Solving large p-median problems using a Lagrangean heuristic

Andréa Cynthia Santos

LIMOS, Université Blaise Pascal, Campus des Cézeaux, 63173, Aubière, France,
`andrea@isima.fr`

Abstract. The *p*-median problem consists in locating *p* medians in a given graph, such that the total cost of assigning each demand to the closest median is minimized. In this work, a Lagrangean heuristic is proposed and it outperforms a classic heuristic based on the same Lagrangean relaxation. Variable fixing tests are used to reduce the size of the problems and a local search procedure is also applied. Variable fixing strategies eliminate 90% of arcs on average. Computational results are reported for large graph instances with about 4000 nodes and 14 millions arcs.
Keywords: Lagrangean heuristics, primal-dual heuristics, p-median problem.

1 Introduction

Let $G = (V, A)$ be a directed graph with a set V of vertices and a set A of arcs. Let $c_{ij} \in \Re$, $\forall(i, j) \in A$, be the cost of assigning a demand $j \in V$ to a median $i \in V$ (medians are also referred to facilities or locations in the literature). Given a positive integer $0 < p \leq |V|$, the p-median problem consists in selecting p medians in V such that the total cost of assigning each demand to the closest median is minimized. This problem belongs to the set of NP-hard problems on general graphs G for an arbitrary p (Garey and Johnson, 1979).

The interest of solving p-median problems appears as it is the core of many location problems and because of its large number of applications. For example, several location applications are found in (Tansel et al., 1983a) and cluster analysis are shown in (Hansen and Jaumard, 1997; Rao, 1971). The amount of work dedicated to this problem has significantly increased in the last decade (Reese, 2006), and there is a clear improvement in the state of the art for the problem. Advanced heuristics such as Lagrangean heuristics (Senne and Lorena, 2000) and metaheuristics (Resende and Werneck, 2004), and sophisticated exact algorithms such as the branch-and-price-and-cut (Avella et al., 2007), and the semi-lagrangean relaxation (Beltran et al., 2006) have been proposed. Improvements on the polyhedral characteristics for the p-median have been done, for example in (de Farias Jr., 2001). Moreover, efficient variable fixing strategies for a problem close to the p-median are presented in Briant and Naddef (2000). For further investigation, readers are referred to the following surveys: work (Reese, 2006) provides insightful information for more than 200 articles on the classic version of the p-median problem. An overview on heuristics and metaheuristics approaches can be found in (Captivo, 2007; Hansen and Mladenović, 2008). For the state of the art and theoretical analysis see, for example, (Tansel et al., 1983a,b).

In this work, a new Lagrangean heuristic is proposed on a classic Lagrangean relaxation, as presented in Narula et al. (1977). The typical heuristic for that Lagrangean relaxation does not perform well as mentioned in Briant and Naddef (2000) and it uses a single criterion to choose the medians. The heuristic proposed in this work uses two criteria to select the medians. The second criterion uses an information from the dual solution to build a primal solution. Some efficient reduction tests are applied and the proposed heuristic is coupled with a local search. The subgradient method of Held and Karp (1970) is used to compute Lagrangean multipliers. The algorithm is tested on the OR-library (Beasley, 1990) and on the TSP-Library (Reinelt,

1991) instances. