CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Benders Decomposition for Large-Scale Uncapacitated Hub Location

Ivan Contreras
Jean-François Cordeau
Gilbert Laporte

June 2010

CIRRELT-2010-26

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Benders Decomposition for Large-Scale Uncapacitated Hub Location

Ivan Contreras[1,2,3,*], Jean-François Cordeau[1,2], Gilbert Laporte[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Canada Research Chair in Logistics and Transportation, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[3] Canada Research Chair in Distribution Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. This paper describes an exact algorithm capable of solving large-scale instances of an important hub location problem called the Uncapacitated Hub Location Problem with Multiple Assignments. The algorithm applies Benders decomposition to a strong path-based formulation of the problem. The standard decomposition algorithm is enhanced through the inclusion of several features such as the use of a multicut reformulation, the generation of strong optimality cuts, the integration of reduction tests, and the execution of a heuristic procedure. Extensive computational experiments were performed to evaluate the efficiency and robustness of the algorithm. Computational results obtained on classical benchmark instances (with up to 200 nodes and 40,000 commodities) and on a new and more difficult set of instances (with up to 500 nodes and 250,000 commodities) confirm the efficiency of the algorithm.

Keywords. Hub location, Benders decomposition, Pareto-optimal cuts, elimination tests.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 227837-09 and 39682-05. This support is gratefully acknowledged.

---

* Corresponding author: Ivan.Contreras@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2010

1 Introduction

Transportation, telecommunications and computer networks frequently employ hub-and-spoke architectures to efficiently route demand between many origins and destinations. Their key feature lies in the use of consolidation, switching, or transshipment points, called *hub facilities*, to connect a large number of origin/destination (O/D) pairs by using a small number of links. This helps reduce setup costs, centralize commodity handling and sorting operations, and achieve economies of scale on routing costs through the consolidation of flows.

Hub Location Problems (HLPs) constitute a challenging class of $\mathcal{NP}$-hard combinatorial optimization problems combining location and network design decisions. Their main difficulty stems from the inherent interrelation between two levels of the decision process. The first level considers the selection of a set of nodes to locate hub facilities, whereas the second level deals with the design of the hub network, usually determined by the allocation pattern of nodes to hub facilities.

The field of hub location is rooted in the work of O'Kelly (1986) and has since evolved into a rich research area. We refer the reader to some of the main survey articles on this topic. The early reviews dealing with HLPs, by O'Kelly and Miller (1994) and Campbell (1994), contain classification schemes for the existing models and for the topological structures applicable to hub networks. Klincewicz (1998) later presented a survey on the design of hub networks in the context of telecommunication networks, and Bryan and O'Kelly (1999) concentrated on air transportation networks. Campbell et al. (2002) wrote a comprehensive survey on network hub location problems in which the location of hubs is the key decision. A more recent paper, by Alumur and Kara (2008), provides an updated and extensive review of the growing literature on network hub location models.

Despite the considerable efforts already made by many researchers, the optimal solution of HLPs remains challenging, particularly when considering more realistic, large-scale instances. To give an idea of the inherent difficulty of HLPs, instances with more than 50 nodes cannot be solved optimally for the vast majority of the variants considered in the literature, and it is only very recently that for some limited classes of HLPs, instances with up to 200 nodes have been solved optimally (see Camargo et al., 2008; Contreras et al., 2010).

In this paper we present an exact algorithm capable of solving large-scale instances for one of the most classical and general problems in the hub location literature, the *Uncapacitated Hub Location Problem with Multiple Assignments* (UHLPMA). In this problem, the capacity on the incoming and outgoing flows at the hub facilities and the amount of flow routed through each link of the hub network are unbounded. The number of hubs to locate is not known a priori, but a fixed set-up cost for each hub is considered. The objective is to minimize the sum of hub fixed costs and of demand transportation costs over the network. We consider the most general version of hub location in which multiple allocations are allowed, i.e., each O/D point may send and receive demand through several hubs. Note that a multiple assignment pattern is crucial when minimizing the total transportation cost, and includes the single assignment as a particular case (see e.g., Campbell, 1996).

There exist several papers on the UHLPMA. The first mathematical programming model was introduced by Campbell (1994) but was not computationally tested. Since then, several efforts have been made to produce better and tighter mixed integer programming (MIP)

formulations. Boland et al. (2004) have developed a multicommodity flow-based formulation capable of producing optimal solutions for instances with up to 50 nodes by using a general purpose solver. Later, Hamacher et al. (2004) and Marín et al. (2005) presented path-based formulations yielding much tighter LP bounds. However, due to their size, these formulations were only able to optimally solve instances with up to 25 nodes using general purpose solvers. The first exact algorithms, put forward by Klincewicz (1996) and by Mayer and Wagner (2002), were branch-and-bound (BB) methods based on dual ascent and dual adjustments techniques. In particular, the HubLocator algorithm (Mayer and Wagner, 2002) was able to obtain optimal solutions for instances with up to 40 nodes. Marín (2005) proposed a relax-and-cut algorithm that could solve to optimality instances with up to 50 nodes. Later, Cánovas et al. (2007) introduced a new BB method, also based on a dual ascent strategy. This method was able to solve to optimality instances with up to 120 nodes. Recently, Camargo et al. (2008) presented an exact Benders decomposition algorithm that was applied to instances involving up to 200 nodes. To the best of our knowledge, these instances are the largest ones ever solved exactly for any type of uncapacitated hub location problem.

The main contribution of this paper is to propose an exact algorithm applicable to large-scale instances of the UHLPMA involving up to 500 nodes and 250,000 commodities. It is a Benders decomposition algorithm based on the path-based formulation of Hamacher et al. (2004). The basic implementation of the algorithm is enhanced through several algorithmic features that make it more robust and efficient. These include: *i)* the use of a stronger multicut Benders reformulation, *ii)* the generation of stronger, almost undominated cuts, *iii)* the inclusion of reduction tests during the inner iterations of the Benders decomposition algorithm and, *iv)* the use of a heuristic for the a priori generation of optimality cuts. In order to evaluate and assess the robustness, efficiency and limitations of our proposed algorithm, extensive computational experiments were performed on the classical Australian Post data set and on a new challenging set of instances.

The remainder of the paper is organized as follows. Section 2 formally defines the problem, and presents an MIP formulation as well as properties of optimal solutions. The basic Benders reformulation, the Benders decomposition algorithm and some aspects of the dual problem are then presented in Section 3. Section 4 introduces several features that improve the convergence and efficiency of the algorithm. Section 5 presents the results of extensive computational experiments performed on a wide variety of instances. Conclusions follow in Section 6.

2 Problem Definition

Let $G = (N, A)$ be a complete digraph, where N is the set of nodes and A is the set of arcs. Let also $H \subseteq N$ represent the set of potential hub locations, and K represent the set of commodities whose origin and destination points belong to N. For each commodity $k \in K$, define W_k as the amount of commodity k to be routed from the origin $o(k) \in N$ to the destination $d(k) \in N$. For each node $i \in H$, f_i is the fixed set-up cost for locating a hub. The distance, or transportation cost d_{ij} between nodes i and j is assumed to satisfy the triangle inequality. The UHLPMA consists in locating a set of hubs and in determining the routing of commodity flows through the network, with the objective of minimizing the

total set-up and transportation cost.

Given that hub nodes are fully interconnected and distances satisfy the triangle inequality, every path between an origin and a destination node will contain at least one and at most two hubs. For this reason, paths between two nodes are of the form $(o(k), i, j, d(k))$, where $(i, j) \in H \times H$ is the ordered pair of hubs to which $o(k)$ and $d(k)$ are allocated, respectively. Therefore, the transportation cost of routing commodity k along the path $(o(k), i, j, d(k))$ is given by $\widehat{F}_{ijk} = W_k \left(\chi d_{o(k)i} + \tau d_{ij} + \delta d_{jd(k)}\right)$, where χ, τ, and δ represent the collection, transfer and distribution costs along the path. To reflect economies of scale between hub nodes, we assume that $\tau < \chi$ and $\tau < \delta$. We define binary location variables z_i, $i \in H$, equal to 1 if and only if a hub is located at node i. We also introduce binary routing variables x_{ijk}, $k \in K$ and $(i, j) \in H \times H$, equal to 1 if and only if commodity k transits via hub arc (i, j). Following Hamacher et al. (2004), the UHLPMA can be stated as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i \in H} f_i z_i + \sum_{i \in H} \sum_{j \in H} \sum_{k \in K} \widehat{F}_{ijk} x_{ijk} & & \\
\text{subject to} \quad & \sum_{i \in H} \sum_{j \in H} x_{ijk} = 1 & \forall\, k \in K & (1) \\
& \sum_{j \in H} x_{ijk} + \sum_{j \in H \setminus \{i\}} x_{jik} \leq z_i & \forall\, i \in H, \forall\, k \in K & (2) \\
& x_{ijk} \geq 0 & \forall\, i, j \in H, \forall\, k \in K & (3) \\
& z_i \in \{0, 1\} & \forall\, i \in H. & (4)
\end{aligned}$$

The first term of the objective function represents the total set-up cost of the hub facilities and the second term is the total transportation cost. Constraints (1) guarantee that there is a single path connecting the origin and destination nodes of every commodity. Constraints (2) prohibit commodities from being routed via a non-hub node. Finally, constraints (3) and (4) are the standard non-negativity and integrality constraints.

2.1 Properties of Optimal Solutions and Preprocessing

Several properties and characteristics of optimal UHLPMA solutions are known and can be used to perform preprocessing. In this section, we unify and summarize the most relevant results and present them in the context of the path-based formulation. Unless otherwise stated, the following properties are a consequence of the assumption of unlimited capacity at the hub nodes.

In any optimal UHLPMA solution, every path uses at most one direction of a hub edge $e = (e_1, e_2) \in H \times H$, the one with lowest transportation cost (Hamacher et al., 2004). We can therefore eliminate approximately half of the x_{ijk} variables associated to non-optimal directions by simply using an *undirected* transportation cost for every hub edge. Let $E = \{L \subseteq H : 1 \leq |L| \leq 2\}$ be the set of subsets of H containing one or two hubs. The *undirected* transportation cost F_{ek} for each $e \in E$ and $k \in K$ is defined as $F_{ek} = \min\{\widehat{F}_{ijk}, \widehat{F}_{jik}\}$.

Moreover, it can be shown that in any optimal UHLPMA solution, no commodity k will be routed through a hub edge e containing two different hubs whenever it is cheaper to route it through only one of them (Boland et al., 2004; Marín et al., 2005).

Property 1 *For every $k \in K$ and $e \in E$, $e_1 \neq e_2$, such that $F_{ek} > \min\left\{F_{(e_1,e_1)k}, F_{(e_2,e_2)k}\right\}$, $x_{ek} = 0$ in any optimal UHLPMA solution.*

We now consider the particular case of commodities k having the same origin and destination points, that is $o(k) = d(k)$. One can observe that such commodities will never be routed through two hubs. Indeed, they will always be collected and distributed by their closest open hub facility (Boland et al., 2004).

Property 2 *For every $e \in E$, such that $e_1 \neq e_2$ and $k \in K$ such that $o(k) = d(k)$, $x_{ek} = 0$ in any optimal UHLPMA solution.*

The above properties lead to a more compact formulation with fewer variables, but with the same number of constraints. We define a set of candidate hub edges for each commodity $k \in K$ as

$$E_k = \begin{cases} \{(i,i)|i \in H\} \bigcup \left\{e : e \in E, (e_1 \neq e_2) \text{ and } \left(F_{ek} < \min\left\{F_{(e_1,e_1)k}, F_{(e_2,e_2)k}\right\}\right)\right\}, & \text{if } o(k) \neq d(k), \\ \{(i,i)|i \in H\}, & \text{otherwise.} \end{cases}$$

The UHLPMA can thus be restated as

$$\text{minimize} \quad \sum_{i \in H} f_i z_i + \sum_{k \in K} \sum_{e \in E_k} F_{ek} x_{ek} \tag{5}$$

$$\text{subject to} \quad \sum_{e \in E_k} x_{ek} = 1 \qquad \forall\, k \in K \tag{6}$$

$$\sum_{e \in E_k : i \in e} x_{ek} \leq z_i \qquad \forall\, i \in H, \forall\, k \in K \tag{7}$$

$$x_{ek} \geq 0 \qquad \forall\, k \in K, \forall\, e \in E_k \tag{8}$$

$$z_i \in \{0,1\} \qquad \forall\, i \in H. \tag{9}$$

Finally, we consider the special case of symmetric transportation costs. Transportation costs are symmetric when the cost of path (i,k,m,j) is equal to the cost of path (j,m,k,i). That is, $F_{ek_1} = F_{ek_2}$ for each $e \in E$ and each pair of commodities (k_1, k_2) such that $o(k_1) = d(k_2)$ and $d(k_1) = o(k_2)$. The only condition for having symmetric transportation costs is that collection and distribution costs should be equal.

Property 3 *If $\chi = \delta$, then transportation costs F_{ek} are symmetric for each $k \in K$ and each $e \in E$.*

Whenever transportation costs are symmetric, we can further reduce the number of x_{ek} variables and constraints by considering as one commodity the sum of the two commodities having the exact same opposite O/D pairs.

3 Benders Decomposition

Benders decomposition is a well-known partitioning method applicable to mixed integer programs (Benders, 1962). It separates the original problem into two simpler ones: an

integer *master problem* and a linear *subproblem*. In this section, we introduce a Benders reformulation of the UHLPMA based on the compact formulation (5)–(8). We then describe a basic Benders decomposition algorithm to solve the reformulation. Because of degeneracy in the primal subproblem, there may exist multiple solutions in the dual. We thus present an efficient procedure to select, among the set of optimal dual solutions, an *appropriate* solution capable of generating a strong cut for the master problem.

3.1 Benders Reformulation

Let $Z = \mathbb{B}^{|H|}$ denote the set of binary vectors associated with the z_i variables. For any fixed vector $\hat{z} \in Z$, the *primal subproblem* (PS) in the space of the x_{ek} variables is

$$\begin{aligned} v(\hat{z}) = \text{minimize} \quad & \sum_{e \in E} \sum_{k \in K} F_{ek} x_{ek} \\ \text{subject to} \quad & (6), (8) \\ & \sum_{e \in E : i \in e} x_{ek} \leq \hat{z}_i \qquad \forall\, i \in H, \forall\, k \in K. \end{aligned} \tag{10}$$

Let α_k and u_{ik} be the dual variables associated with constraints (6) and (10), respectively. The *dual subproblem* (DS), which is the dual of PS, can be stated as follows:

$$\text{maximize} \quad \sum_{k \in K} \alpha_k - \sum_{i \in H} \sum_{k \in K} \hat{z}_i u_{ik} \tag{11}$$

$$\text{subject to} \quad \alpha_k - u_{e_1 k} - u_{e_2 k} \leq F_{ek} \qquad \forall k \in K, \forall e \in E, |e| = 2 \tag{12}$$

$$\alpha_k - u_{e_1 k} \leq F_{ek} \qquad \forall k \in K, \forall e \in E, |e| = 1 \tag{13}$$

$$u_{ik} \geq 0 \qquad \forall\, i \in H, \forall\, k \in K. \tag{14}$$

Let D denote the set of feasible solutions of DS and let P_D denote the set of extreme points of D. Observe that D is not modified when changing $\hat{z}$ and, because $F_{ek} \geq 0$ for each $e \in E_k$ and $k \in K$, the null vector 0 is always a solution to DS. Hence, because of strong duality, either the primal subproblem is feasible and bounded, or it is infeasible. We are thus interested in $\hat{z}$ vectors that give rise to primal subproblems of the former case. The following result establishes under which condition such vectors exist.

Proposition 1 *For any vector $z \in Z$ such that $\sum_{i \in H} z_i \geq 1$, the primal and dual subproblems are feasible and bounded.*

Proof For any vector z such that $\sum_{i \in H} z_i \geq 1$, there exists at least one possible path x_{ek} for every commodity $k \in K$ and thus, the primal problem is feasible. Moreover, since the transportation costs F_{ek} are finite and because of constraints (6) and (10), any feasible solution of PS must be bounded. By strong duality, the dual subproblem is also feasible and bounded. ■

It follows that the dual objective function value is equal to

$$\max_{(\alpha, u) \in P_D} \sum_{k \in K} \alpha_k - \sum_{i \in H} \sum_{k \in K} \hat{z}_i u_{ik}. \tag{15}$$

Introducing an extra variable η for the overall transportation cost, we can formulate the Benders master problem (MP) as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i \in H} f_i z_i + \eta \\
\text{subject to} \quad & \eta \geq \sum_{k \in K} \alpha_k - \sum_{i \in H} \sum_{k \in K} u_{ik} z_i \qquad \forall (\alpha, u) \in P_D \qquad (16) \\
& \sum_{i \in H} z_i \geq 1 \qquad (17) \\
& z_i \in \{0, 1\} \qquad \forall\, i \in H. \qquad (18)
\end{aligned}$$

Observe that Benders feasibility cuts associated with the extreme rays of D are not necessary in the Benders reformulation because the feasibility of PS is ensured by constraints (17). We have thus transformed problem (5)–(8) into an equivalent MIP problem with $|H|$ binary variables and one continuous variable. Nevertheless, the above Benders reformulation contains an exponential number of constraints and must be tackled by an adequate cutting plane approach. Thus, we iteratively solve relaxed master problems containing a small subset of the constraints (16) associated with the extreme points of P_D, and we keep adding these as needed by solving dual subproblems until an optimal solution to the original problem is obtained.

3.2 Basic Benders Decomposition Algorithm

Let ub denote an upper bound on the optimal solution value and let t represent the current iteration number. Let P_D^t denote the restricted set of extreme points of D at iteration t, MP(P_D^t) the relaxed master problem obtained by replacing P_D by P_D^t in MP, and $v(MP(P_D^t))$ its optimal solution value. Also, let z^t be an optimal solution vector of MP(P_D^t), DS(z^t) the dual subproblem for z^t, and $v(DS(z^t))$ its optimal solution value. A pseudo-code of the basic Benders decomposition algorithm is provided in Algorithm 1.

Algorithm 1: Benders decomposition

$ub \leftarrow \infty$, $t \leftarrow 0$
$P_D^t \leftarrow 0$
$terminate \leftarrow$ **false**
while ($terminate =$ **false**) **do**
  Solve MP(P_D^t) to obtain z^t
  if ($v(MP(P_D^t)) = ub$) **then**
    $terminate \leftarrow$ **true**
  else
    Solve DS(z^t) to obtain $(\alpha, u) \in P_D$
    $P_D^{t+1} \leftarrow P_D^t \cup \{(\alpha, u)\}$
    if ($v(DS(z^t)) + \sum_{i \in H} f_i \hat{z}_i < ub$) **then**
      $ub \leftarrow v(DS(z^t)) + \sum_{i \in H} f_i \hat{z}_i$
    end if
  end if
  $t \leftarrow t + 1$
end while

Whenever the problem defined by (5)–(9) is feasible, Algorithm 1 will yield an optimal solution. The computational efficiency of the above Benders decomposition algorithm depends mainly on: *i)* the computational effort needed to solve MP(P_D^t), *ii)* the computational effort needed to solve DS(z^t), and *iii)* the number of iterations required to obtain an optimal solution. Next, we present a methodology for efficiently solving DS(z^t) by exploiting the structure of the primal subproblem. In Section 4, we will present some techniques focusing on *ii)* and *iii)*.

