Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 36 (2009) 1356–1375
computers & operations research
www.elsevier.com/locate/cor

The multi-period incremental service facility location problem

Maria Albareda-Sambola[a,*], Elena Fernández[a], Yolanda Hinojosa[b], Justo Puerto[c]

[a] *Statistics and Operations Research Department, Technical University of Catalonia, Spain*
[b] *Departamento Economía Aplicada I, Universidad de Sevilla, Spain*
[c] *Facultad de Matemáticas, Universidad de Sevilla, Spain*

Available online 10 March 2008

Abstract

In this paper we introduce the multi-period incremental service facility location problem where the goal is to set a number of new facilities over a finite time horizon so as to cover dynamically the demand of a given set of customers. We prove that the coefficient matrix of the allocation subproblem that results when fixing the set of facilities to open is totally unimodular. This allows to solve efficiently the Lagrangean problem that relaxes constraints requiring customers to be assigned to open facilities. We propose a solution approach that provides both lower and upper bounds by combining subgradient optimization to solve a Lagrangean dual with an ad hoc heuristic that uses information from the Lagrangean subproblem to generate feasible solutions. Numerical results obtained in the computational experiments show that the obtained solutions are very good. In general, we get very small percent gaps between upper and lower bounds with little computation effort.

Keywords: Discrete facility location; Lagrangean dual; Multiperiod location

1. Introduction

Locational decisions are proven to be among the most important strategic decisions in the design and management of supply chains since they have a long lasting effect on a company. Therefore, developing efficient tools to guide the locational phase of the decision-making process is crucial to improve supply chain planning and control.

Multi-period location problems are being investigated since the early papers by Warszawski [1] and van Roy and Erlenkotter [2], up to the more recent references by Daskin et al. [3], Galvão and Santibañez-González, [4] and by Current et al. [5], 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. [6], Drezner [7], Hinojosa et al. [8,9]). However, some modeling aspects still require further attention. In this work we shall focus on one of them. In particular, how to address incremental customer service within a planning horizon.

In actual world, it is usual to look for sequential decisions that ensure certain level of coverage at each time period. In the case of essential services, it is required that the population demand be serviced from the first time period. Different works have addressed situations of this type (see the above references). However, in the case of non-essential services

* Corresponding author.
E-mail addresses: maria.albareda@upc.edu (M. Albareda-Sambola), e.fernandez@upc.edu (E. Fernández), yhinojos@us.es (Y. Hinojosa), puerto@us.es (J. Puerto).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.02.010

the full population needs to be serviced only at the end of the planning horizon. Instead, strategic and managerial decisions require that 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. We are not aware of any previous work that addressed this type of model.

It is clear that these types of problems represent actual applications. The simplest application is the sequential optimal location plan and distribution pattern of a good or service to reach full coverage of a set of potential customers in a finite time horizon. This is, for instance, the case of non-essential services like libraries, nursing homes, kindergartens, parking lots, supermarkets, banks, etc., that are planned to provide service to the entire set of potential customers. It is clear that the nature of these applications requires that service to users cannot be interrupted, in the remaining planning horizon, once it has been started. Nevertheless, budget constraints avoid to achieve complete coverage in one single period and therefore, an optimal plan is needed to fulfil the full service goal while ensuring partial achievements at minimum cost over the whole planning horizon.

In this paper we introduce the Multi-period Incremental Service Facility Location Problem, MISFLP for short. The considered problem is quite general and it includes as particular cases some interesting difficult problems. Mainly, we focus on the modeling issues and we propose a solution procedure for MISFLP. The MISFLP consists of minimizing the total costs throughout a finite time planning horizon while ensuring that at each single period, t, a minimum number of customers, n^t, is served. We accept that the allocation of any customer to the servers might change in different time epochs. Nevertheless, once a customer is being served in a period he/she must be served at any subsequent period. Moreover, we assume that exactly p^t new facilities are opened in each time period and that once a facility is opened at any time period, it remains open until the end of the planning horizon. The problem belongs to the class *NP*-hard, since it reduces to the well-known p-median problem [10] when the planning horizon shrinks to one single period.

We show that, in the general case, the allocation subproblem that results when the set of facilities to be opened is fixed can be solved in polynomial time, since its coefficient matrix is totally unimodular. Occasionally, the structure of this subproblem can be further exploited. For instance, for non-negative assignment costs it reduces to a transportation problem, and for the version of the problem when there is no minimum required number of customers to be served per period but there is a profit associated with serviced customers, it reduces to a semi-assignment problem [11]. This allows solving efficiently the Lagrangean problem that relaxes constraints requiring customers to be assigned to open facilities. 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.

Finally, we have run a series of computational experiments, in order to assess the efficiency of our algorithm. We have seen that the problems are difficult to solve and that for medium size instances CPLEX might not be able to obtain the optimal solution within 2 h of cputime. Moreover, in some cases CPLEX is not able to find any feasible solution within the above mentioned cputime. 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. For analyzing the structure of the solutions obtained with our model we also compare the generated solutions with those obtained by solving a series of independent decoupled problems. We also use our model within a what-if scenario analysis for finding appropriate number of periods within the planning horizon yielding to a tradeoff between cost and service level.

The paper is organized as follows. Section 2 describes a mixed integer mathematical programming (MIP) formulation of MISFLP. Section 3 is devoted to the allocation subproblem that results when the set of facilities that are open in each time period is fixed. As we shall see in the paper, this problem is instrumental in our approach to solve MISFLP. In Section 4 we develop a Lagrangean relaxation to get lower bounds on our MIP formulation. We give several results describing the components of the subgradients and the integrality property of our relaxation. The next section deals with our solution approach. Our method combines dual steps in a subgradient algorithm with a heuristic for obtaining improved feasible solutions of MISFLP. Section 6 reports on the results obtained from our computational experiments. In this regard, we have tested several batteries of problems that show the good performance of our algorithm both in cputime and gap. This section also includes an analysis of the solutions yielded by MISFLP as compared to the ones obtained by solving a sequence of decoupled problems for each period, and some insight on how our model can be used as a tool for deciding the number of periods that the planning horizon should have. The paper ends with some conclusions on the models, solution approach and challenging open problems.

2. A mathematical programming formulation of MISFLP

In this section we present the formal description of MISFLP 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. The modeling hypotheses that require a minimum service level per period are stated by means of a minimum number of customers to be served and a number of new plants to be opened. As explained above, the nature of the problem also imposes continuity on customer service.

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

- I: set of customers, indexed by $i \in I$,
- J: set of possible locations for facilities, indexed by $j \in J$,
- T: set of time periods, indexed by $t \in T$.

For each period $t \in T$, define

- n^t: minimum number of customers that must be served at period t,
- p^t: number of facilities that must be opened 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 : total cost of facility j being established at time period t.

This cost includes the opening cost at time period t and the maintenance cost from time period t to the end of the planning horizon. Note that there is no sign constraint on any of the above costs.

We further assume that once a facility is opened, it remains open until the end of the planning horizon. Also, if a customer is served for the first time at period $\hat{t}$, it must be served in all subsequent periods $t \geqslant \hat{t}$ and all customers have to be serviced at the end of the planning horizon. Served customers can be allocated to different open facilities at different periods. Since allocation costs vary with the time periods, customers will patronize different servers in different periods according to a lower cost policy. 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.

Note that MISFLP is quite general and it has as particular cases problems with apparently different characteristics. First of all, a simple transformation of the data will allow to solve problems in which it is not necessary to serve all demands at the end of the planning horizon. This could be done by introducing a dummy facility with:

$$f_j^t = \begin{cases} \infty & t < T \\ 0 & t = T \end{cases} \quad \text{and} \quad c_{ij}^t = \begin{cases} \infty & t < T, \\ \gamma & t = T, \end{cases}$$

where γ is a penalty for leaving a costumer unassigned at the end of the planning horizon. On the other hand, one can also introduce penalties for not servicing the customers in the different periods. It is straightforward that after some rearrangements the resulting objective function is similar to that of our model. In particular, choosing these penalties large enough will force satisfying the demand of all customers in all periods. When the planning horizon reduces to one single period, and the setup costs are zero, we obtain the well-known p-median problem. For a general planning horizon, when the assignment values c are non-negative and there is a pre-specified coverage level at each intermediate period ($n^t > 0$), the resulting problem can be seen as a cost-minimizing problem. Note that also the profit-maximizing version of the problem where there is no required coverage level at each intermediate period ($n^t = 0$), but there is a profit g_i^t associated with serviced customers, fits within MISFLP. The potential applications described in the Introduction, corresponding to private services, fall within this category. In this case, the profit for servicing a customer can be incorporated to its assignment value by defining transformed allocation values $\hat{c}_{ij}^t = c_{ij}^t - g_i^t$. (Notice that the transformed costs might be negative.) We remark that, for the reasons given in the Introduction, also in the profit-maximizing model the constraints that ensure continuity in service are required. In spite of not having a minimum coverage level per period ($n^t = 0$, for all t), costumers will be served starting from some period when the decision is profitable on the overall planning horizon.

Since MISFLP includes as a particular case the p-median problem, it belongs to the class of *NP*-hard problems, see Kariv and Hakimi [10].

