Exactly solving a Two-level Hierarchical Location Problem with modular node capacities

Bernardetta Addis* Giuliana Carello* Alberto Ceselli [1] ◇

*Dipartimento di Elettronica e Informazione, Politecnico di Milano
Via Ponzio 34, 20133 Milano, Italia

◇Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano
via Bramante 65, 26013 Crema, Italia

Abstract

In many telecommunication networks a given set of client nodes must be served by different sets of facilities, providing different services and having different capabilities, which must be located and dimensioned in the design phase. Network topology must be designed as well, by assigning clients to facilities and facilities to higher level entities, when necessary.

We tackle a particular location problem in which two sets of facilities have to be located, and in which different devices can be installed in each site, providing different capacities at different costs. We optimize simultaneously location and dimensioning of these facilities. We introduce a compact formulation for that problem, we exploit discretization and Dantzig-Wolfe reformulation techniques to obtain better models, and we design an exact optimization algorithm. We test our approach on a set of instances derived from the facility location literature.

Keywords: *location, discretization, column generation, branch-and-price, telecommunications*

1 Introduction

In many telecommunication networks different sets of network facilities, equipped with different devices and carrying out different tasks, are needed. In designing such networks the set of client nodes is usually given, while facilities of different kinds must be located and dimensioned. For instance in IP networks *access* nodes are origin and destination of traffic demands but cannot be connected directly one another. Instead, the network structure is hierarchical: access nodes must be connected to *edge* nodes which, in turn, must be connected to the *core* backbone nodes. Traffic is then routed through the different network levels.

A similar structure can be found in fiber-to-the-home networks, where nodes representing clients must be connected to *cabinet* nodes which collect traffic and send it to *central offices*.

The optimal design of networks with the above structure can be seen as a Facility Location Problem in which two different sets of facilities are considered. Facilities in the same sets are similar and are said to belong to the same level. As in the Single Source Facility Location Problem, each client must be assigned to exactly one facility of a mid level set. Besides, each mid level facility must be assigned to exactly one of a high level set. The network topology turns out to be a star-star one. Each client has a demand to be served, and facilities of both levels can serve limited amounts of demand. Moreover, each

[1]Corresponding author: email alberto.ceselli@unimi.it

mid level facility must be *dimensioned*, by installing different kind of devices, capable of serving different amounts of demand at different costs.

A recent review on hierarchical facility location problems, covering papers since the mid-80s, can be found in [11]. Hierarchical facility location problems are classified according to features such as flow pattern and service availability; applications, models and approaches are described.

In a classical generalization of the facility location problem, the so-called Multi-level Facility Location Problem, a set of clients is given together with k sets of facilities, where each set represent a different facility level. Each client must be assigned to a path of k facilities, and its demand must be routed through a facility of each level following a hierarchical order. For that problem, heuristic and exact approaches [12] as well as approximation properties [1] have been investigated. However, unlike the case we consider in this paper, there are no Single Source restrictions: each facility can fractionally be assigned to many higher level facilities.

Another similar generalization of the facility location problem is the Two-level Simple Plant Location Problem proposed in [6]: each client must be assigned to one and only one facility of the mid level which, in turns, must be assigned to one and only one facility of the high level ; facilities of both levels are uncapacitated. The authors proposed both lower and upper bounds for the problems, obtaining good results for problems up to 100 clients and candidate facility sites for both levels.

The problem we considered shares some features also with the Two-echelon Single Source Capacitated Facility Location Problem described in [14], where each client must be assigned to exactly one mid level facility which, in turns, must be assigned exactly to one depot; indeed, depots act as high level facilities, which however are not capacitated. In the paper a branch and bound method is proposed, based on Lagrangian relaxation, solving to optimality instances involving up to 100 clients, 10 mid level candidate facilities and 5 high level candidate facilities.

The problem proposed in [9] is the most similar to that considered in this paper: the location of two different types of facilities, concentrators and routers, in a telecommunication network is considered; both concentrators and routers are capacitated. Each terminal in the network, which represents a client, has to be assigned to exactly one concentrator, which must in turn be assigned to one router. The location of both concentrators and routers has to be chosen. Besides classical fixed installation costs and allocation costs, operational costs related to concentrators and routers are considered, which linearly depend on the amount of capacity used. The problem is heuristically tackled by computing both lower and upper bounds: lower bounds are based on Lagrangian relaxation, and upper bounds are provided through a tabu search heuristic, in which both adding and dropping moves are used. The solution is then completed through a heuristic assignment. The proposed approach is tested on instances with up to 500 terminals, 200 concentrators and 50 routers.

Modeling of telecommunications applications as Two-Level Facility Location Problems are also proposed in [5] and [15]: in [5] the IP network design problem is heuristically tackled; in [15] a hierarchical continuous location problem, where a set of concentrators and one central equipment must be located, is tackled with a column generation approach.

In all the previous papers only location decisions are considered in detail, while dimensioning is only approximated, tackled in a post-processing step or disregarded at all; however, in order to deal with practical applications location and dimensioning have to be optimized simultaneously.

Thus, as first novel contribution, in this paper we consider the problem of simultaneously locating and dimensioning capacitated facilities in a star-star network. We denote the problem as the *Two-level Hierarchical Capacitated Facility Location Problem* (TLHCFLP). TLHCFLP is *NP*-hard, as it generalizes the classical Facility Location Problem.

We provide a compact formulation for the TLHCFLP, and we derive better models by exploiting discretization and Dantzig-Wolfe reformulation techniques. As a second novel contribution, we design an exact optimization algorithm which exploits a hybrid formulation and dynamic column generation within a branch-and-bound framework; it includes ad-hoc stabilization techniques and addition of dual cuts, dynamic generation of general purpose cuts. We test our approach on a set of instances derived from the literature on the Single-Source Capacitated Facility Location Problem.

The paper is organized as follows: in Section 2 the problem is described in detail and our formulations are presented. Our exact optimization algorithm is described in Section 3. The results of an experimental analysis comparing various techniques are reported in Section 4. Final remarks, in Section 5, end the paper.

2 Problem description and formulations

In the TLHCFLP a set of client nodes $\mathcal{I}$ is given. Each client node $i \in \mathcal{I}$ has a demand a_i, it must be connected to a mid level facility which in turn must be connected to a high level facility. Both kinds of facility must be located: the sets of sites which are candidate to respectively host mid level and high level facilities are respectively denoted with $\mathcal{J}$ and $\mathcal{K}$. Placing a mid level facility in a site $j \in \mathcal{J}$ and a high level facility in a site $k \in \mathcal{K}$ implies installation costs c_j and g_k, respectively. Assigning client i to a mid level facility located in $j \in \mathcal{J}$ and a mid level facility located in j to a high level facility located in $k \in \mathcal{K}$ implies connection costs d_{ij} and l_{jk}, respectively.

Each mid level facility must be dimensioned by equipping it with a device chosen in a set $\mathcal{T} = \{1 \dots T\}$. For each device $t \in \mathcal{T}$ a capacity b_t and a setup cost f_t are given; the b_t coefficient represents also the demand to be served by a high level facility to a mid level one equipped with device t. To model features of practical applications we suppose that each device t provides half capacity with respect to device $t+1$; moreover, due to economy of scale, device costs are assumed to be sub-linear with respect to the provided capacity.

Dimensioning of high level facilities is not required: all high level facilities provide the same capacity B.

In the TLHCFL four decisions have to be taken: to open or not a high level facility in each $k \in \mathcal{K}$ (binary variables z_k), to open or not a mid level facility equipped with device $t \in \mathcal{T}$ in each candidate site $j \in \mathcal{J}$ (binary variables y_{jt}), to assign or not a mid level facility opened in $j \in \mathcal{J}$ and equipped with device $t \in \mathcal{T}$ to a high level facility in $k \in \mathcal{K}$ (binary variables w_{jtk}), to assign or not a client $i \in \mathcal{I}$ to a mid level facility located in $j \in \mathcal{J}$ (binary variables x_{ij}).

An ILP model for TLHCFL is the following:

$$\text{IP)} \quad \min \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} d_{ij}x_{ij} + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} (c_j + f_t)y_{jt} + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} l_{jk}w_{jtk} + \sum_{k\in\mathcal{K}} g_k z_k \tag{1}$$

$$\sum_{j\in\mathcal{J}} x_{ij} = 1, \forall\ i \in \mathcal{I} \tag{2}$$

$$\sum_{i\in\mathcal{I}} a_i x_{ij} \leq \sum_{t\in\mathcal{T}} b_t y_{jt}, \forall\ j \in \mathcal{J} \tag{3}$$

$$\sum_{t\in\mathcal{T}} y_{jt} \leq 1, \quad \forall j \in \mathcal{J} \tag{4}$$

$$\sum_{k\in\mathcal{K}} w_{jtk} \geq y_{jt}, \forall\ j \in \mathcal{J}, \forall\ t \in \mathcal{T} \tag{5}$$

$$\sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} b_t w_{jtk} \leq B z_k, \forall\ k \in \mathcal{K} \tag{6}$$

$$x_{ij}, w_{jtk}, y_{jt}, z_k \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{7}$$

The objective function (1) aims at minimizing the total installation, setup and assignment cost. Constraints (2) force each client to be assigned to exactly one mid level facility, while constraints (5) force each open mid level facility to be assigned to a high level one. Inequalities (3) guarantee that each mid level facility is equipped with a device providing enough capacity to serve the demand of all the

assigned clients, while inequalities (4) guarantee that each mid level facility is equipped with at most one device. Finally, inequalities (6) guarantee that each high level facility provides enough capacity to serve the demand of all the assigned mid level facilities. The model is completed with integrality conditions (7). Since it is easy to check that it is never convenient to assign each client to more than one mid level facility, we relax constraints (2) in greater than or equal form.

Model (IP) has a polynomial number of variables and constraints, and is therefore suitable to be optimized by general purpose ILP solvers. As discussed in detail in Section 4, this approach does not allow to solve to proven optimality a large number of instances, whose features are similar to that of practical applications.

Therefore, we propose to reformulate the problem, and solve it by ad-hoc integer programming techniques. In particular,we derive a formulation having two parts: a discretized part which models the high level location problem, and an extended part which models the mid level location problem. Both reformulation steps are described in the following subsections.

2.1 Reformulation by discretization of the high level location problem

In the following we introduce the discretization. Since $b_{t+1} = 2 \cdot b_t$, for each device $t \in \mathcal{T}$ we scale capacity and demand coefficient of each device by b_1: for each $t \in \mathcal{T}$, let $\tilde{b}_t = b_t/b_1$ and $\tilde{B} = B/b_1$. That is, if we say b_1 to represent a *capacity unit*, each $\tilde{b}_t$ and $\tilde{B}$ coefficients are expressed in terms of capacity units. Therefore, we rewrite constraints (6) as follows:

$$\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \tilde{b}_t w_{jtk} \leq \tilde{B} z_k, \forall\ k \in \mathcal{K}.$$

Since each $\tilde{b}_t$ coefficient is integer, these inequalities have always integer left hand sides; therefore, we can strengthen their right hand side by rounding it down to the nearest integer, and replace them by

$$\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \tilde{b}_t w_{jtk} \leq \lfloor \tilde{B} \rfloor z_k, \forall\ k \in \mathcal{K}. \tag{8}$$

Then, let $\mathcal{Q} = \{1 \dots \lfloor \tilde{B} \rfloor\}$ be the set of values which can represent the actual capacity units provided by a single high level facility. Following a discretization technique introduced in [8], we substitute each variable z_k with a set of binary variables z_k^q with $q \in \mathcal{Q}$; each variable z_k^q takes value 1 if a high level facility is built in k and serves exactly q capacity units, 0 otherwise. Therefore $z_k = \sum_{q \in \mathcal{Q}} z_k^q$, provided that

$$\sum_{q \in \mathcal{Q}} z_k^q \leq 1. \tag{9}$$

Finally, if conditions (9) hold, then $\sum_{q \in \mathcal{Q}} q z_k^q \leq \lfloor \tilde{B} \rfloor$; therefore constraints (8) can be replaced as follows:

$$\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \tilde{b}_t w_{jtk} = \sum_{q \in \mathcal{Q}} q z_k^q, \forall\ k \in \mathcal{K}. \tag{10}$$

The whole problem can be reformulated as follows:

$$\text{HD)} \quad \min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} d_{ij} x_{ij} + \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (c_j + f_t) y_{jt} + \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} l_{jk} w_{jtk} + \sum_{k \in \mathcal{K}} g_k \sum_{q \in \mathcal{Q}} z_k^q \tag{11}$$

$$s.t. (2), (3), (4), (5), (9), (10)$$

$$x_{ij}, w_{jtk}, y_{jt}, z_k^q \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \forall q \in \mathcal{Q} \tag{12}$$

As for the compact model, the objective function (11) aims at minimizing the total installation, setup and connection cost.

From a theoretical point of view, the LP relaxation bound given by (HD) is not tighter than the one given by (IP). However, model (HD) can be enriched by the following simple but powerful set of inequalities (see [8] for a similar kind of inequalities): let N be a lower bound on the number of capacity units which have to be provided by the set of opened mid level facilities; N can be computed by simply dividing the demand of all clients by the capacity of the smallest device and rounding up to the nearest integer

$$N = \left\lceil \frac{\sum_{i \in \mathcal{I}} a_i}{b_1} \right\rceil$$

Then

$$\sum_{k \in \mathcal{K}} \sum_{q \in \mathcal{Q}} q z_k^q \geq N.$$

Now, for all $p \in \mathcal{Q}$, we can divide both sides of the inequality by p, round up to the nearest integer the left-hand-side coefficients, still obtaining a valid inequality:

$$\sum_{k \in \mathcal{K}} \sum_{q \in \mathcal{Q}} \left\lceil \frac{q}{p} \right\rceil z_k^q \geq \frac{N}{p}, \quad \forall p \in \mathcal{Q}.$$

Since in this way the left-hand-side never takes fractional values, we can round also each right-hand-side term up to the nearest integer, obtaining the following set of inequalities:

$$\sum_{k \in \mathcal{K}} \sum_{q \in \mathcal{Q}} \left\lceil \frac{q}{p} \right\rceil z_k^q \geq \left\lceil \frac{N}{p} \right\rceil. \tag{13}$$

Besides including these $|\mathcal{Q}|$ inequalities, we strengthen the final model with the following consistency constraints: for all $j \in \mathcal{J}$ and $k \in \mathcal{K}$

$$w_{jkt} \leq \sum_{q \in \mathcal{Q}} z_k^q \tag{14}$$

that is, when no high level facility is built in k, no mid level facility can be assigned to k.

