Solving the Hub Location Problem in a Star–Star Network

Martine Labbé
Université Libre de Bruxelles, Département d'Informatique, Boulevard du Triomphe CP 210/01, 1050 Brussels, Belgium.

Hande Yaman
Bilkent University, Department of Industrial Engineering, Bilkent 06800 Ankara, Turkey.

We consider the problem of locating hubs and assigning terminals to hubs for a telecommunication network. The hubs are directly connected to a central node and each terminal node is directly connected to a hub node. The aim is to minimize the cost of locating hubs, assigning terminals and routing the traffic between hubs and the central node. We present two formulations and show that the constraints are facet-defining inequalities in both cases. We test the formulations on a set of instances. Finally, we present a heuristic based on Lagrangian relaxation. © 2007 Wiley Periodicals, Inc. NETWORKS, Vol. 51(1), 19–33 2008

Keywords: hub location; star–star network; polyhedral analysis; branch and cut; Lagrangian relaxation

1. INTRODUCTION

We consider the problem of locating hubs in a telecommunication network. We are given a set I of terminal nodes and a central node 0. Let $|I| = n$. We assume that $n \geq 3$. We choose a subset of the terminal nodes at which to locate hubs. Each hub node is connected to the central node by a direct link. So the network that links the hubs and the central node is a star. Each terminal node that is not designated a hub is connected directly to a hub node. Hence the network linking a hub and the terminals assigned to it is also a star. The whole network is called a star–star network (see Fig. 1).

Any pair of terminal nodes would like to communicate with each other. Their traffic is routed by hubs. The amount of traffic which must be sent from node $i \in I$ to node $m \in I$ is t_{im}.

There is a cost associated with locating a hub at a given node and a cost associated with assigning a terminal node to a hub node. There is also a cost for routing the traffic on the links between the hubs and the central node. We denote by C_{jj} the cost of locating a hub at node $j \in I$ and by C_{ij} the cost of connecting node $i \in I$ to node $j \in I \setminus \{i\}$. The cost of routing a unit of traffic between nodes j and 0 is denoted by B_j. We assume that $B_j \geq 0$ for all $j \in I$.

If two nodes i and m are assigned to the same hub, say j, the traffic from node i to node m follows the path $i \to j \to m$. So this traffic does not travel on the links between the hubs and the central node. However, if node i is assigned to node j and node m is assigned to node $l \neq j$ (in which case both j and l must be hubs), then the traffic from node i to node m follows the path $i \to j \to 0 \to l \to m$, where node 0 stands for the central node. In Figure 2, we see a network with 10 nodes where nodes 1, 2, 5, and 7 received hubs. The traffic from node 3 to node 10 follows the path $3 \to 2 \to 0 \to 7 \to 10$ since node 3 is assigned to node 2 and node 10 is assigned to node 7. The traffic from node 1 to node 6 follows the path $1 \to 0 \to 5 \to 6$ and the traffic from node 8 to node 9 follows the path $8 \to 7 \to 9$.

So, the total traffic on the link between node j and node 0 is the sum of the traffic between all nodes that are assigned to j and all nodes that are not assigned to j.

The problem is to locate the hubs and to assign the remaining nodes to the hubs in order to minimize the cost of location, assignment and routing. This problem is called the *Uncapacitated Hub Location Problem in a Star–Star Network* (UHLP-S).

The UHLP-S is encountered in designing a telecommunication network where the backbone network, which is the network that connects the hubs is a star and access networks, which are the networks that connect terminals to hubs are also stars. Different from the network design problems where there is a cost for installation of links on the edges of the network, in UHLP-S, there is a cost for routing the traffic on the links. Hence, UHLP-S is a relaxation of the network design problem. Moreover, it is an approximation for that problem

Received October 2004; accepted December 2005
Correspondence to: H. Yaman; e-mail: hyaman@bilkent.edu.tr
Contract grant sponsor: France Telecom R&D; Contract grant number: 99 1B 774
DOI 10.1002/net.20193
Published online 28 November 2007 in Wiley InterScience (www.interscience.wiley.com).

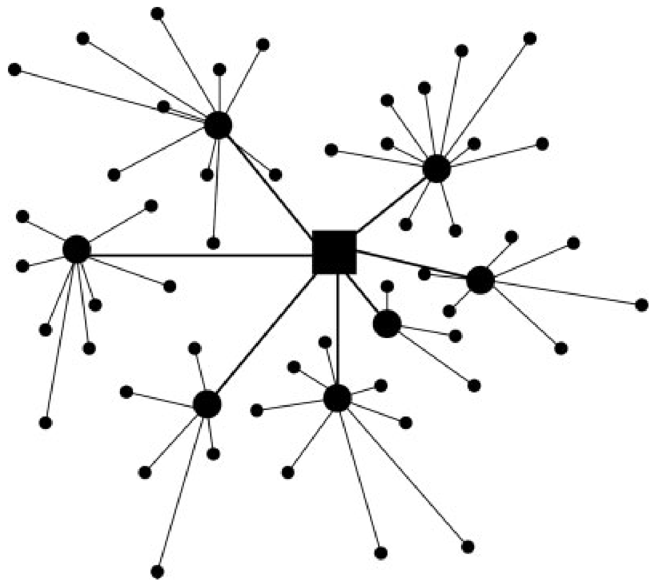

FIG. 1. A star–star network.

where the link installation cost is approximated by the routing cost.

The UHLP-S also appears in the design of a two-level access network. The central node is connected to some backbone network and the terminal nodes are connected to the central node through a two-level access network where both levels are stars. The square nodes are the backbone hubs and they are connected by two rings that share an edge (Fig. 3). Each backbone hub has a two level access network.

Applications of the UHLP-S also arise in transportation. For instance, it arises when one wants to send cargo among cities (terminals). Some of the cities are chosen to be hubs. Each remaining city is served by one hub. The traffic originating at this city is sent to the hub. At the hub, the cargo arriving from different origins are collected and sorted. If the destination is served by the same hub, then the cargo is routed to its destination. The remaining cargo is carried to a central node where it is further routed to the hub of its destination and then to its destination. The lines between hubs and the central node are served by higher capacity trucks. In UHLP-S, we are interested in minimizing the cost of installing hub nodes and the cost of routing the cargo in the network.

Traditionally, problems like UHLP-S are approximated by pure facility location problems. The study of the location problems for telecommunication network design dates back to the 1960s when Hakimi [22, 23] introduced the 1-median and p-median problems to locate switching centers in communication networks. Since then, there has been a lot of work on the facility location problems (see e.g. Refs. [14, 28, 29] for the *Uncapacitated Facility Location Problem* (UFLP) and Refs. [15, 41] for the *Capacitated Facility Location Problem* (CFLP)). These problems are used to design networks with star backbone and star access networks. The aim is to minimize the cost of installing hubs, connecting hubs to the central node, and the cost of assigning terminals to hubs (see e.g. Refs. [33, 35]).

Gourdin et al. [20] gave a survey on location problems which have applications in telecommunications. For earlier surveys, one can refer to Boffey [9] and Klincewicz [27]. Yuan [45] gives an annotated bibliography of network design problems.

Because of economic considerations in network design, star-type networks as backbone or access networks are often studied in the literature. Here, we summarize the work on networks with star components. We focus on problems that are not pure facility location problems.

Chung et al. [13] develops a model for designing a network where the backbone is fully connected and the access networks are stars. They minimize a cost function that includes the cost of installing hubs, cost of assigning terminals to hubs, and the cost of interconnecting hubs. The authors present a dual-based solution procedure and computational results. A similar problem is considered by Hardin et al. [24]. The authors present polyhedral results and develop a method based on these results.

Pirkul and Nagarajan [36] design a network where the backbone is a tree and the access networks are stars using a two-phase algorithm. The first phase uses a sweep algorithm to divide the set of nodes into regions. The second phase, for each region, determines a path from the furthest node of the region to the central node.

Lee et al. [32] also consider the same topology. They present a formulation to determine a network that minimizes the cost of installing hubs, the cost of assigning terminals to

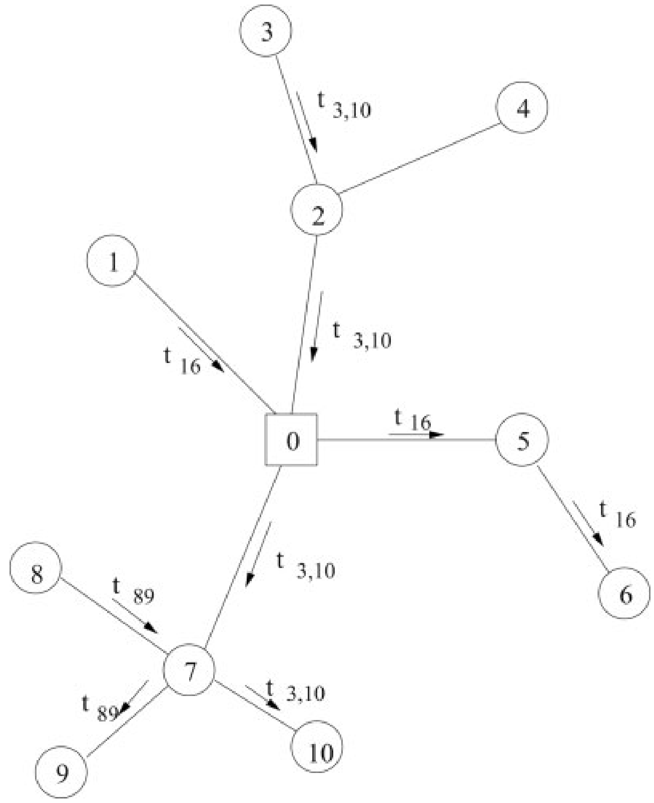

FIG. 2. Routing the traffic.

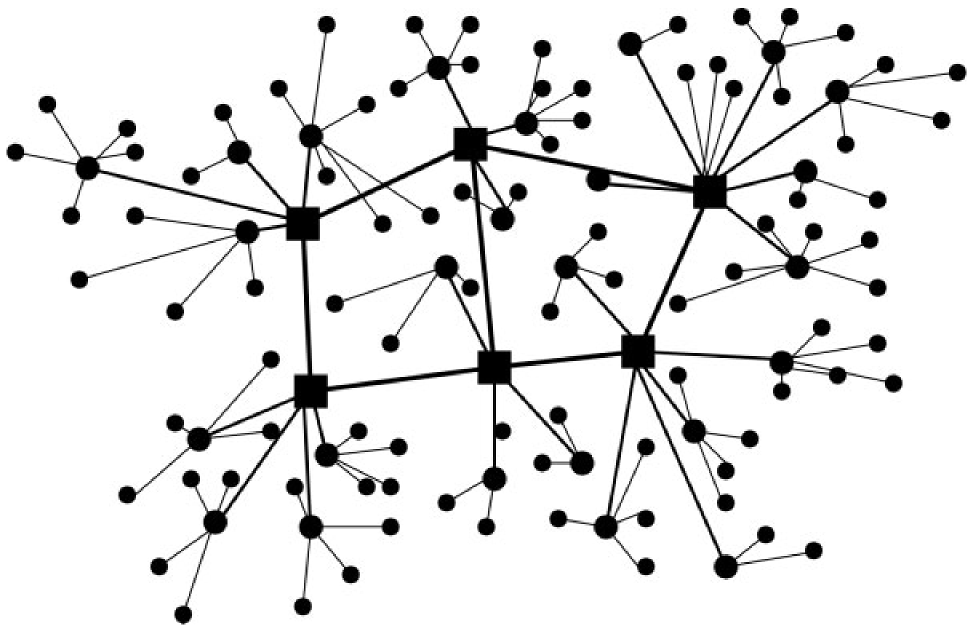

FIG. 3. Star–star two-level access network.

these hubs, and the cost of establishing the links of the spanning tree. The authors apply Lagrangian Relaxation to this model.