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

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

Using these conventions, the mathematical formulation of MISFLP is

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

$$\text{s.t.} \quad \sum_{i\in I}\sum_{j\in J} x_{ij}^t \geqslant n^t \quad \forall t \in T, \tag{2}$$

$$\sum_{j\in J} x_{ij}^t \leqslant 1 \quad \forall i \in I, \ \forall t \in T, \tag{3}$$

$$\sum_{j\in J} x_{ij}^t \geqslant \sum_{j\in J} x_{ij}^{t-1} \quad \forall i \in I, \ \forall t \in T, \ t > 1, \tag{4}$$

$$\sum_{j\in J} x_{ij}^{|T|} = 1 \quad \forall i \in I, \tag{5}$$

$$x_{ij}^t \leqslant \sum_{k\leqslant t} y_j^k \quad \forall i \in I, \ \forall j \in J, \ \forall t \in T, \tag{6}$$

$$\sum_{j\in J} y_j^t = p^t \quad \forall t \in T, \tag{7}$$

$$\sum_{t\in T} y_j^t \leqslant 1 \quad \forall j \in J, \tag{8}$$

$$x_{ij}^t \in \{0,1\} \quad \forall i \in I, \ \forall j \in J, \ \forall t \in T, \tag{9}$$

$$y_j^t \in \{0,1\} \quad \forall j \in J, \ \forall t \in T. \tag{10}$$

The first set of constraints (2) are the covering constraints that impose that in each time period the minimum required number of customers are being served. Constraints (3) ensure that each customer is assigned to at most one facility in each period. Constraints (4) guarantee that once that a customer is served at a given period he/she will be served in any subsequent period. Next set of constraints (5) ensures that all customers are served at the end of the planning horizon. Constraints (6) represent the fact that customers are assigned only to open facilities. The following group of constraints (7) models that exactly p^t facilities are opened in each time period whereas constraints (8) impose that a facility can be opened, at most, once. The last constraints (9) and (10) define the binary variables. The objective function (1) minimizes the overall cost for opening the plants and allocating the customers satisfying the required hypotheses.

3. The allocation 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 the coefficient matrix is totally unimodular. This property will be used later on in our solution approach for MISFLP.

Throughout this section we suppose that we know the set of facilities that are open in each time period. For each $t \in T$ let J^t denote the set of open facilities in time period t and for each $i \in I$ let d_{it} be the total cost for allocating

customer i in period t for the first time. Note that if customer i is allocated for the first time at t, in this period and in any subsequent period he/she will be assigned to the cheapest open facility. Therefore,

$$d_{it} = \sum_{k=t}^{|T|} \left(\min_{j \in J^k} c_{ij}^k \right) \quad \forall i \in I, \ \ \forall t \in T.$$

Hence, we have the following result.

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 whose coefficient matrix is totally unimodular*:

$$(ASP) \quad \min \quad \sum_{t \in T} \sum_{i \in I} d_{it} z_{it} \tag{11}$$

$$\text{s.t.} \quad \sum_{t \in T} z_{it} = 1 \quad \forall i \in I, \tag{12}$$

$$\sum_{i \in I} \sum_{r=1}^{t} z_{ir} \geqslant n^t \quad \forall t \in T, \tag{13}$$

$$z_{it} \geqslant 0 \quad \forall i \in I, \ \ \forall t \in T, \tag{14}$$

where

$$z_{it} = \begin{cases} 1 & \textit{if customer i is allocated in period t for the first time}, \\ 0 & \textit{otherwise}. \end{cases}$$

Proof. When the set of facilities to open is fixed, a model for the allocation subproblem can be directly obtained from (1)–(10):

$$\min \quad \sum_{t \in T} \sum_{j \in J^t} \sum_{i \in I} c_{ij}^t x_{ij}^t \tag{15}$$

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

$$\sum_{j \in J^t} x_{ij}^t \leqslant 1 \quad \forall i \in I, \ \ \forall t \in T, \tag{17}$$

$$\sum_{j \in J^t} x_{ij}^t \geqslant \sum_{j \in J^{t-1}} x_{ij}^{t-1} \quad \forall i \in I, \ \ \forall t \in T, \ \ t > 1, \tag{18}$$

$$\sum_{j \in J^{|T|}} x_{ij}^{|T|} = 1 \quad \forall i \in I, \tag{19}$$

$$x_{ij}^t \in \{0, 1\} \quad \forall i \in I, \ \ \forall j \in J^t, \ \ \forall t \in T. \tag{20}$$

However, from the above discussion it is straightforward that the allocation subproblem can also be formulated by means of model ASP, whose coefficient matrix is given by

$$A = \begin{bmatrix} 1 & & & 1 & & & \cdots & 1 & & \\ & \ddots & & & \ddots & & \cdots & & \ddots & \\ & & 1 & & & 1 & \cdots & & & 1 \\ 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 1 & \cdots & 1 & 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ & \vdots & & & \vdots & & & & \vdots & \\ 1 & \cdots & 1 & 1 & \cdots & 1 & \cdots & 1 & \cdots & 1 \end{bmatrix}.$$

This matrix can be transformed into a transportation matrix by subtracting consecutively rows. The operation consists of subtracting the row $|I|+k-1$ from the row $|I|+k$ for $k=|T|,\ldots,2$. Since, row addition maintains the value of determinants and transportation matrices are totally unimodular (T.U.) we get that A is T.U. as well. □

Therefore, one can see that the opening pattern given by J^t for $t \in T$ and the allocation scheme given by ASP generate a feasible solution for the original problem (1)–(10). In the next sections we will exploit this result to solve efficiently the Lagrangean dual that we will consider, and to obtain feasible solutions for our original problem.

In the particular case of ASP where for each $i \in I$ the cost coefficients d_{it} are non-increasing with respect to t, there is an optimal solution to ASP where all the constraints (13) hold as equality. In this case, ASP admits a simpler transformation into a transportation problem, by subtracting consecutive rows of constraints (13) as indicated in the proof to Proposition 1. Note that when the allocation costs are non-negative the above condition holds. Thus we have the following:

Corollary 1. *If the set of open facilities in period t, J^t, is known for $t \in T$ and $c^t_{ij} \geqslant 0$ for all i, j, t then the optimal allocation for the customers can be obtained by solving the transportation problem*:

$$
\begin{aligned}
(TP)\quad \min\quad & \sum_{t\in T}\sum_{i\in I} d_{it} z_{it} \\
s.t.\quad & \sum_{t\in T} z_{it} = 1 \quad \forall i \in I, \\
& \sum_{i\in I} z_{it} = n^t - n^{t-1} \quad \forall t \in T, \\
& z_{it} \geqslant 0 \quad \forall i \in I,\ \ \forall t \in T,
\end{aligned}
$$

where

$$
z_{it} = \begin{cases} 1 & \textit{if customer } i \textit{ is allocated in period } t \textit{ for the first time}, \\ 0 & \textit{otherwise}. \end{cases}
$$

Also in the case of the profit-maximizing problem, the ASP can be transformed into a much simpler problem. Set

$$
d_{it} = \sum_{k=t}^{|T|} \left(\min_{j\in J^k} (c^k_{ij} - g^k_i) \right) \quad \forall i \in I,\ \ \forall t \in T.
$$

Hence, since in the profit-maximizing model $n^t = 0$ for all t, constraints (13) are redundant because they reduce to a summation of binary variables greater than or equal to 0. In this case the optimal allocation for the customers can be obtained by solving the following series of independent semi-assignment problems [11]:

$$
\left.
\begin{aligned}
(KST)\quad \min\quad & \sum_{i\in I}\sum_{t\in T} d_{it} z_{it} \\
\text{s.t.}\quad & \sum_{t\in T} z_{it} = 1, \quad \forall\ i \in I, \\
& z_{it} \in \{0,1\}, \quad \forall i \in I,\ \ t \in T.
\end{aligned}
\right\}
\Leftrightarrow
\left\{
\begin{aligned}
\sum_{i\in I}\quad \min\quad & \sum_{t\in T} d_{it} z_{it} \\
\text{s.t.}\quad & \sum_{t\in T} z_{it} = 1, \\
& z_{it} \in \{0,1\}, \quad \forall t \in T.
\end{aligned}
\right.
$$

Notice that the above problem can be solved by inspection taking $z_{it^*_i} = 1$ where $t^*_i \in \arg\min_t \{d_{it}\}$ (arbitrarily selected) and $z_{it} = 0$ for all $t \neq t^*_i$, and for all $i \in I$.

