

A computational evaluation of a general branch-and-price framework for capacitated network location problems

Alberto Ceselli, Federico Liberatore, Giovanni Righini *
Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano
via Bramante 65, 26013 Crema, Italy

April 1, 2005

Abstract

The purpose of this paper is to illustrate a general framework for network location problems, based on column generation and branch-and-price. In particular we consider capacitated network location problems with single-source constraints. We consider several different network location models, by combining cardinality constraints, fixed set-up costs, concentrator restrictions and regional constraints. Our general branch-and-price-based approach can be seen as a natural counterpart of the branch-and-cut-based commercial ILP solvers, with the advantage of exploiting the tightness of the lower bound provided by the set partitioning reformulation of network location problems. Branch-and-price and branch-and-cut are compared through an extensive set of experimental tests.

Keywords: network location, branch-and-price, branch-and-cut, integer programming

* Correspondence to: righini@dti.unimi.it

1 Introduction

Finding the optimal location for facilities like warehouses or servers in distribution or telecommunication networks and deciding how to allocate clients to them is a very complex task, since it requires to solve *NP*-Hard integer linear programming problems. Classical and well-studied examples are the Capacitated Facility Location Problem and the P-Median Problem [23]. When tackling an *NP*-hard integer linear programming problem one has the choice between using a general-purpose solver and designing an “ad hoc” algorithm. The most common general-purpose solvers for integer linear programming problems, like ILOG CPLEX and others, use branch-and-cut methods. They combine powerful linear programming solvers with suitable subroutines able to detect violated inequalities and branching strategies to develop a search tree.

The purpose of this paper is to illustrate a general framework for network location problems, which is based on column generation and branch-and-price. In particular we concentrate on capacitated network location problems with single-source constraints. We consider several different network location models, by combining cardinality constraints, fixed set-up costs, concentrator restrictions and regional constraints and we illustrate how all of them can be solved in almost the same way by a generalization of a branch-and-price algorithm we recently designed for the Capacitated P-Median Problem [8].

It is clear that for each particular problem we consider it should be possible to develop more effective “ad hoc” algorithms, both following the branch-and-cut approach and the branch-and-price one (and possibly others, such as Lagrangean relaxation and branch-and-bound): however the accent here is put on generality and flexibility. Our goal in this study was to develop a kind of branch-and-price equivalent of the branch-and-cut-based commercial ILP solvers. Apart from being available for free for research purposes, our branch-and-price framework has the advantage of exploiting the tightness of the lower bound provided by the set partitioning reformulation of network location problems.

The outline of the paper is as follows. In Section 2 we report the models of the single-source capacitated location problems we consider and we review the relevant literature and the solution methods proposed so far. In Section 3 we describe our branch-and-price framework. In Section 4 we review the main algorithmic features of a general-purpose ILP solver we used as a benchmark. In Section 5 we present our experimental results. Conclusions are outlined in Section 6.

2 Single-source capacitated location problems

In a basic facility location scenario the best trade-off has to be found between the cost for building facilities at certain sites and the cost to serve customers. Here we consider min-sum objective functions, where service costs are assumed to be proportional to the distance between each customer and the facility to which it is assigned. The cost of building facilities can be taken into account into two different ways: either with an additional term in the objective function or with a constraint (or both). In the former case fixed set-up costs are specified for each candidate site; these costs are to be paid when a facility is set-up in that site. In the latter case a cardinality constraint is added to the model, so that the number of available facilities is bounded from above. In this section we briefly review the mathematical formulations of network facility location problems with fixed

set-up costs and cardinality constraints. Then we also outline the formulation of the same problems when regional constraints are added. All problems we consider have capacitated facilities and single-source constraints, so that it is not allowed to split customers' demands on more than one facility.

The following definitions apply to all models considered in the remainder. We are given a set \mathcal{N} of customers and a set \mathcal{M} of candidate sites where facilities can be located. An integer weight w_i represents the demand of each customer $i \in \mathcal{N}$. The capacity of a facility built in each site $j \in \mathcal{M}$ is represented by an integer Q_j . Integer coefficients d_{ij} (usually referred to as *distances*) describe the cost of allocating customer $i \in \mathcal{N}$ to a facility located in site $j \in \mathcal{M}$. We make the assumption that $d_{ij} \geq 0 \forall i \in \mathcal{N}, j \in \mathcal{M}$.

2.1 Fixed set-up costs

The set-up cost for a facility in each site $j \in \mathcal{M}$ is represented by an integer coefficient f_j . The problem in which fixed set-up costs are incurred is known as Capacitated Facility Location Problem with Single Source constraints (SS-CFLP).

A formulation for the SS-CFLP is the following:

$$SS - CFLP) \quad \min \quad v = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} + \sum_{j \in \mathcal{M}} f_j y_j \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{M}} x_{ij} = 1 \quad \forall i \in \mathcal{N} \quad (2)$$

$$\sum_{i \in \mathcal{N}} w_i x_{ij} \leq Q_j y_j \quad \forall j \in \mathcal{M} \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \quad (4)$$

$$y_j \in \{0, 1\} \quad \forall j \in \mathcal{M} \quad (5)$$

Binary variables x are assignment variables: $x_{ij} = 1$ if and only if customer i is served by a facility located in site j . Binary variables y correspond to location decisions: $y_j = 1$ if and only if site j is selected to host a facility. The objective is to minimize the sum of allocation costs depending on x variables and set-up costs depending on y variables. Set partitioning constraints (2) impose that each customer is assigned to a facility. Capacity constraints (3) impose that the sum of the customers' demands assigned to a same facility does not exceed the capacity of the facility; these constraints also forbid the assignment of customers to sites which do not host facilities.

To strengthen the linear relaxation of this formulation, disaggregated inequalities

$$x_{ij} \leq y_j, \quad \forall i \in \mathcal{N}, j \in \mathcal{M} \quad (6)$$

are introduced. They arise from constraints (3), and the aggregated demand constraint

$$\sum_{j \in \mathcal{M}} Q_j y_j \geq \sum_{i \in \mathcal{N}} w_i. \quad (7)$$

When single-source constraints are relaxed, that is x variables have a continuous domain in the range $[0, 1]$, the Capacitated Facility Location Problem

(CFLP) arises. The CFLP has been extensively studied and the literature on it is quite rich. Erlenkotter [38] proposed an algorithm in which the continuous relaxation of (1) - (6) is used as a dual bound. Effective optimization algorithms for the CFLP have been developed through Lagrangean methods [10] and cross decomposition [39]. These algorithms allow to solve problem instances involving up to 50 customers and 50 candidate sites. More recent approaches are due to Aardal [1], who exploits the polyhedral structure of the problem to design an effective branch-and-cut algorithm, and to Klose et al. [24], who apply Dantzig-Wolfe decomposition and column generation to obtain better bounds. Aardal [1] attacked instances with up to 100 customers and 75 facilities, proving the optimality of the solutions provided; Klose et al. [24] obtained good approximations on problems with up to 500 customers and 200 candidate facilities, and solved to optimality problem instances with up to 200 customers embedding their column generation routine in a branch-and-price algorithm. Daskin [12] and Drexler [23] give detailed surveys on the CFLP.

The methods proposed in the literature for the single-source constraints could solve only smaller instances. Also in this case, Lagrangean relaxation can be used to design branch-and-bound methods. Dualization of the capacity constraints (3) is discussed in Klincewitz et al. [22], where the authors solve problem instances with 50 customers in few minutes. Pirkul [36] obtained better results by dualizing the partitioning constraints (2): he could solve instances with 100 customers and 20 candidate facilities. Holmberg et al. [20] proposed to couple a Lagrangean relaxation of the capacity constraints with a repeated matching heuristic, to solve problem instances involving up to 200 customers and 30 candidate facilities in some minutes. A column generation approach for the SS-CFLP was proposed by Neebe and Rao [34], who solved problem instances with more than 35 customers and 25 facilities. More recently Fernandez and Diaz [14] implemented a new branch-and-price method, which solved to optimality instances with up to 90 customers and 30 facilities in few hours.

2.2 Concentrators

When the set of customers and the set of candidate sites coincide, the resulting problem is called Capacitated Concentrator Location Problem (CCLP) [17]. This variant has received much attention in telecommunication networks design, where facilities represent electronic devices and customers represent terminals on a telecommunication network. In the CCLP each location variable y_j can be replaced by a corresponding assignment variable x_{jj} . Although some authors [36] used the name ‘‘Capacitated Concentrators’’ to indicate the SS-CFLP, we follow [27] and [26] and we indicate as ‘‘concentrator problems’’ the models in which variables x_{jj} replace variables y_j to represent location decisions.

The mathematical formulation of the CCLP is as follows:

$$\text{CCLP) } \min v = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} + \sum_{j \in \mathcal{M}} f_j x_{jj}$$

$$\text{s.t. (2), (4)}$$

$$\sum_{i \in \mathcal{N}} w_i x_{ij} \leq Q_j x_{jj} \quad \forall j \in \mathcal{M} \quad (8)$$

$$x_{ij} \leq x_{jj} \quad \forall i \in \mathcal{N}, j \in \mathcal{M} \quad (9)$$

$$\sum_{j \in \mathcal{M}} c_j x_{jj} \geq \sum_{i \in \mathcal{N}} w_i \quad (10)$$

The polyhedral structure of these problems has recently been studied in detail by Labbé and Yaman [27] [26]. Problems on networks with up to 100 terminals can be solved to optimality with a branch-and-cut approach in half an hour of CPU time.

2.3 Cardinality constraints

When the set-up costs are taken into account through a limit on the number of facilities that can be built, the resulting problems are usually called “median problems”. The most basic problem of this type is the p -Median Problem (PMP), which is the discrete counterpart of the famous multi-source Fermat-Weber problem [37]. The PMP consists in partitioning the vertices of a given graph into p subsets and to choose a *median* vertex in each subset, minimizing the sum of the distances between each vertex of the graph and the median of its subset. The PMP arises in many different contexts such as network design, telecommunications, distributed database design, transportation and distribution logistics. Kariv and Hakimi [21] proved that the PMP is \mathcal{NP} -hard. Optimization algorithms based on Lagrangean relaxation have been proposed by Narula et al. [33], Cornuéjols et al. [11], Christofides and Beasley [9] and Beasley [4]; approaches based on dual formulations are illustrated in Galvao [15] and Hanjoul and Peeters [18]. Among the most recent contributions we cite the branch-and-cut-and-price algorithm by Avella et al. [2], the branch-and-price algorithm by Senne et al. [41] and the semi-Lagrangean relaxation algorithm by Beltran et al. [5]. The most successful approach is that of Avella et al.: an instance on a graph with 3795 vertices, in which 150 facilities have to be selected was solved, even though this optimization took several hours of CPU time. A survey on the PMP can be found in Labbé et al. [25].

Here we consider the capacitated version of the problem, that is the Capacitated PMP (CPMP), that can be formulated as follows.

$$\begin{aligned}
 \text{CPMP) } \min \quad & v = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} \\
 \text{s.t. } \quad & (2), (3), (4), (5), (6) \\
 & \sum_{j \in \mathcal{M}} y_j = p,
 \end{aligned}$$

Algorithms devised for the uncapacitated PMP cannot be adapted to the CPMP in a straightforward way: even finding a feasible solution is \mathcal{NP} -complete when capacities are considered. Recent contributions to the literature on the CPMP include the algorithm of Baldacci et al. [3], which can give “a posteriori” guarantee on optimality or the approximation achieved and a branch-and-price algorithm of Ceselli and Righini [8]. The branch-and-price approach solved problems on graphs with up to 200 vertices and any number of medians in less than two hours of computation.

Very recently, two heuristic algorithms for the CPMP have been presented: Lorena and Senne [28] followed a column generation approach, finding good solutions on real instances with up to 402 vertices, while Diaz and Fernández [13] attacked an instance with 737 vertices through hybrid scatter search and path relinking.

Both Baldacci et al. [3] and Lorena and Senne [28] replaced the location variables y_j with the assignment variables x_{jj} , so that each median is forced

to be coincident with one of the vertices of its cluster. To be consistent with the terminology explained above, we indicate this problem as the Capacitated P-Concentrator Location Problem (CPCLP).

$$\begin{aligned}
\text{CPCLP)} \quad \min \quad & v = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} \\
\text{s.t.} \quad & (2), (8), (4), (9), (10) \\
& \sum_{j \in \mathcal{M}} x_{jj} = p,
\end{aligned}$$

As reported by Ceselli [7] this restriction can change the optimal solution or even inhibit the existence of feasible solutions.

2.4 Regional Constraints

A region is defined as a subset of candidate sites; a regional constraint imposes an upper or lower bound to the number of facilities that can be built in a certain region. Regional constraints are usually employed to enforce equity in the geographical distribution of the facilities. Following Syam [42] and Murray and Gerrard [32] we formulate the SS-CFLP with regional constraints (RCSS-CFLP) as follows:

$$\begin{aligned}
\text{RCSS - CFLP)} \quad \min \quad & v = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} + \sum_{j \in \mathcal{M}} f_j y_j \\
\text{s.t.} \quad & (2), (3), (4), (5), (6), (7) \\
& \sum_{j \in R} y_j \leq u_R \quad \forall R \in \mathcal{R} \quad (11) \\
& \sum_{j \in R} y_j \geq l_R \quad \forall R \in \mathcal{R} \quad (12)
\end{aligned}$$

Each $R \in \mathcal{R}$ represents a region, that is a subset of the candidate sites. Constraints (11) and (12) impose respectively an upper and a lower bound on the number of facilities that can be located in each region. It is worth noting that in general regions may overlap, while the methods proposed in Syam [42] and Murray and Gerrard [32] are restricted to the case of non-overlapping regions.

Regional constraints can be also added to all the other location problems listed above. The CPMP itself can be viewed as a RCSS-CFLP in which there are no set-up costs ($f_j = 0 \quad \forall j \in \mathcal{M}$) and just one region R with $u_R = l_R = p$.