Gavish [19] formulates a problem where the terminals are connected to the hubs via multidrop links that are capacitated, and hubs are connected to a central unit through a star network. The objective function involves the cost of establishing the links and installing the hubs. There are different types of links with different costs and capacities.

Chardaire et al. [12] consider the design of a network with two levels of hubs, i.e., each terminal is connected to a first-level hub which is connected to a second-level hub. All second-level hubs are connected to a central unit. They present two integer programming formulations and a simulated annealing algorithm.

A very similar problem to the UHLP-S has been considered by Helme and Magnanti [25] for satellite communication networks. The authors propose a quadratic formulation and a linearization that does not increase the order of number of variables. Their formulation can be adapted to the UHLP-S for the special case where the routing cost is the same for all links between the hubs and the central node. Here we consider the general case.

Although the amount of traffic to be routed between origin–destination pairs is a crucial information, it is mostly ignored in location models. The routing of traffic between origin–destination pairs depends on the locations of hubs and on the way the terminals are assigned (connected) to the hubs. In a star–star network, when the hubs are located and the terminals are assigned, between any origin–destination pair, there is a single simple path. In UHLP-S, there is a cost term for routing the traffic in the network. Because of this additional cost, the UHLP-S is a quadratic problem unlike the classical facility location problems and is closely related to the *Uncapacitated Single Allocation Hub Location Problem* (UHLP) (see Ref. [11] for a survey on hub location problems). The difference is that in UHLP, hubs are connected to each other by a complete network. Consider a UHLP for which B'_{jl} represents the cost per unit of traffic routed from node j to node l, in the case that both nodes are hubs. If it is possible to find values B_j for each node j such that $B'_{jl} = B_j + B_l$ for all pairs of nodes j and l, then UHLP is a UHLP-S with the cost of routing traffic from a hub at j to the central node given by B_j. So UHLP-S is a special case of UHLP.

Different formulations of UHLP and of its version where the number of hubs is fixed can be found in Refs. [10, 16, 17, 31, 34, 38]. These formulations differ in the way they linearize the quadratic terms in the problem. The strongest formulations use four-index variables. It is possible to decrease the number of variables by $O(n)$ by viewing the problem as a multicommodity problem and aggregating the commodities by origin or destination. Finally, there are formulations using $O(n^2)$ variables, but such formulations turn out to be much weaker.

Sohn and Park [39] formulate the allocation problem in the case of two hubs as a linear programming problem. It is shown in Sohn and Park [40] that the allocation problem is NP-hard when the number of hubs is more than two and a formulation for the allocation problem with three nodes is given. Ernst and Krishnamoorthy [17] present a branch and bound algorithm that uses the upper bound obtained by simulated annealing, and Ernst and Krishnamoorthy [18] present an exact method based on shortest paths for the case where the number of hubs is fixed.

A Lagrangian Relaxation heuristic is given in Pirkul and Schilling [37]. This heuristic is based on the four-index formulation of UHLP. The authors relax the assignment constraints and the two sets of constraints used for linearization.

They add a relaxed version of the latter constraints. The resulting problem separates into a series of trivial problems. They obtained an average gap of 0.048% over 84 problems.

We observe that problems like UHLP-S are often solved either through MIP formulations or with heuristics. Not much is known about polyhedral properties of these problems. Polyhedral properties have been studied and exact methods based on these results have been developed for pure facility location problems (see e.g. Refs. [1, 2]), but they are rare for location problems that have additional features (like routing cost or delay cost).

Studies have been made recently on the polyhedral properties of hub location problems. Hamacher et al. [21] study the polyhedral properties of the *Uncapacitated Multiple Allocation Hub Location Problem*, i.e., the case where a terminal can be served by several hubs. Sohn and Park [40] consider the polyhedral properties of the allocation problem where the number of hubs is three. Labbé et al. [31] study the polyhedral properties of UHLP and present a branch and cut algorithm based on these results.

In this paper, we present and compare exact and heuristic methods for the UHLP-S. We first prove that UHLP-S is an NP-hard problem. Then we present two formulations. We study the properties of the corresponding polyhedra. We show that the constraints are facet-defining inequalities in both formulations. This suggests that these are strong formulations. Our computational experience also supports this claim.

Motivated by the successful implementation of Pirkul and Schilling [37], we also develop a heuristic based on Lagrangian Relaxation. Unlike in the work of these authors, we use a quadratic formulation of the problem where we relax only the assignment constraints. The Lagrangian subproblem decomposes into a series of mincut problems. The Lagrangian dual gives the same bound as the linear programming relaxations of the two formulations. The heuristic solves the Lagrangian dual with the subgradient algorithm and generates feasible solutions by transforming the solutions of the relaxations. The computational results show that the heuristic is able to find an optimal solution and prove its optimality for most of the instances.

The paper is organized as follows. In Section 2, we prove that UHLP-S is NP-hard and present two formulations. Section 3 is devoted to the polyhedral analysis. In Section 4, we compare the two formulations from the computational side. Finally, in Section 5, we present the Lagrangian Relaxation heuristic.

2. FORMULATIONS

Define $A = \{(i,j) : i \in I, j \in I \setminus \{i\}\}$. Let K and K' be the sets of all directed and undirected pairs of nodes respectively, i.e., $K = \{(i,j) : i,j \in I\}$ and $K' = \{\{i,j\} : i \in I, j \in I \setminus \{i\}\}$. Recall that t_{im} denotes the amount of traffic which must be sent from node $i \in I$ to node $m \in I$. The values t_{ii} are defined to be 0. We can compute the total amount of traffic between nodes i and m as $T_{\{i,m\}} = t_{im} + t_{mi}$. We assume that for each $i \in I$, there exists $m \in I \setminus \{i\}$ such that $T_{\{i,m\}} > 0$.

Define the assignment variable x_{ij} to be 1 if node $i \in I$ is assigned to node $j \in I$ and 0 otherwise. If node j receives a hub then it is assigned to itself, so $x_{jj} = 1$. If $B_j = 0$ for all $j \in I$, then the problem can be formulated as follows:

$$\min \quad \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j\in I} x_{ij} = 1 \quad \forall i \in I \tag{1}$$

$$x_{ij} \le x_{jj} \quad \forall i,j \in I, \ i \ne j \tag{2}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in I. \tag{3}$$

Constraints (1), (2), and (3) imply that each node either receives a hub or it is assigned to exactly one other node which received a hub.

This version of the problem is called the Uncapacitated Concentrator Location Problem (UCLP). In the sequel, we prove that it is NP-hard.

The UCLP is very close to the UFLP. The UFLP is defined as follows: given a set of clients N and a set of possible locations for facilities M, locate facilities on a subset of M and assign each client in N to a facility in order to minimize the cost of facility location and assignment. Let F_j denote the cost of locating a facility at node $j \in N$ and let D_{ij} denote the cost of assigning client i to the facility at node j. Define y_j to be 1 if a facility is located at node j and 0 otherwise and $\bar{x}_{ij}$ to be 1 if client i is assigned to the facility at node j and 0 otherwise. Then the UFLP can be formulated as follows:

$$\min \quad \sum_{j\in M} F_j y_j + \sum_{i\in N}\sum_{j\in M} D_{ij}\bar{x}_{ij}$$

$$\text{s.t.} \quad \sum_{j\in M} \bar{x}_{ij} = 1 \quad \forall i \in N$$

$$\bar{x}_{ij} \le y_j \quad \forall i \in N, \ j \in M$$

$$\bar{x}_{ij}, y_j \in \{0,1\} \quad \forall i \in N, \ j \in M.$$

The UFLP is an NP-hard problem (see Ref. [14]).

The UCLP is a special case of UFLP; it is UFLP with the set of possible locations identical to the set of clients, and with the cost of assigning a client to a facility at the same location no more than that of assigning it to a facility located at a different node. The latter ensures that C_{jj} embeds both the fixed cost of opening a facility at j and the cost of assigning client j to that facility, and there is always an optimal solution in which nodes with facilities are assigned to these facilities.

We prove that UCLP is NP-hard by reduction from UFLP. The proof is given in Appendix A.

Theorem 1. *The UCLP is NP-hard.*

As UCLP is a special case of UHLP-S, UHLP-S is also NP-hard.

We present two formulations for the UHLP-S. Note that it is possible to use the formulations for UHLP to solve UHLP-S. But we obtain smaller size formulations using the special structure of UHLP-S. Our second formulation uses similar ideas as the formulation for a capacitated version of UHLP given in Labbé et al. [31].

In the first formulation, we use the following variables. We define $u^j_{\{i,m\}}$ to be 1 if only one of nodes i and m is assigned to node $j \in I$ and 0 otherwise for $\{i,m\} \in K'$. In other words, $u^j_{\{i,m\}}$ is 1 if the traffic between nodes i and m travels on the link between node j and the central node and 0 otherwise. For a given assignment vector x, we can compute $u^j_{\{i,m\}} = x_{ij}(1-x_{mj}) + x_{mj}(1-x_{ij}) = |x_{ij} - x_{mj}|$ for $\{i,m\} \in K'$ and $j \in I$. We can formulate the UHLP-S as follows:

UHLP-S1

$$\min \quad \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{j\in I} B_j \sum_{\{i,m\}\in K'} T_{\{i,m\}} u^j_{\{i,m\}} \tag{4}$$

s.t. (1), (2), and (3)

$$u^j_{\{i,m\}} \geq x_{ij} - x_{mj} \quad \forall \{i,m\} \in K',\ j \in I \tag{5}$$

$$u^j_{\{i,m\}} \geq x_{mj} - x_{ij} \quad \forall \{i,m\} \in K',\ j \in I. \tag{6}$$

Constraints (5) and (6) compute the vector u in terms of the vector x. For $i,m,j \in I$ such that $T_{\{i,m\}} > 0$ and $B_j > 0$, $u^j_{\{i,m\}}$ takes the minimum value implied, i.e., $u^j_{\{i,m\}} = \max\{x_{ij} - x_{mj}, x_{mj} - x_{ij}\} = |x_{ij} - x_{mj}|$. We do not need to impose integrality on $u^j_{\{i,m\}}$ as its integrality is implied by the integrality of the vector x. The objective function (4) is the sum of the cost of assignment and location and the cost of routing the traffic on the links between hubs and the central node.