4. Lagrangean relaxation

In this section, we consider the Lagrangean relaxation of the problem obtained by relaxing the constraints (6) into the objective function. Let $u^t_{ij} \geqslant 0$ denote the non-negative multipliers, then the relaxed problem, denoted by $L(u)$

is given by

$$L(u) = \min \quad \sum_{t\in T}\sum_{j\in J}\left(\sum_{i\in I} c_{ij}^t x_{ij}^t + f_j^t y_j^t\right) + \sum_{t\in T}\sum_{i\in I}\sum_{j\in J} u_{ij}^t \left(x_{ij}^t - \sum_{k\leqslant t} y_j^k\right)$$

$$\text{s.t.} \quad (2), (3), (4), (5), (7), (8), (9), (10).$$

After some algebra the objective function turns out to be

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

This function can be split into two different functions. The first one depends only on the x variables, while the second one only on the variables y. In addition, constraints (2)–(5) and (9) only relate x variables, while y variables appear alone in constraints (7), (8) and (10). Thus, $L(u)$ separates into two subproblems. The first problem results in:

$$L_x(u) = \min \quad \sum_{t\in T}\sum_{j\in J}\sum_{i\in I}(c_{ij}^t + u_{ij}^t)x_{ij}^t \tag{21}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{j\in J} x_{ij}^t \geqslant n^t \quad \forall t \in T, \tag{22}$$

$$\sum_{j\in J} x_{ij}^t \leqslant 1 \quad \forall i \in I,\ \forall t \in T, \tag{23}$$

$$\sum_{j\in J} x_{ij}^t \geqslant \sum_{j\in J} x_{ij}^{t-1} \quad \forall i \in I,\ \forall t \in T,\ t > 1, \tag{24}$$

$$\sum_{j\in J} x_{ij}^{|T|} = 1 \quad \forall i \in I, \tag{25}$$

$$x_{ij}^t \in \{0, 1\} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in T. \tag{26}$$

This problem can be transformed into a problem of the form ASP, by just taking $J^t = J\ \forall t \in T$ and

$$d_{it} = \sum_{k=t}^{|T|}\left(\min_{j\in J}(c_{ij}^k + u_{ij}^k)\right) \quad \forall i \in I,\ \forall t \in T.$$

On the other hand, the second problem is

$$L_y(u) = \min \quad \sum_{t\in T}\sum_{j\in J}\left(f_j^t - \sum_{i\in I}\sum_{k\geqslant t} u_{ij}^k\right) y_j^t \tag{27}$$

$$\text{s.t.} \quad \sum_{j\in J} y_j^t = p^t \quad \forall t \in T, \tag{28}$$

$$\sum_{t\in T} y_j^t \leqslant 1 \quad \forall j \in J, \tag{29}$$

$$y_j^t \in \{0, 1\} \quad \forall j \in J,\ \forall t \in T, \tag{30}$$

which can be solved as a transportation problem. The following remarks apply to $L(u)$.

Remark 1. $L(u) = L_x(u) + L_y(u)$.

Remark 2. For any given $u \geqslant 0$, let $\tilde{x}$ and $\tilde{y}$ be the optimal solutions of $L_x(u)$ and $L_y(u)$, respectively. Then, a subgradient of $L(u)$ is given by

$$\delta_{ij}^t(u) = \tilde{x}_{ij}^t - \sum_{k \leqslant t} \tilde{y}_j^k = \begin{cases} -1 & \text{if customer } i \text{ is not assigned to facility } j \text{ that is open at } t, \\ 1 & \text{if customer } i \text{ is assigned to facility } j \text{ that is not open at } t, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, we can solve the Lagrangean dual $z_{\mathrm{LD}} = \max_{u \geqslant 0} L(u)$ with subgradient optimization.

Remark 3. $L(u)$ has the integrality property, since all the extreme points both in $L_x(u)$ and $L_y(u)$ are integral. Thus, $z_{\mathrm{LD}} = z_{\mathrm{LP}}$, where z_{LP} denotes the optimal value to the LP relaxation of (1)–(10).

Remark 3 might make us question whether it is computationally worthwhile to solve the Lagrangean dual, instead of the LP relaxation of (1)–(10). As we will see in the computational results section, as the size of the problems increases, the computation times required to solve the Lagrangean dual tend to be considerably smaller than the ones needed to solve the LP relaxation. This gives us a first justification for the selected approach. In addition, our Lagrangean approach provides us information to generate upper bounds for program (1)–(10).

4.1. Upper bounds

Assume that, for a given u, the Lagrangean problem $L(u)$ is solved. Then, a feasible solution for program (1)–(10) can be obtained by solving the allocation subproblem $\mathsf{ASP}(\tilde{y})$ associated with the set $J^t(\tilde{y})$ of open facilities at each period given by the optimal solutions $\tilde{y}$ to $L_y(u)$, that is

$$J^t(\tilde{y}) = \left\{ j \in J : \sum_{k \leqslant t} \tilde{y}_j^k = 1 \right\}. \tag{31}$$

Hence, an upper bound to (1)–(10) can be obtained as

$$val(\mathsf{ASP}(\tilde{y})) + \sum_{t \in T} \sum_{j \in J} f_j^t \tilde{y}_j^t,$$

where $val(P)$ stands for the optimal value of problem P.

In addition, at some steps throughout the algorithm we will use an alternative faster and easier heuristic, aiming to improve the incumbent upper bound. It gives an allocation of customers according to the facilities opened by $\tilde{y}$. Formally, it performs the following:

In each period $t \in T$, if $\tilde{x}_{ij}^t = 1$ and $j \notin J^t(\tilde{y})$, then reassign customer i to facility $j \in J^t(\tilde{y})$ with the minimum cost. That is:

$$j \in \arg\min \{c_{il}^t : l \in J^t(\tilde{y})\}.$$

We will refer to this heuristic procedure, by $HEUT(\tilde{y})$. Then, an upper bound to (1)–(10) can be obtained as

$$val(HEUT(\tilde{y})) + \sum_{t \in T} \sum_{j \in J} f_j^t \tilde{y}_j^t.$$

5. Algorithm

The method that we propose combines the solution of the Lagrangean dual LD with a process for finding feasible solutions to MISFLP. The lower bound z_{LD} of the previous section will be used to compute the gap of our approach. Furthermore, as we have already mentioned, the optimal y-variables of $L_y(u)$ can be used to derive a feasible solution to our original problem by solving the corresponding ASP problem.

Our algorithm consists of an Initialization step, where the initial upper and lower bounds, as well as the initial dual multipliers are set; and a single loop. This loop iterates until the stopping criterion is fulfilled. In each iteration we

perform dual steps of the subgradient algorithm trying to improve the lower bound, and we generate new feasible solutions of MISFLP based on the optimal solutions of $L_y(u)$, which are used to reduce the upper bound. Next we outline the procedure for solving our problem.

Algorithm 1
Initialization.
Set initial step length parameter $\beta = 2$.
Set initial multipliers $u_{ij}^t \quad \forall\, i \in I,\ \ j \in J$ and $t \in T$.
Solve $L(u)$ and let $(\tilde{x}, \tilde{y})$ be an optimal solution. Set the initial lower and upper bounds, respectively, to
$Z_{\mathrm{LB}} := L(u)$ and $Z_{\mathrm{UB}} := val(\mathsf{ASP}(\tilde{y})) + \sum_{t\in T}\sum_{j\in J} f_j^t \tilde{y}_j^t$.
While NOT (STOPPING CRITERION)
Determine the subgradient $\delta(u)$.
Compute the step length $\pi = \frac{\beta\cdot(Z_{\mathrm{UB}} - Z_{\mathrm{LB}})}{\langle \delta(u), \delta(u)\rangle}$.
If $(\pi < \varepsilon)$ then set β to the initial step parameter and recompute π.
Set $u_{ij}^t := \max\{u_{ij}^t + \pi \cdot \delta_{ij}^t(u), 0\}\ \forall\, i \in I,\ \ j \in J$ and $t \in T$.
Solve $L(u)$ and let $(\tilde{x}, \tilde{y})$ be an optimal solution.
If $(Z_{\mathrm{LB}} < L(u))$ set $Z_{\mathrm{LB}} := L(u)$.
Apply $(HEUT(\tilde{y}))$.
If $(Z_{\mathrm{UB}} > val(HEUT(\tilde{y})) + \sum_{t\in T}\sum_{j\in J} f_j^t \tilde{y}_j^t)$ set $Z_{\mathrm{UB}} := val(HEUT(\tilde{y})) + \sum_{t\in T}\sum_{j\in J} f_j^t \tilde{y}_j^t$.
If (MAXIMUM NUMBER OF SUBGRADIENT ITERATIONS WITHOUT IMPROVEMENT) then
Reduce the step length parameter $\beta := \beta/2$.
Solve ASP with set J^t defined by the current $\tilde{y}$ variables and update Z_{UB} accordingly.
End While

The initial choice of the multipliers depends on the problem type. We describe them in detail in Section 6, devoted to computational results.

In the above general description, there are some parameters that need to be specified. The precision parameter ε was taken equal to 10^{-3}, and the maximum number of subgradient iterations without improvement has been set to $|I|/5 + |J|$. In our implementation we have used several conditions as stopping criteria. The first one is the coincidence of the current upper and lower bounds, i.e. $Z_{\mathrm{LB}} = Z_{\mathrm{UB}}$. The second one is the optimality condition $\delta(u) \leqslant 0$ and $< \delta(u), u > = 0$. The third one checks for the convergence of the subgradient algorithm. Formally, let u^m be the dual multiplier in iteration m. According to the third criterion, the algorithm stops if there exists m such that $|L(u^{m+k}) - L(u^{m+k+1})| < 10^{-3}$, for $k = 1, \ldots, 5$. Finally, for avoiding unnecessary iterations, the algorithm also terminates when, after total number of at least 5000 iterations, the maximum number of iterations without improvement is reached. These values were set after some computational experiences, since they gave us the best trade-off between solution quality and computation time.

6. Computational study

The computational tests presented in this section have been designed in order to evaluate the structure of the solutions as well as the performance of the solution procedure developed in Section 5. On this account the algorithm was implemented using *Visual C++6.0*, where *ILOG CPLEX Collable Library* routines have been used for the implementation of the linear programs. Furthermore, *ILOG CPLEX* 8.1 has been used in order to solve these linear programs and to obtain exact solutions of the tested instances. Default parameters have been used. All computational tests have been performed on a PC with a *Pentium IV* processor with 2.0 GHz and 512 MB of RAM.