A similar technique can be applied to the mid level location problem. Such approach is described in the Appendix. We performed preliminary experiments by using the state-of-the-art general purpose solver CPLEX 11 [7] to optimize (a) the compact model (IP), (b) the discretized model (HD) including (13) and (14), (c) a model obtained after reformulation by discretization of both mid level and high level location problems. We observed that (b) substantially improves (a); instead (c) consistently yields out-of-memory problems due to the large number of binary variables introduced, and is therefore unsuitable for general purpose solvers. Solution approach (a) and (b) are also compared in Section 4.

2.2 Dantzig-Wolfe reformulation of the mid level location problem

We perform a second reformulation step to our model by exploiting the following technique.

First we consider, for each site $j \in \mathcal{J}$ and for each device $t \in \mathcal{T}$, each subset of clients which can be assigned to a facility in site j equipped with device t without violating constraints (3); we call these subsets *clusters* of clients, and we indicate as $\mathcal{S}_{jt}$ the set of all clusters which refer to site j and device t. Moreover we describe with coefficients u_{is} the incidence vector of each set s, that is u_{is} is one if i is included in cluster s and zero otherwise. The cost of a cluster $s \in \mathcal{S}_{jt}$ is given by $C_s = c_j + f_t + \sum_{i \in \mathcal{I}} u_{is} d_{ij}$. For all $j \in \mathcal{J}$, for all $t \in \mathcal{T}$ and for all $s \in \mathcal{S}_{jt}$ we introduce a binary variable v_s, which takes value one if a mid level facility equipped with device t is built in j, and cluster s is assigned to that facility, zero otherwise. The w and z variables keep the same role as above; instead, the v variables replace the x and y variables, as one can express each $y_{jt} = \sum_{s \in S_{jt}} v_s$ and each $x_{ij} = \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \sum_{s \in S_{jt}} u_{is} v_s$.

$$\text{IMP)} \quad \min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}_{jt}} C_s v_s + \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} l_{jk} w_{jtk} + \sum_{k \in \mathcal{K}} g_k \sum_{q \in \mathcal{Q}} z_k^q \tag{15}$$

$$\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}_{jt}} u_{is}v_s \geq 1, \forall\ i \in \mathcal{I} \tag{16}$$

$$\sum_{k\in\mathcal{K}} w_{jtk} - \sum_{s\in S_{jt}} v_s \geq 0, \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \tag{17}$$

$$\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} w_{jtk} \leq 1, \forall j \in \mathcal{J} \tag{18}$$

$$\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} \tilde{b}_t w_{jtk} = \sum_{q\in\mathcal{Q}} q z_k^q, \forall k \in \mathcal{K} \tag{19}$$

$$\sum_{q\in\mathcal{Q}} z_k^q \leq 1, \forall k \in \mathcal{K} \tag{20}$$

$$\sum_{k\in\mathcal{K}}\sum_{q\in\mathcal{Q}} \left\lceil \frac{q}{p} \right\rceil z_k^q \geq \left\lceil \frac{N}{p} \right\rceil, \quad \forall p \in \mathcal{Q} \tag{21}$$

$$w_{jkt} \leq \sum_{q\in\mathcal{Q}} z_k^q, \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{22}$$

$$v_s, w_{jtk}, z_k^q \in \{0,1\} \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \forall s \in \mathcal{S}_{jt}, \forall q \in \mathcal{Q}. \tag{23}$$

The objective function (15) represents the total network cost: client assignment costs and mid level facility opening and dimensioning are now described through cluster costs. Constraints (16) guarantee that for each client at least one cluster containing it is assigned to the corresponding mid level facility. Equations (17) force each mid level facility, to which a cluster is assigned, to be connected to a high level one, taking into account the involved device. Since $y_{jt} = \sum_{s\in S_{jt}} v_s$, and $\sum_{s\in S_{jt}} v_s \leq \sum_{k\in\mathcal{K}} w_{jtk}$ due to (17), constraints (18) guarantee that a mid level facility is assigned to at most one high level facility. They also imply $\sum_{t\in\mathcal{T}} w_{jtk} \leq 1$ for each $k \in \mathcal{K}$, that is each mid level facility is equipped with at most one device. Finally, together with (17) they avoid that two or more clusters are assigned to the same mid level facility, thus guaranteeing that the capacity is not violated. Inequalities (20) and equations (19) represent high level facility capacity constraints. As discussed in the previous paragraphs, inequalities (21) and (22) help to improve the LP bound of this formulation.

It is not hard to show that the LP relaxation of model (IMP) corresponds to the Dantzig-Wolfe reformulation of the LP relaxation of model (HD), when the region described by constraints (3) and (4) is replaced by the convex hull of its extreme integer points. Therefore, the dual bound given by the LP relaxation of (IMP) can be much stronger than that given by the LP relaxation of (HD). The actual quality of these bounds is compared in Section 4.

As before, a similar technique can be applied to the high level location problem instead of performing the discretization step. However, preliminary experimental results showed that optimizing such two-level extended formulation is computationally very hard. Further discussion is carried out in Section 4.

3 A branch and price algorithm

We indicate as Master Problem (MP) the continuous relaxation of model (IMP). Since each set $\mathcal{S}_{jt}$ has an exponential number of elements in problem dimension, and therefore the model contains an exponential number of variables, column generation techniques are needed to solve MP.

We compute the optimal MP value and we embed this dual bound in a branch-and-price framework, obtaining an exact optimization algorithm for TLHCFLP. Our method works as follows.

Initialization. We consider a restricted master problem (RMP) including all the constraints, but only z and w variables and columns encoding clusters containing just one client. We enrich this initial

RMP with the solutions obtained by three runs of a simple randomized greedy heuristic, which works as follows. First, mid level facilities are opened in a randomly selected set of sites; each facility is initially equipped with the largest device: the choice on the most suitable device to install on each facility is delayed to subsequent steps. Then, each client is assigned to the nearest facility having enough residual capacity. Finally, the cheapest device providing enough capacity to serve the sum of the demands of the assigned clients is installed on each facility. The above procedure may not produce a complete integer feasible solution, as the assignment of mid level facility to high level ones is not decided, and thus it might be impossible to satisfy high level capacity constraints. Indeed, it proved to be enough to generate good initial clusters.

Column generation. We iteratively solve the RMP and exploit the optimal dual solution to search for new columns having negative reduced cost; the problem of finding such columns is called the pricing problem. Let λ_i and μ_{jt} be the dual variables associated respectively to each constraint of the set (16) and to each constraint of the set (17). For each $j \in \mathcal{J}$ and for each $t \in \mathcal{T}$, the reduced cost of a column encoding a cluster $s \in S_{jt}$ is

$$C_s - \sum_{i \in \mathcal{I}} \lambda_i u_{is} + \mu_{jt}$$

that is

$$c_j + f_t + \sum_{i \in \mathcal{I}} u_{is} d_{ij} - \sum_{i \in \mathcal{I}} \lambda_i u_{is} + \mu_{jt}.$$

We consider the cluster s associated to a value τ_{jt} computed as follows:

$$\tau_{jt} = \min_{s \in S_{jt}} \{c_j + f_t - \sum_{i \in \mathcal{I}} (\lambda_i - d_{ij})\, u_{is} + \mu_{jt}, \text{s.t.} \sum_{i \in \mathcal{I}} a_i u_{is} \leq b_t\}.$$

that corresponds, for each $j \in \mathcal{J}$ and for each $t \in \mathcal{T}$ to the column with minimum reduced cost. In details, the problem of finding such cluster can be stated as follows:

$$c_j + f_t + \mu_{jt} - \max \sum_{i \in \mathcal{I}} (\lambda_i - d_{ij})\, u_i$$

$$\text{s.t.} \sum_{i \in \mathcal{I}} a_i u_i \leq b_t$$

$$u_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}$$

Thus, we obtain each τ_{jt} value by solving a 0-1 knapsack problem; besides classical pseudo-polynomial dynamic programming algorithms, several methods have been proposed in the literature to exactly optimize 0-1 knapsack problems [20], practically showing linear computing time.

The problem of finding the most negative reduced cost column consists in finding the column associated to the value $\sigma = \min_{j \in \mathcal{J}, t \in \mathcal{T}} \{\tau_{jt}\}$. If $\sigma \geq 0$, the current RMP is optimal for MP, and the corresponding objective function value gives a valid dual bound for the whole problem. Otherwise, all the columns having $\tau_{jt} \leq 0$, which are encoded by the coefficients u_{is}, are inserted into the RMP; the new RMP is solved, a new vector of dual variables is obtained and the column generation process is iterated.

Multiple pricing and stabilization. Column generation methods are known to suffer stability problems [21], yielding poor convergence performance. In order to overcome such problems we elaborated on the stabilization method described in [16] in order to devise an ad-hoc stabilization procedure for our problem, obtaining good results with a modest additional computational effort. Our stabilization procedure works as follows.

Let us consider the dual problem of MP (DMP). Each feasible DMP solution would give a valid dual bound to TLHCFLP. However, at each column generation iteration a *restricted* problem is solved: some

MP columns are missing from RMP; since MP columns represent DMP constraints, by solving RMP we obtain a set of dual variables which can represent either a feasible, and therefore optimal DMP solution, or an infeasible super-optimal one.

We indicate as $\pi = (\lambda_i, \mu_{jt})$ the portion of the dual vector which is relevant during pricing;

we keep as *stability center* the feasible dual solution $\bar{\pi} = (\bar{\lambda}_i,\ \bar{\mu}_{jt})$ yielding the best bound so far; initially, $\bar{\pi}$ can be set to a vector with all zero components, which actually compose a feasible dual solution to our problem. Instead of pricing with vector π we choose a value $\alpha \in [0, \ldots, 1]$, and we consider the vector

$$\tilde{\pi} = (\tilde{\lambda}_i, \tilde{\mu}_{jt}) = \alpha(\bar{\lambda}_i, \bar{\mu}_{jt}) + (1 - \alpha)(\lambda_i, \mu_{jt}).$$

We insert in the current RMP all the columns with negative reduced cost found in this way, we decrease α by 0.01, we compute a new vector $(\tilde{\lambda}_i,\ \tilde{\mu}_{jt})$ and we iterate. It might happen that no column with negative reduced cost can be found for a particular value of α; in this case, since each column represents a constraint in the dual problem, the vector $\tilde{\pi}$ does not violate any dual constraint, and therefore represents a feasible dual solution. Moreover, since $\tilde{\pi}$ is found as a linear convex combination of a feasible dual solution and a super-optimal one, it improves the best known dual bound; therefore we set $\bar{\pi} = \tilde{\pi}$.

We initially fix $\alpha = 0.99$ and we repeat this process until no more columns with negative reduced cost can be found, and $\alpha = 0.00$.

In this way we obtain two important effects: (a) the stability center helps to always price with RMP dual solutions near known good ones, (b) many different columns for each candidate facility are added to the RMP at each pricing iterations; this kind of multiple-pricing techniques showed to be useful in column generation frameworks [21].

Reducing pricing effort. Even if the solution of a single 0-1 knapsack problem can be computed efficiently, our pricing problem requires in principle to solve $|\mathcal{J}| \cdot |\mathcal{T}|$ subproblems at each column generation iteration, and for each choice of the α parameter.

In order to reduce the computational effort, we proceed as follows.

First, we introduce the following set of dual cuts [17]: for all $j \in \mathcal{J}$ and for all $t \in \mathcal{T} \setminus \{T\}$

$$\mu_{jt+1} + f_{t+1} \geq \mu_{jt} + f_t; \tag{24}$$

there always exists an optimal dual solution satisfying these constraints. In fact, suppose by contradiction to have an optimal primal solution in which a column encoding a cluster for facility j equipped with device $t < T$ is basic, and has therefore reduced cost 0.0. If $\mu_{jt+1} + f_{t+1}$ was less than $\mu_{jt} + f_t$ in the corresponding optimal dual solution, the column encoding the same cluster for the same facility, equipped with a larger device $t + 1$ would be feasible, since the facility capacity is not decreased, and with negative reduced cost; therefore, the solution could not be optimal. These columns help to speedup the column generation process.

Second, we experimented on the following two techniques.

(a) We start pricing by device $t = T$. Then, whenever a column with negative reduced is found, we downgrade the device of the cluster to the smallest device s which can serve such demand, obtaining a column with non-worse reduced cost. We proceed iteratively, pricing device $s - 1$, possibly downgrading device, until $s = 1$. It is possible to check that when conditions (24) are enforced, no column with minimum reduced cost is lost in this way.

(b) During each column generation iteration, whenever a column with negative reduced cost column is found for a pair (j, t) and a particular choice of α, the candidate location site j is inserted in a *tabu list*. While decreasing α, we do not solve again the pricing problem for any site in the tabu list. In fact since, by decreasing α, the magnitude of dual values in $\tilde{\pi}$ tends to increase; therefore, the chance of finding again the same columns gets higher and higher. The tabu list is cleared at the end of each column generation iteration, that is when $\alpha = 0.0$.

A detailed experimental comparison of pricing techniques is carried out in Section 4.

Cut generation. When column generation is over, we search for valid inequalities which are violated by the current RMP fractional solution. In fact, while cuts on the v column variables are difficult to handle in a branch-and-price-and-cut scheme, cuts involving w and z variables can improve the quality of the bound without affecting the structure of the pricing problem. We search for such violated inequalities in a pool of general purpose cuts, as sketched out in Section 4. Whenever new cuts are found, the whole column generation process is repeated to ensure the dual bound to be valid.

Such price-and-cut loop is repeated until neither new cuts nor new columns are generated.

