A Branch-And-Cut Algorithm for the Partitioning-Hub Location-Routing Problem

F. Aykut Özsoy*, Daniele Catanzaro*, Martine Labbé*, Eric Gourdin†

May 20, 2008

Abstract

We consider the Partitioning-Hub-Location-Routing Problem (PHLRP), a hub location problem involving graph partitioning and routing features. PHLRP consists of partitioning a given network into sub-networks, locating at least one hub in each sub-network and routing the traffic within the network at minimum cost. This problem finds applications in deployment of an Internet Routing Protocol called Intermediate System - Intermediate System (ISIS), and strategic planning of LTL ground freight distribution systems. We describe an Integer Programming (IP) formulation for solving PHLRP. We also explore some valid inequalities for the IP formulation, which we take from the graph partitioning literature. We test effectiveness of the IP formulation and the valid inequalities. Our experiments show that the valid inequalities perform better than the XPRESS proprietary cuts.

Keywords: network design, telecommunication networks, hub-location, branch-and-cut.

1 Introduction

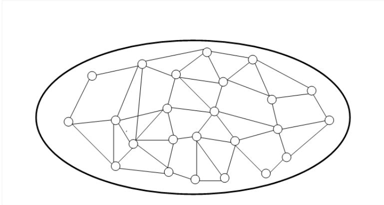
The problem of cost-effectively transferring a set of commodities (e.g., goods, people, data packages, etc.) between source-destination pairs in a network (e.g., a freight distribution network, a rail network, an Internet Protocol network, etc.) is common in many real life applications. Generally in such applications, some nodes of the network are designated as *hubs* (i.e., transshipment or switching points). These hub nodes induce a *backbone* of transfers throughout the network. The backbone, by aggregating the traffic flow corresponding to several source-destination pairs, facilitates exploitation of economies of scale on the transfer costs. Hub location problems, generally speaking, consist of finding in networks the hub locations (i.e., the backbone) that would lead to optimal transfer costs. There are many variations of the hub location problems. We refer the interested reader to two survey papers by Alumur and Kara [1], and by Campbell et al. [3].

In this paper we introduce yet another hub location problem, which involves "graph partitioning" and "routing" features. Namely, we address the problem of partitioning a given network into sub-networks, locating at least one hub in each sub-network and routing the traffic within the network at minimum cost. This problem finds applications in deployment of an Internet Routing Protocol called Intermediate System - Intermediate System (ISIS), and strategic planning of LTL ground (e.g., motor carriers, railways) freight distribution systems. We refer the reader to the appendix of [10] for a detailed description of the ISIS protocol, and to [11] for a detailed treatment of the technical specifications. For detailed introduction to LTL freight distribution systems, see [2] and [6].

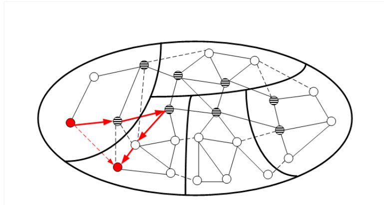
The article is organized as follows. In Section 2, we present a detailed description of the problem under consideration. In Sections 3 and 4 we introduce and explore an IP formulation for solving it. In Sections 5 we present some valid inequalities to strengthen our formulation. Finally, in Section 6 we report on numerical experiments.

*Graphes et Optimisation Mathématique (G.O.M.), D épartement d'Informatique, Université Libre de Bruxelles (U.L.B.), Boulevard du Triomphe, CP 210/01, B-1050, Brussels, Belgium. Phone: 0032 2 650 5628. Fax: 0032 2 650 5970.

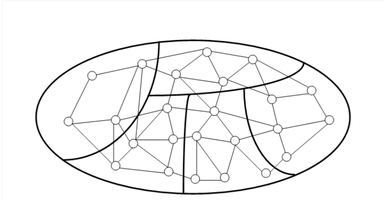
†France Télécom, R.&D. Department



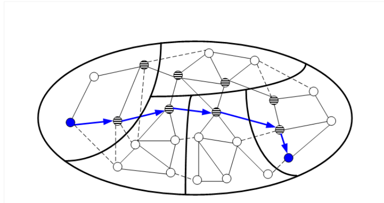
(a) A network example



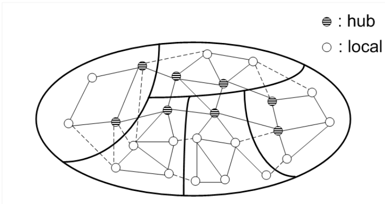
(d) An example of inter-area routing



(b) An example of partitioning of the network into areas



(e) An example of inter-area routing



(c) Hubs located on the network

Figure 1: Partitioning, hub location and routing components of the problem on an example

2 Notation and problem definition

Assume that we are given a network (a digraph) $D = (V, A)$ and a companion graph $G = (V, E)$. Assume that the arc set A is symmetric (i.e., $(j, i) \in A$ for all $(i, j) \in A$) and for each pair of arcs $(i, j), (j, i)$ in A, there exists an edge $\{i, j\} \in E$ with $i < j$.

Let $(V_1, V_2, \ldots, V_k)$ be a partitioning of V into subsets, i.e., $V_1, V_2, \ldots, V_k$ are disjoint subsets of V such that $V = \cup_l V_l$. We assume that $F_L \leq |V_l| \leq F_U$ for $l = 1, 2, \ldots, k$, where F_L and F_U are some lower and upper bound values on the sizes of subsets. Let $D_l = (V_l, A_l)$, $l = 1, 2, \ldots, k$, be sub-networks of D induced by $V_1, V_2, \ldots, V_k$, respectively (see Figure 1(b)). In accordance with the ISIS terminology, in the rest of this paper we refer to these sub-networks using the term *area*.

The use of the parameters F_L and F_U provides a control over the sizes and the number of areas we obtain in the partition. That is, we employ these parameters to protect ourselves against extreme cases in the solution, like very large areas or a very large number of small areas. In fact, partitioning the network into areas of manageable size provides a protection against stability and convergence issues in ISIS (see Chapter 3 of [11] for a detailed treatment of ISIS stability and convergence issues). Capacity or economies-of-scale concerns in freight distribution networks would also justify the use of such bounds on the area sizes.

We designate a subset H of nodes in V as *hub* nodes. We assume $|H| \leq Y$, that is, Y is an upper bound on the number of hubs we can locate in the network. We call the nodes in $V - H$ the *local* nodes (see Figure 1(c)). We call $D_H = (H, A_H)$, the sub-network of D induced by H, the *backbone*.

Let the set T contain all source-sink pairs which have traffic in-between. For each pair $(u, v) \in T$, we call node u the *source* and node v the *sink*. Let $d_{u,v}$ be the amount of required traffic flow between source-sink pair $(u, v) \in T$. A non-negative cost $c_{i,j}^{u,v}$ is incurred when unit flow associated with the pair $(u, v) \in T$ passes through the arc $(i, j) \in A$. The total flow on an arc (i, j) can not exceed a capacity value $C_{i,j}$.

We assume that routing of traffic between (u, v) pairs in T is in accordance with the routing in ISIS protocol (see Appendix). That is, routing in our problem complies with the following assumptions:

(A1) Inter-area traffic flows can be sent through an arc (i, j) only if (i, j) is on the backbone (i.e., $(i, j) \in A_H$).

(A2) Intra-area traffic is realized over paths which entirely lie within the areas.

(A3) Local nodes always use the same hub to send or receive inter-area flows.