The experiments are separated into three parts. The first part has been divided into three separate blocks. Each of them is dedicated to MISFLP instances with different characteristics: (a) general planning horizon, with one new facility opened per period ($p^t = 1, \forall t$); (b) general planning horizon, with an arbitrary number of facilities opened per period ($p^t \geqslant 1, \forall t$); and (c) p-median instances. With blocks (a) and (b), we want to analyze the performance of the proposed solution method with respect to both the quality of the obtained solutions and the efficiency of the method. To the

best of our knowledge there are no available instances for these blocks. Thus, we have generated them as described later in this section. The experiments in block (c) have a different orientation. We note that our algorithm does not exploit specifically the structure of the p-median problem, and we want to evaluate the performance of our algorithm for well-known instances of this structured problem. For these experiments we have used the 40 uncapacitated p-median instances in the OR-Library [12].

The initial multipliers used for the experiments were taken in the following way. In those instances corresponding to blocks (a) and (b), for all i and t, we compute the average assignment value of allocating customer i at period t, i.e., let $m_i^t = \sum_{j\in J} c_{ij}^t/|J|$. Then, we set the initial multipliers as the non-negative deviation of c_{ij}^t with respect to that value:

$$u_{ij}^t = \max\{m_i^t - c_{ij}^t, 0\} \quad \forall\ i \in I,\quad j \in J \ \text{ and } \ t \in T.$$

In block (c) multipliers are set initially to zero, to avoid negative figures since no setup costs are present.

The second part of the experiments analyzes the structure of the solutions generated with our model, and compares it with that of the solutions obtained by solving a series of independent decoupled problems. Finally, in the third part, we use our model within a *what-if* scenario analysis for finding appropriate number of periods within the planning horizon yielding to a tradeoff between cost and service level.

6.1. General planning horizon, with one facility open per period

For generating these instances, we identify three relevant factors in the design of our experiment, namely the total number of customers, the number of candidate sites to open facilities and the planning horizon. For each of these factors we consider different levels that define our battery of test problems: $|I|$ varies in {50, 100, 150, 200, 500}, $|J|$ varies in {8, 10, 12, 15, 20, 30} and $|T|$ in {4, 5, 6, 7, 8, 10, 12} ($|J| \geqslant |T|$). For each combination of factors and levels we generate 10 instances. In total we have generated 1950 instances using the following structure:

- Setup costs drawn from a uniform distribution in [3000, 5000].
- Maintenance costs drawn from a uniform distribution in $[50|I|/|T|, 100|I|/|T|]$.
- Assignment costs drawn from a uniform distribution in [10, 100].
- The values of n^t follow a uniform distribution in $[n^{t-1}, |I|]$, for $t = 1, \ldots, |T| - 1$, $(n^0 = 1)$, whereas $n^{|T|} = |I|$.

Tables 1–3 depict information on these instances and on the average results obtained in each of their meaningful aspects. In particular, Table 1 summarizes the results of the algorithm of Section 5 for solving the Lagrangean dual. The rows of Table 1 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 $|I|$. The columns of Table 1 are separated into two blocks. The first block depicts the percent gaps (average and maximum) between the obtained upper and lower bounds ($100(Z_{UB} - Z_{LB})/z_{LB}$), whereas the second block gives the required cputimes in seconds. Within each block, each column corresponds to instances with a fixed number of facilities $|J|$. Thus, each entry in the tables corresponds to instances of a fixed dimension, and gives the average value over the 10 instances of this size. For the gaps also the maximum value over the 10 instances is reported.

As can be seen, the numerical results are very good, both in terms of the obtained gaps and of the required computation times. It is worth noting that the obtained percent gaps are very small and never exceed 2.75% for the average gap while the maximum among all the instances is 3.71%. This indicates that the duality gap of the considered instances is very small, but also that our solution method is able to find feasible solutions which can be proven to be within a very small gap from optimum. To a large extent these somewhat surprising results motivated our computational experiments with the well-known p-median instances. As will be seen later in this section the computational results with p-median instances are qualitatively similar to the ones given in Table 1. The figures in the second block of columns of Table 1 indicate the computational burden required to obtain these upper and lower bounds is small, taking into account the size of the instances. The largest cputime of 448.71 s corresponds to the largest instances with $|J| = 30$, $|I| = 500$ and $|T| = 12$. Note that these instances have more than 180 000 binary variables. (Fig. 1 illustrates the behavior of cputime of our algorithm as a function of $|I|$ for $|J| = 30$.)

Table 1
Percent gaps and cputimes for Algorithm 1.

		Percent gap												cputime					
		$\|J\|$												$\|J\|$					
		8		10		12		15		20		30		8	10	12	15	20	30
$\|T\|$	$\|I\|$	*av.*	*max*	*av.*	*max*	*av.*	*max*	*av.*	*max*	*av.*	*max*	*av.*	*max*						
4	50	0.09	0.49	0.14	0.58	0.24	0.70	0.43	0.90	0.30	0.90	0.45	0.90	2.69	3.12	5.32	5.71	6.18	9.03
	100	0.35	0.92	0.31	1.12	0.17	0.94	0.65	1.46	0.67	1.21	0.73	1.62	7.87	9.76	8.52	11.81	12.89	19.47
	150	0.20	0.75	0.35	1.03	0.45	1.05	0.50	0.97	1.10	1.88	1.10	2.20	13.69	15.92	16.86	23.73	21.81	28.03
	200	0.41	1.79	0.75	1.55	0.59	1.41	0.78	1.59	1.03	1.77	1.24	2.43	18.21	22.38	22.52	25.88	29.23	36.21
	500	0.27	1.23	0.63	1.41	0.98	1.91	1.04	2.33	1.09	2.55	1.16	2.54	55.78	68.41	65.55	87.46	87.41	118.39
5	50	0.06	0.15	0.24	0.97	0.27	0.65	0.46	1.07	0.39	0.84	0.71	1.16	4.81	5.66	6.03	6.59	9.51	12.41
	100	0.28	1.18	0.28	0.69	0.59	1.61	0.55	1.29	0.77	1.39	0.96	1.68	11.67	11.91	13.13	16.52	19.71	25.28
	150	0.50	1.22	0.69	1.55	0.60	1.39	0.77	2.01	1.19	2.17	1.34	1.92	22.37	20.99	19.68	24.51	28.92	37.87
	200	0.45	0.86	0.54	1.70	0.73	1.42	0.93	1.59	1.06	1.75	1.62	2.36	25.15	19.96	30.21	32.92	38.73	49.15
	500	0.59	1.17	0.89	1.42	1.00	1.89	1.42	2.11	1.72	2.46	1.92	3.02	75.71	90.53	95.50	106.55	105.93	135.92
6	50	0.34	0.83	0.19	0.67	0.41	0.82	0.33	0.88	0.67	1.12	0.82	1.38	7.45	7.89	7.99	9.78	12.47	17.56
	100	0.40	0.87	0.44	1.29	0.70	1.19	0.63	0.99	0.90	1.33	1.30	1.95	14.21	15.62	18.13	19.70	24.35	34.14
	150	0.56	1.21	0.80	1.22	0.80	1.38	1.02	1.97	0.86	1.77	1.61	2.42	26.03	24.89	29.23	33.92	38.78	48.56
	200	0.45	1.43	0.63	1.13	0.84	1.52	1.08	2.39	1.30	1.74	1.93	2.86	36.43	37.14	37.45	42.23	49.52	65.28
	500	0.37	1.21	1.10	2.36	1.31	2.30	1.19	2.34	1.85	2.66	1.94	2.69	104.52	99.61	107.59	115.12	131.79	166.66
7	50	0.23	0.49	0.24	0.53	0.37	0.81	0.43	0.91	0.87	1.32	0.83	1.55	9.21	8.82	10.78	12.45	14.32	18.91
	100	0.28	0.81	0.66	1.06	0.70	1.22	0.79	1.06	1.18	1.89	1.57	1.93	19.67	19.37	22.50	23.61	29.33	35.89
	150	0.39	0.97	0.57	1.03	0.89	1.75	1.00	1.58	0.99	1.82	1.71	2.67	25.19	29.33	35.30	36.49	45.32	52.59
	200	0.40	0.72	0.62	1.29	0.80	1.34	1.12	1.81	1.56	2.70	1.82	2.26	44.40	38.07	44.11	49.34	56.89	72.79
	500	0.53	1.36	1.11	1.65	0.97	2.04	1.85	2.66	2.01	3.02	2.25	3.29	225.91	115.45	132.56	136.92	174.59	195.24
8	50	0.14	0.59	0.33	0.64	0.46	0.98	0.48	0.99	0.66	1.02	0.77	1.24	9.89	9.62	12.41	13.77	16.89	21.74
	100	0.37	0.69	0.58	1.17	0.59	1.07	0.78	1.22	1.15	1.67	1.31	1.95	21.23	21.69	25.04	26.91	32.93	41.88
	150	0.31	0.54	0.74	1.19	0.83	1.68	1.26	2.00	1.42	2.35	2.11	2.48	32.50	37.11	38.96	46.63	54.82	67.58
	200	0.52	1.08	0.88	1.41	0.97	1.77	1.15	1.63	1.52	2.15	2.05	3.05	42.44	52.37	71.43	57.11	72.06	81.05
	500	0.55	1.18	0.96	1.84	1.60	2.47	1.43	1.90	1.95	2.98	2.75	3.50	154.45	140.67	159.11	168.04	176.02	232.67
10	50			0.17	0.38	0.38	0.72	0.48	0.92	0.81	1.31	0.95	1.66		18.52	16.84	21.93	21.89	29.24
	100			0.48	0.69	0.61	1.03	1.00	1.53	1.21	1.73	1.65	2.19		30.12	34.31	38.09	42.58	55.29
	150			0.54	1.02	0.87	1.36	1.02	1.38	1.58	2.17	1.78	2.49		49.03	49.52	56.71	64.09	81.59
	200			0.70	1.34	0.93	1.52	1.18	1.78	1.70	2.29	2.26	3.02		58.42	77.78	79.36	88.56	107.01
	500			1.18	1.52	1.21	1.58	1.58	2.16	2.06	2.59	2.68	3.21		196.91	230.13	251.79	311.09	347.41
12	50					0.18	0.37	0.40	0.70	0.75	1.01	0.97	1.43			24.92	21.56	28.68	36.52
	100					0.65	1.07	0.87	1.20	1.07	1.37	1.77	2.23			41.69	47.50	54.71	60.08
	150					0.90	1.32	1.17	1.71	1.42	1.89	1.80	2.25			59.72	70.29	80.79	107.17
	200					1.06	1.56	1.23	1.69	1.67	2.21	2.06	2.92			85.42	96.55	110.35	142.61
	500					1.37	1.80	1.68	2.17	1.72	2.81	2.64	3.71			242.69	318.56	379.44	448.71

For a better evaluation of our results we have used CPLEX to run some additional experiments. The information we wanted to obtain with these experiments are: (1) the values of the optimal solutions to the instances; (2) the cputime required to solve the LP relaxation of model (1)–(10); and (3) the cputime required by CPLEX to obtain the optimal solution. For these experiments a maximum cputime of 7200 s was fixed.

