Hindawi Publishing Corporation
Advances in Operations Research
Volume 2010, Article ID 120756, 12 pages
doi:10.1155/2010/120756

Research Article

A Decomposition Heuristic for the Maximal Covering Location Problem

Edson Luiz França Senne,[1] Marcos Antonio Pereira,[1] and Luiz Antonio Nogueira Lorena[2]

[1] *Department of Mathematics, Engineering College (FEG), São Paulo State University (UNESP), 12516-410 Guaratinguetá, SP, Brazil*
[2] *Associate Laboratory of Applied Mathematics and Computation (LAC), Brazilian Institute for Space Research (INPE), 12201-970 São José dos Campos, SP, Brazil*

Correspondence should be addressed to Marcos Antonio Pereira, mapereira@feg.unesp.br

Received 29 April 2009; Revised 5 January 2010; Accepted 12 February 2010

Academic Editor: George Steiner

This paper proposes a cluster partitioning technique to calculate improved upper bounds to the optimal solution of maximal covering location problems. Given a covering distance, a graph is built considering as vertices the potential facility locations, and with an edge connecting each pair of facilities that attend a same client. Coupling constraints, corresponding to some edges of this graph, are identified and relaxed in the Lagrangean way, resulting in disconnected subgraphs representing smaller subproblems that are computationally easier to solve by exact methods. The proposed technique is compared to the classical approach, using real data and instances from the available literature.

1. Introduction

The covering class of facility location problems deals with the maximum distance between any client and the facility designed to attend an associated demand. These problems are known as covering problems and the maximum service distance is known as covering distance. The Set Covering Problem [1] determines the minimal number of facilities that are necessary to attend all clients, for a given covering distance. Due to formulation restrictions, this model does not consider the individual demand of each client. In addition, the number of needed facilities can be large, incurring high fixed installation costs. An alternative formulation considers the installation of a limited number of facilities, even if this amount is unable to attend the total demand. In this formulation, the condition that all clients must be served is relaxed and the objective is changed to locate p facilities such that the most part of the existing demand can be attended, for a given covering distance. This model corresponds to the Maximal Covering Location Problem (MCLP).

Covering models are often found in problems of public organizations for the location of emergency services. Early techniques for solving the MCLP tried to obtain integer solutions from the linear relaxation equivalent of the model proposed by Church and ReVelle [2]. This pioneer work formalizes the MCLP and presents a greedy heuristic based on vertices exchange.

MCLP applications range from emergency services [3, 4], hierarchical health services [5], air pollution control [6], to congested systems [7–9]. Solution methods for the MCLP include the linear programming relaxation [2], greedy heuristics [10], and Lagrangean relaxation [11]. Lorena and Pereira [12] report results obtained with a Lagrangean/surrogate heuristic using a subgradient optimization method, in complement to the dissociated Lagrangean and surrogate heuristics presented in [13]. Arakaki and Lorena [14] present a constructive genetic algorithm to solve real case instances with up to 500 vertices. Surveys can be found in [15–18].

In this paper is presented a cluster relaxation technique to solve large-scale maximal covering location problems. The proposed approach requires the identification of a graph related to a set of constraints. If some of these constraints are relaxed, this graph can be partitioned into subgraphs (clusters), corresponding to smaller problems that can be solved independently.

This paper is organized as follows. Section 2 presents a decomposition approach to obtain improved upper bounds to the optimal solution of maximal covering location problems. Section 3 describes computational results. Some conclusions are given in Section 4.

2. The Proposed Approach for the MCLP

The MCLP was formulated in [2] as the following 0-1 linear programming problem:

$$\text{(MCLP)} \qquad v(\text{MCLP}) = \text{Max} \sum_{i \in N} w_i x_i \tag{2.1}$$

$$\text{subject to} \quad \sum_{j \in S_i} y_j \geq x_i, \quad \forall i \in N, \tag{2.2}$$

$$\sum_{j \in M} y_j = p, \tag{2.3}$$

$$x_i \in \{0,1\}, \quad \forall i \in N, \tag{2.4}$$

$$y_j \in \{0,1\}, \quad \forall j \in M, \tag{2.5}$$

where

(i) $M = \{1, 2, \ldots, m\}$ is the set of potential facility locations;

(ii) $N = \{1, 2, \ldots, n\}$ is the set of clients to be covered;

(iii) $D = [d_{ij}]$ is the Euclidean distance matrix between each pair of nodes $i \in N$ and $j \in M$;

(iv) U is the covering distance;

(v) $S_i = \{j \in M \mid d_{ij} \leq U\}$ is the set of facilities that can attend each client $i \in N$;