2.5 Set partitioning formulation

All location problems described in the previous section admit the following alternative formulation.

$$MP) \min \sum_{j \in \mathcal{M}} \sum_{k \in Z^j} (f_j + \sum_{i \in \mathcal{N}} d_{ij} x_i^k) z_k^j \quad (13)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{M}} \sum_{k \in Z^j} x_i^k z_k^j = 1 \quad \forall i \in \mathcal{N} \quad (14)$$

$$- \sum_{k \in Z^j} z_k^j \geq -1 \quad \forall j \in \mathcal{M} \quad (15)$$

$$- \sum_{j \in R} \sum_{k \in Z^j} z_k^j \geq -u_R \quad \forall R \in \mathcal{R} \quad (16)$$

$$\sum_{j \in R} \sum_{k \in Z^j} z_k^j \geq l_R \quad \forall R \in \mathcal{R} \quad (17)$$

$$z_k^j \in \{0, 1\} \quad \forall j \in \mathcal{M}, \forall k \in Z^j. \quad (18)$$

In this set partitioning problem each column corresponds to a feasible cluster, that is an assignment of customers to a facility, that satisfies the capacity constraint. Each cluster k is described by assignment coefficients x_i^k equal to 1 if and only if customer $i \in \mathcal{N}$ belongs to cluster $k \in Z^j$ of the facility located in site $j \in \mathcal{M}$. A binary variable z_k^j is associated with each cluster. Constraints (14) guarantee that each customer is assigned to a facility; constraints (15) impose that no more than one cluster is associated to the same facility site. For each region $R \in \mathcal{R}$, constraints (16) and (17) impose that the number of clusters with facilities in R is between a lower bound l_R and an upper bound u_R .

For concentrator-like models, each cluster always contains the corresponding facility. In this case, constraints (15) are redundant, because they are implied by constraints (14), and they can be removed. When this condition is not enforced, partitioning constraints (14) can be replaced by covering constraints:

$$\sum_{j \in \mathcal{M}} \sum_{k \in Z^j} x_i^k z_k^j = 1 \quad \forall i \in \mathcal{N} \quad (19)$$

because all distances are non-negative and therefore it does always exist an optimal solution in which no customer is assigned more than once.

The set partitioning reformulation is the starting point for developing column generation and branch-and-price algorithms. Many authors have followed this path to develop effective algorithms to solve the SS-CFLP (Diaz and Fernandez [14]) the PMP (Senne et al. [41]) the CPMP (Baldacci et al. [3], Ceselli and Righini [8]) or similar problems (Savelsbergh [40]).

Our purpose however is not to develop an algorithm tailored to any of such problems, but rather to exploit the generality of the approach. This approach works for all problems outlined above and this the reason for which we found worth developing and evaluating a general-purpose branch-and-price solver for network location problems.

3 A branch-and-price algorithm

In this section we present a general branch-and-price framework for single-source capacitated location problems. This framework derives from the algorithm proposed in [8] for the exact optimization of the CPMP.

We describe the main components of the algorithm. These include the column generation subroutine, the branching strategy and the policy for the management of columns. We also discuss the use of Lagrangean lower bounds and primal heuristics. In the exposition of this part we refer to formulation (13) (15) (16) (17) (19) (18), including cardinality constraints, fixed set-up costs and regional constraints. The cardinality constraint has not been taken into account in an explicit way, since it is a special case of regional constraints.

3.1 Column generation

Each set Z^j of feasible clusters served from a facility located in site j contains an exponential number of elements. Since the linear relaxation of the master problem MP (indicated hereafter by LMP) cannot be solved directly because of the exponential number of its columns, column generation is applied (see Gilmore and Gomory [16]): a restricted linear master problem (RLMP), defined by a relatively small subset of columns, is considered and solved to optimality; then, a search is performed for new columns of negative reduced cost and, if any such column is found, it is inserted into the formulation and the RLMP is solved again. When no columns of negative reduced cost exist, the optimal solution of the RLMP is also optimal for the LMP and its value is a valid lower bound to be used in a branch-and-bound framework.

The main advantage of the branch-and-price approach consists in the tightness of the lower bound. The set partitioning formulation can be obtained from the compact formulation by applying Dantzig-Wolfe decomposition [31]. As far as the linear relaxations of the two formulations are concerned, the polyhedron Ω^j described by the capacity constraints in the compact formulation are convexified in the set partitioning formulation. Since each Ω^j is the polyhedron of the linear relaxation of a binary knapsack problem (see [30] for a classical reference), which is known not to have the integrality property, its extreme points can have fractional coordinates; therefore the lower bound computed after the convexification of each set Ω^j can be tighter than that provided by the linear relaxation of the compact formulation. Many experiments [7] [8] show that this is actually the case (see also Section 5).

The implementation of the column generation algorithm requires the repeated solution of a pricing problem. Let $\lambda \in \mathbb{R}_+^N$, $\mu \in \mathbb{R}_+^M$, $\gamma^u \in \mathbb{R}_+^{|\mathcal{R}|}$ and $\gamma^l \in \mathbb{R}_+^{|\mathcal{R}|}$ be the vectors of non-negative dual variables corresponding to constraints (14), (15), (16) and (17) respectively; the reduced cost of column $k \in Z^j$ is

$$r^k(\lambda, \mu, \gamma^u, \gamma^l) = f_j + \sum_{i \in \mathcal{N}} d_{ij} x_i^k - \sum_{i \in \mathcal{N}} \lambda_i x_i^k + \mu_j - \sum_{R \in \mathcal{R} | j \in R} (\gamma_R^l - \gamma_R^u)$$

To find columns with negative reduced cost, one must solve a pricing problem for each candidate facility $j \in \mathcal{M}$:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{N}} (d_{ij} - \lambda_i) x_i^k + f_j + \mu_j - \sum_{R \in \mathcal{R} | j \in R} (\gamma_R^l - \gamma_R^u) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{N}} w_i x_i^k \leq Q_j \\ & x_i^k \in \{0, 1\} \quad \forall i \in \mathcal{N} \end{aligned}$$

and this requires the solution of the following binary knapsack problem:

$$\begin{aligned}
 KP_j) \quad \max \quad & \tau_j = \sum_{i \in \mathcal{N}} (\lambda_i - d_{ij}) x_i^k \\
 \text{s.t.} \quad & \sum_{i \in \mathcal{N}} w_i x_i^k \leq Q_j \\
 & x_i^k \in \{0, 1\} \quad \forall i \in \mathcal{N}
 \end{aligned}$$

When a concentrator-like model is considered, each site selected to host a facility must belong to its cluster and this can easily be handled by fixing $x_j = 1$ in each KP_j .

3.2 Branching scheme

The optimal solution of the RLMP can be fractional; hence the column generation routine is embedded in a branch-and-bound algorithm. Different branching rules can be applied [6] [8]. The choice of a suitable branching strategy is definitely critical because branching must not destroy the combinatorial structure of the pricing problem. The branching rule we used works as follows: for each customer we consider the set of candidate facilities for which there is a fractional assignment in the optimal solution of the RLMP. Then we partition this set in two subsets: aiming at a balanced partition we sort its elements by non-increasing value of fractional assignment, and we insert the elements alternately in the first and in the second subset. Also the set of remaining candidate facilities is partitioned in two subsets. Measuring the balance of the partition as the difference between the value of fractional assignment to the facilities in the first subset and the facilities in the second subset, the customer selected for branching is the one which produces the most balanced partition. Then we generate two subproblems and in each of them we forbid the assignment of the branching customer to the facilities of one of the two subsets. The effect of this branching rule on pricing is simply that of fixing to 0 some of the variables.

3.3 Columns management

The initial RLMP is populated with several columns to allow for a “warm start” of column generation. For all problems outlined above, owing to the capacity constraint, even finding a feasible solution can be difficult, since the decision version of the problems is *NP*-complete. However the initial columns must not necessarily form a complete feasible solution: they must only be in some way “reasonable”, that is they must have a structure similar to those which are likely to be part of feasible and good solutions. Therefore they can be generated in some heuristic way, as described in [8], for instance by selecting at random p candidate sites and then clustering the customers around them. In our implementation the initialization routine is run 10 times, generating 10 p columns. Furthermore we add the following dummy columns: first a dummy column having a 1 corresponding to constraints (19) and 0’s elsewhere; moreover a set of $|\mathcal{M}|$ dummy columns having a 1 corresponding to only one constraint (15) and 0’s elsewhere. This is done to ensure the existence of a feasible starting basis for each iteration of column generation. The second set of dummy columns can be dropped when the concentrator variants of the problems are used.

At each iteration, we insert into the RLMP all the columns with a negative reduced cost found by the pricing algorithm, if any. When a limit of 3000

columns in the RLMP is reached, we remove from the RLMP all the columns with a reduced cost superior to a given threshold, defined as the ratio between the current primal-dual gap and the number of potential facilities p . The removed columns are kept into a pool. At each column generation iteration we scan the pool. If any column is found to have negative reduced cost, it is reinserted into the RLMP. After three subsequent unsuccessful checks, the column is deleted also from the pool. As far as there are negative reduced cost columns in the pool, the pricing algorithm is not invoked.

3.4 Lagrangean bound

The lower bound obtained from the LMP can also be obtained through the Lagrangean relaxation of semi-assignment constraints (2) of the compact formulation:

$$\begin{aligned}
LR) \min \quad \omega_{LR} = & \sum_{j \in \mathcal{M}} f_j y_j + \sum_{i \in \mathcal{N}} d_{ij} x_{ij} + \\
& - \sum_{i \in \mathcal{N}} \lambda_i \left(\sum_{j \in \mathcal{M}} x_{ij} - 1 \right) + \\
& - \sum_{R \in \mathcal{R}} \gamma_R^u \left(u_R - \sum_{j \in R} y_j \right) - \sum_{R \in \mathcal{R}} \gamma_R^l \left(\sum_{j \in R} y_j - l_R \right) \\
\text{s.t.} \quad & \sum_{i \in \mathcal{N}} w_i x_{ij} \leq Q_j y_j & \forall j \in \mathcal{M} & \quad (20) \\
& x_{ij} \in \{0, 1\} & \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \\
& y_j \in \{0, 1\} & \forall j \in \mathcal{M}
\end{aligned}$$

The Lagrangean multipliers in LR correspond to the dual variables in the LMP and the Lagrangean subproblem can be decomposed in the same M binary knapsack problems as the pricing subproblem in the column generation approach (for the equivalence between Dantzig-Wolfe decomposition and Lagrangean relaxation the reader is referred to mathematical programming textbooks like [31]). Therefore column generation can be used as a method alternative to subgradient optimization to update the Lagrangean multipliers.

At each iteration of column generation the current values of the dual variables λ, γ^u and γ^l are used as Lagrangean multipliers to obtain a valid lower bound as follows. First, we solve the allocation subproblem: for each candidate facility j , let a “penalty” value π_j be computed as

$$\pi_j = -f_j + \tau_j - \sum_{R \in \mathcal{R} | j \in R} (\gamma_R^u - \gamma_R^l)$$

where τ_j is the value of optimal solution of KP_j , that has been found solving the pricing problem. Second, we solve the location subproblem, that consists in finding the set \mathcal{M}^{LR} of location sites which is feasible with respect to the regional constraints and contains the most profitable facilities (those with the lowest penalty values).

When regions do not overlap, \mathcal{M}^{LR} can be computed as follows. Let $\underline{\mathcal{M}}_R^{LR}$ be the set of the l_R sites with maximum π_j values in region R and let $\bar{\mathcal{M}}_R^{LR}$ be the set of the u_R sites with maximum π_j values in region R . When regions do not overlap, the best selection of facilities can be computed in two steps. First

the region lower bound constraints are satisfied by selecting the following set of facilities:

$$\mathcal{M}^{LR} := \bigcup_{R \in \mathcal{R}} \underline{\mathcal{M}}_R^{LR}$$

Second, the unselected facility $j^* \in \bigcup_{R \in \mathcal{R}} (\bar{\mathcal{M}}_R^{LR} \cap \underline{\mathcal{M}}_R^{LR})$ with highest π_{j^*} value is iteratively chosen, until the π_j value of the best facility left out is positive, or the upper bound constraints on the number of facilities in each region become tight. It is also easy to handle global upper and lower bounds on the number of facilities: the second step can be halted whenever one of the following conditions holds: (a) the global lower bound on the number of open facilities is satisfied and the π_j value of the best facility left out is positive; (b) the upper bound constraint on the number of facilities in each region becomes tight; (c) the global upper bound becomes tight.

Instead, when regions may overlap it is not easy to find the best set \mathcal{M}^{LR} . Therefore, we heuristically search for a good set of facilities, considering only a subset of regions $\mathcal{S} \subseteq \mathcal{R}$ defined as follows. We start with $\mathcal{S} = \emptyset$. Then, we iteratively choose the region $R \in \mathcal{R} \setminus \mathcal{S}$ with minimum cardinality which has an empty intersection with each region in \mathcal{S} , and we include R in \mathcal{S} until no more such regions can be found. Once \mathcal{S} has been found, we solve the location subproblem as described above, replacing \mathcal{R} with \mathcal{S} .

Thus, a feasible bound at each iteration t is obtained as

$$\omega_{LR}^t = - \sum_{j \in \mathcal{M}^{LR}} \pi_j + \sum_{i \in \mathcal{N}} \lambda_i + \sum_{R \in \mathcal{R}} (\gamma_R^l l_R - \gamma_R^u u_R)$$

where $\mathcal{M}^{LR} \subseteq \mathcal{M}$ is a set of sites selected to host facilities in the Lagrangean-relaxed solution.

In this way a sequence of valid lower bounds is computed during column generation and this allows to fix variables or even to prune the current node of the search tree before column generation is over.

We further exploit the relationship between column generation and Lagrangean relaxation outlined above to improve the dual variables via subgradient optimization [19] after each column generation iteration. Starting with the current optimal values of the dual variables λ , 100 subgradient iterations are executed. The step parameter is initialized at 2 and it is halved after every 10 iterations in which $\omega_{LR}^t \leq \omega_{LR}^{t-1}$, that is the current lower bound has not been improved with respect to the previous iteration.

Column generation is also speeded up by multiple pricing: instead of inserting into the RLMP only the optimal column for each candidate facility, if any is found with negative reduced cost, we add more (suboptimal) columns to enlarge the search space for the linear programming algorithm. This is particularly useful at the root node, when the column pool is still empty and the set of available columns may be small.