Tables 2 and 3 summarize the results obtained with these experiments for $|T| = 4$ and $|T| = 8$. In particular, Table 2 gives: (1) the percent gaps between the upper bounds obtained when solving the Lagrangean dual and the optimal/best solution found with CPLEX (entries labeled *gs*); (2) the number of instances out of the 10 instances of the same dimension for which the best solution found with our algorithm was at least as good as the best solution found with CPLEX (entries labeled *nb*); and (3) the number of instances out of the 10 instances of the same dimension for which

Table 2
Percent gaps of upper bounds with respect to the best solution found with CPLEX.

		$\|I\|=50$			$\|I\|=100$			$\|I\|=150$			$\|I\|=200$			$\|I\|=500$		
$\|T\|$	$\|J\|$	gs	nb	t_{max}	gs	nb	t_{max}	gs	nb	t_{max}	gs	nb	t_{max}	gs	nb	t_{max}
4	8	0.00	10	0	0.00	10	0	0.00	9	0	0.00	10	1	0.00	10	0
	10	0.00	10	0	0.00	10	0	0.00	10	0	0.02	9	1	0.00	10	4
	12	0.01	9	0	0.00	10	0	0.00	10	0	0.00	9	0	−0.01	9	5
	15	0.00	9	0	0.01	9	0	0.00	10	0	0.01	9	1	−0.02	10	6
	20	0.00	10	0	0.00	10	0	0.01	9	1	0.00	10	3	−0.01	9	6
	30	0.00	10	0	0.00	9	0	0.03	9	1	−0.06	9	6	0.02	9	5
8	8	0.00	10	0	0.00	10	0	0.00	10	0	0.02	8	0	0.00	9	5
	10	0.00	9	0	0.00	9	1	0.01	9	2	0.02	6	6	0.00	6	7
	12	0.03	8	0	0.00	10	0	0.04	6	6	−0.04	9	5	−0.14	6	10
	15	0.00	10	0	0.01	9	3	−0.05	10	8	0.00	8	7	−0.29	10	10
	20	0.00	8	0	0.00	7	7	−0.15	7	9	0.00	7	10	*	10	10
	30	0.03	7	1	−0.13	7	8	−0.00	6	10	*	10	10	*	10	10

Table 3
Comparison of cputimes between Algorithm 1, the LP solution, and the best solution found with CPLEX.

		$\|T\|=4$					$\|T\|=8$				
$\|J\|$		$\|I\|=50$	$\|I\|=100$	$\|I\|=150$	$\|I\|=200$	$\|I\|=500$	$\|I\|=50$	$\|I\|=100$	$\|I\|=150$	$\|I\|=200$	$\|I\|=500$
8	Rlag	2.69	7.87	13.69	18.21	55.78	9.89	21.23	32.50	42.44	154.45
	Rlin	0.34	0.51	1.19	3.06	21.76	0.85	2.55	6.29	10.71	105.57
	best	0.62	9.10	30.68	959.10	46.28	5.40	24.80	97.40	973.50	4664.60
10	Rlag	3.12	9.76	15.92	22.38	68.41	9.62	21.69	37.11	52.37	140.67
	Rlin	0.00	0.34	2.04	4.59	34.68	0.51	3.40	11.56	21.42	194.48
	best	1.40	9.41	17.32	840.33	3234.00	24.80	851.00	2460.00	5488.00	5760.10
12	Rlag	5.32	8.52	16.86	22.52	65.55	12.41	25.04	38.96	71.43	159.11
	Rlin	0.51	1.36	2.55	5.10	45.39	1.19	5.61	13.77	31.96	280.84
	best	2.34	4.84	25.79	135.67	3723.78	72.80	910.00	4400.50	4037.00	7215.10
15	Rlag	5.71	11.81	23.73	25.88	87.46	13.77	26.91	46.63	57.11	168.04
	Rlin	0.51	1.87	4.93	7.99	71.06	1.53	8.84	27.71	43.52	371.62
	best	11.28	53.40	447.04	962.94	4446.65	378.20	2772.10	6945.60	5347.90	7202.70
20	Rlag	6.18	12.89	21.81	29.23	87.41	16.89	32.93	54.82	72.06	176.02
	Rlin	0.68	3.06	9.35	13.60	111.69	2.55	18.36	41.82	84.15	772.82
	best	9.15	54.29	1350.60	2250.40	4599.05	746.70	5585.40	6754.78	7201.20	7205.00
30	Rlag	9.03	19.47	28.03	36.21	118.39	21.74	41.88	67.58	81.05	232.67
	Rlin	1.53	6.80	15.13	27.54	177.48	5.44	35.36	106.59	214.54	1488.10
	best	19.19	209.87	1088.89	4501.56	4253.49	1731.16	5861.34	7201.40	7201.90	7208.80

CPLEX could not prove optimality of the best solution found after the limit of 7200 s of cputime (entries labeled t_{max}). The values labeled *gs* correspond to the average over the number of instances of a given dimension for which at least one feasible solution was found by CPLEX. Entries with an asterisk mean that CPLEX could not find any feasible solution for any of the 10 instances of the corresponding dimension within the allowed cputime. These experiments have shown the difficulty of the considered problem, since CPLEX fails to prove the optimality of the best solution found in the 2 h of cputime in most of the medium size instances ($|T| \geqslant 8$ and $|I| \geqslant 150$). Moreover, already for $|T|=8$, CPLEX fails to find one single feasible solution for the instances with $|J|=20$, $|I|=500$ within the allowed 7200 s of cputime.

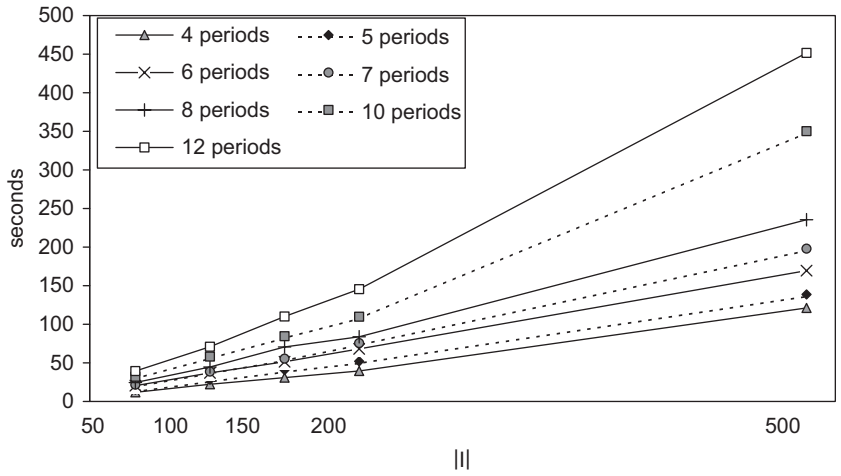

Fig. 1. Averaged CPU time of Algorithm 1 varying $|I|$ for $|J| = 30$.

The results of Table 2 show that in most cases (539 out of 600 instances) the feasible solutions obtained with our algorithm are as good as the optimal/best solution found with CPLEX. The largest average percent gap between the best solution found with our algorithm and the optimal/best solution found with CPLEX is 0.03 which, in our opinion, is extremely good. Moreover, it can be seen that for the larger instances our feasible solutions are on the average better than the best solutions found with CPLEX in 7200 s of cputime (entries labeled *gs* with a negative value).

Table 3 allows to compare the computational cost of our algorithm (entries labeled *Rlag*) with the cost required by CPLEX to obtain the lower bound by solving the LP relaxation of model (1)–(10) (entries labeled *Rlin*), and with the cost required by CPLEX to obtain the optimal/best solution found (entries labeled *best*). As was expected, for small instances CPLEX obtains the lower bound in smaller computation times. But as the sizes of the instances increase obtaining the lower bounds with our solution method becomes considerably more efficient than solving the LP relaxation with CPLEX, even if our method also computes an upper bound. The figures inn Table 3 (together with the entries labeled $t_{\max}$ in Table 2) also indicate that, broadly speaking, only instances with $|T| = 4$ could be solved optimally within the upper cputime limit. However, the computational burden of our algorithm is, excepting for the small size instances, one order of magnitude smaller than the one required by CPLEX to obtain the optimal/best solution found. Despite this, our solutions are comparable in quality and even become better than the ones of CPLEX as the sizes of the instances increase. This indicates that, even if the duality gap for these instances is very small, solving them exactly is extremely time consuming. This highlights the interest of our algorithm, since, as we have seen, we obtain feasible solutions that are either optimal or very close to optimality in small computation times.