3.3 Solving the Subproblem

At any iteration t of Algorithm 1, we obtain an optimal solution vector z^t of MP(P_D^t). Let $H_1^t = \{i : z_i^t = 1\}$ be the set of open hubs and $H_0^t = \{i : z_i^t = 0\}$ be the set of closed hubs. Given that $z^t \in Z$, we can exploit the structure of the primal subproblem to obtain a vector of optimal dual variables (α^t, u^t) more efficiently than by using an LP solver for the explicit solution of DS. In particular, observe that PS can be reduced to the equivalent problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} \sum_{k \in K} F_{ek} x_{ek} & & \\
\text{subject to} \quad & \sum_{e \in E_k \cap \left(H_1^t \times H_1^t\right)} x_{ek} = 1 & \forall\, k \in K, & \qquad (19) \\
& x_{ek} \geq 0 & \forall\, e \in E, \forall\, k \in K. & \qquad (20)
\end{aligned}$$

This problem can be separated into $|K|$ independent subproblems PS_k^t, one for each commodity $k \in K$. Each PS_k^t is a *semi-assignment* problem which can be easily solved by choosing the minimum transportation cost route among those that use open hubs. For a given k, a

primal optimal solution of PS_k^t, denoted by x^t, can be expressed as

$$x^t_{e(k)k} = 1, \qquad \text{for } e(k) = \arg\min\left\{F_{ek} : e \in E_k \cap \left(H_1^t \times H_1^t\right)\right\} \tag{21}$$

$$x^t_{ek} = 0, \qquad \text{for } e \in E_k \setminus \{e(k)\}. \tag{22}$$

The optimal solution value of PS at z^t, denoted as $v(z^t)$, can thus be expressed as

$$v(z^t) = \sum_{k\in K} F_{e(k)k} = \sum_{k\in K} \min_{e\in E}\left\{F_{ek} : e \in E_k \cap \left(H_1^t \times H_1^t\right)\right\}. \tag{23}$$

In order to obtain an associated optimality cut, we still need to produce an optimal dual solution (α^t, u^t). We can use duality theory to recover a dual solution (α^t, u^t) from the primal optimal solution x^t. In particular, the complementary slackness conditions are

$$u^t_{ik}\left(\sum_{e\in E_k: i\in e} x^t_{ek} - z^t_i\right) = 0, \qquad \forall i \in H, k \in K, \tag{24}$$

$$x^t_{ek}\left(\alpha^t_k - u^t_{e_1k} - u^t_{e_2k} - F_{ek}\right) = 0, \qquad \forall k \in K,\ e \in E_k, |e| = 2, \tag{25}$$

$$x^t_{ek}\left(\alpha^t_k - u^t_{ek} - F_{ek}\right) = 0, \qquad \forall k \in K,\ e \in E_k, |e| = 1. \tag{26}$$

First, conditions (24) imply that

$$u^t_{ik} = 0, \qquad \forall i \in H_1^t \setminus \{e_1(k), e_2(k)\}, \forall k \in K. \tag{27}$$

Next, conditions (25) and (26) imply that dual slack variables, associated to optimal primal variables x^t_{ek} set to one, must be equal to zero. For each $k \in K$, this condition is

$$\alpha^t_k - u^t_{e_1(k)k} - u^t_{e_2(k)k} = F_{e(k)k}, \qquad \text{if } |e(k)| = 2, \tag{28}$$

$$\alpha^t_k - u^t_{e_1(k)k} = F_{e(k)k}, \qquad \text{if } |e(k)| = 1. \tag{29}$$

This implies that every feasible solution $(\alpha, u) \in D$ satisfying (27)–(29) is indeed an optimal solution of DS. We thus have characterized the set of optimal solutions of the dual subproblem associated to the optimal primal solution x^t.

Proposition 2 *Let x^t be an optimal solution of PS_k^t. The set of optimal dual solutions of DS^t associated to x^t can be characterized as*

$$DO^t = \{(\alpha, u) \in D :\ (27)\text{–}(29)\ \textit{hold}\}.$$

The above result implies that we can construct optimal dual solutions (α^t, u^t) from the optimal primal solution x^t in two steps. First, we fix each α^t_k, $u^t_{e_1(k)k}$ and $u^t_{e_2(k)k}$, for each $k \in K$, to a particular *feasible* value, with respect to constraints (12)–(13) and conditions (28)–(29), and we fix each u^t_{ik}, such that $i \in H_1^t \setminus \{e_1(k), e_2(k)\}$, to zero. Second, we solve a *reduced* system of inequalities by fixing the variables from the first step in constraints (12) and (13), to obtain an optimal value of the remaining u_{ik} such that $i \in H_0^t$, for each $k \in K$.

In the remainder of this section, we focus on computing an optimal solution (α^t, u^t) from a subset of DO^t associated to solutions in which $\alpha^t_k = F_{e(k)k}$, $u^t_{e_1(k)k} = 0$ and $u^t_{e_2(k)k} = 0$, for

each $k \in K$. By doing so, we avoid checking the feasibility of these variables with respect to constraints (12)–(13). In Section 4 we present some theoretical insights that help us select particular values of the α_k^t, $u_{e_1(k)k}^t$ and $u_{e_2(k)k}^t$ variables associated with optimal dual solutions that could produce stronger optimality cuts.

Observe that some constraints (12) can now be dropped from the model once α_k^t, $u_{e_1(k)k}^t$ and $u_{e_2(k)k}^t$ are set to a particular value for each $k \in K$. Because of constraints (14), we know that constraints (12) having a non-positive right-hand side $\lambda_{ek} = \alpha_k^t - F_{ek}$ are always satisfied. Hence, we only need to consider constraints (12) such that $\lambda_{ek} > 0$. Let $E_k^+ = \{e : \lambda_{ek} > 0, e \in E_k \cap (H_0^t \times H_0^t)\}$ denote this subset of constraints. Moreover, because $u_{ik}^t = 0$ for each $i \in H_1^t$, constraints (12) associated with edges incident to a node $i \in H_1^t$ can be implicitly considered as lower bounds for the remaining u_{ik} variables. In particular, for each node $i \in H_0^t$, let $\mu_{e_1}^i = \max\{\lambda_{ek} : e \in E_k \cap (H_0^t \times H_1^t), e_1 = i\}$ denote the maximum λ_{ek} value of constraints (12) associated with edges having i as first node and any second node $e_2 \in H_1^t$. Similarly, for each node $i \in H_0^t$, let $\mu_{e_2}^i = \max\{\lambda_{ek} : e \in E_k \cap (H_1^t \times H_0^t), e_2 = i\}$ denote the maximum λ_{ek} value of constraints (12) having i as second node and any first node $e_1 \in H_1^t$. Using $\mu_{e_1}^i$, $\mu_{e_2}^i$ and constraints (13), we set the lower bound of u_{ik} variable as $l_{ik} = \max\left\{0,\ \lambda_{(i,i)k},\ \mu_{e_1}^i,\ \mu_{e_2}^i\right\}$ for each node $i \in H_0^t$. From these results, we obtain the *reduced* system

$$u_{e_1k} + u_{e_2k} \geq \lambda_{ek} \qquad \forall e \in E_k^+ \tag{30}$$

$$u_{ik} \geq l_{ik} \qquad \forall\ i \in H_0^t. \tag{31}$$

Nevertheless, not all feasible solutions of (30)–(31) are candidates to generate useful optimality cuts. Given the non-positive coefficients of the z_i variables in (15), optimal dual vectors (α^t, u^t) having large elements in u^t are likely to produce weak optimality cuts. We are therefore interested in (α^t, u^t) vectors for which u^t is as small as possible in order to obtain the largest possible lower bound on $\mathrm{MP}(P_D^{t+1})$.

Algorithm 2 describes a simple procedure for the computation of an optimal solution (α^t, u^t) having small u^t elements. It constructs a solution by directly ensuring feasibility of the system (30)–(31) row by row, while keeping the value of each u_{ik} variable as small as possible. Let ψ and γ be two non-negative parameters such that $\psi + \gamma = 1$.

Algorithm 2: Computing $(\boldsymbol{\alpha^t}, \boldsymbol{u^t})$

forall $(k \in K)$ **do**
  $\alpha_k^t \leftarrow \min\{F_{ek} : e \in E_k \cap (H_1^t \times H_1^t)\}$
  forall $(i \in H_1^t)$ **do**
    $u_{ik} \leftarrow 0$
  end do
  forall $(i \in H_0^t)$ **do**
    $u_{ik} \leftarrow l_{ik}$
  end do
  forall $(e \in E_k^+)$ **do**
    $\Delta \leftarrow u_{e_1k} + u_{e_2k} - \lambda_{ek}$
    if $(\Delta < 0)$ **then**
      $u_{e_1k} \leftarrow u_{e_1k} - \psi\Delta$
      $u_{e_2k} \leftarrow u_{e_2k} - \gamma\Delta$
    end if
  end do
end do

The above algorithm has an $\mathcal{O}(\sum_{k\in K} |E_k^+|)$ time complexity. Note that this procedure does not necessarily produce an extreme point of (30)–(31) as in the case of the simplex method. We could instead obtain a point lying on a face of the polyhedron defined by (12)–(14). However, this does not cause any problem because we are still producing a valid Benders cut which will separate the optimal solution of the current master problem MP^t, thus ensuring convergence.

4 Algorithmic Refinements

We now analyze several ways of improving the convergence and stability of the Benders decomposition algorithm presented in the previous section. We first present a multicut version of the Benders reformulation, which exploits the decomposability of the subproblem. Theoretical aspects concerning stronger, non-dominated optimality cuts are then introduced and used to develop an algorithm capable of efficiently generating stronger cuts that those presented in Section 3. Later, we show how to incorporate some reduction tests into the Benders decomposition algorithm in order to reduce the size of both the master problem and the subproblem, and thus accelerate its convergence. Finally, we present a simple heuristic procedure that can be used to generate an initial set of optimality cuts for the master problem to accelerate the convergence of the algorithm and to improve the efficiency of the reduction tests.

4.1 Multicut Benders Reformulation

It is known that the number of cuts required to obtain an optimal solution of the Benders reformulation will be, in the worst case, equal to the number of extreme points in D. However, this number can be reduced given that the subproblem is decomposable into $|K|$ independent

subproblems (see, e.g. Birge and Louveaux, 1988). We could in principle generate optimality cuts associated to extreme points of each dual polyhedron of the $|K|$ subproblems, but Camargo et al. (2008) show that when adding $|K|$ cuts per iteration, the reduction in the number of iterations is not justified by the increased computational effort required for the solution of the relaxed master problems, even for small size instances.

Instead of adding in a disaggregated way all $|K|$ cuts at each iteration, we can aggregate the information obtained to generate a set of optimality cuts associated with subsets of commodities. In particular, for each node $j \in H$, let $K_j \subset K$ be the subset of commodities whose origin node is j. We can separate the subproblem into $|H|$ independent subproblems, one for each node. Hence, we consider the dual polyhedra of these $|H|$ subproblems and generate cuts from them. Let P_D be the set of extreme points of the dual polyhedron P_{D^j} associated with subproblem i. We thus obtain the following Benders reformulation:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in H} f_i z_i + \sum_{i \in H} \eta_i \\ \text{subject to} \quad & (17), (18) \\ & \eta_j \geq \sum_{k \in K_j} \alpha_k^t - \sum_{i \in H} \sum_{k \in K_j} u_{ik}^t z_i \qquad \forall j \in H,\ \forall (\alpha, u) \in P_{D^j}. \end{aligned} \tag{32}$$

Using this reformulation, only $|H|$ potential optimality cuts will be generated when solving the subproblem, instead of $|K|$ cuts as is the case when considering the complete separability into $|K|$ dual subproblems.

4.2 Pareto-optimal Cuts

One way to improve the convergence of the Benders algorithm is to construct stronger, undominated cuts, known as *Pareto-optimal* cuts (Magnanti and Wong, 1981). We say that the cut generated from the dual solution (α^a, u^a) dominates the cut generated from the dual solution (α^b, u^b) if and only if

$$\sum_{k \in K} \alpha_k^a - \sum_{i \in H} \sum_{k \in K} u_{ik}^a z_i \geq \sum_{k \in K} \alpha_k^b - \sum_{i \in H} \sum_{k \in K} u_{ik}^b z_i$$

for all $z \in Z$ with strict inequality for at least one point. A cut is Pareto-optimal if no other cut dominates it. Let Q be the polyhedron defined by (17) and $0 \leq z_i \leq 1$ for all $i \in H$, and let $ri(Q)$ denote the relative interior of Q. To identify a Pareto-optimal cut at iteration t, we must solve the following *Pareto-optimal subproblem* (PO^t):

$$\text{maximize} \quad \sum_{k \in K} \alpha_k - \sum_{i \in H} \sum_{k \in K} z_i^0 u_{ik} \tag{33}$$

$$\begin{aligned} \text{subject to} \quad & (12) - (13), \\ & \alpha_k - \sum_{i \in H} z_i^t u_{ik} = F_{e(k)k} \qquad \forall k \in K, \end{aligned} \tag{34}$$

where $z^0 \in ri(Q)$ and, as before, $F_{e(k)k}$ is the optimal solution value of subproblem k. Constraints (34) ensure that the optimal solution of PO^t is chosen from the set of optimal

solutions of DS^t. Note that PO^t can also be separated into $|K|$ independent subproblems (PO^t_k), one for each $k \in K$. We thus obtain

$$\text{maximize} \quad \alpha_k - \sum_{i \in H} z^0_i u_{ik} \tag{35}$$

$$\text{subject to} \quad \alpha_k - \sum_{i \in H} z^t_i u_{ik} = F_{e(k)k} \tag{36}$$

$$\alpha_k - u_{e_1 k} - u_{e_2 k} \leq F_{ek} \qquad \forall e \in E_k, |e| = 2 \tag{37}$$

$$\alpha_k - u_{e_1 k} \leq F_{ek} \qquad \forall e \in E_k, |e| = 1 \tag{38}$$

$$u_{ik} \geq 0 \qquad \forall\, i \in H. \tag{39}$$

Because of constraint (36) and of the *fractional* coefficients z^0_i, the primal structure of (35)–(39) cannot be exploited to efficiently obtain an optimal dual solution, as is the case for the DS. This means that we need to solve $|K|$ linear programs, one for each $k \in K$, to obtain a Pareto-optimal cut. Computational experiments indicate that the generation of Pareto-optimal cuts considerably reduces the number of required iterations to converge. However, the time needed to solve the $|K|$ linear programs is not compensated by the improved convergence of the Benders algorithm, even on small-size instances. Given that our goal is to solve large-scale instances, we have developed an efficient procedure capable of producing good approximations of the optimal solution of PO^t_k, and thus of generating stronger optimality cuts, without requiring the explicit solution of (35)–(39).

Here we present an approximate procedure capable of efficiently producing stronger optimality cuts, which are not necessarily Pareto-optimal, by exploiting the fact that PO^t_k can be expressed as the maximization of a *piecewise linear* and *concave* function of α_k. In particular, if we fix the value of the α_k variable in (35)–(39), we can write the resulting subproblem as the following implicit function:

$$L(\alpha_k) = \text{maximize} \quad -\sum_{i \in H} z^0_i u_{ik}$$

$$\text{subject to} \quad \sum_{i \in H} z^t_i u_{ik} = \alpha_k - F_{e(k)k} \tag{40}$$

$$u_{e_1 k} + u_{e_2 k} \geq \alpha_k - F_{ek} \qquad \forall e \in E_k, |e| = 2 \tag{41}$$

$$u_{e_1 k} \geq \alpha_k - F_{ek} \qquad \forall e \in E_k, |e| = 1 \tag{42}$$

$$u_{ik} \geq 0 \qquad \forall\, i \in H. \tag{43}$$

We now can state PO^t_k as

$$\max_{\alpha_k} G(\alpha_k), \tag{44}$$

where $G(\alpha_k) = \alpha_k - L(\alpha_k)$.

Proposition 3 *$G(\alpha_k)$ is a piecewise linear and concave function of α_k.*

Proof Rewriting the right-hand side vector of constraints (40)–(43) as $b + \alpha_k \hat{b}$, where $b = (-F_{e(k)k}, -F_{1k}, \ldots, -F_{|E_k|k})$ and $\hat{b} = (1, 1, \ldots, 1)$, the problem can be viewed as a linear program in which the right-hand side vector is perturbed along the identity vector.

From linear programming theory, we know that parametric analysis on the right-hand side vector in a maximization problem always produces a piecewise linear and concave function (Bazaraa et al., 1990). Therefore, $L(\alpha_k)$ is a piecewise linear and concave function of α_k. ■

By applying parametric analysis over $L(\alpha_k)$, we can determine the ranges of the linear segments and, thus, the break points at which changes of *optimal* bases (with respect to α_k) take place in $G(\alpha_k)$. Moreover, the slope of each linear segment can be computed using the information of its associated optimal basis. We can therefore obtain an optimal solution of PO_k^t as follows. First, set α_k to some initial feasible value and evaluate $L(\alpha_k)$ to obtain an optimal basis associated to a linear segment. Then perform as many dual simplex iterations as break points exist before reaching a point at which the slope of $G(\alpha_k)$ is equal to zero. Even though this procedure is more efficient than solving PO_k^t directly by an LP solver, it still requires the solution of an LP problem to generate an initial optimal basis and its update at each break point.

Instead of optimally solving PO_k^t to produce a Pareto-optimal cut, we solve PO_k^t only approximately and still produce strong, but not necessarily undominated optimality cuts. Our procedure is based on the estimation of the function $G(\alpha_k)$ by using an adaptation of Algorithm 2 presented in Section 3. Using this estimation, we successively evaluate $G(\alpha_k)$ within a given interval $L_k \leq \alpha_k \leq U_k$ and increase α_k until the estimation of $G(\alpha_k)$ stops increasing, or until $\alpha_k = U_k$. In what follows, we present the details on how to efficiently evaluate $G(\alpha_k)$ and how to construct an interval in which the optimal value of α_k is contained. Then, we summarize the overall procedure.

