

The Single Period Coverage Facility Location Problem: Lagrangean heuristic and column generation approaches

Maria Albareda-Sambola · Elena Fernández ·
Yolanda Hinojosa · Justo Puerto

Received: 31 October 2008 / Accepted: 4 May 2009 / Published online: 28 May 2009
© Sociedad de Estadística e Investigación Operativa 2009

Abstract In this paper we introduce the Single Period Coverage Facility Location Problem. It is a multi-period discrete location problem in which each customer is serviced in exactly one period of the planning horizon. The locational decisions are made independently for each period, so that the facilities that are open need not be the same in different time periods. It is also assumed that at each period there is a minimum number of customers that can be assigned to the facilities that are open. The decisions to be made include not only the facilities to open at each time period and the time period in which each customer will be served, but also the allocation of customers to open facilities in their service period.

We propose two alternative formulations that use different sets of decision variables. We prove that in the first formulation the coefficient matrix of the allocation subproblem that results when fixing the facilities to open at each time period is totally unimodular. On the other hand, we also show that the pricing problem of the second

M. Albareda-Sambola
Statistics and Operations Research Department, Technical University of Catalonia, C. Colom, 11,
08222 Terrassa, Spain
e-mail: maria.albareda@upc.edu

E. Fernández
Statistics and Operations Research Department, Technical University of Catalonia, Campus Nord,
C5-208, 08034 Barcelona, Spain
e-mail: e.fernandez@upc.edu

Y. Hinojosa (✉)
Departamento Economía Aplicada I, Universidad de Sevilla, Av. Ramón y Cajal, 1, 41018 Sevilla,
Spain
e-mail: yhinojos@us.es

J. Puerto
Dep Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Sevilla,
C/ Tarfia s/n, 41012 Sevilla, Spain
e-mail: puerto@us.es

model can be solved by inspection. We prove that a Lagrangean relaxation of the first one yields the same lower bound as the LP relaxation of the second one. While the Lagrangean dual can be solved with a classical subgradient optimization algorithm, the LP relaxation requires the use of column generation, given the large number of variables of the second model. We compare the computational burden for obtaining this lower bound through both models.

Keywords Discrete facility location · Multi-period location · Lagrangean heuristic · Column generation

Mathematics Subject Classification (2000) 90B80 · 90C10

1 Introduction

Discrete location models have a wide range of applications that are well documented in the literature (see, for instance, Mirchandani and Francis 1990, and references therein). Quite often, the relevance of discrete location comes from the fact that locational decisions are crucial strategic decisions in the design and management of various types of complex systems, due to their long-lasting effect. Therefore, developing efficient tools to guide the locational phase of the decision-making process is crucial to improve the performance of such systems. One important issue that is often present in real applications is the dynamic nature of the problem, which implies that various elements such as costs, resources, requirements, etc. may change within different time periods throughout the time horizon. From the location point of view this gives rise to different types of multi-period problems. Multi-period location problems are investigated since the early papers by Warszawski (1973), van Roy and Erlenkotter (1982), up to the more recent references by Daskin et al. (1992), Galvão and Santibañez-González (1992), and Current et al. (1997), among others. Most of these approaches have been used for the design of supply structures by deciding which existing facilities should be closed and where new facilities should be opened. In this case, not only the transportation plan but also the time-staged establishment of the facilities are decision variables (see, e.g., Chardaire et al. 1996; Drezner 1995; Hinojosa et al. 2000, 2008; Albareda-Sambola et al. 2009; and Melo et al. 2006, 2009). However, some modeling aspects still require further attention. In this work we shall focus on one of them: how to address the single period coverage of customers within a planning horizon. In the actual world, there are cases where customers require to be serviced only once, and the decision on when to service each customer depends on the current service network. However, we are not aware of any research in the above-mentioned field. Instead, in most (all) known models it is required that either the entire or a pre-specified fraction of the population be reached at each time period, and that the demand of covered customers remains satisfied in the subsequent periods. It is clear that, although not considered before, the single period coverage fits to actual applications. For instance, in vaccination campaigns it is required that the population in different locations be serviced only once (of course before the current infection season). The nature of this application requires that service to users is given

only once. Nevertheless, operational and budget constraints avoid to achieve complete coverage in one single period, and therefore, an optimal plan is needed to reach the full service goal while ensuring total coverage at the end of the planning horizon.

In this paper we introduce the Single Period Coverage Facility Location Problem, SPCFLP for short. The SPCFLP consists of minimizing the total costs throughout a finite time planning horizon while ensuring that at each single period a minimum number of plants are open, each of them having a capacity H , and that each customer is serviced exactly once throughout the entire planning horizon. We accept that plants are opened and closed in different time periods to conform an adequate service network. The SPCFLP is quite general since it includes as particular cases some interesting difficult problems, as, for instance, the well-known capacitated plant location problem (CPLP) (Sridharan 1995), or the variable-size bin packing problem (Correia et al. 2008). The problem belongs to the class NP-hard, since it reduces to the above-mentioned CPLP, when the planning horizon shrinks to one single period.

Mainly we focus on modeling issues, and we propose two solution procedures for the SPCFLP that resort to Lagrangean Relaxation, which has been widely used to address different types of discrete location problems (see, for instance, Guignard 2003). We show that the allocation subproblem that results when the set of facilities to be opened at each time period is known can be solved in polynomial time, since its coefficient matrix is totally unimodular, and that the proposed Lagrangean relaxations can be solved by inspection. Based on these properties, we propose a solution approach that provides both lower and upper bounds by combining subgradient optimization to solve the Lagrangean dual with an ad hoc heuristic that uses information from the Lagrangean subproblem to generate feasible solutions. We also propose another resolution method based on a reformulation of the SPCFLP as a set-partitioning problem, at the cost of introducing exponentially many columns (variables). Despite of that, this formulation is specially suitable for column generation methods, since the corresponding pricing problem can be solved easily by inspection. We prove that the lower bound of the above Lagrangean dual coincides with the LP bound of the set-partitioning reformulation. We have run a series of computational experiments, in order to assess the efficiency of our algorithms and to compare the computational burden required to obtain this lower bound with each model. Our experiments show that the SPCFLP is difficult to solve. Indeed, already for some medium size instances, CPLEX failed to obtain the optimal solution in four hours of cpu time. On the contrary, the numerical results indicate that the proposed approach is very effective and provides very good quality feasible solutions with small duality gaps and small computation times.

The paper is organized as follows. Section 2 describes a mixed integer mathematical programming (MIP) formulation of the SPCFLP. There we also analyze the allocation subproblem that results when the set of facilities that are open in each time period is known, and we introduce two Lagrangean relaxations of the SPCFLP. In Sect. 3 we develop a column generation approach based on a reformulation of the SPCFLP as a set-partitioning problem. Section 4 reports on the results obtained in our computational experiments. In this regard, we have tested several batteries of instances that show a good performance of our algorithms in terms of quality and efficiency. The paper ends with some concluding remarks.