Formulation *UHLP-S1* has $O(n^3)$ variables and $O(n^3)$ constraints.

Next, we give a second formulation of the UHLP-S. We define the traffic variable w_j as the total traffic between node $j \in I$ and the central node.

UHLP-S2

$$\min \quad \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{j\in I} B_j w_j \tag{7}$$

s.t. (1), (2), and (3)

$$w_j \geq \sum_{(i,m)\in S} T_{\{i,m\}}(x_{ij} - x_{mj}) \quad \forall S \subseteq K,\ j \in I. \tag{8}$$

Suppose we are given an assignment vector x. The traffic on the link between node j and the central node is $w_j = \sum_{\{i,m\}\in K'} T_{\{i,m\}}|x_{ij} - x_{mj}|$. For $j \in I$ such that $B_j > 0$, constraints (8) imply that $w_j = \max_{S\subseteq K} \sum_{(i,m)\in S} T_{\{i,m\}}(x_{ij} - x_{mj})$. A maximizing set is $S^* = \{(i,m) \in K : x_{ij} = 1, x_{mj} = 0\}$. Then $\sum_{\{i,m\}\in K'} T_{\{i,m\}}|x_{ij} - x_{mj}| = \sum_{(i,m)\in S^*} T_{\{i,m\}}(x_{ij} - x_{mj})$. So the constraint (8) defined by S^* computes the real value of w_j while the other constraints (8) are redundant.

Formulation *UHLP-S2* has $O(n^2)$ variables, but an exponential number of constraints.

To conclude this section, we compare the strength of LP relaxations of the two formulations. Let LP_1 and LP_2 be the optimal values of the linear programming (LP) relaxations of formulations *UHLP-S1* and *UHLP-S2*, respectively.

Proposition 1. $LP_1 = LP_2$.

Proof. Let F^1 and F^2 be the feasible sets of the LP relaxations of formulations *UHLP-S1* and *UHLP-S2*, respectively. Define F to be the set of triples (x,u,w) which satisfy inequalities (1), (2), (5), (6) and

$$w_j \geq \sum_{\{i,m\}\in K'} T_{\{i,m\}} u^j_{\{i,m\}} \quad \forall j \in I$$

$$0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in A.$$

We can easily show that $F^1 = \text{Proj}_{x,u}(F)$ and $F^2 = \text{Proj}_{x,w}(F)$. So, minimizing (4) on F yields LP_1 and minimizing (7) on F yields LP_2. As $B_j \geq 0$ for all $j \in I$, we have $LP_1 = LP_2$. ■

Proposition 1 says that both formulations *UHLP-S1* and *UHLP-S2* give the same LP bound. Now, the important question is whether this LP bound is a strong bound. The answer is given in the next two sections through polyhedral analysis and computational experiments.

3. POLYHEDRAL ANALYSIS

Note that as $x_{mj} \leq x_{jj}$ for all $m \in I \setminus \{j\}$, $u^j_{\{j,m\}} = |x_{jj} - x_{mj}| = x_{jj} - x_{mj}$. So we can remove the variables $u^j_{\{i,m\}}$ where $i = j$ or $m = j$ from the formulation by modifying the objective function. Define $U = \{(i,m,j) : i \in I, m \in I, j \in I, i < m, i \neq j, m \neq j\}$.

We can also project out the variables x_{jj}'s, i.e., we substitute $x_{jj} = 1 - \sum_{k\in I\setminus\{j\}} x_{jk}$ for all $j \in I$ (see Ref. [2]). Then the formulation *UHLP-S1* becomes

$$\begin{aligned}\min \quad & \sum_{j\in I} C_{jj}\left(1 - \sum_{i\in I\setminus\{j\}} x_{ji}\right) + \sum_{(i,j)\in A} C_{ij}x_{ij} \\ & + \sum_{j\in I} B_j \sum_{m\in I\setminus\{j\}} T_{\{j,m\}}\left(1 - \sum_{i\in I\setminus\{j\}} x_{ji} - x_{mj}\right) \\ & \qquad + \sum_{(i,m,j)\in U} B_j T_{\{i,m\}} u^j_{\{i,m\}}\end{aligned}$$

$$\text{s.t.} \quad x_{ij} + \sum_{m\in I\setminus\{j\}} x_{jm} \leq 1 \quad \forall (i,j) \in A \tag{9}$$

$$u^j_{\{i,m\}} \geq x_{ij} - x_{mj} \quad \forall (i,m,j) \in U \tag{10}$$

$$u^j_{\{i,m\}} \geq x_{mj} - x_{ij} \quad \forall (i,m,j) \in U \tag{11}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A. \tag{12}$$

Let $F_1 = \{(x,u) \in \{0,1\}^{n(n-1)} \times \mathbb{R}^{n(n-1)(n-2)/2} : (x,u)$ satisfies (9)–(11)$\}$ and $P_1 = \text{conv}(F_1)$.

Define $K_j = \{(i,m) \in K, i \neq j, m \neq j\}$ for $j \in I$. We redefine w_j to be the amount of traffic of commodities whose origins and destinations are different from j and which travel between node j and the central node. Then formulation *UHLP-S2* can be rewritten as

$$\min \quad \sum_{j\in I} C_{jj}\left(1-\sum_{i\in I\setminus\{j\}} x_{ji}\right)+\sum_{(i,j)\in A} C_{ij}x_{ij}$$

$$+\sum_{j\in I} B_j \sum_{m\in I\setminus\{j\}} T_{\{j,m\}}\left(1-\sum_{i\in I\setminus\{j\}} x_{ji}-x_{mj}\right)+\sum_{j\in I} B_j w_j$$

$$\text{s.t.} \quad (9) \text{ and } (12)$$

$$w_j \geq \sum_{(i,m)\in S} T_{\{i,m\}}(x_{ij}-x_{mj}) \quad \forall S\subseteq K_j, j\in I. \tag{13}$$

Let $F_2 = \{(x,w) \in \{0,1\}^{n(n-1)} \times \mathbb{R}^n : (x,w)$ satisfies (9) and (13)$\}$ and $P_2 = \text{conv}(F_2)$.

Let e_i be the ith unit vector in $\mathbb{R}^r$ for each $i = 1,\dots,r$. We will use the notation e_i to represent unit vectors of different dimensions r. Similarly, we will use 0 to represent a vector of all zeros, of varying dimensions. In all cases, the dimension intended will be clear from the context. In what follows, we will make particular use of vectors of the form $(e_{ij},0)$ and $(0,e_{imj})$ of the same dimensions as vectors in P_1 and $(e_{ij},0)$ and $(0,e_j)$ of the same dimension as vectors in P_2.

In the remaining part of this section, we study the properties of the polyhedra P_1 and P_2. Most of the results of this section are corollaries to the results in Appendix B.

We project the sets F_1, F_2, P_1, and P_2 on the x space. The result below is a corollary to Proposition 3 of Appendix B.

Corollary 1. $\text{Proj}_x(F_1) = \text{Proj}_x(F_2) = F_0 = \{x \in \{0,1\}^{n(n-1)} : x_{ij} + \sum_{m\in I\setminus\{j\}} x_{jm} \leq 1 \ \forall (i,j) \in A\}$ *and* $\text{Proj}_x(P_1) = \text{Proj}_x(P_2) = P_0 = \text{conv}(F_0)$.

The polytope P_0 is a special stable set polytope. It is full dimensional, i.e., $\dim(P_0) = n(n-1)$. Its polyhedral properties are studied in Labbé and Yaman [30].

We can derive the dimensions of P_1 and P_2 as a corollary to Theorem 6 of Appendix B.

Corollary 2. *The polyhedra P_1 and P_2 are full dimensional.*

Next, we give a characterization of the inequalities which involve only the assignment variables and which define facets of the polyhedra P_1 and P_2 as a corollary to Theorem 7 of Appendix B.

Corollary 3. *The inequality $\pi x \leq \pi_0$ defines a facet of P_1 and of P_2 if and only if it defines a facet of P_0.*

This corollary implies that the polyhedra P_1 and P_2 share some facet-defining inequalities and these are exactly the inequalities that define facets of the polytope P_0.

Labbé and Yaman [30] prove that the nonnegativity constraints $x_{ij} \geq 0$ and inequalities (9) define facets of P_0. Two immediate corollaries of these results and Corollary 3 are given below:

Corollary 4. *For $(i,j) \in A$, inequality $x_{ij} \geq 0$ defines a facet of P_1 and P_2.*

Corollary 5. *For $(i,j) \in A$, inequality (9) defines a facet of P_1 and P_2.*

Now we characterize facet-defining inequalities of P_1 which involve only the variables $u^j_{\{i,m\}}$'s.

Theorem 2. *No inequality of the form $\beta u \geq \beta_0$ defines a facet of P_1.*

Proof. Since $(0,0) \in P_1$, by Theorem 8, if inequality $\beta u \geq \beta_0$ is facet defining for P_1, then it is equivalent to $u^j_{\{i,m\}} \geq 0$ for some $(i,m,j) \in U$. But any $(x,u) \in P_1$ such that $u^j_{\{i,m\}} = 0$ also satisfies $x_{ij} = x_{mj}$. Thus inequality $u^j_{\{i,m\}} \geq 0$ cannot be facet-defining. ∎

We have a similar result for P_2.

Theorem 3. *No inequality of the form $\beta w \geq \beta_0$ defines a facet of P_2.*

Proof. Similar to the proof of Theorem 2. ∎

Theorem 2 (resp., Theorem 3) implies that a facet-defining inequality of P_1 (resp., P_2) either defines a facet of P_0 or it involves both variables x and u (resp., w). Below, we investigate facet-defining inequalities which involve both types of variables.

Theorem 4. *Inequalities* (10) *and* (11) *define facets of P_1 for $(i,m,j) \in U$.*

Proof. Define $P_f = \{(x,u) \in P_1 : u^j_{\{i,m\}} = x_{ij} - x_{mj}\}$. Assume that all $(x,u) \in P_f$ also satisfy $\beta u + \alpha x = \gamma$. As $(0,0) \in P_f$, $\gamma = 0$. Consider $(x,u) \in P_f$ and $(s,t,l) \in U \setminus \{(i,m,j)\}$. As $(x,u) + (0,e_{stl})$ is also in P_f, we have $\beta^l_{\{s,t\}} = 0$. For $(t,s) \in A$ such that $(t,s) \neq (i,j)$ and $(t,s) \neq (m,j)$, as both $(e_{ts}, \sum_{(l,k,v)\in U\setminus\{(i,m,j)\}} e_{lkv})$ and $(0, \sum_{(l,k,v)\in U\setminus\{(i,m,j)\}} e_{lkv})$ are in P_f, we have $\alpha_{ts} = 0$.