4.2.1 Evaluating $G(\alpha_k)$.

Optimal solutions of PS_k^t affect the structure of PO_k^t and we must therefore distinguish between two possible cases when evaluating $G(\alpha_k)$: either the optimal edge $e(k)$ has a single hub node ($|e(k)| = 1$) or it has two different hub nodes ($|e(k)| = 2$). Thus, we need to define two functions $G^1(\alpha_k)$ and $G^2(\alpha_k)$ and two intervals $L_k^1 \leq \alpha_k \leq U_k^1$ and $L_k^2 \leq \alpha_k \leq U_k^2$, respectively. Using constraint (36), the objective function (35) can be expressed as

$$\text{maximize} \quad F_{e(k)k} + \sum_{i \in H} (z_i^t - z_i^0) u_{ik}. \tag{45}$$

For any $i \in H_1^t$, we have $z_i^t = 1$ and the coefficient $\delta_i = z_i^t - z_i^0$ is strictly positive. If $i \in H_0$, we have $z_i^t = 0$ and the coefficient δ_i is strictly negative. Therefore, we would like to increase as much as possible the value of the u_{ik} variables such that $i \in H_1^t$, and keep as low as possible the value of the u_{ik} variables such that $i \in H_0^t$.

For a given α_k, the value of u_{ik} variables such that $i \in H_1^t$ can be already determined using constraints (36)–(39). If $|e(k)| = 2$, given that constraint (36) can be read as $\alpha_k - F_{e(k)k} = \sum_{i \in H_1^t} u_{ik}$, and since $u_{e_1(k)k} + u_{e_2(k)k} \geq \alpha_k - F_{e(k)k}$ and $u_{ik} \geq 0$ for each $i \in H_1^t$, we have

$$u_{e_1(k)k} + u_{e_2(k)k} = \alpha_k - F_{e(k)k}, \tag{46}$$

$$u_{ik} = 0, \qquad \forall i \in H_1^t \setminus \{e_1(k), e_2(k)\}. \tag{47}$$

If $|e(k)| = 1$, by using similar arguments we obtain

$$u_{e_1(k)k} = \alpha_k - F_{e(k)k}, \tag{48}$$
$$u_{ik} = 0, \qquad \forall i \in H_1^t \setminus \{e_1(k)\}. \tag{49}$$

Hence, for a fixed α_k, the optimal solution for the remaining u_{ik} variables can be determined by solving a reduced subproblem. As in the case of the dual subproblem DS, once α_k is fixed some constraints (37) can be eliminated from the model. In particular, we only need to consider constraints (37) whose right-hand side $\lambda_{ek}(\alpha_k) = \alpha_k - F_{ek}$ is strictly positive. Let $E_k^+(\alpha_k) = \{e : \lambda_{ek}(\alpha_k) > 0, e \in E_k \cap (H_0^t \times H_0^t)\}$ denote this set of constraints. Furthermore, because of constraints (47) and (49), constraints associated with edges incident to a node $i \in H_1^t \setminus \{e_1(k), e_2(k)\}$ can be seen as lower bounds for the remaining u_{ik} variables. For every $i \in H_0^t$, let

$$\mu_{e_1}^i(\alpha_k) = \max\left\{\lambda_{ek}(\alpha_k) : e \in E_k \cap \left(H_0^t \times H_1^t\right), e_1 = i, e_2 \neq e_1(k) \text{ and } e_2 \neq e_2(k)\right\}$$

denote the maximum $\lambda_{ek}(\alpha_k)$ value associated to hub edges having i as first node and any second node $e_2 \in H_1^t \setminus \{e_1(k), e_2(k)\}$. Similarly, for every $i \in H_0^t$, let

$$\mu_{e_2}^i(\alpha_k) = \max\left\{\lambda_{ek}(\alpha_k) : e \in E_k \cap \left(H_1^t \times H_0^t\right), e_2 = i, e_1 \neq e_1(k) \text{ and } e_1 \neq e_2(k)\right\}$$

denote the maximum $\lambda_{ek}(\alpha_k)$ value associated to hub edges having i as second node and any first node $e_2 \in H_1^t \setminus \{e_1(k), e_2(k)\}$. Using $\mu_{e_1}^i(\alpha_k)$, $\mu_{e_2}^i(\alpha_k)$ and constraint (38), we set the lower bound of u_{ik} variable, for $i \in H_0^t$, as

$$l_{ik}(\alpha_k) = \max\left\{0,\ \lambda_{(i,i)k}(\alpha_k),\ \mu_{e_1}^i(\alpha_k),\ \mu_{e_2}^i(\alpha_k)\right\}.$$

If $|e(k)| = 2$, the exact value of $u_{e_1(k)k}$ and $u_{e_2(k)k}$ must also be determined by the reduced problem. This also implies that we have to include some additional constraints in the problem. In particular, let

$$EX_k^+(\alpha_k) = \left\{e : \lambda_{ek}(\alpha_k) > 0, e \in E_k \cap \left(H_1^t \times H_0^t\right), e_1 = e_1(k) \text{ and } e_1 = e_2(k)\right\}$$

denote the subset of constraints (38) associated with hub edges containing either $e_1(k)$ or $e_2(k)$ as first node and any second node $e_2 \in H_0^t$. Similarly, let

$$EY_k^+(\alpha_k) = \left\{e : \lambda_{ek}(\alpha_k) > 0, e \in E_k \cap \left(H_0^t \times H_1^t\right), e_2 = e_2(k) \text{ and } e_2 = e_1(k)\right\}$$

denote the subset of constraints (38) associated to hub edges containing any first node $e_1 \in H_0^t$ and either $e_1(k)$ or $e_2(k)$ as second node. Combining the previous results, we can state $G(\alpha_k)$ as

$$G^1(\alpha_k) = F_{e(k)k} + \delta_{e_1(k)}\left(\alpha_k - F_{e(k)k}\right) + f_k^1(\alpha_k), \tag{50}$$

where

$$f_k^1(\alpha_k) = \text{ maximize} \qquad \sum_{i \in H_0} \delta_i u_{ik} \tag{51}$$
$$\text{subject to} \qquad u_{e_1k} + u_{e_2k} \geq \lambda_{ek}(\alpha_k) \qquad \forall e \in E_k^+(\alpha_k) \tag{52}$$
$$u_{ik} \geq l_{ik}(\alpha_k) \qquad \forall\, i \in H_0^t, \tag{53}$$

if $|e(k)| = 1$ and as

$$G^2(\alpha_k) = F_{e(k)k} + f_k^2(\alpha_k), \tag{54}$$

where

$$\begin{aligned} f_k^2(\alpha_k) = \text{ maximize } & \sum_{i \in H_0^t} \delta_i u_{ik} + \delta_{e_1(k)} u_{e_1(k)k} + \delta_{e_2(k)} u_{e_2(k)k} && (55) \\ \text{subject to } & (46), (52), (53) \\ & u_{e_1 k} + u_{e_2 k} \geq \lambda_{ek}(\alpha_k) \qquad \forall e \in EX_k^+(\alpha_k) \cup EY_k^+(\alpha_k), && (56) \end{aligned}$$

if $|e(k)| = 2$.

Given that constraints (52)–(53) and (56) are very similar to the reduced system (30)–(31) of DS_k^t, we can adapt Algorithm 2 to produce feasible solutions to both $f_k^1(\alpha_k)$ and $f_k^2(\alpha_k)$. Using these solutions, we are able to efficiently provide a good estimation of the value of $G(\alpha_k)$ for any feasible α_k value. The main difference with respect to Algorithm 2 is that we now have to consider that $u_{e_1(k)k}$ and $u_{e_2(k)k}$ may take a strictly positive value. If $|e(k)| = 1$, the value of $u_{e_1(k)k}$ for a given α_k is uniquely given by (48). If $|e(k)| = 2$, variables $u_{e_1(k)k}$ and $u_{e_2(k)k}$ can take an infinite number of possible values with respect to (46). However, we have to consider constraints (37)–(38) to ensure the feasibility of the solution vector. Therefore, we need to construct upper bounds on the maximum feasible value that variables $u_{e_1(k)k}$ and $u_{e_2(k)k}$ may take. In particular, constraints (37) having edges e such that $e_1(k) \notin e$ and $e_2(k) \in e$, provide an upper bound for $u_{e_1(k)k}$. Substituting $u_{e_2(k)k} = \alpha_k - F_{e(k)k} - u_{e_1(k)k}$ in these constraints we obtain

$$u_{e_1(k)k} \leq F_{ek} - F_{e(k)k}, \qquad e \in EU1, \tag{57}$$

where

$$EU1 = \left\{ e : e \in E_k \cap \left(H_1^t \times H_1^t\right),\ (e_2 = e_2(k) \text{ and } e_1 \neq e_1(k)) \text{ or } (e_2 = e_1(k) \text{ and } e_1 \neq e_2(k)) \right\}.$$

From constraints (57) we can set an upper bound for $u_{e_1(k)k}$ as $h_k^{e_1} = \min\left\{F_{ek} - F_{e(k)k} : e \in EU1\right\}$. In a similar way, constraints (37) having edges e such that $e_1(k) \in e$ and $e_2(k) \notin e$, provide an upper bound for $u_{e_2(k)k}$. Substituting $u_{e_1(k)k} = \alpha_k - F_{e(k)k} - u_{e_2(k)k}$ in these constraints we obtain,

$$u_{e_2(k)k} \leq F_{ek} - F_{e(k)k}, \qquad e \in EU2 \tag{58}$$

where,

$$EU2 = \left\{ e : e \in E_k \cap \left(H_1^t \times H_1^t\right),\ (e_1 = e_1(k) \text{ and } e_2 \neq e_2(k)) \text{ or } (e_1 = e_2(k) \text{ and } e_2 \neq e_1(k)) \right\}.$$

From constraints (58) we can set an upper bound for $u_{e_2(k)k}$ as $h_k^{e_2} = \min\left\{F_{ek} - F_{e(k)k} : e \in EU2\right\}$. Observe that the feasibility of variables $u_{e_1(k)k}$ and $u_{e_2(k)k}$ can be ensured by fixing them to

$$u_{e_1(k)k} = \frac{h_k^{e_1} \times (\alpha_k - F_{e(k)k})}{h_k^{e_1} + h_k^{e_2}}, \tag{59}$$

and

$$u_{e_2(k)k} = \frac{h_k^{e_2} \times (\alpha_k - F_{e(k)k})}{h_k^{e_1} + h_k^{e_2}}, \tag{60}$$

respectively. Finally, Algorithm 3 summarizes the proposed procedure to obtain an estimation of the value $G(\alpha_k)$ at point α_k.

Algorithm 3: Approximate evaluation of $G(\alpha_k)$

if $(|e(k)| = 1)$ **then**
  $u_{e_1(k)k} \leftarrow \alpha_k - F_{e(k)k}$
  $E \leftarrow E_k^+(\alpha_k)$
else
  $u_{e_1(k)k} \leftarrow h_k^{e_1} \times (\alpha_k - F_{e(k)k})/(h_k^{e_1} + h_k^{e_2})$
  $u_{e_2(k)k} \leftarrow h_k^{e_2} \times (\alpha_k - F_{e(k)k})/(h_k^{e_1} + h_k^{e_2})$
  $E \leftarrow E_k^+(\alpha_k) \cup EX_k^+(\alpha_k) \cup EY_k^+(\alpha_k)$
end if
forall $(i \in H_1^t \setminus \{e_1(k), e_2(k)\})$ **do**
  $u_{ik} \leftarrow 0$
end do
forall $(i \in H_0^t)$ **do**
  $u_{ik} \leftarrow l_{ik}(\alpha_k)$
end do
forall $(e \in E)$ **do**
  $\Delta \leftarrow u_{e_1k} + u_{e_2k} - \lambda_{ek}$
  if $(\Delta < 0)$ **then**
    $u_{e_1k} \leftarrow u_{e_1k} - \psi\Delta$
    $u_{e_2k} \leftarrow u_{e_2k} - \gamma\Delta$
  end if
end do
if $(|e(k)| = 1)$ **then**
  $G(\alpha_k) \leftarrow F_{e(k)k} + \delta_{e_1(k)} \left(\alpha_k - F_{e(k)k}\right) + \sum_{i \in H_0} \delta_i u_{ik}$
else
  $G(\alpha_k) \leftarrow F_{e(k)k} + \delta_{e_1(k)} u_{e_1(k)k} + \delta_{e_2(k)} u_{e_2(k)k} + \sum_{i \in H_0} \delta_i u_{ik}$
end if

4.2.2 Constructing an Interval $L_k \leq \alpha_k \leq U_k$.

Similar to $G(\alpha_k)$, we need to define two intervals, one for $|e(k)| = 1$ and another for $|e(k)| = 2$. Let $L_k^1 \leq \alpha_k \leq U_k^1$ and $L_k^2 \leq \alpha_k \leq U_k^2$, denote these intervals. We can compute the lower bound for both intervals by observing that, regardless of the structure of $e(k)$, constraints (36) can be stated as $\alpha_k = F_{e(k)k} + \sum_{i \in H_1^t} u_{ik}$ and, given that $u_{ik} \geq 0$ for each $i \in H_1^t$, we know that $\alpha_k \geq F_{e(k)k}$. We thus can set the lower bounds L_k^1 and L_k^2 equal to $L_k^1 = L_k^2 = F_{e(k)k}$. However, the upper bounds U_k^1 and U_k^2 can be different. For $|e(k)| = 1$, given that $u_{ik} = 0$ for each $i \in H_1^t \setminus \{e_1(k)\}$, the minimum coefficient F_{ek} associated to constraints (37) having hub edges containing only open nodes different from $e_1(k)$ is a upper bound of α_k. Therefore, we can set upper bound $U_k^1 = \min \{F_{ek} : e \in E_k \cap (H_1^t \times H_1^t), e_1 \neq e_1(k) \text{ and } e_2 \neq e_1(k)\}$.

Using similar arguments, we can derive an upper bound when $|e(k)| = 2$. In particular, since $u_{ik} = 0$ for each $i \in H_1^t \setminus \{e_1(k), e_2(k)\}$, the minimum coefficient F_{ek} associated with constraints (37) having edges that contain only open nodes different from $e_1(k)$ and $e_2(k)$ defines an upper bound of α_k. Let

$$h_k^1 = \min\left\{F_{ek} : e \in E_k \cap \left(H_1^t \times H_1^t\right),\ (e_1 \neq e_1(k) \text{ and } e_2 \neq e_2(k)) \text{ or } (e_1 \neq e_2(k) \text{ and } e_2 \neq e_1(k))\right\}$$

denote such an upper bound. Also, we obtain a second upper bound of α_k from the upper bounds $h_k^{e_1}$ and $h_k^{e_2}$ for the $u_{e_1(k)k}$ and $u_{e_2(k)k}$ variables, respectively. We thus can set upper bound U_k^2 to $U_k^2 = \min\left\{h_k^1, h_k^{e_1} + h_k^{e_2}\right\}$.

4.2.3 Approximate Solution of PO^t.

Instead of computing an optimal solution to PO^t when generating a Pareto-optimal cut, which can be computationally prohibitive, we focus on efficiently generating good solutions that could lead to stronger optimality cuts than those obtained with Algorithm 2, even though they may not necessarily be Pareto-optimal.

We construct promising solutions of PO^t by discretizing the $G(\alpha_k)$ function over the previously constructed interval $L_k \leq \alpha_k \leq U_k$. In particular, we divide the interval into κ equal size smaller intervals and focus the search on the extreme points of these intervals. At each point α_k, the value of $G(\alpha_k)$ is estimated by using Algorithm 3. The proposed procedure to approximately solve PO^t is given in Algorithm 4.

Algorithm 4: Approximate solution of PO^t

forall $(k \in K)$ **do**
  $L_k \leftarrow \min\left\{F_{ek} : e \in E_k \cap (H_1^t \times H_1^t)\right\}$
  if $(|e(k)| = 1)$ **then**
    $U_k \leftarrow \min\left\{F_{ek} : e \in E_k \cap (H_1^t \times H_1^t), e_1 \neq e_1(k) \text{ and } e_2 \neq e_1(k)\right\}$
  else
    $U_k \leftarrow \min\left\{h_k^1, h_k^{e_1} + h_k^{e_2}\right\}$
  end if
  $\Delta \leftarrow (U_k - L_k)/\kappa$
  $G_{max} \leftarrow 0$
  $\alpha_k \leftarrow L_k$
  $terminate \leftarrow$ **false**
  while $(terminate =$ **false**$)$ **do**
    estimate $G(\alpha_k)$
    if $(G(\alpha_k) < G_{max}$ and $\alpha_k = U_k)$ **then**
      $terminate \leftarrow$ **true**
    else
      $G_{max} \leftarrow G(\alpha_k)$
      $\alpha_k \leftarrow \alpha_k + \Delta$
    end if
  end while
end do

4.3 Elimination Tests

The efficiency of the Benders decomposition algorithm can be improved by reducing the size of the original model. By doing so, both the master problem and the subproblem can be solved more efficiently. Moreover, the convergence of the algorithm can also benefit from the solution space reduction. In Section 2 we have presented several optimal UHLPMA properties that can help reduce the size of the model prior to the solution process. Nevertheless, the number of variables and constraints remains very large in large-scale instances.

The size of the model can be further reduced by exploiting the information obtained during the inner iterations of the Benders algorithm. In this section, we develop two different reduction tests capable of eliminating variables which are known not to appear in an optimal solution. Reduction tests have been successfully applied for other HLPs in the context of Lagrangean relaxation (Contreras et al., 2009; Contreras et al., 2010). To the best of our knowledge, the idea of using reduction tests within a Benders decomposition algorithm is new.

The first reduction test uses lower and upper bounds on the optimal solution value to check whether a node may appear in an optimal solution. It exploits the primal information generated during the inner iterations of the Benders algorithm to obtain an estimation of the location and transportation costs associated with feasible solutions containing a hub located at a given node. Using this estimation, we can sometimes determine that the node will not be chosen as a hub. Let MP^t_{LP} denote the linear relaxation of MP^t, $v(MP^t_{LP})$ its optimal solution value, and rc_i the reduced cost associated with variable z_i. The following result provides a reduction test for closing a hub node.

Proposition 4 *Let UB be an upper bound on the optimal solution value of MP. If z_i is a nonbasic variable in the optimal solution to MP^t_{LP} and $v(MP^t_{LP}) + rc_i > UB$, then $z_i = 0$ in any optimal solution.*

Proof The results follows from the fact that $v(MP^t_{LP})+rc_i$ is a lower bound on the objective function value if a hub is located at node i. Therefore, if $v(MP^t_{LP}) + rc_i > UB$, then $z_i = 0$ in any optimal solution. ■

After applying this test, H is updated by removing the eliminated nodes from it. The corresponding node and edge variables are also eliminated from the model.

The second reduction test uses a stronger lower bound that allows checking whether any node in a set of candidate hub nodes $Q \subset H$ may appear in an optimal solution. By solving a slightly modified MP^t, we can obtain an estimation of the total cost associated to feasible solutions containing at least one hub located at a node contained in Q. Using this estimation, we can determine whether $z_i = 0$ for all $i \in Q$ in every optimal solution. We

define the following modified master problem $\mathrm{MP}^t(Q)$:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in H} f_i z_i + \sum_{i\in H} \eta_i \\
\text{subject to} \quad & \eta_i \geq \sum_{k\in K_i} \alpha_k^t - \sum_{i\in H}\sum_{k\in K_i} u_{ik}^t z_i \qquad \forall i \in H,\ (\alpha,u) \in P_{D^i}^t && (61)\\
& \sum_{i\in Q} z_i \geq 1 && (62)\\
& z_i \in \{0,1\} \qquad \forall\, i \in H. && (63)
\end{aligned}$$

The following result provides the reduction test for closing a set of hub nodes.