The first assumption implies that two local nodes in different areas can not exchange traffic flows over an arc that lies in-between. Local nodes in different areas can send flows to each other only through the hubs located within their areas. We illustrate this in Figure 1(c) by displaying in dashed lines the inter-area arcs which are adjacent to local nodes (i.e., the dashed arcs in this Figure are not used at all for transfers). It is worth noting that one can also infer the following from assumptions (A1) and (A2):

- An area can exchange flows with the other areas only if it possesses a hub node (i.e., an area without a hub would be isolated from the rest of the network).
- Hubs of areas with traffic demand in-between have to be connected to each other over the backbone.
- If the source and the destination of a traffic flow are in different areas, then the flow is first directed towards the hub within the source area; subsequently, it is sent over the backbone towards the hub of the destination area; finally, it is directed from this hub towards the destination node.
- If the source and the destination are in the same area, then the flow is simply sent over a path that is entirely contained within the area.

Figures 1(d) and 1(e) display two examples of inter-area flows. Note that in Figure 1(d), the source and the destination are in different areas and although there exists an arc in-between (shown dashed), the flow is sent through the hubs.

Assumption (A3) has significance only if there are more than one hubs in an area. To see this, suppose that area D_l retains at least two hubs, i.e., $|V_l \cap H| \geq 2$, and pick a local node $u \in V_l - H$ in D_l. Then, assumption (A3) implies that u is assigned to one of the hubs in D_l, say $i \in V_l \cap H$, and always uses that hub i to send or receive inter-area flows.

Having set the grounds, we are now ready to give an informal definition of our problem. Given the network $D = (V, A)$, it consists of:

1. partitioning D into areas $D_1, D_2, \ldots, D_k$ of size at least F_L and at most F_U (i.e., partitioning V into subsets $V_1, V_2, \ldots, V_k$ such that $F_L \leq |V_l| \leq F_U$ for $l = 1, 2, \ldots, k$),
2. determining H (i.e., the hub nodes),
3. assigning every local router to a hub within its area (assumption (A3)),
4. routing the traffic for every $(u, v) \in T$ in accordance with the assumptions (A1), (A2), and capacity restrictions over the arcs,
5. with the objective of minimizing the total cost of routing.

In the sequel, we refer to this problem as *the Partitioning-Hub-Location-Routing Problem* ($PHLRP$).

PHLRP has a graph (or network) partitioning component, a hub location component and a routing component. We formulate the graph partitioning component (i.e., partitioning D into $D_1, D_2, \ldots, D_k$ of size at least F_L and at most F_U) as a *size constrained graph partitioning problem*, which is investigated by Özsoy and Labbe ([8], [9]). We model the routing component using multi-commodity flow variables and constraints (i.e., flow balance constraints, arc-capacity constraints).

The hub location component of PHLRP differs from the hub location problems explored in the literature. According to Alumur and Kara [1], hub location problems in the literature are generally based on three prevalent assumptions:

(B1) The hub network is complete with a link between every hub pair.

(B2) Cost of transfer between the hubs is less than the cost of transfer between non-hub nodes and hub nodes (due to economies of scale).

(B3) No direct transfer (between two non-hub nodes) is allowed.

On the other hand, we only keep assumption (B2) in the hub location component of PHLRP. We completely relax assumption (B1), and we adopt a weaker version of assumption (B3).

Generally (B1) does not hold in the application context we consider. In fact, nodes in ground transportation networks are connected to each other over paths that traverse several intermediate nodes. Similarly, in large Internet protocol networks (e.g. country wide large networks) setting up a physical link between every hub, especially between the ones that are far apart, might prove very costly. Furthermore, in logistics networks that involve many nodes or that span very large geographical areas, relaxing (B3) (i.e., permitting transfers between non-hub nodes) might decrease transfer costs significantly. The large distances between the non-hub nodes and the hubs in such networks would justify the use of inter-non-hub means for transfers. Such direct transfers are allowed in ISIS protocol, too. Assumptions (A1) and (A2) account for a weaker version of (B3) and impose that local transfers within the areas can be carried out directly (i.e., not necessarily through hubs).

Note that assumptions (A1)-(A3) of PHLRP are not restrictive at all for ISIS deployment applications and ground LTL freight distribution applications. On the contrary, the first two are necessary to ensure aggregation of inter-area flows on the backbone and to protect the backbone from the burden of intra-area flows.

Moreover, routing in ISIS naturally complies with (A3) because the nodes (which represent *routers*) know only the topology of their areas, i.e., they are blind beyond the borders of their areas. Hence, while sending out inter-area flows, a source router checks if the destination is in the same area as itself. If not, it just sends the flow to the closest hub, no matter what the destination is. In other words, while sending out inter-area flows, routers always minimize the local path within their areas, rather than the global path which extends outside the area towards the destination.

Assumption (A3) has a significant justification in the LTL freight distribution applications, too. In fact, aggregating the inter-area traffic corresponding to a node at a single hub is meaningful only if the fixed cost of dedicating a vehicle to a path is high. For instance, given a local node $u \in V_l - H$ in an area D_l, if we split between $i_1, i_2 \in V_l \cap H$ the corresponding inter-area traffic then we would have to dedicate one vehicle to each of the paths from u to i_1 and i_2. This might prove costly if the associated fixed cost is rather high, e.g., in rail networks. We do not explicitly account for fixed costs in this paper; however, assumption (A3) allows us to take into consideration, albeit implicitly, this relevant fixed cost of dedicating a vehicle to a line.

We shall present now in the next section an IP formulation for the PHLRP. Subsequently, we shall describe the exact algorithm we propose.

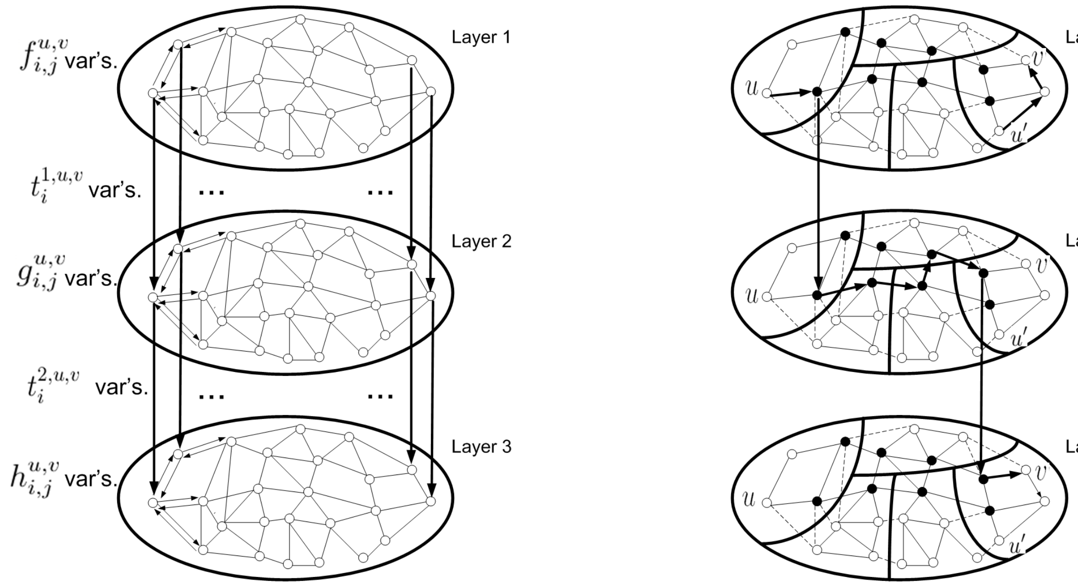
3 An IP Formulation for PHLRP

Our formulation is based on the idea of replicating the digraph $D = (V, A)$ three times (see Figure 2(a)). We refer to these replicates as *layers*. The sets of nodes and the arcs in the three layers are identical. In addition, we define costless and uncapacitated inter-layer arcs between identical nodes of the subsequent layers. Namely, there is an inter-layer arc from node i of Layer 1 (respectively, Layer 2) to node i of Layer 2 (respectively, Layer 3) for each i in V. Note that, unlike the arcs within the layers, the inter-layer arcs are all in one direction. It is only possible to send flow from Layer 1 to Layer 2 and from Layer 2 to Layer 3, but not in the opposite direction.