Since $p = (0, \sum_{(l,k,v)\in U\setminus\{(i,m,j)\}} e_{lkv})$ and $p + (e_{ij}, e_{imj})$ are in P_f, $\alpha_{ij} = -\beta^j_{\{i,m\}}$. Finally, $p + (e_{ij} + e_{mj}, 0)$ is in P_f yielding $\alpha_{mj} = -\alpha_{ij} = \beta^j_{\{i,m\}}$. So $\beta u + \alpha x = \gamma$ is a multiple of $u^j_{\{i,m\}} = x_{ij} - x_{mj}$.

The proof for inequality (11) can be done in a similar way. ∎

We now show that inequalities (13) define facets of P_2 under some conditions. For a given $j \in I$ and $S \subseteq K_j$, inequality (13) is called an *ordering inequality* if there exists an ordering σ on the nodes of $I \setminus \{j\}$ such that $S = \{(i,m) \in K_j, \sigma(i) < \sigma(m)\}$.

Theorem 5. *For a given $j \in I$ and $S \subseteq K_j$, inequality* (13) *defines a facet of P_2 if and only if it is an ordering inequality.*

Proof. Assume that for a given $j \in I$ and $S \subseteq K_j$, inequality (13) is an ordering inequality with ordering σ. Let N be a very large number. Below are n^2 affinely independent points in P_2 that satisfy inequality (13) at equality:

1. $(0, \sum_{m\in I\setminus\{j\}} Ne_m)$.
2. For $i \in I \setminus \{j\}$, $(0, \sum_{m\in I\setminus\{j\}} Ne_m + e_i)$.
3. For $i \in I \setminus \{j\}$, $(e_{ji}, \sum_{m\in I\setminus\{j\}} Ne_m)$.
4. For $i \in I \setminus \{j\}$, $(\sum_{m\in I:\sigma(m)\le\sigma(i)} e_{mj}, \sum_{m\in I\setminus\{j\}} Ne_m + \sum_{m\in I\setminus\{j\}:\sigma(m)\le\sigma(i)} \sum_{l\in I\setminus\{j\}:\sigma(l)>\sigma(i)} T_{ml}e_j)$.
5. For $(l,i) \in A$ such that $l \neq j$ and $i \neq j$, $(e_{ji} + e_{li}, \sum_{m\in I\setminus\{j\}} Ne_m)$.

So inequality (13) defines a facet of P_2.

Now we show that an inequality (13) does not define a facet of P_2 if it is not an ordering inequality. To do so, we show that separating the ordering inequalities is the same as separating inequalities (13).

We can separate inequalities (13) for a given $j \in I$ by taking $S = \{(i,m) \in K_j, x_{ij} - x_{mj} \ge 0\}$. If the corresponding inequality is violated, then it is one of the most violated inequalities (13) for j. Otherwise, there is no violated inequality (13) for j.

For a given $j \in I$, the order σ obtained by ordering the nodes in $I \setminus \{j\}$ in decreasing order of x_{ij} (ties are broken arbitrarily) leads to the set $S = \{(i,m) \in K_j, x_{ij} \ge x_{mj}\}$ which gives a most violated inequality (13). This implies that if there exists a violated inequality (13), then there exists also a violated ordering inequality. So the remaining inequalities (13) are not necessary to describe the polyhedron P_2 and they cannot be facet defining. ∎

The procedure to separate inequalities (13) given in the proof of Theorem 5 suggests the following:

Corollary 6. *Ordering inequalities can be separated in $O(n^3)$ time.*

The results of this section show that the constraints in both formulations are facet-defining inequalities. So we expect these formulations to be strong.

4. EXACT METHODS

In this section, we compare the two formulations from a computational point of view.

As formulation *UHLP-S2* has an exponential number of constraints, we develop a branch and cut algorithm to solve it. This algorithm is implemented in C++ using ABACUS 2.3 (see Ref. [26]) and uses the LP solver CPLEX 7.0. When we start the branch-and-cut algorithm, we do not include inequalities (8) in the formulation. We add these inequalities whenever we find them violated. The separation is done exactly as described in the proof of Theorem 5. When we cannot find any violated inequality (8), then we branch on the assignment constraints (1). Let (x^*, w^*) be the current fractional solution. We find the first node i for which we can find a subset $J \subset I$ such that $\sum_{j\in J} x^*_{ij}$ is close to 0.5. Then in one branch we fix $\sum_{j\in J} x_{ij}$ to 1, and in the other branch we fix $\sum_{j\in I\setminus J} x_{ij}$ to 1. We explore the branch and cut tree using best-first search.

We use the MIP Solver of CPLEX 8.1.0 to solve formulation *UHLP-S1*. We do not use ABACUS since we do not have cuts for *UHLP-S1*. We use the default values of CPLEX for parameters concerning optimality and feasibility tolerances, since initial tests showed that these do not influence significantly the computation times. We let the solver of CPLEX generate cuts.

The instances are generated using the AP data set of hub location problems from the OR Library (see Refs. [6, 17]). This dataset is often used to test solution methods for hub location problems (see e.g. Refs. [8, 16, 18]). It includes the coordinates of 200 districts and the amount of traffic to be routed between origin destination pairs.

We generated problems with 50–150 nodes. For each size, we consider different cost parameters. We define two parameters ϕ and γ which take values in $\{1, 3/4, 1/2, 1/4\}$ such that $\phi \le \gamma$. We multiply C_{jj} by ϕ and B_j by γ for all $j \in I$. As ϕ and γ decrease, more hubs are located. For each n, we have 10 problems with different ϕ and γ values. This lets us see which types of problems are harder to solve.

For the 150 node problem, values of $T_{\{i,m\}}$ differ in the range from 0.01 to 49.6844.

The runs are taken on an Intel Pentium III, 1 GHz, 1 GB RAM running under Suse 7.2. We compare the duality gap at the root node (i.e., gap $= 100\frac{opt-db}{opt}$ where *opt* is the optimal value and *db* is the lower bound before branching) and the CPU time in seconds for the two formulations. Note that despite Proposition 1, the two formulations can lead to different duality gaps as we let CPLEX generate cuts. We set the time limit to four hours. If a problem is not solved to optimality in four hours, we write *time* in the column CPU. We also report the total number of inequalities (8) added during the branch and cut algorithm in column *ineq*.

In Table 1, we report the results for problems with 50–100 nodes. For formulation *UHLP-S1*, if a problem is not solved in 4 h, we do not solve the problems with larger sizes for the same ϕ and γ values.

We observe that the duality gap is zero for all problems except one. This shows that both formulations are strong formulations. For the CPU time, the branch and cut algorithm for formulation *UHLP-S2* is much faster than CPLEX for *UHLP-S1*. Another disadvantage of formulation *UHLP-S1* is that, for unsolved problems CPLEX stops before solving the LP relaxation and does not report any bound.

TABLE 1. Comparison of formulations.

Problem		UHLP-S1		UHLP-S2			Problem		UHLP-S1		UHLP-S2		
ϕ,γ	n	gap	CPU	Gap	CPU	Ineq	ϕ,γ	n	Gap	CPU	Gap	CPU	ineq
1, 1	50	0.00	5,523	0.00	42	273	1/2, 1/2	50	0.00	2,198	0.00	28	441
1, 1	60		Time	0.00	112	370	1/2, 1/2	60	0.00	9,613	0.00	106	910
1, 1	70			0.00	309	660	1/2, 1/2	70		Time	0.00	229	1,196
1, 1	80			0.00	805	920	1/2, 1/2	80			0.00	409	1,218
1, 1	90			0.00	1,867	1,295	1/2, 1/2	90			0.00	763	1,461
1, 1	100			0.00	2,960	1,495	1/2, 1/2	100			0.00	989	1,444
3/4, 1	50	0.00	5,447	0.00	53	344	1/4, 1	50	0.00	5,286	0.00	97	707
3/4, 1	60		Time	0.00	137	506	1/4, 1	60		Time	0.00	303	1,143
3/4, 1	70			0.00	357	763	1/4, 1	70			0.00	721	1,516
3/4, 1	80			0.00	899	1,015	1/4, 1	80			0.00	2,020	2,167
3/4, 1	90			0.00	2,512	1,668	1/4, 1	90			0.00	4,191	2,660
3/4, 1	100			0.00	3,745	1,882	1/4, 1	100			0.00	10,786	4,606
3/4, 3/4	50	0.02	5,360	0.03	60	451	1/4, 3/4	50	0.00	3,352	0.00	50	576
3/4, 3/4	60		Time	0.00	156	708	1/4, 3/4	60		Time	0.00	185	992
3/4, 3/4	70			0.00	251	703	1/4, 3/4	70			0.00	676	1,881
3/4, 3/4	80			0.00	725	1,082	1/4, 3/4	80			0.00	1,446	2,173
3/4, 3/4	90			0.00	1,652	1,539	1/4, 3/4	90			0.00	2,030	2,074
3/4, 3/4	100			0.00	3,035	2,052	1/4, 3/4	100			0.00	3,533	2,642
1/2, 1	50	0.00	6,037	0.00	64	426	1/4, 1/2	50	0.00	1,147	0.00	29	580
1/2, 1	60		Time	0.00	191	704	1/4, 1/2	60	0.00	6,051	0.00	137	1,284
1/2, 1	70			0.00	532	1,009	1/4, 1/2	70		Time	0.00	203	1,176
1/2, 1	80			0.00	1,136	1,355	1/4, 1/2	80			0.00	577	1,932
1/2, 1	90			0.00	3,971	2,560	1/4, 1/2	90			0.00	826	1,787
1/2, 1	100			0.00	6,502	2,985	1/4, 1/2	100			0.00	1,145	1,821
1/2, 3/4	50	0.00	3,761	0.00	52	480	1/4, 1/4	50	0.00	83	0.00	9	307
1/2, 3/4	60		Time	0.00	164	823	1/4, 1/4	60	0.00	347	0.00	22	325
1/2, 3/4	70			0.00	317	981	1/4, 1/4	70	0.00	633	0.00	39	345
1/2, 3/4	80			0.00	860	1,412	1/4, 1/4	80	0.00	3,175	0.00	80	517
1/2, 3/4	90			0.00	1,874	1,741	1/4, 1/4	90	0.00	8,705	0.00	144	702
1/2, 3/4	100			0.00	2,969	1,993	1/4, 1/4	100	0.00	13,142	0.00	218	749

Note that formulation *UHLP-S1* has n^3 constraints. For $n = 50$, this makes 125 thousand constraints. Formulation *UHLP-S2* has n^2 constraints other than constraints (8). For 50 node problems, the largest number of inequalities (8) added during branch and cut is 707. So the LP relaxation with the largest number of constraints has 3207 constraints. For these problems, the LP relaxations solved during branch and cut have less number of variables and constraints compared to the LP relaxation of *UHLP-S1*.

In Table 2, we report the results for problems with 110–150 nodes using formulation *UHLP-S2*. The branch and cut algorithm solved 30 problems out of 50. For all problems, the algorithm stopped at the root node either at optimality or because of the time limit. In the latter case, it reported a lower bound. We use the upper bound given by the Lagrangian Relaxation heuristic (which is presented in the following section) to compute the final gap ($100\frac{ub-lb}{ub}$ where lb is the final lower bound reported by the branch and cut algorithm and ub is the best upper bound given by the heuristic.)