Proposition 5 *Let UB be an upper bound on the optimal solution value of MP. If* $v(MP^t(Q)) > UB$*, then* $z_i = 0$ *for each* $i \in Q$ *in any optimal solution.*

Proof The result follows from the fact that $v(MP^t(Q))$ is a lower bound on the objective function value if a hub is located at some node $i \in Q$. Therefore, if $v(MP^t(Q)) > UB$, then $z_i = 0$ for each $i \in Q$ in any optimal solution. ■

For a particular set $Q \subset H$, the previous test requires the solution of an integer linear program. Therefore, we must carefully choose a candidate set Q containing the largest possible number of nodes, while yielding a lower bound strong enough to close the hub nodes. In particular, we want to exclude nodes associated with good feasible solutions of $\mathrm{MP}^t(Q)$ having an objective function value inferior to the upper bound. If we generate a set Q failing the test, we must remove elements from Q so that the resulting set improves the lower bound and passes the test.

The efficiency of the previous test also relies on the quality of the approximation of MP^t. Thus, we should apply the test once we have constructed a sufficiently good approximation of MP. At the beginning of the Benders algorithm we set $Q = H$. Then, at iteration t of the algorithm we discard from Q the set of open hub nodes from the optimal solution of MP^t (i.e., Q is updated to $Q \setminus \{i \in Q : z_i^t = 1\}$) as well as the nodes that may have been eliminated through the first test.

When we perform the second elimination test with $\mathrm{MP}^t(Q)$ and it fails, we eliminate from Q the set of open hub nodes form an optimal solution, denoted by $z^t(Q)$, i.e., Q is updated to $Q \setminus \{i : z_i^t(Q) = 1\}$. We also eliminate from Q the nodes having small reduced costs $\hat{c}_i$ associated to the LP relaxation, denoted by $\mathrm{MP}_{LP}^t(Q)$, of $\mathrm{MP}^t(Q)$. In particular, we eliminate from Q nodes such that $\hat{c}_i < \nu \times c_{max}$, where c_{max} is the maximum reduced cost associated to nonbasic variables and ν is a control parameter such that $0 < \nu < 1$. These previous nodes are eliminated from Q only when the gap between the upper bound and the optimal solution value of $\mathrm{MP}_{LP}^t(Q)$ exceeds a theshold ζ. The proposed procedure is summarized in Algorithm 5.

Algorithm 5 Elimination test for Q

$terminate \leftarrow$ **false**
$Q \leftarrow H \setminus \{i : z_i^r = 1, r = 1, \ldots, t\}$
while ($terminate =$ **false**) **do**
  Solve $\text{MP}_{LP}^t(Q)$ to obtain $\hat{c}_i$
  if $((UB - v(MP_{LP}^t(Q))/UB > \zeta)$ **then**
    $c_{max} \leftarrow \max\{\hat{c}_i : i \in Q\}$
    $Q \leftarrow Q \setminus \{i : \hat{c}_i < \nu \times c_{max}\}$
  end if
  Solve $\text{MP}^t(Q)$ to obtain $z^t(Q)$
  if $(v(MP^t(Q)) > UB)$ **then**
    $terminate \leftarrow$ **true**
  else
    $Q \leftarrow Q \setminus \{i : z_i^t(Q) = 1\}$
  end if
end while
$H \leftarrow H \setminus Q$

4.4 A Heuristic Procedure for the UHLPMA

In our Benders reformulation, we know that any vector $z \in Z$ such that $\sum_{i \in H} z_i \geq 1$ is a feasible solution for the MP and, thus, has at least one optimality cut associated to it. We can apply a heuristic to produce a diverse set of feasible solutions, yielding optimality cuts that are incorporated at the beginning of the algorithm. In fact, it is known that the use of an initial approximation of the Benders reformulation polyhedron has a major impact on the required number of iterations (see, e.g., Geoffrion and Graves, 1974; Cordeau et al., 2000). The heuristic procedure can also yield good upper bounds that improve the effectiveness of the reduction tests.

Here we present a simple, yet effective heuristic procedure capable of generating high quality solutions and diverse solutions which may provide useful optimality cuts. The proposed heuristic is composed of two phases: an *estimation* phase and an *intensification* phase. The estimation phase is an iterative procedure that constructs a set of initial feasible solutions which are used to construct an interval on the estimated number of open hub facilities in an optimal solution. The intensification phase is an iterative procedure that generates feasible solutions containing sets of open hubs whose cardinality lies in the interval obtained in the previous phase. Within each phase, we use a common constructive procedure that randomly constructs a feasible solution with a given number of open hub facilities, and improves it by means of a local search procedure. In what follows, we first explain the constructive procedure and we then present the overall heuristic.

4.4.1 Constructing Solutions.

Solutions are represented by pairs of the form $s = (H_1, H_0)$ where, as before, H_1 denotes the set of open hubs and H_0 denotes the set of closed hubs. During the inner iterations of both phases of the heuristic, a feasible solution $s^r = (H_1^r, H_0^r)$ is constructed by randomly

selecting a component $z^r \in Z$ such that $|H_1^r| = p$, where p is a fixed parameter. Once the set of open hubs is known, the associated flow routing subproblem is solved by using (23), and the objective value associated to s^r can be evaluated. Each generated solution is then improved by means of a local search procedure which considers three different neighborhoods. The first one considers a subset of feasible solutions that are obtained from the current one by opening a new hub facility. Then, $N_{open}(s) = \{s' = (H_1', H_0') : H_1' = H_1 \cup \{k\}, k \in H_0\}$. To explore $N_{open}(s)$, all nodes $k \in H_0$ are considered. The second one considers a subset of feasible solutions obtained from the current one by closing a hub facility. Then, $N_{close}(s) = \{s' = (H_1', H_0') : H_1' = H_1 \setminus \{k\}, k \in H_1\}$. To explore $N_{close}(s)$, all hub nodes $k \in H_1$ are considered. The last neighborhood examines a subset of feasible solutions obtained from the current one by opening a new facility and closing an open one. Thus, $N_{inter}(s) = \{s' = (H_1', H_0') : H_1' = H_1 \cup \{k_1\} \setminus \{k_2\}, k_1 \in H_0, k_2 \in H_1\}$. To explore $N_{inter}(s)$, all possible combinations of nodes $k_1 \in H_0$ and $k_2 \in H_1$ are considered. The local search procedure is described in Algorithm 6.

Algorithm 6 Local search procedure

terminate ← **false**
while (*terminate* = **false**) **do**
  Explore N_{close}
  if (solution has not been updated in N_{close}) **then**
    Explore N_{open}
  end if
  if (solution has not been updated in N_{close} and N_{open}) **then**
    Explore N_{inter}
  end if
  if (solution has not been updated) **then**
    terminate ← **true**
  end if
end while

4.4.2 Heuristic for the UHLPMA.

During the estimation phase of the heuristic, we construct a total of r_{max} feasible solutions, each one obtained by setting $p = 2$ and randomly generating a z^r vector such that $z^r \in \{z : z \in Z, |H_1^r| = p\}$. After the local search has been applied, the resulting best solution provides an idea of the number of hub facilities that are open in optimal solutions. Using these solutions, we construct a good interval on the required number of open hubs so that the intensification phase then focuses on generating solutions such that $|H_1^r| \in [p_{min}, p_{max}]$, where $p_{min} = \min\{|H_1^r| : r = 1, \ldots, r_{max}\}$ and $p_{max} = \max\{|H_1^r| : r = 1, \ldots, r_{max}\}$. In particular, for each $p \in [p_{min}, p_{max}]$ the intensification phase constructs r_{max} feasible solutions. The overall heuristic procedure is depicted in Algorithm 7.

Algorithm 7 Heuristic for the UHLPMA

$r \leftarrow 1$, $p \leftarrow 2$
while $(r < r_{max})$ **do**
  Randomly select $z^r \in \{z : z \in Z, |H_1^r| = p\}$
  Apply local search
  $r \leftarrow r + 1$
end while
$p_{min} \leftarrow \min\{|H_1^r| : r = 1, \ldots, r_{max}\}$
$p_{max} \leftarrow \max\{|H_1^r| : r = 1, \ldots, r_{max}\}$
$p \leftarrow p_{min}$
while $(p < p_{max})$ **do**
  $r \leftarrow 1$
  while $(r < r_{max})$ **do**
    Randomly select $z^r \in \{z : z \in Z, |H_1^r| = p\}$
    Apply local search
    $r \leftarrow r + 1$
  end while
$p \leftarrow p + 1$
end while

5 Computational Experiments

We now present the results of extensive computational experiments performed to assess the performance our algorithm. In the first part of the computational experiments, we focus on a comparison of different versions of the Benders decomposition algorithm to evaluate the impact of each of the proposed algorithmic features. The second part of the experiments is mainly devoted to a comparison between our exact method and several exact algorithms reported in the literature. In the third part of the experiments, we test the robustness and limitations of our method on large scale instances involving up to 500 nodes. All algorithms were coded in C and run on a Dell Studio PC with an Intel Core 2 Quad processor Q8200 running at 2.33 GHz and 8 GB of RAM under a Linux environment. The master problems of all versions of the algorithm were solved using the callable library CPLEX 10.1.

We have used the well-known Australian Post (AP) set of instances to perform the first two parts of the computational experiments. This data set is the most commonly used in the hub location literature (*mscmga.ms.ic.ac.uk/jeb/orlib/phubinfo.html*). It consists of the Euclidean distances c_{ij} between 200 cities in Australia, of a computer code to reduce the size of the set by grouping cities, and of the values of W_k representing postal flows between pairs of cities. Each instance has a strictly positive flow between every pair of nodes. Therefore, the number of considered commodities is given by $|K| = |H|^2$. From this set of instances, we have selected those with $|H| = 25$, 50, 75, 100, 125, 150, 175 and 200 and with set-up costs of the type loose (L) (see Contreras et al., 2010, for details). We have varied the required number of open hub nodes in an optimal solution by increasing the distances of a particular instance as $d_{ij} = TC \times c_{ij}$ for each pair $(i, j) \in H \times H$, where TC is a scaling parameter for the transportation costs. For each instance size we have generated

nine different instances corresponding to different combinations of values for the inter-hub discount factor $\tau \in \{0.2, 0.5, 0.8\}$ and the transportation cost scaling factor $TC \in \{2, 5, 10\}$. In all these instances, we have considered $\chi = 1$ and $\delta = 1$.

In preliminary experiments, we have used the AP instances to set the values of the parameters of the algorithm. The following values were used in all our tests: $\psi = 0.5$, $\gamma = 0.5$, $\kappa = 10$, $\nu = 0.25$, $\zeta = 0.002$, $r_{max} = 10$, and $z_i^0 = 0.1$ for each $i \in H$. In the first two parts of the experiments, the Benders decomposition algorithm terminated when one of the following criteria was met: *i)* the optimality gap between the upper and lower bounds was below a threshold value ϵ, i.e. $|ub - lb|/ub < \epsilon$, *ii)* the maximum number of iterations $Iter_{max}$ was reached or, *iii)* the maximum time limit $Time_{max}$ was reached. We set the parameter values as $\epsilon = 10^{-6}$, $Iter_{max} = 1000$ and $Time_{max} = 7,200$ seconds.

5.1 Analysis of Algorithmic Refinements

The aim of the first part of the computational experiments is to analyze the effectiveness of each of the algorithmic refinements proposed in Section 4. For presentation purposes, we only include summarized results of all experiments. The interested reader is referred to the Online Supplement for the detailed results.

We first focus on analyzing the benefits of using the multicut Benders reformulation over the standard Benders reformulation. We have implemented two different versions of Algorithm 1. The first one, called *1-cut*, uses the reformulation (16)–(18) in which only one optimality cut is added per iteration. The second one, called $|H|$*-cut*, uses the stronger reformulation (17), (18), (32) in which $|H|$ optimality cuts are added per iteration. Both algorithms use Algorithm 2 to generate the optimality cuts at each iteration. The results of the comparison are summarized in Table 1. The first column gives the number of nodes associated to each group of instances. The next two columns under the heading *Optimal Found* give the number of optimal solutions found for *1-cut* and $|H|$*-cut*, respectively. The next two columns under the heading *Average Time (sec)* give the average CPU time in seconds needed to obtain an optimal solution of the problem by using *1-cut* and $|H|$*-cut*, respectively. The last two columns under the heading *Average Iterations* provide the required number of iterations for each of the algorithms to converge.

Table 1: Comparison of Benders reformulations.

	Optimal found		*Average time (sec)*		*Average iterations*	
$\|H\|$	*1-cut*	$\|H\|$*-cuts*	*1-cut*	$\|H\|$*-cuts*	*1-cut*	$\|H\|$*-cuts*
25	9/9	9/9	8.43	0.67	65.00	9.78
50	9/9	9/9	78.37	4.28	98.11	11.67
75	9/9	9/9	158.37	7.20	87.78	11.11
100	8/9	9/9	1727.41	54.27	171.11	15.67
125	8/9	9/9	950.24	68.64	119.22	13.56
150	8/9	9/9	1040.05	166.83	134.11	15.22
175	7/9	9/9	1885.27	325.66	112.89	12.67
200	7/9	8/9	2069.38	1340.87	118.67	18.56
Average	65/72	71/72	989.69	246.05	113.36	13.53

Table 1 shows that both algorithms *1-cut* and $|H|$*-cut* are able to solve most instances within two hours. However, the strong multicut reformulation is able to solve 71 out of the 72 considered instances whereas the standard Benders reformulation can solve only 65. The

columns *Average time (sec)* indicate that $|H|$-*cut* requires on average much less computation time than *1-cut*. Moreover, as can be seen in the *Average iterations* columns, the convergence of the Benders algorithm is greatly improved by using $|H|$-*cut*. The number of required iterations to converge is reduced by a factor of 10 on average. Given that algorithm $|H|$-*cut* clearly outperforms *1-cut*, we only consider the multicut Benders reformulation in the remainder of the computational experiments.

We next focus on analyzing the effectiveness of generating stronger, possibly undominated, optimality cuts. In particular, we have implemented three different versions of Algorithm 1. The first version, referred to as *NC*, uses the optimality cuts obtained from Algorithm 2. The second version, referred to as *POC*, uses the Pareto-optimal cuts obtained when solving PO^t by using the dual simplex algorithm of CPLEX 10.1. The third version, referred to as *SC*, uses the strong optimality cuts obtained from Algorithm 4. The results of the comparison between these algorithms are summarized in Table 2. The three columns under the heading *Optimal found* give the number of optimal solutions found for each of the considered algorithms. The next columns provide computing times and iterations counts for each version.

Table 2: Comparison of optimality cuts.

	Optimal found			*Average time (sec)*			*Average iterations*		
$\|H\|$	*NC*	*POC*	*SC*	*NC*	*POC*	*SC*	*NC*	*POC*	*SC*
25	9/9	9/9	9/9	0.67	1.13	0.50	9.78	6.78	7.89
50	9/9	9/9	9/9	4.28	9.05	2.10	11.67	7.67	8.00
75	9/9	9/9	9/9	7.20	32.75	5.91	11.11	7.00	7.89
100	9/9	9/9	9/9	54.27	126.78	21.28	15.67	8.22	9.56
125	9/9	9/9	9/9	68.64	366.86	56.72	13.56	9.44	10.67
150	9/9	9/9	9/9	166.83	766.42	123.23	15.22	9.89	11.89
175	9/9	9/9	9/9	325.66	1130.43	254.76	12.67	8.33	9.56
200	8/9	9/9	9/9	1340.87	2275.53	738.05	18.56	10.89	13.78
Average	71/72	72/72	72/72	262.17	588.62	150.32	13.53	8.53	9.90

The results of Table 2 confirm the efficiency of generating stronger optimality cuts. Both the *POC* and *SC* algorithms are able to obtain the optimal solution of all considered instances within two hours of computation time. However, the larger CPU time needed to solve the PO^t problems using *POC* does not compensate for the improvements in convergence, even for the small size instances. As can be seen in the *Average time (sec)* columns, *SC* is considerably more efficient than *NC* and *SC*. Even though *SC* generates optimality cuts that are not necessarily Pareto-optimal, these seem to be stronger than those used in *NC*. The *Average iterations* columns also confirm that the convergence of the Benders algorithm can be improved by using *SC*, but this version is slightly worse than *POC*. Given that algorithm *SC* clearly outperforms *NC* and *POC*, we only consider the generation of optimality cuts with Algorithm 4 in the rest of the experiments.

We next focus on analyzing the performance of the heuristic described in Section 4.4 and its contribution to the convergence of the Benders decomposition algorithm. As mentioned, we can use the solutions obtained from the heuristic procedure to generate a promising initial set of optimality cuts and generate a good approximation of the MP. We first use Algorithm 7 to generate a *diverse* set of feasible solutions, called I, where potential structural cuts can be selected to generate initial optimality cuts. Let $I_p \subset I$ denote the subset of feasible solutions containing exactly p open hub nodes, and let p_{ub} denote the cardinality of the best

solution contained in I. We construct the initial set of cuts, denoted by P_D^I, by selecting solutions from different sets I_p. More specifically, we select the best solution from each set I_p such that $p \in \{1, p_{ub} - r_2, \ldots, p_{ub}, \ldots, p_{ub} + r_2\}$, where $r_2 > 0$ is a parameter controlling the range of the selected p values.

We have tested three different version of Algorithm 1. The first version uses no initial cuts at all, i.e. $|P_D^I| = 0$. The second version uses an initial set P_D^I containing only one cut generated from the best solution obtained from the heuristic. The third version uses an initial set P_D^I containing five different cuts generated by setting $r_2 = 2$. The results of the heuristic procedure as well as the comparison of three algorithms are summarized in Table 3. The column under the heading *Optimal Found* gives the proportion of optimal solutions found by the heuristic procedure for each group of instances. The column *Average % dev.* gives the average percent deviation between the best solution found by the heuristic and the optimal solution, i.e. $\%\ dev = 100(UB_H - OPT)/(UB_H)$, where OPT is the optimal value and UB_H is the best upper bound obtained with Algorithm 7. The remaining column headings are self-explanatory.

Table 3: Effects of using the heuristic procedure.

