CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

The Capacity and Distance Constrained Plant Location Problem

Maria Albareda-Sambola
Elena Fernández
Gilbert Laporte

January 2007

CIRRELT-2007-01

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

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

The Capacity and Distance Constrained Plant Location Problem

Maria Albareda-Sambola[1], Elena Fernández[1] , Gilbert Laporte[2,*]

[1] Departament Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Pau Gargallo, 5, 08028, Barcelona, Spain

[2] Canada Research Chair in Distribution Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7, and Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Abstract. This article introduces a new problem called the *Capacity and Distance Constrained Plant Location Problem*. It is an extension of the discrete capacitated plant location problem, where the customers assigned to each plant have to be packed in groups that will be served by one vehicle each. The paper addresses different modeling aspects of the problem. It describes a tabu search algorithm for its solution. Extensive computational tests indicate that the proposed heuristic consistently yields optimal or near-optimal solutions.

Keywords. Location-allocation, bin packing, tabu search.

Acknowledgements. This work was partially supported by CICYT grant TIC2003-05982-C05-04, by NSERC grant 39682-05 and by CAM grant UC3M-MTM-05-075. This support is gratefully acknowledged.

---

* Corresponding author: gilbert.laporte@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2007

Introduction

One of the most determining strategic decisions in logistics concerns the location of facilities. Optimization problems for this type of decisions have been extensively studied, and a wide variety of problem specifications are now covered by the existing literature. In an important group of variants of such problems, the locations of facilities have to be chosen from a given finite set. These are known as discrete location problems (Mirchandani and Francis, 1990; Daskin, 1995), as opposed to network or continuous location problems. As our knowledge of the basic discrete location problems improves, more attention is being paid to sophisticated versions that are closer to the needs of the real world. Thus, many authors have worked on problems that combine locational decisions with vehicle routing which are central to the design of logistic systems. Most of the advances made in this area have been summarized in the survey papers Laporte (1988), Berman et al. (1995), Min et al. (1998), and more recently Nagy and Salhi (2007).

The introduction of fleet management and routing decisions into location problems gives rise to an important increase in the difficulty of these problems. In this paper, we present a new problem that is halfway between pure discrete location and combined location-routing: the *Capacity and Distance Constrained Plant Location Problem* (CDCPLP). This problem captures some of the intricacies of routing decisions in location problems, but avoids some of the sources of complexity of classical combined location-routing problems.

As in the *Single Source Capacitated Plant Location Problem* (SSCPLP), customers are served from capacitated plants selected from a given candidate set. In addition, in the CDCPLP, each open plant houses a number of identical vehicles that will actually provide the service. It is assumed that customers are served by full return trips from the plant, but the same vehicle can be used for several services as long as its workload does not exceed a prespecified total driving distance. The goal is to select the set of plants to open, determine the number of vehicles needed at each open plant, and assign each customer to a plant and a vehicle, while ensuring that assignments are feasible both with respect to plant capacities and vehicle distance constraints and the total cost, which includes fixed costs for opening plants, fixed vehicle utilization costs and assignment costs, is minimized. We assume that all costs relate to the same planning horizon (one day, say).

We present several integer programming formulations for the CDCPLP and show how this problem relates to several well known combinatorial optimization problems. These include location-allocation, vehicle routing, assignment, and bin packing. The problem is clearly $\mathcal{NP}$-hard, since it contains the SSCPLP as a particular case. In fact, we will show that

general purpose methods fail to solve even small size instances within a reasonable CPU time. The success of tabu search (TS) (Glover, 1989, 1990) on problems related to the CDCPLP (Gendreau et al., 1994; Scholl et al., 1997; Delmaire et al., 1999; Díaz and Fernández, 2001; Albareda-Sambola et al., 2005) have lead us to design and implement a TS based algorithm for this new problem. Since in the case of the CDCPLP the different types of decisions to take are strongly hierarchized, we have designed a TS heuristic that respects the underlying hierarchy, as was done in Albareda-Sambola et al. (2005) for a combined location-routing problem.

The remainder of this paper is organized as follows. In Section 1 we propose a variety of models and a relaxation of the CDCPLP. The different models allow us to relate the CDCPLP to other problems. In Section 2 we describe two proposed heuristics for this problem: a constructive method, and a TS improvement heuristic. Computational results are presented in Section 3. We present our conclusions in the last section.

1 Modelling Issues

Some notation needs to be introduced in order to formally state the CDCPLP. We are given a set J of potential plant locations and a set I of customers. We associate with each plant location $j \in J$ a fixed opening cost f_j, and a capacity b_j. Customer service is provided from the open plants by an homogeneous fleet of vehicles. Each vehicle has a fixed utilization cost g and a maximum (daily) total driving distance ℓ. Servicing customer $i \in I$ from plant $j \in J$ generates a driving distance t_{ij} for the vehicle performing the service, consumes a quantity d_i of the capacitated resource of plant j, and has an associated cost c_{ij}. The vehicles available at the plants are indexed in K, and we will denote by $\bar{k}$ the upper bound on the number of vehicles at any plant. We define binary variables y_j to indicate whether a plant is opened at site j or not, z_{jk} to indicate whether a k^{th} vehicle is assigned to plant j, and x_{ijk} to indicate whether customer i is served by the k^{th} vehicle of plant j.

The problem can then be formulated as follows:

$$\text{(P) Minimize} \quad \sum_{j\in J} f_j y_j + g \sum_{j\in J}\sum_{k\in K} z_{jk} + \sum_{i\in I}\sum_{j\in J} c_{ij} \sum_{k\in K} x_{ijk} \tag{1}$$

$$\text{subject to} \quad \sum_{j\in J}\sum_{k\in K} x_{ijk} = 1 \qquad i \in I, \tag{2}$$

$$\sum_{i\in I} t_{ij} x_{ijk} \leqslant \ell z_{jk} \qquad j \in J, k \in K, \tag{3}$$

$$\sum_{i\in I}\sum_{k\in K} d_i x_{ijk} \leqslant b_j y_j \qquad j \in J, \tag{4}$$

$$z_{jk} \leqslant y_j \qquad j \in J, k \in K, \tag{5}$$

$$x_{ijk} \leqslant z_{jk} \qquad i \in I, j \in J, k \in K, \tag{6}$$

$$z_{jk} \leqslant z_{j,k-1} \qquad j \in J, k \in K\backslash\{1\}, \tag{7}$$

$$x_{ijk}, y_j, z_{jk} \in \{0,1\} \qquad i \in I, j \in J, k \in K. \tag{8}$$

Here, constraints (2) ensure that each customer is served. The driving distances limits and plant capacities are defined by constraints (3) and (4), respectively. Constraints (5) and (6) ensure that a customer cannot be served from a plant that has not been opened, nor by a vehicle that has not been allocated to the corresponding plant. Finally, constraints (7) ensure that vehicle k will not be used before vehicle $k-1$. These constraints avoid awkward symmetries in the set of feasible solutions and multiple representations of the same solution. In fact, other constraints can be added to this end, like, for example:

$$\sum_{i\in I} t_{ij} x_{ijk} \leqslant \sum_{i\in I} t_{ij} x_{ij,k-1} \quad j \in J, k > 1, \tag{9}$$

i.e., in a given feasible solution, vehicles based at the same plant are ordered by non-increasing total travel distances. Since vehicles are identical, such a constraint does not eliminate the optimum, but reduces the number of feasible vectors.