(vi) w_i is the demand (a positive integer value) for each client $i \in N$;

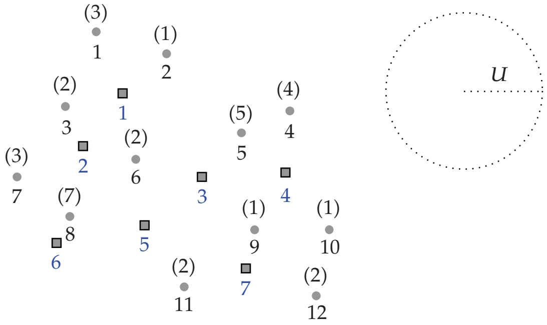


Figure 1: A MCLP instance.

(vii) p is the number of facilities to be located;

(viii) x_i is a decision variable, with $x_i = 1$ if the demand of the client i is covered, and $x_i = 0$, otherwise;

(ix) y_j is a decision variable, with $y_j = 1$ if a facility was installed at the location j, and $y_j = 0$, otherwise.

The objective function maximizes the covered demand. Constraints (2.2) state that a client will be covered if there is at least one facility located within the covering distance. Constraint (2.3) limits to exactly p the number of located facilities and (2.4) and (2.5) express the binary conditions.

The traditional Lagrangean relaxation approach [11] relaxes the set of constraints (2.2) with a vector μ of multipliers $\mu_i \geq 0$, $i \in N$, obtaining

$$(\mathrm{LR}_\mu) \qquad v(\mathrm{LR}_\mu) = \mathrm{Max} \sum_{i \in N} (w_i - \mu_i) x_i + \sum_{i \in N} \sum_{j \in S_i} \mu_i y_j \tag{2.6}$$

subject to (2.3), (2.4), and (2.5).

It is easy to see by the integrality property that $v(\mathrm{LR}_\mu) \geq v(\mathrm{MCLP})$ and the Lagrangean bound cannot be better than the linear relaxation of (MCLP).

In this paper, a decomposition approach based on the Lagrangean relaxation with clusters (LagClus) of Ribeiro and Lorena [19–22] is presented. LagClus is a stronger relaxation that can be useful for several theoretical and practical large-scale problems. The first application of the LagClus was performed on point-feature instances. Later, Ribeiro and Lorena applied this relaxation on pallet loading instances obtaining good results. Besides, the authors proposed a column generation for that problem using this cluster relaxation idea. Another interesting application was performed on wood pulp stowage context. This problem consists of arranging items into holds of dedicated maritime international ships. Recently, [23] applied the LagClus to uncapacitated facility location instances providing better bounds than the ones presented in the literature for a set of difficult instances.

Consider the MCLP instance represented in Figure 1, where the dots correspond to the clients to be covered ($N = \{1, \ldots, 12\}$) and the small squares correspond to the potential facility locations ($M = \{1, \ldots, 7\}$). In this figure, the values in parenthesis are the demand values w_i, for all $i \in N$. For the chosen covering distance, the sets S_i, for all $i \in N$, are defined as follows:

$$\begin{aligned}
&S_1 = \{1\}, \quad &&S_4 = \{4\}, \quad &&S_7 = \{2,6\}, \quad &&S_{10} = \{4\},\\
&S_2 = \{1\}, \quad &&S_5 = \{3,4\}, \quad &&S_8 = \{2,5,6\}, \quad &&S_{11} = \{5,7\},\\
&S_3 = \{1,2\}, \quad &&S_6 = \{1,2,3,5\}, \quad &&S_9 = \{3,4,7\}, \quad &&S_{12} = \{7\}.
\end{aligned} \tag{2.7}$$

Assuming the number of facilities to be installed as $p = 3$, this instance can be formulated as

$$v(\text{MCLP}) = \text{Max}\, 3x_1 + x_2 + 2x_3 + 4x_4 + 5x_5 + 2x_6 + 3x_7 + 7x_8 + x_9 + x_{10} + 2x_{11} + 2x_{12} \tag{2.8}$$

$$\text{subject to} \quad y_1 \geq x_1 \tag{2.9}$$

$$y_1 \geq x_2, \tag{2.10}$$

$$y_1 + y_2 \geq x_3, \tag{2.11}$$

$$y_4 \geq x_4, \tag{2.12}$$

$$y_3 + y_4 \geq x_5, \tag{2.13}$$

$$y_1 + y_2 + y_3 + y_5 \geq x_6, \tag{2.14}$$

$$y_2 + y_6 \geq x_7, \tag{2.15}$$

$$y_2 + y_5 + y_6 \geq x_8, \tag{2.16}$$

$$y_3 + y_4 + y_7 \geq x_9, \tag{2.17}$$

$$y_4 \geq x_{10}, \tag{2.18}$$

$$y_5 + y_7 \geq x_{11}, \tag{2.19}$$

$$y_7 \geq x_{12}, \tag{2.20}$$

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 = 3, \tag{2.21}$$

$$x_i \in \{0,1\}, \quad \forall i \in N, \tag{2.22}$$

$$y_j \in \{0,1\}, \quad \forall j \in M. \tag{2.23}$$

The proposed approach considers the MCLP as a *covering graph*, which is defined as follows.