Branching. Indeed, the MP optimal solution might not be integral, even at the end of the price-and-cut loop. In this case we first compute the fractional solution of the compact formulation (IP) corresponding to the current fractional solution of MP by setting

$$x_{ij} = \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}_{jt}} u_{is} v_s \quad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J}$$

$$y_{jt} = \sum_{s \in \mathcal{S}_{jt}} u_{is} v_s \quad \forall j \in \mathcal{J},\ \forall t \in \mathcal{T}.$$

and keeping the z and w variables at their values. Then we complete the computation by exploring a branching tree, which is composed by six levels.

1. At the first level we branch on the (20) constraints: we select the index $\bar{k}$ for which $\sum_{q \in \mathcal{Q}} z^q_k$ is nearest to 0.5. In fact, in an integer solution $\sum_{q \in \mathcal{Q}} z^q_k$ is 1 if a high level facility is placed in k or 0 otherwise. We impose in one branch that $\sum_{k \in \mathcal{Q}} z^q_{\bar{k}} = 1$ and in the other branch that $\sum_{k \in \mathcal{Q}} z^q_{\bar{k}} = 0$. Both conditions can simply be enforced by fixing to zero either z variables or slack variables in constraints (20).

2. When all the values $\sum_{q \in \mathcal{Q}} z^q_k$ are 0 or 1, we select the index $\bar{k}$ for which the largest number of $z^q_{\bar{k}}$ variables have value greater than 0. That is the site in which the assigned demand is split between the largest number of possible capacity values. Given the capacity value p for which $\sum_{q \in \mathcal{Q} | q \leq p} z^q_{\bar{k}}$ is nearest to 0.5, we impose $\sum_{q \in \mathcal{Q} | q \leq p} z^q_{\bar{k}} = 0$ in one branch and $\sum_{q \in \mathcal{Q} | q > p} z^q_{\bar{k}} = 0$ in the other branch. These constraints are handled by just fixing to 0 some z variables.

3. When all the z variables take integer values, we branch on the (4) constraints: like in level 1, we select the index $\bar{j}$ for which $\sum_{t \in \mathcal{T}} y_{jt}$ is nearest to 0.5, since in an integer solution $\sum_{t \in \mathcal{T}} y_{jt}$ is 1 if a facility is placed in j or 0 otherwise. We impose in one branch that $\sum_{t \in \mathcal{T}} y_{\bar{j}t} = 1$ and in the other branch that $\sum_{t \in \mathcal{T}} y_{\bar{j}t} = 0$. As in the previous level, both conditions can simply be enforced by fixing to zero either slack variables or y variables; furthermore, in the second branch we remove columns related to $\bar{j}$ from the RMP and generate no more column of each set $\mathcal{S}_{\bar{j}t}$.

4. When all the values $\sum_{t \in \mathcal{T}} y_{jt}$ are 0 or 1, we select the index $\bar{j}$ for which the largest number of $y_{\bar{j}t}$ variables have value greater than 0. That is the site in which the assigned demand is split between the largest number of possible devices. Given a capacity b_r for which half of the fractional $y_{\bar{j}t}$ values have $b_t \leq b_r$, we impose $\sum_{t \in \mathcal{T} | b_t \leq b_k} y_{\bar{j}t} = 0$ in one branch and $\sum_{t \in \mathcal{T} | b_t > b_r} y_{\bar{j}t} = 0$ in the other branch. Once again, these constraints are handled by just removing suitable columns from the RMP and not generating any new column in the sets $\mathcal{S}_{\bar{j}t}$.

5. When also each y_{jt} variable takes an integer value, we perform branching on the $\sum_{j \in \mathcal{J}} x_{ij} = 1$ constraint by searching for the client $\bar{i}$ which is (fractionally) assigned to the largest number of facilities. We split $\mathcal{J}$ in two sets $\mathcal{J}^l$ and $\mathcal{J}^r$, each containing at least one site j for which $x_{\bar{i}j} > 0$, and we create two branches imposing respectively $\sum_{j \in \mathcal{J}^l} x_{\bar{i}j} = 0$ and $\sum_{j \in \mathcal{J}^r} x_{\bar{i}j} = 0$. As before, these conditions are easy to enforce by just removing items in the pricing subproblems.

6. Finally, when also these variables take integer value, we perform branching on the $\sum_{k\in\mathcal{K}} w_{jtk} \geq y_{jt}$ constraints by searching for the mid level facility $\bar{j}$ which is (fractionally) assigned to the largest number of high level facilities, and we create two branches as those applied in level 5.

We remark that these branching decisions require no changes in the structure of the pricing problem, which remains a simple 0–1 knapsack problem through all the enumeration process.

Of course, these rules can be applied in an arbitrary order during the exploration of the branching tree: we identified the following sequence of choices giving best performance during preliminary experimental results. First, we consider branching rules in their order from 1 to 6: we skip level 1 rule if each $|\sum_{q\in\mathcal{Q}} z_k^q - 0.5| > 0.4$, level 2 rule if each $|z_k - 0.5| > 0.4$, level 3 rule if each $|\sum_{t\in\mathcal{T}} y_{jt} - 0.5| > 0.4$, and the remaining branching rules if the number of fractional variables in each constraint is less than 3. If all rules are skipped, we consider again all branching rules in their order from 1 to 6, skipping each rule only if all integral values are found.

Still according to preliminary experimental tuning, the branching tree is explored with a best-bound-first policy.

4 Computational results

We implemented our algorithm (DBP) in C++, using SCIP [2] as a framework for building branch-and-price-and-cut algorithms, CPLEX 11.0 dual simplex solver [7] for solving LP subproblems and an adaptation of MINKNAP algorithm [10] [4] for solving the KP subproblems. In our experiments, we also exploited the general purpose cut generation of SCIP.

We consider CPLEX 11.0 ILP solver (CPX), with default parameter settings, as a benchmark method. In fact, CPX relies on a state-of-the-art branch-and-cut method, which includes general purpose cut generation and primal heuristics; it proved to be effective in solving many capacitated location problems [4]. Our experiments ran on a Centrino Core2 3 GHz workstation equipped with 2GB of RAM.

In order to test our method on a wide range of instances, we consider two datasets adapted from those in the literature.

Since our research is motivated by a practical application, Dataset 1 aims at testing our method on instances with a wide range of features but no particular structure, as typically happens in practice. It consists of 71 instances from [18], in which the number of clients range from 50 to 200 and the number of facilities from 10 to 50.

Dataset 2, instead, is selected for a stress-test of our method. It includes 24 instances from [19] and [3], in which the number of clients is 50 and the number of facilities range from 16 to 50. These instances have on the average a low ratio between the overall client weights and the high level facility capacities. In this kind of instances optimal fractional solutions tend to be very different from integer ones, even if the region described by capacity constraints is replaced by the convex hull of its extreme integer points.

The original instances describe single-level single-source capacitated facility location problems. We keep the set of clients and candidate facilities in the original instances respectively as $\mathcal{I}$ and $\mathcal{J}$ and set each coefficient d_{ij} as the distance between client i and facility j in the original instance; then, we consider each candidate location site to be both a mid level and high level candidate location site, that is we set $\mathcal{K} = \mathcal{J}$, we randomly assign each candidate site to a client and keep as distance l_{jk} between candidate mid level and high level location sites the distance between the corresponding client.

Let $\bar{b}$ and $\bar{f}$ respectively be the average capacity and the average installation cost of a facility in an original instance; we consider 5 types of device to be available at each node: we select $T-1$ as *standard* device, fixing its capacity to $\bar{b}$, and its setup cost f_{T-1} to $\bar{f} \cdot 2/3$. In order to model economies of scale of practical applications we set the capacity b_t and the setup cost f_t of each device $t < T-1$ respectively to $b_{t+1}/2$ and $1.1 \cdot f_{t+1}/2$; b_T is fixed to $2 \cdot b_{T-1}$ and f_T is fixed to $2 \cdot f_{T-1}/1.1$. The installation cost c_j of each candidate mid level facility j is set to 1/3 the original one; in this way, the installation and setup costs for installing device $T-1$ in site j is about the installation cost of a facility in site j in the original instance.

Finally, we set the capacity B of a high level facility in the corresponding instance of our dataset as follows

$$\max\left(\left\lceil \frac{1}{3}\cdot N\cdot \frac{\bar{b}}{\sum_{i\in\mathcal{I}} a_i}\cdot \bar{b}\right\rceil, b_T\right)$$

and the installation cost g_k of each candidate high level facility k as follows

$$\left\lceil \frac{B}{\bar{b}}\cdot c_k\right\rceil;$$

these values well represent capacities in practical applications.

Tables 1 and 2 describe all instances in detail: the name of each instance is reported in column (inst); columns ($|\mathcal{I}|$), ($|\mathcal{J}|$) and ($|\mathcal{K}|$) report respectively the number of clients, mid level facility candidate sites and high level facility candidate sites in the instance; column (IC Type) is marked with U if all coefficients c_j and g_k have the same value, that is when the instance has uniform installation costs, is marked with H otherwise, that is the instance has heterogeneous installation costs; columns (mid level install) and (mid level assign) report the average installation and assignment costs for the mid level location problem; columns (high level install) and (high level assign) report the same values for the high level location problem; subsequent columns ($\sum_{i\in\mathcal{I}} a_i$), ($b_{T-1}$), ($B$) and ($|\mathcal{Q}|$) respectively report the sum of client demands, the capacity provided by device $T-1$, the capacity provided by a high level facility and the maximum number of capacity units provided by a high level facility, that is the number of binary z variables for each $k\in\mathcal{K}$; the rightmost four columns respectively report the ratio between average installation and assignment costs in the mid level location problem, the ratio between average installation and assignment costs in the high level location problem, the ratio between customer demands and capacity of device $T-1$ and the ratio between customer demands and capacity of a high level facility. In particular, the last two values represent respectively the minimum number of mid level facilities equipped with device $T-1$ and the minimum number of high level facilities needed to cover all the clients demand in a fractional solution.

4.1 Full discretized and extended formulations

As sketched through the paper the reformulation by discretization we applied to the high level location problem can be applied as well to the mid level location problem; on the other hand, the Dantzig-Wolfe reformulation we applied to the mid level location problem can be applied as well to the high level location problem. However, applying discretization to both levels yielded models with too many binary variables to be effectively managed by general purpose solvers; in the meanwhile, applying Dantzig-Wolfe reformulation to both levels required a two-level column generation scheme which showed to strongly suffer stability problems, and to be numerically very difficult to handle; in any case, no significant improvement in terms of quality of bounds can be obtained in this way.

Therefore, we decided to exclude any method based on these formulations from the following detailed analysis.

4.2 Comparison of pricing policies

The first set of experiments aims at tuning the root node performance of our algorithm DBP. We compare five pricing strategies: solving the pricing problem with no stabilization nor improvement method (plain), with the stabilization method described in Section 3 but no further performance improvement technique (stab) with the addition of dual cuts but no stabilization (dualcuts) with the addition of dualcuts and pricing improvement technique (a) (see Section 3) (dualcuts+imppricing) and the full version including stabilization, addition of dual cuts and pricing improvement technique (a) (full). We also experimented using pricing improving technique (b), observing it is not worth using such technique; therefore we do not include these results.

Our computational results are reported in Table 3 and Table 4 for Dataset 1 and Dataset 2 respectively. Besides including a column reporting the name of each instance, each Table is composed by five horizontal

blocks, one for each pricing technique. In each block the number of pricing iterations and the CPU time needed to complete the column generation process are reported. The last row of each Table contains average values.

Stabilization shows to substantially improve performance. In fact, the number of iterations needed to reach optimality drops by more than 40% in both Datasets; even if the single column generation iteration might take longer, since more 0–1 knapsack problems need to be solved, in Dataset 1 also the CPU time needed to fully optimize the root node drops on the average by 60%; the instances in Dataset 2 seem easier to solve, and therefore the CPU time improvement in much smaller.

The benefit of dual cuts is evident only in terms of CPU time, and only as it allows to include pricing improvement techniques; in this way a further 5% improvement on CPU time with respect to stabilization only can be obtained.

4.3 Performances on the root node

In a second set of experiments we compare the primal and dual bounds which can be achieved by four solution methods. First, we include in the comparison CPX using formulation (IP), after the automatic addition of cuts at the root node (CPX) and CPX using the discretized formulation (HD), after the automatic addition of cuts at the root node, that is only the first step of our method (HD). Then, we consider the column generation algorithm implemented in DBP, using formulation (IMP), before and after the automatic addition of cuts a the root node, that is using all the features of our method (HDCG). Finally, we include in the comparison a column generation algorithm using the following formulation

$$\min \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}_{jt}} C_s v_s + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} l_{jk} w_{jtk} + \sum_{k\in\mathcal{K}} g_k z_k$$

$$s.t.\ \ (16),(17),(18),(6), w_{jtk} \le z_k,\ \ \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \forall k \in \mathcal{K}$$

$$v_s, w_{jtk}, z_k \in \{0,1\} \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \forall s \in \mathcal{S}_{jt}$$

after the automatic addition of cuts a the root node (CG); that is, we perform only the second step of our reformulation method, and therefore we keep capacity constraints (6) of the compact formulation instead of their discretized and strengthened counterpart (19), (20), (21) and (22); this allows us to test the behavior of our method when reformulation by discretization is not performed.

The results of these experiments are split among Tables 5 and 7 for the instances in Dataset 1, and among Tables 6 and 8 for instances in Dataset 2. Besides including the name of each instance in the first column, each of these Tables consists of four horizontal blocks, one for each solution method, as indicated in the leading row. Blocks CPX, HD and CG of Table 5 and 6 include the primal and dual bounds obtained by the corresponding method; block HDCG includes in the following order the dual bound obtained before the automatic addition of cuts, the number of pricing iterations needed for column generation to be over, the number of cuts generated by SCIP, the dual bound obtained after the addition of cuts and the primal bound obtained by SCIP heuristics at the root node.

Instead, in Tables 7 and 8 we report for each instance and for each solution method the gap between the dual bound given by the solution method and the best known primal bound at the root node and the CPU time required to complete the computation; the HDCG block is split in two, the first sub-block refers to HDCG without the SCIP addition of cuts, while the second one refers to HDCG after the automatic addition of cuts. The last row of each Table contains average values.

As expected, we observe a different behavior in Dataset 1 and Dataset 2.