In our formulation, we impose that if the source and the sink nodes, say (u, v), lie in the same area, then the flow starts at u and ends at v of Layer 1. Otherwise, if u and v lie in different areas, then the flow starts at the source node u in Layer 1, passes through Layer 2 and ends at the sink node v of Layer 3 (see Figure 2(b)). We impose that no flow can pass through arcs that lie between nodes belonging to different areas. Moreover, we impose that an inter-area flow can pass from one layer to another only through an inter-layer arc corresponding to a hub.

We shall introduce now the variables of our IP formulation. We define the partitioning variables $w_{u,v} \in \{0, 1\}$, for all $u, v \in V$ such that $u \neq v$, as:

$$w_{u,v} = \begin{cases} 1, & \text{if } u \text{ and } v \text{ are in the same area;} \\ 0, & \text{otherwise.} \end{cases}$$



(a) The flow variables in the formulation. For the sake of simplicity in illustration, only some of the intra-layer and inter-layer arcs are displayed. Each of the undirected links within the layers stands for two arcs in opposite directions.

(b) Examples of intra-area and inter-area flows in the formulation.

Figure 2: Three replicates (i.e., Layers 1, 2 and 3) of the network D in formulation (3.1).

We define the hub-location variables $x_{u,v} \in \{0,1\}$, for all $u, v \in V$, as:

$$x_{u,v} = \begin{cases} 1, & \text{if node } u \text{ is assigned to the hub } v \text{ for exchanging flows} \\ & \text{with outside of its area;} \\ 0, & \text{otherwise.} \end{cases}$$

Setting $x_{u,u}$ to 1 or 0 implies designating node u as a hub or a local node, respectively.

We use the flow variables $f_{i,j}^{u,v}$, $g_{i,j}^{u,v}$, $h_{i,j}^{u,v} \in \{0,1\}$, for all $(u,v) \in T$ and $(i,j) \in A$, to represent the flows in Layers 1,2 and 3, respectively. Besides, we use the variables $t_i^{1,u,v}$, $t_i^{2,u,v} \in \{0,1\}$, for all $(u,v) \in T$ and $i \in V$, to represent the inter-layer flows from Layer 1 to Layer 2 and from Layer 2 to Layer 3, respectively. Specifically, we define $f_{i,j}^{u,v}$ as:

$$f_{i,j}^{u,v} = \begin{cases} 1, & \text{if the flow from } u \text{ to } v \text{ passes through the arc } (i,j) \text{ in Layer 1;} \\ 0, & \text{otherwise.} \end{cases}$$

This definition also applies to $g_{i,j}^{u,v}$ and $h_{i,j}^{u,v}$, if 'Layer 1' is replaced by the appropriate layer label. Similarly, we define $t_i^{1,u,v}$ as:

$$t_i^{1,u,v} = \begin{cases} 1, & \text{if the flow from } u \text{ to } v \text{ passes from Layer 1 to Layer 2 over node } i \\ & \text{(i.e., through the inter-layer arc } (i,i) \text{ between the two layers);} \\ 0, & \text{otherwise.} \end{cases}$$

This definition also applies to $t_i^{2,u,v}$ if the layer labels are changed accordingly.

Hereafter, we use the notation $\delta^+(i)$ and $\delta^-(i)$ to denote the outgoing and ingoing arcs of a node i in D, respectively, i.e., $\delta^+(i) = \{(l_1, l_2) \in A : l_1 = i\}$ and $\delta^-(i) = \{(l_1, l_2) \in A : l_2 = i\}$. The IP formulation can be stated as follows:

$$\min \sum_{(u,v)\in T} \sum_{(i,j)\in A} c^{u,v}_{i,j} d_{u,v}(f^{u,v}_{i,j} + g^{u,v}_{i,j} + h^{u,v}_{i,j}), \tag{3.1a}$$

$$w_{u,v} + w_{u,t} - w_{v,t} \leq 1 \qquad \forall u,v,t \in V : u \neq v, u \neq t, v < t, \tag{3.1b}$$

$$\sum_{v\in V-\{u\}} w_{u,v} \leq F_U - 1 \qquad \forall u \in V, \tag{3.1c}$$

$$\sum_{v\in V-\{u\}} w_{u,v} \geq F_L - 1 \qquad \forall u \in V, \tag{3.1d}$$

$$\sum_{v\in V} x_{u,v} = 1 \qquad \forall u \in V, \tag{3.1e}$$

$$x_{u,v} \leq x_{v,v} \qquad \forall u,v \in V : u \neq v, \tag{3.1f}$$

$$\sum_{u\in V} x_{u,u} \leq Y, \tag{3.1g}$$

$$x_{u,v} + x_{v,u} \leq w_{u,v} \qquad \forall u,v \in V : u < v, \tag{3.1h}$$

$$\sum_{(i,j)\in\delta^+(i)} f^{u,v}_{i,j} - \sum_{(i,j)\in\delta^-(i)} f^{u,v}_{i,j} + t^{1,u,v}_i = \begin{cases} 1 & \text{if } i = u \\ 0 & \text{if } i \in V - \{u,v\} \\ -w_{u,v} & \text{if } i = v \end{cases} \qquad \forall (u,v) \in T, \tag{3.1i}$$

$$\sum_{(i,j)\in\delta^+(i)} g^{u,v}_{i,j} - \sum_{(i,j)\in\delta^-(i)} g^{u,v}_{i,j} + t^{2,u,v}_i - t^{1,u,v}_i = 0, \qquad \forall i \in V, \forall (u,v) \in T, \tag{3.1j}$$

$$\sum_{(i,j)\in\delta^+(i)} h^{u,v}_{i,j} - \sum_{(i,j)\in\delta^-(i)} h^{u,v}_{i,j} - t^{2,u,v}_i = \begin{cases} w_{u,v} - 1 & \text{if } i = v \\ 0 & \text{if } i \in V - \{v\} \end{cases} \qquad \forall (u,v) \in T, \tag{3.1k}$$

$$t^{1,u,v}_i \leq x_{u,i} \qquad \forall (u,v) \in T, \forall i \in V, \tag{3.1l}$$

$$t^{2,u,v}_i \leq x_{v,i} \qquad \forall (u,v) \in T, \forall i \in V, \tag{3.1m}$$

$$\sum_{(i,j)\in\delta^+(i)} g^{u,v}_{i,j} \leq x_{i,i} \qquad \forall (u,v) \in T, \forall i \in V, \tag{3.1n}$$

$$f^{u,v}_{i,j} + f^{u,v}_{j,i} + h^{u,v}_{i,j} + h^{u,v}_{j,i} \leq w_{i,j} \qquad \forall (u,v) \in T, \forall (i,j) \in E, \tag{3.1o}$$

$$\sum_{(u,v)\in T} d_{u,v}\left(f^{u,v}_{i,j} + g^{u,v}_{i,j} + h^{u,v}_{i,j}\right) \leq C_{i,j} \qquad \forall (i,j) \in A. \tag{3.1p}$$

