LOCAL SEARCH HEURISTICS FOR CAPACITATED P-MEDIAN PROBLEMS

Luiz Antonio Nogueira Lorena
LAC/INPE - Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas, 1758 - Caixa Postal 515
12245-970 São José dos Campos, SP – Brazil

Edson Luiz França Senne
FEG/UNESP - Universidade Estadual Paulista
Faculdade de Engenharia - Departamento de Matemática
12500-000 Guaratinguetá, SP - Brazil

Abstract

The search for *p-median* vertices on a network (graph) is a classical location problem. The *p* facilities (medians) must be located so as to minimize the sum of the distances from each demand vertex to its nearest facility. The *Capacitated p-Median Problem (CPMP)* considers capacities for the service to be given by each median. The total service demanded by vertices identified by p-median clusters cannot exceed their service capacity. Primal-dual based heuristics are very competitive and provide simultaneously upper and lower bounds to optimal solutions. The Lagrangean/surrogate relaxation has been used recently to accelerate subgradient like methods. The dual lower bound have the same quality of the usual Lagrangean relaxation dual but is obtained using modest computational times. This paper explores improvements on upper bounds applying local search heuristics to solutions made feasible by the Lagrangean/surrogate optimization process. These heuristics are based on location-allocation procedures that swap medians and vertices inside the clusters, reallocate vertices, and iterate until no improvements occur. Computational results consider instances from the literature and real data obtained using a geographical information system.

Key words: Location problems, Capacitated p-median problems, Clustering, Lagrangean/surrogate relaxation, Subgradient method.

Correspondences to: Luiz Antonio Nogueira Lorena, LAC/INPE - Caixa Postal 515, 12.201-970 São José dos Campos, SP – Brazil

1. Introduction

Clustering problems generally appear in classification of data for some purpose like storage and retrieval or data analysis. Any clustering algorithm will attempt to determine some inherent or natural grouping in the data, using “distance” or “similarity” measures between individual data (Spath [17]). In this paper we examine local search heuristics to a clustering problem in graphs, namely, the *capacitated p-median problem (CPMP)*.

The search for *p-median* vertices on a network (graph) is a classical location problem. The objective is locate *p* facilities (medians) so that the sum of the distances from each demand vertex to its nearest facility is minimized. The *CPMP* considers capacities for the service to be given by each median. The total service demanded by vertices identified by p-median clusters cannot exceed their service capacity.

Apparently the CPMP problem is not as intensively studied as the classical p-median problem. Mulvey and Beck [13] examine the Lagrangean relaxation of assignment constraints in a 0-1 linear integer programming problem formulation. A primal assignment heuristic is embedded within a subgradient method, improved by interchanging medians in clusters. Koskosidis and Powell [7] improve the Mulvey and Beck’s results suggesting various algorithms to find initial solutions for knapsack problems (Lagrangean subproblems). In the same lines Bramel and Simchi-Levi [1] solve a similar problem considering fixed costs. Osman and Christofides [15] apply variations of simulated annealing and tabu search to obtain good approximated solutions to the problem, improving an initial set of medians. An extensive bibliography of related problems, and a set of test problems is also presented in [15]. Similar problems also appear in Klein and Aronson [6], Mulvey and Beck [13] and Maniezzo, Mingozzi and Baldacci [11].

The Lagrangean/surrogate approach presented in this paper goes in the lines of the early Lagrangean heuristics, considering a binary integer programming formulation. The Lagrangean/surrogate relaxation has been used recently to accelerate subgradient-like methods, which are often used to optimize the corresponding Lagrangean dual problem. It was tested before with success on Set Covering problems [8], Generalized Assignment problems [9,14] and some Location problems [10,16]. Narciso and Lorena [14] coined the name *Lagrangean/surrogate* for this kind of relaxation. The Lagrangean/surrogate dual bound has the same quality of the usual Lagrangean dual bound but is obtained with modest computational times.