Let $G(M, A)$ be a graph where M is the set of vertices corresponding to the potential facility locations and $A = \{(p,q) : p \text{ and } q \in S_i,\ i = 1,\ldots,N\}$ is the set of edges. So, in a covering graph there exists an edge (p,q) connecting two potential facility locations, if p and q share, at least, one covered client. Figure 2 shows the covering graph associated with the above MCLP instance.

It is easy to note that the edges in a covering graph are related to the set of constraints (2.2) in the MCLP formulation. For example, the edges shown in Figure 2 correspond to the constraints (2.11), (2.13)–(2.17), and (2.19).

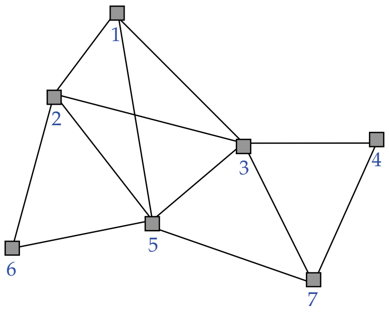
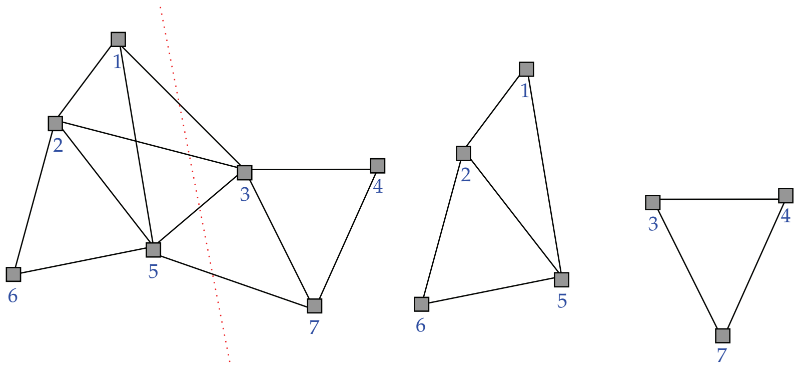


Figure 2: A covering graph.



(a) A possible partition

(b) The corresponding subgraphs

Figure 3: Partitioning a covering graph.

Now, consider that a covering graph is partitioned in some way. Figure 3(a) shows a possible partition. If the edges $(1,3)$, $(2,3)$, $(3,5)$, and $(5,7)$ are removed from the graph, two subgraphs are obtained, as shown in Figure 3(b).

This partition corresponds to relax in the Lagrangean way the constraints (2.14) and (2.19) of the MCLP formulation, using Lagrangean multipliers λ_6 and λ_{11}, respectively. If constraint (2.3) is also relaxed with Lagrangean multiplier μ, the relaxed problem will be

$$\begin{aligned} v(\mathrm{MCLP}_R) = \operatorname{Max} & \, 3x_1 + x_2 + 2x_3 + 4x_4 + 5x_5 + 2x_6 + 3x_7 + 7x_8 + x_9 + x_{10} + 2x_{11} + 2x_{12} \\ & + \lambda_6 \left(y_1 + y_2 + y_3 + y_5 - x_6 \right) + \lambda_{11} \left(y_5 + y_7 - x_{11} \right) \\ & + \mu \left(y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 - 3 \right) \end{aligned} \tag{2.24}$$

subject to (2.9)–(2.13), (2.15)–(2.18), (2.20), (2.22), (2.23), and

$$\begin{gathered} \lambda_6, \lambda_{11} \geq 0, \\ \mu \in R. \end{gathered} \tag{2.25}$$