For what concerns Dataset 1, by looking at block HDCG in Table 5, one can notice that the dynamic addition of valid cuts to formulation (IMP) yields on the average only a slight improvement in the quality of the bound; indeed, very few cuts are generated on the average. This is not the case for CG method, where we observed up to 131 cuts, and almost 15 cuts to be on the average dynamically generated, yielding an improvement of about 1% in the quality of the bound. We argue that by discretizing the high level location problem one gets a better description of the convex hull of extreme integer points for the region described by constraints (6), and therefore less violated cuts arise from these constraints. At the

same time, we observed a reduction of about 30% on the number of pricing iterations needed by HDCG with respect to CG to complete the column generation process, proving HDCG to be more stable. Then, by comparing block CPX with block HD, and block CG with block HDCG in Table 7, it is clear that the discretization step of our reformulation is crucial: the primal–dual gap nearly halves, and the CPU time substantially drops. The Dantzig-Wolfe step of our reformulation helps to further improve the quality of dual bounds, at the expense of higher CPU time.

A different behavior is observed in Dataset 2, by comparing block CPX with block CG, and block HD with block HDCG of Table 8: here the quality of dual bounds improves only after the Dantzig-Wolfe step of our reformulation. Also in this case the quality improvement comes at the price of CPU time.

For any solution method, the bounds obtained in Dataset 2 are tighter than those obtained in Dataset 1.

In almost all the cases, the best primal bound on the root node is either found by CPX or by HD, stressing the need for special purpose heuristics when using column generation techniques.

4.4 Obtaining optimal solutions

In a third set of experiments we tried to optimize the whole problem, imposing a time limit of 7200s of CPU time to each computation. We compared three solution methods: CPX using formulation (IP) (CPX), CPX using the discretized formulation (HD) and DBP using formulation (IMP), with stabilization and improved pricing strategy (DBP).

The results of these experiments are reported in Table 9 and Table 10 for Dataset 1 and Dataset 2 respectively. The name of each instance is indicated in the first column; then, each Table is composed by three horizontal blocks, one for each solution method. Each block contains two columns. In the first one we report the CPU time required to solve each instance; when optimality is not proved, the first column is marked with a dash, and in the second column we report the gap between the best primal (PB) and dual bound (DB) found during optimization, computed as (PB - DB) / DB. In the last row of each table we report, for each solution method, the average CPU time needed to solve the instances which all methods were able to solve to proven optimality, and the number of instances in the dataset solved to proven optimality.

Also in this case, a different behavior is observed in the two Datasets.

CPX is able to solve to optimality only 43 of the 71 instances in Dataset 1 (that is about 60%). By performing the discretization step of our reformulation, 19 more instances can be solved, and the average CPU time needed to optimize each instance strongly decreases. A detailed look at the 'time' column of Table 10 show that no strict domination can be observed between HD and HDCG. Our full method DBP fails in optimizing instance p21 due to memory consumption problems, but allows to solve two more instances (p56 and p57), yielding proven optimal solutions for more than 91% of the instances in the Dataset. It is interesting to note that in some instances the primal solution found by HD after two hours of computation is substantially better than that found by HDCG; this once again highlights how special purpose heuristics could increase the performances of HDCG.

Instances in Dataset 2 are generally easier, as all methods are able to solve most of them in few seconds.

Among these, however, instances cap122, cap123 and cap124 show a particular behavior, as they turn out to be very hard to be solved: CPX performs best on the average, and HDCG is not able to prove optimality. On the average, by looking at the instance details in Table 2, we observed that when the capacity of high level facilities is large, the hardness of the problem increases as the ratio between installation and assignment costs increases. We tried to explain this behavior as follows.

In this kind of instances, the continuous relaxation of IMP tends to be highly fractional, as it is convenient to open several fractions of the facilities as close as possible to the clients, instead of serving all demand from one site; however, as the installation costs increase, in an integer solution becomes appealing to open just one facility, even at the expense of higher assignment costs. Thus, optimal fractional and integer solutions can be very different, making it hard to reach optimality through branching. Among the instances with this structure, cap121, cap122, cap123 and cap124 have the highest number of candidate

sites for both levels of facility: they have as many clients as candidate sites of both levels; therefore they represent the worst case of this scenario. On the other hand, instances from cap131 to cap134, share many features with instances from cap121 to cap124, but are not so difficult to solve: the different behavior may be explained by the fact that in this case the ratio between total client demand mid level facility capacity is such that only one facility can provide enough capacity for all clients, making these instances pretty like single-level single-facility location problems.

Finally, as a general observation, the value $|\mathcal{Q}|$, indicating the number of binary z variables for each $k \in \mathcal{K}$, does not seem to be crucial for the performance of neither HD nor HDCG.

5 Conclusions

In this paper we tackle a Two-level facility location problem, with practical applications in telecommunications network design, in which is need to both locate and dimension facilities. The dimensioning features of our problem have never been considered in the literature in such level of detail, and similar previous models have always been solved heuristically.

We propose different formulations for the problem, discussing how to successfully exploit reformulation by discretization and Dantzig-Wolfe decomposition techniques: neither of these two reformulation tools is fully useful on its own, but we experimented on how to gracefully integrate them.

As a result, we devise an exact optimization algorithm based on a hybrid formulation and column generation techniques, enriched with ad-hoc stabilization and pricing improvement techniques.

Our experimental analysis show that this exact optimization algorithm is much more effective than a general purpose solver in optimizing instances with no particular structure, as those arising in practice; in the meanwhile, we identified a class of hard synthetic instances, in which our approach shows performance problems, and we tried to give explanations to this phenomenon.

As highlighted by our experimental analysis, primal bounds are worth of further studying: suitable and efficient ad-hoc heuristics, to be embedded in column generation schemes, can improve the performance of our approach.

Appendix

The reformulation by discretization of model (HD) can be completed as follows. We suppose data to be integer, otherwise suitable scaling has to be applied.

For the ease of notation, we define a fictious device capacity $b_0 = 0$. Let $\mathcal{P}_t = \{b_{t-1} + 1 \dots b_t\}$ be the set of capacity values which can actually be provided by a mid level facility equipped with device t. Still following the technique proposed in [8], we substitute each variable y_{jt} with a set of binary variables y^q_{jt} for each j in $\mathcal{J}$, for each $t \in \mathcal{T}$ and for each $q \in \mathcal{P}_t$; each variable y^q_{jt} takes value 1 if a high level facility is built in j, is equipped with device t and serves exactly q capacity units, 0 otherwise. Therefore $y_{jt} = \sum_{q \in \mathcal{P}_t} y^q_{jt}$ and $\sum_{q \in \mathcal{P}_t} y^q_{jt} \leq 1$.

The reformulation by discretization of the whole problem, in which the total installation, setup and connection cost is minimized, is thus the following:

$$\text{FD)} \min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} d_{ij} x_{ij} + \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (c_j + f_t) \sum_{q \in \mathcal{P}_t} y^q_{jt} + \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} l_{jk} w_{jtk} + \sum_{k \in \mathcal{K}} g_k \sum_{q \in \mathcal{Q}} z^q_k \tag{25}$$

$$\sum_{j \in \mathcal{J}} x_{ij} \geq 1, \forall\ i \in \mathcal{I} \tag{26}$$

$$\sum_{i \in \mathcal{I}} a_i x_{ij} = \sum_{t \in \mathcal{T}} \sum_{q \in \mathcal{P}_t} q y^q_{jt}, \quad \forall\ j \in \mathcal{J} \tag{27}$$

$$\sum_{t \in \mathcal{T}} \sum_{q \in \mathcal{P}_t} y^q_{jt} \leq 1, \quad \forall j \in \mathcal{J} \tag{28}$$

$$\sum_{k\in\mathcal{K}} w_{jtk} \geq \sum_{q\in\mathcal{P}_t} y^q_{jt}, \forall\ j \in \mathcal{J}, \forall\ t \in \mathcal{T} \tag{29}$$

$$\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} \tilde{b}_t w_{jtk} = \sum_{q\in\mathcal{Q}} q z^q_k, \forall\ k \in \mathcal{K} \tag{30}$$

$$\sum_{q\in\mathcal{Q}} z^q_k \leq 1, \forall k \in \mathcal{K} \tag{31}$$

$$x_{ij}, w_{jtk}, y^p_{jt}, z^q_k \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \forall p \in \mathcal{P}_t, \forall q \in \mathcal{Q} \tag{32}$$

As for model (HD), the LP relaxation bound obtained by (FD) is not tighter than the one given by (IP), but model (FD) can be enriched by the following strengthening constraints [8].

First, by introducing variables y^q_{jt}, we can express the overall capacity actually provided by mid level facility as $\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{q\in\mathcal{P}_t} q y^q_{jt}$. This has to be equal to the overall customers demand

$$\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{q\in\mathcal{P}_t} q y^q_{jt} = \sum_{i\in\mathcal{I}} a_i. \tag{33}$$

This constraint can be relaxed in a greater than or equal form, and the division and rounding method described in Subsection 2.1 can be applied, obtaining for each $p \in \bigcup_{t\in\mathcal{T}} \mathcal{P}_t$ the following set of inequalities

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\sum_{q\in\mathcal{P}_t} \left\lceil \frac{q}{p} \right\rceil y^q_{jt} \geq \left\lceil \frac{\sum_{i\in\mathcal{I}} a_i}{p} \right\rceil. \tag{34}$$

A similar method can be applied by relaxing (33) in less than or equal form; for each $p \in \bigcup_{t\in\mathcal{T}} \mathcal{P}_t$ we consider the following inequality

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\sum_{q\in\mathcal{P}_t} \frac{q}{p} y^q_{jt} \leq \frac{\sum_{i\in\mathcal{I}} a_i}{p}.$$

and we round down to the nearest integer each term on the left hand side, still obtaining a valid inequality

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\sum_{q\in\mathcal{P}_t} \left\lfloor \frac{q}{p} \right\rfloor y^q_{jt} \leq \frac{\sum_{i\in\mathcal{I}} a_i}{p};$$

since now the left hand side is always integer, it is possible to round down to the nearest integer the right hand side, still obtaining a valid inequality

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\sum_{q\in\mathcal{P}_t} \left\lfloor \frac{q}{p} \right\rfloor y^q_{jt} \leq \left\lfloor \frac{\sum_{i\in\mathcal{I}} a_i}{p} \right\rfloor \tag{35}$$

The following consistency constraints are also valid for this formulation: for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$

$$x_{ij} \leq \sum_{t\in\mathcal{T}}\sum_{q\in\mathcal{P}_t} y^q_{jt} \tag{36}$$

that is, when no mid level facility is built in j, no client can be assigned to j.

Inequalities (13) and (14) are valid for this formulation as well.

References

[1] K. Aardal, F.A. Chudak, and D.B Shmoys. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Information Processing Letters*, 72:161–167, 1999.

[2] T. Achterberg. Constraint Integer Programming. *PhD Thesis, Technische Universität Berlin*, available online, 2007.

[3] R.K. Ahuja, J.B. Orlin, S. Pallottino, M.P. Scaparra, and M.G. Scutellà. A multi-exchange heuristic for the single source capacitated facility location problem. *Management Science*, 50(6):749:760, 2004.

[4] A. Ceselli, F. Liberatore, and G. Righini. A computational evaluation of a general branch-and-price framework for capacitated network location problems. *Annals of Operations Research*, available online, 2008.

[5] S. Chamberland. An Efficient Algorithm for Designing Reliable IP Networks with an Access/Edge/Core Hierarchical Structure. *Networks 2008*, conference presentation, Budapest, 2008.

[6] P. Chardaire, J.-L. Lutton, and A. Sutter. Upper and lower bounds for the two-level simple plant location problem. *Annals of Operations Research*, 86:117–140, 1999.

[7] ILOG CPLEX 11.0 Users Manual. ILOG Inc, 2007

[8] L. Gouveia, F. Saldanha-da-Gama. On the capacitated concentrator location problem: a reformulation by discretization. *Computers and Operations Research*, 33(5):1242–1258, 2006.

[9] A.A.V.. Ignacio, V.J.M.F. Filho, and R.D. Galvao. Lower and upper bounds for a two-level hierarchical location problem in computer networks. *Computers and Operations Research*, 35:1982–1998, 2008.

[10] D. Pisinger. A minimal algorithm for the 0-1 knapsack problem. *Operations Research*, 45:758–767, 1997.

[11] G. Sahin and H. Süral. A review of hierarchical facility location models. *Computers and Operations Research*, 34:2310–2331, 2007.

[12] D. Tcha and B. Lee. A branch-and-bound algorithm for the multi-level uncapacitated location problem. *European Journal of Operations Research*, 18:35–43, 1984.

[13] S. Tragantalerngsak, J. Holt, and M. Ronnqvist. Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, 102(3):611–625, 1997.

[14] S. Tragantalerngsak, J. Holt, and M. Ronnqvist. An exact method for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, 123:473–489, 2000.

[15] M. Trampont, C. Destr, and A. Faye. Solving a hierarchical network design problem with two stabilized column generation approaches. *INOC 2009 - International Network Optimization Conference*, Pisa, 2009.

[16] E. Uchoa. A new stabilization method for column generation. *Column Generation Workshop*, conference presentation, Aussois, 2008.

[17] J.M. Valerio de Carvalho. Using Extra Dual Cuts to Accelerate Column Generation. *INFORMS Journal on Computing 17* 2:175-182, 2005.

[18] K. Holmberg, M. Ronnqvist, D. Yuan. An exact algorithm for the capacitated facility location problem with single sourcing. *European Journal of Operational Research 113*, 544-559, 1999.

[19] J.E. Beasley. OR-Library: Distributing test problems by electronic mail. *Journal of Operational Research Society 41*, 1069-1072, 1990.

[20] H. Kellerer, U. Pferschy, D. Pisinger (Eds). Knapsack Problems. Springer-Verlag, 2004.