This paper explores improvements on upper bounds, the Lagrangean/surrogate primal counterpart. The Lagrangean/surrogate relaxation is combined with location-allocation heuristics, proposed by Cooper [2] and used before on Taillard [18] and Senne and Lorena [16]. The heuristics improve solutions made feasible by the Lagrangean/surrogate optimization process, swapping medians and vertices inside the clusters, reallocating vertices, and iterating until no more improvements occur. One of the goals of our approach is to apply to a large scale real data obtained using Geographic Information Systems. Our proposal compares favorably to some metaheuristics and seems to be indicated to real data due to small computational times.

The paper is organized in the following sections. Section 2 summarizes the Lagrangean/surrogate approach. The Lagrangean/surrogate heuristic to *CPMP* is presented in section 3. Section 4 presents the local search heuristics and section 5 the computational results for a set of classical instances and a set of real data collected at the central area of a Brazilian 500,000 inhabitant's city.

2. Lagrangean/Surrogate Approach

The Lagrangean/surrogate approach is a successful substitute to the ordinary Lagrangean relaxation. The Lagrangean/surrogate dual bound has the same quality as the usual Lagrangean dual bound but is obtained with modest computational effort. A general description for Lagrangean/surrogate relaxation appears in [10] and [14]. This section reviews the Lagrangean/surrogate relaxation for a general 0-1 linear integer programming. The next section applies the Lagrangean/surrogate relaxation to *CPMP.*

Let us suppose, in general, the following 0-1 linear programming problem:

$$(P) \qquad \begin{aligned} v(P) = Min \;\; & cx \\ subject\;\; to \;\; & Ax \geq b, \\ & Dx \geq e, \\ & x \in \{0,1\}^n \end{aligned}$$

where $c \in R^n, A \in R^{mxn}, b \in R^m, D \in R^{pxn} and \; e \in R^p$.

Think $Dx \geq e$ as the easily enforced constraints and $Ax \geq b$ as complicating ones. Defining X = $\{x \in \{0,1\}^n \mid Dx \geq e\}$, for a given multiplier $l \in R^m_+$ the Lagrangean relaxation of (P) is:

$$(LP^l) \qquad \begin{aligned} & v(LP^l) = Min\{cx - l\,(Ax - b)\} \\ & subject\; to \quad x \in X. \end{aligned}$$

The Lagrangean dual is then the optimization problem in λ, that is:

$$(D_\lambda) \qquad \begin{array}{l} v(D_\lambda) = Max\{v(LP^\lambda)\} \\ subject\ to\ \ \lambda \in R^m_+ . \end{array}$$

The Lagrangean multipliers can be considered *surrogate* multipliers for the same set of constraints relaxed, and a new Lagrangean relaxation is identified for the surrogate constraint. The surrogate problem of (P) is:

$$(SP^\lambda) \qquad \begin{array}{l} v(SP^\lambda) = Min\ cx \\ subject\ \ to \quad \lambda\,(Ax-b) \geq 0, \\ \ x \in X . \end{array}$$

Frequently (SP^λ) is a difficult problem (like (P)) justifying a Lagrangean relaxation of the surrogate constraint in problem (SP^λ). Given a $\lambda \in R^m_+$, and a parameter $t \geq 0$, the ***Lagrangean/surrogate*** relaxation of (P) is then defined by:

$$(LP^{t\lambda}) \qquad \begin{array}{l} v(LP^{t\lambda}) = Min\ \{cx - t\lambda\,(Ax-b)\} \\ subject\ \ to\ \ x \in X . \end{array}$$