The objective function (3.1a) stands for the total cost of routing the traffic within the network. Valid partitioning of the network into areas is ensured by means of the constraints (3.1b), the so-called triangle inequalities. These constraints, first used by Grötschel and Wakabayashi [7] for the clique partitioning problem, guarantee that if two edges of a triangle lie *within* an area, the third edge lies within that same area, too. Constraints (3.1c) and (3.1d) impose upper and lower bounds, F_U and F_L respectively, on the sizes of the areas. Constraints (3.1e) ensure that each node is allocated to a hub to communicate with nodes outside its area. Note that when $x_{u,u} = 1$ for a node u, u cannot be allocated to a node other than itself (i.e., $x_{u,v} = 0$ for $v \in V \setminus \{u\}$); otherwise, when $x_{u,u} = 0$, u is necessarily allocated to a hub. Constraints (3.1f) prevents a local node from being assigned to a local node. Constraint (3.1g) imposes an upper bound Y on the number of hubs that can be located in the network. Finally, constraints (3.1h) ensure that each local node is assigned to a hub that lies within the same areas as itself.

Constraints (3.1i)-(3.1k) are the flow balance equations of Layers 1-3, respectively. These constraints ensure that for each source-sink pair $(u,v) \in T$:

- u in Layer 1 sends out one unit of flow, and
- that one unit of flow is absorbed by
  - v in Layer 1 if $w_{u,v} = 1$ (i.e., if u and v are in the same area),
  - v in Layer 3 if $w_{u,v} = 0$ (i.e., if u and v are in different areas).

Constraints (3.1l) and (3.1m) ensure that an inter-area flow from a source u to a sink v gets onto the backbone on the hub node to which the source is allocated and gets off the backbone on the hub node to which the sink is allocated. The flow from u to v is allowed to pass from Layer 1 to Layer 2 over node i only if i is a hub and the source u is allocated to i (i.e., only if $x_{u,i} = 1$). Similarly, the flow is allowed to pass from Layer 2 to Layer 3 over node i only if i is a hub and the sink v is allocated to i (i.e., $x_{v,i} = 1$). Constraints (3.1n) guarantee that the backbone is composed only of hubs. Specifically, in (3.1n), if a node is not a hub then it does not receive any flow from the other nodes in Layer 2. Constraints (3.1o) ensure that the flows in Layer 1 and Layer 3 do not cut across area borders. Finally, constraints (3.1p) stand for the capacity restrictions over the arcs.

It is worth noting that this formulation complies with the assumptions (A1)-(A3). In fact, constraints (3.1n) and (3.1o) represent assumption (A1); constraints (3.1i) and (3.1o) represent assumption (A2); finally, constraints (3.1e) stand for assumption (A3).

Note further that assumption (B2) of general hub location problems (i.e., transfers on the backbone costs less per unit flow), can also be accommodated by replacing the cost coefficient of $g_{i,j}^{u,v}$ variables in the objective function.

4 More on Formulation (3.1)

In this section, we show that some simplifications can be introduced in formulation (3.1) when capacity constraints (3.1p) are relaxed.

In absence of capacity restriction on arcs, partitioning the network into areas and locating the corresponding hubs are the main tasks to be performed for solving PHLRP. Once these tasks are accomplished (i.e., once D_1, $D_2, \ldots, D_k$ and D_H are determined), allocation and routing can be carried out in a simple way. In fact, when $D_1, D_2, \ldots, D_k$ and D_H are known, routing of the traffic for each $(u, v) \in T$ and allocation of local nodes to hubs reduces to solving a shortest path problem in the aforementioned three-layer network (see Figure 2(a)). This in turn suggests that in absence of capacity restriction on arcs, if variables $w_{u,v}$ and $x_{u,u}$ take on integral values in an optimal solution of the LP relaxation of the formulation, the flow variables $f_{i,j}^{u,v}$, $g_{i,j}^{u,v}$, $h_{i,j}^{u,v}$, $t_i^{1,u,v}$, $t_i^{2,u,v}$ and the allocation variables $x_{u,v}$, $u \neq v$, would take on integral values as well. In other words, if capacity restrictions are ignored, we can relax the integrality restrictions of the allocation variables (i.e., $x_{u,v}$ with $u \neq v$) and the flow variables (i.e., $f_{i,j}^{u,v}$, $g_{i,j}^{u,v}$, $h_{i,j}^{u,v}$, $t_l^{1,u,v}$ and $t_l^{2,u,v}$ for all $(u, v) \in T$, $(i, j) \in A$ and $l \in V$). Hence, only a small portion of variables in the formulation (i.e., only variables $w_{u,v}$ and $x_{u,u}$) need to be binary. This result is highlighted by the following lemma, which is taken from [10]. We refer the interested reader to [10] for a complete proof.

Lemma 4.1. *Let* $\mathbf{z} = \left(\mathbf{x}, \mathbf{w}, \mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{t^1}, \mathbf{t^2}\right)$ *and let* μ *denote the number of variables in the IP formulation introduced (i.e.,* $\mu = n^2 + \binom{n}{2} + 3|T||A| + 2n|T|$*). Further, let* $\mathcal{P}$ *denote the convex hull of feasible solutions of the formulation, i.e.,*

$$
\begin{aligned}
\mathcal{P} = conv\,\{\mathbf{z} \in \mathbb{R}^\mu |\, \mathbf{z} \ & \textit{satisfies } (3.1b) - (3.1o), \\
& w_{u,v} \in \{0,1\} && \forall u, v \in V : u < v \\
& x_{u,v} \in \{0,1\} && \forall u, v \in V \\
& f_{i,j}^{u,v}, g_{i,j}^{u,v}, h_{i,j}^{u,v} \in \{0,1\} && \forall (u,v) \in T, (i,j) \in A \\
& t_i^{1,u,v}, t_i^{2,u,v} \in \{0,1\} && \forall (u,v) \in T, i \in V\ \}.
\end{aligned}
$$

Then,

$$
\begin{aligned}
\mathcal{P} = conv\,\{\mathbf{z} \in \mathbb{R}^\mu |\, \mathbf{z} \ & \textit{satisfies } (3.1b) - (3.1o), \\
& w_{u,v} \in \{0,1\} && \forall u, v \in V : u < v \\
& x_{u,u} \in \{0,1\} && \forall u \in V \\
& x_{u,v} \in [0,1] && \forall u, v \in V : u \neq v \\
& f_{i,j}^{u,v}, g_{i,j}^{u,v}, h_{i,j}^{u,v} \in [0,1] && \forall (u,v) \in T, (i,j) \in A \\
& t_i^{1,u,v}, t_i^{2,u,v} \in [0,1] && \forall (u,v) \in T, i \in V\ \}.
\end{aligned}
$$

The absence of the capacity constraints (3.1p) also leads to a more compact formulation. In fact, constraints (3.1i) imply $\sum_{i \in V} t_i^{1,u,v} = 1 - w_{u,v}$ for all $(u, v) \in T$. In any optimal solution, integrality of the variables $t_i^{1,u,v}$ and $w_{u,v}$ implies, for each $(u, v) \in T$, the existence of at most one $i \in V$ such that $t_i^{1,u,v} = 1$. Hence, the right-hand-sides of constraints (3.1l) can be replaced by $x_{i,i}$, for all $i \in V$, i.e.,

$$t_i^{1,u,v} \leq x_{i,i} \qquad \forall (u,v) \in T, \forall i \in V. \tag{4.1a}$$

Similarly, we can also replace (3.1m) by

$$t_i^{2,u,v} \leq x_{i,i} \qquad \forall (u,v) \in T, \forall i \in V. \tag{4.1b}$$

In turn, constraints (3.1e), (3.1f) and (3.1h) become independent of the rest of the formulation and the corresponding allocation variables $x_{u,v}$, $u \neq v$, become redundant so that they can be removed. The following lemma from [10] highlights this result.