To this purpose we exploit the subgradient optimization algorithm and we insert into the RLMP the set of columns corresponding to each solution of the LR for which the Lagrangean lower bound improves upon the best incumbent Lagrangean lower bound.

3.5 Variable fixing

Given a solution of the Lagrangean relaxation LR, let ω_{LR} be its value and let $j_R^{WI} \in \operatorname{argmin}_{j \in \mathcal{M}^{LR} \cap R} \{\pi_j\}$ be the site in region R with minimum π_j value

that hosts a facility and $j_R^{BO} \in \operatorname{argmax}_{j \notin \mathcal{M}^{LR} \cap R} \{\pi_j\}$ be the site in region R with maximum π_j value that does not host a facility (WI stands for “worst in”, BO for “best out”). Let also v^* be a primal bound. The idea is to compute how forbidding the location of a facility in a site $j \in \mathcal{M}^{LR}$ would affect the dual bound. If $\pi_{j_R^{BO}} > 0$, or the lower bound constraint on region R is active (once building a facility in site j is forbidden, it is worth or necessary to open a facility in site j_R^{BO}), and $\lceil \omega_{LR} + \pi_j - \pi_{j_R^{BO}} \rceil \geq v^*$, then y_j can be fixed to 1. In a similar way, if the lower bound constraint is not active, $\pi_{j_R^{BO}} \leq 0$ and $\lceil \omega_{LR} + \pi_j \rceil \geq v^*$, then y_j can be fixed to 1. Analogously, if $\pi_{j_R^{WI}} < 0$ or the upper bound constraint on region R is active (once building a facility in j is imposed, it is worth or necessary to close the facility in site j_R^{WI}) and $\lceil \omega_{LR} - \pi_j + \pi_{j_R^{WI}} \rceil \geq v^*$, then y_j can be fixed to 0; in a similar way, if the upper bound constraint is not active, $\pi_{j_R^{WI}} \geq 0$ and $\lceil \omega_{LR} - \pi_j \rceil \geq v^*$, then y_j can be fixed to 0.

Once the π_j values have been computed, this variable fixing step takes $O(\sum_{R \in \mathcal{R}} |R|)$ time and it may reduce the problem size considerably. In our experiments variable fixing was done at each iteration of the subgradient optimization algorithm at the root node and only at the end of column generation at the other nodes in the search tree.

3.6 Primal heuristics

In order to find good feasible solutions early in the search tree, we integrated two primal heuristics in the main algorithm. Both of them are extensions of heuristics presented in [8] and consist of two phases: the selection of facility locations and the allocation of customers to the facilities.

The first one is a Lagrangean-based algorithm: let \mathcal{M}^{LR} be the set of sites in which a facility is activated in a Lagrangean-relaxed solution. We fix each location variable y_j to 1 if $j \in \mathcal{M}^{LR}$, to 0 otherwise. This selection of facility sites can violate some regional constraint. In this case the heuristic fails in identifying a feasible solution. Otherwise, we proceed to the allocation step as in [8]: when partitioning constraints are not violated, we use the same assignments which appear in the Lagrangean-relaxed solution, and the allocation of the other vertices is done as in the MTHG algorithm [29], with desirability coefficients $f_{ij} = -d_{ij}$; the heuristic could fail during the second step too, since it could be impossible to find a feasible allocation pattern. However, this heuristic proved to be sufficiently fast and effective to be run at each evaluation of a Lagrangean bound, that is, several times for each column generation iteration.

Another primal bound is computed with a rounding technique. In order to measure the desirability of building a facility in each site we define two sets of coefficients:

$$f_{ij} = \sum_{k \in Z^j} x_i^k z_k^j \quad \forall i \in \mathcal{N} \quad \forall j \in \mathcal{M}$$

and

$$\psi_j = \sum_{i \in \mathcal{N}} f_{ij} \quad \forall j \in \mathcal{M}.$$

Then, the selection of facility locations is done in two steps. In the first step the facility with highest ψ_j value is selected, among those which belong to some region whose lower bound is not satisfied yet, and do not belong to any region

whose upper bound is tight; this operation is repeated until all region lower bounds are satisfied, or no such facility can be found. In the second case the heuristic fails in finding a feasible solution and stops. In the second step, the facility with highest ψ_j value is selected, among those which do not belong to a region whose upper bound is tight; this operation is repeated until no such facility can be found, or the ψ_j value of the selected facility is less than 0.5. For the allocation phase, we follow the three steps of the original MTHG algorithm: first, the customers are assigned to the facilities in decreasing order of desirability coefficients f_{ij} as far as the capacity constraint allows the assignments; if some vertex remains unassigned, local search iterations are performed to produce a feasible solution; finally, if this step succeeds, a local search tries to improve the solution. This heuristic is much more time-consuming than the previous one. Therefore, it is used only in two cases: (a) at the root node, at each column generation iteration, provided that the value of the fractional LRMP solution is less than the double of the best known lower bound; (b) at each node of the search tree, only once the column generation process is over.

4 Branch-and-cut

The structure of a branch-and-cut algorithm is the following. First, the continuous relaxation of a compact formulation is solved. When the optimal solution of this relaxation is integral, it is also optimal. Otherwise, integrality is enforced in two ways, that is cutting planes and branching. Cutting planes are inequalities which are redundant for the original integer program, but are violated in the relaxed solution. Adding these cuts and re-optimizing the problem may yield a tighter lower bound. This cut-generation process can be iterated in order to obtain tighter approximations to the optimal integer solution. Branching is usually performed by selecting a variable whose value is fractional and considering two (or more) subproblems in which this variable is fixed to an integer value.

To benchmark our branch-and-price approach we chose as a competitor a general-purpose MIP solver, which uses branch-and-cut, that is ILOG CPLEX 8.1. In this section we review the classes of inequalities that CPLEX automatically generates and that we found to have a greater effect on the solution of single-source capacitated location problems.

Minimal cover cuts and generalized upper bound cover cuts. Each capacity constraint (3) is analyzed in order to find a group of variables forming a minimal cover. A minimal cover is a set of variables such that if all of them were set to 1, the constraint would be violated, but if any of them is set to 0, the constraint would not be violated. Therefore a valid inequality, called cover inequality, imposes that the sum of these variables has to be strictly less than their number. These cover inequalities can be strengthened in many ways. In particular CPLEX implements a search for generalized upper bound (GUB) cover cuts. A GUB cover imposes that at most one element in a subset of variables can be selected. CPLEX dynamically generates violated minimal cover cuts and GUB cover cuts, automatically finding how often to start this generation process and how many cuts to generate. These inequalities proved to be the most effective cuts for our class of integer programs.

Clique cuts. Clique cuts are added whenever a set of binary variables is identified such that at most one of the variables can be set to 1. These cuts are

derived from the examination of the relationship between the variables through constraint propagation techniques. This is done by CPLEX in a preprocessing step, before optimization starts.

Gomory fractional cuts. Gomory fractional cuts are an algebraic method for generating valid inequalities through integer rounding. This is a general purpose technique, that does not rely on any particular structure of the model. CPLEX allows the user to decide a number of parameters, including how many Gomory cuts must be generated and when. We kept the standard settings of CPLEX in our experiments since we observed they were quite effective.

5 Computational analysis

In this section we present the experimental results of our tests, in which we compared our general branch-and-price algorithm with the branch-and-cut-based solver implemented in ILOG CPLEX 8.1.

We have divided our experiments into three different parts. The purpose of the experiments in the first part is to evaluate the effect of fixed costs, capacities and cardinality constraints on the computing time required to achieve a provably optimal solution. The second part concerns the effect of the introduction of regional constraints and concentrator models. The third part includes experiments on large-size instances for which neither approach could reach proven optimality within a time-out of several hours; the purpose of these last experiments is to compare branch-and-price with branch-and-cut in terms of approximation, measured by the primal-dual gap.

The branch-and-price algorithm has been implemented in C++. ILOG CPLEX 8.1 libraries have been used to solve the LP relaxations. The program was compiled with GNU C/C++ compiler version 3.2.2 with full optimizations. All internal parameters of CPLEX have been kept to default values. All tests have been run on a Pentium IV 1.6GHz machine, running a Linux RedHat 9 operating system. Resource limitations were imposed to both algorithms: computation was halted after one hour of CPU time and the available RAM memory was limited to 512 MB.

5.1 Cardinality constraints and fixed costs

This first set of experiments is aimed at studying the effect of cardinality constraints and fixed costs.

To this purpose we considered two SSCFLP datasets, both taken from the literature: the first one (indicated as HOLM) consists of 71 instances and it is described in [20]; the second (indicated as DIAZ) consists of 57 instances and it is described in [14]. The instances in these datasets have up to 200 users and 30 candidate sites, with non-uniform fixed costs and capacities.

We designed our experiments to investigate to main questions: (Q1) “*What happens when a cardinality constraint is introduced?*” (Q2) “*How does the computing time change when fixed costs are made more uniform?*”

To answer question (Q1) we added a constraint on the maximum number of facilities, p . The value of p has been defined so that the average demand

satisfied by each of p facilities would be equal to 0.8:

$$p = \lfloor \frac{\sum_{i \in \mathcal{N}} w_i}{0.8 \sum_{j \in \mathcal{M}} Q_j / |\mathcal{M}|} \rfloor$$

To answer question (Q2) we also considered four different scenari: scenario (a) corresponds to the original HOLM and DIAZ instances with the additional cardinality constraint; in scenario (b) all fixed costs have been halved; in scenario (c) all fixed costs have been set to 0; in scenario (d) the fixed cost for each site has been set equal to the average fixed cost in its instance.

In tables 2 and 3 we report the results obtained for HOLM and DIAZ datasets respectively.

Table 2 consists of four horizontal blocks. The first one indicates the characteristics of each instance (in turn, problem id, number of users, number of candidate location sites, range in which the setup costs are generated, range in which the capacity constraints are generated); each of the three subsequent blocks refers to one of the scenari (a) - (c) described above. In each of these blocks we report the value of the solutions found by the branch-and-price and branch-and-cut algorithms, and the CPU time spent in proving its optimality. When the test exceeded the resource limitations, the corresponding ‘time’ column is marked with a dash. The last three rows of the table indicate the average computing times (neglecting the instances in which optimality was not proved), the number of instances solved to proven optimality and the average gap between the primal and the dual bounds at the end of computation (for the instances whose solutions were not proved optimal).

Table 3 has an analogous structure: in the first block we report the size of the instances. In the other blocks we report the results for scenari (a) to (d).

Question (Q1). In the rightmost columns of each table we mark with a capital ‘T’ the instances for which the cardinality constraint is tight in an optimal solution. Considering scenario (a), the cardinality constraint is tight only 6 of the 57 DIAZ instances, but has impact on many of the HOLM instances.

Question (Q2). From the analysis of the average results it is easy to see that non-uniform fixed costs make the instances much harder to solve for both branch-and-price and branch-and-cut. Computing times in scenario (c), without fixed costs, are two orders of magnitude lower than those in the other scenari and scenario (c) is the only one in which all DIAZ instances were solved within the time limit. In scenario (c) the branch-and-price algorithm could solve 63 HOLM instances out of 71 and CPLEX solved all of them; the increase in the average computing time of the branch-and-price algorithm with respect to scenari (a) and (b) is a consequence of the larger number of instances solved, since the average time is computed only on the solved instances.

Branch-and-price is more effective when fixed costs are absent or uniform: for instance in dataset DIAZ, scenario (c), it takes less average time than CPLEX to solve the same number of instances and in scenario (d) it takes about 75% the average time required by CPLEX and solves approximately the same number of instances. In dataset HOLM the number of instances closed by branch-and-price increases when costs become more uniform or vanish, while branch-and-cut is rather insensitive to this variation. When fixed costs are non-uniform branch-and-cut performed better than branch-and-price. This outcome was expected since our branch-and-price algorithm had been originally devised for the CPMP without fixed costs. When fixed costs are significant, location decisions (that is, where to open the facilities) are likely to become more critical than allocation

decisions (that is, to which facility each user must be assigned). Hence in these cases a two-levels branching policy like that proposed by Pirkul [36] and Diaz and Fernandez[14] can be more indicated for branch-and-price algorithms.

5.2 Regional constraints and concentrators

The purpose of the second set of experiments is to evaluate the effect of introducing cardinality constraints and fixed costs in SS-CFLP and CPMP models, from the viewpoint of the computational resources required to reach provable optimality and from the viewpoint of the optimal value. Then we also tried to evaluate the effect of regional constraints and of constraints imposing that a user hosting a facility must be allocated to it (i.e. the “concentrator” case). Hence the questions we investigated are the following: (Q3) “*What is the effect of introducing cardinality constraints in SS-CFLP?*” (Q4) “*What is the effect of introducing fixed costs in the CPMP?*” (Q5) “*What is the effect of introducing regional constraints of different types?*” (Q6) “*What is the effect of the constraint $y_i = x_{ii}$ in these models?*”

For this second set of experiments we used the dataset based on 20 CPMP instances taken from the OR Library web site. and already used in several papers [3] [35]. This dataset is more significant for these experiments for the following reasons: (1) they are Euclidean, in that distances between users and facilities are computed according to the Euclidean metric in two dimensions; therefore their distance matrices are more realistic than random matrices; (2) in the literature regional constraints have been so far considered in addition to models as CPMP [32] and SS-CFLP with cardinality constraints [42] and the instances considered in these papers are similar to ours; (3) the set of sites which can host facilities coincides with the set of the users; this restriction contributes to make these instances more realistic. Moreover it allows to evaluate the effect of the constraint $y_i = x_{ii}$.

The original instances have a cardinality constraint with $p = N/10$ and no fixed costs. We considered three variants: CPMP with cardinality constraint and no fixed costs, SS-CFLP with fixed costs and no cardinality constraint, and CARD+FIX with cardinality constraint and fixed costs. Fixed costs have been randomly generated from a uniform distribution in the range $[Q_j/2, \dots, 3Q_j/2]$.

We generated seven types of regional constraints, indicated by capital letters from *A* to *G*. Regions in types *A* to *E* correspond to partitions of the instance graph, while regions in type *F* and *G* instances may overlap. In table 1 we describe in details how these instances were generated. We report the regions type (first column) the number of regions (second column), the ranges in which the lower and the upper bounds of each region are chosen (third and fourth column) and the average overlapping of the regions (fifth column).