2 A mathematical programming formulation of the SPCFLP

In this section we present the formal description of the SPCFLP, and we model it by means of a mixed integer linear program. Recall that this model looks for the locational and assignment decisions, over a given planning horizon, that lead to the minimization of the total operation costs.

To simplify the mathematical formulation of our problem, we use the following notation.

- I : the set of customers, indexed by $i \in I$.
- J : the set of possible locations for facilities, indexed by $j \in J$.
- T : the set of time periods, indexed by $t \in T$.
- H : a constant that denotes the maximum number of customers that can be assigned to a plant if it is open.

For each period $t \in T$, define

- p^t : minimum number of facilities to open at period t .

In addition, two types of costs are combined in this problem:

- c_{ij}^t : assignment value of allocating customer i to facility j at time period t .
- f_j^t : cost of opening facility j at period t .

We assume that a facility that is opened in one period can be closed in the next one and that a closed facility can be reopened in subsequent time periods. We require that each customer is serviced in exactly one period of the planning horizon.

The goal is to find the facilities to open at each period $t \in T$ and the allocation of customers to the open facilities that satisfy the above requirements at the minimum total cost.

In order to build a mathematical programming formulation for the SPCFLP, we define the following decision variables:

- $x_{ij}^t = \begin{cases} 1 & \text{if customer } i \text{ is assigned to facility } j \text{ at time period } t, \\ 0 & \text{otherwise.} \end{cases}$
- $y_j^t = \begin{cases} 1 & \text{if facility } j \text{ is open at time period } t, \\ 0 & \text{otherwise.} \end{cases}$

Using these conventions, a formulation of the SPCFLP is

$$(M1) \quad \min \quad \sum_{t \in T} \sum_{j \in J} \left(f_j^t y_j^t + \sum_{i \in I} c_{ij}^t x_{ij}^t \right) \quad (1)$$

$$\text{s.t.} \quad \sum_{t \in T} \sum_{j \in J} x_{ij}^t = 1 \quad \forall i \in I, \quad (2)$$

$$\sum_{i \in I} x_{ij}^t \leq H y_j^t \quad \forall j \in J, \forall t \in T, \quad (3)$$

$$\sum_{j \in J} y_j^t \geq p^t \quad \forall t \in T, \quad (4)$$

$$x_{ij}^t \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \forall t \in T, \quad (5)$$

$$y_j^t \in \{0, 1\} \quad \forall j \in J, \forall t \in T. \quad (6)$$

Constraints (2) ensure that each customer is serviced exactly one time throughout the planning horizon. Constraints (3) guarantee that no more than H customers are assigned to one plant if it is open. The following group of constraints (4) ensure that at least p^t plants are opened at time period t . Constraints (5) and (6) are the binary conditions on the decision variables. As we will see, constraints (5) can be substituted by continuous nonnegativity bounds on their value.

If constraints (4) were not included and constraints (3) were relaxed to $x_{ij}^t \leq y_j^t$, the problem would just be an Uncapacitated Facility Location Problem (Cornuejols et al. 1990). Since constraints (3) are a particular case of capacity constraints, if we just omitted constraints (4), the problem would be a particular case of the Single Source Capacitated Plant Location Problem (Cortinhal and Captivo 2003). Moreover, if constraints (4) were omitted and the planning horizon shrank to one period, the problem would be a Variable-Size Bin Packing Problem (Correia et al. 2008).

2.1 The assignment subproblem

In this section we study the structure of the allocation subproblem that results when the set of facilities that are open in each time period is fixed. As we will see, the allocation subproblem can be solved in polynomial time, since its coefficient matrix is totally unimodular. This property will be used later on in our solution approach for $M1$.

Proposition 1 *If the set of open facilities in period t , J^t , is known for $t \in T$, then the optimal allocation for the customers can be obtained by solving the following linear program:*

$$(TP) \quad \min \quad \sum_{t \in T} \sum_{j \in J^t} \sum_{i \in I} c_{ij}^t x_{ij}^t \tag{7}$$

$$\text{s.t.} \quad \sum_{t \in T} \sum_{j \in J^t} x_{ij}^t = 1 \quad \forall i \in I, \tag{8}$$

$$\sum_{i \in I} x_{ij}^t \leq H \quad \forall t \in T, j \in J^t, \tag{9}$$

$$x_{ij}^t \geq 0 \quad \forall i \in I, \forall t \in T, \forall j \in J^t, \tag{10}$$

whose coefficient matrix is totally unimodular, since it can be transformed into a transportation problem.

Corollary 1 *In formulation $M1$ the binary conditions $x_{ij}^t \in \{0, 1\}$, $\forall i \in I, \forall j \in J, \forall t \in T$, can be substituted by the nonnegativity conditions $x_{ij}^t \geq 0$, $\forall i \in I, \forall j \in J, \forall t \in T$.*

2.2 Lagrangean relaxations of the SPCFLP

We next present two Lagrangean relaxations of model $M1$. The first one incorporates to the objective function capacity constraints (3), whereas the second one relaxes assignment constraints (2).

2.2.1 Lagrangean relaxation of capacity constraints

If we relax constraints (3), for a given set of multipliers $u_j^t \geq 0$, the relaxed problem, denoted by $L_1(u)$, is given by

$$L_1(u) = \min \sum_{t \in T} \sum_{j \in J} \left(f_j^t y_j^t + \sum_{i \in I} c_{ij}^t x_{ij}^t \right) + \sum_{t \in T} \sum_{j \in J} u_j^t \left(\sum_{i \in I} x_{ij}^t - H y_j^t \right) \quad (11)$$

$$\text{s.t.} \quad \sum_{t \in T} \sum_{j \in J} x_{ij}^t = 1 \quad \forall i \in I, \quad (12)$$

$$\sum_{j \in J} y_j^t \geq p^t \quad \forall t \in T, \quad (13)$$

$$x_{ij}^t \geq 0, \quad y_j^t \in \{0, 1\} \quad \forall j \in J, \forall t \in T. \quad (14)$$

After some algebra, the objective function (11) can be rearranged to

$$\min \sum_{t \in T} \sum_{j \in J} (f_j^t - H u_j^t) y_j^t + \sum_{t \in T} \sum_{j \in J} \sum_{i \in I} (c_{ij}^t + u_j^t) x_{ij}^t.$$

The objective function value of $L_1(u)$ can be obtained by decomposing problem (11)–(14) in two independent problems: one in the y variables, and another one in the x variables. Each of them can be solved by inspection.