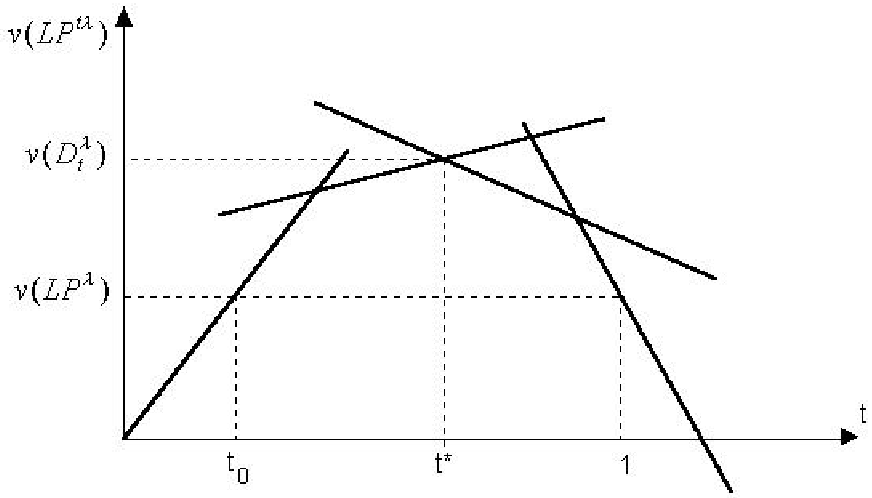

Figure 1: Lagrangean/surrogate bounds.

Parameter t is called Lagrangean/surrogate multiplier and the Lagrangean multipliers l are scaled by t. If t = 1 the Lagrangean/surrogate is the usual Lagrangean relaxation using the multiplier l. Considering the duality theory, it is well known that the Lagrangean function is concave and piecewise linear (see *Figure 1*). The best Lagrangean/surrogate multiplier t^* is obtained solving the dual $v(D_t^l) = \underset{t \geq 0}{Max}[v(LP^{tl})] = v(LP^{t^* l})$. However, in general, we have an interval of values t_0 £ T £ t_1 (with t_0 = 1 or t_1 = 1) which also produces improved bounds for the usual Lagrangean relaxation (in *Figure 1* we set, arbitrarily, $t_1 = 1$).

The following naive line search is used to approximately solve $v(D_t^l)$, calculating a T belonging to the interval $t_0 \leq T \leq t_1$ ($t_0 = 1$ or $t_1 = 1$):

Search Heuristic (SH)

s	is the initial step size;
k	is the number of iterations;
kmax	is the maximum number of iterations;
tinit	is the initial value of Lagrangean/surrogate multiplier;
t	is the current value of Lagrangean/surrogate multiplier;
tleft	is the current best left-approximate value for t*;
tright	is the current best right-approximate value for t*;
T	is the value of Lagrangean/surrogate multiplier (the best approximate value found for t*);
zleft	is the value of (LP^{tl}) for t = tleft;
zright	is the value of (LP^{tl}) for t = tright;
z	is the maximum value found for (LP^{tl}), that is, the value of (LP^{Tl});

Set

k = 0;
z = 0;
t = tinit;
T = t;
tleft = tright = undefined;
zleft = zright = undefined;

While (k < kmax) **do**

k = k + 1;
Solve (LP^{tl}) obtaining x^{λ};
If ($v(LP^{tl})$ > z) **then**

```
        z = v(LP^{tl});
        T = t;
        Calculate μ^λ = λ(Ax^λ − b) (μ^λ is the slope of the Lagrangean/surrogate function);
        If ( μ^λ > 0 ) then
            tleft = t;
            zleft = z;
            If ( tright is undefined ) then
                t = t + s;
            Else
                t = (tleft × zleft + tright × zright) / (zleft + zright);
                If ( v(LP^{tl}) > z ) then
                    z = v(LP^{tl});
                    T = t;
                End_If
                Stop.
            End_If
        Else
            tright = t;
            zright = z;
            If ( tleft is undefined ) then
                t = t – s;
            Else
                t = (tleft × zleft + tright × zright) / (zleft + zright);
                If ( v(LP^{tl}) > z ) then
                    z = v(LP^{tl});
                    T = t;
                End_If
                Stop.
            End_If
        End_If
    Else
        t = t − s/2;
        If ( v(LP^{tl}) > z ) then
            z = v(LP^{tl});
            T = t;
        End_If
        Stop.
    End_If
End_While.
```

The Lagrangean/surrogate relaxation can provide better bounds than their Lagrangean (alone) counterpart. For a fixed l and T (after the *SH* application), the following inequalities are valid:

$$v(LP^l) \leq v(LP^{Tl}) \leq v(D_t^l) \leq v(SP^l) \leq v(D_l) \leq v(P).$$

The Lagrangean dual (D_l) is optimized by a subgradient method. If the Lagrangean/surrogate relaxation is used, the new dual (D_{tl}) will have the same bound as (D_l), but the sequence of $v(LP^{Tl})$ can provide better bounds than the sequence $v(LP^l)$, accelerating the subgradient method.