	Optimal	*Average*	*Average time (sec)*				*Average iterations*		
$\|H\|$	*found*	*% dev.*	*Heur*	$\|P_D^I\| = 0$	$\|P_D^I\| = 1$	$\|P_D^I\| = 5$	$\|P_D^I\| = 0$	$\|P_D^I\| = 1$	$\|P_D^I\| = 5$
25	9/9	0.00	0.06	0.50	0.24	0.24	7.89	5.22	6.00
50	9/9	0.00	0.44	2.10	1.79	1.77	8.00	5.78	7.00
75	9/9	0.00	1.48	5.91	5.41	5.80	7.89	6.11	7.67
100	9/9	0.00	3.57	21.28	19.62	22.16	9.56	8.44	10.22
125	7/9	0.02	6.11	56.72	52.78	57.01	10.67	9.56	11.22
150	8/9	0.02	14.48	123.23	110.64	115.28	11.89	9.78	11.22
175	9/9	0.00	16.40	254.76	232.30	244.22	9.56	8.33	9.33
200	8/9	0.03	33.44	738.05	679.33	652.30	13.78	12.44	13.44
Average	68/72	0.01	9.50	150.32	137.76	137.35	9.90	8.21	9.51

The results of Table 3 confirm the efficiency of the heuristic procedure. It is able to find the optimal solution in 68 out of the 72 tested instances. Moreover, for the instances in which the optimal solution could not be found, the percent deviation never exceeds 0.3%. The fact that our heuristic is not very sophisticated and yet able to produce very good results, indicates that the UHLPMA is a problem for which good solutions can be obtained easily. However, proving optimality remains challenging. The computational time required by the heuristic to produce a good solution is just a fraction of the time needed for the Benders algorithm to obtain the optimal solution. The columns *Average iterations* confirm that the convergence of the Benders algorithm can be further improved by using good feasible solutions to obtain an initial set of cuts. However, the best results are obtained when considering only the best solution produced by the heuristic. Although the improvement in computational time is relatively small for these instances, we will show later that it becomes more important on the larger size instances. Furthermore, the best upper bound provided by the heuristic also has a positive effect on the performance of the elimination tests, as we will show.

We now analyze the effect of incorporating the elimination tests of Section 4.3 into the Benders decomposition algorithm. We have implemented four versions of the algorithm. The first two consider the case in which no initial cuts are generated, i.e. $|P_D^I| = 0$, whereas the last two work with $|P_D^I| = 1$. Moreover, the first and the third versions only execute the first

reduction test, denoted as EI, whereas the second and fourth version execute both the first set EI and the second test, called EII, performed by applying Algorithm 5. The test EI is applied at every iteration of the Benders algorithm whereas EII is applied once the relative optimality gap is below 1% and is performed only once. The results of the elimination tests are summarized in Table 4. The headings of this table are again self-explanatory.

Table 4: Effects of elimination tests.

	Average time (sec)					*Average iterations*				
		$\|P_D^I\| = 0$		$\|P_D^I\| = 1$			$\|P_D^I\| = 0$		$\|P_D^I\| = 1$	
$\|H\|$	*No tests*	*E I*	*EI + EII*	*E I*	*EI + EII*	*No tests*	*EI*	*EI + EII*	*EI*	*EI + EII*
25	0.50	0.35	0.33	0.29	0.31	7.89	7.22	6.89	5.11	5.11
50	2.10	2.12	2.14	2.80	2.94	8.00	7.78	7.89	6.33	6.56
75	5.91	5.29	5.42	7.13	7.08	7.89	7.00	7.11	6.33	6.44
100	21.28	19.52	20.81	22.00	24.52	9.56	9.33	9.44	7.67	8.44
125	56.72	55.05	53.69	54.37	56.25	10.67	10.44	10.22	9.89	9.00
150	123.23	105.54	108.94	107.24	118.24	11.89	10.56	10.11	10.22	8.44
175	254.76	236.17	250.11	243.14	249.27	9.56	9.11	9.33	9.22	8.00
200	738.05	673.49	590.76	632.12	540.32	13.78	13.11	12.11	12.67	10.00
Average	150.32	137.19	129.02	133.64	124.87	9.90	9.32	9.14	8.43	7.75

The columns *Average iterations* show that for small size instances, the improved convergence of the algorithms with the elimination tests does not yield much lower CPU times. However, for the larger 200 node instances, improved convergence translates into shorter computational time, particularly with $|P_D^I| = 1$ and *EI + EII*. Observe that the average required CPU time for this version is 73% of the required CPU time without any reduction tests and initial cuts. In the last part of the computational experiments, we will perform additional experiments to confirm and assess the efficiency of the elimination tests and the heuristic on more difficult and larger instances.

5.2 Comparison with Alternative Solution Methods

We now present a comparison between our best version of the Benders decomposition algorithm and several exact solution methods previously proposed in the literature. In particular, we compare our exact method with the following five exact algorithms: *i)* the Benders decomposition algorithm of Camargo et al. (2008), *ii)* the dual adjustment procedure developed by Cánovas et al. (2007), *iii)* the relax-and-cut algorithm proposed by Marín (2005), *iv)* the solution of a flow-based formulation using CPLEX as described in Boland et al. (2004), and *v)* the solution of the strong path-based formulation presented in Section 2.1, using CPLEX. To provide a fair comparison, we have run all algorithms on the same computer. The dual adjustment procedure and the relax-and-cut algorithm were obtained from their respective authors, whereas the remaining algorithms were coded by us.

The detailed results of the comparison between the exact methods using the AP data set are provided in Table 5. The first three columns give the number of nodes, the discount factor and the transportation scale factor. The remaining columns give the CPU time in seconds needed to obtain an optimal solution for each exact algorithm. The *Benders* column provides the results obtained with the best version of our Benders decomposition algorithm. Whenever a solution method cannot optimally solve an instance within two hours of CPU time, we write *time* in the corresponding entry of the table. If an algorithm runs out of memory we then write *memory*.

Table 5: Comparison of exact methods with AP instances from 25 to 200 nodes.

			Total time (sec)					
$\|H\|$	τ	TC	*CPLEX*	*Boland et al. (2004)*	*Marín (2005)*	*Cánovas et al. (2007)*	*Camargo et al. (2008)*	*Benders*
25	0.2	2	1.69	1.37	5.24	0.30	0.24	0.11
	0.2	5	1.36	3.59	2.13	0.19	4.01	0.30
	0.2	10	0.80	2.20	3.07	0.28	73.00	0.85
	0.5	2	1.12	13.78	2.38	0.11	0.20	0.08
	0.5	5	5.29	18.35	3.74	0.29	1.76	0.56
	0.5	10	0.82	5.63	1.29	0.10	10.78	0.44
	0.8	2	0.99	36.22	2.04	0.09	0.19	0.06
	0.8	5	1.04	17.83	1.01	0.06	0.56	0.17
	0.8	10	0.93	9.83	1.11	0.11	2.54	0.25
50	0.2	2	874.46	526.70	786.14	21.36	4.18	0.96
	0.2	5	159.27	644.38	1292.35	6.39	20.38	1.76
	0.2	10	57.70	218.54	80.64	12.34	624.41	11.75
	0.5	2	61.62	4982.49	119.75	5.20	2.65	0.85
	0.5	5	102.06	4128.70	72.27	5.52	12.04	1.76
	0.5	10	54.58	1030.34	50.36	6.20	67.16	3.50
	0.8	2	34.34	time	80.91	2.54	2.33	0.80
	0.8	5	61.31	6406.17	28.09	1.31	5.35	0.87
	0.8	10	166.38	2471.61	39.33	7.87	35.04	2.24
75	0.2	2	memory	6817.61	time	104.31	27.43	4.73
	0.2	5	memory	2579.95	time	86.34	73.94	5.80
	0.2	10	memory	3045.77	time	61.12	2208.01	19.06
	0.5	2	memory	time	time	40.89	21.40	4.78
	0.5	5	memory	time	time	59.71	53.98	5.83
	0.5	10	memory	time	time	24.14	129.12	7.72
	0.8	2	memory	time	time	35.03	17.02	4.30
	0.8	5	memory	time	time	12.52	50.29	4.43
	0.8	10	memory	time	time	37.82	77.54	7.05
100	0.2	2	memory	time	time	568.43	210.72	15.14
	0.2	5	memory	time	time	1196.08	1364.60	28.09
	0.2	10	memory	time	time	1244.14	time	58.48
	0.5	2	memory	time	time	182.37	165.75	13.81
	0.5	5	memory	time	time	417.31	787.01	22.61
	0.5	10	memory	time	time	1762.05	4586.82	34.77
	0.8	2	memory	time	time	102.84	126.38	13.68
	0.8	5	memory	time	time	155.74	259.70	15.32
	0.8	10	memory	time	time	415.01	1198.83	18.79
125	0.2	2	memory	time	memory	2399.08	747.44	47.77
	0.2	5	memory	time	memory	1584.67	2406.95	56.34
	0.2	10	memory	time	memory	2691.29	time	112.20
	0.5	2	memory	time	memory	625.49	503.83	38.23
	0.5	5	memory	time	memory	836.01	2150.63	49.51
	0.5	10	memory	time	memory	3420.10	time	73.52
	0.8	2	memory	time	memory	179.88	459.49	36.78
	0.8	5	memory	time	memory	547.98	1061.16	43.55
	0.8	10	memory	time	memory	1556.03	time	48.36
150	0.2	2	memory	time	memory	memory	2360.82	121.83
	0.2	5	memory	time	memory	memory	time	123.90
	0.2	10	memory	time	memory	memory	2360.82	121.83
	0.5	2	memory	time	memory	memory	1814.04	96.16
	0.5	5	memory	time	memory	memory	time	112.44
	0.5	10	memory	time	memory	memory	time	135.71
	0.8	2	memory	time	memory	memory	1278.42	86.26
	0.8	5	memory	time	memory	memory	3816.60	91.66
	0.8	10	memory	time	memory	memory	6638.50	105.07
175	0.2	2	memory	memory	memory	memory	2062.87	237.55
	0.2	5	memory	memory	memory	memory	time	256.08
	0.2	10	memory	memory	memory	memory	time	525.67
	0.5	2	memory	memory	memory	memory	1776.36	196.25
	0.5	5	memory	memory	memory	memory	6575.61	206.45
	0.5	10	memory	memory	memory	memory	time	263.70
	0.8	2	memory	memory	memory	memory	1425.14	176.31
	0.8	5	memory	memory	memory	memory	3621.65	184.16
	0.8	10	memory	memory	memory	memory	time	197.27
200	0.2	2	memory	memory	memory	memory	time	483.91
	0.2	5	memory	memory	memory	memory	time	485.47
	0.2	10	memory	memory	memory	memory	time	1271.34
	0.5	2	memory	memory	memory	memory	time	393.36
	0.5	5	memory	memory	memory	memory	time	394.65
	0.5	10	memory	memory	memory	memory	time	750.23
	0.8	2	memory	memory	memory	memory	time	338.44
	0.8	5	memory	memory	memory	memory	time	361.97
	0.8	10	memory	memory	memory	memory	time	383.49

The results of Table 5 clearly indicate that our exact method outperforms all previously proposed methods. Observe that our algorithm is able to solve all 72 instances whereas the algorithm of Camargo et al. (2008) is only able to optimally solve 52 within two hours of CPU time. The dual adjustment approach by Cánovas et al. (2007) is able to find the optimal solution in 45 out of the 72 instances, and the relax-and-cut approach of Marín (2005) can only solve instances with up to 50 nodes. Similar results are obtained when solving the flow-based formulation of Boland et al. (2004) and the path-based formulation (5)–(9) with CPLEX. This is a clear indication of the limitations of using a commercial solver to solve the UHLPMA. In the case of the path-based model, eight GB of memory are not sufficient to load the model into CPLEX when $|H| > 50$. With the flow-based model, larger size instances can be loaded into CPLEX, but their weaker LP bounds do not allow solving instances with more than 75 nodes within two hours of CPU time. It can be seen that our algorithm is always at least one order of magnitude faster than the other exact methods, with the exception of the small instances involving 25 nodes.

5.3 A New Data Set

In the previous experiments, we have shown that the largest size instances of the AP set, containing 200 nodes, can be optimally solved by our algorithm within less than 20 minutes of CPU time. Given that this set contains the largest size instances currently available, we have generated a set of larger instances in order to test the robustness and limitations of our Benders decomposition algorithm.

At this stage, some comments on the structure of flows in the AP set are in order. We have observed that the amount of flow originating at each node is highly variable in every instance of this set: all instances have a very small number of nodes for which the outgoing flow is much larger than for the other nodes. For instance, the 200 nodes instance of the AP set has one node generating 15% of the total flow of the network, another generating 7%, and the remaining ones each generating less than 1%. This situation seems to make the solution of these instances rather easy since very few nodes have a large impact on the overall cost of the network and thus greatly influence the hub location decisions. As we will show next, instances in which the outgoing flow of each node is within a narrow range are considerably more difficult to solve.

For this reason, we introduce three different sets of instances with diverse structural characteristics in the flow network. In particular, we consider different levels of magnitude for the amount of flow originating at a given node to obtain three different sets of nodes: low-level (LL) nodes, medium-level (ML) nodes, and high-level (HL) nodes. The total outgoing flow of LL, ML and HL nodes lies in the interval $[1, 10]$, $[10, 100]$, and $[100, 1000]$, respectively. Using these nodes, we generate three different classes of instances. In the first set of instances, called *Set I*, the number of HL, ML, and LL nodes is 2%, 38% and 60% of the total number of nodes, respectively. In the second set, called *Set II*, we construct an instance is such a way that the number of HL, ML, and LL nodes is 30%, 35% and 35% of the total number of nodes, respectively. Finally, in the third set, called *Set III*, the number of HL, ML, and LL nodes is 0%, 1% and 99% of the total number of nodes, respectively. In *Set I* we generate instances with $|H| = 50$, 100, 150, 200, 250, 300, 350, 400, 450 and 500. In *Set II* and *Set III*, we generate instances with $|H| = 50$, 100, 150, and 200. For each value of

n in each set, we randomly generate the (x, y)-coordinates of the nodes from a continuous uniform distribution in $[0, 1000] \times [0, 1000]$ and define the distance between pairs of nodes as the Euclidean distance. We generate the fixed costs for the hub facilities as $f_i = \theta \times AD$, where $\theta \sim U[0.3, 0.8]$ and $AD = \sum_{k \in K} W_k$. Finally, for each basic instance we generate nine instances corresponding to different combinations of values for the inter-hub discount factor $\tau \in \{0.2, 0.5, 0.8\}$ and the transportation costs scale factor $TC \in \{2, 5, 10\}$. Therefore, *Set I* contains a total of 90 instances whereas sets *Set II* and *Set III* contain 36 instances each.

In these final computational experiments, we further analyze and evaluate the performance of the algorithmic refinements, especially the heuristic procedure and the elimination tests. To this end, we consider two different versions of Algorithm 1. The first version, referred to as $B1$, uses the multicut reformulation and the strong optimality cuts obtained from Algorithm 4. However, it does not include the initial cuts nor the elimination tests. The second version, referred to as $B2$, uses the multicut reformulation, the strong cuts, an initial cut associated to the best upper bound found by Algorithm 7, and the two elimination tests. Because of the increase in instance size, we have extended the CPU time limit to one day, i.e. $Time_{max} = 86,400$ seconds.

Computational results are summarized in Tables 6, 7 and 8. The columns *Optimal found* give the number of optimal solutions found by the heuristic, $B1$ and $B2$. The columns *Average % gap* provide the average percent deviation between the best upper and lower bounds, for the heuristic, B1 and B2 when the optimal solution cannot be found within the given time limit. That is % $gap = 100(UB_T - LB_T)/(UB_T)$, where UB_T and LB_T are the upper an lower bounds, respectively, obtained with $T = B1, B2$. The columns *Average time (sec)* provide the average CPU time in seconds needed to obtain an upper bound, in the case of the heuristic, and an optimal solution of the problem by using $B1$ and $B2$, respectively. The columns *Average iterations* give the average number of iterations for $B1$ and $B2$. The column *% Closed hubs* gives the average percent of hubs that were closed by the reduction tests in $B2$.

Table 6: Summary results of 54 instances of *Set I* with $|H| = 50$, 100, 150, 200, 250 and 300.

	Optimal found			*Average % gap*			*Average time (sec)*			*Average iterations*		*% Closed*
$\|H\|$	*Heur*	$B1$	$B2$	*Heur*	$B1$	$B2$	*Heur*	$B1$	$B2$	$B1$	$B2$	*hubs*
50	7/9	9/9	9/9	0.08	0.00	0.00	0.62	1.69	2.11	9.89	8.00	70.89
100	9/9	9/9	9/9	0.00	0.00	0.00	4.52	10.21	12.67	9.33	7.89	79.11
150	9/9	9/9	9/9	0.00	0.00	0.00	21.92	80.44	72.41	14.11	11.33	84.37
200	8/9	9/9	9/9	0.03	0.00	0.00	39.80	321.43	236.49	14.89	11.00	86.06
250	8/9	9/9	9/9	0.00	0.00	0.00	114.58	4976.90	2597.97	24.56	17.33	82.89
300	7/9	8/9	8/9	0.12	0.07	0.07	169.05	15380.02	8832.61	35.67	30.33	74.41
Average	48/54	53/54	53/54	0.04	0.01	0.01	58.41	3461.78	1959.04	18.07	14.31	79.62

Table 6 shows that both $B1$ and $B2$ algorithms are able to obtain an optimal solution in all instances, except one. The relative gap in the remaining instance is 0.67% for $B1$ and 0.59% for $B2$. The heuristic reaches an optimal solution 48 times out of 54. Moreover, the percent deviation in the instances in which the optimal solution could not be found never exceeds 0.8% and the total average deviation is 0.04%. Algorithm $B2$ is clearly faster than $B1$ on large-scale instances. On 250-nodes and 300-nodes instances, the average CPU time is reduced by half when using the reduction tests and the heuristic procedure. Also, from the *Average iterations* columns we observe that the convergence of the Benders algorithm can be

improved by incorporating these features. Moreover, column *% Closed hubs* indicates that a considerable number of candidate hub nodes can be eliminated by using the elimination tests. The percent of closed hubs ranges from 46% to 98%, with an average of 79.62%.

Table 7: Summary results of 36 instances of *Set II* with $|H| = 50$, 100, 150, and 200.

	Optimal found			*Average % gap*			*Average time (sec)*			*Average iterations*		*% Closed*
$\|H\|$	*Heur*	$B1$	$B2$	*Heur*	$B1$	$B2$	*Heur*	$B1$	$B2$	$B1$	$B2$	*hubs*
50	9/9	9/9	9/9	0.00	0.00	0.00	0.86	3.00	3.81	8.11	7.22	69.33
100	8/9	9/9	9/9	0.06	0.00	0.00	8.20	27.23	21.61	14.44	11.00	74.56
150	7/9	9/9	9/9	0.09	0.00	0.00	24.80	1729.73	601.38	23.89	18.22	78.22
200	6/9	7/9	7/9	0.09	0.08	0.06	71.98	1880.77	861.39	37.78	33.89	71.78
Average	30/36	34/36	34/36	0.06	0.02	0.02	26.46	910.18	372.05	21.06	17.58	73.47

Similar observations can be drawn from Table 7 for *Set II* instances. We observe that these instances are more difficult than those of *Set I* and the largest instances solved contain only 200 nodes. One possible explanation for this behavior is that the instances in *Set II* do not longer have the peculiarity that very few nodes generate a large proportion of the total flow of the network and thus, the decision of where to locate the hubs becomes much more difficult.