We observe that for a fixed ϕ, problems with high γ values are harder to solve. For example, for $\phi = 1/4$ and $n \geq 110$, none of the problems with $\gamma = 1$ is solved to optimality in 4 h, while the problems with $\gamma = 1/4$ are all solved in less than 40 min. Similarly, for fixed γ, problems with small ϕ values are harder to solve.

5. LAGRANGIAN RELAXATION HEURISTIC

The computational results show that for most problems the LP relaxation has an integer solution. So it is important to compute efficiently the lower bound of the LP relaxation. In this section, we present a Lagrangian relaxation which is as strong as the LP relaxations of formulations *UHLP-S1* and *UHLP-S2* and a heuristic method.

Lagrangian Relaxation is used often to solve location and design problems. For instance, Beasley [7] develops Lagrangian heuristics for location problems. Pirkul and Schilling [37] present a successful implementation of Lagrangian Relaxation for UHLP. Lee et al. [32] apply Lagrangian Relaxation to the problem of designing a network with a tree backbone and star access networks.

The UHLP-S can also be formulated as a quadratic mixed integer programming problem. We call this third formulation *UHLP-S3*.

UHLP-S3

$$\min \quad F(x) = \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{j\in I} B_j \sum_{i\in I}\sum_{m\in I} T_{\{i,m\}}x_{ij}(1 - x_{mj})$$

s.t. (1), (2), and (3).

TABLE 2. Larger instances with branch-and-cut.

Problem						Problem					
ϕ, γ	n	No. of LP's	No. of ineq.'s (8)	CPU	Final gap	ϕ, γ	n	No. of LP's	No. of ineq.'s (8)	CPU	Final gap
1, 1	110	105	1,643	4,395	0.00	1/2, 1/2	110	82	1,647	1,506	0.00
1, 1	120	108	1,937	7,195	0.00	1/2, 1/2	120	92	2,013	2,645	0.00
1, 1	130	122	2,429	13,579	0.00	1/2, 1/2	130	126	2,791	5,099	0.00
1, 1	140	84	2,141	Time	0.52	1/2, 1/2	140	123	3,102	7,461	0.00
1, 1	150	65	1,800	Time	1.78	1/2, 1/2	150	123	3,368	10,878	0.00
3/4, 1	110	116	2,085	5,589	0.00	1/4, 1	110	272	4,919	Time	0.01
3/4, 1	120	139	2,473	12,730	0.00	1/4, 1	120	159	3,811	Time	0.31
3/4, 1	130	119	2,639	Time	0.10	1/4, 1	130	99	2,974	Time	1.47
3/4, 1	140	80	2,118	Time	1.17	1/4, 1	140	72	2,442	Time	3.00
3/4, 1	150	65	1,880	Time	2.55	1/4, 1	150	58	2,180	Time	5.03
3/4, 3/4	110	131	2,441	4,526	0.00	1/4, 3/4	110	115	2,760	4,932	0.00
3/4, 3/4	120	131	2,550	6,583	0.00	1/4, 3/4	120	116	2,926	7,613	0.00
3/4, 3/4	130	131	2,822	11,068	0.00	1/4, 3/4	130	133	3,649	13,155	0.00
3/4, 3/4	140	103	2,816	Time	0.35	1/4, 3/4	140	94	3,222	Time	0.53
3/4, 3/4	150	74	2,288	Time	1.84	1/4, 3/4	150	71	2,789	Time	2.24
1/2, 1	110	183	3,172	8,573	0.00	1/4, 1/2	110	83	2,008	1,634	0.00
1/2, 1	120	210	3,835	Time	0.00	1/4, 1/2	120	95	2,478	2,954	0.00
1/2, 1	130	111	2,728	Time	0.72	1/4, 1/2	130	105	2,972	5,231	0.00
1/2, 1	140	76	2,202	Time	1.95	1/4, 1/2	140	124	3,569	8,682	0.00
1/2, 1	150	61	1,971	Time	3.58	1/4, 1/2	150	137	4,276	13,763	0.00
1/2, 3/4	110	110	2,224	4,247	0.00	1/4, 1/4	110	49	856	323	0.00
1/2, 3/4	120	116	2,493	6,795	0.00	1/4, 1/4	120	45	958	493	0.00
1/2, 3/4	130	120	2,937	11,239	0.00	1/4, 1/4	130	51	1,189	891	0.00
1/2, 3/4	140	101	2,937	Time	0.31	1/4, 1/4	140	68	1,563	1,380	0.00
1/2, 3/4	150	75	2,550	Time	1.72	1/4, 1/4	150	76	1,731	2,228	0.00

If we dualize constraints (1), we obtain:

$$LR(\lambda) = \min F(x) + \sum_{i\in I} \lambda_i \left(1 - \sum_{j\in I} x_{ij}\right)$$
$$\text{s.t.} \quad (2) \text{ and } (3).$$

Let $LD = \max_\lambda LR(\lambda)$.

Proposition 2. $LD = LP_1 = LP_2$.

Proof. We first present a linearization of $LR(\lambda)$:

$$\begin{aligned} LR(\lambda) = \min \sum_{i\in I}\sum_{j\in I} &\left(C_{ij} + B_j \sum_{m\in I} T_{\{i,m\}}\right) x_{ij} \\ &- \sum_{j\in I} B_j \sum_{(i,m)\in K} T_{\{i,m\}} v^j_{im} + \sum_{i\in I}\lambda_i\left(1 - \sum_{j\in I} x_{ij}\right) \end{aligned}$$

$$\text{s.t.} \quad (2) \text{ and } (3)$$

$$v^j_{im} \le x_{ij} \quad \forall (i,m) \in K,\ j \in I \tag{14}$$

$$v^j_{im} \le x_{mj} \quad \forall (i,m) \in K,\ j \in I \tag{15}$$

$$v^j_{im} \in \{0,1\} \quad \forall (i,m) \in K,\ j \in I. \tag{16}$$

Let X denote the set of of feasible solutions of the above linearization and D be the constraint matrix. The matrix D is totally unimodular as each row has two entries that sum to 0. It is known that (see e.g. Ref. [43])

$$\begin{aligned} LD = \min \sum_{i\in I}\sum_{j\in I} &\left(C_{ij} + B_j \sum_{m\in I} T_{\{i,m\}}\right) x_{ij} \\ &- \sum_{j\in I} B_j \sum_{(i,m)\in K} T_{\{i,m\}} v^j_{im} \end{aligned}$$

$$\text{s.t.} \quad (1) \text{ and } (x, v) \in \text{conv}(X).$$

As D is totally unimodular,

$$\begin{aligned} LD = \min \sum_{i\in I}\sum_{j\in I} &\left(C_{ij} + B_j \sum_{m\in I} T_{\{i,m\}}\right) x_{ij} \\ &- \sum_{j\in I} B_j \sum_{(i,m)\in K} T_{\{i,m\}} v^j_{im} \end{aligned}$$

$$\text{s.t.} \quad (1), (2), (14), (15)$$

$$0 \le x_{ij} \le 1 \quad \forall i \in I,\ j \in I \tag{17}$$

$$0 \le v^j_{im} \le 1 \quad \forall (i,m) \in K,\ j \in I. \tag{18}$$

For a given x which satisfies (1), (2), and (17), there is an optimal v which satisfies $v^j_{im} = \min\{x_{ij}, x_{mj}\}$ for all $(i,m) \in K$

and $j \in I$. For this (x, v), the objective value is

$$\begin{aligned}
&\sum_{i\in I}\sum_{j\in I}\left(C_{ij} + B_j \sum_{m\in I} T_{\{i,m\}}\right) x_{ij} \\
&\qquad\qquad - \sum_{j\in I} B_j \sum_{(i,m)\in K} T_{\{i,m\}} \min\{x_{ij}, x_{mj}\} \\
&= \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{j\in I} B_j \sum_{(i,m)\in K} T_{\{i,m\}}(x_{ij} - \min\{x_{ij}, x_{mj}\}) \\
&= \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{j\in I} B_j \sum_{(i,m)\in K} T_{\{i,m\}}(x_{ij} - x_{mj})^+ \\
&= \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{j\in I} B_j \sum_{(i,m)\in K'} T_{\{i,m\}}|x_{ij} - x_{mj}| \\
&= \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{j\in I} B_j \sum_{(i,m)\in K'} T_{\{i,m\}}u^j_{\{i,m\}}.
\end{aligned}$$

So $LD = LP_1$. ■

In the remaining part of this section, we present a heuristic based on this Lagrangian Relaxation. First we discuss how to compute LD. For a given λ, $LR(\lambda)$ can be computed by solving n independent problems, i.e.,

$$\begin{aligned}
LR(\lambda) = &\sum_{i\in I} \lambda_i \\
&+ \sum_{j\in I} \min\left\{C_{jj} + \sum_{m\in I\setminus\{j\}} B_j T_{\{j,m\}} - \lambda_j + LR_j(\lambda), 0\right\}
\end{aligned}$$

where

$$\begin{aligned}
LR_j(\lambda) = \min &\sum_{i\in I\setminus\{j\}} (C_{ij} - B_jT_{\{i,j\}} - \lambda_i)x_{ij} \\
&+ \sum_{i\in I\setminus\{j\}}\sum_{m\in I\setminus\{j\}} B_jT_{\{i,m\}}x_{ij}(1 - x_{mj}) \\
\text{s.t. } & x_{ij} \in \{0, 1\} \quad \forall i \in I \setminus \{j\}.
\end{aligned}$$

Notice that if $i \in I \setminus \{j\}$ is the only node assigned to node j, its contribution to the objective function is $C_{ij} - B_jT_{\{i,j\}} - \lambda_i + \sum_{m\in I\setminus\{j\}} B_jT_{\{i,m\}}$. Assigning more nodes to j cannot increase this contribution. So if this value is not positive, then there is an optimal solution where $x_{ij} = 1$.

If node $i \in I \setminus \{j\}$ is the only node not assigned to node j, then assigning it to j will cause an increase of $C_{ij} - B_jT_{\{i,j\}} - \lambda_i - \sum_{m\in I\setminus\{j\}} B_jT_{\{i,m\}}$ in the objective function. Changing the assignment of other nodes cannot decrease the value of this term. So if $C_{ij} - B_jT_{\{i,j\}} - \lambda_i - \sum_{m\in I\setminus\{j\}} B_jT_{\{i,m\}} \geq 0$, then there exists an optimal solution where $x_{ij} = 0$.