6.2. *General planning horizon, with an arbitrary number of facilities to open per period*

These experiments were run in order to see if we could appreciate any difference on the performance of our solution method when the number of facilities to open at each period was not necessarily 1. For allowing more combinatorics in the possibilities for the sets of facilities to open at each period, in these experiments we have only considered the instances with $|J| = 30$ whereas, as before, values of $|I|$ vary in {50, 100, 150, 200, 500}, and values of $|T|$ vary in {4, 5, 6, 7, 8, 10, 12}. In fact, for these experiments we have used the corresponding instances of Section 6.1, but for each instance, the value p^t, $t \in T$ was generated as follows:

- We first obtain a number p from a discrete uniform distribution in $[|T|, |J|]$. This number is an approximation of the total number of facilities to be open $\sum_{t\in T} p^t$, and $p/|T|$ is the average number of facilities to open per period.
- For $t = 1, \ldots, |T|$, generate p^t from a discrete uniform distribution in $[a, b]$ with $a = 1$ and $b = \lceil 2p/|T| - a \rceil$.
- If $\sum_{t\in T} p^t \geqslant |J|$, we reject the instance and restart the process again.

Table 4
Gaps (left) and CPU times (right) for Algorithm 1 with general p^t for $|J| = 30$.

	Percent gap					cputime				
$\|T\|$	$\|I\| = 50$	$\|I\| = 100$	$\|I\| = 150$	$\|I\| = 200$	$\|I\| = 500$	$\|I\| = 50$	$\|I\| = 100$	$\|I\| = 150$	$\|I\| = 200$	$\|I\| = 500$
4	0.02	0.11	0.16	0.14	0.20	2.80	14.50	31.80	22.70	119.40
5	0.12	0.08	0.37	0.28	0.37	7.80	13.40	36.10	132.00	210.40
6	0.09	0.09	0.08	0.32	0.49	12.30	21.00	129.30	80.10	217.10
7	0.09	0.28	0.44	0.18	0.54	15.50	42.50	58.50	87.80	728.10
8	0.19	0.21	0.21	0.63	0.76	20.70	47.80	74.40	94.20	351.50
10	0.40	0.13	0.22	0.64	0.81	34.60	64.20	95.40	139.50	426.90
12	0.38	0.65	0.43	0.58	1.27	47.20	86.30	134.70	198.90	543.00

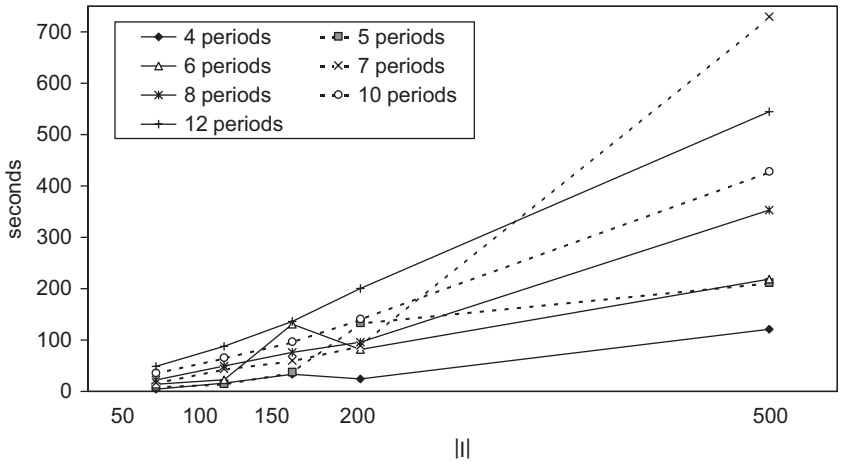

Fig. 2. Averaged CPU time of Algorithm 1 for the instances with $|J| = 30$ and $p^t > 1$.

The results are presented in Table 4. Columns have been separated in two blocks: the first one corresponds to the percent gaps between our upper and lower bounds, and the columns of the second block to the consumed cputime. Once more the values correspond to the average over the 10 instances of a given dimension. As can be seen the results are also very good. In fact, the average percent gaps never exceed 1.27% and are even smaller than the ones that we obtained in Section 6.1 for similar size instances. This is due to the fact that, since several facilities are opened per period, the objective function values are bigger in magnitude than in the case where only one facility is opened per period. Thus, even if the absolute gaps between the upper and lower bounds are similar to those of Section 6.1, they result in smaller relative gaps. As for the cputimes, we can see that they are still small for instances of the considered sizes, although they have increased slightly with respect to the ones of the previous subsection. Note that when $p^t = 1$, the Lagrangean subproblem $L_y(u)$ reduces to an assignment problem, whereas for general values of p^t, $L_y(u)$ is a transportation problem. Even if both are structured problems that can be solved efficiently, the later takes in general more computational effort than the former.

In addition, Fig. 2 shows, for instances with $|J| = 30$, the evolution of the cputime when the number of customers varies, for each possible number of periods. The reader may notice the different behavior of Algorithm 1 for $p^t = 1$ (Fig. 1) and $p^t > 1$ (Fig. 2). In the former case, cputime increases linearly with $|I|$ whereas in the latter the behavior is not that clear due to increasing combinatorics induced by the larger number of facilities to open per period.

6.3. The p-median problem

We conclude the first part of this section by reporting the numerical results of the computational experiments that we have run with the well-known p-median problem which, as we have already mentioned, is a particular case of MISFLP.

Table 5
Results of Algorithm 1 on the p-median instances of the OR-Library.

	Z_{LB}	opt	Z_{UB}	gap_L	gap_U	gap
pmed1	5798.12	5819	5819	0.36	0.00	0.36
pmed2	4067.68	4093	4093	0.62	0.00	0.62
pmed3	4234.25	4250	4254	0.37	0.09	0.47
pmed4	3033.02	3034	3034	0.03	0.00	0.03
pmed5	1352.21	1355	1355	0.21	0.00	0.21
pmed6	7754.25	7824	7842	0.89	0.23	1.13
pmed7	5619.79	5631	5650	0.20	0.34	0.54
pmed8	4438.73	4445	4445	0.14	0.00	0.14
pmed9	2727.97	2734	2734	0.22	0.00	0.22
pmed10	1252.44	1255	1258	0.20	0.24	0.44
pmed11	7675.05	7696	7709	0.27	0.17	0.44
pmed12	6623.39	6634	6634	0.16	0.00	0.16
pmed13	4368.33	4374	4374	0.13	0.00	0.13
pmed14	2962.30	2968	2971	0.19	0.10	0.29
pmed15	1727.01	1729	1738	0.12	0.52	0.64
pmed16	8062.28	8162	8177	1.22	0.18	1.42
pmed17	6955.82	6999	7095	0.62	1.37	2.00
pmed18	4806.41	4809	4811	0.05	0.04	0.10
pmed19	2843.00	2845	2850	0.07	0.18	0.25
pmed20	1786.41	1789	1796	0.14	0.39	0.54
pmed21	9120.11	9138	9138	0.20	0.00	0.20
pmed22	8523.10	8579	8656	0.65	0.90	1.56
pmed23	4618.46	4619	4625	0.01	0.13	0.14
pmed24	2955.48	2961	2971	0.19	0.34	0.53
pmed25	1826.98	1828	1837	0.06	0.49	0.55
pmed26	9823.48	9917	9933	0.94	0.16	1.11
pmed27	8292.10	8307	8321	0.18	0.17	0.35
pmed28	4497.49	4498	4507	0.01	0.20	0.21
pmed29	3029.60	3033	3054	0.11	0.69	0.81
pmed30	1985.00	1989	1997	0.20	0.40	0.60
pmed31	9998.40	10 086	10 097	0.87	0.11	0.99
pmed32	9277.12	9297	9376	0.21	0.85	1.07
pmed33	4695.97	4700	4707	0.09	0.15	0.23
pmed34	3012.46	3013	3022	0.02	0.30	0.32
pmed35	10 261.47	10 400	10 414	1.33	0.13	1.49
pmed36	9806.53	9934	9979	1.28	0.45	1.76
pmed37	5054.56	5057	5071	0.05	0.28	0.33
pmed38	10 893.28	11 060	11 087	1.51	0.24	1.78
pmed39	9350.71	9423	9455	0.77	0.34	1.12
pmed40	5126.37	5128	5138	0.03	0.20	0.23

Given that our solution algorithm does not exploit explicitly the structure of p-median problems, the aim with these experiments is to analyze the ability of our algorithm to obtain good quality solutions and good lower bounds for these problems. The test instances that we have used are the ones of the OR-Library [12]. It is a battery of 40 instances with $I = J$ and $|I|$ ranging in $\{100, 200, 300, 400, 500, 600, 700, 800, 900\}$ and $|T| = 1$. These instances are available at http://people.brunel.ac.uk/~mastjjb/jeb/info.html.