Lemma 4.2. *If capacity restrictions are ignored, formulation* (3.1) *can be rewritten using constraints* (3.1b)-(3.1d)*,* (3.1g)*,* (3.1i)-(3.1k)*,*(4.1a) *and* (4.1b)*.*

The following proposition is a direct consequence of Lemmas 4.1 and 4.2.

Proposition 4.1. *If capacity restrictions are ignored, formulation* (3.1) *reduces to the following:*

$$\begin{array}{rcll} min & & (3.1a) & \\ s.\ to & & (3.1b) - (3.1d), (3.1g), (3.1i) - (3.1k), (4.1a) \textit{ and } (4.1b), & \\ w_{u,v} & \in & \{0,1\} & \forall u,v \in V : u < v, \\ x_{u,u} & \in & \{0,1\} & \forall u \in V, \\ f_{i,j}^{u,v}, g_{i,j}^{u,v}, h_{i,j}^{u,v} & \in & [0,1] & \forall (u,v) \in T, (i,j) \in A, \\ t_i^{1,u,v}, t_i^{2,u,v} & \in & [0,1] & \forall (u,v) \in T, i \in V. \end{array}$$

5 Valid Inequalities

In this section we present some strong valid inequalities for the PHLRP. Özsoy and Labbé [9] show that these inequalities we present below are facet defining under some conditions for the the Size Constrained Graph Partitioning Polytope (SCGP), i.e., the polytope defined by inequalities (3.1b)-(3.1d). We refer the interested reader to [8] and [9] for a detailed treatment of the SCGP polytope and the corresponding inequalities; in the sequel we will just recall them.

As implied by Proposition 4.1, valid inequalities for SCGP are especially important for solving the uncapacitated version of PHLRP. Because the partitioning variables $w_{u,v}$ are the only binary variables remaining in the reduced form of formulation (3.1) given in Proposition 4.1.

Before proceeding with the inequalities, we need to introduce the notation we will be using hereafter in this section. We denote the complete graph defined over the vertex set V by $K_V = (V, E_V)$. Let $Q \subseteq V$ and $S \subseteq V$. We denote the set of all edges in K_V whose both endnodes are in Q by $E(Q)$, i.e., $E(Q) = \{\{i,j\} \in E_V : i,j \in Q\}$. The set of edges with one endnode in Q and the other endnode in S are denoted by $E(Q,S)$, i.e., $E(Q,S) = \{\{i,j\} \in E_V : i \in Q, j \in S\}$. For any $\alpha \in \mathbb{R}^{|E_V|}$ and $F \subseteq E_V$, we denote $\sum_{e \in F} \alpha_e$ by $\alpha(F)$.

The 2-partition inequalities Grötschel and Wakabayashi [7] first introduced the 2-partition inequalities for the Clique Partitioning Polytope (CPP), i.e., the polytope defined by (3.1b). Such inequalities are defined over two disjoint subsets S, T of V:

$$w(E(S,T)) - w(E(S)) - w(E(T)) \leq \min\{|S|, |T|\} \qquad \text{for } S,T \subset V : S \cap T = \emptyset. \tag{5.1}$$

Grötschel and Wakabayashi prove that the 2-partition inequalities are facet defining for the CPP if and only if $|S| \neq |T|$. Recently, Özsoy and Labbé [9] further investigated the 2-partition inequalities and stated the conditions for which they are facet defining for SCGP polytope.

The lower general clique inequalities Chopra and Rao [4] introduced a family of valid inequalities for the graph partitioning problem. Such inequalities can be adapted to SCGP polytope as follows. Let Q be a subset of V such that $|Q| > \left\lfloor \frac{|V|}{F_L} \right\rfloor$ and $\left(|Q| \text{mod} \left\lfloor \frac{|V|}{F_L} \right\rfloor\right) \neq 0$. Moreover, let k, r, p and q be defined as follows:

- $k = \left\lfloor \frac{|V|}{F_L} \right\rfloor$;
- $r = |V| \bmod F_L$, i.e., $|V| = kF_L + r$,

- $p = \left\lfloor \frac{|Q|}{k} \right\rfloor$ $(p \geq 1)$
- $q = (|Q| \bmod k)$ $(q > 0)$.

Then, the following inequality is valid and facet defining (see [9]) for the SCGP polytope:

$$w(E(Q)) \geq (k-q)\binom{p}{2} + q\binom{p+1}{2}. \tag{5.2}$$

The upper general clique inequalities Let $Q \subseteq V$ and consider the values ϕ_i^U computed as follows:

$$\phi_i^U = max\left\{\phi \in \mathbb{Z}_+ \mid F_L \leq \phi \leq F_U, \left\lfloor \frac{R_{i-1} - \phi}{F_L} \right\rfloor \geq \left\lceil \frac{R_{i-1} - \phi}{\phi} \right\rceil \right\},\ i = 1, \ldots, \left\lceil \frac{|V|}{F_U} \right\rceil$$

where $R_0 = |V|$ and $R_i = R_{i-1} - \phi_i^U$. Moreover, let k_Q and n_Q be defined as follows:

- $k_Q = \max\{i | \sum_{l=1}^{i} \phi_l^U \leq |Q|\}$,
- $n_Q = |Q| - \sum_{l=1}^{k_Q} \phi_l^U$.

Then, the following inequality is valid and facet defining (see [9]) for the SCGP polytope (see [8]):

$$w(E(Q)) \leq \sum_{l=1}^{k_Q} \binom{\phi_l^U}{2} + \binom{n_Q}{2} \qquad \forall Q \subseteq V. \tag{5.3}$$

Note that in the statement of this inequality, we adopt the convention that $\binom{0}{2} = \binom{1}{2} = 0$.

The cycle inequalities Conforti, Rao and Sassano [5] introduced the cycle inequalities for the equipartition polytope. Such inequalities can be stated as follows. Let $G_C = (V_C, E_C)$ be a cycle in G, then

$$w(E_C) \leq |V_C| - 2 \tag{5.4}$$

is valid for the equipartition polytope, and is facet defining when $|V|$ is odd and $|V_C| = \left\lceil \frac{|V|}{2} \right\rceil + 1$. Özsoy and Labbé [9] further investigated the cycle inequalities and stated the conditions for which they are facet defining for the SCGP polytope.

6 Numerical Results

In this section we analyze the performances of our model to tackle instances of PHLRP. Our experiments have been motivated by two goals: (i) to estimate the maximum size of PHLRP instances that can be optimally solved in a fixed amount of time, and (ii) to evaluate the benefits obtained from the inclusion of the valid inequalities previously discussed. We shall describe now the nature of the instances analyzed, the separation oracles used to implement the valid inequalities, and the results obtained.