3. Lagrangean/Surrogate application to CPMP

The *CPMP* is known to be NP-hard and some earlier approaches applying Lagrangean heuristics to *CPMP* are proposed in Koskosidis and Powell [7] and in [13]. Recent approaches apply metaheuristics, such as simulated annealing and tabu search (as in França, Sosa and Pureza [4] and in [15]), and genetic algorithms (Maniezzo, Mingozzi and Baldacci [11]). Good results are reported for a set of standard test problems (OR-Library - http://mscmga.ms.ic.ac.uk/info.html).

The *CPMP* considered in this paper is modeled as the following binary integer programming problem:

$$v(CPMP) = \textbf{\textit{Min}} \sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} \qquad (1)$$

(CPMP) subject to $$\sum_{j \in M} x_{ij} = 1 \; ; \; i \in N \qquad (2)$$

$$\sum_{j \in M} y_j = p \qquad (3)$$

$$\sum_{i \in N} q_i x_{ij} \le Q_j y_j \ ; \ j \in M \qquad (4)$$

$$y_j \in \{0,1\};\ x_{ij} \in \{0,1\};\ i \in N,\ j \in M \qquad (5)$$

where:

N = {1,...,n} is the index set of entities to allocate and M = {1,...,m} is the index set of possible medians, where p medians will be located;

q_i is the demand of each entity and Q_j the capacity of each possible median;

$[d_{ij}]_{n \times m}$ is a distance matrix;

$[x_{ij}]_{n \times m}$ is the allocation matrix, with x_{ij}=1 if entity i is allocated to median j, and x_{ij}=0, otherwise; y_j = 1 if median j is selected and y_j = 0, otherwise.

Constraints (2) and (3) enforce that each entity is allocated to only one median. Constraint (4) imposes that a total median capacity must be respected, and (5) gives the integer conditions.

For a given $\lambda \in R_+^n$ and $t \geq 0$ the Lagrangean/surrogate relaxation of *CPMP* is given by:

$$v(LCPMP^{tl}) = \boldsymbol{Min} \sum_{i \in N} \sum_{j \in M} (d_{ij} - t\,l_i) x_{ij} + t \sum_{i \in N} l_i \qquad (6)$$

$(LCPMP^{tl})$ subject to (3), (4) and (5).

Problem $(LCPMP^{tl})$ is solved considering the implicit constraint (3). Decomposing by index j, the following m 0-1 knapsack problems are obtained:

$$v(knap^j) = \boldsymbol{Min} \sum_{i \in N} (d_{ij} - t\, l_j) x_{ij} \qquad (7)$$

subject to (4) and (5).

Each problem is solved using the Horowitz and Sahni code (see Martello and Toth [12]).

Let J be the index set of the p smallest $v(knap^j)$, j $\in$ M (here constraint (3) is considered implicitly). The Lagrangean/surrogate value is given by:

$$v(LCPMP^{tl}) = \sum_{j \in J} v(knap^j) + t \sum_{i \in N} l_i \ . \qquad (8)$$

The best Lagrangean/surrogate relaxation value gives an improved bound to the usual Lagrangean relaxation. To find an approximated best Lagrangean/surrogate multiplier T we have used the search procedure *SH* described in section 2.

The following general subgradient algorithm is used to optimize the Lagrangean/surrogate dual. It combines the Lagrangean/surrogate bounds with improved primal feasible solutions.

