ELSEVIER

European Journal of Operational Research 124 (2000) 377–389

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem

Roberto D. Galvão [a,*], Luis Gonzalo Acosta Espejo [a], Brian Boffey [b]

[a] *Programa de Engenharia de Produção, COPPE/UFRJ, Caixa Postal 68507, Rio de Janeiro 21945-970, Brazil*
[b] *Department of Mathematical Sciences, The University of Liverpool, Liverpool L69 3BX, UK*

Received 1 November 1997; accepted 1 March 1999

Abstract

We compare heuristics based on Lagrangean and surrogate relaxations of the Maximal Covering Location Problem (MCLP). The Lagrangean relaxation of MCLP used in this paper has the integrality property and the surrogate relaxed problem we solve is the LP relaxation of the original 0–1 knapsack problem. The heuristics were compared using 331 test problems available in the literature, corresponding to networks ranging from 55 to 900 vertices. The gaps obtained with both heuristics were very low and did not differ substantially among themselves for the several problem sets used, in accordance with theoretical results reviewed in the paper. When the initial set of multipliers was determined using a valid bound for MCLP the computing times did not differ significantly between the Lagrangean and surrogate heuristics.

Keywords: Location; Heuristics; Lagrangean relaxation; Surrogate relaxation; Covering problems

1. Introduction

The Maximal Covering Location Problem (MCLP) is a well-studied problem in the category of models that provide coverage to demand areas. In these models, often used in applications related to the location of emergency facilities, a demand area is considered *covered* if it is within a predefined *service distance* from at least one of the existing facilities. MCLP was proposed by Church and ReVelle [6] and does not require that all demand areas be covered; the objective is to locate p facilities such that the maximal population is covered within the service distance.

Since its proposal MCLP has been generalized in different ways (Boffey and Narula [4]). Applications are found both in the public and private sectors. Chung [5] reviews several applications of MCLP. In relation to emergency services, Eaton et al. [13] used it to determine ambulance deployment in Santo Domingo (Dominican Republic), Current and O'Kelly [7] to locate warning sirens in cases of emergency. In the private sector MCLP has been used to locate bank branches, see Pastor [27].

* Corresponding author. Tel.: +55 21 560 88 32 ext. 415; fax: +55 21 290 6626.
E-mail address: galvao@pep.ufrj.br. (R.D. Galvão)

0377-2217/00/$ - see front matter
PII: S0377-2217(99)00171-X

Other applications of MCLP can be found in Dwyer and Evans [12] (selection of mailing lists); Daskin et al. [8] (flexible manufacturing); and Hougland and Stephens [23] (air pollution control), among others.

Early solution methods proposed for MCLP include the Linear Programming (LP) relaxation of the 0–1 integer programming formulation of the problem and "greedy"-interchange heuristics (see Church and ReVelle [6]). Galvão and ReVelle [15] developed a Lagrangean heuristic for the problem; they report computational experience using data from the literature and randomly generated networks. Exact methods include the algorithm of Dwyer and Evans [12], developed for the particular case where all demand areas have equal weights, and the dual-based algorithm of Downs and Camm [9]. The latter authors present an extensive computational evaluation of their method, in terms of both variety of applications and problem size.

In this paper we compare Lagrangean and surrogate-based heuristics for MCLP. In Section 2 we give Lagrangean and surrogate relaxations of MCLP and discuss how the integrality property of the problem relates the corresponding duals. The heuristics are described in Section 3 and computational results are given in Section 4; this is followed by the conclusions.

2. Lagrangean and surrogate relaxations of MCLP

The use of surrogate constraints in integer programming was introduced by Glover [18,19]. Theoretical treatments of surrogate duality in mathematical programming are given in Greenberg and Pierskalla [21,22] and Glover [20]. Karwan and Rardin [24] investigated relationships between bounds obtained from Lagrangean and surrogate duals and proposed a two-phase algorithm which optimizes first the Lagrangean and then the surrogate dual.

Due to the successful use of Lagrangean relaxation in solving an extensive list of combinatorial optimization problems, and to the fact that surrogate bounds are more difficult to obtain because of the "knapsack" nature of the corresponding relaxed problems, surrogate duals are lesser known and less used than Lagrangean duals. However, a very important result (see Greenberg and Pierskalla [21]) is that surrogate duality gaps are at least as small as Lagrangean duality gaps. Surrogate bounds might therefore still provide a powerful alternative for calculating bounds for tree search procedures. Within this context the conjunction of Lagrangean and surrogate duals, for example as proposed by Karwan and Rardin [24], could be of great value in some applications.

Lorena and Lopes [25] and Lorena and Narciso [26] used, respectively, surrogate relaxation in heuristics for the Set Covering and Generalized Assignment problems, solving continuous surrogate relaxations (where 0–1 knapsack problems are replaced by their continuous relaxations) in order to reduce computational times. In both cases they report their approach to be superior to Lagrangean heuristics, mainly with respect to computing times. Almiñana and Pastor [1] devised a very effective surrogate-based heuristic for the Location Set Covering Problem (a special case of the Set Covering Problem (SCP)).

In this paper we consider two relaxations for MCLP: (i) a Lagrangean Relaxation (LR), obtained by dualizing the cover constraints, as in Galvão and ReVelle [15]; (ii) a Surrogate Relaxation (SR), obtained by combining the cover contraints into a single knapsack constraint.

2.1. Mathematical formulation of MCLP

Let S be the service distance for MCLP, i.e., a demand area is covered if it is within distance S from at least one of the existing facilities. Now let $J = \{1, 2, \ldots, m\}$ be the set of demand areas, $I = \{1, 2, \ldots, n\}$ the set of potential facility sites, f_j the population of demand area j, $a_{ij} = 1$ if demand area j can be covered by a facility located at $i \in I$ within service distance S ($a_{ij} = 0$ otherwise), and p the number of facilities to be located. $x_j = 1$ if demand area j is covered ($x_j = 0$ otherwise); $y_i = 1$ means that a facility must be located at site $i \in I$ ($y_i = 0$ otherwise). The mathematical formulation of MCLP is given by:

(MCLP)

$$v(\text{MCLP}) = \max \sum_{j\in J} f_j x_j \tag{1}$$

s.t.

$$\sum_{i\in I} a_{ij} y_i - x_j \geqslant 0, \quad j \in J, \tag{2}$$

$$\sum_{i\in I} y_i = p, \tag{3}$$

$$x_j \in \{0,1\}, \quad j \in J, \tag{4}$$

$$y_i \in \{0,1\}, \quad i \in I. \tag{5}$$

In the formulation above the objective function seeks to maximize the total population covered. Constraints (2) state that a demand area $j \in J$ is covered if there is at least one facility within distance S from it. Constraint (3) limits the number of facilities in the solution to p. Finally, constraints (4) and (5) define the 0–1 nature of the decision variables.