The Lagrangean Dual associated with $L_1(u)$ is thus

$$(D1) \quad d_1 = \max_{u \geq 0} L_1(u).$$

Problem $D1$ can be solved with basic subgradient optimization (the implementation is similar to the one followed in Albareda-Sambola et al. 2009, where further details on parameters can be found). For a given vector $u \geq 0$, let $(y(u), x(u))$ denote the optimal solution to $L_1(u)$. Then, a subgradient of $L_1(u)$ is given by $\gamma = (\gamma_j^t)_{j \in J, t \in T}$, where $\gamma_j^t = \sum_{i \in I} x(u)_{ij}^t - H y(u)_j^t$.

Proposition 2 *The optimal value of the Lagrangean Dual D_1 , d_1 , coincides with the value of the LP relaxation of program $M1$.*

Proof The result follows since $L_1(u)$ has the integrality property (cf. Theorem 2 in Geoffrion 1974). \square

2.2.2 Lagrangean relaxation of assignment constraints

If we relax constraints (2), for a given set of multipliers $v_i \in \mathbb{R}$, the relaxed problem, denoted by $L_2(v)$, is given by

$$L_2(v) = \min \sum_{t \in T} \sum_{j \in J} \left(f_j^t y_j^t + \sum_{i \in I} c_{ij}^t x_{ij}^t \right) + \sum_{i \in I} v_i \left(1 - \sum_{t \in T} \sum_{j \in J} x_{ij}^t \right) \tag{15}$$

$$\text{s.t. } \sum_{i \in I} x_{ij}^t \leq H y_j^t \quad \forall j \in J, \forall t \in T, \tag{16}$$

$$\sum_{j \in J} y_j^t \geq p^t \quad \forall t \in T, \tag{17}$$

$$0 \leq x_{ij}^t \leq 1, \quad y_j^t \in \{0, 1\} \quad \forall j \in J, \forall t \in T. \tag{18}$$

After some algebra the objective function (15) can be rearranged to

$$\min \sum_{i \in I} v_i + \sum_{t \in T} \sum_{j \in J} \left(f_j^t y_j^t + \sum_{i \in I} (c_{ij}^t - v_i) x_{ij}^t \right).$$

A solution to $L_2(v)$ can be obtained as follows:

- For each $j \in J$ and each $t \in T$, do the following:
 - let $\{i_1, i_2, \dots, i_H, \dots, i_{|I|}\}$ be the index set of customers ordered by increasing values of $c_{ij}^t - v_i$.
 - Let $s_{jt}^* = \min\{H, \max\{r : c_{i_r j}^t - v_{i_r} < 0\}\}$.
 - Define $w_j^t = f_j^t + \sum_{r=1}^{s_{jt}^*} (c_{i_r j}^t - v_{i_r})$.
- For each $t \in T$, do:
 - let $\{j_1, \dots, j_{p^t}, \dots, j_{|J|}\}$ be the index set of plants ordered by increasing values of w_j^t .
 - Let $j(t)^* = \max\{p^t, \max\{r : w_{j_r}^t < 0\}\}$.
 - For each $r \in \{1, \dots, j(t)^*\}$, do $y_{j_r}^t = 1$, and $x_{i_s j_r}^t = 1$, for $s = 1, \dots, s_{j_r t}^*$.

Now, the Lagrangean Dual associated with $L_2(v)$ is

$$(D2) \quad d_2 = \max_{v \in \mathbb{R}^{|I|}} L_2(v).$$

Problem $D2$ can be solved with basic subgradient optimization (the implementation is similar to the one followed in Albareda-Sambola et al. 2009, where further details on parameters can be found). For a given vector v , let $(y(v), x(v))$ denote the optimal solution to $L_2(v)$. Then, a subgradient of $L_2(v)$ is given by $\varphi = (\varphi_i)_{i \in I}$, where $\varphi_i = 1 - \sum_{t \in T} \sum_{j \in J} x_{ij}^t(v)$.

2.3 Upper bounds

Assume that, for a given multipliers vector of appropriate dimensions, the Lagrangean problem $L_i(\cdot)$ for $i = 1, 2$ is solved. Then, a feasible solution for program $M1$ can be obtained by solving the allocation subproblem $TP(\tilde{y})$ associated with the set $J^t(\tilde{y})$ of open facilities at each period given by the optimal solutions \tilde{y} to $L_i(\cdot)$ for $i = 1, 2$, that is,

$$J^t(\tilde{y}) = \{j \in J : \tilde{y}_j^t = 1\}. \tag{19}$$

Hence, when the solution of $TP(\tilde{y})$ is a feasible solution to $M1$ (i.e., $\sum_t |J^t(\tilde{y})| \geq |I|/H$) an upper bound to $M1$ can be obtained as $\text{val}(TP(\tilde{y})) + \sum_{t \in T} \sum_{j \in J^t(\tilde{y})} f_j^t$, where $\text{val}(P)$ stands for the optimal value of problem P .

Note that if $\sum_t |J^t(\tilde{y})| < |I|/H$, the relaxed problems can be reinforced by adding the constraint $\sum_{j,t} y_j^t \geq |I|/H$. In this case, the solution of the relaxed problems is completed by opening the cheapest plants until this constraint is satisfied.

In the next section we present another formulation for the SPCFLP that is suitable for a column generation approach.

3 Set-Partitioning formulation of the SPCFLP

Feasible assignments to plants are associated with subsets of customers $S \subset I$ such that $|S| \leq H$. Let K denote the index set of such subsets. For each triplet (S_k, j, t) , $k \in K, j \in J, t \in T$, let p_{kj}^t denote the cost of opening facility j in period t and assigning to it all the customers of S_k . That is, $p_{kj}^t = f_j^t + \sum_{i \in S_k} c_{ij}^t$.

For building another formulation for the SPCFLP, we define the following decision variables:

- $z_{kj}^t = \begin{cases} 1 & \text{if the subset of customers } S_k \text{ is assigned to facility } j \text{ at time period } t, \\ 0 & \text{otherwise.} \end{cases}$

Then, a model formulation for the SPCFLP is

$$(M2) \quad \min \quad \sum_{t \in T} \sum_{j \in J} \sum_{k \in K} p_{kj}^t z_{kj}^t \tag{20}$$

$$\text{s.t.} \quad \sum_{t \in T} \sum_{j \in J} \sum_{k \in K} a_{ik} z_{kj}^t = 1 \quad \forall i \in I, \tag{21}$$

$$\sum_{k \in K} z_{kj}^t \leq 1 \quad \forall j \in J, \forall t \in T, \tag{22}$$

$$\sum_{k \in K} \sum_{j \in J} z_{kj}^t \geq p^t \quad \forall t \in T, \tag{23}$$

$$z_{kj}^t \in \{0, 1\} \quad \forall k \in K, \forall j \in J, \forall t \in T, \tag{24}$$

where the coefficients of matrix $A = (a_{ik})$ are given by

- $a_{ik} = \begin{cases} 1 & \text{if customer } i \in S_k, \\ 0 & \text{otherwise.} \end{cases}$

Constraints (21) ensure that each customer is serviced in exactly one period. Constraints (22) guarantee that at each period at most one feasible assignment is selected for each plant, whereas constraints (23) impose that at least p^t assignments take place at each period (and thus at least p^t plants are open).