Let r be the number of regions in which a graph with N vertices has to be partitioned. Consider as “covered” a vertex belonging to a region: so each of the vertices in the original CPMP instances is initially “uncovered”. In each instance regions are created as follows: first, the vertex with minimum average distance between the other vertices is selected. This is the “seed” of the new region. Second, the uncovered vertex with minimum distance between one of the vertices in the new region is iteratively included in the region, until the number of vertices in the region is N/r . This process is repeated, partitioning in $r - 1$ regions the graph of the uncovered nodes.

Regions overlapping is allowed by iteratively selecting the pair of vertices with minimum distance, which belong to different regions, and imposing that

Constraint Type	Number of regions	Lower bound	Upper bound	Overlap percentage
A	1	0	p	0%
B	2	2	3	0%
C	$\lceil \frac{1}{2}p \rceil$	$[0, 1]$	$[1 \dots 3]$	0%
D	$\lceil \frac{2}{3}p \rceil$	$[0, 1]$	$[1 \dots 2]$	0%
E	p	1	1	0%
F	$\lceil \frac{3}{5}p \rceil$	$[0, 1]$	$[1 \dots 3]$	50%
G	$\frac{3}{5}p$	$[0, 1]$	$[1 \dots 3]$	80%

Table 1: Generation of regional constraints

each of these vertices belongs to both regions, until a predefined overlapping ratio is reached.

The outcome of the experiments is reported in Tables 4 and 5. Each Table is divided in seven vertical blocks and four horizontal blocks. Each vertical block corresponds to a region type (whose ID is reported in the first cell of each block). The first horizontal block includes the instance parameters (region type and instance ID). The second, third and fourth horizontal blocks refer to the CPMP, SSCFLP and CARD+FIX variants of the problem respectively. In each of these blocks we indicate, for each instance, the value of the optimal solution, the maximum distance between a facility and one of its assigned users and the maximum load of a facility (computed as the ratio between the sum of the demands of users assigned to the facility and the capacity of the facility). For each of these values, the percentage increase (or decrease) with respect to the value in the optimal solution of the problem without regional constraints is indicated. Finally, the time (in seconds) required to prove optimality by both branch-and-price and branch-and-cut is reported. In the last row of each vertical block we indicate the average results for each region type.

Question (Q3). Comparing the results reported in block SSCFLP with those in block CARD+FIX in Tables 4 and 5 it can be seen that the cardinality constraint has little effect on the computing time for both methods. Against the intuition, the introduction of this constraint does not contribute to reduce the search space and thus the difficulty of the problem: in some cases the opposite effect is observed. We remark that these experiments have been made only with the given value of the ratio $p/N = 1/10$. Also the effect on the value of the optimal solution was almost negligible: for 50 users instances we observed a 2.25% increase in the minimum cost, while for 100 users instances the observed increase was about 0.65%. Furthermore, this increase vanishes as tightest regional constraints are introduced.

Question (Q4). On the opposite, comparing the results in block CPMP with those in block CARD+FIX in Tables 4 and 5 it is clear that removing the fixed costs really changes the structure of the problem. For the branch-and-cut, the CPMP is more difficult than the CARD+FIX variant: computing the optimal solution of the CPMP instances requires about a double CPU time with respect to the variant with fixed costs. Branch-and-price does not have a regular behaviour: on some instances (e.g. 8 and 18) the optimization of the CPMP version is more difficult than that of the CARD+FIX version, while on other instances the opposite trend is observed.

If CPMP is interpreted as a CARD+FIX problem in which obtaining the best average service for the users (minimum allocation costs) is much more relevant than searching for the best trade-off between fixed costs and allocation costs, then it makes sense to compare the optimal CPMP solution value with the

contribution of the allocation costs to the optimal CARD+FIX solution value. When the CPMP model is used, a reduction of about 6% and 9.5% in the allocation costs is obtained for the $N = 50$ and $N = 100$ instances respectively, while the corresponding increase on the overall solution value is about 9.5% and 15.5% respectively.

Question (Q5). A comparison of the values reported on the last row of each block in Tables 4-5 (the “avg” row), it is possible to evaluate the effect of the different regional constraints on the computing time and the optimal value. From the viewpoint of computational resources no relevant differences are observed: the computing time required by branch-and-price and branch-and-cut remains of the same order of magnitude.

In CPMP instances, with no fixed costs, when the number of regions increases (from A instances to E instances) the computing time decreases and this holds for both approaches. On the contrary in SSCFLP instances, without cardinality constraints, the computing time increases when the number of regions increases. The computing time with overlapping regional constraints (types F and G) are not very different from those without regional constraints.

The value of the optimal solution is strongly affected by regional constraints: for instance the average optimal value of all the instances in class E is about 25% worse than that for class A . When regions are large and overlap (types F and G) the effect on the optimal value is small.

Instances with fixed costs, that is those in classes SSCFLP and CARD+FIX, are more affected by regional constraints. This was expected because constraints on location variables have impact both on assignment costs and on location costs: in the former case, because it can be necessary to locate facilities in sites farther away from users; in the second case, because it can be necessary to use sites with higher fixed costs.

Since regional constraints are usually considered as an option to enforce some kind of “equity” in the geographical distribution of the facilities, we analyzed how the worst-case service varies when the regional constraints are introduced. In particular we observed the maximum user-facility distance and the maximum load assigned to a facility. Neither of these parameters significantly decreases when the number of regions increases. This puts some doubt on the actual effectiveness of regional constraints to achieve fairness among the users and equitable distribution of the workload among the facilities. It seems to us that equity can be better pursued by suitable models in which min-max objective functions are explicitly considered and optimized.

Question (Q6). All tests described in the previous paragraph have been repeated for the concentrator variant, in which user i must be assigned to facility j whenever the facility is located in the same site of the user. Results are reported in Tables 6 and 7.

We can observe that the average computing time of branch-and-cut is improved; this was also expected since branch-and-cut takes advantage of a reduced number of variables. On the contrary the computing time of branch-and-price is worse, even if the N constraints (15) can be removed from formulation (13) – (18).

The value of the optimal solution is not affected: it did not change in any of our tests.

5.3 Lower bounds and gaps

We compared the lower bounds achieved by branch-and-cut and branch-and-price at the root node and we observed that the lower bound provided by branch-and-price is consistently tighter: branch-and-cut could never reduce the duality gap with respect to the best known feasible solution below 1.2%; on the contrary branch-and-price achieved duality gaps ranging from the 0.5 to the 0.8 as much.

We also compared the best solution found and the primal-dual gaps obtained by branch-and-price and branch-and-cut when solving large scale instances (900 customers). In this case the superiority of branch-and-price is overwhelming: CPLEX experienced memory overflow problems and could neither solve the linear relaxation at the root node nor find any feasible solution. On the contrary branch-and-price yielded feasible solution with an approximation guarantee of about 4% in average. Moreover, we observed that on large-scale instances, when the concentrator restriction is introduced, the computing time required by branch-and-price is reduced to approximately one half.

6 Conclusions

From the average results reported in the last rows of the tables above the following observations can be done. Both branch-and-cut and branch-and-price could solve all instances with 50 users: branch-and-cut was in average faster; branch-and-price worked better when both cardinality constraints and fixed costs were present (class CARD+FIX). For instances with 100 users branch-and-cut was clearly superior to branch-and-price, mainly in terms of instances solved to proven optimality within the time limit. Looking at the detailed results reported in the tables however one can see that there are several instances in which branch-and-price wins. Therefore the superiority of branch-and-cut holds in average, on a sufficiently large number of instances, but there is no guarantee that it will be the best approach on any given single instance.

A very important remark concerns the trade-off between the width of applicability and the effectiveness of the algorithms examined here. As stated in the introduction, our goal was not to compare two specific algorithms but rather two approaches, designed to be widely applicable to all location problems considered here. This strive for generality is obviously paid in terms of effectiveness. Therefore for each single location problem considered here it may be possible to obtain better results than those presented above, by incorporating specialized cutting, branching and heuristic procedures. Examples of specialized cutting planes devised for particular network location problems are those of [2] and [1]; specialized branching procedures are illustrated for instance in [14] and [36]; specialized heuristic procedures can be found in [20] and [13]. The performances of both branch-and-cut and branch-and-price general solvers can be strongly affected by such tailored additions. Branch-and-price exploits a tight lower bound, provided by the set partitioning reformulation of the problems. This gives advantage on large scale instances compared with branch-and-cut algorithms.

Last but not least, the branch-and-cut code we have used as a benchmark for our branch-and-price framework is a commercial solver not available for free, while our code is freely available upon request for scientific purposes. It needs a linear programming subroutine to solve the linear relaxation of the restricted master problem: the results presented above have been obtained with the ILOG

CPLEX simplex algorithm but any linear programming solver can be used instead.

Acknowledgments. We are grateful to Zvi Drezner for encouraging us to write this paper. We also acknowledge the kind support of ACSU - Associazione Cre-masca Studi Universitari for its support to the Operations Research Laboratory of our Department, where this work was done.

Problem	N	M	F	CAP	Setup * 1.0				load factor 0.8				Setup * 0.5				load factor 0.8				S. 1.0	S. 0.5
					Value	B&P time	CPX value	CPX time	Value	B&P time	CPX value	CPX time	Value	B&P time	CPX value	CPX time	Value	B&P time	CPX value	CPX time		
1	50	10	[300..700]	[100..400]	8848	0.38	8848	0.17	7331	0.44	7331	0.17	5814	0.27	5814	0.16	T	T				
2			300		7914	0.38	7914	0.15	6864	0.27	6864	0.16	5814	0.26	5814	0.16	T	T				
3			500		9314	0.30	9314	0.15	7564	0.34	7564	0.16	5814	0.26	5814	0.15	T	T				
4			700		10714	1.53	10714	0.74	8264	0.26	8264	0.15	5814	0.26	5814	0.14	T	T				
5			[300..700]	200	8838	0.41	8838	0.40	7026	0.38	7026	0.12	5077	0.41	5077	0.12	T	T				
6			300		7777	0.20	7777	0.06	6427	0.46	6427	0.11	5077	0.19	5077	0.05	T	T				
7			500		9488	1.32	9488	0.28	7327	0.54	7327	0.12	5077	0.40	5077	0.12	T	T				
8			700		11088	1.10	11088	0.45	8227	1.05	8227	0.19	5077	0.40	5077	0.12	T	T				
9			[300..700]	300	8462	0.30	8462	0.14	7197	0.36	7197	0.14	5932	0.32	5932	0.14	T	T				
10			300		7732	0.38	7732	0.14	6832	0.35	6832	0.14	5932	0.31	5932	0.14	T	T				
11			500		8932	0.52	8932	0.14	7432	0.42	7432	0.14	5932	0.30	5932	0.14	T	T				
12			700		10132	1.03	10132	0.15	8032	0.34	8032	0.14	5932	0.32	5932	0.14	T	T				
13	50	20	[300..700]	[100..400]	8252	0.40	8252	0.32	6382	0.50	6382	0.26	4437	0.34	4437	0.23	T	T				
14			300		7137	0.48	7137	0.25	5787	0.67	5787	0.22	4437	0.34	4437	0.23	T	T				
15			500		8808	0.56	8808	0.28	6687	0.37	6687	0.24	4437	0.33	4437	0.24	T	T				
16			700		10408	1.25	10408	0.57	7587	0.41	7587	0.26	4437	0.32	4437	0.23	T	T				
17			[300..700]	200	8227	0.93	8227	0.44	6370	0.32	6370	0.23	4425	0.53	4425	0.23	T	T				
18			300		7125	0.37	7125	0.23	5775	0.48	5775	0.23	4425	0.54	4425	0.24	T	T				
19			500		8886	2.09	8886	1.43	6675	0.27	6675	0.24	4425	0.55	4425	0.22	T	T				
20			700		10486	2.17	10486	1.44	7575	0.32	7575	0.27	4425	0.54	4425	0.24	T	T				
21			[300..700]	300	8171	0.76	8171	0.18	6785	0.84	6785	0.22	5373	3.14	5373	2.13	T	T				
22			300		7473	4.50	7473	2.54	6423	2.94	6423	1.95	5373	3.14	5373	2.17	T	T				
23			500		8873	3.41	8873	2.23	7123	2.30	7123	0.61	5373	3.09	5373	2.14	T	T				
24			700		10273	2.99	10273	1.66	7823	2.18	7823	0.46	5373	3.10	5373	2.04	T	T				
25	150	30	[300..700]	[100..400]	11630	-	11630	15.62	9978	682.60	9978	1.15	8086	31.01	8086	0.98	T	T				
26			300		10771	-	10771	15.27	9436	54.79	9436	0.98	8086	30.72	8086	0.97	T	T				
27			500		12322	-	12322	43.97	10336	1594.03	10336	2.66	8086	31.01	8086	0.97	T	T				
28			700		13722	-	13722	43.55	11171	-	11171	9.23	8086	30.69	8086	0.96	T	T				
29			[300..700]	200	12371	-	12371	226.02	10257	-	10257	130.12	7967	2690.72	7967	4.65	T	T				
30			300		11394	-	11331	670.97	9744	-	9742	46.10	7967	2697.21	7967	4.66	T	T				
31			500		13483	-	13331	546.72	10844	-	10831	273.28	7967	2682.70	7967	4.63	T	T				
32			700		15450	-	15331	1040.26	11944	-	11831	763.61	7967	2687.10	7967	4.64	T	T				
33			[300..700]	300	11629	512.66	11629	3.22	9932	498.81	9932	1.51	8068	31.36	8068	1.06	T	T				
34			300		10632	40.10	10632	1.33	9418	102.52	9418	1.27	8068	31.13	8068	1.06	T	T				
35			500		12232	29.18	12232	1.34	10232	25.66	10232	1.32	8068	31.29	8068	1.06	T	T				
36			700		13832	31.97	13832	1.35	11032	39.33	11032	1.33	8068	31.10	8068	1.06	T	T				
37			[300..700]	600	11258	34.80	11258	1.03	10049	39.92	10049	0.99	8824	74.40	8824	1.02	T	T				
38			300		10624	44.66	10624	0.95	9724	36.20	9724	0.98	8824	74.88	8824	0.99	T	T				
39			500		11824	51.50	11824	1.02	10324	44.59	10324	0.98	8824	75.36	8824	0.99	T	T				
40			700		13024	28.89	13024	0.97	10924	38.64	10924	0.99	8824	74.80	8824	1.01	T	T				
41	90	10	[300..700]	[100..400]	6700	76.08	6700	18.14	5728	29.09	5728	8.90	4730	79.82	4730	8.30	T	T				
42	80	20	[300..700]	[100..400]	5663	100.83	5663	6.90	4666	23.80	4666	6.33	3669	21.64	3669	2.63	T	T				
43	70	30	[300..700]	[100..400]	5214	1.01	5214	0.55	4292	1.32	4292	0.55	3368	1.62	3368	0.54	T	T				
44	90	10	[300..700]	[100..400]	7270	8.06	7270	2.08	5829	21.17	5829	0.57	4388	3.33	4388	0.30	T	T				
45	80	20	[300..700]	[100..400]	6251	4.34	6251	0.49	5018	3.63	5018	0.50	3785	3.32	3785	0.53	T	T				
46	70	30	[300..700]	[100..400]	5965	2.48	5965	0.78	4949	1.31	4949	0.79	3919	2.78	3919	0.80	T	T				
47	90	10	[300..700]	[100..400]	6719	1598.23	6719	3.16	5354	65.78	5354	4.43	3975	41.87	3975	0.85	T	T				
48	80	20	[300..700]	[100..400]	6179	29.11	6179	9.20	5017	19.06	5017	9.54	3857	20.75	3857	11.75	T	T				
49	70	30	[300..700]	[100..400]	5609	61.82	5609	13.68	4529	5.68	4529	0.96	3417	3.41	3417	0.58	T	T				
50	100	10	[300..700]	[100..400]	8808	-	8808	6.44	7654	-	7654	6.63	6500	-	6500	11.29	T	T				
51			20	[300..700]	[100..400]	7422	-	7414	40.35	6219	-	6219	9.71	4971	2475.37	4971	14.56	T	T			
52			10	[300..700]	[100..400]	9178	16.32	9178	1.47	7785	39.61	7785	1.51	6390	36.24	6390	1.00	T	T			
53			20	[300..700]	[100..400]	8531	6.63	8531	0.58	7151	12.04	7151	0.59	5770	9.99	5770	0.57	T	T			
54			10	[300..700]	[100..400]	8777	4.39	8777	0.95	7416	62.76	7416	1.19	5861	13.41	5861	1.10	T	T			
55			20	[300..700]	[100..400]	7654	210.13	7654	2.62	6324	73.49	6324	2.49	4894	13.29	4894	2.23	T	T			
56	200	30	[300..700]	[100..400]	21120	-	21103	119.79	16620	-	16618	130.69	12178	-	12118	104.75	T	T				
57			300		26411	-	26039	939.41	19318	-	19318	127.78	12178	-	12118	104.94	T	T				
58			500		37871	-	37239	796.17	25720	-	25239	611.74	12178	-	12118	104.88	T	T				
59			700		27311	-	27282	341.88	20519	1434.28	20519	27.72	12178	-	12118	104.66	T	T				
60			[300..700]	200	20854	92.33	20854	1.88	17854	51.74	17854	1.89	14854	107.52	14854	1.89	T	T				
61			300		24454	42.29	24454	2.20	19654	45.97	19654	1.95	14854	107.05	14854	1.90	T	T				
62			500		32689	-	32643	193.20	23854	37.10	23854	2.08	14854	107.71	14854	1.91	T	T				
63			700		25105	525.87	25105	6.84	20701	56.66	20701	2.50	14854	107.46	14854	1.91	T	T				
64			[300..700]	300	22476	-	22476	68.19	20226	2891.49	20226	52.19	17976	-	17976	64.74	T	T				
65			300		25176	2326.14	25176	59.09	21576	2266.30	21576	59.30	17976	-	17976	64.63	T	T				
66			500		31657	-	31415	330.87	24726	-	24726	73.22	17976	-	17976	63.69	T	T				
67			700		24848	422.25	24848	14.67	20248	32.85	20248	0.92	14619	37.51	14619	0.87	T	T				
68			[300..700]	600	20932	17.19	20932	1.13	17932	55.73	17932	1.10	14932	18.91	14932	1.10	T	T				
69			300		24532	15.47	24532	1.11	19732	20.84	19732	1.17	14932	18.86	14932	1.09	T	T				
70			500		32392	-	32321	135.92	23932	39.66	23932	2.27	14932	39.90	14932	2.28	T	T				
71			700		25880	-	25893	-	21019	-	20973	270.53	14932	39.98	14932	2.32	T	T				
Avg computing time							122.36	82.11		177.38	37.54		232.33			10.24						
Solved instances							52.00	70.00		59.00	71.00		63.00			71.00						