Subgradient Heuristic (SubG)

Given $\lambda \geq 0$, $\lambda \neq 0$;

Set lb = $-\infty$, ub = $+\infty$;

Repeat

Solve relaxation $(LCPMP^{Tl})$ obtaining x^{λ} and $v(LCPMP^{Tl})$;

Obtain a feasible solution x^f and their value v_f using x^f (see section 4);

Update lb = max [lb, $v(LCPMP^{Tl})$];

Update ub = min [ub, v_f];

Set $g_i^l = 1 - \sum_{j \in M} x_{ij}^l$, i ∈ N;

Update the step size θ;

Set $\lambda_i = \max \{ 0, \lambda_i + \theta g_i^l \}$, i ∈ N;

Until (stopping tests).

In this algorithm, *T* is the approximately optimal value for *t* obtained by the procedure *SH* (with parameters *s* = 0.25, *kmax* = 10 and *tinit* = 1.0). The multiplier *T* is updated for each iteration of *SubG*. However, if the procedure *SH* produces the same multiplier *T* for 5 consecutive iterations of *SubG*, then the next Lagrangean/surrogate relaxations will use this fixed value *T* as the multiplier and the search *SH* is not performed.

The initial λ used is $l_i = \min_{j \in M} \{d_{ij}\}$, i ∈ N. The step sizes used are: $\theta = \pi \, (ub - lb) / \| g^\lambda \|^2$.

The control of parameter π is the Held and Karp [5] classical control. It makes $0 \le \pi \le 2$, beginning with π = 2 and halving π whenever lb does not increase for 30 successive iterations. The stopping tests used are:

a) $\pi \le 0.005$;

b) ub - lb < 1;

c) $\| g^\lambda \|^2 = 0$.

4. The Local Search Heuristics

The Lagrangean/surrogate approach described in section 3 is integrated with local search heuristics to make primal feasible a sequence of intermediate dual solutions. These heuristics will be described as follows.

Solution x^l in procedure *SubG* is not necessarily feasible to *CPMP*, but the set *J* identifies median nodes that can be used to produce feasible solutions. In order to allocate the non-median nodes to the identified set of medians we approximately solve the following generalized assignment problem:

$$\boldsymbol{Max} \sum_{i \in N} \sum_{j \in J} p_{ij} \, x_{ij}^f \tag{9}$$

(*GAP*) subject to:

$$\sum_{i \in N} q_i x_{ij}^f \leq Q_j \,, \; j \in J \tag{10}$$

$$\sum_{j \in J} x_{ij}^f = 1 \,, \; i \in N \tag{11}$$

$$x_{ij}^f \in \{0,1\} \,, \; i \in N \,; \, j \in J \tag{12}$$

where $p_{ij} = -d_{ij}$ ($i \in N$; $j \in J$) is the profit of node i if assigned to median j.

The algorithm *MTHG* proposed in [12] is used to provide approximated solutions x^f to *GAP*. Heuristic *MTHG* uses $f_{ij} = p_{ij}$ as a measure of the "desirability" of allocating item i to median j. It iteratively considers all the unallocated items, determining the item i' having the maximum difference between the largest and the second largest f_{ij} ($i \in N$); i' is then assigned to the knapsack for which $f_{i'j}$ is a maximum. If the solution obtained is feasible, in a second part of the algorithm the current solution is improved through local exchanges.

Solution x^f is further improved by an additional location-allocation heuristic (*LAH*) based on the observation that after the definition of x^f exactly p clusters can be identified, C_1, ..., C_p, corresponding to the p medians and their allocated non-medians. Solution x^f can be improved by searching for a new median inside each cluster, swapping the current median by a non-median and reallocating. If the set J changes we recalculate the x^f value on the new *GAP*, and if the new solution is better, we can repeat the reallocation process inside the new clusters, and the process until no improvements occur.

Specifically, in order to improve solutions x^f we have used the following heuristic:

Location-Allocation Heuristic (LAH)

For each cluster C_j (j = 1, ..., p) let $m_j \in C_j$ be the index of the median in the cluster C_j and $z_j = \sum_{k \in C_j} d_{km_j}$ be the cost of cluster C_j. Let M be the index set of possible medians and Q_i $(i \in M)$ be the capacity of each possible median.