The CDCPLP is $\mathcal{NP}$-hard because the SSCPLP, which is known to be $\mathcal{NP}$-hard, corresponds to the particular case of (P) in which $g = 0$, $L = 1$ and $t_{ij} = 1\ \forall i \in I, j \in J$.

1.1 Bilevel model

In a first alternative model for this problem we consider as the main decisions the location of plants and the assignment of customers to plants. The objective is to use the minimum required vehicles to satisfy the customer demands as they have been assigned. To this end, we use the following binary variables:

y_j equal to 1 if and only if a plant is opened at site j or not,

s_{ij} equal to 1 if and only if customer i is served from plant j,

z_{jk} equal to 1 if and only if a k^{th} vehicle is assigned to plant j, and

x_{ijk} equal to 1 if and only if customer i is served by the k^{th} vehicle of plant j.

These variables allow us to model the CDCPLP in as follows:

$$\text{(BP) Minimize } \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J} c_{ij}s_{ij} + A(y,s) \tag{10}$$

$$\text{subject to } \sum_{j\in J} s_{ij} = 1 \qquad \forall i \in I, \tag{11}$$

$$\sum_{i\in I} t_{ij}s_{ij} \leqslant \bar{k}\ell \qquad \forall j \in J, \tag{12}$$

$$\sum_{i\in I} d_i s_{ij} \leqslant b_j y_j \qquad \forall j \in J, \tag{13}$$

$$s_{ij} \leqslant y_j \qquad \forall i \in I, j \in J, \tag{14}$$

$$s_{ij}, y_j \in \{0,1\} \qquad \forall i \in I, j \in J, \tag{15}$$

where

$$A(y,s) = \min\ g\sum_{j\in J}\sum_{k\in K} z_{jk} \tag{16}$$

$$\text{subject to } \sum_{k\in K} x_{ijk} = s_{ij} \qquad \forall i \in I, j \in J, \tag{17}$$

$$\sum_{i\in I} t_{ij}x_{ijk} \leqslant \ell z_{jk} \qquad \forall j \in J, k \in K, \tag{18}$$

$$x_{ijk} \leqslant z_{jk} \qquad \forall i \in I, j \in J, k \in K, \tag{19}$$

$$z_{jk} \leqslant y_j \qquad \forall j \in J, k \in K, \tag{20}$$

$$x_{ijk}, z_{jk} \in \{0,1\} \qquad \forall i \in I, j \in J, k \in K. \tag{21}$$

Observe that $A(y,s)$ can be separated in $|J|$ *Bin Packing Problems*. The function $A(y,s)$ is, in fact, independent of the y vector, since, for feasible solutions of the first level problem, its last set of constraints can be replaced by

$$z_{jk} \leqslant \sum_{i\in I} s_{ij} \quad \forall i \in I, j \in J, k \in K.$$

1.2 Relaxed model

To generate good lower bounds, we relax the problem in the following manner. We allow different vehicles of the same plant to share the travel load to a given customer as if, instead of having k vehicles on a plant, we had a single vehicle capable of driving k times the distance limit. To model this relaxation it is only necessary to relax vehicle distance constraints (3) of (P) and include the surrogate aggregated capacity constraint for all vehicles in the same plant. We can rewrite the resulting model by means of the following binary variables:

y_{jk} with value 1 if plant j is used with exactly k vehicles, and 0 otherwise,

x_{ij} with value 1 if customer i is assigned to plant j.

Using $f_{jk} = f_j + kg$ to denote the cost for opening plant j with k vehicles, the resulting model is:

$$\text{(RP) Minimize} \quad \sum_{j\in J}\sum_{k\in K} f_{jk}y_{jk} + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \tag{22}$$

$$\text{subject to} \quad \sum_{j\in J} x_{ij} = 1 \qquad \forall i \in I, \tag{23}$$

$$\sum_{i\in I} t_{ij}x_{ij} \leqslant \ell \sum_{k\in K} k\, y_{jk} \qquad \forall j \in J, \tag{24}$$

$$\sum_{i\in I} d_i x_{ij} \leqslant b_j \sum_{k\in K} y_{jk} \qquad \forall j \in J, \tag{25}$$

$$\sum_{k\in K} y_{jk} \leqslant 1 \qquad \forall j \in J, \tag{26}$$

$$x_{ij} \leqslant \sum_{k\in K} y_{jk} \qquad \forall i \in I, j \in J, \tag{27}$$

$$x_{ij}, y_{jk} \in \{0,1\} \qquad \forall i \in I, j \in J, k \in K. \tag{28}$$

Constraints (23) ensure that all customers are assigned. The surrogate distance constraints are given by (24), while constraints (25) ensure that plant capacities are respected. Constraints (26) avoid making more than one choice on the number of vehicles for plant j, and finally, constraints (27) forbid assignments to plants that are not open.

Proposition 1. *The linear relaxation of (RP) has the same optimal value as the linear relaxation of (P), reinforced with the constraints*

$$\sum_{k\in K} x_{ijk} \leqslant y_j, \qquad \forall i, j, \tag{29}$$

and

$$y_j \leqslant \sum_{k \in K} z_{jk} \qquad \forall j. \tag{30}$$

Proof. First observe that constraints (29) are valid for (P), since if customer i is served from plant j, it is only served from one of the routes of the plant. Inequalities (30) are dominance constraints since in any optimal solution every open plant will use, at least, one vehicle. We will see how optimal solutions of these problems can be recoded into either format. Denote by (LP) the linear relaxation of (P) reinforced with the above stated constraints.

1.- Given an optimal solution of (LP) (y, z, x) satisfying (29) and (30), the vectors (x_{ij}) and (y_{jk}) with components

$$x_{ij} = \sum_{k \in K} x_{ijk}, \quad i \in I, j \in J,$$

$$y_{jk} = \begin{cases} \left(\bar{k} y_j - \sum\limits_{s \in K} z_{js} \right) \Big/ (\bar{k} - 1) & \text{if } k = 1, j \in J, \\ 0 & \text{if } 1 < k < \bar{k}, j \in J, \\ \left(\sum\limits_{s \in K} z_{js} - y_j \right) \Big/ (\bar{k} - 1) & \text{if } k = \bar{k}, j \in J, \end{cases}$$

provide a feasible solution for the linear relaxation of the relaxed model, (LRP) with value $\sum_{j \in J} f_j y_j + g \sum_{j \in J} \sum_{k \in K} z_{jk}$.