Problem Id	N	M	Setup * 1.0 load factor = 0.8				Setup * 0.5 load factor = 0.8				Setup * 0.0 load factor = 0.8				AVGFIX				S. 1.0	S. 0.5
			B&P Value	B&P time	CPX Value	CPX time	B&P Value	B&P time	CPX Value	CPX time	B&P Value	B&P time	CPX Value	CPX time	B&P Value	B&P time	CPX Value	CPX time		
1	15	10	2014	2.51	2014	1.42	1147	0.30	1147	0.33	207	0.07	207	0.08	2081	0.08	2081	0.13		T
2			4251	0.63	4251	1.70	2319	2.11	2319	1.46	201	0.07	201	0.04	3668	0.21	3668	1.30		T
3			6051		6051	2.40	2159	1.47	2159	0.86	185	0.05	185	0.03	5427	0.12	5427	0.12		T
4			7168		7168	10.91	3785	506.69	3785	1.05	195	0.06	195	0.03	6328	0.18	6328	0.14		
5			4551		4551	9.24	2466	3.97	2466	1.23	269	0.06	269	0.04	4273	0.14	4273	0.16		
6			2269	1.81	2269	1.67	1272	0.11	1272	0.24	213	0.03	213	0.03	2189	0.11	2189	0.17		
7	30	15	4366	1201.23	4366	8.96	2422	45.49	2422	10.25	269	0.06	269	0.04	4069	0.44	4069	2.82		
8			1244	0.06	1244	0.03	1231	0.05	1231	0.04	381	0.13	381	0.11	2216	0.05	2216	0.04		
9			2480		2480	19.22	1447	1.93	1447	7.99	275	0.09	275	0.08	2664	0.57	2664	14.58		
10			23112	2102.66	23112	4.40	11745	1.37	11745	20.82	172	0.08	172	0.05	21733	0.37	21733	19.75		
11			3447	7.53	3447	59.36	1937	0.33	1937	5.34	255	0.10	255	0.06	3245	0.58	3245	139.32		
12			3711	440.25	3711	39.34	2103	0.73	2103	4.60	297	0.08	297	0.08	3394	0.33	3394	0.62		
13			3760	335.45	3760	178.79	2090	0.97	2090	5.78	288	0.23	288	0.19	3380	1.52	3380	11.07		
14			6767		6579	4.43	3654	2220.36*			323	0.08	323	0.06	6571	0.24	6571	0.24	T	T
15			7816	47.29	7816	21.93	4095	1.18	4095	28.08	175	0.11	175	0.08	7868	0.44	7868	9.20		
16			11543		11543	16.71	5958	5.96	5958	28.27	175	0.10	175	0.07	11648	0.39	11648	7.31		
17			9932	2025.03*	9884	18.93	5239		5239	11.56	243	0.07	243	0.08	9054	0.65	9054	1.90		
18	40	20	15628		15607	68.21	8037	150.39	8037	35.43	238	0.11	238	0.11	14427	1.28	14427	555.28		
19			18769	1506.45*	18683	2535.92	9596		9557	4.14	221	0.13	221	0.18	17301	1.49	17301	13.73		
20			26680		26584		13558		13504	1053.73	185	0.19	185	0.11	24862	2.65	24862	604.14		
21			7302		7301		3872	10.39	3872	400.90	265	0.19	265	0.15	6937	0.43	6937	26.11		
22			3315		3314	119.82	1926	4.78	1926	22.85	328	0.30	328	0.29	3166	64.81	3166	5.16	T	T
23			6036	118.78	6036	28.22	3338	0.84	3338	10.61	320	0.22	320	0.25	6160	1.11	6160	2.26	T	T
24			6357	2683.41*	6346	14.73	3474		3474	14.57	407	0.71	407	1.96	6550	100.17	6550	7.46	T	T
25			8947		8947	14.87	4745		4745	8.08	359	3.19	359	1.91	10024	29.35	10024	5.44		
26	50	20	4448	9.92	4448	365.19	2502	14.21	2502	384.02	288	0.19	288	0.22	4347	2.04	4347	12.97		
27			10974		10963		5981		5979		455	4.28	455	3.42	11734	81.82	11734	5.38	T	T
28			11590		11500	63.49	6080		6025	149.40	351	0.64	351	3.04	11327	478.67	11327	26.09	T	T
29			9941		9832	104.28	5239		5239	234.44	402	0.25	402	0.18	9662	224.95	9662	17.01		
30			10935		11074		5735		2424.67	5721	249	0.35	249	0.14	9497	0.54	9497	8.33		
31			4525	2271.85*	4466	2864.39	2466		2466	17.81	285	0.21	285	0.12	3789	26.47	3789	20.39		
32			10266	1385.73*	9881	37.29	5295		5295	354.32	387	1.57	387	5.52	8965	19.66	8965	7.21	T	
33			39463		41362		20026	1468.96	20943		311	0.89	311	0.33	38010	1.55	38010	740.29		
34	60	30	4819		4736		2796		2769	527.66	315	0.30	315	0.37	8330	2900.03	8330	187.95		
35			5456		5456	26.82	2999	61.11	2999	21.20	256	1.06	256	1.33	8184	26.98	8184	459.94		
36			17243		16781	2008.69	8939		8749	2340.74	244	0.25	244	0.22	27416	7.44	27416	48.28		
37			14783		14803		7771	2742.9*	7692		262	0.99	262	6.04	23147	4.51	23147	34.45		
38			47280		47249	112.95	23863		23875		199	0.30	199	0.44	54037	2.86	54037	142.28		
39			41197		41007	856.11	20851		20766	3111.67	233	0.26	233	0.25	55083	8.67	55083	535.18		
40			62004		64108		31267		31056		183	0.71	183	0.98	73969	2.44	73969	465.55		
41			20202	462.65	17246	17.73	10551	815.93	9196	45.63	506	2.00	506	13.43	60052		60052	46.30		
42	75	30	7920		7887	159.80	4366		4348		318	0.42	318	0.38	11070	205.36	11070	903.45		
43			3627	3.00	5114	499.35	3020		3020	38.69	378	9.49	378	13.43	10274	8.52	10274	45.89		
44			36688		37636		18389		18959		211	0.55	211	0.29	38065	5.25	38065	211.58		
45			17854		17676	18.23	9308		9233	135.81	335	1.01	335	1.64	29359	63.67	29359	190.55		
46			51409		50250		26078		24718	2676.39	348	1.42	348	0.42	71251	70.66	71251	131.01		
47			68426		68452		34603		34995		291	0.43	291	0.31	67682	3.79	67682	416.88		
48			63385		58964	314.92	32181		30036	130.99	423	1.37	423	8.37	85726	1053.53	85729	1408.68		
49			85668		81301		42803		41184		285	0.45	285	0.57	92208	95.09	92208	1325.22		
50	90	30	6047		5937	248.99	3570		3522	643.45	550	14.38	550	15.47	12720		12712	66.75		
51			9123	3277.43	9123		5053		5055		369	0.70	369	0.68	12946	466.02	12946	301.92		
52			38575		35324		19572		17681		357	1.78	357	1.30	47157	30.77	47157	245.88		
53			30140		30038	339.95	15558		15379	808.02	322	0.88	322	0.41	37580	9.57	37580	1639.23		
54			51461		43853	39.32	26203		22485	59.48	462	15.70	462	12.22	84560	37.89	84562	35.64		
55			71009		72034		35849		36422		351	0.76	351	0.44	84255		84313			
56			64474		64474	118.36	32759		32755	127.05	474	2.18	474	6.58	110009	2240.50	110009	82.10		
57			49791	1605.43	49791	6.80	25456	2239.79	25456	5.48	560	3.77	560	10.10	107410		107410	323.92		
Avg computing time			565.76		271.04		301.31		309.38		1.32		2.01		156.36		205.54			
Solved instances			17.00		42.00		26.00		44.00		57.00		57.00		53.00		56.00			
Avg. (PB - DB) / DB gap			6.70%				9.05%				0.00%				2.14%					