Set nchanges = 0;

Repeat

  For (each cluster C_j, j = 1, ..., p) **do**:

    For (each non-median node $i \in C_j$ such that $i \in M$ and $Q_i \geq Q_{m_j}$) **do**:

      If (nchanges < Max_Changes) **then**

        Interchange i with m_j and update the cluster C_j calculating $z_i = \sum_{k \in C_j} d_{ki}$;

        If $z_i < z_j$ **then**

          nchanges = nchanges + 1;

          Update the set *J*, excluding m_j and including i;

Else

Interchange m_j with i, maintaining the cluster C_j and z_j unchanged;

End_If

End_If

End_For

End_For

Solve *GAP* considering the set J obtaining a new set of clusters $C_1, ..., C_p$;

While (it is possible to improve the feasible solution).

In addition, the following interchange heuristics are used, trying a further improvement to the feasible solution:

Interchange-Transfer Heuristic (ITH)

Let C_j and z_j be as in the algorithm LAH. Let $D_j = \sum_{k \in C_j} q_k$ be the total demand of the cluster C_j, j = 1, ..., p.

For (each cluster C_j, j = 1, ..., p) **do**:

For (each cluster C_i, i = 1, ..., p, i $\neq$ j) **do**:

For (each non-median nodes $k \in C_j$ and $l \in C_i$ such as:

$q_l \le Q_j - (D_j - q_k)$ and

$q_k \le Q_i - (D_i - q_l)$ and

$(z_j - d_{km_j} + d_{lm_j}) + (z_i - d_{lm_i} + d_{km_i}) < (z_j + z_i)$)

do interchange *k* with *l*.

End_For;

For (each non-median node $k \in C_i$ and a cluster C_j such as:

$q_k \le Q_j - D_j$ and

$(z_j + d_{km_j}) + (z_i - d_{km_i}) < (z_j + z_i)$)

do transfer k from cluster C_i to cluster C_j.

End_For;

End_For;

End_For.

The resulting algorithm that uses *SubG* (described in section 3) integrated with *LAH* and *ITH* will be called in the sequence of the paper as the *Lagrangean/Surrogate Local Search Heuristic (LSLSH)*, and can be described as:

Lagrangean/Surrogate Local Search Heuristic (LSLSH)

Given $\lambda \geq 0$, $\lambda \neq 0$;

Set lb = $-\infty$, ub = $+\infty$;

Repeat

Use the **algorithm SH** with parameters s = 0.25, kmax = 10 and tinit = 1.0, to find the best value T for t in the problem $(LCPMP^{tl})$;

Solve relaxation $(LCPMP^{Tl})$ obtaining x^{λ} and $v(LCPMP^{Tl})$;

Update lb = max [lb, $v(LCPMP^{Tl})$];

Obtain a feasible solution x^f and their value v_f using x^f;

Use the **algorithms LAH and ITH** while is possible to improve the feasible solution x^f;

Update ub = min [ub, v_f];

Set $g_i^{\lambda} = 1 - \sum_{j \in M} x_{ij}^{\lambda}$, $i \in N$;

Update the step size θ;

Set $\lambda_i = \max \{ 0, \lambda_i + \theta g_i^l \}$, $i \in N$;

Until (stopping tests).

5. Computational results

Two problem sets are used in computational tests: a classical set frequently used in others papers, and a set of real data collected at the central area of the São José dos Campos city. The first set was used before [15] and is formed by 2 sets of 10 instances, with (50×5) and (100×10) vertices and medians, respectively (available in the OR-Library - http://mscmga.ms.ic.ac.uk/info.html).

The algorithm *LSLSH* is coded in *C* and the tests executed on a *SUN Ultra30* machine. The *LAH* parameter *Max_Changes* is set to 3. *Table 1* reports the *LSLSH* application to these instances. The results are compared to the ones of two metaheuristics, the *HSS.OC* heuristic that presented the best performance among those reported on [15], and the *ATS* heuristic of [4].