2.- Given a feasible solution of (LRP) $((y_{jk}), (x_{ij}))$,we can build a solution for (LP) as follows:

$$y_j = \sum_{k \in K} y_{jk}, \quad j \in J, \tag{31}$$

$$z_{jk} = \sum_{s=k}^{\bar{k}} y_{js}, \quad j \in J, k \in K. \tag{32}$$

Note that constraint (4) will be satisfied. Any feasible solution of (LRP) satisfies the length and capacity conditions: $x_{ij} \leqslant \sum_{k \in K} y_{jk}$ and $\sum_{i \in I} d_i x_{ij} \leqslant b_j \sum_{k \in K} y_{jk}$. Therefore, if we take x_{ijk} such that $\sum_{k \in K} x_{ijk} = x_{ij}$, then:

$$\begin{aligned}\sum_{i\in I}\sum_{k\in K} d_i x_{ijk} &= \sum_{i\in I} d_i \sum_{k\in K} x_{ijk} \\ &= \sum_{i\in I} d_i x_{ij} \\ &\leqslant b_j \sum_{k\in K} y_{jk} \\ &= b_j y_j.\end{aligned}$$

By construction, (29) will also hold. It is therefore sufficient to prove that, for a given j, there exists a set of values x_{ijk} with $\sum_{k\in K} x_{ijk} = x_{ij}$ and $x_{ijk} \leqslant z_{jk}$ for which (3) holds. It is easy to check that the vector with components $x_{ijk} = z_{jk} x_{ij} / \sum_{k\in K} k y_{jk}$ satisfies the above conditions.

□

1.3 Valid inequalities for (PR)

In this subsection we present four different types of valid inequalities for (PR).

Define ℓ_t as the value of the ratio between the sum of the distances from each customer to its closest plant and the maximum distance a vehicle can drive, rounded to the next integer, i.e., $\ell_t = \left\lceil \sum_{i\in I} \min_{j\in J} \left\{ t_{ij}/\ell \right\} \right\rceil$. It follows from constraints (24) that ℓ_t is a lower bound on the total number of routes in a feasible solution. Thus, the inequality $\sum_{j\in J}\sum_{k\in K} k y_{jk} \geq \ell_t$ is valid.

Analogously, let ℓ_d be the value of the ratio between the aggregated demand and the maximum capacity of a plant rounded to the next integer, i.e., $\ell_d = \left\lceil \sum_{i\in I} d_i \big/ \max_{j\in J} \{b_j\} \right\rceil$. From constraints (25) it follows that ℓ_d is a lower bound on the number of plants to be open. Thus, the inequality $\sum_{j\in J}\sum_{k\in K} y_{jk} \geq \ell_d$ is valid.

From constraints (24) we can also derive the following inequalities. For a given plant $\hat{\jmath} \in J$, and a subset of customers $C_1 \subset I$, let $\ell_t(C_1) = \left\lceil \sum_{i\in C_1} t_{ij}/\ell \right\rceil$ denote the minimum number of routes needed to service all clients indexed in C_1 from plant $\hat{\jmath}$. The bound $\ell_t(C_1)$ can be established using the following valid inequality:

$$\sum_{i\in C_1} x_{i\hat{\jmath}} \leq (|C_1| - 1) \sum_{s=1}^{\ell_t(C_1)-1} y_{\hat{\jmath}s} + |C_1| \sum_{s=\ell_t(C_1)}^{|K|} y_{\hat{\jmath}s}.$$

For a given plant $\hat{\jmath} \in J$ and any set of customers $C_2 \subset I$ with a total demand larger than the capacity of plant $\hat{\jmath}$ ($\sum_{i \in C_2} d_i > b_{\hat{\jmath}}$) we can derive the following valid inequality from constraints (25):

$$\sum_{i \in C_2} x_{i\hat{\jmath}} \leq (|C_2| - 1) \sum_{k \in K} y_{\hat{\jmath}k}.$$

2 Algorithms

In this section we describe the two methods we have developed to solve the CDCPLP. We have first designed a simple constructive heuristic that decomposes the problem into two sub-problems; first choose the set of plants to open, and then assign customers to them allocating vehicles as needed. We have also implemented a TS based heuristic with three levels of search, according to the structure of the problem. We now describe both algorithms in detail.

2.1 Constructive heuristic

To construct an initial feasible solution, we divide the CDCPLP into two separate problems, namely, the choice of the set of plants to open, and the assignment of customers to plants and allocation of vehicles.

First, we choose the set of plants to open. To do so, we first compute an estimate of the number of plants $\hat{m}$ that must be opened, and we then select a candidate set of $\hat{m}$ plants, taking into account their characteristics. As shown in Section 1.3, ℓ_d is a lower bound on the number of open plants necessary to satisfy all customer demand. Since demands are unsplittable, and there are also constraints on the vehicles total driven distance, this bound can be rather loose. In this algorithm we compute $\hat{m}$ as $\hat{m} = \beta \ell_d$, where the factor β depends on the value of the demands relative to the capacities. A good range for β is the interval $(1, 1.5)$. If demands are much smaller than capacities, the plant capacities are expected to be used almost completely, whereas, if they are large, a considerable fraction of the capacity of the open plants is likely to be unused. Once the value of $\hat{m}$ has been fixed, plants are taken in non-decreasing order of the weights

$$(f_j / I_d(j))\,(g / I_t(j))\,\bar{c}_j,$$

where $I_d(j)$ is the maximum number of customers whose total demand does not exceed the capacity of plant j, $I_t(j)$ is the maximum number of customers whose total travel distance to plant j is smaller than or equal to ℓ, and $\bar{c}_j$ is the average assignment cost of the different customers to plant j.

Once the set of open plants is fixed, customers are taken one at a time, and assigned to the open plant with sufficient residual capacity that has the smallest value of the auxiliary costs $c_{ij}t_{ij}$. If no feasible assignment exists, then the customer is assigned to the plant with the largest residual capacity to incur the smallest violation possible, and to the vehicle with the largest actual load, among those that could serve it without violating their total driving distance constraint. If no such vehicle exists, then a new vehicle is allocated to the plant and the customer is assigned to it, unless the plant has already $\bar{k}$ vehicles, in which case, the vehicle with the smallest travel distance assigned is chosen.