R. Type	Inst.	CPMP				SS-CFLP				CARD+FIX			
		Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time
A	1	713	0.00%	3.58	4.29	1217	0.00%	0.65	0.27	1242	0.00%	5.94	3.84
	2	740	0.00%	0.36	0.28	1169	0.00%	0.68	0.27	1169	0.00%	0.24	0.25
	3	751	0.00%	1.41	1.33	1141	0.00%	1.17	0.39	1162	0.00%	0.36	0.5
	4	651	0.00%	0.46	0.56	1141	0.00%	0.89	0.55	1141	0.00%	0.37	0.41
	5	664	0.00%	0.94	0.75	1103	0.00%	4.24	0.44	1103	0.00%	0.45	0.38
	6	778	0.00%	0.54	0.67	1240	0.00%	1.33	0.44	1257	0.00%	5.73	16.39
	7	787	0.00%	5.08	17.14	1175	0.00%	1.64	1.93	1213	0.00%	1.71	6.45
	8	820	0.00%	1891.39	207.91	1184	0.00%	1.84	0.39	1248	0.00%	4.89	8.14
	9	715	0.00%	1.92	10.5	1141	0.00%	2.31	0.37	1152	0.00%	1.25	0.74
	10	829	0.00%	10.81	84.3	1159	0.00%	0.75	0.28	1248	0.00%	39.25	49.57
A	Avg		0.00%	191.65	32.77		0.00%	1.55	0.53		0.00%	6.02	8.67
B	1	713	0.00%	3.18	3.65	1217	0.00%	0.29	0.27	1242	0.00%	5.43	2.44
	2	740	0.00%	0.4	0.29	1169	0.00%	0.26	0.26	1169	0.00%	0.23	0.26
	3	751	0.00%	1.29	5.65	1150	0.79%	1.13	0.49	1162	0.00%	0.37	0.51
	4	651	0.00%	0.56	0.56	1141	0.00%	0.4	0.92	1141	0.00%	0.53	0.47
	5	664	0.00%	0.84	0.79	1103	0.00%	4.61	0.43	1103	0.00%	0.37	0.34
	6	778	0.00%	0.54	0.77	1240	0.00%	1.05	0.45	1257	0.00%	4.9	11.19
	7	787	0.00%	4.69	22.73	1175	0.00%	0.64	0.61	1213	0.00%	1.7	10.86
	8	821	0.12%	2559.2	219.64	1190	0.51%	0.74	0.64	1248	0.00%	4.32	7.28
	9	715	0.00%	1.36	9.56	1141	0.00%	1.11	0.37	1152	0.00%	1.19	0.79
	10	829	0.00%	11.09	77.86	1183	2.07%	0.98	0.69	1248	0.00%	21.28	28.88
B	Avg		0.01%	258.32	34.15		0.34%	1.12	0.51		0.00%	4.03	6.30
C	1	713	0.00%	2.01	1.26	1258	3.37%	1.34	0.61	1258	1.29%	0.57	0.44
	2	761	2.84%	1.05	0.38	1216	4.02%	1.35	0.63	1216	4.02%	1.65	0.61
	3	751	0.00%	1.12	1.16	1155	1.23%	1.01	0.62	1162	0.00%	0.49	0.64
	4	652	0.15%	0.81	0.41	1141	0.00%	0.49	0.47	1141	0.00%	0.4	0.51
	5	664	0.00%	0.64	0.92	1103	0.00%	5.75	0.98	1103	0.00%	0.59	0.47
	6	799	2.70%	0.76	0.84	1274	2.74%	7.1	14.2	1274	1.35%	1.7	11.89
	7	787	0.00%	5.45	21.38	1175	0.00%	1.55	0.43	1213	0.00%	2.03	7.39
	8	820	0.00%	1843.11	205.44	1195	0.93%	0.5	0.32	1248	0.00%	3.02	13.08
	9	715	0.00%	0.85	0.69	1152	0.96%	1.57	0.72	1152	0.00%	1.87	0.61
	10	835	0.72%	23.33	79.78	1183	2.07%	0.39	0.35	1248	0.00%	22.55	46.35
C	Avg		0.64%	187.91	31.23		1.53%	2.11	1.93		0.67%	3.49	8.20
D	1	744	4.35%	3.45	2.26	1270	4.35%	0.43	0.39	1270	2.25%	0.42	0.38
	2	770	4.05%	1.91	0.93	1209	3.42%	2.15	0.81	1209	3.42%	1.82	1.2
	3	755	0.53%	3.15	1.72	1186	3.94%	2	2.02	1186	2.07%	3.96	0.72
	4	651	0.00%	1.16	0.57	1163	1.93%	0.73	0.5	1165	2.10%	0.78	0.47
	5	664	0.00%	0.5	0.64	1103	0.00%	0.58	0.32	1103	0.00%	0.56	0.31
	6	778	0.00%	0.67	0.7	1240	0.00%	1.44	0.44	1257	0.00%	3.86	10.79
	7	845	7.37%	8.2	20.93	1315	11.91%	54.87	17.3	1315	8.41%	30.29	21.86
	8	820	0.00%	41.67	32.87	1285	8.53%	1.8	5.96	1285	2.96%	1.84	6.25
	9	717	0.28%	3.02	9.29	1146	0.44%	0.4	0.39	1165	1.13%	1.48	13.5
	10	848	2.29%	52.83	60.16	1217	5.00%	0.53	0.52	1253	0.40%	7.27	18.2
D	Avg		1.89%	11.66	13.01		3.95%	6.49	2.87		2.27%	5.23	7.37
E	1	714	0.14%	2.35	0.64	1266	4.03%	2.01	4.39	1266	1.93%	1.68	4.44
	2	740	0.00%	0.65	0.26	1170	0.09%	0.21	0.24	1170	0.09%	0.22	0.25
	3	751	0.00%	1.1	0.83	1162	1.84%	0.6	0.38	1162	0.00%	0.49	0.37
	4	652	0.15%	0.58	0.38	1141	0.00%	0.45	0.43	1141	0.00%	0.61	0.36
	5	664	0.00%	1.5	0.69	1103	0.00%	0.55	0.31	1103	0.00%	0.38	0.31
	6	787	1.16%	2.72	0.56	1274	2.74%	1.93	12.05	1274	1.35%	2.53	14.78
	7	789	0.25%	2.74	12.14	1255	6.81%	16.66	17.32	1255	3.46%	26.15	14.93
	8	822	0.24%	507.72	41.87	1266	6.93%	1.62	4.97	1266	1.44%	1.38	7.14
	9	718	0.42%	3.51	12	1183	3.68%	1.83	6.45	1183	2.69%	1.83	7.78
	10	829	0.00%	2.03	46.01	1277	10.18%	62.53	28.94	1277	2.32%	46.9	30.03
E	Avg		0.24%	52.49	11.54		3.63%	8.84	7.55		1.33%	8.22	8.04
F	1	713	0.00%	3.51	3.74	1217	0.00%	0.81	0.24	1244	0.16%	2.68	3.41
	2	750	1.35%	0.79	0.65	1181	1.03%	1.11	0.26	1181	1.03%	0.43	0.28
	3	755	0.53%	3.05	1.19	1148	0.61%	1.59	0.35	1177	1.29%	2.39	5.62
	4	651	0.00%	0.37	0.46	1141	0.00%	0.65	0.57	1141	0.00%	0.7	0.45
	5	666	0.30%	1.01	0.76	1161	5.26%	3.25	0.73	1161	5.26%	0.72	0.94
	6	778	0.00%	0.45	0.68	1248	0.65%	3.3	3.5	1257	0.00%	2.17	8.33
	7	787	0.00%	3.78	17.21	1179	0.34%	1.4	0.8	1232	1.57%	17.54	24.94
	8	820	0.00%	1726.75	105.1	1195	0.93%	0.88	0.3	1248	0.00%	3.79	10.93
	9	715	0.00%	1.73	7.78	1145	0.35%	0.67	0.29	1156	0.35%	1.07	0.56
	10	829	0.00%	5.89	48.53	1188	2.50%	1.79	2.16	1261	1.04%	11.09	34.95
F	Avg		0.22%	174.73	18.61		1.17%	1.55	0.92		1.07%	4.26	9.04
		714	0.14%	1.6	1.42	1217	0.00%	0.51	0.26	1244	0.16%	6.57	3.09
		740	0.00%	0.57	0.3	1181	1.03%	0.7	0.22	1181	1.03%	0.65	0.24
		751	0.00%	0.62	0.38	1152	0.96%	1.36	0.29	1177	1.29%	0.49	0.37
		652	0.15%	0.55	0.42	1141	0.00%	1.04	0.57	1141	0.00%	0.65	0.43
		664	0.00%	1.37	0.71	1103	0.00%	0.76	0.3	1103	0.00%	0.67	0.28
		778	0.00%	0.7	0.71	1240	0.00%	1.62	0.42	1257	0.00%	2.13	6.16
		805	2.29%	4.5	25.5	1197	1.87%	1.4	0.57	1274	5.03%	9.97	21.54
		829	1.10%	1476.55	149.64	1232	4.05%	0.65	0.57	1267	1.52%	6.62	17.29
		715	0.00%	2.05	11.76	1141	0.00%	0.59	0.31	1171	1.65%	0.55	0.42
		829	0.00%	3.58	74.21	1183	2.07%	1.54	0.55	1289	3.29%	345.39	43.35
G	Avg		0.37%	149.21	26.51		1.00%	1.02	0.41		1.40%	37.37	9.32
				70.00	70.00			70.00	70.00			70.00	70.00
				146.57	23.97			3.24	2.10			9.80	8.13

R. Type	Inst.	CPMP				SS-CFLP				CARD+FIX			
		Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time
A	1	1006	0.00%	16.87	66.5	1819	0.00%	9	3.08	1835	0.00%	20.47	69.29
	2	966	0.00%	106.59	96.01	1848	0.00%	62.34	23.58	1848	0.00%	50.72	34.07
	3	1026	0.00%	15.53	16.48	1834	0.00%	6.24	2.75	1842	0.00%	6.82	31.51
	4	982	0.00%	341.49	326.52	1845	0.00%	52.06	26.16	1850	0.00%	194.37	63.89
	5	1091	0.00%	279.72	273.44	1876	0.00%	52.46	16.71	1897	0.00%	397.41	227.42
	6	954	0.00%	9.54	138.49	1732	0.00%	12.37	2.3	1740	0.00%	7.33	7.35
	7	1034	0.00%	91.62	153.59	1834	0.00%	53.56	21.59	1851	0.00%	1853.16	177.35
	8	1043	0.00%	698.39	304.22	1812	0.00%	36.58	10.61	1812	0.00%	27.26	38.58
	9	1031	0.00%	36.45	185.51	1854	0.00%	239.58	39	1854	0.00%	34.42	43.27
	10	1006	0.00%	-	-	1768	0.00%	81.51	22.82	1814	0.00%	-	338.87
A	Avg		0.00%	177.36	173.42		0.00%	60.57			0.00%	288.00	103.16
B	1	1006	0.00%	15.03	52.08	1819	0.00%	8.59	3.5	1835	0.00%	23.23	35.56
	2	966	0.00%	93.55	116.12	1848	0.00%	64.6	36.36	1848	0.00%	57.31	33.26
	3	1026	0.00%	13.15	15.46	1834	0.00%	1.79	2.72	1842	0.00%	8.24	40.27
	4	982	0.00%	334.76	315.12	1845	0.00%	83.9	30.31	1850	0.00%	354.32	65.39
	5	1091	0.00%	376.13	286.33	1886	0.53%	101.12	88.5	1897	0.00%	472.52	122.37
	6	954	0.00%	10.25	91.65	1732	0.00%	4.76	2.14	1740	0.00%	9.63	8.15
	7	1034	0.00%	101.9	79.09	1834	0.00%	45.79	38.04	1851	0.00%	1638.57	116.28
	8	1043	0.00%	865.7	151.23	1812	0.00%	17.07	33.63	1812	0.00%	18.09	30.43
	9	1031	0.00%	17.38	137.35	1854	0.00%	215.81	33.2	1854	0.00%	40.08	52.49
	10	1005	-0.10%	-	-	1768	0.00%	49.89	38.26	1814	0.00%	-	556.08
B	Avg		-0.01%	203.09	138.27		0.05%	59.33			0.00%	291.33	106.03
C	1	1077	7.06%	513.24	259.72	2014	10.72%	180.4	210.34	2014	9.75%	126.37	247.74
	2	968	0.21%	21.13	61.22	1861	0.70%	1604.47	81.65	1861	0.70%	1621.47	80.73
	3	1080	5.26%	13.46	36.99	1848	0.76%	2.41	3.08	1848	0.33%	4.5	2.56
	4	1036	5.50%	68.14	80.43	1911	3.58%	-	291.57	1911	3.30%	-	355.95
	5	1464	34.19%	1717.63	2346.76	2198	17.16%	19.5	183.63	2198	15.87%	21.29	138.42
	6	970	1.68%	17.26	51.63	1756	1.39%	5.61	2.71	1770	1.72%	74.25	95.81
	7	1034	0.00%	82.54	107.31	1861	1.47%	110.7	62.73	1861	0.54%	115.61	83.54
	8	1046	0.29%	212.37	135.45	1812	0.00%	4.33	17.81	1812	0.00%	4.42	19.22
	9	1077	4.46%	948.6	652.17	1930	4.10%	79.43	58.99	1930	4.10%	39.59	56.08
	10	1008	0.20%	-	-	1768	0.00%	5.86	4.94	1814	0.00%	1769.3	229.4
C	Avg		5.88%	399.37	554.42		3.99%	223.63			3.63%	419.64	130.95
D	1	1082	7.55%	29.59	135.19	1896	4.23%	178.24	136.26	1896	3.32%	201.72	81.01
	2	984	1.86%	23.16	29.04	1905	3.08%	253	33.79	1905	3.08%	285.22	30.02
	3	1046	1.95%	10.77	65.05	1935	5.51%	36.97	119.56	1935	5.05%	31.79	39.17
	4	1155	17.62%	-	664.03	1954	5.91%	-	630.64	1954	5.62%	-	589.41
	5	1137	4.22%	2553.9	581.63	1979	5.49%	100.41	55.76	1979	4.32%	74.97	86.89
	6	954	0.00%	3.48	33.01	1740	0.46%	39.95	4.23	1740	0.00%	13.34	7.73
	7	1060	2.51%	48.3	84.21	1899	3.54%	20.73	83.24	1899	2.59%	18.57	111.19
	8	1043	0.00%	653.97	154.21	1812	0.00%	39.03	11.73	1812	0.00%	47.46	28.93
	9	-	-	-	-	-	-	-	-	-	-	-	-
	10	1012	0.60%	-	1205.99	1847	4.47%	345.26	254.34	1847	1.82%	269.18	241.53
D	Avg		4.03%	474.74	328.04		3.63%	126.70			2.87%	117.78	135.10
E	1	1014	0.80%	14.82	31.35	1864	2.47%	19.08	4.54	1864	1.58%	20.05	5.47
	2	966	0.00%	29	20.61	1862	0.76%	25.79	19.2	1862	0.76%	24.83	13.08
	3	1029	0.29%	4.52	19.64	1842	0.44%	6.05	4.95	1842	0.00%	4.37	5.23
	4	990	0.81%	18.72	114.3	1867	1.19%	117.85	34.87	1867	0.92%	93.79	45.01
	5	1091	0.00%	43.82	127.41	1931	2.93%	38.73	68.86	1931	1.79%	82.88	107.76
	6	960	0.63%	19.58	66.91	1772	2.31%	13.12	27.05	1772	1.84%	22.59	20.44
	7	1056	2.13%	698.57	141.39	1869	1.91%	122.1	31.98	1869	0.97%	43.26	18.26
	8	1049	0.58%	12.6	6.24	1815	0.17%	4.21	3.38	1815	0.17%	4.08	3.97
	9	1036	0.48%	37.84	234.26	1907	2.86%	15.23	33.91	1907	2.86%	13.84	34.25
	10	1005	-0.10%	-	409.08	1840	4.07%	67.42	63.53	1840	1.43%	139.12	93.6
E	Avg		0.56%	97.72	117.12		1.91%	42.96			1.23%	44.88	34.71
F	1	1020	1.39%	3.05	39.05	1862	2.36%	35.51	5.34	1877	2.29%	17.84	30.62
	2	966	0.00%	32.83	53.94	1851	0.16%	21.54	32.64	1851	0.16%	21.36	34.95
	3	1032	0.58%	3.26	5.68	1839	0.27%	3.97	2.07	1849	0.38%	44.69	11.16
	4	988	0.61%	261.95	236.38	1851	0.33%	85	34.85	1874	1.30%	-	220.17
	5	1109	1.65%	152.23	109.61	1902	1.39%	3.88	1.87	1949	2.74%	108.74	138.02
	6	974	2.10%	17.45	86.05	1764	1.85%	20.81	2	1796	3.22%	17.42	25.82
	7	1036	0.19%	137.07	68.78	1834	0.00%	22.57	19.32	1851	0.00%	364.59	35.68
	8	1043	0.00%	10.53	80.73	1812	0.00%	5.76	3.33	1812	0.00%	3.88	2.73
	9	1036	0.48%	40.01	188.75	1886	1.73%	10.32	3.28	1899	2.43%	27.85	62.37
	10	1009	0.30%	-	1193.61	1803	1.98%	54.57	15.86	1854	2.21%	659.57	212.28
F	Avg		0.73%	73.15	206.26		1.01%	26.39			1.47%	140.66	77.38
G	1	1027	2.09%	69.35	135.44	1849	1.65%	6.38	3.51	1882	2.56%	21.53	52.14
	2	977	1.14%	233.75	121.98	1848	0.00%	23.22	4.93	1848	0.00%	21.26	3.47
	3	1043	1.66%	13.44	30.43	1867	1.80%	5.64	2.11	1875	1.79%	5.05	2.27
	4	982	0.00%	414.54	284.28	1847	0.11%	51.05	11.57	1869	1.03%	-	134.18
	5	1127	3.30%	520.23	126.18	1916	2.13%	42.98	57.01	1929	1.69%	9.73	125.11
	6	986	3.35%	12.28	110.19	1825	5.37%	9.77	1.78	1874	7.70%	51.44	36.14
	7	1037	0.29%	269.35	146.99	1834	0.00%	40.38	30.93	1901	2.70%	-	769.95
	8	1043	0.00%	445.19	307.44	1813	0.06%	6.8	3.08	1813	0.06%	3.1	20.01
	9	1037	0.58%	108.42	183.79	1863	0.49%	7.96	3.14	1910	3.02%	113.29	72.16
	10	1017	1.09%	-	1307.76	1768	0.00%	76.79	17.07	1814	0.00%	118.87	255.88
G	Avg		1.35%	231.84	275.45		1.16%	27.10	13.51		2.05%	43.03	147.13
				61.00	67.00			67.00	63.00			62.00	69.00
				236.75	256.14			80.95	13.51			192.19	104.92