Columns in *Table 1* are composed of: the problem identification, the best-known solution and the gaps (%) to the best solutions. Heuristic *HSS.OC* is a simulated annealing probabilistic acceptance approach that makes use of a non monotonic cooling schedule, a systematic neighborhood search, and a termination condition based on the number of temperature resets performed without improving the best solution. Heuristic *ATS* is an adaptive tabu search algorithm that systematically perturbs selected tabu elements, promoting intensification of the search when some indicators identify promising regions, and diversification if improvements seem to be minimal.

The last line in *Table 1* shows the average gaps for the instances. Results are very good and *LSLSH* seems to be better than the corresponding metaheuristics.

Table 2 reports the average running times for the heuristics. The times for *HSS.OC* were obtained with a VAX 8600, while the times for *ATS* were obtained on a SUN Sparc20.

Although the tests were performed on different machines, it can be conjectured that the *LSLSH* is faster than the other approaches, as it obtains a smaller number of feasible solutions.

The second set of problems is composed of real data collected using the Geographical Information System *ArcView* (ESRI [3]), and reporting the central area of São José dos Campos city. Six instances (100×10), (200×15), (300×25), (300×30), (402×30) and (402×40) are considered. Each point is located on a block, which presents a demand and is also a possible place to locate medians. The demand was estimated considering the number of houses (apartments) at each block. An empty block receives value 1. Capacities are then estimated as $C = \left\lceil \frac{\sum demands}{number\ \ of\ medians\ \times \alpha} \right\rceil$, where α is 0.9 or 0.8. These problems are available at http://www.lac.inpe.br/~lorena/instancias.html.

Table 3 presents the results. All the dual gaps are lower than 1% and results are obtained at reasonable computer times.

In order to isolate the effects of the local search heuristics *LAH* and *ITH* on heuristic *LSLSH*, the algorithm *SubG* was run without the local improvements in primal feasible solutions. The results are compiled in *Table 4*. The gaps increased up to 5.5% and also, as a side effect, the overall computational times (almost all the tests stopped at the $\pi \leq 0.005$ condition).

6. Conclusions

This work considers Lagrangean/surrogate local search heuristics for capacitated p-median problems. The Lagrangean/surrogate approach was able to generate as good approximate solutions as the obtained by metaheuristic approaches but employing small computational times.

The use of location-allocation followed by interchange heuristics has been proved to be useful for the primal feasibility of intermediate dual solutions. Heuristic *LSLSH* has been shown flexible and fast for large-scale real data obtained using Geographical Information Systems. These data present a spatially distributed set of points where location-allocation based heuristics perform best.

We conjecture that these ideas can be explored for larger-scale problems to produce high quality approximate solutions at reasonable computational cost.

Acknowledgments The authors acknowledge Fundação de Amparo a Pesquisa do Estado de S. Paulo - FAPESP (proc. 96/04585-6) and Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (proc. 350034/91-5 and 302408/88-6), for partial financial support. We greatly acknowledge the useful suggestions made in two revisions by the Area editor and a referee.

References

[1] J. Bramel, D. Simchi-Levi, *A location based heuristic for general routing problems*, Operations Research 43(1995) 649-660.

[2] L. Cooper, Location-allocation problems, Operations Research 11 (1963) 331-343.

[3] ESRI - Environmental Systems Research Institute, Inc. Avenue Customization and Application Development for ArcView, 1996.

[4] P.M. França, N.M. Sosa, V. Pureza, An adaptive tabu search algorithm for the capacitated p-median problem, International Transactions in Operations Research 6 (1999) 665-678.

[5] M. Held, R.M. Karp, The Traveling salesman problem and minimum spanning trees. Operations Research 18 (1970) 1138-1162.

[6] K. Klein, J.E. Aronson, Optimal clustering: a model and method, Naval Research Logistics 38 (1991) 447-461.

[7] Y.A. Koskosidis, W.R. Powell, Clustering algorithms for consolidation of costumes orders into vehicle shipments, Transportation Research B 26 (1992) 365-379.

[8] L.A.N. Lorena, F.B. Lopes, A surrogate heuristic for set covering problems, European Journal of Operational Research 79 (1994) 138-150.