The results are presented in Table 5. The first column gives the names of the instances. Then, in the next three columns, our lower and upper bounds (z_{LB} and Z_{UB}, respectively) are compared with the optimal solution value (opt), which is known for all these instances. The following three columns give the percent gap between our lower bound and the optimal solution ($gap_L = 100 \times (opt - Z_{LB})/opt$); the percent gap between our upper bound and the optimal solution ($gap_U = 100 \times (Z_U - opt)/opt$); and the percent gap between our lower and upper bounds $gap = 100 \times (Z_{UB} - Z_{LB})/z_{LB}$. In fact, column gap_L gives the duality gap between the optimal value and the LP value for these instances. Note that the values in column gap_L are very small and similar to the ones of the instances

that we generated. Note also that, in general, the values in gap_U are very small. Nine instances were optimally solved; the average percent deviation between our best solution and the optimal value is 0.26%, and only for one instance this deviation is above 1%. We believe that these results are also very good for instances of this sizes with an algorithm that is not designed to exploit the specific characteristics of the p-median problem.

6.4. Comparison with a decoupled model

For capturing better the performance of our model we next compare the solutions that we have obtained with the ones generated with a decoupled model, where instead of solving one single problem for deciding over all the planning horizon, independent problems are solved at each period for deciding the plants and the assignments for the specific period. For the decoupled model, we also establish at each period the number of new plants to be opened (p^t), and the minimum number of customers to be serviced (n^t).

For a fair comparison between the two models, in the decoupled model, we use part of the output of a given period, as part of the input for the following period. In particular, the set of plants that are opened for the decoupled subproblem at period t are fixed to be opened at the subproblem at period $t+1$. In addition, for the reasons explained in page 2 once a customer is serviced at a period, it must also be serviced at all subsequent periods. Thus, for each customer i that is serviced at period t, in the subproblem for period $t+1$ we add the constraint, $\sum_{j\in J} x_{ij}^{t+1} = 1$, to ensure that customer i is also serviced at period $t+1$.

For this experiment we have solved the decoupled model for the instances of Section 6.2 for which CPLEX found the optimal solution of MISFLP within the time limit. The results indicate that the decoupled model behaves like a myopic greedy heuristic, for our integrated model. That is, since the decisions made in the first periods "*ignore*" the costs they will imply in later periods, the decoupled model tends to provide with very good solutions (small values) in the first periods, which force solutions that are not so good in the later periods. Since the optimal solution to the decoupled model is feasible for MISFLP, in all the cases the optimal value to the former model is higher than that of the integrated model. The percent deviation of the optimal decoupled value relative to the optimal integrated value lies in the interval [1.5, 4] (for the instances that we have solved), and the trend is that this percent gap increases with the size of the instances. We have also observed that at the earlier periods the percent deviation with respect to the optimal solution of the integrated model is negative, but it ends up by being positive at the end of the planning horizon.

We have observed that the relationship between these values is quite similar for all the instances, so we have arbitrarily chosen a representative instance with $|I| = 100$, $|J| = 30$ and $|T| = 12$ for illustration. For the integrated model, for each period we have isolated the contribution to the objective function of the variables corresponding to that period, o^t, and we have calculated the accumulated value of the optimal solution so far: $ac_{\text{int}}^t = \sum_{r\leqslant t} o^r$. Indeed $opt = ac_{\text{int}}^{|T|}$. For the decoupled model, the accumulated value for each period is just the sum of the optimal values until that period, $ac_{\text{dec}}^t = \sum_{r\leqslant t} d^r$, where d^r denotes the optimal value to the decoupled model in period r. Then, for each period t the accumulated value to the decoupled model, ac_{dec}^t, is compared to ac_{int}^t. Fig. 3 depicts the percent deviation $100 \times (ac_{\text{dec}}^t - ac_{\text{int}}^t)/ac_{\text{int}}^t$ at each time epoch.

We next compare the solutions obtained with the integrated and the decoupled models. Indeed the solutions obtained with both models must have some similarities since both require that at each period the same number of plants is opened, and the same number of customers is serviced. However, for instances with six or more periods, the number of new plants that are opened in the same period in the two models over the planning horizon is typically smaller than 50%. In addition, we can observe a percentage of about 15% of the plants that are opened at the end of the planning horizon, that were opened for the first time in different periods. For example, in the instance selected for reference, only 9 out of the 21 plants opened throughout the planning horizon were opened in the same period with the two models.

Since in both models the set of plants that are opened at each period determines the set of customers who are assigned, to a large extent the coincidence in the assignment of the customers depends on the coincidence on the plants that are opened. Once more, our test instances behave quite similarly with respect to this point. Fig. 4 further illustrates the coincidence in the assignment of customers who the selected reference instance. For each period, customers have been classified in to four groups. Group A contains all the customers who are assigned to the same plant in both models. Groups B and C contain customers who are not treated similarly by the two models.

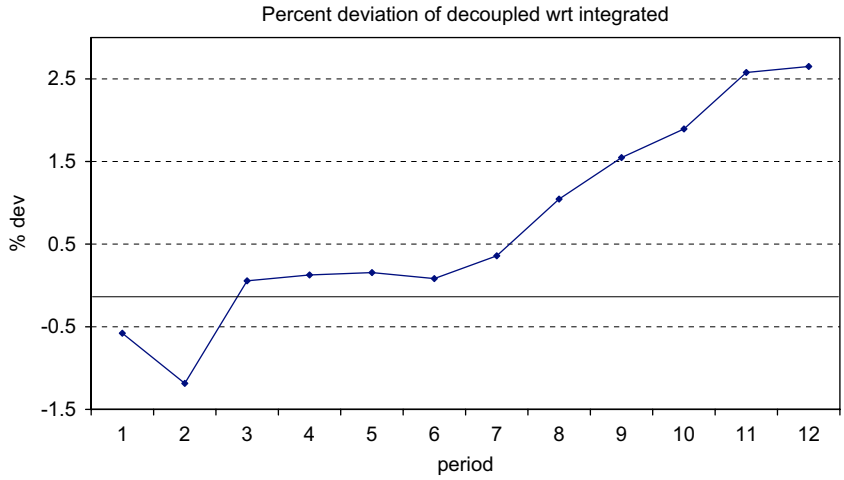

Fig. 3. Deviation of decoupled optimal value from integrated optimal value.

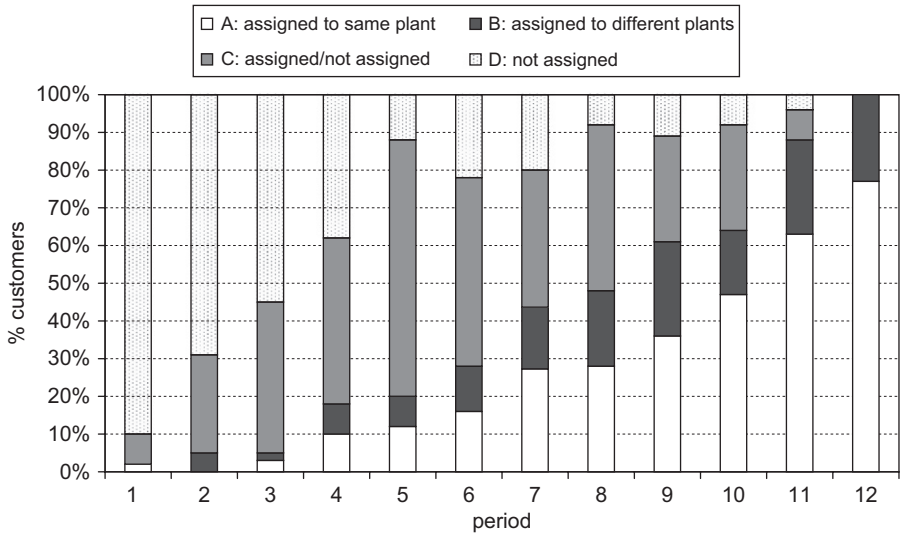

Fig. 4. Comparing assignments of customers between MISFLP and decoupled model.

In particular, group B contains the customers who are serviced with the two models, but the plant they are assigned to is not the same, whereas group C contains the customers who are serviced in one model and not serviced with the other one. Finally, group D has all the customers who are not serviced in any of the two models. As can be seen, the range of customers who are assigned to the same plant in both models ranges from 0% to 78%. The tendency is to have fewer coincidences in the medium periods and an increasing number of coincidences in the extreme periods.

6.5. Analysis on the number of periods over the planning horizon

Multi-period problems assume that the number of periods in which the planning horizon is divided is given. For enhancing the applicability of model MISFLP we next describe how it can be used as a *what-if* tool within a scenario analysis for deciding the most appropriate number of periods when decisions are made, for achieving an equilibrium between cost and quality of service. Note that, for a given data set and a fixed planning horizon, the value and the

structure of the optimal solution can vary significantly depending on the number of periods in which the planning horizon is divided. For example, if the planning horizon is one year, it is most likely that we obtain very different solutions if we make decisions every month (12 periods) or if we make decisions on months 1, 4, 8 and 12 (4 periods), even if the overall number of plants that are opened and the number of customers who are assigned throughout the planning horizon are the same. Therefore, before setting the number of periods for a multi-period problem within a given planning horizon, the decision maker should apply some trade-off analysis between cost and quality in order to select appropriate number of periods.