The large number of variables of formulation $M2$ makes it suitable for column generation.

Proposition 3 *The optimal value of the Lagrangean dual D_2 , d_2 , coincides with the value of the LP relaxation of program $M2$.*

Proof Consider the following Lagrangean problem resulting from relaxing assignment constraints (21) in $M2$ in a Lagrangean fashion:

$$LM2(\pi) = \min \sum_{i \in T} \sum_{j \in J} \sum_{k \in K} p_{kj}^t z_{kj}^t + \sum_{i \in I} \pi_i \left(1 - \sum_{t \in T} \sum_{j \in J} \sum_{k \in K} a_{ik} z_{kj}^t \right) \quad (25)$$

$$\text{s.t. } \sum_{k \in K} z_{kj}^t \leq 1 \quad \forall j \in J, \forall t \in T, \quad (26)$$

$$\sum_{k \in K} \sum_{j \in J} z_{kj}^t \geq p^t \quad \forall t \in T, \quad (27)$$

$$z_{kj}^t \in \{0, 1\} \quad \forall k \in K, \forall j \in J, \forall t \in T. \quad (28)$$

In view of the unimodularity of the set of constraints (26)–(27), this Lagrangean problem has the integrality property, and, thus (cf. Theorem 2 in Geoffrion 1974), the bound obtained with its Lagrangean dual coincides with the value of the LP relaxation of $M2$.

Note that, for each feasible solution \hat{z} to (26)–(28), for any fixed $j \in J, t \in T$, there is a one-to-one correspondence between $\hat{z}_{kj}^t, k \in K$, and a vector $(\hat{y}_j^t, \hat{x}_{ij}^t), i \in I$ that satisfies constraints (16)–(18). In particular, $\hat{z}_{kj}^t = 0$ for all $k \in K$ if and only if $\hat{y}_j^t = \hat{x}_{ij}^t = 0$ for all $i \in I$, whereas there exists (a unique) $\hat{k} \in K$ such that $\hat{z}_{\hat{k}j}^t = 1$ (associated with the subset of customers $S_{\hat{k}}$) if and only if $\hat{y}_j^t = 1$ and $\hat{x}_{ij}^t = 1$ for all $i \in S_{\hat{k}}$. Therefore, by the definition of the variables z_{kj}^t for $j \in J, t \in T, (\hat{x}_{ij}^t)_{i \in I}$ represents a feasible assignment to facility j at time period t , i.e., $\sum_{i \in I} \hat{x}_{ij}^t \leq H \hat{y}_j^t$. Also, constraints (27) are equivalent to $\sum_{j \in J} \hat{y}_j^t \geq p^t$.

On the other hand, the objective function of $LM2(\pi)$ can be expressed as

$$\sum_{i \in I} \pi_i + \min \left[\sum_{t \in T} \sum_{j \in J} \sum_{k \in K} p_{kj}^t z_{kj}^t - \sum_{t \in T} \sum_{j \in J} \sum_{k \in K} \sum_{i \in I} \pi_i a_{ik} z_{kj}^t \right]. \quad (29)$$

Observe that, by the definition of the cost coefficients and by constraints (26), for a given $j \in J, t \in T, \sum_{k \in K} p_{kj}^t z_{kj}^t = f_j^t \hat{y}_j^t + \sum_{i \in I} c_{ij}^t \hat{x}_{ij}^t$. In addition, by the definition of the coefficients $a_{ik}, \sum_{k \in K} \pi_i a_{ik} z_{kj}^t = \pi_i \hat{x}_{ij}^t$. Taking into account the above considerations, program $LM2(\pi)$ can be rewritten as

$$\begin{aligned} & \sum_{i \in I} \pi_i + \min \sum_{t \in T} \sum_{j \in J} \left(f_j^t y_j^t + \sum_{i \in I} c_{ij}^t x_{ij}^t \right) - \sum_{i \in I} \sum_{t \in T} \sum_{j \in J} \pi_i x_{ij}^t \quad (30) \\ & \text{s.t. } \sum_{i \in I} x_{ij}^t \leq H y_j^t \quad \forall j \in J, \forall t \in T, \\ & \sum_{j \in J} y_j^t \geq p^t \quad \forall t \in T, \\ & x_{ij}^t \in \{0, 1\}, \quad y_j^t \in \{0, 1\} \quad \forall j \in J, \forall t \in T, \end{aligned}$$

which is indeed the Lagrangean problem $L_2(\pi)$ resulting from relaxing assignment constraints (2) in $M1$. □

3.1 The pricing subproblem

Let $\widehat{K} \subseteq K$ denote the index set of a known set of columns, and let us suppose that we have solved $LM2^{\widehat{K}}$, the LP relaxation of $M2$ restricted to the columns indexed in \widehat{K} , with optimal value $\text{val}(LM2^{\widehat{K}})$. Let π , λ , and μ denote the optimal values of dual variables associated with constraints (21), (22), and (23), respectively. Then, for finding, if it exists, a variable that will improve the current LP solution, we must find the column of the coefficient matrix of $M2$ with the smallest reduced cost. For variable z_{kj}^t $j \in J, k \in K, t \in T$, its reduced cost is given by $r_{kj}^t = p_{kj}^t - \sum_{i \in I} \pi_i a_{ik} - \lambda_j^t - \mu^t$.

Thus, for finding the column that yields the smallest reduced cost, we must solve the following pricing problem:

$$(PP) \quad \min_{j \in J, k \in K, t \in T} \quad r_{kj}^t = p_{kj}^t - \sum_{i \in I} \pi_i a_{ik} - \lambda_j^t - \mu^t.$$

Since $p_{kj}^t = f_j^t + \sum_{i \in S_k} c_{ij}^t$, we have $r_{kj}^t = f_j^t + \sum_{i \in I} (c_{ij}^t - \pi_i) a_{ik} - \lambda_j^t - \mu^t$. Note also that feasible columns a_k are characterized by the condition $\sum_{i \in I} a_{ik} \leq H$. Thus, the solution to PP can be obtained by solving a series of independent problems, one for each $j \in J, t \in T$. Since for given $j \in J, t \in T$, the value $f_j^t - \lambda_j^t - \mu^t$ is fixed, the corresponding problem reduces to

$$(PP_j^t) \quad \min \quad \sum_{i \in I} (c_{ij}^t - \pi_i) a_i \tag{31}$$

$$\text{s.t.} \quad \sum_{i \in I} a_i \leq H, \tag{32}$$

$$a_i \in \{0, 1\} \quad \forall i \in I. \tag{33}$$

Note that PP_j^t can be solved by inspection as follows:

- Let $\{i_1, i_2, \dots, i_H, \dots, i_{|I|}\}$ be the index set of customers ordered by increasing values of $c_{ij}^t - \pi_i$.
- Set $s^* = \min\{H, \max\{r : c_{i_r j}^t - \pi_{i_r} < 0\}\}$.
- The optimal solution to PP_j^t is $\hat{a}_{i_r} = 1, r = 1, \dots, s^*; \hat{a}_{i_r} = 0$, otherwise, which gives a column of A with reduced cost $\hat{r}_j^t = f_j^t + \sum_{i \in I} (c_{ij}^t - \pi_i) \hat{a}_i - \lambda_j^t - \mu^t$.