The objective function can now be rewritten as

$$\begin{aligned} v(\mathrm{MCLP}_R) = \mathrm{Max}\, & 3x_1 + x_2 + 2x_3 + 4x_4 + 5x_5 + (2 - \lambda_6)x_6 + 3x_7 + 7x_8 + x_9 + x_{10} \\ & + (2 - \lambda_{11})x_{11} + 2x_{12} + (\lambda_6 + \mu)y_1 + (\lambda_6 + \mu)y_2 + (\lambda_6 + \mu)y_3 \\ & + \mu y_4 + (\lambda_6 + \lambda_{11} + \mu)y_5 + \mu y_6 + (\lambda_{11} + \mu)y_7 - 3\mu. \end{aligned} \tag{2.26}$$

Then, the problem can be decomposed in two subproblems:

$$v(\mathrm{MCLP}_R) = v(\mathrm{SP}_1) + v(\mathrm{SP}_2) + (2 - \lambda_6)x_6 + (2 - \lambda_{11})x_{11} - 3\mu, \tag{2.27}$$

where

$$\begin{aligned} v(\mathrm{SP}_1) = \mathrm{Max}\, & 3x_1 + x_2 + 2x_3 + 3x_7 + 7x_8 \\ & + (\lambda_6 + \mu)y_1 + (\lambda_6 + \mu)y_2 + (\lambda_6 + \lambda_{11} + \mu)y_5 + \mu y_6 \end{aligned} \tag{2.28}$$

subject to (2.9)–(2.11), (2.15), (2.16), (2.22), (2.23), and

$$\begin{gathered} \lambda_6, \lambda_{11} \geq 0, \\ \mu \in R, \end{gathered} \tag{2.29}$$

and

$$v(\mathrm{SP}_2) = \mathrm{Max}\, 4x_4 + 5x_5 + x_9 + x_{10} + 2x_{12} + (\lambda_6 + \mu)y_3 + \mu y_4 + (\lambda_{11} + \mu)y_7 \tag{2.30}$$

subject to (2.12), (2.13), (2.17), (2.18), (2.20), (2.22), (2.23), and

$$\begin{gathered} \lambda_6, \lambda_{11} \geq 0, \\ \mu \in R. \end{gathered} \tag{2.31}$$

Note that these subproblems correspond to the following clusters (which are associated with the subgraphs of the covering graph):

(i) Cluster 1: $M_1 = \{1, 2, 3, 7, 8\}$ and $N_1 = \{1, 2, 5, 6\}$;

(ii) Cluster 2: $M_2 = \{4, 5, 6, 9, 10, 11, 12\}$ and $N_2 = \{3, 4, 7\}$.

The resulting Lagrangean relaxation does not have the integrality property, being stronger than (LR_μ) of Galvão and ReVelle [11]. As the clusters are smaller than the original covering graph, exact methods can be employed to solve each corresponding subproblem, obtaining better quality bounds in shorter computational times.

For the above example, one can apply a subgradient optimization method in order to determine the values of the dual variables λ_6, λ_{11}, and μ. At each iteration of this method, the values of the dual variables are used to update the coefficients of the objective function of each subproblem, which can be solved by exact methods.

It is interesting to observe that, due to the relaxation of constraints (2.14) and (2.19), the variables x_6 and x_{11} disappeared from the formulations of the subproblems. According to the objective function (2.27) their values will be set as 0 or 1, depending on their respective coefficient at each iteration:

$$x_i = \begin{cases} 1, & \text{if } w_i - \lambda_i > 0, \\ 0, & \text{otherwise} \end{cases} \tag{2.32}$$

for all $i \in I = \{i \in N \mid \text{constraint (2.2) containing variable } x_i \text{ is relaxed}\}$.

Therefore, the proposed decomposition approach can be established in the following steps.

(a) Build a covering graph $G(M, A)$ corresponding to the MCLP.

(b) Apply a graph partitioning heuristic to divide the covering graph G into k clusters. The MCLP can be written through the objective function defined in (2.1) subject to (2.2)–(2.5) where the constraints (2.2) are now divided into two groups: one with constraints corresponding to intracluster edges and other formed by constraints that correspond to edges connecting the clusters.

(c) Using distinct nonnegative multipliers, relax in the Lagrangean way the constraints corresponding to the edges connecting the clusters (defining the set I) and also relax constraint (2.3).

(d) The resulting Lagrangean relaxation is decomposed into k subproblems.

(e) Apply the standard subgradient method in order to optimize the dual variables λ and μ.

The subgradient method used in the step (e) can be written as follows.

Set, initially,

$$\begin{gathered} \lambda_i = \begin{cases} w_i, & \text{if } i \in I, \\ 0, & \text{otherwise,} \end{cases} \\ \mu = 0, \\ \theta = 2, \\ \text{LB} = -\infty, \\ \text{UB} = +\infty, \end{gathered} \tag{2.33}$$

While (the stop conditions are not satisfied) do the following

Solve subproblems SP_k for x and y, using the current values of λ and μ.