The outcome of the assignment-allocation phase strongly depends on the order in which customers are processed. To determine this order, we proceed as follows. For each customer $i \in I$ three values are considered: the ratio between the average assignment cost of the customer to the open plants and the average assignment cost to open plants, the ratio between its demand and the average demand, and the ratio between the average distance from the customer to the open plants and the overall average distance from customers to open plants:

$$\bar{c}_i = \sum_{j \in \hat{J}} c_{ij} \Big/ |\hat{J}|\bar{c}, \quad \bar{d}_i = d_i/\bar{d} \quad \text{and} \quad \bar{t}_i = \sum_{j \in \hat{J}} t_{ij} \Big/ |\hat{J}|\bar{t},$$

where $\hat{J}$ is the set of open plants, $\bar{c}$ is the average of all assignment costs to open plants, $\bar{d}$ is the average customer demand, and $\bar{t}$ is the average travel distance of customers to the open plants.

A single list is built with the $3|I|$ values, and it is sorted in non-increasing order. One value is taken at a time, according to this order and the corresponding customer is considered if it was not yet assigned.

2.2 Nested tabu search algorithm

When solving a CDCPLP, a series of decisions of fairly different natures have to be taken. The strong interrelationships between these decisions (location, assignment, and packing) make it important to take all them together to obtain the best solution possible. Nevertheless, there is a clear hierarchy in these decisions, which should be reflected in an algorithm designed for this problem.

In this section we present a TS improvement heuristic. A variety of neighborhoods with different purposes are used within the algorithm. While some of them focus on the selection of an appropriate set of plants to open, others focus on determining a good assignment of customers to plants for a given such set, and the remaining neighborhoods are designed to improve the packing of customers into vehicle routes.

More precisely, the neighborhoods "Empty a plant" (N_{ep}), "Open a new plant" (N_{op}) (that also assigns some customers to it) and "Interchange an open plant by a closed one" (N_{oc}) affect the set of open plants, while the neighborhoods "Reassign a customer to another open plant" (N_{cp}) , "Interchange two customers" (N_{ic}) and "Transfer a complete route to another plant" (N_{tr}) that reassigns one vehicle (and all its customers) to a different plant, affect the assignment of customers to plants. Finally, the neighborhoods "Reassign a customer to another route" (N_{cr}), "Split a route into two" (N_{sr}) and "Merge two routes" (N_{mr}) only affect the packing of customers into a vehicle within a specified plant.

The inclusion of so many different neighborhoods within one single search algorithm suggests several search strategies. Computational experiments have been conducted to compare different alternatives, which have led us to a nested TS structure. The search structure contains three levels. At the innermost level, we explore the neighborhoods that affect the packing of customers into vehicles (N_{cr}, N_{sr}, and N_{mr}). At the intermediate level we explore the neighborhoods that affect the assignment of customers to plants (N_{cp}, N_{ic} and N_{tr}), whereas at the outermost level the neighborhoods affect the set of open plants (N_{op}, N_{ep} and N_{oc}).

As first proposed in Gendreau et al. (1994) for a vehicle routing problem, infeasibilities are allowed throughout the search, both with respect to distance constraints and to capacity constraints. Violations of plant capacity and vehicle distance constraints are penalized with weights P_p and P_v, respectively. As in Díaz and Fernández (2001) these weights are periodically updated according to the expressions $P_p = P_p\alpha^{\beta_p}$ and $P_v = P_v\alpha^{\beta_v}$, where the common parameter α reflects the quality of the solutions found previously and parameters β_p and β_v reflect the infeasibilities of either type (capacity or distance constraint) that have been encountered recently. The technical details about the settings used for these parameters are given in Section 3. At each level of the search process, the infeasibilities encountered in the last iteration play an important role in the selection of the neighborhood to explore.

At the outermost level, the search is conducted using a best improve strategy. If the current solution is feasible with respect to plant capacity, the three neighborhoods N_{op}, N_{ep} and N_{oc} are explored, otherwise only solutions in N_{op} or N_{oc} are considered. Moreover, in the neighborhood N_{oc} a given interchange is not considered if it leads to a configuration where the total capacity of the set of open plants is smaller than the aggregated demand. Opening a recently closed plant is set tabu for a number of iterations, and vice versa.

At the intermediate level, where the assignment of customers to plants is to be improved, neighborhoods N_{cp}, N_{ic} and N_{tr} are completely explored, and the best improving move is selected. A customer-plant assignment is kept in the tabu list for a random number of

iterations after it has been modified; a solution in N_{tr} is considered tabu if at least half of the new assignments are forbidden.

Finally, at the innermost level, the neighborhoods N_{cr}, N_{sr} and N_{mr} are considered. In this case, the search only affects the customers currently assigned to a given plant. The search is performed on the plants that have been modified at the other levels of search. During the search, neighborhoods N_{sr} and N_{mr} are never explored in iterations that are close to each other; after performing a move in N_{sr}, N_{mr} is considered tabu for a number of iterations, and vice versa. Regarding the moves in N_{cr}, once a customer is reassigned, it is forbidden to move it again for a number of iterations. If the current packing is feasible with respect to the distance constraint, then solutions in N_{mr} or in N_{cr} are considered. Otherwise, if there is at least one vehicle with an assigned distance that exceeds the limit, neighborhood N_{sr} or N_{cr} is explored. In both cases, the choice of the neighborhood depends on the tabu list.

The structure of the search is outlined in Algorithm 1. In the updates of the best known solution, we consider that a new solution is better than the incumbent when they are both feasible and the new one has a smaller cost, when the incumbent is infeasible and the new one is feasible, or when, being both infeasible, neither the total plant capacity violation nor the total vehicle travel distance excess are larger in the new solution than in the incumbent, and at least one is smaller.

Details on the termination criteria of the three levels are given in Section 3.

Algorithm 1 TS heuristic search structure

```
Initialize solution x  # might violate length and/or capacity constraints #
Initialize x*  # best known solution #
while no stop criterion 1 do  # plant-level search #
  if x is plant feasible then
    Explore non-tabu moves in N_op, N_ep and N_oc
  else
    Explore non-tabu moves in N_op and N_oc
  end if
  Perform selected move
  Update tabu list 1 and x*, if applicable
  while no stop criterion 2 do  # assignment-level search #
    Explore non-tabu(2) moves in N_cp, N_ic and N_tr
    Perform the chosen move. Let j ∈ J be the affected plant
    Update tabu list 2, and x*, if applicable
    while no stop criterion 3 do  # packing-level search at plant j #
      if x is vehicle-feasible at plant j then
        if merge is not tabu then
          Explore N_mr
        else
          Explore non-tabu(3) moves in N_cr
        end if
      else
        if split is not tabu then
          Explore N_sr
        else
          Explore non-tabu(3) moves in N_cr
        end if
        Perform chosen move
        Update tabu list 3 and x*, if applicable
      end if
      Update TS-3 parameters
    end while
    Update TS-2 parameters
  end while
  Update TS-1 parameters
end while
```

3 Computational Experiments

A series of computational experiments were carried out to assess the quality of our proposed lower and upper bounds. The algorithms described in the previous section were coded in C, compiled with Microsoft Visual C 6.0, and run on a PC with a Pentium IV processor at 2.4MHz. To evaluate the quality of the results we have applied CPLEX 10.0 to all instances.

