	0	2.52	0.79	4.0	0	1.14	0.3
56703.3	4	3.5	2	14.27	0.61	3.67	4	30.82	1.61	3.67	3	5.32	0.37
56851.9	4	3.33	3	21.46	0.75	3.6	3	26.45	1.92	3.57	11	12.4	0.45
57731.5	4	3.12	10	24.28	1.0	3.44	4	50.97	2.59	3.25	7	10.18	0.53
58778.3	4	3.0	7	35.61	1.25	3.19	8	161.16	3.48	3.0	7	28.71	0.74
59687.7	4	3.0	104	756.79	1.04	3.08	7	226.02	5.5	3.0	15	63.2	0.78
59844.1	4	3.0	24	67.61	0.74	3.07	8	203.02	5.62	3.0	22	73.9	1.04
60770.0	3	3.0	11	29.41	1.16	3.0	0	9.31	5.71	3.0	2	13.71	0.8
61871.9	3	3.0	3	27.32	0.99	3.0	0	11.61	7.56	3.0	0	3.15	0.92
62993.8	3	2.12	4	48.69	2.03	2.5	3	253.75	8.51	2.5	5	22.82	2.23

Table 4: AP data with 50 nodes $\alpha = 0.75$. Comparisons of all three SAQI variants using plain branch and bound (all preprocessing, cuts and heuristics turned off).

The new results, as shown in Table 4, indicate that for this plain version of the MILP solver the anomalies disappear. The strengthened constraints always produce a tighter lower bound. Interestingly the reduced version in which coeffiecients were dropped from the maximum clique constraints stil produce lower bounds that are at least as good and sometimes better than the original formulation. The run times for the root lower bound also agree more closely with the difference in the number of non-zero coefficients (note that the "NZ" columns have not been shown as they are identical to those in Tables 2 and 3). The overall run time is slightly less predictable since it partly depends on when the branch and bound finds the optimal solution. However overall the SAQI-Min formulation with the reduced set of cuts appears to be the most effective.

4.4. Results for USAHCoP-r on large instances

Given the poor performance of the different SAQI variants we only provide detailed results for large instances using the USAHCoP-r formulation. Indeed for 200 node AP data sets none of the SAQI variants were able to solve any of the instances and in most cases failed to even solve the root node. Table 5 shows that the USACoP-r managed to solve all of these problems, though in the case of $n = 200$, $\beta = 63903.5$ CPLEX reported using slightly more CPU time than the time limit of 1800 seconds given to it. It is also interesting to note that CPLEX as of version 12 can carry out the branch and bound in parallel while only using a single CPU core for the root node. This means that for USAHCoP-r the elapsed (wall-clock) time is often significantly less than the CPU times reported in the table. In fact the only problem that required more than 500 seconds of real time was $n = 200$,$\beta = 67266.8$ which took 707.55 seconds. The only 200 node instance for which SAQI-W2 solved the root node within the 1800 second time limit was for $\beta = 57672.9$ where it obtained a lower bound of 5.67 in 1624.04 seconds.

[3]Specifically we turned off the parameters: PreInd, PrePass, AggInd, RelaxPreInd, RINSHeur, FPHeur, HeurFreq, CutPass, EachCutLim, FracCuts

n	β	p*	LB	B&B	t_{all}	t_{root}	NZ
100	52376.0	7	6.0	31	3.17	0.53	69,500
100	55132.6	5	4.0	36	4.73	0.7	69,500
100	58034.4	4	3.0	13	3.46	0.77	69,500
100	61088.8	3	3.0	78	23.19	1.2	69,500
200	57672.9	6	5.0	151	1,167.1	8.08	279,000
200	60708.3	4	4.0	257	1,189.37	12.84	279,000
200	63903.5	4	3.0	165	1,871.56	11.01	279,000
200	64892.1	3	2.0	85	551.99	10.29	279,000
200	65062.2	3	2.0	183	1,610.0	10.04	279,000
200	66068.8	3	2.0	78	536.8	14.82	279,000
200	67266.8	3	2.0	103	1,483.32	18.41	279,000
200	68486.6	2	2.0	100	726.63	25.51	279,000
200	69546.1	2	2.0	4	69.75	30.37	279,000
200	70807.2	2	2.0	90	1,195.72	26.49	279,000
200	73206.4	2	2.0	0	35.55	19.47	279,000
200	74533.9	2	2.0	51	350.14	42.52	279,000
200	77059.4	1	1.0	0	22.34	18.63	279,000

Table 5: USAHCoP-r results for large AP data sets with 100 and 200 nodes, $\alpha = 0.75$.