MCLP is related to the classical SCP. The Lagrangean relaxation of MCLP, obtained by incorporating constraint (3) into the objective function, is an SCP. By varying the Lagrangean multiplier we may solve MCLP for most values of p but not necessarily for the specific value(s) we may be interested in. Furthermore, the computational effort to solve the resulting (parametric) SCP may be greater than that required for typical SCPs of the same size. For these reasons we prefer to develop algorithms specialised for MCLP.

2.2. Lagrangean relaxation $MCLP_\lambda$

Let $\lambda_j \geqslant 0$ be multipliers associated with constraints (2). The Lagrangean relaxation MCLP_λ is defined in the following way:

(MCLP_λ)

$$\begin{aligned} v(\text{MCLP}_\lambda) &= \max \left\{ \sum_j f_j x_j + \sum_j \lambda_j \left[\sum_i a_{ij} y_i - x_j \right] \right\} \\ &\equiv \max \left\{ \sum_j (f_j - \lambda_j) x_j + \sum_i \sum_j a_{ij} \lambda_j y_i \right\}. \end{aligned}$$

Or, making $\alpha_i = \alpha_i(\lambda) = \sum_j a_{ij}\lambda_j$,

$$v(\text{MCLP}_\lambda) = \max \left\{ \sum_j (f_j - \lambda_j) x_j + \sum_i \alpha_i y_i \right\} \tag{6}$$

subject to (3)–(5).

The solution of this Lagrangean problem is straightforward and its optimal value is given by

$$v(\text{MCLP}_\lambda) = \sum_j \max(0, f_j - \lambda_j) + \sum^{p} \alpha_i, \tag{7}$$

where $\sum^p \alpha_i$ is the sum of p largest α_i, ties being broken arbitrarily, with $x_j = 1$ if $\lambda_j \leqslant f_j$ ($x_j = 0$ otherwise) and $y_i = 1$ for p largest α_i ($y_i = 0$ otherwise). As noted by Galvão and ReVelle [15], a primal solution v_{primal} can be readily obtained from the optimal y^*_λ's of MCLP_λ: for each of the p $y^*_{\lambda i} = 1$ make $x_j^{\text{primal}} = 1$ if $a_{ij} = 1$, all other $x_j^{\text{primal}} = 0$. Then $v_{\text{primal}} = \sum_j f_j x_j^{\text{primal}}$ is a lower bound for MCLP.

Theorem. *There is an optimal solution of the Lagrangean dual*

(DMCLP_λ)

$$v(\text{DMCLP}_\lambda) = \min_{\lambda \geqslant 0} v(\text{MCLP}_\lambda)$$

for which $\lambda_j \leqslant f_j$ *for all* $j \in J$.

Proof. Suppose λ^* is an optimal solution of DMCLP_λ and $\lambda_k^* > f_k$ for some k. A new feasible solution $\hat{\lambda}$ is defined by: $\hat{\lambda}_k = f_k$ and $\hat{\lambda}_j = \lambda_j^*$ for all $j \neq k$. Then:

$$\begin{aligned} v(\text{MCLP}_{\lambda^*}) = &\sum_j \max(0, f_j - \lambda_j^*) + \alpha_{i(1)}(\lambda^*) + \cdots \\ &+ \alpha_{i(p)}(\lambda^*), \end{aligned} \tag{8}$$

where $\alpha_{i(1)}(\lambda^*) \geqslant \alpha_{i(2)}(\lambda^*) \geqslant \cdots \geqslant \alpha_{i(n)}(\lambda^*)$.

From the definition of $\hat{\lambda}$ and the α_i, it is readily verified that $\max(0, f_j - \hat{\lambda}_j) = \max(0, f_j - \lambda_j^*)$ and $\alpha_i(\hat{\lambda}) \leqslant \alpha_i(\lambda^*)$ for all $i \in I$. Consequently,

$$\begin{aligned} &v(\text{DMCLP}_{\lambda^*}) \\ &\geqslant \sum_j \max(0, f_j - \hat{\lambda}_j) + \alpha_{i(1)}(\hat{\lambda}) + \cdots + \alpha_{i(p)}(\hat{\lambda}) \\ &\geqslant v(\text{DMCLP}_{\hat{\lambda}}), \end{aligned} \tag{9}$$

the second inequality resulting from the observation that $\alpha_{i(1)}(\hat{\lambda}), \ldots, \alpha_{i(p)}(\hat{\lambda})$ are not necessarily the p smallest $\alpha_i(\hat{\lambda})$. Since λ^* is optimal, (9) implies $\hat{\lambda}$ is optimal also. The desired result follows by repeating the above argument to each k for which $\lambda_k^* > f_k$. □

Corollary. *$v(\mathrm{DMCLP}_\lambda) = \sum_j (f_j - \lambda_j) + \sum^p \alpha_i$ for some λ with $0 \leqslant \lambda_j \leqslant f_j$ for all $j \in J$.*

2.3. Surrogate relaxation MCLP$^\mu$

For any vector $\mu \geqslant 0,\ \mu \neq 0$ we consider the following surrogate relaxation, obtained by combining cover constraints (2) into a single knapsack constraint:

(MCLP$^\mu$)

$$v(\mathrm{MCLP}^\mu) = \max \sum_j f_j x_j \tag{10}$$

s.t.

$$\sum_j \mu_j \left\{ \sum_i a_{ij} y_i - x_j \right\} \geqslant 0 \equiv \sum_i \beta_i y_i - \sum_j \mu_j x_j \geqslant 0 \tag{11}$$

and (3)–(5), where $\beta_i = \sum_j a_{ij}\mu_j$.

Clearly those y_i corresponding to p largest β_i should be set to 1. Call $y_{\mu i}^*$ the y_i's set in this way and make $c = \sum_i \beta_i y_{\mu i}^*$; the resulting problem in x_j's is the following 0–1 knapsack problem:

$$v(\mathrm{MCLP}^\mu) = \max \sum_j f_j x_j$$

s.t.

$$\sum_j \mu_j x_j \leqslant c, \tag{12}$$

and (4).