[21] G. Desaulniers, J. Desrosiers, M.M. Solomon (Eds). Column Generation. Springer-Verlag, 2005

Inst.	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{K}\rvert$	IC Type	mid level install	mid level assign	high level install	high level assign	$\sum_{i\in\mathcal{I}} a_i$	b_{T-1}	B	$\lvert\mathcal{Q}\rvert$	mid level costs ratio	high level costs ratio	$\frac{\sum_{i\in\mathcal{I}} a_i}{b}$	$\frac{\sum_{i\in\mathcal{I}} a_i}{B}$
p10	50	10	10	U	100.00	398.46	1200.00	371.38	1456	300	1031	27	0.25	3.23	4.85	1.41
p11	50	10	10	U	167.00	398.46	2000.00	371.38	1456	300	1031	27	0.42	5.39	4.85	1.41
p12	50	10	10	U	234.00	398.46	2800.00	371.38	1456	300	1031	27	0.59	7.54	4.85	1.41
p13	50	20	20	H	147.80	400.53	1327.80	395.24	2855	396	912	18	0.37	3.36	7.21	3.13
p14	50	20	20	U	100.00	400.53	900.00	395.24	2855	396	912	18	0.25	2.28	7.21	3.13
p15	50	20	20	U	167.00	400.53	1500.00	395.24	2855	396	912	18	0.42	3.80	7.21	3.13
p16	50	20	20	U	234.00	400.53	2100.00	395.24	2855	396	912	18	0.58	5.31	7.21	3.13
p17	50	20	20	H	147.80	400.53	1327.80	395.24	2855	400	935	18	0.37	3.36	7.14	3.05
p18	50	20	20	U	100.00	400.53	900.00	395.24	2855	400	935	18	0.25	2.28	7.14	3.05
p19	50	20	20	U	167.00	400.53	1500.00	395.24	2855	400	935	18	0.42	3.80	7.14	3.05
p1	50	10	10	H	144.10	398.46	1294.20	371.38	1456	254	735	22	0.36	3.48	5.73	1.98
p20	50	20	20	U	234.00	400.53	2100.00	395.24	2855	400	935	18	0.58	5.31	7.14	3.05
p21	50	20	20	H	147.80	400.53	1327.80	395.24	2855	500	1460	23	0.37	3.36	5.71	1.96
p22	50	20	20	U	100.00	400.53	900.00	395.24	2855	500	1460	23	0.25	2.28	5.71	1.96
p23	50	20	20	U	167.00	400.53	1500.00	395.24	2855	500	1460	23	0.42	3.80	5.71	1.96
p24	50	20	20	U	234.00	400.53	2100.00	395.24	2855	500	1460	23	0.58	5.31	5.71	1.96
p25	150	30	30	H	138.07	613.43	2891.23	639.33	2970	408	2800	54	0.23	4.52	7.28	1.06
p26	150	30	30	U	100.00	613.43	2100.00	639.33	2970	408	2800	54	0.16	3.28	7.28	1.06
p27	150	30	30	U	167.00	613.43	3500.00	639.33	2970	408	2800	54	0.27	5.47	7.28	1.06
p28	150	30	30	U	234.00	613.43	4900.00	639.33	2970	408	2800	54	0.38	7.66	7.28	1.06
p29	150	30	30	H	138.07	613.43	2478.20	639.33	2970	300	1516	39	0.23	3.88	9.90	1.96
p2	50	10	10	U	100.00	398.46	900.00	371.38	1456	254	735	22	0.25	2.42	5.73	1.98
p30	150	30	30	U	100.00	613.43	1800.00	639.33	2970	300	1516	39	0.16	2.82	9.90	1.96
p31	150	30	30	U	167.00	613.43	3000.00	639.33	2970	300	1516	39	0.27	4.69	9.90	1.96
p32	150	30	30	U	234.00	613.43	4200.00	639.33	2970	300	1516	39	0.38	6.57	9.90	1.96
p33	150	30	30	H	138.07	613.43	2891.23	639.33	2970	400	2694	53	0.23	4.52	7.43	1.10
p34	150	30	30	U	100.00	613.43	2100.00	639.33	2970	400	2694	53	0.16	3.28	7.43	1.10
p35	150	30	30	U	167.00	613.43	3500.00	639.33	2970	400	2694	53	0.27	5.47	7.43	1.10
p36	150	30	30	U	234.00	613.43	4900.00	639.33	2970	400	2694	53	0.38	7.66	7.43	1.10
p37	150	30	30	H	138.07	613.43	4543.37	639.33	2970	600	6061	80	0.23	7.11	4.95	0.49
p38	150	30	30	U	100.00	613.43	3300.00	639.33	2970	600	6061	80	0.16	5.16	4.95	0.49
p39	150	30	30	U	167.00	613.43	5500.00	639.33	2970	600	6061	80	0.27	8.60	4.95	0.49
p3	50	10	10	U	167.00	398.46	1500.00	371.38	1456	254	735	22	0.42	4.04	5.73	1.98
p40	150	30	30	U	234.00	613.43	7700.00	639.33	2970	600	6061	80	0.38	12.04	4.95	0.49
p41	90	10	10	H	129.70	136.26	2713.90	143.69	1299	276	1751	50	0.95	18.89	4.71	0.74
p42	80	20	20	H	132.15	137.99	2765.70	142.29	1163	291	1932	52	0.96	19.44	4.00	0.60
p43	70	30	30	H	133.93	137.88	2804.90	129.94	1068	295	1901	51	0.97	21.59	3.62	0.56
p44	90	10	10	H	129.70	175.74	2326.20	175.71	1568	276	1451	41	0.74	13.24	5.68	1.08
p45	80	20	20	H	132.15	172.38	2370.60	157.92	1401	291	1603	43	0.77	15.01	4.81	0.87
p46	70	30	30	H	133.93	179.91	2404.20	173.29	1246	295	1629	44	0.74	13.87	4.22	0.76
p47	90	10	10	H	129.70	158.16	2326.20	152.25	1560	276	1458	41	0.82	15.28	5.65	1.07
p48	80	20	20	H	132.15	160.03	2370.60	140.10	1430	291	1571	42	0.83	16.92	4.91	0.91
p49	70	30	30	H	133.93	159.30	2404.20	156.01	1250	295	1624	43	0.84	15.41	4.24	0.77
p4	50	10	10	U	234.00	398.46	2100.00	371.38	1456	254	735	22	0.59	5.65	5.73	1.98
p50	100	10	10	H	129.70	162.05	2713.90	156.94	1458	276	1733	49	0.80	17.29	5.28	0.84
p51	100	20	20	H	132.15	163.09	2765.70	164.17	1458	291	1926	52	0.81	16.85	5.01	0.76
p52	100	10	10	H	129.70	213.64	2326.20	248.12	1724	276	1466	41	0.61	9.38	6.25	1.18
p53	100	20	20	H	132.15	215.09	2370.60	211.50	1724	291	1629	44	0.61	11.21	5.92	1.06
p54	100	10	10	H	129.70	191.31	2326.20	160.65	1810	276	1396	39	0.68	14.48	6.56	1.30
p55	100	20	20	H	132.15	191.53	2370.60	171.28	1810	291	1551	41	0.69	13.84	6.22	1.17
p56	200	30	30	U	167.00	369.56	2500.00	349.06	6076	400	1756	35	0.45	7.16	15.19	3.46
p57	200	30	30	U	267.00	369.56	4000.00	349.06	6076	400	1756	35	0.72	11.46	15.19	3.46
p58	200	30	30	U	500.00	369.56	7500.00	349.06	6076	400	1756	35	1.35	21.49	15.19	3.46
p59	200	30	30	H	336.30	369.56	5039.83	349.06	6076	400	1756	35	0.91	14.44	15.19	3.46
p5	50	10	10	H	144.10	398.46	1294.20	371.38	1456	200	458	18	0.36	3.48	7.28	3.18
p60	200	30	30	U	167.00	369.56	3500.00	349.06	6076	600	3950	52	0.45	10.03	10.13	1.54
p61	200	30	30	U	267.00	369.56	5600.00	349.06	6076	600	3950	52	0.72	16.04	10.13	1.54
p62	200	30	30	U	500.00	369.56	10500.00	349.06	6076	600	3950	52	1.35	30.08	10.13	1.54
p63	200	30	30	H	336.30	369.56	7055.77	349.06	6076	600	3950	52	0.91	20.21	10.13	1.54
p64	200	30	30	U	167.00	369.56	4500.00	349.06	6076	800	7023	70	0.45	12.89	7.60	0.87
p65	200	30	30	U	267.00	369.56	7200.00	349.06	6076	800	7023	70	0.72	20.63	7.60	0.87
p66	200	30	30	U	500.00	369.56	13500.00	349.06	6076	800	7023	70	1.35	38.68	7.60	0.87
p67	200	30	30	H	343.07	370.78	7196.47	349.07	6120	600	3922	52	0.93	20.62	10.20	1.56
p68	200	30	30	U	167.00	369.56	3500.00	349.06	6076	604	3993	52	0.45	10.03	10.06	1.52
p69	200	30	30	U	267.00	369.56	5600.00	349.06	6076	604	3993	52	0.72	16.04	10.06	1.52
p6	50	10	10	U	100.00	398.46	900.00	371.38	1456	200	458	18	0.25	2.42	7.28	3.18
p70	200	30	30	U	500.00	369.56	10500.00	349.06	6076	604	3993	52	1.35	30.08	10.06	1.52
p71	200	30	30	H	336.30	369.56	7055.77	349.06	6076	604	3993	52	0.91	20.21	10.06	1.52
p7	50	10	10	U	167.00	398.46	1500.00	371.38	1456	200	458	18	0.42	4.04	7.28	3.18
p8	50	10	10	U	234.00	398.46	2100.00	371.38	1456	200	458	18	0.59	5.65	7.28	3.18
p9	50	10	10	H	144.10	398.46	1725.60	371.38	1456	300	1031	27	0.36	4.65	4.85	1.41

Table 1: Dataset 1 (derived from Holmberg dataset)

Inst.	$\|\mathcal{I}\|$	$\|\mathcal{J}\|$	$\|\mathcal{K}\|$	IC Type	mid level install	mid level assign	high level install	high level assign	$\sum_{i\in\mathcal{I}} a_i$	b_{T-1}	B	$\|\mathcal{Q}\|$	mid level costs ratio	high level costs ratio	$\frac{\sum_{i\in\mathcal{I}} a_i}{b}$	$\frac{\sum_{i\in\mathcal{I}} a_i}{B}$
cap101	50	25	25	H	2400.00	48600.60	122400.00	43285.90	58268	58268	971134	133	0.05	2.83	1.00	0.06
cap102	50	25	25	H	4000.32	48600.60	204000.00	43285.90	58268	58268	971134	133	0.08	4.71	1.00	0.06
cap103	50	25	25	H	5600.64	48600.60	285600.00	43285.90	58268	58268	971134	133	0.12	6.60	1.00	0.06
cap104	50	25	25	H	8000.64	48600.60	408000.00	43285.90	58268	58268	971134	133	0.16	9.43	1.00	0.06
cap121	50	50	50	H	2450.00	51501.10	36750.00	51501.10	58268	15000	64358	34	0.05	0.71	3.88	0.91
cap122	50	50	50	H	4083.66	51501.10	61250.00	51501.10	58268	15000	64358	34	0.08	1.19	3.88	0.91
cap123	50	50	50	H	5717.32	51501.10	85750.00	51501.10	58268	15000	64358	34	0.11	1.67	3.88	0.91
cap124	50	50	50	H	8167.32	51501.10	122500.00	51501.10	58268	15000	64358	34	0.16	2.38	3.88	0.91
cap131	50	50	50	H	2450.00	51501.10	124950.00	51501.10	58268	58268	971134	133	0.05	2.43	1.00	0.06
cap132	50	50	50	H	4083.66	51501.10	208250.00	51501.10	58268	58268	971134	133	0.08	4.04	1.00	0.06
cap133	50	50	50	H	5717.32	51501.10	291550.00	51501.10	58268	58268	971134	133	0.11	5.66	1.00	0.06
cap134	50	50	50	H	8167.32	51501.10	416500.00	51501.10	58268	58268	971134	133	0.16	8.09	1.00	0.06
cap51	50	16	16	H	5469.38	44663.40	49218.80	39403.10	58268	10000	28604	22	0.12	1.25	5.83	2.04
cap61	50	16	16	H	2343.75	44663.40	35156.20	39403.10	58268	15000	64358	34	0.05	0.89	3.88	0.91
cap62	50	16	16	H	3906.56	44663.40	58593.80	39403.10	58268	15000	64358	34	0.09	1.49	3.88	0.91
cap63	50	16	16	H	5469.38	44663.40	82031.20	39403.10	58268	15000	64358	34	0.12	2.08	3.88	0.91
cap64	50	16	16	H	7813.12	44663.40	117188.00	39403.10	58268	15000	64358	34	0.17	2.97	3.88	0.91
cap71	50	16	16	H	2343.75	44663.40	119531.00	39403.10	58268	58268	971134	133	0.05	3.03	1.00	0.06
cap72	50	16	16	H	3906.56	44663.40	199219.00	39403.10	58268	58268	971134	133	0.09	5.06	1.00	0.06
cap73	50	16	16	H	5469.38	44663.40	278906.00	39403.10	58268	58268	971134	133	0.12	7.08	1.00	0.06
cap74	50	16	16	H	7813.12	44663.40	398438.00	39403.10	58268	58268	971134	133	0.17	10.11	1.00	0.06
cap91	50	25	25	H	2400.00	48600.60	36000.00	43285.90	58268	15000	64358	34	0.05	0.83	3.88	0.91
cap92	50	25	25	H	4000.32	48600.60	60000.00	43285.90	58268	15000	64358	34	0.08	1.39	3.88	0.91
cap93	50	25	25	H	5600.64	48600.60	84000.00	43285.90	58268	15000	64358	34	0.12	1.94	3.88	0.91
cap94	50	25	25	H	8000.64	48600.60	120000.00	43285.90	58268	15000	64358	34	0.16	2.77	3.88	0.91

Table 2: Dataset 2 (derived from Orlib dataset)