Table 8: Summary results of 36 instances of *Set III* with $|H| = 50$, 100, 150, and 200.

	Optimal found			*Average % gap*			*Average time (sec)*			*Average iterations*		*% Closed*
$\|H\|$	*Heur*	$B1$	$B2$	*Heur*	$B1$	$B2$	*Heur*	$B1$	$B2$	$B1$	$B2$	*hubs*
50	8/9	9/9	9/9	0.06	0.00	0.00	0.85	8.70	8.22	10.89	10.89	61.56
100	7/9	9/9	9/9	0.04	0.00	0.00	8.51	56.60	42.73	16.11	12.00	61.33
150	7/9	9/9	9/9	0.01	0.00	0.00	30.08	5373.33	1226.42	34.89	23.44	63.56
200	7/9	7/9	8/9	0.14	0.09	0.02	70.00	5912.76	2727.85	38.56	34.67	63.67
Average	29/36	34/36	35/36	0.06	0.02	0.00	27.36	2837.85	1001.31	25.11	20.25	62.53

Table 8 shows that $B2$ is still superior to $B1$ and that the instances of *Set III* are the most difficult of the test bed. This translates into a smaller percentage of closed hubs and into much longer CPU times.

To better analyze the limit of our algorithm, we have run a final series of computational experiments using the 36 instances of *Set I* with $|H| = 350$, 400, 450, and 500. Given that algorithm $B2$ has proven to be the best version of our Benders decomposition algorithm, these experiments were performed only with this variant.

The results of these experiments are summarized in Table 9. They confirm the efficiency and robustness of our algorithm on very large-scale instances. We have proved optimality of 26 out of the 36 considered instances. For the remaining instances, the relative duality gap is below 1%, with a maximum of 1.5% in one instance. The heuristic was able to obtain the optimal or best known solution in 25 cases out of 36, and the relative deviation for the remaining instances never exceeds 0.7%, except for one instance with 2.56%. From column *% Closed hub* we note that the elimination tests can again close a considerable number of candidate hub nodes. The percent of closed hubs ranges from 2% to 98%, with an average of 72.02%.

Table 9: Summary results of 36 instances of *Set I* with $|H| = 350$, 400, 450, and 500.

	Optimal found		*Average % gap*		*Average time (sec)*		*Average iterations*	*% Closed*
$\|H\|$	*Heur*	*B2*	*Heur*	*B2*	*Heur*	*B2*	*B2*	*hubs*
350	6/9	7/9	0.33	0.27	359.24	32141.39	30.33	69.11
400	8/9	8/9	0.04	0.03	610.69	41844.17	31.67	78.89
450	5/9	5/9	0.17	0.23	1100.41	19819.68	33.67	75.38
500	6/9	6/9	0.12	0.23	912.02	31108.20	23.71	64.69
Average	25/36	26/36	0.16	0.19	745.59	31228.36	29.85	72.02

6 Conclusions

We have presented an exact Benders decomposition algorithm for large-scale instances of the classical Uncapacitated Hub Location Problem with Multiple Assignments. A standard Benders decomposition was enhanced through the incorporation of several algorithmic features such as a multicut reformulation, the generation of stronger optimality cuts, the incorporation of reduction tests, and the use of a heuristic procedure. Extensive computational experiments on a large set of existing and new instances with up to 500 nodes and 250,000 commodities have clearly confirmed the efficiency and robustness of the algorithm. To the best of our knowledge, the new instances are by far the largest and most difficult ever solved for any type of hub location problem.

Acknowledgments

This work was partly founded by the Canadian Natural Sciences and Engineering Research Council under grants 227837-09 and 39682-10. This support is gratefully acknowledged.

References

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

Bazaraa, M.S., J.J. Jarvis, H.D. Sherali. 1990. Linear Programming and Network Flows, 2nd Edition, Wiley, New York.

Benders, J.F. 1962. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik* **4** 238–252.

Birge, J.R., F.V. Louveaux. 1988. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research* **34** 384–392.

Boland, N., M. Krishnamoorthy, A.T. Ernst, J. Ebery. 2004. Preprocessing and cutting for multiple allocation hub location problems. *European Journal of Operational Research* **155** 638–653.

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

Camargo, R.S., G. Miranda Jr., H.P. Luna. 2008. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research* **35** 1047–1064.

Campbell, J.F. 1994. A survey of network hub location. *Studies in Locational Analysis* **6** 31–43.

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

Campbell, J.F. 1996. Hub location and the p-hub median problem. *Operations Research* **44** 1–13.

Campbell, J.F., A.T. Ernst, M. Krishnamoorthy. 2002. Hub location problems. In: Z. Drezner, H.W. Hamacher (Eds.), *Facility Location. Applications and Theory*, Springer, Heidelberg, pp. 373-408.

Cánovas, L., S. Garcia, A. Marín. 2007. Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique *European Journal of Operational Research* **179** 990–1007.

Contreras, I., J.A. Díaz, E. Fernández. 2010. Branch and price for large-scale capacitated hub location problems with single assignment. *INFORMS Journal on Computing*. Forthcomming.

Contreras, I., J.A. Díaz, E. Fernández. 2009. Lagrangean relaxation for the capacitated hub location problem with single assignment, *Operations Research Spectrum*, **31** 483–505.

Cordeau, J.-F., F. Soumis, J. Desrosiers. 2000. A Benders decomposition approach for the locomotive and car assignment problem. *Transportation Science* **34** 133–149.

Cordeau, J.-F., F. Soumis, J. Desrosiers. 2001. Simultaneous assignment of locomotives and cars to passenger trains. *Operations Research* **49** 531–548.

Geoffrion, A.M., G.W. Graves. 1974. Multicommodity distribution system design by Benders decomposition. *Management Science*, **20** 822–844.

Hamacher, H.W., M. Labbé, S. Nickel, T. Sonneborn. 2004. Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics* **145** 104–116.

Klincewicz, J.G. 1996. A dual algorithm for the uncapacitated hub location problem. *Location Science* **4** 173–184.

Klincewicz, J.G. 1998. Hub location in backbone/tributary network design: A review. *Location Science* **6** 307–335.

Magnanti, T.L., R.T. Wong. 1981. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* **29** 464–484.

Marín, A., L. Canovas, M. Landete. 2005. New formulations for the uncapacitated multiple allocation hub location problem. *European Journal of Operational Research* **33** 393–422.

Marín, A. 2005. Uncapacitated Euclidean hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. *Journal of Global Optimization* **33** 393–422.

Mayer, G., B. Wagner. 2002. HubLocator: An exact solution method for the multiple allocation hub location problem. *Computers & Operations Research* **29** 715–739.

O'Kelly, M.E. 1986. The location of interacting hub facilities. *Transportation Science* **20** 92–106.

O'Kelly, M.E., H.J. Miller. 1994. The hub network design problem: A review and synthesis. *Journal of Transport Geography* **2** 31–40.

Online Supplement

In this Online Supplement we present the detailed computational results of Section 5. In particular, Tables 10 and 11 provide the computational results for the comparison of the Benders reformulations. The results for the comparison of algorithms for generating optimality cuts are given in Tables 12 and 13. The results of the heuristic procedure and some variants of the Benders algorithm containing different sets of initial optimality cuts are provided in Tables 14 and 15. The results of the elimination tests are given in Tables 16 and 17. Finally, Tables 18–21 provide the detailed computational results relative to the new set of instances. The headings of these tables are self-explanatory or have been defined in the paper.

Table 10: Comparison of Benders reformulations using 36 instances of AP set with $|H| =$ 25, 50, 75 and 100.

			Total time (sec)		*Iterations*		*Optimal solution*	
$\|H\|$	τ	*TC*	*1-cut*	$\|H\|$*-cuts*	*1-cut*	$\|H\|$*-cuts*	*Value*	*Set of hubs*
25	0.2	2	0.20	0.11	16	5	198152.69	8 18
	0.2	5	4.38	0.77	83	11	372917.47	2 8 15 16 18
	0.2	10	58.18	1.97	206	17	575514.42	2 4 5 8 11 15 17 18
	0.5	2	0.21	0.13	14	5	206559.35	8 18
	0.5	5	1.69	1.03	60	15	420252.84	2 8 14 18
	0.5	10	8.72	1.18	109	13	695200.36	2 5 8 14 16 18
	0.8	2	0.14	0.08	12	5	212227.82	8 18
	0.8	5	0.40	0.18	25	7	438906.58	8 18
	0.8	10	1.97	0.55	60	10	770835.99	2 8 14 16 18
50	0.2	2	1.94	1.11	28	8	198606.10	15 36
	0.2	5	12.02	2.58	90	11	384589.91	6 22 27 35
	0.2	10	618.99	15.13	352	20	629478.04	3 9 15 21 28 33 35
	0.5	2	1.06	0.82	18	6	206938.91	15 36
	0.5	5	4.93	3.12	62	13	425809.79	6 27 35
	0.5	10	49.66	6.95	173	16	730413.83	3 9 15 28 32 35
	0.8	2	0.91	0.82	15	6	211809.78	15 36
	0.8	5	1.72	1.22	30	8	443079.34	15 36
	0.8	10	14.09	6.73	115	17	794145.54	3 15 27 32 35
75	0.2	2	7.25	5.20	30	9	195761.92	22 68
	0.2	5	18.59	7.00	69	10	377893.29	8 40 47 52
	0.2	10	1323.30	16.23	361	16	629153.82	4 7 11 22 47 52 58
	0.5	2	4.79	4.47	23	8	204199.76	22 52
	0.5	5	8.07	6.99	49	12	419794.51	8 40 47 52
	0.5	10	38.82	7.15	116	11	724355.30	4 11 22 47 52 58
	0.8	2	3.96	4.50	19	9	208876.08	22 68
	0.8	5	6.85	6.08	51	12	442308.85	22 52
	0.8	10	13.66	7.19	72	13	787147.05	8 22 47 52 58
100	0.2	2	21.38	14.64	28	7	198620.55	29 73
	0.2	5	162.22	69.15	168	23	391144.24	29 44 54 71
	0.2	10	time	170.00	500	23	642832.63	5 18 29 52 58 64 70
	0.5	2	15.11	13.66	21	7	206992.89	29 73
	0.5	5	46.50	38.85	96	20	429862.06	29 54 71
	0.5	10	2514.49	113.32	500	24	747284.96	6 18 30 56 64 70
	0.8	2	13.27	12.91	17	6	211399.54	29 73
	0.8	5	16.45	17.54	34	11	447136.10	29 71
	0.8	10	125.05	38.38	176	20	802972.37	29 52 64 70

Table 11: Comparison of Benders reformulations using 36 instances of AP set with $|H| =$ 125, 150, 175 and 200.

			Total time (sec)		*Iterations*		*Optimal solution*	
$\|H\|$	τ	TC	*1-cut*	$\|H\|$*-cuts*	*1-cut*	$\|H\|$*-cuts*	*Value*	*Set of hubs*
125	0.2	2	60.07	47.20	25	10	195896.01	37 86
	0.2	5	117.79	52.92	84	12	377694.40	11 47 81 86
	0.2	10	time	170.82	383	25	628314.63	7 20 28 37 68 81 86
	0.5	2	41.22	36.63	18	6	203742.38	37 86
	0.5	5	73.36	53.26	72	16	419899.65	33 47 81 86
	0.5	10	806.1	137.54	305	25	732973.18	7 37 68 81 86
	0.8	2	36.25	35.70	16	6	208128.33	37 86
	0.8	5	43.60	38.06	48	9	441429.82	33 81 90
	0.8	10	100.06	45.66	122	13	790675.99	11 60 81 86
150	0.2	2	225.26	138.66	38	12	192280.70	40 100
	0.2	5	359.6	143.40	131	16	375646.21	11 80 94 100
	0.2	10	time	455.93	427	25	623773.71	7 25 40 62 82 94 100
	0.5	2	126.83	91.57	29	6	199546.92	40 99
	0.5	5	222.75	155.32	130	22	418433.02	12 80 94 100
	0.5	10	783.4	202.96	245	20	726703.57	11 25 40 82 94 100
	0.8	2	87.23	82.82	22	5	203945.31	40 99
	0.8	5	104.78	107.40	65	15	439397.54	40 100
	0.8	10	155.2	123.39	120	16	786644.08	11 40 82 94 100
175	0.2	2	430.36	287.08	21	7	188078.83	44 121
	0.2	5	648.64	276.74	102	15	372751.79	46 93 109 121
	0.2	10	time	1031.62	255	32	628214.33	8 30 44 93 109 121
	0.5	2	244.74	210.40	16	5	196283.17	44 121
	0.5	5	308.17	203.82	53	10	411927.96	46 93 109 121
	0.5	10	time	391.21	451	24	729241.06	8 30 44 93 109 121
	0.8	2	177.27	172.08	13	4	200335.63	44 121
	0.8	5	189.4	177.84	31	8	432513.74	46 93 121
	0.8	10	226.62	180.13	74	9	781209.12	46 93 108 121
200	0.2	2	1139.49	656.69	21	8	187460.90	53 184
	0.2	5	1269.81	463.78	96	12	368847.02	22 77 126 184
	0.2	10	time	time	207	47	631458.20	13 32 53 97 113 126 184
	0.5	2	653.74	419.86	19	6	197011.18	53 184
	0.5	5	841.89	408.06	93	16	411477.77	22 104 126 184
	0.5	10	time	2436.64	372	39	733190.17	14 32 61 113 126 184
	0.8	2	351.28	332.53	14	5	201286.60	53 184
	0.8	5	400.77	362.31	47	13	436165.81	57 126 184
	0.8	10	761.51	455.65	199	21	787170.71	22 104 126 140

Table 12: Comparison of optimality cuts using 36 instances of AP set with $|H| = 25$, 50, 75 and 100.

			Total time (sec)			*Iterations*		
$\|H\|$	τ	TC	NC	POC	SC	NC	POC	SC
25	0.2	2	0.11	0.96	0.11	5	4	4
	0.2	5	0.77	1.67	1.19	11	8	12
	0.2	10	1.97	1.92	1.16	17	9	12
	0.5	2	0.13	0.64	0.13	5	4	4
	0.5	5	1.03	1.72	0.56	15	11	11
	0.5	10	1.18	1.13	0.75	13	7	11
	0.8	2	0.08	0.44	0.09	5	4	4
	0.8	5	0.18	0.78	0.19	7	7	6
	0.8	10	0.55	0.89	0.29	10	7	7
50	0.2	2	1.11	11.81	1.17	8	6	7
	0.2	5	2.58	10.88	1.38	11	6	6
	0.2	10	15.13	22.57	8.21	20	13	15
	0.5	2	0.82	5.49	0.78	6	5	5
	0.5	5	3.12	7.83	1.50	13	8	9
	0.5	10	6.95	8.24	1.82	16	8	7
	0.8	2	0.82	3.17	0.79	6	5	5
	0.8	5	1.22	4.18	1.08	8	7	8
	0.8	10	6.73	7.26	2.19	17	11	10
75	0.2	2	5.20	63.09	6.08	9	7	9
	0.2	5	7.00	48.18	6.02	10	6	7
	0.2	10	16.23	45.53	11.48	16	6	10
	0.5	2	4.47	31.99	4.49	8	7	7
	0.5	5	6.99	23.10	4.60	12	5	6
	0.5	10	7.15	28.66	6.18	11	7	8
	0.8	2	4.50	17.42	4.43	9	8	8
	0.8	5	6.08	17.45	4.10	12	8	6
	0.8	10	7.19	19.30	5.78	13	9	10
100	0.2	2	14.64	209.46	15.55	7	6	6
	0.2	5	69.15	254.53	32.89	23	11	15
	0.2	10	170.00	207.16	32.77	23	9	12
	0.5	2	13.66	79.08	13.59	7	5	5
	0.5	5	38.85	136.65	23.83	20	11	13
	0.5	10	113.32	123.48	28.16	24	11	13
	0.8	2	12.91	34.59	12.97	6	5	5
	0.8	5	17.54	44.45	14.74	11	7	7
	0.8	10	38.38	51.61	17.02	20	9	10

Table 13: Comparison of optimality cuts using 36 instances of AP set with $|H| = 125$, 150, 175 and 200.

			Total time (sec)			*Iterations*		
$\|H\|$	τ	TC	NC	POC	SC	NC	POC	SC
125	0.2	2	47.20	734.78	48.67	10	8	8
	0.2	5	52.92	527.13	54.07	12	8	9
	0.2	10	170.82	670.35	107.67	25	12	17
	0.5	2	36.63	221.91	38.54	6	5	6
	0.5	5	53.26	280.32	43.97	16	8	9
	0.5	10	137.54	511.92	98.41	25	19	22
	0.8	2	35.70	97.19	36.65	6	6	6
	0.8	5	38.06	115.43	38.71	9	8	8
	0.8	10	45.66	142.71	43.81	13	11	11
150	0.2	2	138.66	1310.32	141.06	12	8	10
	0.2	5	143.40	1370.39	154.09	16	11	15
	0.2	10	455.93	1157.68	170.20	25	10	15
	0.5	2	91.57	582.45	95.22	6	6	6
	0.5	5	155.32	970.64	133.45	22	14	17
	0.5	10	202.96	674.03	127.31	20	12	13
	0.8	2	82.82	190.32	84.28	5	5	5
	0.8	5	107.40	355.26	104.40	15	12	14
	0.8	10	123.39	286.69	99.07	16	11	12
175	0.2	2	287.08	1675.29	311.86	7	7	7
	0.2	5	276.74	1586.16	287.86	15	8	12
	0.2	10	1031.62	2614.34	442.48	32	16	19
	0.5	2	210.40	864.59	215.48	5	5	5
	0.5	5	203.82	983.15	209.37	10	8	8
	0.5	10	391.21	1219.99	288.62	24	12	16
	0.8	2	172.08	347.30	174.02	4	4	4
	0.8	5	177.84	468.69	181.61	8	8	8
	0.8	10	180.13	414.40	181.52	9	7	7
200	0.2	2	656.69	2431.71	613.25	8	6	6
	0.2	5	463.78	2623.35	485.63	12	8	10
	0.2	10	time	6698.61	2541.63	47	21	31
	0.5	2	419.86	1495.22	432.01	6	6	6
	0.5	5	408.06	1681.55	418.60	16	9	13
	0.5	10	2436.64	2755.24	1059.11	39	18	27
	0.8	2	332.53	636.47	336.72	5	5	5
	0.8	5	362.31	1034.27	365.87	13	11	12
	0.8	10	455.65	1123.34	389.65	21	14	14

Table 14: Results of heuristic using 36 instances of AP set with $|H| = 25$, 50, 75 and 100.