As the 0–1 knapsack problem is NP-hard (see Garey and Johnson [16]), a strategy often used (see, for example, Lorena and Lopes [25]; Lorena and Narciso [26]) is to relax the integrality of the x_j's. The knapsack constraint (12) will now be satisfied with equality and the solution of the resulting problem, which we call RMCLP$^\lambda$, is straightforward (see Duszinski and Waluckiewicz [10,11]). We note that there will (at most) be a single fractional variable, say x_{j0}, which we will refer to as the critical variable.

Since MCLP$^\mu$ and MCLP$^{\mu'}$ are equivalent problems if $\mu' = K\mu$, K a strictly positive number, it follows that there is an optimal solution μ of the surrogate dual DMCLP$^\mu$ for which $\mu_j \leqslant f_j$ for all j (cf. theorem earlier). Similar to MCLP$_\lambda$, a primal solution v_{primal} can be readily obtained from the optimal y_μ^*'s of MCLP$^\mu$.

2.4. Lagrangean and surrogate bounds

The Lagrangean and surrogate relaxations described above provide upper bounds for MCLP for any $\lambda \geqslant 0,\ \mu \geqslant 0,\ \mu \neq 0$. The central problem here is to find vectors λ and μ that minimize these upper bounds. It is also important to determine how the bounds produced by these two relaxations relate to each other.

It is clear that the best choice for the multipliers would be an optimal solution that minimizes the corresponding duals. In addition to DMCLP$_\lambda$, defined in the theorem and repeated below for the sake of convenience, let DMCLP$^\mu$ and DRMCLP$^\mu$ be respectively the duals of MCLP$^\mu$ and RMCLP$^\mu$:

$$(\mathrm{DMCLP}_\lambda)\colon v(\mathrm{DMCLP}_\lambda) = \min_{\lambda \geqslant 0} v(\mathrm{MCLP}_\lambda).$$

$$(\mathrm{DMCLP}^\mu)\colon v(\mathrm{DMCLP}^\mu) = \min_{\mu \geqslant 0, \mu \neq 0} v(\mathrm{MCLP}^\mu).$$

$$(\mathrm{DRMCLP}^\mu)\colon v(\mathrm{DRMCLP}^\mu) = \min_{\mu \geqslant 0, \mu \neq 0} v(\mathrm{RMCLP}^\mu).$$

These duals are solved approximately through a subgradient optimization algorithm described in Section 3. We turn now to theoretical relationships that exist between LPMCLP, the LP relaxation of MCLP and the duals described above.

Let P_λ and P^μ be, respectively, Lagrangean and surrogate relaxations of a combinatorial optimization problem P (maximization), v(LP) the optimal solution of the LP relaxation of P, and D_λ, D^μ and RD^μ, respectively, the duals of the LR, SR and relaxed SR of P. Let finally DLP$_\lambda$ and DLP$^\mu$ be the

LP relaxations of D_λ and D^μ and $v(.)$ the corresponding optimal values. Geoffrion [17] proved that $v(\mathrm{LP}) = v(\mathrm{DLP}_\lambda) \geqslant v(D_\lambda)$. He also proved that if P_λ has the integrality property, then $v(\mathrm{LP}) = v(\mathrm{DLP}_\lambda) = v(D_\lambda)$.

Greenberg and Pierskalla [21] proved that any true surrogate duality gap, $v(P) - v(D^\mu)$, is at least as small and often smaller than the Lagrangean duality gap, $v(P) - v(D_\lambda)$, i.e., that $v(\mathrm{LP}) \geqslant v(D_\lambda) \geqslant v(D^\mu) \geqslant v(P)$. Karwan and Rardin [24] obtained important results that relate Lagrangean and surrogate duality. They proved that $v(\mathrm{LP}) = v(\mathrm{DLP}_\lambda) = v(\mathrm{DLP}^\mu)$ and that, if P^μ has the integrality property, then $v(\mathrm{LP}) = v(D_\lambda) = v(D^\mu)$ (Note that if P^μ has the integrality property, so does P_λ).

When these results are applied to MCLP, noting that MCLP_λ has the integrality property, it is possible to write:

$$\begin{aligned} v(\mathrm{LPMCLP}) &= v(\mathrm{DMCLP}_\lambda) \\ &= v(\mathrm{DRMCLP}^\mu) \geqslant v(\mathrm{DMCLP}^\mu) \\ &\geqslant v(\mathrm{MCLP}). \end{aligned} \tag{13}$$

Note therefore that for MCLP neither the LR bound, nor the surrogate bound obtained from the LP relaxation of the corresponding knapsack problem, will improve the bound obtained from the LP relaxation of the problem.

As noted by Galvão and ReVelle [15], the justification for the use of other than LP relaxations in this case lies in lower computing times for large problems, and on the possibility of obtaining good primal solutions as a byproduct in the course of Lagrangean/surrogate-based heuristics. Our interest in this paper lies in comparing Lagrangean and surrogate relaxations of MCLP, both with respect to quality of bounds and computing times. Specifically, we are interested in determining how the convergence of the subgradient optimization algorithms compares with the theoretical results shown above and what are the corresponding computing times.

3. A subgradient-based heuristic for MCLP

We use a subgradient optimization algorithm in the search of multipliers both to minimize $v(\mathrm{DMCLP}_\lambda)$ (Lagrangean dual) and $v(\mathrm{DRMCLP}^\mu)$ (relaxed surrogate dual). The structure of the algorithm is therefore the same for both cases, the only difference being whether we solve a Lagrangean problem or a surrogate problem for a fixed vector of multipliers (λ-Lagrangean, μ-surrogate).

In the algorithm given below we obtain upper bounds for MCLP using either Lagrangean relaxation MCLP_λ or surrogate relaxation RMCLP^μ. Lower bounds are obtained through 'greedy'-interchange and interchange heuristics. After an initial lower bound is obtained through a 'greedy'-interchange heuristic, the subgradient optimization algorithm is activated. In each iteration of the subgradient optimization algorithm the upper bound is tentatively updated. A primal solution (lower bound v_{primal}) for MCLP is also obtained using the procedure described in Section 2. This lower bound is generally a weak bound, so under certain conditions (described in (b). *Feasible solutions generated throughout the procedure* below) the corresponding primal solution is used as an initial solution in an interchange heuristic that attempts to obtain a stronger lower bound. The MCLP lower bound is then tentatively updated before a new iteration of the subgradient optimization algorithm takes place.

We are now able to describe, in pseudocode, a general subgradient-based heuristic for MCLP. This description is valid for use with either the Lagrangean or surrogate relaxation of MCLP; use with the surrogate relaxation is indicated by the **or** operator.