Inst.	plain		stab		dualcuts		dualcuts+IP		full	
	iter	time (s)	iter	time (s)	iter	time (s)	iter	time (s)	iter	time (s)
p10	42	0.62	50	0.44	39	0.63	40	0.58	50	0.43
p11	40	0.65	51	0.48	38	0.59	38	0.61	50	0.49
p12	42	0.66	41	0.41	50	0.72	46	0.73	43	0.39
p13	22	0.89	22	0.68	23	0.90	25	0.95	24	0.72
p14	21	1.02	27	0.94	23	0.98	23	0.98	24	0.77
p15	25	1.15	27	0.94	24	1.16	22	1.11	25	0.88
p16	24	1.04	23	0.75	26	1.37	26	1.24	27	0.89
p17	23	0.93	24	0.73	23	0.98	23	0.84	24	0.68
p18	21	1.01	20	0.83	21	1.01	19	0.89	25	0.73
p19	21	1.07	25	0.89	22	1.10	22	0.95	25	0.82
p1	40	0.42	53	0.39	36	0.51	39	0.44	47	0.34
p20	22	0.97	30	0.81	24	1.30	24	1.09	30	0.90
p21	25	1.28	26	1.06	24	1.41	23	1.22	27	1.06
p22	19	1.38	25	1.13	21	1.49	20	1.20	21	0.89
p23	26	1.65	27	1.31	26	1.83	28	1.63	24	1.20
p24	25	1.55	27	1.22	28	2.06	28	1.79	28	1.36
p25	193	55.21	95	21.46	212	61.18	208	47.70	88	17.43
p26	174	47.06	99	19.66	226	58.00	199	48.68	103	19.90
p27	357	65.05	137	25.48	295	76.90	295	59.62	108	22.19
p28	301	78.88	121	28.35	308	116.03	313	79.56	117	31.60
p29	165	44.89	93	19.50	182	63.05	190	46.49	83	19.65
p2	37	0.56	50	0.40	32	0.59	35	0.52	54	0.45
p30	198	38.50	130	24.03	218	73.91	188	48.15	92	21.00
p31	244	59.99	108	27.00	210	82.98	246	64.34	150	33.62
p32	304	68.59	184	36.71	363	100.18	328	67.27	145	30.60
p33	224	62.76	102	22.34	223	72.62	229	54.37	97	20.31
p34	211	48.23	100	19.43	215	64.06	210	45.05	101	20.01
p35	233	62.47	129	27.44	226	70.41	269	58.92	109	22.15
p36	386	75.42	125	27.23	367	112.24	324	80.23	123	32.23
p37	299	81.01	136	35.49	298	135.21	276	88.83	118	35.13
p38	256	61.58	141	28.06	243	80.11	293	58.76	109	24.70
p39	323	82.94	122	29.25	352	128.28	333	85.77	121	29.41
p3	38	0.59	61	0.49	40	0.62	39	0.50	51	0.43
p40	287	102.64	127	33.94	349	117.85	355	93.31	116	25.69
p41	116	6.48	82	2.91	113	7.07	120	6.18	83	2.58
p42	56	6.68	44	3.96	55	6.48	57	5.59	43	3.23
p43	40	7.89	34	5.25	41	7.98	43	6.66	31	3.78
p44	135	4.61	107	2.72	135	5.96	135	5.08	100	2.36
p45	84	8.70	79	5.53	84	10.37	81	8.70	56	4.52
p46	36	9.96	36	7.26	38	11.30	39	9.15	34	5.69
p47	389	13.58	127	4.47	372	15.83	372	13.50	152	4.45
p48	163	12.74	96	6.86	186	14.37	186	12.06	114	6.04
p49	46	8.02	45	6.84	47	10.43	54	8.43	44	6.86
p4	42	0.58	58	0.49	39	0.64	41	0.61	52	0.45
p50	163	10.04	120	4.15	172	10.91	164	9.64	109	4.17
p51	147	16.50	87	7.66	147	16.65	178	14.66	82	6.38
p52	267	14.99	149	4.41	254	14.98	254	13.02	194	5.18
p53	142	13.21	70	5.52	133	13.04	131	12.54	79	6.02
p54	440	31.28	148	4.27	353	26.35	353	26.36	143	4.68
p55	422	35.43	142	11.10	452	31.57	486	32.60	147	10.48
p56	86	55.97	73	29.50	89	63.80	92	63.69	74	30.48
p57	97	92.02	78	40.49	96	93.27	96	92.69	76	44.79
p58	138	96.69	75	34.99	137	100.28	137	99.85	81	34.53
p59	128	57.47	75	20.31	122	60.73	122	60.66	82	25.30
p5	31	0.38	54	0.37	31	0.35	31	0.37	51	0.36
p60	105	99.65	78	41.51	90	92.37	98	94.44	71	37.56
p61	104	112.54	75	42.33	107	113.08	108	111.01	72	39.87
p62	121	142.59	86	51.17	120	201.71	120	200.95	83	67.38
p63	108	114.66	74	40.33	105	131.22	108	130.56	75	44.61
p64	92	188.58	78	74.80	86	115.80	92	112.18	72	47.48
p65	106	208.67	88	87.36	110	163.13	109	158.19	76	56.41
p66	129	280.76	91	88.36	136	180.39	136	180.10	87	58.95
p67	122	88.31	80	33.89	113	144.91	123	146.54	79	51.85
p68	95	93.36	81	39.55	92	81.63	101	79.51	74	35.47
p69	92	140.88	69	52.16	97	151.15	101	148.38	71	53.32
p6	38	0.49	56	0.45	32	0.48	35	0.49	53	0.39
p70	121	207.40	80	61.46	118	204.50	118	203.79	83	66.97
p71	110	91.31	76	32.65	115	106.26	115	106.23	74	32.15
p7	36	0.47	56	0.47	34	0.48	34	0.48	56	0.43
p8	35	0.43	51	0.44	35	0.55	35	0.55	53	0.44
p9	37	0.47	55	0.43	43	0.50	40	0.49	35	0.27
Average	131	46.85	78	18.25	131	51.31	133	45.36	75	17.25

Table 3: Different Princing Strategies: Dataset 1

Inst.	plain		stab		dualcuts		dualcuts+IP		full	
	iter	time (s)	iter	time (s)	iter	time (s)	iter	time (s)	iter	time (s)
cap101	86	5.82	34	4.08	107	6.61	76	5.26	39	4.73
cap102	74	4.72	54	5.77	75	4.48	76	4.47	49	5.11
cap103	109	6.85	58	5.87	100	6.15	125	5.87	53	5.34
cap104	193	9.45	76	7.44	204	9.58	224	10.60	55	5.96
cap111	3	0.84	4	1.37	4	0.81	3	0.88	5	1.49
cap112	4	0.87	4	1.29	4	0.88	5	0.97	6	1.47
cap113	4	0.83	5	1.48	6	0.88	5	0.94	10	1.82
cap114	4	0.84	4	1.45	6	0.86	6	0.96	11	2.12
cap121	63	5.59	39	8.37	59	5.19	60	5.62	30	5.35
cap122	78	7.47	39	9.09	69	5.91	76	6.64	39	7.55
cap123	63	7.58	35	7.97	66	7.52	75	7.02	38	7.65
cap124	81	9.80	44	10.64	80	10.38	79	9.89	45	10.53
cap131	67	15.37	30	12.03	80	18.46	72	14.58	35	13.51
cap132	78	19.07	42	15.76	79	19.29	77	17.13	35	15.95
cap133	76	20.64	44	18.13	89	17.70	102	19.32	44	17.69
cap134	146	29.61	62	22.34	128	25.06	134	25.02	54	19.54
cap41	6	0.11	5	0.11	5	0.10	5	0.13	5	0.16
cap42	6	0.11	10	0.15	4	0.10	4	0.12	6	0.17
cap43	8	0.11	7	0.14	6	0.10	7	0.12	8	0.20
cap44	9	0.11	7	0.13	4	0.11	5	0.13	6	0.18
cap51	76	1.47	67	1.64	72	1.10	67	1.17	43	1.18
cap61	57	1.19	64	1.46	56	1.07	51	1.15	46	1.27
cap62	104	2.23	70	1.91	100	1.76	94	1.82	71	2.20
cap63	102	2.10	63	1.81	100	1.85	102	2.07	63	1.81
cap64	176	3.42	79	2.81	152	3.18	160	3.80	83	3.07
cap71	171	5.19	92	5.36	186	5.78	172	5.38	86	4.80
cap72	271	9.48	125	6.86	310	9.12	335	9.35	96	4.67
cap73	248	7.66	94	5.05	307	9.05	260	7.12	145	6.61
cap74	272	8.20	98	5.02	251	8.31	192	6.03	101	4.81
cap81	4	0.29	4	0.36	6	0.29	7	0.29	8	0.45
cap82	7	0.31	9	0.47	10	0.31	11	0.32	12	0.57
cap83	8	0.29	8	0.41	7	0.32	7	0.31	8	0.48
cap84	5	0.26	9	0.44	6	0.19	11	0.21	3	0.27
cap91	37	1.35	25	1.23	40	1.41	40	1.35	29	1.44
cap92	51	1.75	33	1.69	39	1.37	52	1.34	30	1.38
cap93	69	2.47	41	2.34	70	2.23	70	1.93	43	2.30
cap94	116	3.55	60	3.74	161	4.57	132	3.83	68	3.88
Average	79	5.32	42	4.76	82	5.19	81	4.95	41	4.53

Table 4: Different Princing Strategies: Dataset 2

Inst	CPX		HD		CG				HDCG		
	dual bound	primal bound	dual bound	primal bound	Dual Bnd	Primal Bnd	CG dual bnd	pricing iter	cuts	CRG dual bnd	CRG primal bnd
p10	11794.44	12280.00	12087.81	12259.00	11857.07	13062.00	12141.93	50	0	12141.93	12803.00
p11	13904.21	15131.00	14851.49	15282.00	14309.92	16005.00	14932.88	50	4	14932.88	17625.00
p12	16155.35	18372.00	17582.98	18543.00	16679.38	20474.00	17681.27	43	3	17681.27	19669.00
p13	13338.50	14345.00	13945.59	14338.00	13858.06	15200.00	14102.77	28	7	14108.50	15246.00
p14	11522.88	12330.00	12045.28	12803.00	11943.81	12732.00	12134.53	24	0	12134.53	13918.00
p15	15029.81	16768.00	16049.27	17815.00	15715.17	17504.00	16185.26	26	2	16187.46	17264.00
p16	18336.62	20743.00	19987.42	22647.00	19267.33	22523.00	20166.47	30	3	20174.05	24555.00
p17	13286.59	14345.00	13996.43	14371.00	13784.37	14968.00	14075.05	24	3	14075.10	16217.00
p18	11484.01	12296.00	12044.27	12736.00	11873.35	12313.00	12108.97	25	0	12108.97	13887.00
p19	14999.43	16339.00	16046.29	16542.00	15601.73	16863.00	16151.75	27	3	16152.90	18935.00
p1	12808.44	14071.00	13585.63	13909.00	13179.13	14390.00	13648.31	50	4	13671.39	14370.00
p20	18419.18	20709.00	19953.04	22489.00	19184.95	24340.00	20138.74	35	5	20148.65	20782.00
p21	11818.23	13320.00	11807.11	13079.00	12223.18	13546.00	11974.98	31	4	12051.76	14471.00
p22	10415.97	11249.00	10378.65	11411.00	10707.63	12386.00	10555.81	21	0	10555.81	12712.00
p23	12866.81	15121.00	12942.26	14909.00	13495.58	17469.00	13125.99	24	0	13125.99	16695.00
p24	15319.85	18090.00	15454.45	17869.00	16212.58	20471.00	15650.45	28	0	15650.45	20248.00
p25	14859.55	16399.00	15794.08	16132.00	15016.12	18337.00	15988.12	88	1	15988.12	16719.00
p26	13294.91	15113.00	14857.50	17055.00	13388.63	15810.00	14951.38	103	0	14951.38	17526.00
p27	16306.45	19735.00	19247.95	19551.00	16533.54	21196.00	19381.70	108	0	19381.70	20700.00
p28	19362.92	24113.00	23592.24	24007.00	19530.08	25340.00	23734.62	117	1	23734.62	24830.00
p29	16133.17	18015.00	17320.69	17748.00	16291.79	18530.00	17475.36	91	4	17479.47	19129.00
p2	11770.51	12578.00	12411.83	12532.00	12047.43	12710.00	12441.91	54	4	12441.91	13260.00
p30	14716.97	16737.00	16385.07	18342.00	14810.00	17423.00	16485.52	92	2	16485.52	18918.00
p31	19014.34	22493.00	21937.20	25221.00	19077.99	23093.00	22038.82	150	0	22038.82	23235.00
p32	23207.96	27946.00	27398.00	27702.00	23267.76	28715.00	27480.38	145	2	27480.38	28170.00
p33	14964.99	16822.00	15872.04	16197.00	15124.23	17449.00	16029.55	97	0	16029.55	16518.00
p34	13410.66	15334.00	14871.50	15192.00	13487.32	16699.00	14978.38	101	0	14978.38	17033.00
p35	16558.42	19735.00	19293.84	19511.00	16642.94	20684.00	19427.42	109	4	19427.42	20157.00
p36	19569.87	24027.00	23645.67	24044.00	19777.28	25407.00	23799.18	123	3	23799.18	24949.00
p37	15128.05	15409.00	15035.95	17265.00	15217.01	20287.00	15241.75	118	0	15241.75	16173.00
p38	13926.02	14028.00	13921.94	14025.00	13970.83	14623.00	13970.86	109	0	13970.86	14628.00
p39	16535.86	19412.00	17315.69	17582.00	16702.86	19283.00	17411.11	121	2	17411.11	18428.00
p3	14216.39	15813.00	15515.58	15877.00	14795.27	15998.00	15592.63	57	2	15604.89	17566.00
p40	18874.96	22802.00	20693.21	21137.00	19000.56	23842.00	20823.38	116	2	20823.38	22123.00
p41	10365.61	10619.00	10295.97	10846.00	9962.46	12319.00	10352.66	83	0	10352.66	12381.00
p42	7669.29	9778.00	8406.35	9251.00	7784.97	9805.00	8499.73	43	0	8499.73	10748.00
p43	6968.89	9375.00	7635.06	10012.00	6940.81	9961.00	7742.54	31	0	7742.54	9174.00
p44	13805.61	15364.00	15234.90	15369.00	13819.49	16177.00	15329.65	100	3	15329.65	15417.00
p45	10403.87	12497.00	10411.78	12419.00	10346.84	13446.00	10477.49	56	0	10477.49	13088.00
p46	8390.97	13180.00	8832.40	10889.00	8588.01	10495.00	8946.99	34	0	8946.99	12374.00
p47	14452.71	16216.00	15919.62	16279.00	14466.03	17931.00	15973.74	152	2	15973.74	19712.00
p48	11293.44	11608.00	11292.23	11357.00	11296.40	11932.00	11312.14	183	4	11334.47	–
p49	8767.46	9982.00	9175.71	11255.00	8945.58	10839.00	9281.31	44	0	9281.31	9935.00
p4	16700.90	18992.00	18586.56	20803.00	17534.48	19150.00	18703.56	53	6	18720.43	21903.00
p50	11021.85	11147.00	10992.15	11163.00	10901.60	12132.00	11080.71	109	0	11080.71	11787.00
p51	9279.93	9926.00	9272.82	9557.00	9024.25	10551.00	9389.46	82	0	9389.46	11491.00
p52	17419.74	17623.00	17366.22	17754.00	16081.85	18480.00	17449.82	194	3	17449.82	21719.00
p53	13902.18	15786.00	15395.88	15699.00	13981.54	16336.00	15480.58	79	2	15480.58	16430.00
p54	16252.21	16609.00	16226.18	16801.00	15099.62	16735.00	16276.73	158	1	16303.20	16493.00
p55	13016.37	14480.00	14421.71	14541.00	13037.36	16186.00	14480.00	147	0	14480.00	14480.00
p56	34617.78	40552.00	35803.26	39081.00	34965.86	45539.00	36099.21	74	1	36099.21	44800.00
p57	44697.44	54314.00	46746.46	52822.00	45124.79	57971.00	47106.54	76	1	47106.54	57803.00
p58	67129.13	76118.00	71195.81	83853.00	67798.46	109855.00	71731.05	81	0	71731.05	99059.00
p59	44523.01	56779.00	46086.02	50601.00	45090.07	57028.00	46500.98	84	4	46504.97	60370.00
p5	14867.10	15501.00	15227.55	15454.00	15099.44	16916.00	15333.46	60	9	15366.40	–
p60	30903.83	34874.00	32112.93	36038.00	31232.54	41015.00	32416.50	71	0	32416.50	38064.00
p61	37851.41	56808.00	40129.50	46495.00	38294.13	55146.00	40492.62	72	0	40492.62	52028.00
p62	53086.54	64973.00	57846.20	70683.00	53703.42	80938.00	58293.03	83	0	58293.03	82303.00
p63	38053.73	59832.00	39843.40	46038.00	38678.08	54728.00	40257.02	75	1	40257.02	52630.00
p64	29765.21	36875.00	29704.97	34487.00	29736.35	41821.00	30026.81	72	0	30026.81	40215.00
p65	34722.57	49033.00	35567.71	43735.00	35254.28	52518.00	35968.84	76	0	35968.84	50455.00
p66	46534.79	71553.00	48495.63	65685.00	47472.44	75432.00	49115.17	87	0	49115.17	74997.00
p67	37587.80	50422.00	39013.88	44395.00	38119.67	52877.00	39487.63	79	0	39487.63	47477.00
p68	30777.02	37417.00	32059.61	35979.00	31171.89	36298.00	32397.06	74	0	32397.06	38636.00
p69	37640.73	47152.00	39958.60	46591.00	38175.27	53627.00	40458.10	71	0	40458.10	50937.00
p6	13151.48	13709.00	13553.56	13732.00	13472.71	13808.00	13643.82	60	14	13658.46	15318.00
p70	52669.32	77076.00	57739.44	70319.00	53466.02	80977.00	58231.53	83	0	58231.53	87834.00
p71	37902.90	58709.00	39818.19	45448.00	38583.67	46943.00	40213.15	74	0	40213.15	53366.00
p7	16669.58	17924.00	17449.51	17828.00	17182.63	18034.00	17614.56	67	18	17628.10	18302.00
p8	20064.89	21808.00	21332.70	21908.00	20769.22	24207.00	21516.06	58	16	21537.42	25093.00
p9	12613.05	13326.00	13034.74	13254.00	12790.57	13762.00	13133.43	35	0	13133.43	14622.00