Proposition 4

1. If $\min_{j \in J, t \in T} \hat{r}_j^t \geq 0$, then the solution to $LM2^{\widehat{K}}$ is optimal to the LP relaxation of $M2$.
2. $LB1 = \text{val}(LM2^{\widehat{K}}) + \sum_{j \in J} \sum_{t \in T} \min\{\hat{r}_j^t, 0\}$ is a valid lower bound for the optimal value of the SPCFLP.

3.2 Initial set of columns

In order to obtain the initial subset of columns, two different GRASP heuristics (Feo and Resende 1995) have been applied. In both heuristics columns are iteratively built with some randomized greedy criterion. The basic difference between the two procedures is that in the first one, in the iterative step the customer i to be assigned is fixed, and the pair (j, t) to which i is assigned is selected, among the ones that fulfil some additional conditions, relative to the greedy function, whereas in the second heuristic the pair (j, t) is fixed, and the set of customers assigned to that pair is randomly selected (among the unassigned ones). Thus, whereas in the first heuristic several “partially built” columns can exist simultaneously, in the second heuristic each column is completely built before a new one is started. Like in formulation $M2$, in both heuristics each column k is associated with a facility j and a time period t , and it is defined by a subset of customers.

Apart from the obvious condition that no column contains more than H customers, for avoiding generating too many columns in the initial set of columns, in both heuristics we exclude columns with very few customers by ensuring that each column has at least $\max\{\lfloor \frac{H}{2} \rfloor, \lfloor \frac{|I|}{|T||J|} \rfloor\}$ customers. In the first heuristic, this is achieved as follows. Each time a new customer is assigned to a “partially built” column, the current size of the column is checked. When this number $r = h(j, t)$ is at least $\max\{\lfloor \frac{H}{2} \rfloor, \lfloor \frac{|I|}{|T||J|} \rfloor\}$, we “close” the column (do not allow more customers to be added to it) with probability $\pi_r = \frac{r-1}{2\lfloor \frac{H}{2} \rfloor}$ if $r < H$ or with probability $\pi_r = 1$ if $r = H$. When a column is “closed,” additional columns with the same pair (j, t) can be generated in subsequent iterations.

Each iteration of the first heuristic consists of the following. First, an unassigned customer i is randomly selected and assigned to a column associated with a facility and time period (j, t) randomly selected from a Restricted Candidate List (RCL). The RCL contains “good” facilities for customer i , among the ones of the “partially built” columns. In particular, a facility j is considered “good” for customer i at time period t if $c_{i,j}^t$ does not exceed a fraction of the maximum assignment cost of customer i . The fraction is initially set to 0.5, but if the resulting RCL is empty, it is increased (in steps of 0.1) until a suitable facility is found (now, possibly corresponding to an empty column). The pseudocode of the first heuristic for generating the initial set of columns is presented in Fig. 1.

The second heuristic consists of two nested loops, where each outer iteration builds a different block of columns. Columns of each block are built at the inner iterations in such a way that each customer belongs to exactly one column of the block. For the current block, let \hat{I} denote the set of unassigned customers, and \hat{T} the set of time periods that have less than p^t associated columns. Initially, each block contains no columns, so that $\hat{I} = I$ and $\hat{T} = T$. For building each column, first a time period t is randomly selected. The time period t is randomly selected from \hat{T} when $\hat{T} \neq \emptyset$, whereas, otherwise, it is randomly generated from T . Once the time period t is fixed, the facility j_t is randomly selected from a Restricted Candidate List, $RCL1$, constructed as follows. For each pair (j, t) , one tentative column is built. This column contains $h(j, t)$ unassigned customers, where $h(j, t)$ is

```

for ( $iter = 1, \dots, numiter$ ) do
   $h(j, t) = 0 \forall j \in J, t \in T,$ 
   $I_{assign} := \emptyset,$ 
  while ( $I_{assign} \neq I$ ) do
    Select randomly  $i \in I \setminus I_{assign},$ 
     $c_{max} := \max\{c_{ij}^t : j \in J\}, \quad frac := 0.5,$ 
     $RCL := \{(j, t) : h(j, t) \geq 1 \text{ and } c_{ij}^t \leq frac \times c_{max}\},$ 
    while ( $RCL \neq \emptyset$ ) do
       $frac := frac + 0.1,$ 
       $RCL := \{(j, t) : c_{ij}^t \leq frac \times c_{max}\},$ 
    endwhile
    Randomly select  $(j, t) \in RCL,$ 
    Assign  $i$  to column  $(j, t),$ 
     $h(j, t) := h(j, t) + 1,$ 
    if ( $h(j, t) \geq \max\{\lfloor \frac{H}{2} \rfloor, \lfloor \frac{|I|}{|T|} \rfloor\}$ ) then close column  $(j, t)$  with probability  $\pi_r,$  with  $r = h(j, t),$ 
     $I_{assign} := I_{assign} \cup \{i\},$ 
  endwhile
endfor

```

Fig. 1 GRASP Heuristic 1 for building the initial set of columns

randomly selected in the interval $[\lfloor \frac{H}{2} \rfloor, \min\{H, |\widehat{I}|\}]$. The customers of the tentative column for the pair (j, t) are randomly selected from another Restricted Candidate list, $RCL2$, that contains the $2h(j, t)$ unassigned customers with smaller assignment costs c_{ij}^t . The value of this tentative column is $v_j = f_j^t + \sum_{i \in I_{jt}} c_{ij}^t$. Then, $RCL1 = \{j \in J : v_j \leq v_{min} + 0.5(v_{max} - v_{min})\}$, where v_{max} and v_{min} respectively denote the larger and smaller values among all generated tentative columns. The pseudocode of the second heuristic for generating the initial set of columns is presented in Fig. 2.