Procedure Subgradient_Heuristic for MCLP;
{Initialize}
{Uses the 'greedy'-interchange heuristic to find an initial solution for MCLP}
Lb: = 'Greedy'-interchange (MCLP); {lower bound for MCLP}
Ub: $= +\infty$; {upper bound for MCLP}
{Apply dual descent procedure to obtain initial set of multipliers}
inva: = 0; {counts the number of consecutive iterations without upper bound improvement}

$s[\lambda_j] := 0\ \forall j \in J$ **or** $s[\mu_j] = 0\ \forall j \in J$;

while (stop_test = **false**) **do**
  solve relaxation MCLP$_\lambda$ **or** RMCLP$^\mu$ obtaining [$(x^*_\lambda, y^*_\lambda)$ and v(MCLP$_\lambda$)] **or** [(x^*_μ, y^*_μ) and v(RMCLP$^\mu$)]; {optimal solution of relaxed problem}
  obtain a feasible solution (x^{primal}, v_{primal}). Make v_f = val_FeasSol;
  if (Ub $\leqslant$ v(MCLP$_\lambda$) **or** Ub $\leqslant$ v(RMCLP$^\mu$)) **then** inva: = inva + 1
  else
    inva : = 0;
    if conditions described in *b* below are met **then** vertex_substitution (x^{primal});
    obtain v_f = val_FeasSol;
    end_if
  end_if
  Lb: = max[Lb,v_f];
  Ub: = min[Ub, (v(MCLP$_\lambda$) **or** v(RMCLP$^\mu$))];
  compute $g[\lambda_j] := \sum_i a_{ij} y_{\lambda i} - x_{\lambda j}$ **or** $g[\mu_j] := \sum_i a_{ij} y_{\mu i} - x_{\mu j}$; {subgradient in $(x^*_\lambda, y^*_\lambda)$ **or** (x^*_μ, y^*_μ) };
  make $s[\lambda_j] := g[\lambda_j] + 0.5 * s[\lambda_j]$ **or** $s[\mu_j] := g[\mu_j] + 0.5 * s[\mu_j]$; {using a 'discounted' subgradient};
  update step_size;
  make $\lambda_j := \max\{0, \lambda_j - \text{step\_size} * s[\lambda_j]\}$ **or** $\mu_j := \max\{0, \mu_j - \text{step\_size} * s[\mu_j]\}$; {update multipliers}
  make stop tests;
end_while;
write results;
end_procedure.

Corresponding to the solution (x^*_μ, y^*_μ) of RMCLP$^\mu$ we define x^U_μ and x^L_μ to be the same as x^*_μ but with the critical variable x_{j0} set to 1 and 0, respectively. Then, as in Lorena and Lopes [25], the vector with components $\sum_i a_{ij} y_{\mu i} - x^U_{\mu j}$ is a subgradient of v(MCLP$_\lambda$) where $\lambda = (f_{j0}/\mu_{j0})\mu$. It may be verified that this remains true if x^L_μ replaces x^U_μ and hence if x^*_μ replaces x^U_μ since a convex combination of subgradients is again a subgradient (for the same λ).

The factors that influence the convergence of the heuristic are: the relaxation used (Lagrangean or surrogate); the value of the initial lower bound; the quality of feasible solutions (lower bounds) generated throughout the procedure; the initial set of multipliers; the step size; the stop rules. Among these we will now examine the issues not yet addressed in this paper.

(a) *Value of the initial lower bound.* A 'greedy'-interchange heuristic is used to obtain an initial lower bound for MCLP, such as in Galvão and ReVelle [15]. This heuristic consists of a vertex addition phase used in conjunction with a vertex substitution phase and is similar to heuristics developed for other facility location problems, for example the heuristic of Eilon and Galvão [14] for the *p*-median problem.

(b) *Feasible solutions generated throughout the procedure.* A 'stand-alone' interchange heuristic is used to improve the primal solution v_{primal} under certain conditions. In preliminary runs of the Subgradient_Heuristic described above two facts were observed: (i) upper bounds provided by the heuristic are highly variable in the initial iterations (tending to stabilize as the number of iterations increases); (ii) the interchange heuristic is very time consuming. Given these two observations, we decided to activate the interchange heuristic depending on the percentage decrease of the upper bound as a function of the number of iterations, in the following way.

In the first 100 iterations, starting from the 0.750% level, the threshold improvement in the upper bound to activate the interchange heuristic is multiplied by 0.50 every 20 iterations. Accordingly, in the first 20 iterations the interchange heuristic is activated whenever the upper bound decreases by at least 0.750%; in iterations 21–40 it is activated whenever the upper bound decreases by at least 0.375%; and so on, up to iteration 100. In iterations 101–200, the threshold value is multiplied by 0.75 every 30 iterations; in iterations 201–400 this value is multiplied by 0.90 every 50 iterations. This threshold value, however, cannot be inferior to 0.050%.

(c) *Initial set of multipliers.* Motivated by the work of Beasley [3] on the Set Covering Problem a dual descent procedure was used to choose the initial set of multipliers.

Looking at the form of v(MCLP$_\lambda$) in Eq. (6) it is clearly desirable that for all *j*,*k*:

(i) λ_j be as large as possible (subject to $\lambda_j \leqslant f_j$);

(ii) α_k be as small as possible.

These are conflicting aims. However, since an increase in λ_j leads to an increase in α_k for all

$k \in N_j = \{i : a_{ij} = 1\}$, it seems sensible to put more emphasis on increasing those λ_j for which $n_j = |N_j|$ is relatively small. The routine SetMultipliers(M) that follows makes use of this observation.

SetMultipliers(M)

{Set λ subject to λ_j, $\alpha_i \leqslant M$ for all i,j}

For all j taken in increasing order of the n_j set

$$e_j = \min_{i \in Nj}(M - \alpha_i), \lambda_j = \min(M, f_j, e_j).$$

If $W(M)$ denotes $v(\mathrm{MCLP}_\lambda)$ when λ is determined by SetMultipliers(M), we may expect some values of M to lead to better bounds $W(M)$ than others. For a given p, we use a Golden Section Search (see for example Walsh [29]) to find an M which approximately minimises $W(M)$. The initial 'range of uncertainty' is taken to be $0 \leqslant M \leqslant 3n\sum_j f_j/2pm$, where $n = |I|$ and $m = |J|$. The search terminates when the range has been reduced by a factor of approximately 15.