Calculate

$$v(\text{MCLP}_R) = \sum_k v(\text{SP}_k) + \sum_{i \in I} \max\{0, (w_i - \lambda_i)\} - p\mu. \tag{2.34}$$

Update $\text{UB} = \min\{\text{UB}, v(\text{MCLP}_R)\}$.

```
    If $\sum_{j\in N} y_j = p$ (a feasible solution for MCLP is found), then do the following.
        Calculate $v(\text{MCLP})$ using the obtained values for $x$ and $y$.
        Update $\text{LB} = \max\{\text{LB}, v(\text{MCLP})\}$.
    Calculate
        $g_i^{\lambda} = x_i - \sum_{j\in S_i} y_j, \quad i \in I,$
        $g^{\mu} = p - \sum_{j\in N} y_j.$                                  (2.35)
    Update the step size $\theta$.
    Update
        $\lambda_i = \min\left\{0, \lambda_i + \theta g_i^{\lambda}\right\}, \quad i \in I,$
        $\mu = \mu + \theta g^{\mu}.$                                        (2.36)
End-While.
```

The step size θ used in this algorithm is the one proposed in [24], beginning with $\theta = 2$ and halving it whenever the upper bound does not decrease for a certain number of successive iterations. The stopping tests used are the following:

(a) $\theta \leq 0.005$, or

(b) $\text{UB} - \text{LB} < 1$, or

(c) the subgradient vector $g = [g_i^{\lambda}, g^{\mu}] = 0$.

3. Computational Results

The LagClus algorithm was coded in C and the tests were conducted on a notebook with Intel Core 2 Duo 2.0 GHz processor and 2.0 GB RAM, running Windows XP (Service Pack 3), and ILOG CPLEX 10.1.1 [25]. The data were obtained from TSPLIB PCB3038 [26] and real case instances for facility location problems in São José dos Campos city, Brazil (available for download at http://www.lac.inpe.br/~lorena/instancias.html).

For the graph partitioning task was used the well-known METIS heuristic for graph partitioning problems [27], with default values. Given a covering graph G and a predefined number k of clusters, METIS divides the graph in k clusters minimizing the number of edges with terminations in different clusters.

The results obtained are shown in Tables 1 to 3. These tables use the following legend in the columns

(i) *k:* number of clusters;

(ii) *n*: number of potential facilities locations and clients to be covered;

(iii) *p*: number of facilities to be installed;

(iv) *Optimal*: optimal solution of the corresponding MCLP obtained by CPLEX;

(v) *GapLP*: linear relaxation gap provided by CPLEX (in percentage);

Table 1: Computation times for SJC instances, $U = 150$.

					$k = 10$				$k = 50$			
n	*p*	*Optimal*	*GapLP*	*CPU*	*Cuts*	*Gap*	*CPUSeq*	*CPUPar*	*Cuts*	*Gap*	*CPUSeq*	*CPUPar*
324	20	7302	0.000	0.015	296	**0.000**	2.543	2.537	1449	**0.000**	9.875	9.739
	30	9127	0.027	0.047		**0.027**	24.650	24.635		**0.027**	44.596	41.101
	40	10443	0.156	0.188		**0.108**	25.985	25.969		0.157	45.122	42.353
	50	11397	0.180	0.391		**0.138**	24.452	24.422		0.195	43.748	43.459
	60	11991	0.184	0.235		**0.024**	44.514	44.421		0.221	48.049	47.909
	80	12152	0.000	0.031		**0.000**	8.876	8.816		0.003	109.129	108.115
	108	12152	0.000	0.016		**0.000**	1.595	1.595		2.977	26.533	26.487
500	40	13340	0.000	0.047	108	**0.000**	3.453	3.361	804	**0.000**	17.186	11.499
	50	14773	0.014	0.047		**0.000**	4.938	4.514		**0.014**	59.019	36.293
	60	15919	0.048	0.063		**0.000**	8.233	7.243		**0.048**	57.157	34.737
	70	16908	0.000	0.031		**0.000**	3.723	3.370		**0.000**	22.891	14.421
	80	17749	0.000	0.015		**0.000**	5.406	4.766		**0.000**	26.686	16.697
	100	18912	0.098	0.109		**0.000**	10.276	7.171		**0.056**	62.071	37.748
	130	19664	0.041	0.297		**0.015**	30.827	24.588		**0.041**	69.934	43.373
	167	19706	0.005	0.047		**0.003**	14.600	14.235		**0.005**	46.078	35.414
818	80	23325	0.055	0.140	166	**0.003**	45.564	21.880	1649	0.061	85.922	45.819
	90	24455	0.123	0.266		**0.041**	56.388	24.747		0.143	87.797	47.001
	100	25435	0.127	0.344		**0.012**	87.279	34.481		0.140	96.124	52.060
	120	26982	0.084	0.297		**0.015**	69.658	31.368		**0.062**	105.547	54.658
	140	28002	0.140	0.359		**0.095**	52.966	26.271		**0.128**	121.127	63.713
	160	28699	0.128	0.391		**0.107**	58.453	24.904		**0.126**	96.017	50.828
	200	29153	0.018	0.234		**0.011**	61.531	28.301		0.039	253.908	135.048
	273	29168	0.000	0.031		**0.000**	3.343	2.545		0.554	46.766	37.178