The approaches have been tested on Turkish high way data, as well. The value of the threshold parameter β is defined in the same way as for AP. For each threshold we tested with 5 different discount factors 0.6, 0.7, 0.8 and 0.9. For some combinations of α and β there is no feasible solution and hence the corresponding parts of the table are left blank.

The test results are shown in Table 6, though for the SAQI-W2 and SAQI-Min formulations we only show the solution status in the 1800 second time limit, that is whether CPLEX managed to solve the root node or to solve the instance to optimality. We also indicate where the solver is able to find at least a non-trivial heuristic solution. Typically CPLEX would start with the trivial solution of making every node a hub and only after some time find a better solution with no more than 10 hubs.

The results indicate that SAQI-Min tended to perform slightly better than SAQI-W2. However the results are not consistent nor the differences very large. For example for $\alpha = 0.7$, $\beta = 1962.3$ SAQI-W2 found the optimal solution but it took 1041.9 seconds almost all of which were required to solve the root node requiring just 1 second to find the optimal solution with a root node heuristic. On the other hand SAQI-Min which had far fewer rows and non-zero coefficients took 1444.6 seconds for the root node and found no heuristic solution in the remaining time. To get an indication of the difference in solution quality between the different formulations we calculated the average number of hubs over all instances where both SAQI-W2 and SAQI-Min at least found a non-trivial heuristic solution. This gives 5.10 hubs for SAQI-W2 and 4.41 for SAQI-Min, though the optimal solutions found by USAHCoP-r require only 3.38 hubs on average for these instances.

It is clear that the USAHCoP-r formulation performed very well on a range of problems that could not be solved using SAQI-W2 and SAQI-Min. The only instance for which USAHCoP-r required more than one minute of CPU time was with $\alpha = 0.6$ and $\beta = 1518.4$. Here the lower bound was quite weak. However even though SAQI-W2 and SAQI-Min could both obtain a stronger lower bound (2.86 and 3.48 respectively), they each only found a heuristic solution with 6 hubs without being able to complete the branch and bound within the time limit.

	$\alpha = 0.6$							$\alpha = 0.7$						
β	p*	LB	B&B	t_{all}	t_{root}	SW	SM	p*	LB	B&B	t_{all}	t_{root}	SW	SM
1518.4	4	2.0	1197	707.79	0.25	H	H	7	5.5	363	14.69	0.18	O	O
1598.3	4	2.0	368	28.59	0.37	H	H	5	3.12	76	4.28	0.24	H	H
1602.5	4	2.0	342	40.88	0.4	H	H	5	3.12	69	4.08	0.25	H	H
1627.3	4	2.0	294	26.68	0.51	H	H	5	3.0	475	22.06	0.29	H	H
1686.9	3	2.0	251	31.18	0.64	H	H	4	2.0	238	19.73	0.33	H	H
1708.5	3	2.0	203	27.64	0.75	R	H	4	2.0	447	34.45	0.41	H	H
1739.5	3	2.0	109	19.06	0.84	H	R	3	2.0	27	2.64	0.39	H	H
1744.0	3	2.0	273	18.32	0.86	R	H	3	2.0	37	4.52	0.47	H	H
1775.7	2	2.0	22	2.8	0.92	H	R	3	2.0	29	2.87	0.64	H	H
1807.8	2	2.0	5	3.21	1.79	-	R	3	2.0	97	4.71	0.96	H	O
1835.8	2	2.0	7	2.48	1.12	-	-	3	2.0	231	27.59	1.25	-	H
1869.1	2	2.0	15	3.71	1.83	-	R	2	2.0	31	4.82	1.2	R	H
1927.4	2	2.0	4	2.71	1.13	-	-	2	2.0	7	2.18	0.77	H	H
1962.3	2	2.0	0	2.03	1.2	-	-	2	2.0	0	2.46	1.05	O	R
1967.5	2	2.0	6	3.63	2.04	-	R	2	2.0	0	2.04	1.24	R	R
2028.8	2	2.0	0	2.25	1.26	-	-	2	2.0	0	1.73	1.19	R	R
2071.0	2	2.0	11	4.14	2.12	-	-	2	2.0	0	1.95	1.32	-	R
2180.0	1	1.0	0	1.39	0.96	-	O	1	1.0	0	1.32	0.9	-	-

	$\alpha = 0.8$							$\alpha = 0.9$						
β	p*	LB	B&B	t_{all}	t_{root}	SW	SM	p*	LB	B&B	t_{all}	t_{root}	SW	SM
1518.4						I	I						I	I
1598.3						I	I						I	I
1602.5						I	I						I	I
1627.3						I	I						I	I
1686.9	6	5.09	0	0.46	0.22	O	O						I	I
1708.5	6	4.0	46	1.44	0.22	O	O						I	I
1739.5	6	4.0	568	33.34	0.26	O	O						I	I
1744.0	5	4.0	256	6.39	0.23	H	H						I	I
1775.7	4	3.0	222	13.42	0.35	O	O						I	I
1807.8	4	2.0	99	2.85	0.28	H	O						I	I
1835.8	3	2.0	38	2.83	0.45	H	O						I	I
1869.1	3	2.0	83	13.88	0.67	H	H	4	4.0	0	0.45	0.25	O	O
1927.4	2	2.0	53	3.37	0.61	H	O	3	2.0	23	0.94	0.24	O	O
1962.3	2	2.0	11	1.91	0.92	H	H	3	2.0	25	1.36	0.37	O	O
1967.5	2	2.0	6	2.46	1.5	H	O	3	2.0	21	1.64	0.33	O	O
2028.8	2	2.0	47	2.95	0.87	O	H	2	2.0	15	2.11	0.75	O	O
2071.0	2	2.0	0	1.09	1.0	H	O	2	2.0	0	1.08	0.7	O	O
2180.0	1	1.0	0	1.16	0.77	O	O	1	1.0	0	0.92	0.6	O	O

Table 6: USAHCoP-r results for the Turkish data with 81 cities.
Column "SW" and "SM" indicate the solution status for SAQI-W2 and SAQI-Min respectively:
I – Infeasible
- – Failed to complete even the root node
R – Root node solved only
H – Heuristic solution found
O – Solved to optimality

5. Concluding Remarks

We have considered a previously published [19] formulation for the single allocation hub covering problem (USAHCoP) and shown that this could be tightened by lifting some of the constraints. We have also proven that the lifted constraints are facet defining for a slightly modified version of the problem. However in computational experiments it is shown that this lifting of the constraints actually has a negative impact on the computational performance when solving the MILPs using a recent version of the CPLEX MILP solver. A better outcome can often be achieved by first lifting the constraints and then paring back the coefficients to get a minimal set of constraints. However in comparing different formulation variants empirically the actual effectiveness of the formulation depends not only on their size and tightness but also on how it interacts with the CPLEX MILP algorithms including pre-processing, cutting and heuristics.

A new formulation for USAHCoP has been proposed based on the radius concept. This uses auxiliary variables to determine the distance of the furthest node allocated to each hub. This formulation, while slightly weaker than any of the others is much more compact and was shown empirically to perform better. This conclusion is analogous to the situation for p-hub median problems where Ernst and Krishnamoorthy [10, 11] showed that a smaller, weaker formulation can outperform a larger tighter one . Using the new formulation we were able to solve larger problem instances with up to 200 nodes that were previously out of reach for exact solution approaches.

Further work will be required to see whether the facet defining inequalities for USAHCoP can be used to produce higher performing algorithms. For example it may be possible to develop a branch and cut approach to exploit the stronger constraints without paying as larger a price for the exponential growth in the number of such constraints and non-zero coefficients.

References

[1] Alumur, S. A., Kara, B. Y., & Karasan, O. E. (2009). The design of single allocation incomplete hub networks. *Transportation Research Part B*, *43*, 936–951.

[2] Atamturk, A., G.L.Nemhouser, & Savelsbergh, M. (2000). Conflict graphs in integer programming. *European Journal of Operational Research*, *121*, 40–55.

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

[4] Calik, H. S., Alumur, A., Kara, B. Y., & Karasan, O. E. (2009). A tabu search based heuristic for hub covering problem over incomplete hub networks. *Computers & Operations Research*, *36*, 3088–3096.

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

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

[7] Campbell, J. F., Ernst, A., & Krishnamoorthy, M. (2001). Hub location problems. In H. Hamacher, & Z. Drezner (Eds.), *Location Theory: Applications and Theory* (pp. 373–406). Springer-Verlag.

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

[9] Ernst, A. T., Hamacher, H., Jiang, H., Krishnamoorthy, M., & Woeginger, G. (2009). Uncapacitated single and multiple allocation p-hub center problems. *Computers and Operations Research*, *36*, 2230–2241.

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

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

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

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

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

[15] Kara, B. Y., & Tansel, B. (2003). The single-assignment hub covering problem: Models and linearizations. *Journal of the Operational Research Society*, *54*, 59–64.

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

[17] Padberg, M. (1973). On the facial structure of set packing polyhedra. *Mathematical Programming*, *5*, 199–215.

[18] Tan, P. Z., & Kara, B. Y. (2007). A hub covering model for cargo delivery systems. *Networks*, *49*, 28–39.

[19] Wagner, B. (2008). Model formulations for hub coverig problems. *Journal of the Operational Research Society*, *59*, 932–938.