It may be thought that use of Golden Section Search is invalid since the function W is only 'approximately convex'. This is not so as we are using the procedure to calculate $W(M)$ for several suitable values of M with the best set of multipliers encountered being retained.

(d) *Step size.* We calculate the step size at iteration k applying the commonly used formula $\theta_k = \alpha_k(Ub_k - Lb_k)/\|s\|^2$, where α_k, the value of the step size parameter at iteration k, is a scalar satisfying $0 < \alpha_k \leqslant 2$, Ub_k and Lb_k are, respectively, the upper and lower bounds at iteration k, and $\|s\|^2$ is the norm of the discounted subgradient at iteration k. In view of the theorem, if ($s_j < 0$ and $u_j = f_j$) or ($s_j > 0$ and $u_j = 0$) then s_j is reset to zero before the update step. If after the update $u_j > f_j$ then, again in view of the theorem, we set $u_j = f_j$. We start with the value of the step size parameter at $\alpha_0 = 2$ and halve it whenever there is no improvement in the upper bound after 15 iterations.

(e) *Stop rules.* In all cases the solution for MCLP is given by the primal solution corresponding to the current value of the best lower bound found. The stop rules for the algorithm are the following:

(i) $(Ub - Lb) < 1$. As the data for all test problems are integers, in this case the optimal solution of MCLP has been reached.
(ii) Total coverage has been achieved by the primal solution. In this case the optimal solution of MCLP has also been reached.
(iii) $g(\lambda_j)[g(\mu_j)] = 0\ \forall j \in J$. In this case the optimal solution of the dual (Lagrangean or surrogate) has been reached. There may or may not be a duality gap in relation to the optimal solution of MCLP.
(iv) $\theta_k < 0.01$. In this case there exists a gap between the best upper and lower bounds obtained through the procedure and optimality is not proven.
(v) Num_iter (number of iterations) > 400. In this case a gap also exists.

(f) *Problem reduction.* There are various problem reductions for the SCP. However, for MCLP the restriction to p facilities implicitly implies equal cost facilities and there is only one relevant reduction. This is that if a facility at r can cover the same demand as a facility at s (and possibly more) then there is an optimal solution in which no facility is placed at s. Precisely, we may set $y_s = 0$ if $a_{ir} \geq a_{is}$ for all i. We performed such an initial problem reduction not that it improves convergence but because it permits the interchange heuristic to be implemented more efficiently.

As the algorithm progresses we may be able to fix more y_i to be zero. This is determined by considering the effect of adding the constraint $y_i = 1$ to the appropriate relaxation. Then, if this yields a value no greater than the best primal solution so far found, we may infer that there is an optimal solution for which $y_i = 0$. We checked for such variable fixing only at iterations leading to dual bound improvement, but the number of situations in which it was successful did not justify the computational effort involved. Of course, a similar possibility for setting y_i to be 1 exists, but again the computational effort involved did not justify its implementation.

4. Computational results

In order to compare the Lagrangean and surrogate relaxations of MCLP we used test problems

available in the literature. These are the 55-vertex problem of Swain [28], the 100 and 150-vertex problems of Galvão and ReVelle [15] and the following problems, whose distance matrices were taken from Beasley's OR Library [2] *Pmed* problems: (i) five sets each of 100, 300, 500 and 600-vertex networks (Beasley's problems Pmed01 to Pmed05; Pmed11 to Pmed15 and Pmed21 to Pmed30), (ii) four sets of 700-vertex networks (Beasley's problems Pmed31 to Pmed34); (iii) three sets each of 800 and 900-vertex networks (Beasley's problems Pmed35 to Pmed40). The Galvão and Revelle problems were randomly generated; a description of how these networks were generated can be found in the corresponding paper.

The values of p and S used in each test problem are not meaningful when taken separately, but their combined values determine the percentage of the total population covered in each case. We chose values of p and S that generated a range of total population covered varying from 59% to 100% for the 331 test problems used in this paper.

Beasley's OR Library *Pmed* problems are problems generated for the p-median problem and as such his data do not include the population of each vertex (demand area). We randomly generated these populations, from a uniform distribution defined in the range [20,30] for the 100-vertex problems, and from a normal distribution with mean 80 and standard deviation 15 for the problems with 300 or more vertices. A summary of the test problems used is given in Table 1.

The two heuristics (Lagrangean and surrogate) were implemented under identical hardware and software configurations. The heuristics were coded in *Digital's Pascal* and run on a Digital Alpha 3000/300 workstation. The solutions obtained from the heuristics were compared with optimal solutions obtained via CPLEX, Version 4.0.7, run on the same workstation, for problems with up to 600 vertices and for two sets of the 700-vertex problems, CPU times becoming prohibitive for larger problems. For these larger problems (Table 3) the gaps shown are % gaps between the upper and lower bounds provided by the heuristics.

A summary of the computational results is given in Tables 2 and 3. In Table 2 we show a summary of the results for problems for which it was possible to obtain the optimal solution through CPLEX: we give average CPLEX times, average and maximal heuristic times and average

Table 1
Summary of test problems[a]

Problem set	Number of vertices	Values of p	Values of s	Source	No. of problems
S55	55	[03,04,05,06,07,08]	[06,08,10,12]	Swain [28]	24
G&R100	100	[08,10,12]	[50,65,80]	Galvão and Revelle [15]	9
B100_01	100	[08,10,12]	[55,65,75]	(Pmed01)	9
B100_02 to B100_05	100	[08,10,12]	[55,70,85]	(Pmed02–Pmed05)	36
G&R150	150	[05,06,07,08,10,12,14, 16,18,20]	[70,75,80,85, 90,95]	Galvão and Revelle [15]	28
B300_11 to B300_15	300	[18,20,22]	[25,30,35]	(Pmed11–Pmed15)	45
B500_21	500	[18,23,28]	[15,20,25]	(Pmed21)	9
B500_22	500	[15,18,21]	[20,25,30]	(Pmed22)	9
B500_23 & B500_24	500	[24,27,30]	[15,20,25]	(Pmed23 and Pmed24)	18
B500_25	500	[24,28,32]	[15,20,25]	(Pmed25)	9
B600_26	600	[15,20,25]	[15,20,25]	(Pmed26)	9
B600_27, B600_29 &B600_30	600	[17,20,23]	[15,20,25]	(Pmed27, Pmed29 and Pmed30)	27
B600_28	600	[16,19,22]	[15,20,25]	(Pmed28)	9
B700_31 to B700_34	700	[20,24,28]	[13,15,20]	(Pmed31–Pmed34)	36
B800_35 to B800_37	800	[18,21,24]	[13,15,20]	(Pmed35–Pmed37)	27
B900_38 to B900_40	900	[20,24,28]	[10,13,16]	(Pmed38–Pmed40)	27

[a] All problems of the form 'Pmed**' are from Beasley's OR Library [2].