We have generated 91 CDCPLP instances as extensions of the SSCPLP ones used in Barceló, Fernández, and Jörnsten, 1991, available at `www-eio.upc.es/~elena/sscplp/`. The sizes $|J| \times |I|$ of the considered instances, p_1 to p_{25}, are the following: 10×20 for p_1 to p_6, 15×30 for p_7 to p_{17}, and 20×40 for p_{18} to p_{25}. To complete the SSCPLP instances to CDCPLP instances, it is necessary to generate all vehicle data: their fixed cost g, the travel distances t_{ij}, and the total distance limit ℓ.

For the smallest instances, we have generated 12 variants of each, divided into two groups. The first group constists of six instances where assignment costs and travel distances are uncorrelated (*uncorrelated instances*), the second group contains another six instances with correlated assignment costs and travel distances (*correlated instances*). Within each group, we have taken different combinations of the vehicle utilisation cost, g, and the maximum total driving distance, ℓ. The considered combinations of ℓ and g, and the corresponding instance labels are depicted in Table 1.

Label	A	B	C	D	E	F
ℓ	40	40	50	50	100	100
g	50	100	80	150	150	300

Table 1: Considered combinations of ℓ and g

For the uncorrelated instances, values t_{ij} have been randomly taken from the interval $[10, 50]$. In the case of the correlated instances, t_{ij} have been computed as the assignment costs scaled so as to fall in the interval $[15, 45]$ plus a random noise taken from $[-5, 5]$. Since the considered SSCFLPs have very tight capacity constraints and CDCPLP has more constraints than SSCFLP, we have increased the plant capacities by a factor of 1.5.

We solved the small instances using model P enhanced with constraints (9), and the aggregated relaxation using model RP, with CPLEX 10.0. In both cases, we have guided the search of the branch-and-bound tree by setting the CPLEX MIP emphasis parameter to "optimality" which proved to give the best results in our preliminary computational experiments.

The obtained results are sumarized in Table 2. In both cases, columns under the heading

		Uncorrelated Instances					Correlated Instances				
Instance	Label	*opt*	*time*	*lb*	*time lb*	*dev*	*opt*	*time*	*lb*	*time lb*	*dev*
p1	A	2126	7604	2075	80	2.40	1789	111	1779	10	0.56
	B	2656	594	2592	19	2.41	2375	292	2329	15	1.94
	C	2288	3885	2226	11	2.71	1976	666	1976	19	0.00
	D	2918	4005	2881	71	1.27	2606	400	2606	37	0.00
	E	2176	26	2176	2	0.00	1996	38	1996	5	0.00
	F	2926	39	2926	3	0.00	2746	73	2746	10	0.00
p2	A	3561	192	3494	440	1.88	2807	179	2807	9	0.00
	B	4146	645	4064	182	1.98	3307	42	3307	12	0.00
	C	3754	2884	3692	658	1.65	3055	2129	2990	3	2.13
	D	4363	2269	4288	476	1.72	3685	2251	3620	15	1.76
	E	3700	66	3700	51	0.00	3020	30	3020	2	0.00
	F	4450	84	4450	93	0.00	3770	254	3770	8	0.00
p3	A	4637	10901	4554	263	1.79	3759	652	3744	108	0.40
	B	5187	561	5104	1017	1.60	4359	387	4316	800	0.99
	C	4771	3548	4754	2920	0.36	3914	279	3910	248	0.10
	D	5407	4066	5397	1260	0.18	4544	700	4540	189	0.09
	E	4835	771	4835	95	0.00	3978	54	3978	10	0.00
	F	5585	1142	5585	103	0.00	4728	176	4728	27	0.00
p4	A	5283	563	5241	12	0.80	4315	6749	4313	148	0.05
	B	5801	87	5791	88	0.17	4865	29677	4813	147	1.07
	C	5401	198	5401	14	0.00	4491	236	4491	46	0.00
	D	6031	386	6031	71	0.00	5121	760	5121	956	0.00
	E	5414	20	5414	1	0.00	4521	18	4521	2	0.00
	F	6164	39	6164	5	0.00	5271	109	5271	15	0.00
p5	A	3756	460	3713	52	1.14	3142	286	3111	54	0.99
	B	4356	4200	4263	367	2.13	3692	222	3664	35	0.76
	C	3897	96	3869	97	0.72	3311	258	3311	48	0.00
	D	4559	1456	4532	417	0.59	4011	269	3950	89	1.52
	E	3932	118	3932	16	0.00	3292	40	3292	1	0.00
	F	4532	593	4532	6	0.00	4042	32	4042	1	0.00
p6	A	2364	1985	2309	33	2.33	1956	774	1956	23	0.00
	B	2864	806	2835	39	1.01	2506	260	2506	35	0.00
	C	2524	2210	2488	30	1.43	2148	4215	2069	8	3.68
	D	3163	23479	3163	671	0.00	2734	2336	2629	4	3.84
	E	2536	732	2536	33	0.00	2116	21	2116	3	0.00
	F	3286	428	3286	19	0.00	2866	114	2866	12	0.00

Table 2: Results obtained with CPLEX for 10×20 instances

opt contain the value of the optimal solution to P found by CPLEX. The required CPU times (in seconds) are given in columns *time* and *time lb*, for models P and RP, respectively. The values of the lower bound provided by RP are depicted in column *lb* and, finally, the percent deviation $100(opt - lb)/opt$ is presented under the heading *dev*.

As can be seen in Table 2, the lower bound given by RP was optimal for P in 15 out of the 36 uncorrelated instances. For the correlated instances, the lower bound was optimal in 21 cases. The deviations between the optimal solution to P and the lower bound obtained with RP tend to decrease for each instance as the limit on the total distance per vehicle, ℓ, increases. Indeed, all the lower bounds provided by RP for instances with $\ell = 100$ were optimal for P. For all other instances but two variants of the uncorrelated version of p_6, the deviations were under 3%. These small deviations assess the suitability of model RP to produce tight lower bounds for CDCPLP.

The CPU time required to solve exactly RP is, in general, much smaller than the time needed to solve P. Although in seven out of the 72 considered instances the time needed to solve RP was slightly larger than that needed to solve P, RP was always solved in less than one hour of CPU time, whereas CPLEX needed up to 8 hours to solve P for the most demanding instance. On the average, solving RP required three minutes per instance, while solving P required 30 minutes. Table 2 also shows how, in general terms, correlated instances were easier to solve than uncorrelated ones, as was expected.

We next comment on the numerical results obtained with the TS procedure. For implementing it, a few parameters had to be set. These include the termination criteria of the three levels of search, the tenure of the tabu attributes and the frequency for updating violation penalties.