Let U_j^1 and U_j^0 be the sets of variables x_{ij}'s that are fixed to 1 and 0 respectively. Define $I_j = I \setminus (U_j^1 \cup U_j^0 \cup \{j\})$. Then the objective function becomes:

$$\begin{aligned}
&= \sum_{i\in U_j^1}(C_{ij} - B_jT_{\{i,j\}} - \lambda_i) + \sum_{i\in I_j}(C_{ij} - B_jT_{\{i,j\}} - \lambda_i)x_{ij} \\
&\quad + \sum_{i\in U_j^1}\sum_{m\in U_j^0} B_jT_{\{i,m\}} + \sum_{i\in I_j}\left(\sum_{m\in U_j^1} B_jT_{\{i,m\}}(1 - x_{ij}) \right. \\
&\qquad \left. + \sum_{m\in U_j^0} B_jT_{\{i,m\}}x_{ij}\right) + \sum_{i\in I_j}\sum_{m\in I_j} B_jT_{\{i,m\}}x_{ij}(1 - x_{mj}) \\
&= \sum_{i\in U_j^1}\left(C_{ij} - B_jT_{\{i,j\}} - \lambda_i + \sum_{m\in I\setminus(U_j^1\cup\{j\})} B_jT_{\{i,m\}}\right) \\
&\quad + \sum_{i\in I_j}\left(C_{ij} - B_jT_{\{i,j\}} - \lambda_i - \sum_{m\in U_j^1} B_jT_{\{i,m\}} + \sum_{m\in U_j^0} B_jT_{\{i,m\}}\right)x_{ij} \\
&\quad + \sum_{i\in I_j}\sum_{m\in I_j} B_jT_{\{i,m\}}x_{ij}(1 - x_{mj}).
\end{aligned}$$

The objective function has a fixed term and then the same structure as before but with different coefficients for linear terms. Let FC denote the fixed term of the objective function and C'_{ij} be the coefficient of x_{ij} for $i \in I_j$ in the linear term, i.e., $FC = \sum_{i\in U_j^1}(C_{ij} - B_jT_{\{i,j\}} - \lambda_i + \sum_{m\in I\setminus(U_j^1\cup\{j\})} B_jT_{\{i,m\}})$ and $C'_{ij} = C_{ij} - B_jT_{\{i,j\}} - \lambda_i - \sum_{m\in U_j^1} B_jT_{\{i,m\}} + \sum_{m\in U_j^0} B_jT_{\{i,m\}}$ for $i \in I_j$. Then

$$\begin{aligned}
LR_j(\lambda) = FC + \min &\sum_{i\in I_j} C'_{ij}x_{ij} + \sum_{i\in I_j}\sum_{m\in I_j} B_jT_{\{i,m\}}x_{ij}(1 - x_{mj}) \\
\text{s.t. } & x_{ij} \in \{0, 1\} \quad \forall i \in I_j.
\end{aligned}$$

The preprocessing algorithm (Algorithm 1) is based on these observations and it finds sets U_j^1 and U_j^0 and computes FC and C'_{ij} for $i \in I_j$.

The value $LR_j(\lambda)$ can then be computed by solving a min-cut problem on the graph $G_j = (I_j \cup \{o, d\}, A_j)$ where o and d are the dummy origin and destination nodes respectively and $A_j = \{(o, i), (i, d) : i \in I_j\} \cup \{(i, m) : i \in I_j, m \in I_j, i \neq m\}$ (see Ref. [42]) since $B_jT_{\{i,m\}} \geq 0$ for all $i, m, j \in I$. Let c_{im} denote the capacity of arc $(i, m) \in A_j$. The capacities are as follows:

$$\begin{aligned}
c_{im} &= B_jT_{\{i,m\}} \quad \text{for } i \in I_j, m \in I_j, i \neq m \\
c_{id} &= (C'_{ij})^+ \quad \text{for } i \in I_j \\
c_{oi} &= (-C'_{ij})^+ \quad \text{for } i \in I_j.
\end{aligned}$$

Algorithm 1 Preprocess

for all $i \in I \setminus \{j\}$ **do**
  $C'_{ij} \leftarrow C_{ij} - B_j T_{\{i,j\}} - \lambda_i$
$change \leftarrow 1$, $U_j^1 \leftarrow \emptyset$, $U_j^0 \leftarrow \emptyset$, $FC \leftarrow 0$
while *change* **do**
  $change \leftarrow 0$
  for all $i \in I \setminus (U_j^1 \cup U_j^0 \cup \{j\})$ **do**
    if $C'_{ij} + \sum_{m \in I \setminus (U_j^1 \cup U_j^0 \cup \{j\})} B_j T_{\{i,m\}} \le 0$ **then**
      $U_j^1 \leftarrow U_j^1 \cup \{i\}$
      $FC \leftarrow FC + C'_{ij} + \sum_{m \in I \setminus (U_j^1 \cup U_j^0 \cup \{j\})} B_j T_{\{i,m\}}$
      $C'_{mj} \leftarrow C'_{mj} - B_j T_{\{i,m\}}$ for $m \in I \setminus (U_j^1 \cup U_j^0 \cup \{j\})$
      $change \leftarrow 1$
    else if $C'_{ij} - \sum_{m \in I \setminus (U_j^1 \cup U_j^0 \cup \{j\})} B_j T_{\{i,m\}} \ge 0$ **then**
      $U_j^0 \leftarrow U_j^0 \cup \{i\}$
      $C'_{mj} \leftarrow C'_{mj} + B_j T_{\{i,m\}}$ for $m \in I \setminus (U_j^1 \cup U_j^0 \cup \{j\})$
      $change \leftarrow 1$

If the minimum cut separating nodes o and d is $(S \cup \{o\} : I_j \setminus S \cup \{d\})$, then the capacity of the cut is:

$$\begin{aligned} c(S) &= \sum_{i \in S} (C'_{ij})^+ + \sum_{i \in I_j \setminus S} (-C'_{ij})^+ + \sum_{i \in S} \sum_{i \in I_j \setminus S} B_j T_{\{i,m\}} \\ &= \sum_{i \in S} (C'_{ij})^+ + \sum_{i \in I_j} (-C'_{ij})^+ - \sum_{i \in S} (-C'_{ij})^+ \\ &\qquad + \sum_{i \in S} \sum_{i \in I_j \setminus S} B_j T_{\{i,m\}} \\ &= \sum_{i \in I_j} (-C'_{ij})^+ + \sum_{i \in S} (C'_{ij}) + \sum_{i \in S} \sum_{i \in I_j \setminus S} B_j T_{\{i,m\}}. \end{aligned}$$

So $LR_j(\lambda) = FC + c(S) - \sum_{i \in I_j} (-C'_{ij})^+$.

Hence $LR(\lambda)$ can be computed by solving n mincut problems. To compute LD we use the subgradient method.

The heuristic is given in Algorithm 2. The algorithm solves the Lagrangian Relaxations and generates feasible solutions using the optimal solutions of the relaxations. It stops either when the gap is no more than 0.0001% or when the lower bound does not improve.

Here, σ denotes the parameter that multiplies the stepsize, s denotes the stepsize and lb and ub denote the lower and upper bounds respectively. If at an iteration, the solution

TABLE 3. Lagrangian Relaxation heuristic.

Problem						Problem					
ϕ, γ	n	Final gap	No. of iters	CPU	%Imp. in CPU	ϕ, γ	n	Final gap	No. of iters	CPU	%Imp. in CPU
1, 1	50	0.00	107	24	42.86	1/2, 1/2	50	0.00	95	1	96.43
1, 1	60	0.00	37	10	91.07	1/2, 1/2	60	0.00	785	21	80.19
1, 1	70	0.00	83	78	74.76	1/2, 1/2	70	0.00	192	8	96.51
1, 1	80	0.00	179	394	51.06	1/2, 1/2	80	0.00	151	10	97.56
1, 1	90	0.00	143	505	72.95	1/2, 1/2	90	0.00	123	14	98.17
1, 1	100	0.00	117	616	79.19	1/2, 1/2	100	0.00	272	41	95.85
3/4, 1	50	0.00	74	13	75.47	1/4, 1	50	0.00	67	10	89.69
3/4, 1	60	0.00	48	20	85.40	1/4, 1	60	0.00	135	56	81.52
3/4, 1	70	0.00	138	124	65.27	1/4, 1	70	0.00	131	113	84.33
3/4, 1	80	0.00	109	194	78.42	1/4, 1	80	0.00	214	418	79.31
3/4, 1	90	0.00	161	717	71.46	1/4, 1	90	0.00	134	442	89.45
3/4, 1	100	0.00	159	967	74.18	1/4, 1	100	0.01	331	1,646	84.74
3/4, 3/4	50	0.03	302	31	48.33	1/4, 3/4	50	0.00	35	1	98.00
3/4, 3/4	60	0.00	515	102	34.62	1/4, 3/4	60	0.00	104	13	92.97
3/4, 3/4	70	0.00	70	13	94.82	1/4, 3/4	70	0.00	151	35	94.82
3/4, 3/4	80	0.00	119	84	88.41	1/4, 3/4	80	0.00	125	56	96.13
3/4, 3/4	90	0.00	72	63	96.19	1/4, 3/4	90	0.00	75	45	97.78
3/4, 3/4	100	0.00	264	516	83.00	1/4, 3/4	100	0.00	97	83	97.65
1/2, 1	50	0.00	66	12	81.25	1/4, 1/2	50	0.00	94	1	96.55
1/2, 1	60	0.00	105	47	75.39	1/4, 1/2	60	0.00	159	3	97.81
1/2, 1	70	0.00	231	226	57.52	1/4, 1/2	70	0.00	173	6	97.04
1/2, 1	80	0.00	150	296	73.94	1/4, 1/2	80	0.00	167	10	98.27
1/2, 1	90	0.00	246	898	77.39	1/4, 1/2	90	0.00	162	14	98.31
1/2, 1	100	0.00	147	849	86.94	1/4, 1/2	100	0.00	104	10	99.13
1/2, 3/4	50	0.00	50	3	94.23	1/4, 1/4	50	0.00	170	1	88.89
1/2, 3/4	60	0.00	203	26	84.15	1/4, 1/4	60	0.00	193	2	90.91
1/2, 3/4	70	0.00	105	31	90.22	1/4, 1/4	70	0.00	133	3	92.31
1/2, 3/4	80	0.00	110	58	93.26	1/4, 1/4	80	0.00	95	3	96.25
1/2, 3/4	90	0.00	65	55	97.07	1/4, 1/4	90	0.00	165	7	95.14
1/2, 3/4	100	0.00	111	134	95.49	1/4, 1/4	100	0.00	524	29	86.70

Algorithm 2 Lagrangian Relaxation Heuristic

$\sigma \leftarrow 2$, $\lambda \leftarrow 0$, $noimp \leftarrow 0$
$lb \leftarrow 0$ and $ub \leftarrow N$ where N is a large number
while $100\frac{ub-lb}{ub} > 10^{-4}$ and $\sigma > 10^{-4}$ **do**
  Compute $LR(\lambda)$ and let x be the optimal solution
  if x is feasible, i.e. satisfies (1) **then**
    STOP, optimal!
  else
    if $LR(\lambda) > lb$ **then**
      $lb \leftarrow LR(\lambda)$
      $noimp \leftarrow 0$
    else
      increment $noimp$
  if $\sum_{i\in I} x_{ii} \geq 1$ **then**
    ImproveSolution(x)
  if $noimp > 15$ **then**
    $noimp \leftarrow 0$
    $\sigma \leftarrow \sigma/2$
  $s \leftarrow \sigma \frac{ub-LR(\lambda)}{\sum_{i\in I}(1-\sum_{j\in I} x_{ij})^2}$
  $\lambda_i \leftarrow \lambda_i - s(1-\sum_{j\in I} x_{ij})$

is feasible for *UHLP-S3*, then the solution is optimal. Otherwise, we try to find a feasible solution using Algorithm 3. We update the Lagrange multipliers as $\lambda_i = \lambda_i - s(1-\sum_{j\in I} x_{ij})$ where $s = \sigma \frac{ub-LR(\lambda)}{\sum_{i\in I}(1-\sum_{j\in I} x_{ij})^2}$. The value *noimp* is the number of consecutive iterations where the lower bound does not improve. If *noimp* is more than 15, then we halve the parameter σ.