		CPMP				SS-CFLP				CARD+FIX			
R. Type	Inst.	Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time
A	1	713	0.00%	2.79	6.16	1217	0.00%	0.33	0.28	1242	0.00%	7.19	2.87
	2	740	0.00%	0.9	0.42	1169	0.00%	0.61	0.26	1169	0.00%	0.56	0.25
	3	751	0.00%	1.72	1.39	1141	0.00%	1.42	0.4	1162	0.00%	0.81	0.47
	4	651	0.00%	0.91	0.59	1141	0.00%	1.28	0.44	1141	0.00%	0.67	0.47
	5	664	0.00%	1.07	0.81	1103	0.00%	3.73	0.89	1103	0.00%	0.43	0.52
	6	778	0.00%	0.58	0.67	1240	0.00%	1.43	0.44	1257	0.00%	6.36	9.49
	7	787	0.00%	4.09	17.17	1175	0.00%	1.8	0.67	1213	0.00%	1.74	4.69
	8	820	0.00%	1989.29	202.28	1184	0.00%	1.66	0.31	1248	0.00%	4.61	8.19
	9	715	0.00%	2.16	9.98	1141	0.00%	1.84	0.35	1152	0.00%	1.91	0.8
	10	829	0.00%	9.33	71.52	1159	0.00%	0.73	0.29	1248	0.00%	26.37	45.52
A	Avg		0.00%	201.28	31.10		0.00%	1.48	0.43		0.00%	5.07	7.33
B	1	713	0.00%	5.67	4.89	1217	0.00%	0.32	0.27	1242	0.00%	9.93	3.39
	2	740	0.00%	1	0.36	1169	0.00%	0.27	0.27	1169	0.00%	0.64	0.27
	3	751	0.00%	2.08	1.21	1150	0.79%	1.22	0.58	1162	0.00%	0.78	0.46
	4	651	0.00%	1.12	0.64	1141	0.00%	0.29	0.47	1141	0.00%	0.67	0.47
	5	664	0.00%	0.91	0.76	1103	0.00%	3.1	0.53	1103	0.00%	0.37	0.41
	6	778	0.00%	1.05	0.71	1240	0.00%	1.56	0.45	1257	0.00%	7.41	14
	7	787	0.00%	5.08	24.78	1175	0.00%	1.1	0.57	1213	0.00%	2.37	8.44
	8	821	0.12%	2142.42	203.07	1190	0.51%	0.48	0.47	1248	0.00%	2.91	7.85
	9	715	0.00%	1.77	15.73	1141	0.00%	0.89	0.37	1152	0.00%	1.96	0.7
	10	829	0.00%	9.95	83.41	1183	2.07%	0.95	0.59	1248	0.00%	32.15	47.76
B	Avg		0.01%	217.11	33.56		0.34%	1.02	0.46		0.00%	5.92	8.38
C	1	713	0.00%	1.79	2.05	1258	3.37%	1.13	0.52	1258	1.29%	0.76	0.44
	2	761	2.84%	0.81	0.51	1216	4.02%	1.57	1.77	1216	4.02%	2.26	0.55
	3	751	0.00%	1.99	1.3	1155	1.23%	0.95	0.53	1162	0.00%	0.74	0.46
	4	652	0.15%	0.82	0.5	1141	0.00%	0.5	0.5	1141	0.00%	0.7	0.47
	5	664	0.00%	1.28	0.94	1103	0.00%	2.37	0.51	1103	0.00%	0.34	0.35
	6	799	2.70%	1.02	0.93	1274	2.74%	7.18	13.41	1274	1.35%	3.5	10.85
	7	787	0.00%	8.03	29.17	1175	0.00%	0.71	0.43	1213	0.00%	2.21	4.88
	8	820	0.00%	1702.21	167.71	1195	0.93%	0.7	0.29	1248	0.00%	3.86	9.36
	9	715	0.00%	0.93	0.55	1152	0.96%	1.8	0.61	1152	0.00%	1.57	0.59
	10	835	0.72%	37.26	59.06	1183	2.07%	0.54	0.37	1248	0.00%	19.19	26.98
C	Avg		0.64%	175.61	26.27		1.53%	1.75	1.89		0.67%	3.51	5.49
D	1	744	4.35%	3.7	5.55	1270	4.35%	0.78	0.37	1270	2.25%	0.69	0.4
	2	770	4.05%	2.53	0.88	1209	3.42%	2.53	0.69	1209	3.42%	2.23	0.74
	3	755	0.53%	2.61	1.91	1186	3.94%	2.07	2.01	1186	2.07%	4.1	0.66
	4	651	0.00%	0.97	0.59	1163	1.93%	0.41	0.48	1165	2.10%	1.41	0.44
	5	664	0.00%	0.64	0.8	1103	0.00%	0.92	0.34	1103	0.00%	0.717	0.44
	6	778	0.00%	0.79	0.7	1240	0.00%	1.35	0.46	1257	0.00%	4.7	9.59
	7	845	7.37%	9.55	24.22	1315	11.91%	45.34	18.25	1315	8.41%	45.76	18.81
	8	820	0.00%	71.68	26.44	1285	8.53%	2.66	6.95	1285	2.96%	2.38	6.97
	9	717	0.28%	3.08	21.18	1146	0.44%	0.72	0.37	1165	1.13%	1.83	6.34
	10	848	2.29%	32.91	88.74	1217	5.00%	0.41	0.44	1253	0.40%	9.22	15.71
D	Avg		1.89%	12.85	17.10		3.95%	5.72	3.04		2.27%	7.30	6.01
E	1	714	0.14%	1.87	0.7	1266	4.03%	2.85	3.28	1266	1.93%	3.29	3.26
	2	740	0.00%	0.71	0.34	1170	0.09%	0.65	0.24	1170	0.09%	0.66	0.24
	3	751	0.00%	1.3	0.98	1162	1.84%	0.76	0.36	1162	0.00%	0.85	0.35
	4	652	0.15%	0.72	0.42	1141	0.00%	1.2	0.37	1141	0.00%	1.2	0.37
	5	664	0.00%	1.13	0.85	1103	0.00%	0.73	0.27	1103	0.00%	0.77	0.27
	6	787	1.16%	2.2	0.63	1274	2.74%	3.18	9.31	1274	1.35%	4.08	9.25
	7	789	0.25%	2.33	20.35	1255	6.81%	41.83	17.23	1255	3.46%	24.35	17.25
	8	822	0.24%	667.17	50.25	1266	6.93%	1.46	4.13	1266	1.44%	1.72	4.09
	9	718	0.42%	3.29	10.07	1183	3.68%	2.5	6.16	1183	2.69%	2.45	6.14
	10	829	0.00%	1.81	35.08	1277	10.18%	78.67	29.24	1277	2.32%	59.32	29.21
E	Avg		0.24%	68.25	11.97		3.63%	13.38	7.06		1.33%	9.87	7.04
F	1	713	0.00%	3.92	4.66	1217	0.00%	0.92	0.24	1244	0.16%	3.2	2.9
	2	750	1.35%	1.12	0.84	1181	1.03%	0.91	0.29	1181	1.03%	0.86	0.25
	3	755	0.53%	2.65	2.73	1148	0.61%	1.53	0.35	1177	1.29%	2.43	0.93
	4	651	0.00%	0.94	0.55	1141	0.00%	1	0.4	1141	0.00%	0.62	0.42
	5	666	0.30%	1.08	0.74	1161	5.26%	3.61	1.62	1161	5.26%	0.87	0.53
	6	778	0.00%	1.06	0.66	1248	0.65%	2.06	3.77	1257	0.00%	3.1	6.18
	7	787	0.00%	2.62	21.56	1179	0.34%	1.88	0.73	1232	1.57%	13.23	15.73
	8	820	0.00%	1606.42	87.3	1195	0.93%	0.76	0.28	1248	0.00%	2.97	10.85
	9	715	0.00%	2.41	10.14	1145	0.35%	1.04	0.27	1156	0.35%	1.55	0.52
	10	829	0.00%	5.16	45.19	1188	2.50%	1.81	3.12	1261	1.04%	19.63	28.56
F	Avg		0.22%	162.74	17.44		1.17%	1.55	1.11		1.07%	4.85	6.69
G	1	714	0.14%	2.19	1.85	1217	0.00%	0.71	0.28	1244	0.16%	15.44	2.97
	2	740	0.00%	0.73	0.37	1181	1.03%	0.93	0.23	1181	1.03%	1.38	0.22
	3	751	0.00%	0.81	0.51	1152	0.96%	0.98	0.27	1177	1.29%	0.78	0.38
	4	652	0.15%	1.27	0.55	1141	0.00%	1.15	0.53	1141	0.00%	0.83	0.43
	5	664	0.00%	1.17	0.74	1103	0.00%	0.44	0.29	1103	0.00%	0.602	0.26
	6	778	0.00%	1.02	0.59	1240	0.00%	1.14	0.42	1257	0.00%	4.12	5.58
	7	805	2.29%	7.89	23.76	1197	1.87%	1.24	0.55	1274	5.03%	17.1	17.58
	8	829	1.10%	742.11	99.29	1232	4.05%	0.7	0.48	1267	1.52%	8.97	13.21
	9	715	0.00%	3.91	9.69	1141	0.00%	0.71	0.31	1171	1.65%	0.93	0.48
	10	829	0.00%	3.39	33.45	1183	2.07%	1.36	0.44	1289	3.29%	289.07	39.83
G	Avg		0.37%	76.45	17.08		1.00%	0.94	0.38		1.40%	33.92	8.09
				70.00	70.00			70.00	70.00			70.00	70.00
				130.61	22.07			3.69	2.05			10.06	7.00