In all three loops the penalties for infeasibilities are updated every ten iterations starting from $P_p = P_v = 1000$. Parameter α is initialized at value 2, increased by 0.01 every 100 iterations, and reset to 2 every time the incumbent solution improves. Parameters β_p, β_v count the number of infeasible solutions with respect to plant capacities and to vehicle length constraints, respectively, in the last ten iterations.

The outer loop is terminated after at least 300 iterations, when the best solution has not improved in the last 50 iterations. The risk of cycling in this type of search is small because of the extra moves inside each loop. Thus, it is not necessary to impose long tabu tenures. For the outer loop, we take them uniformly from the interval $[3, 5]$. Tabu statuses are overridden when this leads to a feasible solution that is better than the incumbent.

The intermediate loop, where the assignments of customers to plants are modified, is

terminated when no improvement of the incumbent solution has occurred in the last 30 iterations and at least 3000 iterations have been performed. In this case tabu tenures are randomly taken from $[7, 10]$. Finally, the iteration limit for the innermost loop is five times the number of customers currently assigned to the plant being explored. As in the intermediate loop, tabu tenures are taken from the interval $[7, 10]$.

To measure the quality of the solutions provided by our TS we have computed, for each instance, the deviation between the value of the obtained solution and the optimal solution. This value is depicted in Figures 1 and 2 for the uncorrelated and the correlated instances, respectively. The values of the percent deviations are given with respect to the left vertical

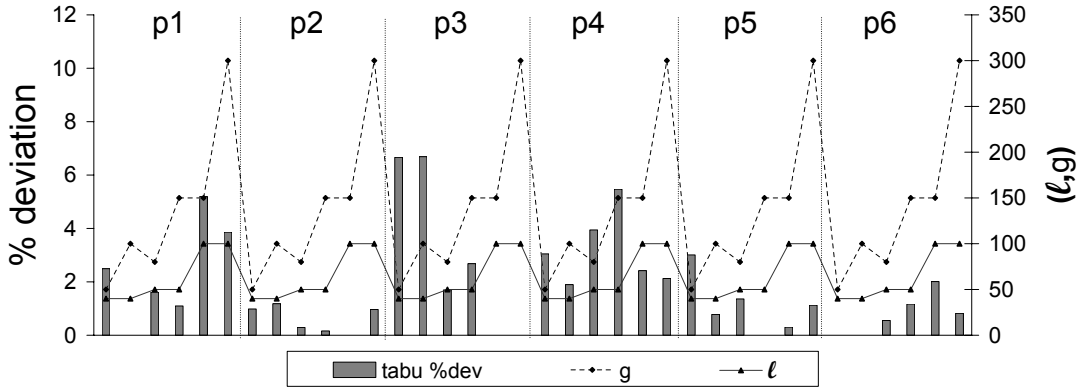

Figure 1: TS deviation from the optimal solution for uncorrelated instances

axis, whereas the values of the parameters g and ℓ are shown relative to the right vertical axis. In general, TS gives good solutions, especially taking into account the difficulty of the problem. In particular, seven/nine uncorrelated/correlated instances were optimally solved with TS. For the other ones the percent deviation from the optimal solution is not homogeneous, although it is below 1% for fifteen/seventeen uncorrelated/correlated instances.

The time needed by TS to solve each of the instances has also been recorded. Average times (in seconds) over the six instances with the same parameters g and ℓ, and the same cost structure are displayed in Table 3.

	A	B	C	D	E	F
Uncorrelated	26.35	25.37	18.61	18.42	6.07	4.41
Correlated	35.65	33.60	24.95	25.42	2.65	2.92

Table 3: Average CPU times (in seconds) of TS

As can be seen, the time required by our TS method is quite sensitive to the characteristics of the vehicles. Instances with loose distance constraints on the vehicles were solved up to

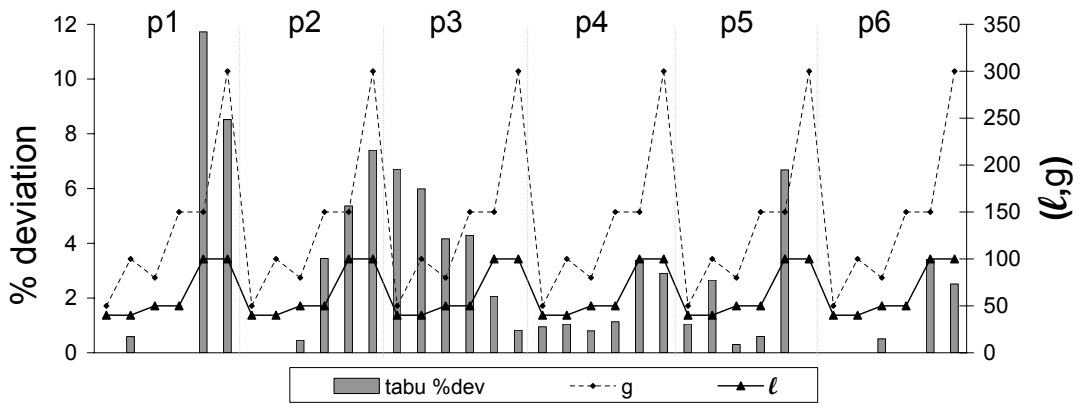

Figure 2: TS deviation from the optimal solution for correlated instances

ten times faster than those with very tight constraints. Vehicle costs had less influence on the time requirements of our algorithm. As far as the cost structure is concerned, and contrary to what happened for the exact algorithm, correlated instances were slightly more demanding than uncorrelated ones. The time required to solve an instance never exceeded one minute.

Figures 3 and 4 compare the sets of open plants in the TS and optimal solutions. Each bar consists of three parts: the middle one (in gray) shows the number of common open plants in both solutions, the lower one (in white) gives the number of plants that are open in the TS solution but are closed in the optimal solution, whereas the upper part (in black) displays the number of plants that are open in the optimal solution but are closed in the TS one. In general, the coincidences in the sets of open plants are quite large, specially for the uncorrelated instances. The figures also show how the set of plants that are open by TS is never larger than the set of plants opened in the optimal solution.