Algorithm 3 ImproveSolution(x)

$y \leftarrow 0$
for all $i \in I$ such that $x_{ii} = 1$ **do**
  $y_{ii} \leftarrow 1$
for all $i \in I$ such that $y_{ii} = 0$ **do**
  $I(i) \leftarrow \{j \in I : y_{jj} = 1, x_{ij} = 1\}$
  if $I(i) = \emptyset$ **then**
    $I(i) \leftarrow \{j \in I : y_{jj} = 1\}$
  $j' \leftarrow argmin_{j\in I(i)} C_{ij}$
  $y_{ij'} \leftarrow 1$
Compute the cost of y and update the upper bound and the best solution if necessary

The function *ImproveSolution*(x) constructs a feasible solution y using the actual solution of the Lagrangian Relaxation x if there is at least one hub open, i.e., if $\sum_{i\in I} x_{ii} \geq 1$. It keeps all open hubs of x. If node i is assigned to a single node j in x, then this assignment is also kept. Otherwise, if i is assigned to several nodes, then the algorithm picks the cheapest one in terms of costs C_{ij}. If node i is not assigned at all in x, then it is assigned to the hub with the cheapest C_{ij}.

In Table 3, we present the computational results for problems with 50–100 nodes. For each problem, we report the final gap ($100\frac{ub-lb}{ub}$ where lb and ub are the final lower and bounds given by the heuristic, respectively), the number of

TABLE 4. Larger instances with Lagrangian relaxation heuristic.

Problem						Problem					
ϕ, γ	n	Final gap	No. of iters	CPU	%Imp. in CPU	ϕ, γ	n	Final gap	No. of iters	CPU	%Imp. in CPU
1, 1	110	0.00	148	1,202	72.65	1/2, 1/2	110	0.00	304	59	96.08
1, 1	120	0.00	272	3,603	49.92	1/2, 1/2	120	0.00	248	57	97.84
1, 1	130	0.00	179	4,122	69.64	1/2, 1/2	130	0.00	155	44	99.14
1, 1	140	0.00	251	9,133	–	1/2, 1/2	140	0.00	350	140	98.12
1, 1	150	0.00	99	5,408	–	1/2, 1/2	150	0.00	304	171	98.43
3/4, 1	110	0.00	122	1,054	81.14	1/4, 1	110	0.01	404	3,136	–
3/4, 1	120	0.00	126	1,256	90.13	1/4, 1	120	0.01	376	4,104	–
3/4, 1	130	0.00	122	2,512	–	1/4, 1	130	0.01	728	14,577	–
3/4, 1	140	0.00	187	8,237	–	1/4, 1	140	0.00	383	12,976	–
3/4, 1	150	0.00	223	14,425	–	1/4, 1	150	0.00	182	7,161	–
3/4, 3/4	110	0.00	150	323	92.86	1/4, 3/4	110	0.00	174	190	96.15
3/4, 3/4	120	0.00	232	764	88.39	1/4, 3/4	120	0.00	96	164	97.85
3/4, 3/4	130	0.00	163	794	92.83	1/4, 3/4	130	0.00	119	297	97.74
3/4, 3/4	140	0.01	453	3,818	–	1/4, 3/4	140	0.00	90	310	–
3/4, 3/4	150	0.00	201	2,801	–	1/4, 3/4	150	0.00	112	763	–
1/2, 1	110	0.00	207	1,857	78.34	1/4, 1/2	110	0.00	144	20	98.78
1/2, 1	120	0.00	162	2,116	–	1/4, 1/2	120	0.00	142	25	99.15
1/2, 1	130	0.00	167	3,609	–	1/4, 1/2	130	0.00	174	41	99.22
1/2, 1	140	0.00	338	11,017	–	1/4, 1/2	140	0.00	212	66	99.24
1/2, 1	150	0.00	232	13,368	–	1/4, 1/2	150	0.00	207	82	99.40
1/2, 3/4	110	0.00	95	171	95.97	1/4, 1/4	110	0.00	270	19	94.12
1/2, 3/4	120	0.00	96	231	96.60	1/4, 1/4	120	0.00	239	22	95.54
1/2, 3/4	130	0.00	142	631	94.39	1/4, 1/4	130	0.00	330	39	95.62
1/2, 3/4	140	0.00	107	527	–	1/4, 1/4	140	0.00	295	43	96.88
1/2, 3/4	150	0.00	124	1,062	–	1/4, 1/4	150	0.00	217	39	98.25

iterations (the number of executions of the "while" loop in Algorithm 2), the CPU time in seconds and the percentage improvement in the CPU time compared to the branch and cut algorithm.

For two problems, the final gap is nonzero. For problem with $\phi = 3/4$, $\gamma = 3/4$, and $n = 50$, the final gap is 0.03% which is same as the gap of the LP relaxation of *UHLP-S2*. The heuristic finds an optimal solution but cannot prove it. For the problem with $\phi = 1/4$, $\gamma = 1$, and $n = 100$, the lower bound is the same as the optimal value but the heuristic cannot find an optimal solution and so the gap is between the optimal value and the upper bound.

For the remaining problems, the heuristic algorithm finds an optimal solution and proves optimality. It is very advantageous in terms of the CPU time compared to the branch-and-cut algorithm.

The results for problems with 110–150 nodes are given in Table 4. For four problems, the final gap is 0.01%. For the remaining problems, the algorithm proves optimality. There are two problems for which the heuristic took more than 4 h, still the max time is less than 4 h and 3 min.

In conclusion, the branch and cut algorithm for formulation *UHLP-S1* is able to solve problems of 100 nodes in a reasonable amount of time. For larger sizes, it can solve easy instances.

The Lagrangian Relaxation heuristic is rather efficient and gives very good solutions for instances up to 150 nodes. In 110 problems tested, the algorithm found an optimal solution and proved its optimality for 104 instances.

Acknowledgments

The authors thank anonymous referees for their valuable comments.

APPENDIX A: PROOF OF THEOREM 1

We show that the decision version of the UCLP is NP-complete by a reduction from the decision version of the UFLP. The decision version of the UFLP (DUFLP) is as follows: given sets M and N, vector F, matrix D, and a constant b, does there exist a solution $(\bar{x}, y)$ to UFLP with cost less than or equal to b? Similarly, the decision version of the UCLP (DUCLP) is: given set I, cost matrix C, and a constant b, does there exist a solution x to UCLP with cost less than or equal to b? The problem DUCLP is in NP since given a solution x we can verify in polynomial time that it has cost less than or equal to b.

Given an instance of the DUFLP, define $\bar{b} = b + 1$ and $I = N \cup M \cup \{o\}$ where node o is a dummy node. Let $C_{oj} = \bar{b}$ for all $j \in N \cup M$ and $C_{oo} = 0$. Set $C_{jo} = 0$ for each node j in M and set $C_{io} = \bar{b}$ for each node $i \in N$. Define $C_{jj} = F_j$ for all $j \in M$ and $C_{jl} = \bar{b}$ for all $j \in M$ and $l \in N \cup M$ such that $j \neq l$. Set $C_{il} = \bar{b}$ if $i \in N$ and $l \in N$ and set $C_{ij} = D_{ij}$ if $i \in N$ and $j \in M$. In Figure A1, we show this instance of the DUCLP (we removed the arcs with cost equal to $\bar{b}$).

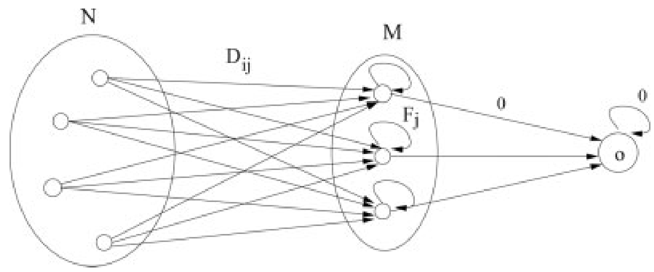

FIG. A1. Reduction from DUFLP to DUCLP.

Assume that the DUCLP has a solution x with cost less than or equal to b. Clearly, this solution does not use any arc with cost $\bar{b}$. So in x, node o is assigned to itself and each node in set M is either assigned to itself or it is assigned to node o. Moreover, each node in N is assigned to a node in M. Now define $(\bar{x}, y)$ as follows: Let $y_j = 1$ if $x_{jj} = 1$ for $j \in M$ and 0 otherwise and $\bar{x}_{ij} = 1$ if $x_{ij} = 1$ for $i \in N$ and $j \in M$ and 0 otherwise. The cost of such a solution is $\sum_{j \in M} F_j y_j + \sum_{i \in N} \sum_{j \in M} D_{ij}\bar{x}_{ij} = \sum_{i \in N \cup M \cup \{o\}} \sum_{j \in N \cup M \cup \{o\}} C_{ij}x_{ij}$ and is therefore less than or equal to b.

Equivalently, any solution $(\bar{x}, y)$ of the DUFLP can be transformed to a solution of DUCLP as follows: Set $x_{oo} = 1$, $x_{jj} = 1$ if $y_j = 1$ and $x_{jo} = 1$ otherwise for each $j \in M$ and set $x_{ij} = 1$ if $\bar{x}_{ij} = 1$ for each $i \in N$ and $j \in M$ and $x_{ij} = 0$ otherwise. It can be shown easily that the two solutions have the same cost. So we can conclude that there exists a solution $(\bar{x}, y)$ to DUFLP if and only if there exists a solution x to DUCLP of the same cost. Hence, DUCLP is NP-complete. ■

APPENDIX B: PROJECTION RESULTS

Projection has been a tool widely used in polyhedral analysis. One of the main concerns in this area has been the relationship between the dimension and facets of a polyhedron and the ones of its projection onto a subspace. Balas and Oosten [5] give necessary and sufficient conditions for a face of a polyhedron to project into a face of the projection of the polyhedron (see also Refs. [3, 4] for more results on projection).