It is worth noting that the initial subset of columns obtained in both heuristics may not define a feasible solution to formulation $M2$ because it may not fulfil constraints (23) regarding the minimum number of plants that need to be opened at each time period (p^t). To avoid this inconvenience, the solution is completed by opening the cheapest plants in each stage t until the constraint is satisfied. Finally, to improve the quality of this solution, at each iteration a local search plus an interchange is performed.

4 Computational study

The computational tests presented in this section have been designed in order to evaluate and compare the performance of the two Lagrangean relaxations developed in Sect. 2.2 and the column generation proposed in Sect. 3.1. On this account, all the solution procedures were implemented using *Visual C++ 6.0*, where *ILOG CPLEX 8.1 Callable Library* routines have been used for the implementation of the linear and integer programs. Default parameters have been used. All computational tests have been performed on a PC with a *Pentium IV* processor with 3 GHz and 1 GB of RAM.

For generating the instances, we identify three relevant factors in the design of our experiment, namely the number of candidate facilities sites, the planning hori-

```

for (iter = 1, ..., numiter) do
     $\hat{T} := I, \quad \hat{T} = T,$ 
     $\hat{p}^t = 0 \quad \forall t \in T,$ 
    while ( $\hat{T} \neq \emptyset$ ) do
        if ( $\hat{T} \neq \emptyset$ ) then
            Randomly select  $t \in \hat{T},$ 
        otherwise
            Randomly select  $t \in T,$ 
         $v_{\min} = \infty, \quad v_{\max} = 0,$ 
        for ( $j \in J$ ) do
            Randomly generate  $h(j, t) \in [H/2, \min\{H, |\hat{T}\}],$ 
            Define RCL2 as the set of the  $\min\{2h(j, t), |\hat{T}|\}$ 
                customers in  $\hat{T}$  with smaller assignment costs  $c_{ij}^t,$ 
            Randomly select a set  $I_{jt} \subseteq \hat{T}$  of  $h(j, t)$  indices from RCL2,
            Evaluate  $v_j = f_j^t + \sum_{i \in I_{jt}} c_{ij}^t,$ 
            if ( $v_j > v_{\max}$ ) then  $v_{\max} = v_j,$ 
            if ( $v_j < v_{\min}$ ) then  $v_{\min} = v_j,$ 
        endfor
        Randomly select a plant  $j_t \in RCL1 = \{j \in J : v_j \leq v_{\min} + 0.5(v_{\max} - v_{\min})\},$ 
         $\hat{T} := \hat{T} \setminus I_{j_t t},$ 
         $\hat{p}^t := \hat{p}^t + 1,$ 
        if ( $\hat{p}^t = p^t$ ) then  $\hat{T} := \hat{T} \setminus \{t\},$ 
    endwhile
endfor

```

Fig. 2 GRASP Heuristic 2 for building the initial set of columns

zon, and the total number of customers. For each of these factors, we consider different levels that define our battery of test instances: $|J|$ varies in $\{10, 15, 30\}$ and $|T|$ in $\{4, 8, 12\}$. Finally, different values of $|I|$ have been considered for each value of $|T|$. For $|T| = 4,$ $|I|$ varies in $\{100, 150, 200, 500\}$; for $|T| = 8,$ $|I|$ varies in $\{150, 200, 500\}$; for $|T| = 12,$ $|I|$ varies in $\{200, 500\}$.

In addition, for each combination of $|T|$ and $|I|,$ we have considered two different values for the maximum capacity of the facilities, $H,$ a small capacity, and a large capacity (see Tables 1 and 2). For each combination of factors, levels, and capacities, we have generated 10 instances. In total we have generated 540 instances using the following structure:

- Opening costs in the first period drawn from a uniform distribution in $[75, 125]$ and opening costs in period $t > 1,$ $f_j^t,$ are given by $f_j^t = f_j^{t-1} + U[-7, 7].$
- Assignment costs in the first period drawn from a uniform distribution in $[10, 50]$ and assignment costs in period $t > 1,$ $c_{ij}^t,$ are given by $c_{ij}^t = c_{ij}^{t-1} + U[-5, 5].$
- For each instance, the value $p^t, t \in T,$ was generated as the minimum between the closest integer to $\frac{|I|}{|T|H} U[0.9, 1.3]$ and $|J|.$

For a better evaluation of our results, we have used CPLEX to get the values of the optimal solutions to the instances and the cpu time required to obtain the optimal solution using model *M1*. For these experiments, a maximum of four hours (14000 seconds) of cpu time was fixed. Tables 1 and 2 depict information on the above described instances and on the average results obtained in each of their meaningful aspects. In particular, Table 1 depicts the average percent deviations of the obtained lower and upper bounds with respect to the optimal/best solution found by CPLEX,

Table 1 Lower and upper bounds percent deviations with respect to the optimum

T	I	H	J = 10			J = 15			J = 30							
			o-lb		ub-o		o-lb		ub-o		o-lb		ub-o			
			LR ₁ /LP	LR ₂ /GC	GC	LR ₂	LR ₁ /LP	LR ₂ /GC	GC	LR ₂	LR ₁ /LP	LR ₂ /GC	GC	LR ₂		
4	100	3	22.10	0.00	1.00	0.00	0.00	20.27	0.01	2.51	0.00	18.15	0.01	5.29	0.00	
		5	33.27	0.08	2.89	0.01	30.84	0.07	4.87	0.03	27.90	0.09	27.90	0.09	9.10	0.04
	150	5	32.08	0.02	1.58	0.00	30.09	0.03	3.73	0.01	27.29	0.05	27.29	0.05	7.47	0.02
		10	50.16	0.15	4.19	0.13	47.78	0.13	6.20	0.16	44.42	0.28	44.42	0.28	8.67	0.84
	200	8	42.70	0.03	1.90	0.02	40.97	0.08	3.99	0.09	37.50	0.12	37.50	0.12	6.88	0.30
		15	60.79	0.30	5.30	0.51	58.40	0.44	6.50	1.22	*	*	*	*	*	*
500	15	58.61	0.03	1.32	0.00	55.85	0.09	2.85	0.15	51.20	0.31	51.20	0.31	4.87	1.06	
	30	75.14	0.46	2.64	2.09	72.86	0.44	3.97	2.94	70.41	2.19	70.41	2.19	4.77	3.54	
	3	19.36	0.01	1.65	0.00	17.75	0.01	3.50	0.01	15.80	0.03	15.80	0.03	6.86	0.01	
8	150	5	29.72	0.07	3.87	0.10	26.99	0.07	6.21	0.03	24.63	0.09	24.63	0.09	9.49	0.24
		5	28.88	0.05	3.08	0.01	26.57	0.04	5.18	0.05	24.29	0.08	24.29	0.08	8.48	0.20
	10	46.64	0.33	6.18	0.81	44.03	0.36	8.08	1.29	42.02	0.81	42.02	0.81	10.26	2.36	
500	8	36.80	0.01	1.21	0.00	35.32	0.05	2.88	0.08	32.38	0.10	32.38	0.10	5.59	0.31	
	15	54.71	0.19	3.28	0.71	52.88	0.51	4.96	1.65	50.27	1.33	50.27	1.33	6.94	2.94	
12	200	3	17.36	0.01	1.96	0.00	16.44	0.02	4.47	0.01	15.20	0.02	15.20	0.02	6.55	0.03
		5	27.36	0.07	4.62	0.13	25.66	0.06	7.35	0.09	23.93	0.10	23.93	0.10	10.32	0.26
	500	5	24.40	0.00	0.82	0.00	23.16	0.01	2.40	0.02	22.75	0.02	22.75	0.02	6.45	0.02
		10	41.28	0.12	3.31	0.38	39.64	0.23	4.89	0.70	39.00	0.62	39.00	0.62	8.18	1.18