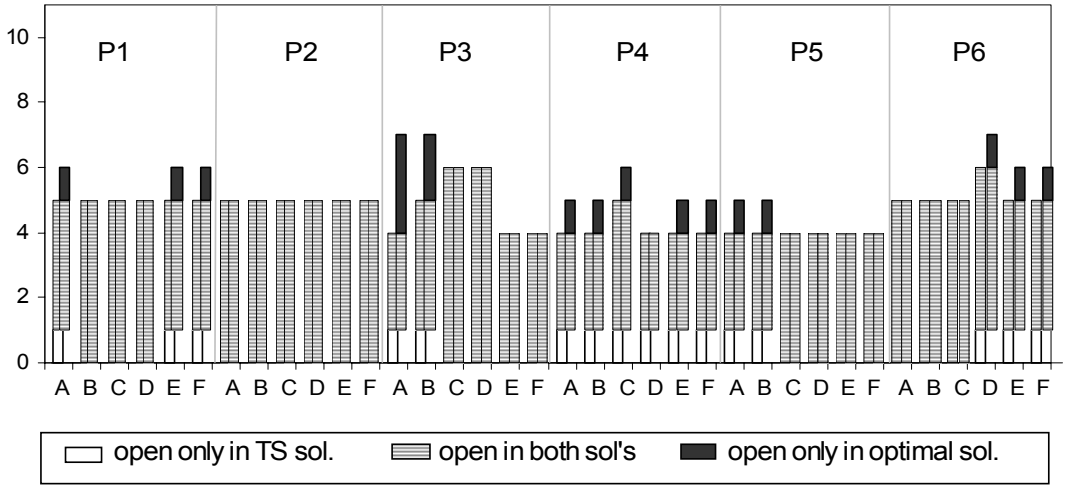

Figure 3: Coincidences in sets of open plants for uncorrelated instances

It is clear that full coincidence in the set of open plants does not imply optimality of the TS solution, since the vehicle allocations and the customer assignments need not coincide. However, there exists a strong relationship beetween the coincidences in the sets of open plants and the percent deviations of the TS solution with respect to the optimal. For example, in the case of the uncorrelated instances derived from p_2, TS obtains especially small deviations and all the sets of open plants coincide. This strong relationship is partly due to the cost

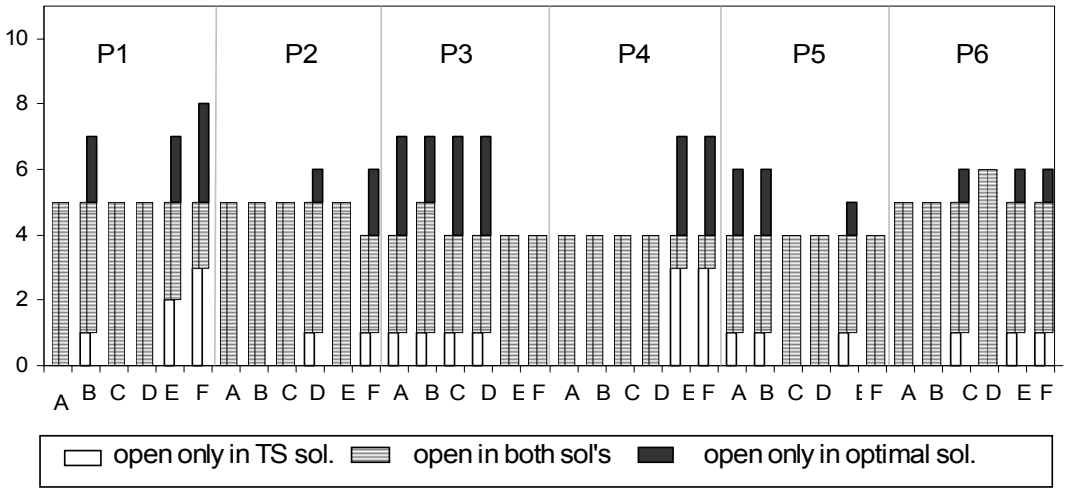

Figure 4: Coincidences in sets of open plants for correlated instances

structure of the solutions. Figure 5 shows separately the contributions of fixed opening costs, vehicle utilisation costs and assignment costs to the total value of the optimal solution of each correlated instance. For uncorrelated instances we get similar figures. As can be seen in Figure 5 the different SSCPLP instances gave raise to CDCPLP instances with quite diverse cost structures. In most cases, fixed costs for opening plants constitute between 50% and 75% of the total investment and they are never under 25%. The assignment cost contribution ranges from 5% to 20% of the total cost and the contribution of vehicle utilisation costs ranges from 30% to 60%. The major impact of fixed costs for opening plants in the total cost of the solutions explains why the choice of the adequate set of plants to open is so deeply related to the quality of a solution.

In order to evaluate the performance of the TS algorithm, we also solved larger instances, derived from SSCPLP instances from p_7 to p_{25}. To capture the average performance among different types of instances, we generated correlated instances corresponding to label "C", *i.e.*, instances with $g = 80$ and $\ell = 50$. Again, for each instance, both P and RP were solved using CPLEX, but this time we imposed a limit of two hours of CPU time and we recorded the best bounds obtained by CPLEX up to that moment.

	CPLEX(P) 2h		CPLEX(RP) 2h		Tabu Search			*best*
Instance	*best*	%*gap*	*lb comp*	*time*	*lb dev*	*CPX dev*	*time*	%*gap*
p7	3757	0.94	-0.51	510	2.31	1.36	69.02	0.938
p8	5680	2.63	-0.31	–	4.80	2.11	66.14	2.631
p9	2526	4.12	1.49	45	3.05	0.44	71.59	2.600
p10	12360	1.49	0.70	1218	8.49	7.65	35.30	0.783
p11	3150	9.67	4.07	747	3.88	-1.43	54.42	3.881
p12	3375	7.29	4.18	948	5.55	2.49	61.92	2.991
p13	3416	8.67	5.01	1252	6.76	3.16	78.77	3.484
p14	4343	0.83	0.76	4782	6.41	6.33	73.89	0.069
p15	5265	6.95	3.36	–	4.20	0.70	38.83	3.469
p16	7072	5.27	2.34	–	5.43	2.49	35.73	2.865
p17	6650	5.47	0.19	–	12.39	6.77	79.91	5.269
p18	9486	3.69	1.20	–	8.87	6.26	85.20	2.458
p19	11463	3.71	0.26	–	11.18	7.48	116.63	3.436
p20	13650	3.19	0.53	–	7.74	4.96	151.27	2.646
p21	5146	9.85	4.84	–	7.24	2.35	160.91	4.775
p22	3474	12.63	4.79	2671	19.21	10.91	165.11	7.488
p23	5387	12.80	4.63	–	28.02	18.75	172.20	7.812
p24	5351	7.91	1.79	–	12.59	6.20	171.44	6.016
p25	6328	2.59	1.29	150	3.04	1.74	166.44	1.280

Table 4: Results for large instances.