				Total time (sec)				*Iterations*		
$\|H\|$	τ	*TC*	*% Dev*	*Heur*	$\|P_D^I\| = 0$	$\|P_D^I\| = 1$	$\|P_D^I\| = 5$	$\|P_D^I\| = 0$	$\|P_D^I\| = 1$	$\|P_D^I\| = 5$
25	0.2	2	0.00	0.01	0.11	0.11	0.07	4	3	3
	0.2	5	0.00	0.06	1.19	0.23	0.32	12	5	7
	0.2	10	0.00	0.2	1.16	0.61	0.53	12	9	9
	0.5	2	0.00	0.01	0.13	0.09	0.06	4	2	3
	0.5	5	0.00	0.04	0.56	0.5	0.68	11	10	13
	0.5	10	0.00	0.12	0.75	0.32	0.25	11	7	7
	0.8	2	0.00	0.01	0.09	0.05	0.05	4	2	3
	0.8	5	0.00	0.03	0.19	0.13	0.1	6	5	5
	0.8	10	0.00	0.06	0.29	0.14	0.07	7	4	4
50	0.2	2	0.00	0.14	1.17	0.85	1.2	7	4	7
	0.2	5	0.00	0.5	1.38	1.11	1.34	6	4	6
	0.2	10	0.00	1.23	8.21	7.15	6.1	15	10	11
	0.5	2	0.00	0.11	0.78	0.73	0.83	5	4	5
	0.5	5	0.00	0.28	1.5	1.31	1.22	9	7	7
	0.5	10	0.00	0.94	1.82	2.14	2.16	7	7	8
	0.8	2	0.00	0.1	0.79	0.71	0.81	5	4	5
	0.8	5	0.00	0.14	1.08	0.78	0.8	8	5	6
	0.8	10	0.00	0.55	2.19	1.3	1.49	10	7	8
75	0.2	2	0.00	0.38	6.08	5.52	5.6	9	8	8
	0.2	5	0.00	1.37	6.02	4.8	5.57	7	4	6
	0.2	10	0.00	4.79	11.48	11.73	11.71	10	9	10
	0.5	2	0.00	0.39	4.49	4.25	4.58	7	6	7
	0.5	5	0.00	1.25	4.6	4.07	4.52	6	4	6
	0.5	10	0.00	2.5	6.18	5.42	5.83	8	6	7
	0.8	2	0.00	0.39	4.43	3.9	4.41	8	5	8
	0.8	5	0.00	0.55	4.1	3.89	4.27	6	5	7
	0.8	10	0.00	1.68	5.78	5.12	5.68	10	8	10
100	0.2	2	0.00	1.39	15.55	15.68	15.65	6	6	6
	0.2	5	0.00	3.29	32.89	25.83	32.6	15	12	15
	0.2	10	0.00	10.26	32.77	30.73	33.25	12	10	11
	0.5	2	0.00	0.86	13.59	13.12	14.31	5	4	6
	0.5	5	0.00	2.16	23.83	22.78	28.68	13	13	16
	0.5	10	0.00	8.2	28.16	24.01	26.51	13	11	12
	0.8	2	0.00	0.8	12.97	12.87	14.45	5	4	7
	0.8	5	0.00	1.67	14.74	13.25	14.79	7	5	8
	0.8	10	0.00	3.46	17.02	18.34	19.23	10	11	11

Table 15: Results of heuristic using 36 instances of AP set with $|H| = 125$, 150, 175 and 200.

				Total time (sec)				*Iterations*		
$\|H\|$	τ	*TC*	*% Dev*	*Heur*	$\|P_D^I\| = 0$	$\|P_D^I\| = 1$	$\|P_D^I\| = 5$	$\|P_D^I\| = 0$	$\|P_D^I\| = 1$	$\|P_D^I\| = 5$
125	0.2	2	0.00	2.46	48.67	50.26	50.52	8	8	9
	0.2	5	0.00	7.17	54.07	57.12	57.52	9	10	10
	0.2	10	0.00	15.64	107.67	102.29	116.97	17	15	15
	0.5	2	0.00	1.85	38.54	37.17	41.02	6	5	8
	0.5	5	0.00	5.63	43.97	48.28	49.61	9	11	12
	0.5	10	0.00	10.09	98.41	64.08	69.27	22	15	16
	0.8	2	0.00	1.5	36.65	35.37	36.85	6	4	6
	0.8	5	0.13	3.83	38.71	39.66	43.82	8	9	12
	0.8	10	0.02	6.85	43.81	40.83	47.5	11	9	13
150	0.2	2	0.00	3.73	141.06	135.96	137.28	10	9	11
	0.2	5	0.00	13.89	154.09	118.11	133.07	15	8	12
	0.2	10	0.17	50.87	170.2	132.85	146.88	15	11	12
	0.5	2	0.00	3.12	95.22	95.4	99.13	6	6	8
	0.5	5	0.00	10.58	133.45	129.41	126.03	17	15	16
	0.5	10	0.00	26.73	127.31	111	113.88	13	12	13
	0.8	2	0.00	2.7	84.28	84.22	86.1	5	5	6
	0.8	5	0.00	3.7	104.4	95.36	98.12	14	12	12
	0.8	10	0.00	14.97	99.07	93.45	97.05	12	10	11
175	0.2	2	0.00	5.8	311.86	246.19	280.14	7	3	6
	0.2	5	0.00	17.55	287.86	264.21	267.17	12	11	11
	0.2	10	0.00	47.63	442.48	395.09	421.18	19	18	18
	0.5	2	0.00	4.84	215.48	196.53	214.17	5	3	5
	0.5	5	0.00	14.71	209.37	195.51	199.22	8	7	8
	0.5	10	0.00	27.3	288.62	261.83	281.35	16	15	18
	0.8	2	0.00	4.59	174.02	171.71	175.77	4	3	5
	0.8	5	0.00	7.65	181.61	180.38	179.16	8	8	7
	0.8	10	0.00	17.57	181.52	179.22	179.78	7	7	6
200	0.2	2	0.00	9.44	613.25	548.34	603.02	6	5	7
	0.2	5	0.00	29.83	485.63	494.38	535.71	10	9	10
	0.2	10	0.00	108.43	2541.63	2274.19	1825.69	31	28	28
	0.5	2	0.00	8.06	432.01	420.51	432.78	6	4	6
	0.5	5	0.00	17.57	418.6	431.21	444.05	13	13	14
	0.5	10	0.28	64.21	1059.11	880.76	948.57	27	26	26
	0.8	2	0.00	6.76	336.72	335.9	341.78	5	4	6
	0.8	5	0.00	18.47	365.87	356.56	347.28	12	11	9
	0.8	10	0.00	38.2	389.65	372.11	391.78	14	12	15

Table 16: Results of elimination tests using 36 instances of AP set with $|H| = 25$, 50, 75 and 100.

				Average time (sec)				*Average iterations*				*% Closed hubs*			
				$\|P_D^I\| = 0$		$\|P_D^I\| = 1$		$\|P_D^I\| = 0$		$\|P_D^I\| = 1$		$\|P_D^I\| = 0$		$\|P_D^I\| = 1$	
$\|H\|$	τ		TC	*E I*	*EI + EII*	*E I*	*EI + EII*	*EI*	*EI + EII*	*EI*	*EI + EII*	*E I*	*EI + EII*	*E I*	*EI + EII*
25	0	2	2	0.11	0.11	0.11	0.11	4	4	3	3	88.00	92.00	44.00	44.00
	0	2	5	0.84	0.85	0.30	0.30	11	11	5	5	20.00	28.00	32.00	48.00
	0	2	10	0.59	0.35	0.74	0.85	10	7	8	8	8.00	4.00	0.00	4.00
	0	5	2	0.08	0.09	0.07	0.08	4	4	2	2	84.00	88.00	0.00	0.00
	0	5	5	0.53	0.55	0.55	0.56	11	11	10	10	32.00	36.00	28.00	48.00
	0	5	10	0.46	0.48	0.41	0.44	8	8	7	7	4.00	12.00	12.00	16.00
	0	8	2	0.08	0.08	0.07	0.06	4	4	2	2	88.00	92.00	0.00	0.00
	0	8	5	0.20	0.20	0.17	0.17	6	6	5	5	40.00	56.00	28.00	28.00
	0	8	10	0.29	0.30	0.21	0.25	7	7	4	4	4.00	16.00	0.00	32.00
50	0	2	2	0.91	0.89	1.00	0.96	6	6	4	4	88.00	90.00	58.00	58.00
	0	2	5	1.43	1.46	1.69	1.76	7	7	4	5	58.00	66.00	38.00	74.00
	0	2	10	8.49	8.42	12.72	13.75	13	14	14	14	8.00	32.00	10.00	18.00
	0	5	2	0.69	0.71	0.83	0.85	5	5	4	4	90.00	92.00	76.00	92.00
	0	5	5	1.59	1.51	1.96	1.76	9	9	8	8	60.00	70.00	50.00	76.00
	0	5	10	1.73	1.77	3.16	3.50	7	7	7	7	32.00	42.00	30.00	38.00
	0	8	2	0.67	0.68	0.82	0.80	5	5	4	4	92.00	96.00	74.00	92.00
	0	8	5	0.96	0.95	0.83	0.87	7	7	4	4	80.00	76.00	80.00	84.00
	0	8	10	2.58	2.89	2.18	2.24	11	11	8	9	52.00	54.00	52.00	64.00
75	0	2	2	4.75	4.75	4.74	4.73	7	7	6	6	89.33	88.00	88.00	92.00
	0	2	5	5.34	5.28	5.89	5.80	7	7	4	4	77.33	81.33	80.00	90.67
	0	2	10	10.51	10.60	20.12	19.06	10	10	11	11	38.67	60.00	36.00	62.67
	0	5	2	4.06	4.06	4.65	4.78	7	7	8	8	92.00	93.33	89.33	89.33
	0	5	5	4.22	4.24	5.86	5.83	5	5	6	6	76.00	81.33	80.00	86.67
	0	5	10	5.57	6.78	7.74	7.72	7	8	5	5	52.00	64.00	53.33	66.67
	0	8	2	3.86	3.83	4.20	4.30	6	6	5	5	89.33	92.00	82.67	92.00
	0	8	5	4.11	4.09	4.43	4.43	6	6	5	5	82.67	81.33	81.33	92.00
	0	8	10	5.15	5.13	6.56	7.05	8	8	7	8	68.00	73.33	66.67	78.67
100	0	2	2	15.15	15.27	15.18	15.14	6	6	5	5	77.00	89.00	76.00	95.00
	0	2	5	24.84	24.12	24.22	28.09	13	12	11	13	70.00	67.00	71.00	77.00
	0	2	10	32.05	37.35	49.42	58.48	12	13	12	14	51.00	56.00	51.00	61.00
	0	5	2	13.28	13.30	13.59	13.81	5	5	4	4	90.00	92.00	74.00	95.00
	0	5	5	18.59	20.77	20.09	22.61	12	13	11	12	79.00	80.00	77.00	77.00
	0	5	10	27.36	28.66	28.29	34.77	13	12	9	11	49.00	56.00	55.00	60.00
	0	8	2	12.91	12.91	13.59	13.68	5	5	4	4	95.00	96.00	74.00	74.00
	0	8	5	15.28	15.90	14.87	15.32	8	8	5	6	68.00	70.00	83.00	92.00
	0	8	10	16.19	18.98	18.75	18.79	10	11	8	7	76.00	73.00	77.00	71.00

Table 17: Results of elimination tests using 36 instances of AP set with $|H| = 125$, 150, 175 and 200.

				Average time (sec)				*Average iterations*				*% Closed hubs*			
				$\|P_D^I\| = 0$		$\|P_D^I\| = 1$		$\|P_D^I\| = 0$		$\|P_D^I\| = 1$		$\|P_D^I\| = 0$		$\|P_D^I\| = 1$	
$\|H\|$	τ		TC	*E I*	*EI + EII*	*E I*	*EI + EII*	*EI*	*EI + EII*	*EI*	*EI + EII*	*E I*	*EI + EII*	*E I*	*EI + EII*
125	0	2	2	44.42	44.40	44.41	47.77	7	7	7	7	89.60	91.20	89.60	93.60
	0	2	5	50.40	50.28	50.34	56.34	9	9	9	8	74.40	81.60	81.60	85.60
	0	2	10	119.43	104.19	111.81	112.20	18	17	17	14	40.00	63.20	40.00	64.00
	0	5	2	36.47	36.48	36.48	38.23	5	5	5	5	89.60	93.60	89.60	96.80
	0	5	5	40.88	41.65	41.27	49.51	9	10	9	9	84.80	85.60	83.20	83.20
	0	5	10	88.30	90.90	89.60	73.52	21	20	18	15	44.80	60.00	47.20	59.20
	0	8	2	36.01	35.86	35.94	36.78	6	6	6	5	92.80	95.20	93.60	96.80
	0	8	5	37.30	37.69	37.50	43.55	8	8	8	9	86.40	86.40	86.40	89.60
	0	8	10	42.20	41.72	41.96	48.36	11	10	10	9	64.00	78.40	67.20	81.60
150	0	2	2	109.71	110.40	110.06	121.83	9	9	9	7	93.33	90.00	93.33	94.00
	0	2	5	117.21	116.37	116.79	123.90	12	11	11	9	82.00	75.33	84.00	87.33
	0	2	10	151.17	169.74	160.46	191.13	14	13	12	12	62.67	65.33	62.67	63.33
	0	5	2	89.51	89.95	89.73	96.16	6	6	6	6	92.67	91.33	92.67	93.33
	0	5	5	102.15	103.66	102.91	112.44	14	13	14	10	86.00	84.67	78.00	83.33
	0	5	10	114.80	121.15	117.98	135.71	13	13	13	10	68.00	72.67	69.33	80.67
	0	8	2	83.40	83.65	83.53	86.26	5	5	5	5	89.33	91.33	76.67	78.00
	0	8	5	90.04	90.98	90.51	91.66	11	11	11	9	86.67	83.33	86.00	89.33
	0	8	10	91.88	94.56	93.22	105.07	11	10	11	8	77.33	72.67	74.67	82.00
175	0	2	2	268.83	268.73	268.78	237.55	7	7	7	3	94.86	96.00	71.43	77.71
	0	2	5	254.92	255.49	255.21	256.08	10	10	10	10	77.71	86.86	81.71	85.71
	0	2	10	410.31	520.80	465.56	525.67	19	20	20	20	48.57	61.71	55.43	69.14
	0	5	2	202.80	202.69	202.75	196.25	5	5	5	3	97.14	98.29	94.29	98.29
	0	5	5	200.88	201.97	201.43	206.45	8	8	8	7	90.29	85.71	88.57	95.43
	0	5	10	258.70	268.96	263.83	263.70	15	15	15	14	53.14	65.14	56.57	65.71
	0	8	2	173.48	173.57	173.53	176.31	4	4	4	3	94.86	97.71	91.43	98.29
	0	8	5	177.39	179.25	178.32	184.16	7	8	7	7	82.86	90.86	85.14	96.57
	0	8	10	178.23	179.50	178.87	197.27	7	7	7	5	72.00	82.29	75.43	85.71
200	0	2	2	523.96	523.78	523.87	483.91	6	6	6	5	95.50	97.50	86.00	90.00
	0	2	5	447.96	448.25	448.11	485.47	10	9	10	7	89.50	80.50	81.50	87.50
	0	2	10	2395.21	1603.71	1999.46	1271.34	29	26	28	21	32.50	62.00	34.00	63.50
	0	5	2	390.77	392.37	391.57	393.36	6	6	6	4	95.50	97.50	92.50	96.50
	0	5	5	383.47	389.44	386.46	394.65	11	11	11	10	92.50	87.50	85.50	89.50
	0	5	10	874.78	914.56	894.67	750.23	26	23	26	21	48.00	57.50	46.50	60.00
	0	8	2	329.82	332.90	331.36	338.44	5	5	5	4	97.50	98.00	96.50	98.00
	0	8	5	342.15	347.76	344.96	361.97	11	11	11	9	92.00	83.50	82.50	92.50
	0	8	10	373.31	364.04	368.68	383.49	14	12	11	9	76.50	73.00	71.50	83.50

Table 18: Comparison of Benders algorithm using 54 instances of *Set I* with $|H| = 50$, 100, 150, 200, 250 and 300.