6.1 Instances

We have carried out our experiments on a set of randomly generated PHLRP instances. We classify the instances by:

- the network size $|V|$, i.e., the number of nodes in the network;
- the edge density, i.e., $|E|$ divided by $\binom{|V|}{2}$;
- the demand density, i.e., the ratio $|T|/\binom{|V|}{2}$.

For sake of simplicity in the following we denote the edge density by Ed, and the demand density by Dd.

We have considered instances having $|V|$ equal to 9, 10, 12, 15, and 20 respectively. Moreover, we have further considered three possible values for Ed, $\{0.3, 0.6, 0.9\}$, and three possible values for Dd, $\{0.3, 0.6, 1.0\}$. Due to the randomness of the instance generation the Ed and Dd values of our instances generally may be quite close but not exactly equal to the above values.

For each possible assignment of the network size, the edge density, and the demand density, we have generated 10 PHLRP instances, leading to an overall number of 450 PHLRP instances downloadable at `http://homepages.ulb.ac.be/%7Efozsoy/`.

6.2 Computational experimentation details

We have implemented the model by means of Xpress-MP Mosel 2 using Xpress Optimizer 18 running on a Pentium 4, 3.2 GHz, equipped with 2 GByte RAM and operating system Gentoo release 7 (kernel linux 2.6.17). The running time was limited to a maximum of 3 hours.

PHLRP is a very difficult problem. For this reason, in our computational experimentation we considered only the un-capacitated version of the problem, solution of which is already very hard and time-consuming as will be seen in Section 6.3 and Tables 1, 2. We exploit the reduction implied by Proposition 4.1 and run our experiments on this reduced version of formulation (3.1).

Our experiments consists of two different implementations:

- a Basic Implementation (hereafter indicated as BI) in which the Xpress Optimizer proprietary cuts are activated and the valid inequalities (5.1)-(5.4) are deactivated;
- a Strengthened Implementation (hereafter indicated as SI) in which the Xpress Optimizer proprietary cuts are deactivated and the valid inequalities (5.1)-(5.4) are activated.

The valid inequalities previously discussed are $\mathcal{NP}$-hard problems, for this reason we have used heuristic oracles to separate them. We shall describe now each separation oracle in detail.

Separation oracle for the 2-partition inequalities The 2-partition inequalities are separated by means of a slightly modified version of the heuristic proposed by Grötschel and Wakabayashi [7] and Tcha *et al.* [12]. Specifically, for every node $v \in V$ we consider the set of nodes $W := \{v' \in V \setminus \{v\} \mid 0 < w(v, v') < 1\}$. After imposing a random ordering W_o of the set W, we pick the first node $v' \in W_o$, and we set $T := \{v'\}$. Subsequently, for every node $i \in W_o \setminus \{v'\}$ we set

$$T := T \cup \{i\} \qquad \text{if } w_{ij} = 0 \text{ for all } j \in T.$$

We check whether the set T constructed this way satisfies $w(E(T)) > 1$. If so we add the inequality $w(E(T)) \leq 1$.

Separation oracle for the lower general cliques inequalities The lower general clique inequalities are separated by means of a heuristic algorithm working as follows. Before starting the exact search we pre-compute the interval of feasible values for $|Q|$ i.e., the values of $|Q|$ such that $|Q| > \left\lfloor \frac{|V|}{F_L} \right\rfloor$ and $\left(|Q| \bmod \left\lfloor \frac{|V|}{F_L} \right\rfloor\right) \neq 0$. During the exact search we choose at random a subset Q of V having a feasible cardinality $|Q|$, and we check whether $w(E(Q)) < (k-q)\binom{p}{2} + q\binom{p+1}{2}$. If so we add the inequality $w(E(Q)) \geq (k-q)\binom{p}{2} + q\binom{p+1}{2}$. We iterate this procedure for all the feasible values $|Q|$.

Separation oracle for the upper general cliques inequalities The upper general clique inequalities are separated by means of a heuristic algorithm similar to the one used for the lower general cliques inequalities. Before starting the exact search we pre-compute the values $\{\phi_i^U\}$, $i = 1, \ldots, \lceil \frac{|V|}{F_U} \rceil$. During the exact search for all the values of $|Q| = 1, \ldots, \lfloor \frac{|V|}{2} \rfloor$ we choose at random a subset Q of $|Q|$ nodes of V and we check whether $w(E(Q)) > \sum_{l=1}^{k_Q} \binom{\phi_l^U}{2} + \binom{n_Q}{2}$. If so we add the inequality $w(E(Q)) \leq \sum_{l=1}^{k_Q} \binom{\phi_l^U}{2} + \binom{n_Q}{2}$.

Separation oracle for the cycle inequalities The cycle inequalities are separated by means of a heuristic algorithm working as follows. For every node $v \in V$ we set $V_c := \{v\}$, and we add the remaining nodes to V_c by picking at random $(\phi_1 - 1)$ nodes in $V \setminus \{v\}$. Subsequently, we check whether $w(E_c) > \phi_1 - 2$. If so we add the inequality $w(E_c) \leq \phi_1 - 2$.

6.3 Discussion

The performances obtained by both BI and SI are shown in Tables 1 and 2, respectively. Each row of the tables represents a set of 10 instances characterized by the given network size, edge density and demand density. The fourth, the fifth, and the sixth entries of each row evidence the average, the maximum, and the minimum of the solution times needed to solve the corresponding set of 10 instances, respectively. Similarly, the seventh, the eighth, and the ninth entries of each row evidence similar information about the gap (i.e., the difference between the Best Primal Bound (BPP) found and the value of linear relaxation at the root node of the search tree, divided by BPP) for the corresponding set of instances. Finally, the last three entries of each row evidence similar information about the number of nodes expanded in the search tree.

As general trend, SI is characterized by a solution time in average always smaller than BI, with a time performance that degrades by increasing the dimension of the instance analyzed $|V|$. This phenomenon is due to the fact that the generation of the Xpress proprietary cuts at each node of the search tree is in general more time consuming than the generation of the valid inequalities (5.1)-(5.4). On the contrary, as seen in the Nodes column of the Tables 1 and 2, the numbers of nodes expanded in the search trees of SI is much higher than BI. This shows that the Xpress proprietary cut generation and separation procedure is more thorough than the procedures we use to generate and separate the valid inequalities (5.1)-(5.4). One may suspect that, by increasing the size of the instances, the search tree of SI might end up so large that the solution time of SI could excess the solution time of BI. However, this seems not to be a real issue because in preliminary experiments no formulation was able to tackle instances larger than 25 nodes within one computing day; even in this case, SI performed better than BI.

As regard to the Gap columns of the Tables 1 and 2, the larger gap of SI is justified by the fact that the pre-solving strategies have been deactivated in SI. In fact, preliminary experiments showed that better performance can be achieved by SI when the pre-solving strategies is switched off. On the other hand, preliminary experiments showed that the pre-solving strategies enhance the performance of BI, for this reason it was switched on for BI.