When the number of periods is set to $|T|$, at each period $t = 1, \ldots, |T|$ decisions must be made for selecting the new plants to be opened and the new customers to be serviced, in such a way that $\sum_{r \leqslant t} p^r$ plants are operating and n^t customers are being serviced. When the number of periods reduces from $|T|$ to K, decisions will be made at instants $t_1, t_2, \ldots, t_K$. Now it must be decided "when" to make the decisions, that otherwise would have been made at $t \in T$ different from $t_1, t_2, \ldots, t_K$, relative to the plants to be opened and the customers to be serviced at t. When such decisions are made in an anticipatory fashion, at instant t_r, $\sum_{t_r \leqslant t < t_{r+1}} p^t$ plants will be opened and $n^{t_{r+1}-1}$ customers will be serviced. On the other hand, if the decisions corresponding to instances different from $t_1, t_2, \ldots, t_K$ are postponed, then, at instant t_r, $\sum_{t_{r-1} < t \leqslant t_r} p^t$ plants will be opened and n^{t_r} customers will be serviced. Another possibility is to "distribute" the instants when the decisions are made in such a way that some are made in advance, and some other are postponed. Indeed, if decisions are made in an anticipatory fashion, the level of service (number of open plants and number of serviced customers) at instant t will be higher than when decisions are made in $|T|$ periods. However, anticipating the decisions also implies anticipating their associated costs. Thus, the overall cost over the planning horizon will be higher. On the other hand, if we postponed the decisions, then the level of service at instants t different from $t_1, \ldots, t_K$, would be lower than when decisions are made in$|T|$ periods. Hence, the overall cost over the planning horizon will be smaller. When decisions are distributed, at some instants the level of service will be higher than when $|T|$ periods are taken, whereas at some other instants the level of service would be smaller. Summarizing, the decision maker can use model MISFLP to analyze several possibilities regarding the number of periods and the strategy chosen to make the decisions, according to the relationship between the level of service and its corresponding cost in each case.

For illustrating our experiments again we use the reference instance with $|T| = 12$, $|I| = 100$ and $|J| = 30$. We have compared the following scenarios:

(A) One single anticipatory decision in period 1.
(B) Four decision instants, $t_1 = 1$, $t_2 = 4$, $t_3 = 8$ and $t_4 = 12$, with anticipatory policy.
(C) Four decision instants, $t_1 = 1$, $t_2 = 4$, $t_3 = 8$ and $t_4 = 12$, with "distributed" policy, in such a way that the number of decisions that are made in advance coincides with the number of decisions that are postponed. The number of plants opened in period $t_1 = 1$ is $p^1 + p^2$ and the number of assigned customers n^2; the number of plants opened in period $t_2 = 4$ is $p^3 + p^4 + p^5$ and the number of customers who are serviced is n^5; the number of plants opened in period $t_3 = 8$ is $p^6 + p^7 + p^8 + p^9$ and the number of customers who are serviced n^9; and the number of plants opened in period $t_4 = 12$ is $p^{10} + p^{11} + p^{12}$ and the number of customers who are serviced n^{12}.
(D) Twelve decision instants $t = 1, 2, \ldots, 12$, where at instant t, p^t new plants are opened and n^t customers are serviced.
(E) Four decision instants, $t_1 = 1$, $t_2 = 4$, $t_3 = 8$ and $t_4 = 12$, with postponed policy.
(F) One single postponed decision in period 12.

For each of the considered strategies at each instant $t = 1, \ldots, 12$ we have calculated the accumulated objective function so far, that is, the value of all the decisions made until that instant. These values are depicted in Fig. 5 for each scenario. Table 6 reports the number of customers serviced at each instant with each strategy.

As was expected, since F and A have the lowest and the highest service levels, respectively, they also have the smallest and the largest costs at all instants. Due to the way the strategies have been defined, $z^t_A > z^t_B > z^t_C > z^t_E > z^t_F, \forall t \in T$, where $z^t_\bullet$ stands for the accumulated value for strategy $\bullet$ at instant t. Notice that the same relationship holds for the respective service levels, as can be observed in Table 6. If we consider the strategy D with $|T|$ periods, we also have $z^t_B > z^t_D > z^t_E, \forall t \in T$, as it happens with the associated levels of service. The only strategies that do not compare uniformly are C and D. Both strategies coincide in level of service at times $t = 2, 5, 9, 12$. At these times, the associated values are always very close, and note that, in this example, strategy D gives always smaller costs. At the other instants,

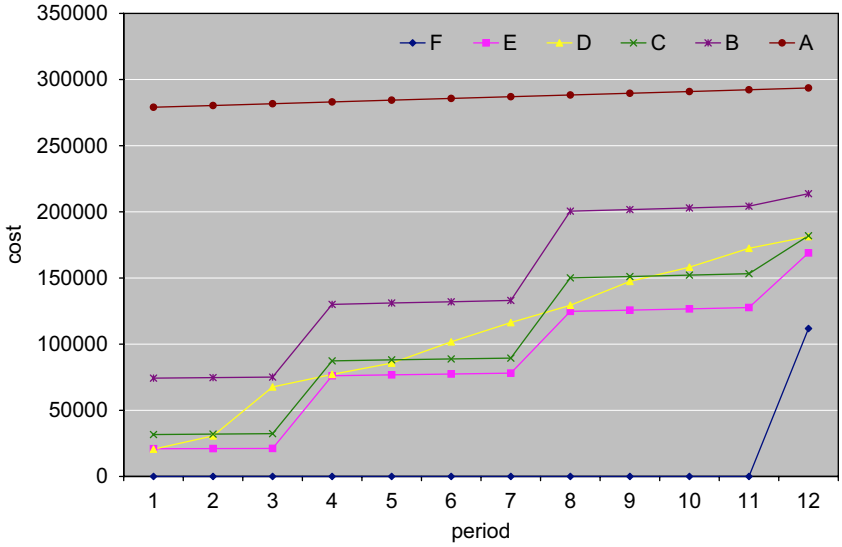

Fig. 5. Accumulated objective function value for different scenarios.

Table 6
Service levels attained by each strategy.

	1	2	3	4	5	6	7	8	9	10	11	12
F	0	0	0	0	0	0	0	0	0	0	0	100
E	6	6	6	40	40	40	40	70	70	70	70	100
D	**6**	**18**	**25**	**40**	**44**	**53**	**68**	**70**	**77**	**82**	**94**	**100**
C	18	18	18	44	44	44	44	77	77	77	77	100
B	25	25	25	68	68	68	68	94	94	94	94	100
A	100	100	100	100	100	100	100	100	100	100	100	100

it always holds that higher values correspond to better services. Finally, notice that higher number of time periods increase the level of service more gradually.

7. Conclusions

This paper introduces the multi-period incremental service facility location problem. The problem consists of finding the pattern for opening plants and assigning customers to them throughout a finite time planning horizon that guarantees the service to a pre-specified number of customers per period at the minimum total cost. The first part of the paper studies different modeling issues. We propose an MIP formulation, and study some of its structural properties. In the second part of the paper we use these properties in the design of an algorithm based on Lagrangean relaxation. As it is shown in the computational experiments, this algorithm is able to find remarkably good solutions in small computation times, despite the complexity of the considered problem. The computational experience also shows the convenience of considering all decisions concerning different periods of the planning horizon within one single optimization problem, and illustrates how to use the proposed model as a tool for choosing appropriate number of periods within a planning horizon.

The study of locational issues related to the incremental service of a set of demand points has received relatively little attention in the literature. This paper focuses on a specific multi-period location problem, but many variants can be formulated with clear practical interest. In particular, whereas the objective function considered here includes the costs of all the assignments in each period, variants considering only the largest assignment cost incurred per period are also challenging problems that deserve the attention of researchers in the area. Other interesting related problems include covering problems with sequential reduction of the coverage radius throughout a planning horizon.

Acknowledgments

This research was partially supported by Spanish Ministry of Science and Education Grants: MTM2004-22566-E, MTM2004-0909 and MTM2006-14961-C05-01. This support is gratefully acknowledged.

References

[1] Warszawski A. Multi-dimensional location problems. Operational Research Quarterly 1973;24:165–79.
[2] van Roy TJ, Erlenkotter D. A dual-based procedure for dynamic facility location. Management Science 1982;28:1091–105.
[3] Daskin MS, Hopp WJ, Medina B. Forecast horizons and dynamic facility location planning. Annals of Operations Research 1992;40:125–52.
[4] Galvão RD, Santibañez-González ER. A Lagrangean heuristic for the p_k-median dynamic location problem. European Journal of Operational Research 1992;58:250–62.
[5] Current JR, Ratick S, ReVelle CS. Dynamic facility location when the total number of facilities is uncertain: a decision analysis approach. European Journal of Operational Research 1997;110:597–609.
[6] Chardaire P, Sutter A, Costa MC. Solving the dynamic facility location problem. Networks 1996;28:117–24.
[7] Drezner Z. Dynamic facility location: the progressive *p*-median problem. Location Science 1995;3:1–7.
[8] Hinojosa Y, Puerto J, Fernández FR. A multiperiod two-echelon multicommodity capacitated plant location problem. European Journal of Operational Research 2000;123:45–65.
[9] Hinojosa Y, Kalcsics J, Nickel S, Puerto J, Velten S. Dynamic supply chain design with inventory. Computers & Operations Research 2008;35:373–91.
[10] Kariv O, Hakimi SL. An algorithmic approach to network location problems ii: the *p*-medians. SIAM Journal of Applied Mathematics 1979;37:539–60.
[11] Kennington J, Wang Z. A shortest augmenting path algorithm for the semiassignment problem. Operations Research 1992;40:178–87.
[12] Beasley JE. OR-Library: distributing test problems by electronic mail. Journal of Operational Research Society 1990;41(11):1069–72 (Available at: ⟨http://people.brunel.ac.uk/∼mastjjb/jeb/info.html⟩).