Table 2
Summary of computational results, optimal solutions included

Problem set	CPLEX	Lagrangean heuristic			Surrogate heuristic		
	Average time[a]	Average time[a]	Maximal time[a]	% gap[b]	Average time[a]	Maximal time[a]	% gap[b]
S55	0.08	0.17	2.00	0.24	0.13	1.00	0.16
G&R100	62.34	2.44	4.00	0.65	1.89	3.00	0.47
B100_01	0.15	0.78	1.00	0.06	1.00	2.00	0.06
B100_02	1.31	1.00	2.00	0.14	0.89	2.00	0.33
B100_03	0.18	0.67	2.00	0.28	1.22	2.00	0.19
B100_04	0.17	0.78	2.00	0.00	1.00	2.00	0.00
B100_05	0.18	1.00	2.00	0.32	0.78	1.00	0.49
G&R150	692.76	8.57	15.00	0.54	8.86	14.00	0.41
B300_11	3.90	81.67	133.00	0.47	83.56	142.00	0.40
B300_12	3.38	53.22	90.00	0.44	55.67	84.00	0.34
B300_13	5.02	65.67	112.00	0.43	65.56	115.00	0.24
B300_14	58.36	63.89	104.00	0.33	63.56	105.00	0.30
B300_15	3.20	71.56	125.00	0.35	71.33	127.00	0.32
B500_21	28.62	346.44	784.00	0.32	357.56	748.00	0.33
B500_22	869.38	416.78	785.00	0.38	400.89	668.00	0.38
B500_23	49.58	365.44	696.00	0.51	369.56	686.00	0.55
B500_24	30.57	429.67	758.00	0.36	375.89	664.00	0.45
B500_25	47.07	464.67	962.00	0.27	456.89	915.00	0.28
B600_26	302.36	794.56	1769.00	0.30	791.56	1847.00	0.27
B600_27	527.26	839.11	1632.00	0.42	806.67	1482.00	0.63
B600_28	105.52	799.56	1644.00	0.15	759.44	1360.00	0.17
B600_29	543.96	777.67	1541.00	0.22	756.11	1301.00	0.24
B600_30	697.16	589.56	1176.00	0.33	579.78	1166.00	0.34
B700_31	1206.32	1119.00	2391.00	0.44	1142.56	2222.00	0.21
B700_32	18028.92	950.89	2158.00	0.26	942.44	2076.00	0.29
Average				0.33			0.31

[a] Cpu seconds on a Digital 3000/300 workstation, exclusive of I/O times.
[b] % gap calculated in relation to the optimal solution provided by CPLEX: % gap = (opt. sol. − heur. sol.)/(opt. sol.) × 100.

Table 3
Summary of computational results, optimal solutions not included

Problem set	Lagrangean heuristic			Surrogate heuristic		
	Average time[a]	Maximal time[a]	% gap[b]	Average time[a]	Maximal time[a]	% gap[b]
B700_33	913.56	1836.00	1.32	970.67	1820.00	1.29
B700_34	1049.00	2064.00	0.85	1046.78	2094.00	0.87
B800_35	2272.00	5275.00	0.79	2232.89	4508.00	0.77
B800_36	1448.00	3012.00	1.23	1466.56	3048.00	1.15
B800_37	1367.44	2698.00	1.73	1322.56	2815.00	1.80
B900_38	2256.33	4165.00	0.79	2038.67	4263.00	0.86
B900_39	2049.89	4292.00	0.71	2032.00	4244.00	0.73
B900_40	1454.33	2877.00	0.91	1398.33	2781.00	0.92
Average			1.04			1.05

[a] Cpu seconds on a Digital 3000/300 workstation, exclusive of I/O times.
[b] % gap calculated between upper and lower bounds provided by the heuristics: % gap = (upper bound − lower bound)/(upper bound) × 100.

% gaps in relation to the optimal solution, both for the Lagrangean and surrogate heuristics, for each group of problems. In Table 3 we show a summary of the results for the problems where CPLEX times became prohibitive: we give average and maximal heuristic times and average % gaps between upper and lower bounds.

As can be seen in Tables 2 and 3, the gaps produced by both heuristics are very close to each other in all cases. The computing times also do not differ significantly between the two heuristics. These computing times do not confirm the conclusions of Lorena and Lopes [25] and Lorena and Narciso [26], who applied Lagrangean and surrogate relaxations to the Set Covering and Generalized Assignment problems, respectively. In both cases they report significantly lower computing times for their surrogate heuristics (as compared with Lagrangean heuristic times), especially for large problems.

In an earlier version of this paper we obtained surrogate heuristic times that were approximately half of the corresponding Lagrangean times for networks with more than 300 vertices (a sample of these preliminary results is shown in Table 4). On that occasion, however, we did not use the theorem of Section 2 to obtain the initial set of multipliers. When we used the theorem and applied a dual descent procedure to determine the initial multipliers, Lagrangean times were halved but surrogate times remained practically unchanged. An explanation for this is that with a carefully chosen set of initial multipliers both heuristics quickly achieve stability and the surrogate heuristic loses its "advantage" of greater stability reported by Lorena and Lopes (see [25], Figs. 1 and 2).

Tables 5–7 show complete results for selected problem sets. These tables are self-explanatory. The (% Cover) columns indicate the percentage of the total population covered in the solutions found by the heuristic procedures.

A word should be said about CPLEX times. These times are generally low for the Beasley problems of up to 500 vertices (with the exception of the B500_22 problem set), but high for the randomly generated networks G&R100 and G&R150. CPLEX times became prohibitive for the Beasley problems with more than 600 vertices. It is also interesting to note that these times are highly variable among individual problems, as can be easily seen in Tables 5 and 6.