Here we are interested in the projection of some specific family of polyhedra. Namely, we consider polyhedra P_C and P_I defined as follows: $P_C = \text{conv}(F_C)$ where

$$F_C = \{(x, z) \in \{0, 1\}^p \times \mathbb{R}^q : Ax \leq a, G_x x + g \leq G_z z\}$$

and $P_I = \text{conv}(F_I)$ where

$$F_I = \{(x, z) \in \{0, 1\}^p \times \mathbb{Z}^q : Ax \leq a, G_x x + g \leq G_z z\}.$$

We assume that matrix G_z has nonnegative entries and that every row and column of G_z has at least one positive entry. This implies that (x, z) where $x = 0$, $z = e_j$ for some $1 \leq j \leq q$ and e_j is the jth unit vector of size q is a ray of P_C and P_I.

We study the relationship between the polyhedra P_C, P_I and $P = \text{conv}(F)$ where $F = \{x \in \{0, 1\}^p : Ax \leq a\}$. Define

$\text{Proj}_x(F)$ to be the projection of the set F onto x space, i.e., $\text{Proj}_x(F) = \{x \in \{0,1\}^p : \exists\ (x,z) \in F\}$. We can easily show that:

Proposition 3. *$F = \text{Proj}_x(F_C) = \text{Proj}_x(F_I)$ and $P = \text{Proj}_x(P_C) = \text{Proj}_x(P_I)$.*

Next, we investigate how the dimensions of the three polyhedra are related.

Theorem 6. $\dim(P_C) = \dim(P_I) = \dim(P) + q$.

Proof. Assume that all points $(x,z) \in P_C$ satisfy an equality $\alpha x + \beta z = \gamma$. Choose $1 \le j \le q$ and $(x,z) \in P_C$. Consider (x',z') which is the same as (x,z) except that $z'_j = z_j + 1$. As both (x,z) and (x',z') are in P_C, we have $\alpha x + \beta z = \gamma$ and $\alpha x' + \beta z' = \gamma$. This implies that $\beta_j = 0$ for all $1 \le j \le q$.

Let $A^= x \le a^=$ be the system of inequalities that are satisfied at equality by all points (x,z) in P_C. As $P = \text{Proj}_x(P_C)$, they are also satisfied at equality by all points $x \in P$. So $\dim(P_C) = \dim(P) + q$.

The proof for P_I can be done similarly. ∎

The following theorem characterizes the facet-defining inequalities that are common to the three polyhedra.

Theorem 7. *The inequality $\alpha x \le \alpha_0$ defines a facet of P if and only if it defines a facet of P_C and of P_I.*

Proof. Clearly, the inequality $\alpha x \le \alpha_0$ is valid for P if and only if it is valid for P_C and P_I.

Let $F^\alpha = \{x \in P : \alpha x = \alpha_0\}$ and $F_C^\alpha = \{(x,z) \in P_C : \alpha x = \alpha_0\}$. By Proposition 3, we have $F^\alpha = \text{Proj}_x(F_C^\alpha)$. Let $r = \dim(P)$. Theorem 6 implies that $\dim(F^\alpha) = r - 1$ if and only if $\dim(F_C^\alpha) = r - 1 + q$.

The proof for P_I can be done similarly. ∎

This theorem gives a characterization of facet-defining inequalities of P_C and P_I which involve only the x variables in terms of the facet-defining inequalities of P.

Now consider $F_C^+ = \{(x,z) \in F_C : z \ge 0\}$ and $F_I^+ = \{(x,z) \in F_I : z \ge 0\}$.

Theorem 8. *If $(x,0) \in F_C^+$ (resp., F_I^+) for some $x \in F$ and if $\beta z \ge \beta_0$ defines a facet of $\text{conv}(F_C^+)$ (resp., $\text{conv}(F_I^+)$), then it is equivalent to $z_j \ge 0$ for some $j \in \{1,\ldots,q\}$.*

Proof. Assume that $(x,0) \in F_C^+$ for some $x \in F$ and that $\beta z \ge \beta_0$ defines a facet of $\text{conv}(F_C^+)$. Let $(x,z) \in F_C^+$ such that $\beta z = \beta_0$ and $j \in \{1,\ldots,q\}$. Since (x,z') where $z'_j = z_j + 1$ and $z'_l = z_l$ for $l \ne j$ is also in F_C^+, $\beta_j \ge 0$. As $z \ge 0$ and $(x,0) \in F_C^+$ for some $x \in F$, $\beta_0 = 0$. Then the inequality $\beta z \ge \beta_0$ should be equivalent to $z_j \ge 0$ for some $j \in \{1,\ldots,q\}$.

The proof for $\text{conv}(F_I^+)$ can be done similarly. ∎

REFERENCES

[1] K. Aardal, Capacitated facility location: Separation algorithms and computational experience, Math Program 81 (1988), 149–175.

[2] P. Avella and A. Sassano, On the p-median polytope, Math Program 89 (2001), 395–411.

[3] E. Balas, Projection with a minimal system of inequalities, Computat Optim Applic 10 (1998), 189–193.

[4] E. Balas, "Projection and lifting in combinatorial optimization", Computational Combinatorial Optimization, M. Jünger and D. Naddef (Editors), Springer, 2001.

[5] E. Balas and M. Oosten, On the dimension of the projected polyhedra, Discrete Appl Math 87 (1998), 1–9.

[6] E.J. Beasley, OR-Library: Distributing test problems by electronic mail, J Oper Res Soc 41 (1990), 1069–1072.

[7] E.J. Beasley, Lagrangean heuristics for location problems, Eur J Oper Res 65 (1993), 383–399.

[8] N. Boland, M. Krishnamoorthy, A.T. Ernst, and J. Ebery, Preprocessing and cutting for multiple allocation hub location problems, Eur J Oper Res 155 (2004), 638–653.

[9] T.B. Boffey, Location problems arising in computer networks, J Oper Res Soc 40 (1989), 347–354.

[10] J.F. Campbell, Integer programming formulations of discrete hub location problems, Eur J Oper Res 72 (1994), 387–405.

[11] J.F. Campbell, A.T. Ernst, and M. Krishnamoorthy, "Hub location problems", Facility Location: Applications and Theory, Z. Drezner and H.W. Hamacher (Editors), Springer, 2002, pp. 373–407.

[12] P. Chardaire, J.L. Lutton, and A. Sutter, Upper and lower bounds for the two-level simple plant location problem, Ann Oper Res 86 (1999), 117–140.

[13] S. Chung, Y. Myung, and D. Tcha, Optimal design of a distributed network with a two-level hierarchical structure, Eur J Oper Res 62 (1992), 105–115.

[14] G. Cornuéjols, G.L. Nemhauser, and L.A. Wolsey, "The uncapacitated facility location problem", Discrete Location Theory, P.B. Mirchandani and R.L. Francis (Editors), Wiley, New York, 1990, pp. 119–171.

[15] G. Cornuéjols, R. Sridharan, and J.M. Thizy, A comparison of heuristics and relaxations for the capacitated plant location problem, Eur J Oper Res 50 (1991), 280–297.

[16] J. Ebery, Solving large single allocation p-hub problems with two or three hubs, Eur J Oper Res 128 (2001), 447–458.

[17] A.T. Ernst and M. Krishnamoorthy, Efficient algorithms for the uncapacitated single allocation p-hub median problem, Loc Sci 4 (1996), 139–154.

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

[19] B. Gavish, Topological design of centralized computer networks: Formulations and algorithms, Networks 12 (1982), 355–377.

[20] E. Gourdin, M. Labbé, and H. Yaman, "Telecommunication and location", Facility Location: Applications and Theory, Z. Drezner and H.W. Hamacher (Editors), Springer, 2002, pp. 275–305.

[21] H.W. Hamacher, M. Labbé, S. Nickel, and T. Sonneborn, Adapting polyhedral properties from facility to hub location problems, Discrete Appl Math 145 (2004), 104–116.

[22] S.L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph, Oper Res 12 (1964), 450–459.

[23] S.L. Hakimi, Optimum distribution of switching centers in a communication network and some related graph theoretic problems, Oper Res 13 (1965), 462–475.

[24] J. Hardin, J. Lee, and J. Leung, On the Boolean-quadratic packing uncapacitated facility-location polytope, Ann Oper Res 83 (1998), 77–94.

[25] M.P. Helme and T.L. Magnanti, Designing satellite communication networks by zero-one quadratic programming, Networks 19 (1989), 427–450.

[26] M. Jünger and S. Thienel, The ABACUS System for branch-and-cut-and-price algorithms in integer programming and combinatorial optimization, Software Pract Exper 30 (2000), 1325–1352.

[27] J.G. Klincewicz, Hub location in backbone/tributary network design: A review, Loc Sci 6 (1998), 307–335.

[28] J. Krarup and P.M. Pruzan, The simple plant location problem: Survey and synthesis, Eur J Oper Res 12 (1983), 36–81.

[29] M. Labbé, D. Peeters, and J.F. Thisse, “Location on networks”, Network Routing, Handbooks in Operations Research and Management Sciences, M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser (Editors), Vol. 8, North-Holland, Amsterdam, 1995, pp. 551–624.

[30] M. Labbé and H. Yaman, Polyhedral analysis for concentrator location problems, Computational Optimization and Applications 34 (2006), 377–407.

[31] M. Labbé, H. Yaman, and E. Gourdin, A branch and cut algorithm for hub location problems with single assignment, Math Program 102 (2005), 371–405.

[32] Y. Lee, B.H. Lim, and J.S. Park, A hub location problem in designing digital data service networks: Lagrangian relaxation approach, Loc Sci 4 (1996), 185–194.

[33] A. Mirzaian, Lagrangian relaxation for the star–star concentrator location problem: Approximation algorithm and bounds, Networks 15 (1985), 1–20.

[34] M.E. O’Kelly, A quadratic integer program for the location of interacting hub facilities, Eur J Oper Res 32 (1987), 393–404.

[35] H. Pirkul, Efficient algorithms for the capacitated concentrator location problem, Comput Oper Res 14 (1987), 197–208.

[36] H. Pirkul and V. Nagarajan, Locating concentrators in centralized computer networks, Ann Oper Res 36 (1992), 247–262.

[37] H. Pirkul and D. Schilling, An efficient procedure for designing single allocation hub and spoke systems, Mgmt Sci 44 (1998), S235–S242.

[38] D. Skorin-Kapov, J. Skorin-Kapov, and M. O’Kelly, Tight linear programming relaxations of uncapacitated p-hub median problem, Eur J Oper Res 94 (1996), 582–593.

[39] J. Sohn and S. Park, A linear program for the two hub location problem, Eur J Oper Res 100 (1997), 617–622.

[40] J. Sohn and S. Park, The single allocation problem in the interacting three hub network, Networks 35 (2000), 17–25.

[41] R. Sridharan, The capacitated plant location problem, Eur J Oper Res 87 (1995), 203–213.

[42] J.C. Picard and H.D. Ratliff, Minimum cuts and related problems, Networks 5 (1975), 357–370.

[43] L.A. Wolsey, Integer Programming, Wiley, New York, 1998.

[44] H. Yaman, Concentrator Location in Telecommunications Networks, Springer, 2005.

[45] D. Yuan, “An annotated bibliography in communication network design and routing,” Optimization Models and Methods for Communication Network Design and Routing, Ph.D., Thesis, Department of Mathematics, Linköping University, Sweden, 2001.