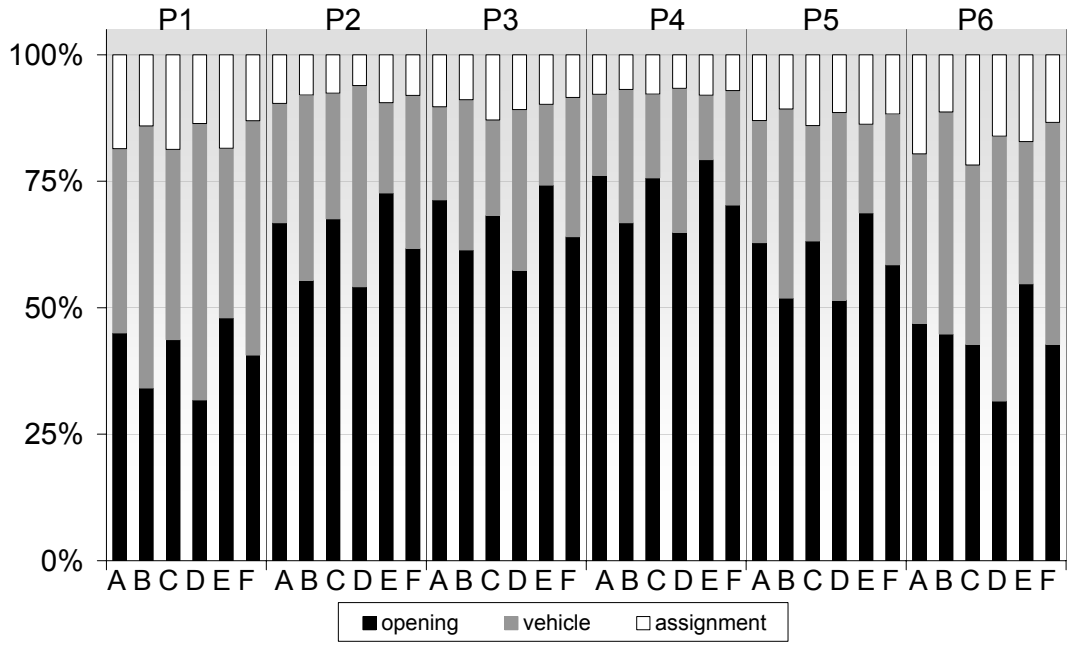

Figure 5: Contributions to total cost (correlated instances)

Table 4 displays the results obtained for the large instances. The first two columns refer to the solutions obtained by CPLEX with model P, whereas the following two columns refer to model RP. Recall that in both cases CPLEX was interrupted after two hours of CPU time. Columns *best* and $\%gap$ depict, respectively, the value of the best obtained solution and its corresponding percent gap relative to the best lower bound at the end of the execution. In the next column, *lb comp*, the deviation $100(lb_{RP} - lb_P)/lb_P$ is used to compare this lower bound (lb_P) with the lower bound produced by model RP (lb_{RP}). In addition, for the instances for which RP could be solved exactly within the time limit, column *time* gives the required CPU time. Columns under the heading Tabu Search give information on the behavior of TS. Column *lb dev* gives the percent deviation between the TS solution and the best of the above two lower bounds, $\max\{lb_P, lb_{RP}\}$. The TS solution is also compared to the best solution obtained by CPLEX in column *CPX dev* and TS execution time is given in column *time*. Finally, column *best* $\%gap$ shows the percent gap between the best known upper and lower bounds for each instance.

Note that none of these instances could be solved to optimality by CPLEX within the two hour limit. The smallest percent gap at termination is nearly 1% and for some of the largest instances it is close to 13%. In contrast, RP could be solved exactly for nine out of the 19 instances. It is remarkable that the lower bound lb_{RP} is tighter than lb_P for all but two (smaller) instances. In our opinion the results of TS are satisfactory, especially taking

into account the required CPU time and the difficulty and sizes of the instances. Note that the execution time does not exceed three minutes for the larger 20×40 instances, and, if we exclude two of the larger instances for which the results are not so good, the average deviation of the obtained solutions relative to the ones obtained by CPLEX after two hours of CPU time is 3.6% . Column "best %gap", which is a summary of the bounds of the previous columns, indicates that the considered instances could be solved quite efficiently. Note that the percent gap between the better upper and lower bounds never exceeds 8%. As can be observed from the results given in the previous columns, for most of the instances, this gap was obtained with the upper bound provided by CPLEX and the lower bound given by RP.

4 Conclusions

We have introduced the *Capacity and Distance Constrained Plant Location Problem*, an extension of the capacitated plant location problem that includes fleet management but, as opposed to typical location-routing problems, does not include route design. Different models were presented, including an integer programming relaxation, and four different valid inequalities for that relaxation. We have also proposed a tabu search algorithm, structured in different levels of search, obeying to the hierarchy in the decisions to take. Extensive computational experiments were carried out to assess the effectiveness of the algorithm and the quality of the lower bounds provided by the relaxation. The tabu search heuristic consistently provided optimal or near-optimal solutions within reasonable computational times, given the difficulty of the problem. These results also confirm the quality of our lower bounding procedure. On the other hand, the presented relaxation provided tight lower bounds.

Acknowledgements

This work was partially supported by CICYT grant TIC2003-05982-C05-04, by NSERC grant 39682-05 and by CAM grant UC3M-MTM-05-075. This support is gratefully acknowledged.

References

Albareda-Sambola, M., J. A. Díaz, and E. Fernández (2005). A compact model and tight bounds for a combined location-routing problem. *Computers & Operations Research 32*, 407–428.

Barceló, J., E. Fernández, and K. Jörnsten (1991). Computational results from a new lagangean relaxation algorithm for the capacitated plant location problem. *European Journal of Operational Research 53*, 38–45.

Berman, O., P. Jaillet, and D. Simchi-Levi (1995). Location-routing problems with uncertainty. In Z. Drezner (Ed.), *Facility Location: A Survey of Applications and Methods*, pp. 427–453. Springer, New York.

Daskin, M. S. (1995). *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley, New York.

Delmaire, H., J. A. Díaz, E. Fernández, and M. Ortega (1999). Reactive grasp and tabu search based heuristics for the single source capacitated plant location problem. *INFOR 37*, 194–225.

Díaz, J. A. and E. Fernández (2001). A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research 132*, 22–38.

Gendreau, M., A. Hertz, and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. *Management Science 40*, 1276–1290.

Glover, F. (1989). Tabu search: Part I. *ORSA Journal of Computing 1*, 190–206.

Glover, F. (1990). Tabu search: Part II. *ORSA Journal of Computing 2*, 4–32.

Laporte, G. (1988). Location-Routing problems. In B. L. Golden and A. A. Assad (Eds.), *Vehicle Routing: Methods and Studies*, pp. 163–197. North-Holland, Amsterdam.

Min, H., V. Jayaraman, and R. Srivastava (1998). Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research 108*, 1–15.

Mirchandani, P. B. and R. L. Francis (1990). *Discrete Location Theory*. Wiley, Chichester.

Nagy, G. and S. Salhi (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research 177*, 649–672.

Scholl, A., R. Klein, and C. Jürgens (1997). BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers & Operations Research 24*, 627–645.