Table 2 Average CPU times (in seconds)

T	J	H	J = 10				J = 15				J = 30				
			J = 10		J = 15		J = 15		J = 30		J = 30		J = 30		
			LR ₁	opt	No	gc	LR ₂	LR ₁	opt	No	gc	LR ₂	LR ₁	opt	No
4	100	3	9.2	0.0	0.32	0.8	11.1	0.5	0.40	5.2	12.8	1.0	0.76	2.7	
		5	8.4	1.1	0.74	5.3	9.0	1.7	0.86	5.9	11.3	6.5	1.12	14.3	
	150	5	11.8	0.4	1.90	4.5	13.4	2.0	1.48	10.6	16.6	10.5	2.00	19.3	
		10	9.0	8.1	4.28	9.2	10.6	17.2	4.54	12.2	14.3	82.7	4.70	19.2	
	200	8	12.4	3.1	7.26	13.6	14.5	16.8	5.78	17.1	19.7	91.0	5.94	26.5	
		15	10.0	27.2	47.70	12.1	12.3	634.7	36.24	15.1	17.9	12754.9	4	27.78	25.9
500	15	15	24.7	6.8	2182.78	21.1	30.3	398.1	273.64	38.5	48.9	14400.8	10	157.76	66.0
		30	19.6	835.5	2123.38	27.9	25.0	12519.7	8	1617.28	35.2	43.9	14400.3	10	636.16
	8	150	3	20.3	0.8	1.36	10.5	24.0	2.3	1.56	8.7	9.8	2.30	21.0	
		5	16.7	8.1	1.76	17.8	19.3	17.9	2.06	21.6	26.3	78.0	2.96	35.0	
	200	5	22.0	8.5	3.16	23.9	25.8	27.9	3.68	30.0	38.1	237.7	4.82	48.2	
		10	17.1	240.6	9.34	20.5	20.6	826.7	9.04	26.2	33.8	13310.6	8	9.58	46.6
500	8	49.6	9.9	203.32	47.2	64.8	497.2	84.66	73.6	101.8	12439.9	6	80.28	126.0	
	15	37.7	8373.2	3	140.70	46.7	51.9	14400.5	10	146.50	14401.6	10	164.70	121.6	
12	200	3	35.1	2.8	3.58	25.7	40.7	61.5	6.04	71.2	8.0	4.28	38.1		
		5	26.2	60.8	3.96	29.1	32.0	570.2	6.04	69.3	54.2	76.4	4.50	39.4	
	500	5	93.5	3.2	101.96	63.5	113.9	910.2	124.16	197.9	167.2	99.7	107.78	113.6	
		10	60.5	8760.0	4	100.68	68.8	81.4	14407.4	10	130.84	14400.8	10	107.32	97.2

whereas Table 2 gives information about the average cpu times in seconds. In both tables, the rows are grouped into blocks, each of them corresponding to a fixed number of periods $|T|$. The rows within each block correspond to instances with a fixed number of customers $|I|$ and a fixed value for the maximum capacity of the plants H . The columns are grouped into three blocks, each of them corresponding to a fixed number of facilities $|J|$. Thus, each entry in the tables corresponds to the average value over 10 instances of a fixed dimension.

The columns within each column block of Table 1 are, in turn, grouped into two blocks. The first one (labeled as $o - lb$) gives the average percent gaps between the optimal/best solution found with CPLEX and the two obtained lower bounds: 1) the entries labeled as LR_1/LP correspond to the Lagrangean bound associated with $L1(u)$ (which coincides with the LP bound of program $M1$); and 2) the entries labeled as LR_2/GC correspond to the Lagrangean bound associated with $L2(v)$ (which coincides with the LP bound of program $M2$). The second block (labeled as $ub - o$) gives the average percent gaps between the optimal/best solution found with CPLEX and the two obtained upper bounds: 1) the upper bound obtained with the initial set of columns used for the column generation approach to $M2$ (entries labeled GC); and 2) the best solution found with the heuristic applied to the solutions of $L2(v)$ (entries labeled LR_2). Entries with an asterisk mean that either CPLEX could not find any feasible solution for any of the 10 instances of the corresponding dimension within the allowed cpu time or that a memory overflow occurred.

As can be seen, the obtained gaps of the LP relaxation of program $M1$ are very large, especially for instances with large facility capacities, where this gap exceeded 50% in several instance sets. This motivated the study of a different Lagrangean relaxation and another reformulation based on set partitioning. Since these two approaches have dramatically outperformed in all cases the first Lagrangean relaxation, in the following we just report on the results of these latter approaches.

The columns within each column block of Table 2 give: 1) the cpu time required to solve the Lagrangean Dual $D1$ (entries labeled LR_1); 2) the cpu time required by CPLEX to obtain the optimal solution with an upper limit of four hours (entries labeled opt); 3) the number of instances out of the 10 instances of the same dimension for which CPLEX could not prove optimality of the best solution found after the limit of four hours of cpu time (entries labeled No); 4) the cpu time required to solve the LP relaxation of program $M2$ using column generation (entries labeled gc); and 5) the cpu time required to solve the Lagrangean Dual $D2$ (entries labeled LR_2). These experiments have shown the difficulty of large size problems, since CPLEX fails to prove the optimality of the best solution found in the four hours of cpu time in most of the large size instances ($|I| = 500$ or $|J| = 30$ and $|I| \geq 200$).