			% Gap			*Total time (sec)*			*Iterations*		*% Closed*	*Optimal solution*	
$\|H\|$	τ	TC	*Heur*	$B1$	$B2$	*Heur*	$B1$	$B2$	$B1$	$B2$	*hubs*	*Value*	*Set of hubs*
50	0.2	2	0.53	0.00	0.00	0.2	1.63	1.17	11	8	80.00	95453.05	17 20 43
	0.2	5	0.00	0.00	0.00	1.01	1.76	2.77	10	9	66.00	176755.87	17 19 28 43 50
	0.2	10	0.00	0.00	0.00	1.33	3.68	4.12	10	8	54.00	280490.13	3 17 19 28 31 34 47
	0.5	2	0.00	0.00	0.00	0.22	0.7	0.7	8	7	86.00	99796.22	43 50
	0.5	5	0.00	0.00	0.00	0.5	1.29	1.76	11	9	64.00	199897.39	17 43 47
	0.5	10	0.00	0.00	0.00	1.14	4.34	6.17	15	13	46.00	342846.18	17 19 28 43 47
	0.8	2	0.17	0.00	0.00	0.21	0.43	0.59	7	7	90.00	101431.59	43 50
	0.8	5	0.00	0.00	0.00	0.22	0.69	0.58	9	6	78.00	210616.58	17 43 47
	0.8	10	0.00	0.00	0.00	0.71	0.68	1.12	8	5	74.00	375563.34	17 19 28 43 47
100	0.2	2	0.00	0.00	0.00	1.61	6.04	7.26	5	7	94.00	958448.03	47 67
	0.2	5	0.00	0.00	0.00	5.27	13.62	13.56	11	8	81.00	1717247.23	19 20 67 87
	0.2	10	0.00	0.00	0.00	11.84	21.29	29.9	12	11	78.00	2733587.92	17 19 20 43 67 87
	0.5	2	0.00	0.00	0.00	1.29	4.54	4.32	7	5	77.00	1022032.03	50 67
	0.5	5	0.00	0.00	0.00	3.28	5.68	9.4	9	10	84.00	2032155.05	19 67 81
	0.5	10	0.00	0.00	0.00	9.47	30.72	34.91	19	15	56.00	3479465.06	20 43 67 87 96
	0.8	2	0.00	0.00	0.00	1.67	1.56	2.89	4	3	80.00	1022322.83	77
	0.8	5	0.00	0.00	0.00	1.91	1.85	3.34	5	4	90.00	2189049.87	50 67
	0.8	10	0.00	0.00	0.00	4.34	6.56	8.41	12	8	72.00	3945197.08	19 67 81
150	0.2	2	0.00	0.00	0.00	7.2	91.55	67.02	11	7	94.00	1981357.21	20 66 128
	0.2	5	0.00	0.00	0.00	23.64	80.77	76.45	10	9	86.00	3617316.49	20 43 66 137 150
	0.2	10	0.00	0.00	0.00	48.66	138.36	177.33	15	15	75.33	5818351.14	19 36 43 103 137 138 141 150
	0.5	2	0.00	0.00	0.00	6.13	60.18	38.57	14	11	92.67	2192049.63	66 103 150
	0.5	5	0.00	0.00	0.00	18.65	62.4	59.53	15	13	87.33	4370594.00	20 66 96 137
	0.5	10	0.00	0.00	0.00	54.64	203.75	145.83	23	18	67.33	7459760.40	19 34 36 103 137 138 150
	0.8	2	0.00	0.00	0.00	6.3	15.53	14.42	10	7	95.33	2260991.32	118 150
	0.8	5	0.00	0.00	0.00	10.41	19.37	23.33	11	9	86.00	4767506.82	20 66 150
	0.8	10	0.00	0.00	0.00	21.65	52.08	49.17	18	13	75.33	8582350.85	19 20 137 138 150
200	0.2	2	0.00	0.00	0.00	16.59	721.55	408.63	12	10	93.00	4581052.59	17 152 180
	0.2	5	0.00	0.00	0.00	78.95	529.97	466.82	22	15	84.50	8690186.86	17 43 141 152 180
	0.2	10	0.00	0.00	0.00	113.33	434.75	512.53	17	18	68.00	14029111.00	17 42 43 66 97 115 180
	0.5	2	0.00	0.00	0.00	16.54	476.25	173.92	18	12	92.50	4882414.46	88 115
	0.5	5	0.00	0.00	0.00	26.88	161.05	139.13	10	9	87.00	9987580.92	17 115 118 180
	0.5	10	0.00	0.00	0.00	76.55	416.02	266.05	23	15	72.00	17427726.67	17 28 63 115 118 180
	0.8	2	0.00	0.00	0.00	14.94	44.64	38.65	10	5	97.00	4983776.59	88 103
	0.8	5	0.00	0.00	0.00	17.1	37.44	42.48	9	6	96.50	10614381.69	17 115 180
	0.8	10	0.23	0.00	0.00	39.8	71.17	80.18	13	9	84.00	19410105.16	17 81 100 180
250	0.2	2	0.00	0.00	0.00	33.38	3314.85	1423.53	14	10	92.80	3874482.18	88 103
	0.2	5	0.00	0.00	0.00	142.65	1755.94	1232.83	22	16	80.40	7282336.18	28 78 88 141 213
	0.2	10	0.01	0.00	0.00	346.38	31189.44	17222.68	71	52	69.20	11873568.51	19 28 88 141 164 213 235 237
	0.5	2	0.00	0.00	0.00	35.09	1016.14	515.34	10	6	97.20	4062441.10	88 103
	0.5	5	0.00	0.00	0.00	108.6	1049.36	544.68	23	13	86.00	8425749.10	88 141 152 201
	0.5	10	0.00	0.00	0.00	157.84	5079.08	1660.4	34	25	61.60	14584699.51	24 28 88 141 213 235
	0.8	2	0.00	0.00	0.00	29.55	100.17	109.33	7	4	98.00	4146606.99	88 103
	0.8	5	0.00	0.00	0.00	51.16	152.9	152.23	13	10	92.40	8941909.55	88 152 201
	0.8	10	0.00	0.00	0.00	126.6	1134.23	520.73	27	20	68.40	16327319.27	24 28 66 88 213 235
300	0.2	2	0.00	0.00	0.00	102.57	20624.68	5149.04	24	23	91.67	3928387.32	77 103
	0.2	5	0.74	0.00	0.00	168.47	28914.76	17453.65	51	48	66.67	7425734.30	141 150 198 254
	0.2	10	0.00	0.67	0.59	496.46	time	time	58	54	47.00	12240606.61	28 43 128 141 243 256
	0.5	2	0.00	0.00	0.00	57.61	3656.84	1651.73	9	5	98.33	4018433.63	172
	0.5	5	0.00	0.00	0.00	115.23	9231.32	3596.2	38	31	71.67	8321461.85	77 152 180
	0.5	10	0.00	0.00	0.00	297.21	21107.77	11922.04	45	35	60.00	14339548.71	128 152 254 256 275
	0.8	2	0.00	0.00	0.00	42.59	322.7	248.09	6	4	94.33	4018433.63	172
	0.8	5	0.36	0.00	0.00	86.4	916.53	435.92	20	15	87.33	8687780.51	77 152 180
	0.8	10	0.00	0.00	0.00	154.91	38265.58	30204.22	70	58	52.67	15724133.58	77 128 152 254

Table 19: Comparison of Benders algorithm using 36 instances of *Set II* with $|H| = 50$, 100, 150 and 200.

			% Gap			*Total time (sec)*			*Iterations*		*% Closed*	*Optimal solution*	
$\|H\|$	τ	TC	*Heur*	*B1*	*B2*	*Heur*	*B1*	*B2*	*B1*	*B2*	*hubs*	*Value*	*Set of hubs*
50	0.2	2	0.00	0.00	0.00	0.23	1.86	0.99	10	7	80.00	1068384.96	19 47
	0.2	5	0.00	0.00	0.00	1.24	1.93	2.65	9	8	62.00	1982230.84	11 17 19 31 42 47
	0.2	10	0.00	0.00	0.00	1.82	16.89	19.16	13	15	28.00	3053193.54	3 11 17 19 22 27 28 31 47
	0.5	2	0.00	0.00	0.00	0.18	0.46	0.41	5	4	90.00	1117953.42	19 47
	0.5	5	0.00	0.00	0.00	0.82	0.99	1.5	8	7	74.00	2290402.89	11 17 19 43 47
	0.5	10	0.00	0.00	0.00	1.92	3.32	7.23	11	11	42.00	3896604.16	3 11 17 19 22 31 47
	0.8	2	0.00	0.00	0.00	0.20	0.53	0.37	6	4	92.00	1144745.11	19 47
	0.8	5	0.00	0.00	0.00	0.45	0.47	0.6	5	3	94.00	2432686.22	11 19 47
	0.8	10	0.00	0.00	0.00	0.86	0.57	1.35	6	6	62.00	4381754.77	11 17 19 43 47
100	0.2	2	0.00	0.00	0.00	2.45	26.34	11.2	15	10	87.00	6475483.69	17 19 28
	0.2	5	0.00	0.00	0.00	11.73	43.71	29.36	18	13	74.00	12000667.43	17 28 43 66 79
	0.2	10	0.00	0.00	0.00	19.02	24.82	33.63	11	12	80.00	18760244.63	28 43 66 67 75 79 81 88
	0.5	2	0.00	0.00	0.00	1.47	11.8	4.85	11	6	84.00	7014709.34	28 88
	0.5	5	0.00	0.00	0.00	6.26	43.47	18.65	20	14	76.00	14097698.70	28 34 79 88
	0.5	10	0.00	0.00	0.00	17.17	20.87	36.12	14	13	71.00	23962527.74	28 43 66 67 79 81 88
	0.8	2	0.00	0.00	0.00	1.41	4.42	2.86	7	4	66.00	7194920.97	28 88
	0.8	5	0.51	0.00	0.00	4.42	8.95	6.92	10	7	65.00	15198974.03	19 28 77
	0.8	10	0.00	0.00	0.00	9.86	60.68	50.93	24	20	68.00	27446996.05	28 34 79 88
150	0.2	2	0.00	0.00	0.00	9.28	87.73	63.23	8	7	92.67	13840337.13	28 66 128
	0.2	5	0.00	0.00	0.00	24.83	405.71	156.2	26	19	75.33	26725935.73	17 20 43 66 128
	0.2	10	0.38	0.00	0.00	75.74	8163.97	3588.99	51	43	51.33	44590550.97	19 28 43 47 66 67 137 150
	0.5	2	0.00	0.00	0.00	7.40	89.61	41.49	16	13	90.67	15485950.81	66 103 128
	0.5	5	0.00	0.00	0.00	18.47	104.44	54.9	17	13	82.00	31169206.40	19 20 66 150
	0.5	10	0.00	0.00	0.00	45.15	6445.11	1379.4	50	36	57.33	54965468.04	20 24 66 90 150
	0.8	2	0.00	0.00	0.00	4.93	33.35	13.47	11	8	94.67	15969490.54	96 118
	0.8	5	0.00	0.00	0.00	13.80	65.17	33.73	16	12	88.00	33599533.73	20 66 128
	0.8	10	0.46	0.00	0.00	23.63	172.48	81.02	20	13	72.00	61091408.94	19 20 66 150
200	0.2	2	0.80	0.00	0.00	17.60	870.61	397.49	15	16	93.00	28312270.32	66 103 128
	0.2	5	0.00	0.00	0.00	74.05	2208.66	1209.08	37	29	75.00	54925668.74	17 19 66 115
	0.2	10	0.00	0.26	0.30	258.48	time	time	70	68	41.00	90790523.74	17 19 20 43 66 113 137
	0.5	2	0.00	0.00	0.00	14.28	347.62	142.64	16	10	95.50	29883949.79	50 128
	0.5	5	0.00	0.00	0.00	60.42	3815.15	1865.61	57	46	66.50	62434123.12	17 66 115 180
	0.5	10	0.05	0.49	0.28	105.75	time	time	63	75	45.00	109722466.01	17 24 34 113 152
	0.8	2	0.00	0.00	0.00	12.79	26.94	42.7	12	8	93.00	30467892.95	50 128
	0.8	5	0.00	0.00	0.00	29.56	310.74	143.41	26	18	84.50	65867505.7	47 88 180
	0.8	10	0.00	0.00	0.00	74.93	5585.69	2228.77	44	35	52.50	119827042.48	17 19 50 66 103

Table 20: Comparison of Benders algorithm using 36 instances of *Set III* with $|H| = 50$, 100, 150 and 200.

			% Gap			*Total time (sec)*			*Iterations*		*% Closed*	*Optimal solution*	
$\|H\|$	τ	TC	*Heur*	*B1*	*B2*	*Heur*	*B1*	*B2*	*B1*	*B2*	*hubs*	*Value*	*Set of hubs*
50	0.2	2	0.00	0.00	0.00	0.43	1.21	1.12	8	6	84.00	213384.42	17 20 43
	0.2	5	0.00	0.00	0.00	1.50	1.87	3.79	9	10	62.00	396277.88	17 19 28 34 43 47
	0.2	10	0.00	0.00	0.00	1.70	49.51	27.64	17	16	22.00	637858.99	3 17 19 28 31 34 47
	0.5	2	0.00	0.00	0.00	0.21	1.24	0.56	9	5	62.00	229430.34	19 47
	0.5	5	0.00	0.00	0.00	0.73	3.16	11.88	16	24	60.00	465625.17	17 19 20 43 50
	0.5	10	0.00	0.00	0.00	1.21	2.62	10.08	13	16	40.00	791449.46	17 19 28 34 43 47
	0.8	2	0.00	0.00	0.00	0.34	0.58	0.63	7	5	90.00	233796.77	19 47
	0.8	5	0.52	0.00	0.00	0.45	1.59	1.34	11	10	72.00	497326.01	4 20 43
	0.8	10	0.00	0.00	0.00	1.10	1.23	1.68	8	6	62.00	886675.04	17 19 28 34 43 47
100	0.2	2	0.00	0.00	0.00	2.34	54.25	14.86	21	13	82.00	907379.18	20 43 88
	0.2	5	0.00	0.00	0.00	12.15	31.9	26.12	15	10	69.00	1661263.35	17 19 20 43 66
	0.2	10	0.00	0.00	0.00	19.93	45.62	35.25	13	9	77.00	2687610.56	19 28 43 47 60 66 67 88
	0.5	2	0.00	0.00	0.00	1.44	16.89	6.66	14	9	82.00	961919.22	47 88
	0.5	5	0.00	0.00	0.00	7.71	88.08	44.74	24	20	72.00	1964190.84	17 19 20 34 43
	0.5	10	0.29	0.00	0.00	16.76	91.82	92.98	21	20	55.00	3378760.50	17 19 28 34 43 47
	0.8	2	0.00	0.00	0.00	1.56	4.62	4.00	8	7	91.00	977835.63	47 88
	0.8	5	0.10	0.00	0.00	3.76	19.21	9.88	15	10	75.00	2090495.80	19 20 77
	0.8	10	0.00	0.00	0.00	10.98	28.26	21.33	14	10	58.00	3762525.47	19 20 34 43 88
150	0.2	2	0.03	0.00	0.00	11.79	266.72	89.07	23	14	85.33	1982556.18	103 118 150
	0.2	5	0.00	0.00	0.00	34.96	1530.73	207.36	41	20	73.33	3754614.74	17 19 20 43 141
	0.2	10	0.00	0.00	0.00	89.08	8701.56	3011.01	47	36	52.00	6189074.11	19 28 43 47 60 67 141 150
	0.5	2	0.00	0.00	0.00	5.90	164.59	46.59	22	16	87.33	2129448.17	88 103
	0.5	5	0.00	0.00	0.00	20.08	1405.08	164.85	39	21	76.67	4343129.83	17 19 115 141
	0.5	10	0.00	0.00	0.00	59.32	32454.78	5255.47	70	52	48.00	7597481.26	17 24 28 43 47 141
	0.8	2	0.00	0.00	0.00	5.47	36.61	16.18	12	9	92.00	2166739.51	88 103
	0.8	5	0.00	0.00	0.00	13.04	117.51	33.60	19	12	82.67	4620013.77	103 118 150
	0.8	10	0.03	0.00	0.00	31.09	2084.38	615.66	41	31	50.67	8396538.43	19 34 115 150
200	0.2	2	0.00	0.00	0.00	21.02	954.27	364.19	17	12	88.00	3491300.05	103 118 128
	0.2	5	0.59	0.00	0.00	79.96	24896.54	10182.09	73	59	56.00	6723977.42	17 19 43 141 152
	0.2	10	0.00	0.27	0.00	221.95	time	62960.59	62	73	44.00	11055338.21	19 28 43 75 103 141 164 172
	0.5	2	0.00	0.00	0.00	13.48	461.66	155.96	18	12	90.50	3726506.36	50 128
	0.5	5	0.00	0.00	0.00	49.19	3457.64	1550.68	46	35	64.00	7650937.45	17 19 103 118
	0.5	10	0.00	0.51	0.17	117.28	time	time	62	71	41.50	13406864.52	17 24 43 113 141 152
	0.8	2	0.00	0.00	0.00	17.68	59.53	48.98	11	8	91.50	3786269.19	88 103
	0.8	5	0.66	0.00	0.00	33.52	63.88	103.53	12	11	84.00	8093276.59	34 103 128
	0.8	10	0.00	0.00	0.00	75.94	10399.75	5593.50	46	39	47.50	14741708.60	19 88 103 151

Table 21: Computational results of 36 instances of *Set I* with $|H| = 350$, 400, 450 and 500.

			% Gap		*Total time (sec)*		*Iterations*	*% Closed*	*Optimal solution*	
$\|H\|$	τ	*TC*	*Heur*	*B2*	*Heur*	*B2*	*B2*	*hubs*	*Value*	*Set of hubs*
350	0.2	2	0.00	0.00	149.27	9867.9	14	93.14	4814837.17	103 128 275
	0.2	5	0.00	0.00	423.60	42209.42	49	70.00	9266821.81	17 275 323 333 348
	0.2	10	0.00	1.55	888.77	time	40	1.71	15414338.11	19 28 43 150 226 235 275
	0.5	2	0.00	0.00	145.65	2939.2	10	97.71	5182254.33	34 180
	0.5	5	0.37	0.00	293.12	40300	60	67.14	10679368.78	17 103 275 323
	0.5	10	0.00	0.88	497.13	time	41	46.57	18668497.14	50 110 156 323 333 348
	0.8	2	2.56	0.00	120.15	609.29	8	98.57	5182254.33	34 180
	0.8	5	0.00	0.00	156.42	755.06	12	89.71	11332464.07	34 103 128
	0.8	10	0.00	0.00	345.46	19791.7	39	57.43	20669490.79	77 103 156 323
400	0.2	2	0.00	0.00	214.92	24506.22	31	85.75	11647182.77	66 235 323
	0.2	5	0.00	0.00	655.28	80603.09	56	79.75	21918766.87	11 55 66 235 323 390
	0.2	10	0.00	0.29	2008.59	time	32	67.50	34991534.99	28 50 118 271 308 323 336 390
	0.5	2	0.00	0.00	210.69	6304.99	14	95.00	12246703.45	118 180
	0.5	5	0.00	0.00	382.69	12036.96	34	81.50	25158258.50	17 235 275 323
	0.5	10	0.00	0.00	1181.91	83162.59	39	69.00	43348414.05	50 60 275 333 348 390 397
	0.8	2	0.00	0.00	98.60	1130.79	9	96.25	12503643.67	118 180
	0.8	5	0.00	0.00	221.13	2426.71	25	79.75	26889674.22	180 320 390
	0.8	10	0.35	0.00	522.41	80026.2	45	55.50	48694157.48	66 86 103 320 397
450	0.2	2	0.00	0.00	358.53	50970.6	33	88.22	15231788.68	180 226 348
	0.2	5	0.70	0.74	825.23	time	35	74.22	28590433.45	17 225 302 320 379
	0.2	10	0.00	0.80	2686.16	time	26	51.78	46044048.31	17 103 128 132 225 299 407
	0.5	2	0.00	0.00	429.17	15741.29	32	90.67	16434881.90	180 348 390
	0.5	5	0.37	0.27	786.33	time	49	77.11	33189201.77	17 225 226 235 254
	0.5	10	0.00	0.26	1503.69	time	35	66.44	56769271.61	17 103 128 218 225 226 407
	0.8	2	0.00	0.00	231.03	1981.52	19	92.00	16762883.77	103 390
	0.8	5	0.46	0.00	442.49	10585.3	29	78.67	35580614.86	180 348 390
	0.8	10	0.00	0.00	874.98	85105.27	45	59.33	63935092.11	96 254 353 407 412
500	0.2	2	0.00	0.00	534.15	59474.39	10	87.80	18367424.23	118 235 271
	0.2	5	0.00	0.56	1229.15	time	40	69.40	35417722.40	11 43 47 270 291
	0.2	10	0.00	0.97	2739.08	time	26	17.80	575116180.16	43 270 271 291 300 348 386 469
	0.5	2	0.00	0.00	639.64	26133.4	15	94.40	19828698.61	103 225
	0.5	5	0.00	0.00	1062.82	34151.86	41	77.80	40697258.74	118 260 271 477
	0.5	10	0.20	0.54	2049.76	time	34	69.20	70787217.57	11 110 242 260 271 291
	0.8	2	0.00	0.00	271.06	4053.58	11	97.20	20200841.36	103 225
	0.8	5	0.34	0.00	530.23	4170.19	15	89.60	43227344.28	103 180 271
	0.8	10	0.28	0.00	1264.02	58665.77	40	66.20	78492890.70	118 226 260 271 477