R. Type	Inst.	CPMP				SS-CFLP				CARD+FIX			
		Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time	Avg	cost increase	B&P Time	CPX Time
A	1	1006	0.00%	26.84	79.34	1819	0.00%	8.15	2.89	1835	0.00%	30.07	37.90
	2	966	0.00%	96.04	95.86	1848	0.00%	61.14	22.86	1848	0.00%	78.58	24.36
	3	1026	0.00%	13.88	27.59	1834	0.00%	5.12	2.48	1842	0.00%	19.80	16.23
	4	982	0.00%	234.13	210.62	1845	0.00%	121.26	36.39	1850	0.00%	418.73	88.07
	5	1091	0.00%	385.70	312.55	1876	0.00%	63.01	21.85	1897	0.00%	432.94	115.58
	6	954	0.00%	5.07	92.97	1732	0.00%	10.94	2.11	1740	0.00%	20.47	6.07
	7	1034	0.00%	125.05	112.10	1834	0.00%	60.19	15.77	1851	0.00%	1700.40	111.42
	8	1043	0.00%	801.15	225.63	1812	0.00%	20.07	20.68	1812	0.00%	38.43	27.34
	9	1031	0.00%	16.27	185.24	1854	0.00%	336.79	21.40	1854	0.00%	41.76	35.92
	10	1009	0.00%	-	-	1768	0.00%	193.46	22.68	1814	0.00%	-	563.39
A	Avg		0.00%	189.35	149.10		0.00%	88.01			0.00%	309.02	102.63
B	1	1006	0.00%	29.83	76.75	1819	0.00%	3.59	3.74	1835	0.00%	17.88	45.40
	2	966	0.00%	147.42	100.78	1848	0.00%	85.45	27.99	1848	0.00%	77.28	22.72
	3	1026	0.00%	17.24	34.53	1834	0.00%	1.80	2.37	1842	0.00%	12.88	12.95
	4	982	0.00%	318.39	302.26	1845	0.00%	83.60	28.99	1850	0.00%	322.68	154.24
	5	1091	0.00%	527.71	215.07	1886	0.53%	132.47	42.22	1897	0.00%	586.07	226.39
	6	954	0.00%	14.31	151.42	1732	0.00%	6.47	2.04	1740	0.00%	11.08	3.89
	7	1034	0.00%	110.62	90.89	1834	0.00%	45.29	35.37	1851	0.00%	2006.64	75.54
	8	1043	0.00%	1102.96	291.21	1812	0.00%	26.07	29.70	1812	0.00%	28.35	30.97
	9	1031	0.00%	17.77	232.00	1854	0.00%	450.44	39.76	1854	0.00%	31.56	38.95
	10	1005	-0.40%	-	-	1768	0.00%	173.77	33.10	1814	0.00%	-	254.55
B	Avg		-0.04%	254.03	166.10		0.05%	100.90			0.00%	343.82	86.56
C	1	1077	7.06%	488.07	396.71	2014	10.72%	223.26	165.67	2014	9.75%	197.89	206.83
	2	968	0.21%	37.93	57.88	1861	0.70%	1844.79	93.03	1861	0.70%	1782.91	69.62
	3	1080	5.26%	8.51	47.26	1848	0.76%	1.99	2.74	1848	0.33%	1.85	2.61
	4	1036	5.50%	60.25	61.63	1911	3.96%	-	286.15	1911	3.68%	-	125.93
	5	1464	34.19%	1229.49	2395.68	2198	17.16%	29.42	97.58	2198	15.87%	33.88	114.57
	6	970	1.68%	29.83	61.27	1756	1.39%	4.77	3.82	1770	1.72%	79.99	23.29
	7	1034	0.00%	83.06	145.29	1861	1.47%	116.95	66.91	1861	0.54%	122.32	54.17
	8	1046	0.29%	243.77	131.71	1812	0.00%	3.60	7.31	1812	0.00%	3.50	22.08
	9	1077	4.46%	544.75	503.58	1930	4.10%	149.92	67.29	1930	4.10%	59.19	47.88
	10	1008	-0.10%	-	2487.29	1768	0.00%	9.19	6.60	1814	0.00%	3076.97	195.62
C	Avg		5.85%	302.85	628.83		4.03%	264.88			3.67%	595.39	86.26
D	1	1082	7.55%	25.13	136.42	1896	4.23%	541.27	61.75	1896	3.32%	707.12	82.34
	2	984	1.86%	34.68	24.46	1905	3.08%	378.81	44.40	1905	3.08%	416.34	41.19
	3	1046	1.95%	5.96	47.97	1935	5.51%	69.47	61.30	1935	5.05%	71.78	40.46
	4	1155	19.25%	-	851.72	1954	5.91%	-	352.02	1954	5.62%	3661.00	690.02
	5	1137	4.22%	1169.39	1071.69	1979	5.49%	126.98	64.11	1979	4.32%	110.15	66.64
	6	954	0.00%	2.56	69.86	1740	0.46%	21.09	6.56	1740	0.00%	8.96	6.60
	7	1060	2.51%	34.98	77.52	1899	3.54%	15.92	103.92	1899	2.59%	16.49	92.62
	8	1043	0.00%	845.98	437.80	1812	0.00%	34.30	16.22	1812	0.00%	58.20	23.84
	9	-	-	-	-	-	-	-	-	-	-	-	-
	10	1012	0.30%	-	456.60	1847	4.47%	593.93	124.35	1847	1.82%	493.66	135.71
D	Avg		4.18%	302.67	352.67		3.63%	222.72			2.87%	615.97	131.05
E	1	1014	0.80%	23.62	251.21	1864	2.47%	21.16	21.86	1864	1.58%	21.69	21.95
	2	966	0.00%	32.14	23.74	1862	0.76%	57.28	12.95	1862	0.76%	47.48	13.09
	3	1029	0.29%	1.99	5.78	1842	0.44%	2.22	4.27	1842	0.00%	3.01	4.31
	4	990	0.81%	44.7	49.54	1867	1.19%	104.2	44.91	1867	0.92%	76.43	45.13
	5	1091	0.00%	61.7	97.65	1931	2.93%	79.51	75.74	1931	1.79%	118.32	76.25
	6	960	0.63%	48.44	131.38	1772	2.31%	32	16.93	1772	1.84%	38.6	16.92
	7	1056	2.13%	1235.52	143.33	1869	1.91%	152.4	16.6	1869	0.97%	84.82	16.63
	8	1049	0.58%	12.57	46.01	1815	0.17%	3.1	3.11	1815	0.17%	3.92	3.14
	9	1036	0.48%	41.59	234.74	1907	2.86%	14.12	33.79	1907	2.86%	19.01	33.89
	10	1005	-0.40%	3661	403.3	1840	4.07%	104.82	114.49	1840	1.43%	109.16	113.8
E	Avg		0.53%	516.33	138.67		1.91%	57.08			1.23%	52.24	34.51
F	1	1020	1.39%	2.59	34.27	1862	2.36%	56.29	3.94	1877	2.29%	22.18	38.03
	2	966	0.00%	64.55	56.57	1851	0.16%	69.81	24.15	1851	0.16%	22.87	22.43
	3	1032	0.58%	1.92	6.65	1839	0.27%	5.88	1.93	1849	0.38%	66.74	10.51
	4	988	0.61%	958.77	358.10	1851	0.33%	112.76	27.49	1874	1.30%	3661.00	273.69
	5	1109	1.65%	183.12	158.03	1902	1.39%	4.78	1.73	1949	2.74%	144.38	141.36
	6	974	2.10%	11.05	166.72	1764	1.85%	33.32	1.87	1796	3.22%	21.24	67.55
	7	1036	0.19%	133.31	66.98	1834	0.00%	35.01	20.13	1851	0.00%	606.29	78.48
	8	1043	0.00%	11.76	75.27	1812	0.00%	6.40	3.11	1812	0.00%	3.29	3.46
	9	1036	0.48%	18.02	178.43	1886	1.73%	28.03	4.31	1899	2.43%	23.82	62.28
	10	1009	0.00%	-	699.05	1803	1.98%	222.18	26.82	1854	2.21%	197.66	408.88
F	Avg		0.70%	153.90	180.01		1.01%	57.45			1.47%	476.95	110.67
G	1	1027	2.09%	93.48	109.17	1849	1.65%	7.21	3.51	1882	2.56%	11.83	61.79
	2	977	1.14%	267.20	167.44	1848	0.00%	51.77	8.04	1848	0.00%	31.49	7.07
	3	1043	1.66%	41.32	55.09	1867	1.80%	4.77	2.18	1875	1.79%	3.55	2.11
	4	982	0.00%	662.23	227.18	1847	0.11%	177.36	11.77	1869	1.08%	-	72.14
	5	1127	3.30%	500.26	124.00	1916	2.13%	59.01	40.03	1929	1.69%	8.32	46.28
	6	986	3.35%	18.83	88.46	1825	5.37%	31.98	1.53	1874	7.70%	42.17	28.24
	7	1037	0.29%	328.74	195.26	1834	0.00%	88.70	15.34	1901	2.70%	-	379.74
	8	1043	0.00%	1603.61	152.28	1813	0.06%	6.62	2.47	1813	0.06%	2.55	17.42
	9	1037	0.58%	88.21	142.41	1863	0.49%	6.44	2.43	1910	3.02%	179.88	87.22
	10	1017	0.79%	3661.00	1052.60	1768	0.00%	94.35	10.42	1814	0.00%	177.12	46.38
G	Avg		1.32%	726.49	231.39		1.16%	52.82	9.77		2.06%	57.11	74.84
				63.00	67.00			67.00	63.00			64.00	69.00
				349.37	263.82			120.55	9.77			350.07	89.50

References

- [1] K. Aardal. Capacitated facility location: Separation algorithms and computational experience. *Mathematical Programming*, 81:149–175, 1998.
- [2] P. Avella, A. Sassano, and I. Vasilév. Computational study of large scale p-median problems. *Optimization Online*, 2003. Technical Report.
- [3] R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, and A. Mingozzi. A new method for solving capacitated location problems based on a set partitioning approach. *Computers and Operations Research*, 29:365–386, 2002.
- [4] J.E. Beasley. A note on solving large p-median problems. *European Journal of Operational Research*, 21:270–273, 1985.
- [5] C. Beltran, C. Tadonki, and J.Vial. Solving the p-median problem with a semi-lagrangian relaxation. Technical report, University of Geneva, 2004.
- [6] A. Ceselli. *Algoritmi branch and bound e branch and price per il problema delle p-mediane con capacità*. PhD thesis, Dipartimento di Tecnologie dell’Informazione, Università degli Studi di Milano, Milano, IT, 2002. In Italian.
- [7] A. Ceselli. Two exact algorithms for the capacitated p-median problem. *4OR*, 1(4), 2003.
- [8] A. Ceselli and G. Righini. A branch-and-price algorithm for the capacitated p-median problem. *Networks*, In press, 2005.
- [9] N. Christofides and J.E. Beasley. A tree search algorithm for the p-median problem. *European Journal of Operational Research*, 10:196–204, 1981.
- [10] N. Christofides and J.E. Beasley. Extensions to a lagrangean relaxation approach for the capacitated warehouse location problem. *European Journal of Operational Research*, 12:19–28, 1983.
- [11] G. Cornuejols, M.L. Fisher, and G.L. Nemhauser. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.
- [12] J. Current, M. Daskin, and D. Schilling. *Discrete Network Location Models*, pages 81–118. Springer Verlag, Berlin, 2002.
- [13] J.A. Diaz and E. Fernández. Hybrid scatter search and path relinking for the capacitated p-median problem. *European journal of Operational Research*. In press.
- [14] J.A. Diaz and E. Fernández. A branch-and-price algorithm for the single source capacitated plant location problem. *Journal of the Operational Research Society*, 53:728–740, 2002.
- [15] R.D. Galvao. A dual-bounded algorithm for the p-median problem. *Operations Research*, 28(5), 1979.
- [16] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, 1961.

- [17] E. Gourdin, M. Labbé, and H. Yaman. *Telecommunication in location*, pages 275–305. Springer, Berlin, 2003.
- [18] P. Hanjoul and D. Peeters. A comparison of two dual-based procedures for solving the p-median problem. *European Journal of Operational Research*, 20:387–396, 1985.
- [19] M. Held, P. Wolfe, and H.P. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6:62–88, 1974.
- [20] K. Holmberg, M. Rönnqvist, and D. Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, 113:544–559, 1999.
- [21] O. Kariv and S.L. Hakimi. Reducibility among combinatorial problems. *SIAM Journal of Applied Mathematics*, 37:539–560, 1979.
- [22] J.G. Klincewicz and H. Luss. A lagrangean relaxation heuristic for capacitated facility location with single-source constraints. *Journal of the Operational Research Society*, 37(5):495–500, 1986.
- [23] A. Klose and A. Drexl. Facility location models for distribution system design. *European Journal of Operational Research*, 2004.
- [24] A. Klose and S. Görtz. *An exact column generation approach to the capacitated facility location problem*, volume 544 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, 2004.
- [25] M. Labbé, D. Peeters, and J.F. Thisse. *Location on Networks, in Network Routing*, volume 8. Elsevier Science B.V., 1995.
- [26] M. Labbé and H. Yaman. A note on the projection of polyhedra. 2003.
- [27] M. Labbé and H. Yaman. Polyhedral analysis for concentrator location problem. *Optimization Online*, 694, 2003. Optimization Online site : www.optimization-online.org.
- [28] L. Lorena and E. Senne. A column generation approach to capacitated p-median problems. *Computers and Operations Research*, 31(6):863–876, 2004.
- [29] S. Martello and P. Toth. *An algorithm for the generalized assignment problem*, pages 589–603. 1981.
- [30] S. Martello and P. Toth. *Knapsack problems: Algorithms and Computer Implementations*. Wiley, New York, 1990.
- [31] R.K. Martin. *Large scale linear and integer optimization*. Kluwer, 1999.
- [32] A.T. Murray and R.A. Gerrard. Capacitated service and regional constraints in location-allocation modeling. *Location Science*, 5(2):103–118, 1997.
- [33] S.C. Narula, U.I. Ogbu, and H.M. Samuelsson. An algorithm for the p-median problem. *Operations Research*, 25(4), 1977.

- [34] A.W. Neebe and M.R. Rao. An algorithm for the fixed charge assigning users to sources problem. *Journal of the Operational Research Society*, 34(11):1107–1113, 1983.
- [35] I.H. Osman and N. Christofides. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 13:317–336, 1994.
- [36] H. Pirkul. Efficient algorithms for the capacitated concentrator location problem. *Computers and Operations Research*, 14(3):197–208, 1987.
- [37] K.E. Rosling. Towards the solutions of the (generalised) multi-weber problem. *Environment and Planning, Series B*, 18:347–360, 1991.
- [38] T.J. Van Roy and D. Erlenkotter. Dual-based procedure for dynamic facility location. *Management Science*, 28(10), 1982.
- [39] Van Roy and D. Erlenkotter. A cross decomposition algorithm for capacitated facility location. *Operations Research*, 34:145–163, 1986.
- [40] M. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45(6), 1997.
- [41] E.L.F. Senne, L.A.N. Lorena, and M.A. Pereira. A branch-and-price approach to p-median location problems. *Computers and Operations Research*, 32(6):1655–1664, 2005.
- [42] S.S. Syam. A model for the capacitated p-facility location problem in global environments. *Computers Ops Res.*, 24(11):1005–1016, 1997.