Table 5: Root node results: Dataset 1

Inst	CPX		HD		CG				HDCG		
	dual bound	primal bound	dual bound	primal bound	Dual Bnd	Primal Bnd	CG dual bnd	pricing iter	cuts	CRG dual bnd	CRG primal bnd
cap101	868268.09	868268.09	868268.09	868268.09	868268.10	868268.10	868268.09	39	4	868268.10	868268.10
cap102	896653.09	896653.09	896653.09	896653.09	896423.60	898853.10	896423.60	49	4	896423.60	911425.90
cap103	919208.43	919835.98	919381.98	919381.98	919382.00	919382.00	919381.98	53	4	919382.00	919382.00
cap104	948036.00	948124.28	948124.28	948124.28	948124.30	948124.30	948124.28	55	0	948124.30	948124.30
cap121	964898.41	991863.05	964977.98	982277.25	966081.50	1002947.00	965354.50	43	5	966112.30	1003380.00
cap122	994474.16	1046004.48	994337.76	1049647.88	998392.00	1068245.00	997894.91	66	4	998499.60	1060067.00
cap123	1022802.05	1099414.48	1021680.90	1099414.48	1027404.00	1135122.00	1027531.34	55	2	1027988.00	1135122.00
cap124	1059626.07	1117362.08	1061801.97	1102919.77	1068962.00	1214033.00	1069775.58	45	0	1069776.00	1226000.00
cap131	947238.35	947238.35	947238.35	947238.35	946719.30	967606.90	946719.32	35	0	946719.30	967606.90
cap132	965484.88	965484.88	965484.88	965484.88	965480.10	986718.90	965480.05	35	0	965480.10	986718.90
cap133	982135.89	982135.89	982135.89	982135.89	982135.90	982135.90	982135.89	44	2	982135.90	982135.90
cap134	1002623.14	1002623.14	1002623.14	1002623.14	1002623.00	1002623.00	1002623.14	54	0	1002623.00	1002623.00
cap51	1139661.27	1174972.70	1162933.98	1171287.86	1143815.00	1252437.00	1163653.07	56	7	1164136.00	1263186.00
cap61	1017415.47	1017415.47	1017415.47	1017415.47	1017113.00	1017415.00	1016734.62	66	4	1016901.00	1017415.00
cap62	1063890.93	1073499.17	1064206.68	1072680.47	1065120.00	1101143.00	1065295.88	71	1	1065296.00	1114671.00
cap63	1102137.12	1123027.49	1092787.60	1143914.40	1105435.00	1150258.00	1105757.64	63	1	1105758.00	1159120.00
cap64	1144232.78	1215745.81	1145935.50	1217282.45	1150260.00	1228657.00	1147627.61	108	1	1150330.00	1279600.00
cap71	1050285.16	1050285.16	1048606.65	1078084.06	1048738.00	1121470.00	1048737.61	86	1	1048738.00	1125788.00
cap72	1063829.94	1063829.94	1063829.94	1063829.94	1063830.00	1063830.00	1063829.94	99	4	1063830.00	1063830.00
cap73	1076463.94	1076463.94	1076463.94	1076463.94	1076464.00	1076464.00	1076463.94	145	4	1076464.00	1076464.00
cap74	1094888.34	1094888.34	1094888.34	1094888.34	1094888.00	1094888.00	1094888.34	101	4	1094888.00	1094888.00
cap91	863875.76	866015.11	863875.76	866015.11	862481.60	899442.10	862481.61	29	0	862481.60	899442.10
cap92	915676.46	932484.76	913577.25	936558.03	915620.00	988708.00	915619.99	30	2	915620.00	988708.00
cap93	960034.68	999693.85	959968.56	998616.85	962132.60	1033589.00	962154.14	43	0	962154.10	1027673.00
cap94	1005701.00	1009319.68	1006217.07	1019087.84	1007536.00	1026436.00	1007309.16	80	4	1007517.00	1017027.00

Table 6: Root node results: Dataset 2

	CPX		HD		CG		HDCG			
Inst.	gap	time (s)	gap	time (s)	gap	time (s)	gap	time (s)	gap	time (s)
p10	3.79%	1.35	1.40%	0.64	3.28%	0.94	0.96%	0.49	0.96%	0.78
p11	8.11%	1.05	1.85%	0.54	5.43%	1.05	1.33%	0.57	1.33%	0.92
p12	12.07%	1.34	4.29%	0.72	9.21%	0.97	3.91%	0.46	3.91%	0.72
p13	6.97%	2.87	2.74%	1.85	3.35%	1.70	1.67%	1.26	1.63%	2.04
p14	6.55%	2.12	2.31%	1.12	3.13%	1.47	1.61%	0.86	1.61%	1.69
p15	10.37%	2.22	4.29%	1.53	6.28%	2.38	3.60%	1.12	3.59%	2.09
p16	11.60%	2.61	3.64%	1.68	7.11%	2.83	2.86%	1.25	2.82%	2.05
p17	7.38%	3.23	2.43%	2.38	3.91%	1.62	1.92%	1.16	1.92%	1.57
p18	6.60%	1.92	2.05%	1.44	3.44%	1.75	1.54%	1.07	1.54%	1.56
p19	8.20%	2.26	1.79%	1.09	4.51%	2.17	1.16%	1.34	1.15%	1.92
p1	7.91%	0.73	2.32%	0.48	5.25%	0.89	1.91%	0.63	1.74%	0.72
p20	11.06%	3.37	3.65%	1.91	7.36%	3.01	2.83%	1.45	2.78%	2.03
p21	9.64%	2.63	9.72%	2.28	6.54%	1.68	9.22%	1.76	8.52%	2.45
p22	7.41%	2.64	7.74%	1.93	4.81%	1.73	6.57%	1.28	6.57%	1.85
p23	13.70%	3.98	13.19%	2.45	9.48%	2.10	13.58%	1.66	13.58%	2.61
p24	14.27%	2.57	13.51%	2.55	9.27%	2.90	14.18%	1.85	14.18%	2.74
p25	7.89%	4.89	2.09%	6.20	6.92%	34.69	0.90%	22.84	0.90%	26.64
p26	12.03%	8.25	1.69%	4.76	11.41%	35.22	1.08%	25.62	1.08%	28.71
p27	16.60%	6.59	1.55%	5.43	15.43%	58.41	0.87%	27.85	0.87%	32.49
p28	19.34%	10.33	1.73%	6.25	18.65%	64.48	1.15%	42.80	1.15%	47.93
p29	9.10%	5.35	2.41%	4.92	8.20%	33.90	1.56%	27.97	1.54%	36.36
p2	6.08%	1.29	0.96%	0.39	3.87%	0.87	0.72%	0.71	0.72%	0.86
p30	12.07%	6.66	2.10%	4.00	11.51%	41.01	1.53%	27.60	1.53%	34.15
p31	15.47%	6.60	2.47%	3.39	15.18%	37.74	2.06%	43.07	2.06%	45.50
p32	16.22%	7.84	1.10%	3.99	16.01%	39.15	0.81%	32.50	0.81%	34.70
p33	7.61%	4.81	2.01%	5.64	6.62%	50.09	1.04%	20.84	1.04%	23.83
p34	11.73%	8.47	2.11%	6.72	11.22%	33.26	1.43%	20.53	1.43%	23.26
p35	15.13%	7.60	1.11%	5.63	14.70%	35.80	0.43%	23.14	0.43%	29.85
p36	18.55%	7.33	1.59%	5.56	17.69%	56.58	0.96%	34.31	0.96%	40.83
p37	1.82%	5.59	2.42%	7.58	1.25%	46.95	1.10%	35.92	1.10%	41.75
p38	0.71%	3.66	0.73%	5.68	0.39%	35.00	0.39%	25.39	0.39%	29.48
p39	5.95%	3.02	1.51%	6.37	5.00%	40.35	0.98%	30.87	0.98%	35.93
p3	10.10%	2.26	1.88%	0.46	6.44%	1.00	1.41%	0.53	1.33%	0.68
p40	10.70%	7.05	2.10%	6.63	10.11%	55.03	1.51%	27.64	1.51%	33.22
p41	2.39%	1.38	3.04%	1.37	6.18%	5.50	2.57%	2.75	2.57%	3.27
p42	17.10%	3.16	9.13%	2.29	15.85%	4.69	8.84%	3.61	8.84%	4.50
p43	24.04%	5.14	16.78%	6.32	24.34%	6.11	18.49%	4.16	18.49%	5.83
p44	10.14%	1.31	0.84%	0.92	10.05%	6.01	0.22%	2.46	0.22%	2.71
p45	16.23%	2.52	16.16%	3.31	16.69%	6.75	18.53%	4.71	18.53%	5.80
p46	20.05%	9.18	15.84%	6.09	18.17%	6.57	17.30%	6.04	17.30%	8.37
p47	10.87%	0.69	1.83%	0.57	10.79%	22.86	1.52%	4.58	1.52%	5.62
p48	0.56%	1.26	0.57%	1.63	0.53%	8.62	0.40%	8.49	0.20%	10.45
p49	11.75%	4.73	7.64%	4.89	9.96%	7.18	7.04%	7.22	7.04%	8.90
p4	12.06%	1.38	2.13%	0.55	7.67%	1.04	1.54%	0.54	1.45%	0.71
p50	1.12%	1.16	1.39%	1.28	2.20%	8.88	0.60%	4.26	0.60%	4.87
p51	2.90%	2.33	2.97%	2.48	5.57%	11.72	1.78%	6.83	1.78%	7.93
p52	1.15%	0.96	1.46%	0.69	8.75%	12.46	0.99%	5.36	0.99%	6.30
p53	11.45%	3.39	1.93%	3.07	10.94%	10.71	1.41%	6.43	1.41%	8.40
p54	1.46%	0.80	1.62%	0.44	8.45%	15.20	1.33%	5.00	1.16%	5.53
p55	10.11%	2.68	0.40%	1.65	9.96%	17.08	0.00%	10.57	0.00%	10.52
p56	11.42%	19.88	8.39%	8.46	10.53%	53.34	8.26%	31.31	8.26%	39.02
p57	15.38%	22.09	11.50%	16.03	14.57%	54.69	12.13%	45.88	12.13%	65.28
p58	11.81%	18.80	6.47%	15.36	10.93%	68.39	6.12%	35.02	6.12%	48.15
p59	12.01%	17.03	8.92%	12.62	10.89%	39.10	8.82%	26.70	8.81%	34.80
p5	3.80%	0.79	1.47%	0.48	2.29%	1.02	0.79%	0.54	0.57%	0.66
p60	11.38%	19.54	7.92%	13.62	10.44%	51.51	7.58%	38.13	7.58%	43.80
p61	18.59%	23.75	13.69%	15.58	17.64%	49.45	14.82%	40.40	14.82%	47.92
p62	18.29%	17.94	10.97%	14.87	17.35%	73.41	11.46%	67.98	11.46%	83.94
p63	17.34%	21.96	13.46%	14.23	15.99%	52.24	14.36%	45.66	14.36%	59.08
p64	13.69%	17.88	13.87%	15.04	13.78%	74.06	14.85%	48.38	14.85%	56.99
p65	20.61%	26.30	18.67%	20.54	19.39%	81.64	21.59%	57.06	21.59%	64.38
p66	29.15%	28.86	26.17%	28.05	27.73%	71.39	33.74%	59.63	33.74%	71.88
p67	15.33%	20.54	12.12%	14.70	14.14%	45.20	12.43%	52.63	12.43%	61.76
p68	14.46%	25.57	10.89%	11.24	13.36%	45.30	11.06%	36.02	11.06%	40.74
p69	19.21%	26.33	14.24%	11.97	18.06%	60.84	15.16%	53.90	15.16%	67.38
p6	4.07%	1.24	1.13%	0.46	1.72%	0.85	0.48%	0.62	0.37%	0.80
p70	25.10%	34.50	17.89%	16.07	23.97%	69.77	20.76%	67.56	20.76%	79.49
p71	16.60%	19.43	12.39%	11.85	15.10%	42.86	13.02%	32.73	13.02%	38.94
p7	6.50%	1.50	2.12%	0.33	3.62%	1.05	1.21%	0.76	1.08%	0.85
p8	7.99%	1.50	2.18%	0.79	4.76%	1.31	1.36%	0.64	1.26%	0.91
p9	4.84%	0.71	1.65%	0.44	3.50%	0.72	0.92%	0.29	0.92%	0.40
Avg.	11.18%	7.78	5.75%	5.56	9.98%	25.61	5.67%	18.50	5.64%	22.26