A more detailed analysis shows that SI is able to solve instances of size 9 in at most 15 seconds with an average time of at most 5 seconds. On the contrary, BI takes four times more than SI to solve the most difficult instance of size 9 and in average up to five times more than SI to solve the same datasets. Similarly, SI is able to solve instances of size 10 in at most 95 seconds with an average time of at most 22 seconds. On the contrary, BI takes three times more than SI to solve the most difficult instance of size 10 and in average up to four times more than SI to solve the same dataset. SI solves instances of size 12 in at most 675 seconds with an average time of at most 107 seconds. On the contrary, BI takes 2.5 times more than SI to solve the most difficult instance of size 12 and in average up to 3.7 times more than SI to solve the same dataset. SI solves instances of size 15 in at most 5243 seconds with an average time of at most 805 seconds. On the contrary, BI is unable to tackle the most difficult instances of size 15 within the limit time and in average takes up to 4.7 times more than SI to solve the same dataset. Finally, both SI and BI are in general unable to solve instances of size 20 within the limit time (3 hours). Specifically, SI is able to solve the first and the third datasets in at most 6325 seconds, nevertheless the remaining datasets contain some instances which need more than three hours to be solved. Moreover, none of the instances belonging to the last datasets were solved within the limit time (see Table 4). On the contrary, BI is characterized by a higher percentage of unsolved instances. Specifically, the percentage of unsolved instances approaches 0 when analyzing datasets having edge density larger than 0.6 (see Table 3).

7 Conclusion

In this paper we consider the Partitioning-Hub-Location-Routing Problem (PHLRP), which is a hub location problem with graph partitioning and routing features. We introduce an IP formulation for solving it, and investigate further characteristics of this formulation for the un-capacitated case. We also test effectiveness

BI

$\lvert V \rvert$	Ed	Dd	Time (sec.)			Gap (%)			Nodes		
			Average	Max	Min	Average	Max	Min	Average	Max	Min
9	0.3	0.3	1.178	3.21	0.13	6.183	18.5888	0	4.900	17	0
9	0.3	0.6	1.376	5.66	0.06	1.206	6.36361	0	1.300	3	0
9	0.3	1	4.844	19.1	0.08	3.597	11.2566	0	3.200	13	0
9	0.6	0.3	3.127	7.68	0.17	4.938	14.9183	0	8.200	27	1
9	0.6	0.6	8.693	20.83	1.43	10.541	21.9765	1.65202	19.600	95	1
9	0.6	1	21.903	47.03	2.31	9.617	23.519	0.888299	28.600	97	1
9	0.9	0.3	5.052	14.04	0.9	8.705	15.198	0.782532	18.600	59	1
9	0.9	0.6	9.772	17.62	0.45	7.618	14.8799	0	14.400	39	1
9	0.9	1	27.005	61.75	2.95	7.248	13.3142	1.35227	21.800	91	1
10	0.3	0.3	4.082	9.2	0.31	7.516	23.4174	0	24.600	71	1
10	0.3	0.6	10.342	16.9	1.61	10.909	30.478	0	35.600	85	1
10	0.3	1	32.161	130.05	0.11	8.376	19.9038	0	43.100	213	0
10	0.6	0.3	7.826	11.86	3.07	6.943	13.4231	3.06962	31.200	75	5
10	0.6	0.6	19.740	52.32	6.97	8.234	19.8715	2.71859	47.000	185	9
10	0.6	1	81.607	150.68	25.25	11.308	16.8102	6.18679	97.800	211	13
10	0.9	0.3	13.998	37.47	2.98	8.104	15.3023	0.141476	43.400	137	7
10	0.9	0.6	34.675	76.81	11.86	9.515	16.1959	2.55618	79.400	227	5
10	0.9	1	91.252	280.81	15.02	9.905	17.8419	1.00561	94.200	389	1
12	0.3	0.3	34.303	93.73	2.51	7.420	13.7804	0.381655	89.600	287	1
12	0.3	0.6	56.244	125.91	19.22	9.037	19.0824	4.88025	56.600	141	7
12	0.3	1	163.730	774.4	34.94	7.748	15.421	0.181952	72.400	337	5
12	0.6	0.3	28.501	107.53	5.29	5.875	14.5091	1.25651	87.000	471	5
12	0.6	0.6	143.520	351.98	43.65	9.431	16.3633	3.04573	136.200	455	9
12	0.6	1	379.523	819.98	63.55	8.147	13.9666	1.91166	139.900	337	15
12	0.9	0.3	59.841	320.35	13.32	7.808	15.3416	1.74104	136.800	739	23
12	0.9	0.6	166.590	324.77	20.21	8.193	14.8881	1.84757	108.400	207	3
12	0.9	1	398.894	1693.35	32.42	7.237	14.8154	0.794988	107.400	579	1
15	0.3	0.3	133.914	397.32	31.59	5.486	9.27182	1.78006	152.400	469	11
15	0.3	0.6	1044.506	2387.84	108.46	7.875	12.413	2.72327	356.000	967	19
15	0.3	1	2382.889	5337.89	501.25	8.504	11.3151	3.89735	297.800	631	59
15	0.6	0.3	193.853	427.44	78.12	6.256	10.9274	2.27197	205.400	539	21
15	0.6	0.6	569.994	930.78	120.43	5.556	8.99907	1.8603	166.400	379	37
15	0.6	1	2641.774	10835.3	188.56	5.216	9.24387	2.01189	260.700	1176	9
15	0.9	0.3	351.339	749.08	37.53	7.263	9.63293	1.46339	320.800	631	31
15	0.9	0.6	2634.632	10819	260.95	7.469	13.6094	1.15943	584.900	2567	39
15	0.9	1	3801.558	10136.8	273.24	6.848	12.0653	2.14586	328.200	799	9
20	0.3	0.3	5153.653	10832.3	74.66	5.042	8.50953	1.18672	699.400	1457	7
20	0.3	0.6	9911.467	10935.5	3942.22	6.323	14.7954	1.57351	260.000	415	109
20	0.3	1	10703.338	11104.7	7632.17	9.649	32.7458	2.61005	83.100	108	57
20	0.6	0.3	5743.235	10849.3	2408.75	5.912	11.2135	1.33713	528.600	1141	227
20	0.6	0.6	10064.342	10979.3	2331.82	4.834	13.5294	0.638326	203.100	288	67
20	0.6	1	10882.537	11192.7	8718.96	12.283	29.6195	3.43702	74.400	96	56
20	0.9	0.3	8189.168	10841	1535.67	4.826	11.6135	0.348171	718.400	1090	105
20	0.9	0.6	10946.990	10976.4	10912	8.646	23.2627	0.395442	227.600	350	158
20	0.9	1	11185.799	11307.8	11054.9	9.382	22.6831	0.597229	74.800	135	51

Table 1: Results obtained by Xpress Optimizer with proprietary cut-strategy.