We formulated MCLP as efficiently as we could for the CPLEX runs (for example, only the y_i variables were defined as binary: the x_j variables were defined in the range $0 \leqslant x_j \leqslant 1\ \forall j \in J$, as the problem formulation implies that these variables will be binary in the solution anyway; this saved considerable computational effort when CPLEX was executed). However, it was found that CPU times depend more on the data of specific problems than on problem size. This fact justifies the

Table 4
Sample of preliminary results: Dual descent procedure not used for obtaining initial set of multipliers

Problem set	CPLEX	Lagrangean heuristic			Surrogate heuristic		
	Average time[a]	Average time[a]	Maximal time[a]	% gap[b]	Average time[a]	Maximal time[a]	% gap[b]
G&R100	62.34	3.44	6.00	0.23	2.67	5.00	0.63
B100_01	0.15	2.33	3.00	0.01	2.33	3.00	0.07
G&R150	692.76	12.54	20.00	0.23	10.07	18.00	0.35
B300_11	3.90	245.00	430.00	0.09	97.00	125.00	0.30
B300_13	5.02	248.78	423.00	0.11	99.67	129.00	0.14
B500_21	28.62	949.33	1838.00	0.16	434.11	703.00	0.29
B500_25	47.07	1192.00	2208.00	0.16	550.11	862.00	0.19
B700_32	18028.92	3048.11	5077.00	0.14	1479.67	1968.00	0.17
B700_33	–	2532.11	3734.00	–	1399.67	1888.00	–
B700_34	–	2910.78	3969.00	–	1378.78	2008.00	–

[a] Cpu seconds on a Digital 3000/300 workstation, exclusive of I/O times.
[b] % gap calculated in relation to the optimal solution provided by CPLEX: % gap = (opt. sol. − heur. sol.) / (opt. sol.) × 100.

Table 5
Selected results for problem set G&R150

Problem			CPLEX		Lagrangean heuristic					Surrogate heuristic				
n	*p*	*s*	Optimal solution	Total time[a]	No. iter.	Heuristic solution	% gap	% Cover.	Total time[a]	No. iter.	Heuristic solution	% gap	% Cover.	Total time[a]
150	10	70	8272	15.73	376	8272	0.00	68.86	9.00	400	8272	0.00	68.86	11.00
150	12	70	9291	50.23	339	9198	1.00	76.57	8.00	400	9261	0.32	77.09	11.00
150	14	70	10022	277.32	306	10012	0.10	83.34	12.00	369	10012	0.10	83.34	13.00
150	16	70	10587	2253.72	355	10490	0.92	87.32	12.00	340	10541	0.43	87.75	13.00
150	18	70	11153	6373.78	353	11099	0.48	92.39	12.00	376	10961	1.72	91.24	8.00
150	20	70	11610	5830.93	271	11286	2.79	93.95	6.00	168	11286	2.79	93.95	6.00
150	8	80	7387	4.45	335	7387	0.00	61.49	4.00	382	7387	0.00	61.49	7.00
150	10	80	8518	13.98	324	8518	0.00	70.91	6.00	336	8518	0.00	70.91	8.00
150	12	80	9394	100.63	392	9387	0.07	78.14	10.00	400	9387	0.07	78.14	12.00
150	14	80	10147	427.77	396	10097	0.49	84.05	11.00	345	10147	0.00	84.47	12.00
150	6	90	9907	3.05	323	9907	0.00	82.47	4.00	400	9907	0.00	82.47	5.00
150	8	90	10787	6.58	347	10787	0.00	89.79	8.00	397	10787	0.00	89.79	9.00
150	10	90	11311	30.17	345	11241	0.62	93.57	10.00	262	11297	0.12	94.04	7.00
150	12	90	11759	53.42	354	11616	1.22	96.70	9.00	153	11644	0.98	96.93	5.00
150	14	90	12013	17.92	279	11896	0.97	99.03	8.00	122	11896	0.97	99.03	5.00
150	5	95	10479	8.02	313	10479	0.00	87.23	4.00	382	10479	0.00	87.23	7.00
150	7	95	11285	20.88	348	11208	0.68	93.30	6.00	365	11285	0.00	93.94	8.00

[a] Cpu seconds on a Digital 3000/300 workstation, exclusive of I/O times.

Table 6
Complete results for problems set B700_32

Problem			CPLEX		Lagrangean heuristic					Surrogate heuristic				
n	*p*	*s*	Optimal solution	Total time[a]	No. iter.	Heuristic solution	% gap	% Cover.	Total time[a]	No. iter.	Heuristic solution	% gap	% Cover.	Total time[a]
700	20	13	38993	258.47	331	38946	0.12	70.03	334.00	379	38907	0.22	69.96	329.00
700	24	13	41402	479.60	323	41402	0.00	74.44	399.00	366	41402	0.00	74.44	615.00
700	28	13	43408	488.75	349	43379	0.07	78.00	448.00	345	43408	0.00	78.05	536.00
700	20	15	44310	9146.02	345	44247	0.14	79.56	592.00	359	44138	0.39	79.36	424.00
700	24	15	46297	45649.63	384	46256	0.09	83.17	764.00	380	46256	0.09	83.17	662.00
700	28	15	48006	98769.20	354	47932	0.15	86.18	890.00	375	47932	0.15	86.18	841.00
700	20	20	53608	1607.87	360	53261	0.65	95.76	1332.00	316	53261	0.65	95.76	1281.00
700	24	20	54270	4750.65	339	53956	0.58	97.01	1641.00	400	53956	0.58	97.01	1718.00
700	28	28	54825	1110.08	400	54514	0.57	98.02	2158.00	350	54514	0.57	98.02	2076.00

development of specialized algorithms for MCLP. Besides becoming prohibitive for the larger problems, low average CPLEX times for some sets of the smaller problems do not mean that these problems may in general be expected to run in predictable computing times when CPLEX is used. In this respect the Lagrangean and surrogate times are more well-behaved, increasing moderately with problem size and not showing high variability among problems of the same set.

5. Conclusions

In this paper we compare heuristics based on Lagrangean and surrogate relaxations of the Maximal Covering Location Problem. The surrogate relaxed problem reduces to a 0−1 knapsack problem; as 0−1 knapsack problems are NP-hard, we solve instead the corresponding relaxed knapsack problem. The Lagrangean relaxation of MCLP used in this paper has the integrality

Table 7
Complete results for problem set B900_39

Problem			Lagrangean heuristic					Surrogate heuristic				
n	*p*	*s*	No. iter.	Heuristic solution	% gap	% Cover.	Total time[a]	No. iter.	Heuristic solution	% gap	% Cover.	Total time[a]
900	20	10	400	48502	0.96	67.72	763.00	369	48123	1.76	67.19	671.00
900	24	10	337	51047	1.49	71.28	916.00	400	51265	1.07	71.58	1099.00
900	28	10	347	53720	0.98	75.01	1096.00	400	53800	0.83	75.12	1272.00
900	20	13	400	63048	0.38	88.03	1433.00	400	63048	0.39	88.03	1272.00
900	24	13	400	64799	0.36	90.48	1656.00	400	64799	0.36	90.48	1698.00
900	28	13	325	66102	0.42	92.30	1989.00	400	66102	0.42	92.30	2036.00
900	20	16	287	69277	0.53	96.73	2725.00	299	69277	0.53	96.73	2727.00
900	24	16	355	69944	0.61	97.66	3579.00	238	69944	0.62	97.66	3269.00
900	28	16	321	70493	0.62	98.43	4292.00	318	70493	0.62	98.43	4244.00

[a] Cpu seconds on a Digital 3000/300 workstation, exclusive of I/O times.

property; therefore, in this case, neither the LR bound nor the surrogate bound obtained from the LP relaxation of the corresponding knapsack problem, will improve the bound obtained from the LP relaxation of the problem.