[9] L.A.N. Lorena, M.G. Narciso, Relaxation heuristics for a generalized assignment problem, European Journal of Operational Research 91 (1996) 600-610.

[10] L.A.N. Lorena, E.L.F. Senne, Improving traditional subgradient scheme for Lagrangean relaxation: an application to location problems, International Journal of Mathematical Algorithms 1 (1999) 133-151.

[11] V. Maniezzo, A. Mingozzi, R. Baldaci, A bionomic approach to the capacitated p-median problem. Journal of Heuristics 4 (1998) 263-280.

[12] S. Martello, P. Toth, Knapsack problems: Algorithms and computer implementations, John Wiley & Sons, 1990.

[13] J.M. Mulvey, M.P. Beck, Solving capacitated clustering problems. European Journal of Operational Research 18 (1984) 339-348.

[14] M.G. Narciso, L.A.N. Lorena, Lagrangean/surrogate relaxation for generalized assignment problems. European Journal of Operational Research 114 (1999) 165-177.

[15] I.H. Osman, N. Christofides, Capacitated clustering problems by hybrid simulated annealing and tabu search. International Transactions in Operational Research 1 (1994) 317-336.

[16] E.L.F. Senne, L.A.N. Lorena, Lagrangean/Surrogate Heuristics for p-Median Problems. In Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research, M. Laguna and J.L. Gonzalez-Velarde (Eds.), Kluwer Academic Publishers, 2000, 115-130.

[17] H. Spath, Cluster Analysis Algorithms for data reduction and classification of objects, Ellis Horwood Publishers, 1980.

[18] E.D. Taillard, Heuristic methods for large centroid clustering problems, Technical report IDSIA96-96, IDSIA, 1996.

Table 1: Results for the first set of instances

Problem	**Vertices**	**Medians**	**Best known solution**	**HSS.OC gap (%)**	**ATS gap (%)**	**LSLSH gap (%)**
1	50	5	713	0	0	0
2	50	5	740	0	0	0
3	50	5	751	0	0	0
4	50	5	651	0	0	0
5	50	5	664	0	0	0
6	50	5	778	0	0	0
7	50	5	787	0	0	0
8	50	5	820	0	0	0
9	50	5	715	0	0	0
10	50	5	829	0	0	0
11	100	10	1006	0	0	0
12	100	10	966	0	0	0
13	100	10	1026	0	0	0
14	100	10	982	0.30	0.30	0
15	100	10	1091	0	0.27	0.09
16	100	10	954	0	0	0
17	100	10	1034	0.48	0	0
18	100	10	1043	0.19	0.19	0
19	100	10	1031	0	0.19	0
20	100	10	1005	0	0	0.39
Mean				0.049	0.047	0.024

Table 2: Average CPU times - comparison (seconds)

Medians	HSS.OC	ATS	LSLSH
50	23.23	13.89	3.81
100	338.19	304.67	37.1

Table 3: Results for São José dos Campos city set of instances

Problem	Size	Bound LSLSH dual	Bound LSLSH primal	Gap (%)	Time (sec.)
1	100 x 10	17252.12	17288.99	0.21	68.62
2	200 x 15	33223.66	33395.38	0.51	2083.37
3	300 x 25	45313.43	45364.30	0.11	2604.92
4	300 x 30	40634.91	40635.90	0.00	867.68
5	402 x 30	61842.49	62000.23	0.25	27717.11
6	402 x 40	52396.54	52641.79	0.46	4649.47

Table 4: Results without local search heuristics

Problem	Size	Bound SubG dual	Bound SubG primal	Gap (%)	Time (sec.)
1	100 x 10	17256.97	17288.99	0.19	55.09
2	200 x 15	33223.63	34130.32	2.66	2742.19
3	300 x 25	45315.43	45758.39	0.97	6692.47
4	300 x 30	40634.91	40635.91	0.00	2466.18
5	402 x 30	61842.65	63028.14	1.88	51887.37
6	402 x 40	52396.62	55474.61	5.55	6927.82