(vi) *CPU*: computing time of the corresponding optimal solution obtained by CPLEX (in seconds);

(vii) *Cuts*: number of relaxed constraints;

(viii) $Gap = 100\% \times (UB - Optimal)/Optimal$;

(ix) *CPUSeq*: sum of the computing times for every cluster, at each iteration (in seconds);

(x) *CPUPar*: the largest computing time for a cluster, at each iteration (in seconds).

The values marked with an asterisk in Table 3 denote instances where CPLEX failed to produce an optimal solution within the time limit of 20000 seconds. The presented figures are suboptimal values.

From these results one can observe that the smaller the number of clusters is, the better are the upper bounds obtained (smaller are the gaps). On the other hand, as the number of clusters increases, the computational effort for solving the subproblems is reduced.

In the results shown that for $k = 10$ (SJC instances) and $k = 5$ (TSPLIB PCB3038 instance), the gaps obtained for 92.3% of the instances (48 out of 52) are equal or smaller (values presented in bold face) than the gaps obtained by the linear relaxation, which demonstrates the effectiveness of the decomposition heuristic. However, improved gaps could be obtained by reducing the number of clusters, at the cost of larger computational times.

Therefore, as shown in Tables 1, 2, and 3, the bounds obtained by the proposed approach are better than the ones produced by the linear relaxation and, consequently,

Table 2: Computation times for SJC instances, $U = 200$.

					$k = 10$				$k = 50$			
n	*p*	*Optimal*	*GapLP*	*CPU*	*Cuts*	*Gap*	*CPUSeq*	*CPUPar*	*Cuts*	*Gap*	*CPUSeq*	*CPUPar*
324	20	9670	0.334	0.172	980	**0.243**	19.293	19.293	3008	0.347	32.943	32.911
	30	11737	0.087	0.484		**0.060**	28.943	28.943		0.094	69.959	69.959
	40	12151	0.005	0.094		**0.008**	31.066	31.066		0.138	77.872	77.612
	50	12152	0.000	0.015		**0.000**	9.926	9.926		**0.000**	61.456	61.395
	60	12152	0.000	0.047		**0.000**	4.575	4.575		**0.000**	14.376	14.376
	80	12152	0.000	0.016		**0.000**	3.670	3.670		**0.000**	33.936	33.936
	108	12152	0.000	0.031		0.248	11.343	11.343		0.001	897.830	893.405
500	40	17077	0.453	0.203	657	**0.387**	24.668	20.669	2625	0.469	55.001	51.048
	50	18361	0.014	0.109		**0.003**	39.109	32.248		0.025	67.626	62.596
	60	19153	0.035	0.063		**0.005**	52.639	35.363		0.112	85.374	76.578
	70	19551	0.110	1.078		**0.069**	43.946	29.817		0.170	76.671	68.721
	80	19703	0.013	0.156		**0.008**	35.495	27.927		0.150	102.056	95.253
	100	19707	0.000	0.078		**0.000**	16.624	16.501		0.001	89.858	86.698
	130	19707	0.000	0.047		**0.000**	1.986	1.864		0.001	26.546	25.809
	167	19707	0.000	0.016		0.016	22.379	22.225		0.859	22.314	22.133
818	80	27945	0.070	0.203	840	**0.069**	57.835	27.423	4910	0.121	147.155	115.605
	90	28519	0.138	1.141		**0.071**	114.145	45.536		0.177	128.574	99.585
	100	28910	0.103	1.391		**0.036**	88.885	33.153		0.175	101.875	80.758
	120	29165	0.002	1.234		**0.002**	55.710	31.180		0.117	141.246	115.434
	140	29168	0.000	0.125		**0.000**	11.643	8.940		0.021	171.961	143.737
	160	29168	0.000	0.062		**0.000**	9.738	7.598		0.878	39.343	36.244
	200	29168	0.000	0.032		**0.000**	5.762	4.610		0.847	37.205	34.871
	273	29168	0.000	0.031		0.207	24.698	18.505		2.904	25.282	23.772

Table 3: Computation times for TSPLIB PCB3038 instance, $U = 400$.