SI

$\lvert V\rvert$	Ed	Dd	Time (sec.)			Gap (%)			Nodes		
			Average	Max	Min	Average	Max	Min	Average	Max	Min
9	0.3	0.3	0.259	0.55	0.05	7.175	19.2707	0	5.600	13	1
9	0.3	0.6	0.444	1	0.02	2.826	11.3032	0	3.200	9	0
9	0.3	1	0.929	3.3	0.04	5.663	15.6252	0	8.000	33	0
9	0.6	0.3	0.755	1.99	0.09	5.488	16.6148	0	10.800	29	1
9	0.6	0.6	2.012	3.99	0.7	12.293	24.1318	3.50706	30.400	77	5
9	0.6	1	4.911	11.13	0.89	11.765	25.5876	2.11634	41.200	115	1
9	0.9	0.3	1.472	4.42	0.49	8.943	15.8677	1.33777	27.000	79	5
9	0.9	0.6	2.115	4.12	0.18	8.970	15.9507	0	20.000	51	1
9	0.9	1	5.134	14.68	1.93	9.833	16.7337	3.96084	31.800	123	3
10	0.3	0.3	1.332	3.7	0.09	7.622	23.8011	0	43.700	143	1
10	0.3	0.6	2.350	7.48	0.45	11.472	31.3654	0	42.800	169	1
10	0.3	1	8.327	44.65	0.06	10.052	21.7457	0	73.700	387	0
10	0.6	0.3	1.974	4.01	0.71	7.246	14.5422	3.33255	35.800	81	1
10	0.6	0.6	5.098	12.91	1.71	8.773	20.4081	3.11298	70.000	217	15
10	0.6	1	17.739	39.42	4.35	12.778	18.0462	7.20175	132.000	285	23
10	0.9	0.3	3.214	8.52	0.8	8.545	15.8594	0.141476	47.800	177	5
10	0.9	0.6	10.901	32.88	2.02	10.418	17.8768	3.12856	125.200	455	7
10	0.9	1	21.961	94.94	2.34	11.157	19.076	1.34333	142.200	647	1
12	0.3	0.3	8.574	22.66	0.82	7.941	14.7622	0.753457	129.200	403	1
12	0.3	0.6	13.150	29.72	2.99	10.791	21.5661	5.41576	124.800	349	11
12	0.3	1	28.762	179.67	1.43	9.455	19.6357	0	116.600	799	1
12	0.6	0.3	12.136	49.99	1.63	6.176	14.7709	1.42392	216.000	931	9
12	0.6	0.6	49.141	127.48	7.84	11.283	19.4915	3.77272	355.200	835	35
12	0.6	1	86.184	234.08	8.31	9.892	17.029	2.30431	237.600	745	7
12	0.9	0.3	17.528	93.44	2.02	8.097	15.414	1.78825	230.400	1111	19
12	0.9	0.6	32.798	61.3	5.32	9.380	15.7852	2.17027	158.000	353	3
12	0.9	1	106.726	674.53	3.62	8.807	17.9588	0.0320767	260.400	1787	1
15	0.3	0.3	23.443	52.84	4.72	5.042	9.48492	1.57925	143.000	339	15
15	0.3	0.6	140.459	384.32	18.09	7.500	11.4465	2.69284	409.200	1483	43
15	0.3	1	406.246	1390	47.83	9.566	14.3973	4.17014	525.300	1635	55
15	0.6	0.3	43.929	108.49	18.15	5.519	10.0754	2.51041	270.600	737	63
15	0.6	0.6	150.432	346.81	24.71	5.965	8.96121	2.40958	369.800	1059	37
15	0.6	1	585.383	3035.98	56.95	7.999	13.4062	2.51279	489.400	2369	39
15	0.9	0.3	112.585	248.24	11.16	6.926	9.08745	0.608663	569.900	1033	23
15	0.9	0.6	804.534	5242.47	46.31	9.179	14.7264	3.45587	1279.300	8538	43
15	0.9	1	706.995	1635.89	98.81	8.301	12.5914	3.3096	493.200	1300	35
20	0.3	0.3	1391.972	4904.51	33.02	8.977	13.3628	1.40949	1369.200	4010	29
20	0.3	0.6	8645.743	10822	756.09	5.704	11.7623	0.542981	2224.000	3552	463
20	0.3	1	7372.315	10845.5	1908.89	3.143	7.27825	0.001532	888.200	1564	191
20	0.6	0.3	2378.024	6324.98	559.9	10.558	15.8532	7.23804	1605.600	4087	275
20	0.6	0.6	7801.546	10833.6	410.37	3.102	7.15811	0.706479	1567.000	2370	143
20	0.6	1	10533.928	10905.6	8610.32	6.914	11.3189	3.44158	755.375	1067	517
20	0.9	0.3	6972.569	10816.4	698.53	5.050	12.1208	0.137965	2029.778	3548	281
20	0.9	0.6	10440.939	10868.2	8241.82	5.035	17.4275	0.250289	1011.111	1509	759
20	0.9	1	10932.791	10947.1	10905.3	7.386	25.2158	0.159251	510.500	639	411

Table 2: Results obtained by Xpress Optimizer using cuts (5.1)-(5.4) and deactivating proprietary cut strategy.

BI	Ed		
Dd	0.3	0.6	0.9
0.3	7/10	2/10	1/10
0.6	7/10	1/10	1/10
1.0	4/10	0/10	0/10

Table 3: Number of instances of size 20 solved by BI in function of the edge density Ed and demand density Dd.

SI	Ed		
Dd	0.3	0.6	0.9
0.3	10/10	2/10	4/10
0.6	10/10	3/10	2/10
1.0	4/10	2/10	0/10

Table 4: Number of instances of size 20 solved by SI in function of the edge density Ed and demand density Dd.

of some valid inequalities, which are already investigated by Özsoy and Labbe [9] for *the size constrained graph partitioning polytope*, for solving the problem in a branch-and-cut algorithm.

We report on our computational experiments in XPRESS using our IP formulation. Our experiments show that the valid inequalities we consider perform better than the XPRESS proprietary cuts in solving PHLRP instances. However, so far, the IP model we present is unable to tackle instances larger than 20 nodes within 3 hours. The analysis of real life instances deserves further research effort.

Acknowledgements

The first author acknowledges support from France Télécom (Contract number 46126988) and TUBITAK (The Scientific and Technological Research Council of Turkey). The second author acknowledges support from the Belgian National Fund for Scientific Research (F.N.R.S.), of which he is a Research Fellow, and from Communauté Française de Belgique — Actions de Recherche Concertées (ARC).

References

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

[2] J. F. Campbell. Strategic network design for motor carriers. In A. Langevin and D. Riopel, editors, *Logistics Systems: Design and Optimization*. Springer, U.S.A., 2005.

[3] J. F. Campbell, A. T. Ernst, and M. Krishnamoorthy. Hub location problems. In Z. Drezner and H. Hamacher, editors, *Facility Location: Applications and Theory*. Springer-Verlag, 2002.

[4] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59:87.

[5] M. Conforti, M. R. Rao, and A. Sassano. The equipartition polytope. ii: Valid inequalities and facets. *Mathematical Programming*, 49:71.

[6] T. G. Crainic. Long haul freight transportation. In R. Hall, editor, *Handbook of Transportation Science*. Kluwer Academic Publishers, 2003.

[7] M. Grötschel and Y. Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*, 47:367.

[8] F. A. Özsoy and Labbé M. Size constrained graph partitioning polytope. part i: Dimension and trivial facets. Technical Report 577, ULB, Department of Computer Science, 2007.

[9] F. A. Özsoy and Labbé M. Size constrained graph partitioning polytope. part ii: Non-trivial facets. Technical Report 578, ULB, Department of Computer Science, 2007.

[10] F. A. Özsoy, Labbé M., and E. Gourdin. Analytical and empirical comparison of integer programming formulations for a partitioning-hub location-routing problem. Technical Report ?, ULB, Department of Computer Science, 2008.

[11] A. Retana and R. White. *IS-IS: Deployment in IP Networks*. Addison-Wesley, Boston, USA, 2003.

[12] D-W. Tcha, T-J. Choi, and Y-S. Myung. Location-area partition in a cellular radio network. *The Journal of Operational Research Society*, 48(11):1076–1081, 1997.