The results obtained when solving the LP relaxation of program $M2$ using column generation and the Lagrangean Dual $D2$ are good, both in terms of the obtained bounds and of the required computation times. It is worth noting that the obtained average percent gaps between the optimal/best solution found with CPLEX and the lower bounds obtained with these two approaches are very small; indeed, they are under 1% in all but two cases, where they do not exceed 2.20%. On the other hand, the average percent gaps between the upper bounds and the optimal/best solution found

with CPLEX depend on the heuristic used. For the heuristic that builds the initial set of columns of program $M2$, these gaps range between 0.82% and 10.32%, whereas for the heuristic used to obtain feasible solutions when solving the Lagrangean Dual $D2$, the final gaps range between 0% and 3.54%. Actually, the best feasible solution found with this relaxation is optimal for 178 out of the 540 instances, and, in general, this last solution method is able to find feasible solutions which can be proven to be within a small gap from the optimum. From this observation we conclude that the latter method seems to be the most appropriate to solve this problem. This assertion is also supported by the observed cpu times with both methods. Nevertheless, one may observe that the column generation approach requires, in some occasions, less cpu time than the Lagrangean one on some instances of small to medium sizes. In spite of that, the Lagrangean method shows less variability in terms of the time so that we considered this one as the best method. For this reason, computational results on larger instances are only given for this approach.

Table 3 shows the average percent gaps between the upper and the lower bounds obtained when solving the Lagrangean Dual $D2$ and the cpu times required for solving instances with 700 and 1000 customers. The entries in Table 3 indicate that the computational burden required to obtain these upper and lower bounds is small, taking into account the size of the instances. The largest cpu time, which is around 6.5 minutes, corresponds to instances with $|J| = 30$, $|I| = 1000$, and $|T| = 12$. Note that, for these instances, formulation $M1$ has more than 360000 variables. Our experiments show that CPLEX was not able to find the optimal solutions of these problems except for small size instances (within the allowed cpu time limit). This indicates that solving these instances exactly is extremely time consuming and that our approach is an appropriate alternative.

Table 3 Average gaps and CPU time for LR_2

$ T $	$ I $	H	$ J = 10$		$ J = 15$		$ J = 30$	
			gap ub-lb	CPU	gap ub-lb	CPU	gap ub-lb	CPU
4	700	20	0.02	30.90	0.42	52.80	2.26	97.30
		35	1.71	37.10	3.33	49.40	5.64	88.80
	1000	30	0.03	47.40	1.04	73.00	3.62	137.50
		50	1.13	49.80	2.74	65.10	5.53	124.40
8	700	15	0.30	66.30	1.40	98.10	3.16	176.10
		30	4.41	60.50	6.70	90.50	9.14	166.50
	1000	20	0.25	99.10	2.01	140.80	4.03	262.40
		35	4.68	94.70	6.80	131.80	9.68	244.50
12	700	8	0.02	107.10	0.16	158.00	0.27	273.60
		15	1.26	96.10	2.75	133.40	3.61	251.40
	1000	15	0.47	147.20	1.25	206.10	2.33	387.10
		30	5.68	131.60	7.81	189.40	9.51	375.10

5 Conclusions

In this work we have presented the Single Period Coverage Facility Location Problem. It is a dynamic facility location problem where service to each customer needs to be provided in exactly one time period within the planning horizon. To provide these services, a minimum number of capacitated plants are opened at each time period (among the entire set of candidate sites) so that their locations can be chosen in different sites at different time periods.

For this problem, two different formulations have been studied, and different bounding procedures based on these formulations are presented. In particular, for the first model, we have studied two Lagrangean relaxations, and for the second model, we have proposed a column generation approach to obtain its LP bound, given that its number of variables increases exponentially with the number of customers. Additionally, we have proved that its LP bound coincides with the lower bound associated with one of the previous Lagrangean duals.

Extensive computational experiments have been carried out to evaluate the proposed methods. Our results indicate that the Single Period Coverage Facility Location Problem is very difficult to solve using standard optimization software, which gives an extra motivation to develop heuristics and lower bounds. The results show that the proposed procedures allow one to identify efficiently good quality feasible solutions and that the lower bound given by the second Lagrangean relaxation (and, thus, the LP bound of the second formulation) is very tight. However, our results indicate that it is more efficient to find this common lower bound through the Lagrangean relaxation of the first model.

Acknowledgements This research is partially supported by Spanish Ministry of Science and Education grants: MTM2004-22566-E, MTM2004-0909, and MTM2006-14961-C05-01, MTM2007-67433-C02-01 and P06-FQM-01366, and through ERDF funds. These supports are gratefully acknowledged.

References

- Albareda-Sambola M, Fernández E, Hinojosa Y, Puerto J (2009) The multi-period incremental service facility location problem. *Comput Oper Res* 36(5):1356–1375
- Chardaire P, Sutter A, Costa MC (1996) Solving the dynamic facility location problem. *Networks* 28:117–124
- Cornuejols G, Nemhauser GL, Wolsey LA (1990) The uncapacitated facility location problem. In: Mirchandani PB, Francis RL (eds) *Discrete location theory*. Wiley, New York, pp 119–171
- Correia I, Gouveia L, Saldanha da Gama F (2008) Solving the variable size bin packing problem with discretized formulations. *Comput Oper Res* 35(6):2103–2113
- Cortinhal MJ, Captivo ME (2003) Upper and lower bounds for the single source capacitated location problem. *Eur J Oper Res* 151(2):333–351
- Current J, Ratick S, ReVelle C (1997) Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. *Eur J Oper Res* 110:597–609
- Daskin MS, Hopp WJ, Medina B (1992) Forecast horizons and dynamic facility location planning. *Ann Oper Res* 40:125–152
- Drezner Z (1995) Dynamic facility location: the progressive p -median problem. *Location Sci* 3:1–7
- Feo TA, Resende MGC (1995) Greedy randomized adaptive search procedures. *J Glob Optim* 6:109–133
- Galvão RD, Santibañez-González ER (1992) A Lagrangean heuristic for the p_k -median dynamic location problem. *Eur J Oper Res* 58:250–262
- Geoffrion AM (1974) Lagrangean relaxation for integer programming. *Math Program Study* 2:82–114

- Guignard M (2003) Lagrangean relaxation. *TOP* 11(2):151–228
- Hinojosa Y, Puerto J, Fernández FR (2000) A multiperiod two-echelon multicommodity capacitated plant location problem. *Eur J Oper Res* 123(2):45–65
- Hinojosa Y, Kalcsics J, Nickel S, Puerto J, Velten S (2008) Dynamic supply chain design with inventory. *Comput Oper Res* 35(2):373–391
- Melo MT, Nickel S, Saldanha-da-Gama F (2006) Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Comput Oper Res* 33(1):181–208
- Melo MT, Nickel S, Saldanha-da-Gama F (2009) Facility location and supply chain management—A review. *Eur J Oper Res* 196(2):401–412
- Mirchandani PB, Francis RL (eds) (1990) *Discrete location theory*. Wiley, New York
- Sridharan R (1995) The capacitated plant location problem. *Eur J Oper Res* 87(2):203–213
- van Roy TJ, Erlenkotter D (1982) A dual-based procedure for dynamic facility location. *Manag Sci* 28(10):1091–1105
- Warszawski A (1973) Multi-dimensional location problems. *Oper Res Q* 24:165–179