Table 7: Computational time and gap at root node: Dataset 1

	CPX		HD		CG		HDCG			
Inst.	gap	time (s)	gap	time (s)	gap	time (s)	gap	time (s)	gap	time (s)
cap101	0.00%	0.21	0.00%	1.20	0.00%	2.70	0.00%	7.53	0.00%	6.87
cap102	0.00%	0.38	0.00%	1.45	0.03%	3.52	0.03%	9.04	0.03%	10.10
cap103	0.02%	0.36	0.00%	1.60	0.00%	3.18	0.00%	8.02	0.00%	8.02
cap104	0.01%	0.42	0.00%	1.45	0.00%	3.91	0.00%	6.40	0.00%	5.88
cap121	1.77%	4.26	1.76%	4.67	1.65%	10.05	1.75%	11.20	1.67%	14.08
cap122	4.93%	8.47	4.94%	4.40	4.55%	9.54	4.82%	14.41	4.76%	17.40
cap123	6.97%	4.72	7.07%	5.95	6.55%	9.81	7.00%	11.85	6.95%	15.64
cap124	3.93%	5.62	3.73%	6.47	3.08%	11.74	3.10%	12.97	3.10%	15.71
cap131	0.00%	1.02	0.00%	3.98	0.05%	7.12	0.05%	18.20	0.05%	22.10
cap132	0.00%	1.20	0.00%	4.00	0.00%	7.09	0.00%	22.22	0.00%	26.41
cap133	0.00%	1.44	0.00%	4.60	0.00%	6.46	0.00%	24.15	0.00%	24.58
cap134	0.00%	1.20	0.00%	4.38	0.00%	8.35	0.00%	21.43	0.00%	21.60
cap51	2.70%	0.82	0.71%	0.85	2.35%	1.16	0.66%	1.47	0.61%	1.72
cap61	0.00%	0.23	0.00%	0.31	0.03%	1.64	0.07%	1.75	0.05%	1.56
cap62	0.82%	0.48	0.79%	0.63	0.70%	2.10	0.69%	2.43	0.69%	2.59
cap63	1.86%	0.90	2.69%	0.71	1.57%	2.20	1.56%	2.14	1.56%	2.55
cap64	5.88%	0.61	5.74%	0.71	5.39%	2.71	5.94%	3.61	5.69%	3.95
cap71	0.00%	0.27	0.16%	1.36	0.15%	2.78	0.15%	5.80	0.15%	7.06
cap72	0.00%	0.21	0.00%	0.92	0.00%	3.10	0.00%	6.47	0.00%	6.86
cap73	0.00%	0.21	0.00%	0.86	0.00%	2.98	0.00%	8.86	0.00%	8.98
cap74	0.00%	0.31	0.00%	0.92	0.00%	2.79	0.00%	7.25	0.00%	6.81
cap91	0.25%	0.71	0.25%	0.82	0.41%	2.15	0.41%	1.97	0.41%	2.45
cap92	1.80%	0.86	2.03%	1.03	1.81%	2.40	1.84%	2.23	1.84%	2.65
cap93	3.86%	1.60	3.87%	1.57	3.65%	2.71	3.79%	2.88	3.79%	3.45
cap94	0.36%	0.97	0.31%	1.58	0.18%	3.22	0.20%	5.29	0.18%	6.84
Avg.	1.41%	1.50	1.36%	2.26	1.29%	4.62	1.28%	8.78	1.26%	9.83

Table 8: Computational time and gap at root node: Dataset 2

Inst.	CPX		HD		DBP	
	time (s)	gap $\frac{PB-DB}{DB}$	time(s)	gap $\frac{PB-DB}{DB}$	time (s)	gap $\frac{PB-DB}{DB}$
p10	11.9	0.0%	0.92	0.0%	1.74	0%
p11	54.22	0.0%	1.86	0.0%	1.75	0%
p12	743.18	0.0%	5.71	0.0%	1.58	0%
p13	777.4	0.0%	11.36	0.0%	3.62	0%
p14	259.97	0.0%	5.33	0.0%	2.47	0%
p15	-	0.6%	10.58	0.0%	2.79	0%
p16	-	5.4%	21.31	0.0%	5.45	0%
p17	1117.26	0.0%	4.76	0.0%	4.51	0%
p18	215.55	0.0%	3.42	0.0%	1.8	0%
p19	-	0.9%	11.93	0.0%	3.05	0%
p1	166.92	0.0%	1.22	0.0%	1.26	0%
p20	-	4.9%	19.47	0.0%	4.34	0%
p21	4099.17	0.0%	6120.44	0.0%	2706.98 (MEM)	2.9%
p22	735.41	0.0%	1759.25	0.0%	1440.5	0%
p23	-	4.4%	-	5.8%	2929.48 (MEM)	5.4%
p24	-	8.5%	-	8.5%	1908.54 (MEM)	9.0%
p25	-	3.0%	12.87	0.0%	87.69	0%
p26	-	10.6%	35.66	0.0%	89.08	0%
p27	-	16.3%	28.9	0.0%	118.48	0%
p28	-	19.7%	17.69	0.0%	161.3	0%
p29	57.04	0.0%	31.37	0.0%	127	0%
p2	238.54	0.0%	0.55	0.0%	1.71	0%
p30	39.18	0.0%	33.81	0.0%	74.61	0%
p31	39.65	0.0%	25.33	0.0%	144.53	0%
p32	36.72	0.0%	12.11	0.0%	335.86	0%
p33	-	2.0%	23.54	0.0%	148.7	0%
p34	-	8.7%	28.76	0.0%	70.14	0%
p35	-	13.9%	37.88	0.0%	214.31	0%
p36	-	18.8%	32.32	0.0%	326.46	0%
p37	10.01	0.0%	25.71	0.0%	203.5	0%
p38	10.07	0.0%	7.93	0.0%	97.31	0%
p39	31.45	0.0%	16.6	0.0%	141.94	0%
p3	5739.89	0.0%	1.43	0.0%	1.81	0%
p40	48.38	0.0%	10.88	0.0%	126.02	0%
p41	8.99	0.0%	5.43	0.0%	29.05	0%
p42	39.44	0.0%	46.8	0.0%	1133.94	0%
p43	111.02	0.0%	79.77	0.0%	20.68	0%
p44	3.81	0.0%	1.54	0.0%	12.88	0%
p45	6.84	0.0%	7.84	0.0%	855.68	0%
p46	101.62	0.0%	35.2	0.0%	898.93	0%
p47	280.79	0.0%	5.47	0.0%	516.52	0%
p48	3.16	0.0%	2.19	0.0%	35.27	0%
p49	129.1	0.0%	138.23	0.0%	23.17	0%
p4	-	3.7%	4.61	0.0%	3.89	0%
p50	2.89	0.0%	2.33	0.0%	23	0%
p51	23.91	0.0%	21.42	0.0%	68.33	0%
p52	6.57	0.0%	4.72	0.0%	114.81	0%
p53	-	2.4%	12.43	0.0%	33.87	0%
p54	9.86	0.0%	1.25	0.0%	57.79	0%
p55	2123.65	0.0%	2.18	0.0%	37.06	0%
p56	-	4.1%	-	0.3%	726.02	0%
p57	-	5.4%	-	0.4%	3888.75	0%
p58	-	7.0%	-	0.8%	-	2.2%
p59	-	2.6%	1256.37	0.0%	694.63	0%
p5	31.03	0.0%	0.79	0.0%	1.19	0%
p60	-	2.3%	5863.28	0.0%	232.32	0%
p61	-	5.9%	-	0.1%	624.02	0%
p62	-	10.3%	6453.77	0.0%	-	1.1%
p63	3909.3	0.0%	825.19	0.0%	4995.94	0%
p64	1250.55	0.0%	661.88	0.0%	211.03	0%
p65	2949.67	0.0%	4366.29	0.0%	350.41	0%
p66	-	0.7%	6697.26	0.0%	3096.4	0%
p67	1271.62	0.0%	1160.84	0.0%	1267.44	0%
p68	-	3.2%	3659.98	0.0%	264.71	0%
p69	-	7.0%	4318.73	0.0%	2148.87	0%
p6	20.18	0.0%	0.81	0.0%	0.95	0%
p70	-	10.7%	-	0.3%	-	1.2%
p71	-	0.1%	634.63	0.0%	328.73	0%
p7	479.77	0.0%	1.67	0.0%	1.6	0%
p8	2447.69	0.0%	6.38	0.0%	155.53	0%
p9	1.6	0.0%	0.61	0.0%	0.74	0%
Av./Succ	608.23	43	222.34	64	322.03	65

Table 9: Solving TLHCFL to proven optimality: Dataset 1

Inst.	CPX		HD		DBP	
	time (s)	gap $\frac{PB-DB}{DB}$	time(s)	gap $\frac{PB-DB}{DB}$	time (s)	gap $\frac{PB-DB}{DB}$
cap101	0.23	0.0%	1.17	0.0%	4.02	0%
cap102	0.38	0.0%	1.44	0.0%	6.73	0%
cap103	0.37	0.0%	1.61	0.0%	5.62	0%
cap104	0.42	0.0%	1.45	0.0%	9.95	0%
cap111	0.12	0.0%	0.18	0.0%	1.18	0%
cap112	0.14	0.0%	0.18	0.0%	1.23	0%
cap113	0.13	0.0%	0.18	0.0%	2.76	0%
cap114	0.13	0.0%	0.2	0.0%	2.86	0%
cap121	12.78	0.0%	13.14	0.0%	748.16	0%
cap122	900.67	0.0%	802.52	0.0%	4403.4 (MEM)	1.4%
cap123	185.11	0.0%	437.21	0.0%	-	1.1%
cap124	26.8	0.0%	19.92	0.0%	-	0.7%
cap131	0.95	0.0%	3.97	0.0%	23.7	0%
cap132	1.16	0.0%	4.02	0.0%	35.58	0%
cap133	1.42	0.0%	4.61	0.0%	17.76	0%
cap134	1.14	0.0%	4.39	0.0%	18.85	0%
cap41	0.02	0.0%	0.02	0.0%	0.15	0%
cap42	0.02	0.0%	0.03	0.0%	0.2	0%
cap43	0.02	0.0%	0.03	0.0%	0.19	0%
cap44	0.02	0.0%	0.04	0.0%	0.2	0%
cap51	21.62	0.0%	1.01	0.0%	4.85	0%
cap61	0.22	0.0%	0.31	0.0%	2.08	0%
cap62	0.58	0.0%	0.73	0.0%	4.34	0%
cap63	2.01	0.0%	2.98	0.0%	62.71	0%
cap64	2.22	0.0%	1.42	0.0%	18.25	0%
cap71	0.25	0.0%	1.37	0.0%	9.7	0%
cap72	0.21	0.0%	0.91	0.0%	10.49	0%
cap73	0.2	0.0%	0.87	0.0%	7.27	0%
cap74	0.3	0.0%	0.92	0.0%	6.75	0%
cap81	0.04	0.0%	0.06	0.0%	0.27	0%
cap82	0.04	0.0%	0.05	0.0%	0.35	0%
cap83	0.05	0.0%	0.05	0.0%	0.77	0%
cap84	0.04	0.0%	0.06	0.0%	0.41	0%
cap91	0.71	0.0%	0.83	0.0%	2.25	0%
cap92	1.69	0.0%	2.18	0.0%	4.74	0%
cap93	2.37	0.0%	2.15	0.0%	8.33	0%
cap94	1.24	0.0%	2.18	0.0%	14.7	0%
Av./Succ.	1.57	37	1.61	37	30.51	34

Table 10: Solving TLHCFL to proven optimality: Dataset 2