The two heuristics were compared using several test problems available in the literature. For several of these problems the corresponding optimal solutions were obtained through CPLEX. In line with the theoretical results mentioned above, the % gaps calculated for the two heuristics did not differ substantially for any group of problems used in this paper; in Table 2 for example the average gap was 0.33% for the Lagrangean heuristic and 0.31% for the surrogate heuristic. In this sense both heuristics can be considered very efficient in finding good quality solutions for MCLP.

The computing times we report do not confirm the findings of Lorena and Lopes [25] and Lorena and Narciso [26] for the Set Covering and Generalized Assignment problems: when we used the theorem and a dual descent algorithm to determine the initial set of multipliers the computing times did not differ significantly between the Lagrangean and surrogate heuristics.

Due to an efficient formulation of MCLP as a mixed integer programming problem for the CPLEX runs, for networks of up to 600 vertices whose distances were calculated according to the Euclidean metric (the Beasley problems) average CPLEX times were generally lower than times obtained using the two heuristics (this was not true for the randomly generated networks G&R100 and G&R150). Nevertheless, even for the more favourable data sets, the high variability of CPLEX times among individual problems does not recommend its use as a standard technique for solving MCLP for large data sets. In this respect these computing times are not predictable, depending more on the data of specific problems than on problem size.

References

[1] M. Almiñana, J.T. Pastor, An adaptation of SH heuristic to the location set covering problem, European Journal of Operational Research 100 (1997) 586–593.
[2] J.E. Beasley, OR-Library: Distributing test problems by electronic mail, Journal of the Operational Research Society 41 (1990) 1069–1072.
[3] J.E. Beasley, An algorithm for the set covering problem, European Journal of Operational Research 31 (1987) 85–93.
[4] T.B. Boffey, C.S. Narula, Multiobjective covering and routing problems, in: M. Karwan, J. Spronk and J. Wallenius (Eds.), Essays in Decision Making: A Volume in Honor of Stanley Zionts, Springer, Berlin, 1997, pp. 342–370.
[5] C.H. Chung, Recent applications of the Maximal Covering Location Problem (MCLP) model, Journal of the Operational Research Society 37 (1986) 735–746.
[6] R.L. Church, C.S. ReVelle, The maximal covering location problem, Papers of the Regional Science Association 32 (1974) 101–118.
[7] J. Current, M. O'Kelly, Locating emergency warning sirens, Decision Sciences 23 (1992) 221–234.
[8] M.S. Daskin, P.C. Jones, T.J. Lowe, Rationalizing tool selection in a flexible manufacturing system for sheet metal products, Operations Research 38 (1990) 1104–1115.

[9] B.T. Downs, J.D. Camm, An exact algorithm for the maximal covering location problem, Naval Research Logistics Quarterly 43 (1996) 435–461.
[10] K. Duszinski, S. Waluckiewicz, A fast algorithm for the linear multiple choice knapsack problem, Operational Research Letters 3 (1984) 205–209.
[11] K. Duszinski, S. Waluckiewicz, Exact methods for the knapsack problem and its generalizations, European Journal of Operational Research 28 (1987) 3–21.
[12] F.R. Dwyer, J.R. Evans, A branch and bound algorithm for the list selection problem in direct mail advertising, Management Science 27 (1981) 658–667.
[13] D. Eaton, M. Hector, V. Sanchez, R. Latingua, J. Morgan, Determining ambulance deployment in Santo Domingo, Dominican Republic, Journal of the Operational Research Society 37 (1986) 113–126.
[14] S. Eilon, R.D. Galvão, Single and double vertex substitution in heuristic procedures for the p-median problem, Management Science 24 (1978) 1763–1766.
[15] R.D. Galvão, C.S. ReVelle, A Lagrangean heuristic for the maximal covering location problem, European Journal of Operational Research 88 (1996) 114–123.
[16] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, W.H. Freeman and Co., San Francisco, 1979.
[17] A.M. Geoffrion, Lagrangean relaxation for integer programming, Mathematical Programming Study 2 (1974) 82–114.
[18] F. Glover, A multiphase-dual algorithm for the zero–one integer programming problem, Operations Research 13 (1965) 879–919.
[19] F. Glover, Surrogate constraints, Operations Research 16 (1968) 741–749.
[20] F. Glover, Surrogate constraint duality in mathematical programming, Operations Research 23 (1975) 434–451.
[21] H.J. Greenberg, W.P. Pierskalla, Surrogate mathematical programming, Operations Research 18 (1970) 924–939.
[22] H.J. Greenberg, W.P. Pierskalla, Quasi-conjugate functions and surrogate duality, Cahiers du Centre d'Études de Recherche Opérationnelle 15 (1973) 437–448.
[23] E.S. Hougland, N.T. Stephens, Air pollutant monitor siting by analytical techniques, Journal of the Air Polution Control Association 26 (1976) 52–53.
[24] M.H. Karwan, R.L. Rardin, Some relationships between Lagrangian and surrogate duality in integer programming, Mathematical Programming 17 (1979) 320–334.
[25] L.A.N. Lorena, F.B. Lopes, A surrogate heuristic for set covering problems, European Journal of Operational Research 79 (1994) 138–150.
[26] L.A.N. Lorena, M.G. Narciso, Relaxation heuristics for a generalized assignment problem, European Journal of Operational Research 91 (1996) 600–610.
[27] J.T. Pastor, Bicriterion programs and managerial locations: Application to the banking sector, Journal of the Operational Research Society 45 (1994) 1351–1362.
[28] R. Swain, A decomposition algorithm for a class of facility location problems, Ph.D. Thesis, Cornell University, Ithaca, NY, 1971.
[29] G.R. Walsh, Methods of Optimization, Wiley, London, 1975.