					$k = 5$				$k = 10$			
n	*p*	*Optimal*	*GapLP*	*CPU*	*Cuts*	*Gap*	*CPUSeq*	*CPUPar*	*Cuts*	*Gap*	*CPUSeq*	*CPUPar*
3038	17	125320	0.368	802.390	165579	**0.205**	843.838	235.541	291363	0.470	582.528	223.245
	18	130004	0.517	10265.016		**0.372**	817.076	283.400		0.712	634.402	243.747
	19	134262*	0.605	20000.049		**0.382**	1483.237	598.653		0.793	576.821	222.087
	20	139028*	0.698	20000.156		**0.500**	1712.078	798.911		0.973	628.288	236.767
	21	141279*	0.853	20000.094		**0.654**	3117.174	1448.730		1.128	646.765	243.302
	22	143809*	1.196	20000.123		**0.992**	6656.267	3094.410		1.598	615.783	231.525

by any Lagrangean relaxation as the ones presented in [11–13]. Nevertheless, comparing the values of the CPU times presented by this method and those presented by CPLEX, it is clear that the proposed decomposition approach is appropriate only for large-scale problems.

Comparing the values of *CPUSeq* and *CPUPar* for each value of k, it is also possible to note that *CPUPar* values are becoming relatively smaller than *CPUSeq* values as the size of the instances increases, indicating that the proposed decomposition approach can substantially reduce the time for solving large instances of MCLP in parallel (or multicore CPU) computers.

4. Conclusions

This paper presents a decomposition approach based on cluster partitioning to calculate improved upper bounds to the optimal solution of maximal covering location problems. The partitioning is based on the covering graph of potential facility locations that attend a same client. The corresponding coupling constraints are identified and some of them are relaxed in the Lagrangean way, resulting in subproblems that can be solved independently. Each subproblem represents a cluster smaller than the original problem and can be solved by exact methods in smaller computational times. Computational tests using real data and instances from the available literature were conducted and confirmed the effectiveness of the proposed approach.

An important characteristic of large-scale problems addressed by the proposed approach is the tradeoff between gap values and CPU times. Depending on the application, one may choose to sacrifice the quality of the bounds (increasing the number of clusters) in order to obtain shorter processing times. On the other hand, if quality is the issue, the processing times needed to solve instances with only a few clusters may be longer.

In this study, the number of clusters was fixed *a priori* and then the number of intercluster edges was minimized by the partitioning algorithm. In this way, for a chosen k, the number of relaxed constraints is minimized and, consequently, better bounds are obtained. The partitioning is clearly crucial and other choices could be considered, for example, minimize the maximal size of the clusters, so as to obtain the smallest possible subproblems. This has yet to be investigated.

The heuristic presented in this article can be used in a branch-and-bound exact method. As, in general, the upper bounds obtained with this heuristic are better than those obtained by the linear relaxation, one would expect many more nodes be pruned, with a possibly significant size reduction of the search tree.

Advances in applied mathematics and computer science have resulted in high-performance tools for mathematical programming, allowing tough optimization problems to be solved. However, as the problem size increases, the computational time may grow excessively, making the problem intractable even for the most efficient tools. In such a case an approach in which a large-scale problem is divided into a number of smaller-scale subproblems can be a nice solution possibility.

Acknowledgments

The authors acknowledge CNPq—*Conselho Nacional de Desenvolvimento Científico e Tecnológico* for partial financial research support. They also would like to thank the valuable suggestions of two anonymous referees.

References

[1] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, “The location of emergency service facilities,” *Operations Research*, vol. 19, pp. 1363–1373, 1971.

[2] R. Church and C. ReVelle, “The maximal covering location problem,” *Papers of the Regional Science Association*, vol. 32, no. 1, pp. 101–118, 1974.

[3] D. J. Eaton, M. L. Hector, U. Sanchez, R. Lantigua, and J. Morgan, “Determining ambulance deployment in Santo Domingo. Dominican Republic,” *Journal of the Operational Research Society*, vol. 37, no. 2, pp. 113–126, 1986.

[4] J. R. Current and M. O’Kelly, “Locating emergency warning sirens,” *Decision Sciences*, vol. 23, pp. 221–234, 1981.

[5] G. C. Moore and C. S. ReVelle, "The hierarchical service location problem," *Management Science*, vol. 28, pp. 775–780, 1982.

[6] E. S. Hougland and N. T. Stephens, "Air pollutant monitor sitting by analytical techniques," *Journal of the Air Pollution Control Association*, vol. 26, pp. 52–53, 1976.

[7] V. Marianov and D. Serra, "Probabilistic, maximal covering location-allocation models from congested systems," *Journal of Regional Science*, vol. 38, no. 3, pp. 401–424, 1998.

[8] V. Marianov and D. Serra, "Hierarchical location-allocation models for congested systems," *European Journal of Operational Research*, vol. 135, no. 1, pp. 195–208, 2001.

[9] F. A. Corrêa and L. A. N. Lorena, "Using the constructive genetic algorithm for solving the probabilistic maximal covering location-allocation problem," in *Proceedings of the 1st Workshop on Computational Intelligence/SBRN*, 2006, http://www.lac.inpe.br/~lorena/correa/Correa_Lorena_Wci_2006.pdf.

[10] M. Daskin, *Network and Discrete Location: Models Algorithms and Applications*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, NY, USA, 1995.

[11] R. D. Galvão and C. ReVelle, "A Lagrangean heuristic for the maximal covering location problem," *European Journal of Operational Research*, vol. 88, no. 1, pp. 114–123, 1996.

[12] L. A. N. Lorena and M. A. Pereira, "A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition," *International Journal of Industrial Engineering*, vol. 9, no. 1, pp. 57–67, 2002.

[13] R. D. Galvão, L. G. A. Espejo, and B. Boffey, "A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem," *European Journal of Operational Research*, vol. 124, no. 2, pp. 377–389, 2000.

[14] R. G. I. Arakaki and L. A. N. Lorena, "A constructive genetic algorithm for the maximal covering location problem," in *Proceedings of the 4th Metaheuristics International Conference (MIC '01)*, pp. 13–17, Porto. Portugal, July 2001, http://www.lac.inpe.br/~lorena/arakaki/cgalap.pdf.

[15] C.-H. Chung, "Recent applications of the maximal covering location problem (MCLP) model," *Journal of the Operational Research Society*, vol. 37, no. 8, pp. 735–746, 1986.

[16] T. S. Hale and C. R. Moberg, "Location science research: a review," *Annals of Operations Research*, vol. 123, no. 1–4, pp. 21–35, 2003.

[17] D. Serra and V. Marianov, "New trends in public facility location modeling," Working Paper 755, Universitat Pompeu Fabra Economics and Business, 2004, http://www.econ.upf.edu/docs/papers/downloads/755.pdf.

[18] R. D. Galvão, "Uncapacitated facility location problems: contributions," *Pesquisa Operacional*, vol. 24, pp. 7–38, 2004.

[19] G. M. Ribeiro and L. A. N. Lorena, "Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem," *Computers & Operations Research*, vol. 34, no. 9, pp. 2695–2708, 2007.

[20] G. M. Ribeiro and L. A. N. Lorena, "Lagrangean relaxation with clusters for point-feature cartographic label placement problems," *Computers & Operations Research*, vol. 35, no. 7, pp. 2129–2140, 2008.

[21] G. M. Ribeiro and L. A. N. Lorena, "Optimizing the woodpulp stowage using Lagrangean relaxation with clusters," *Journal of the Operational Research Society*, vol. 59, no. 5, pp. 600–606, 2008.

[22] G. M. Ribeiro and L. A. N. Lorena, "Column generation approach for the point-feature cartographic label placement problem," *Journal of Combinatorial Optimization*, vol. 15, no. 2, pp. 147–164, 2008.

[23] F. A. Corrêa, L. A. N. Lorena, and E. L. F. Senne, "Lagrangean relaxation with clusters for the uncapacitated facility location problem," in *Proceedings of the 13th CLAIO—Congreso Latino-Iberoamericano de Investigación Operativa*, Montevideo, Uruguay, November 2006, http://www.lac.inpe.br/~lorena/correa/LagClus_UFLP_CLAIO2006.pdf.

[24] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees: part II," *Mathematical Programming*, vol. 1, no. 1, pp. 6–25, 1971.

[25] ILOG, "CPLEX 10.1.1: User's Manual," France, 2006.

[26] G. Reinelt, *The Traveling Salesman Problem: Computational Solutions for TSP Applications*, vol. 840 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 1984.

[27] G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *Journal of Parallel and Distributed Computing*, vol. 48, no. 1, pp. 96–129, 1998.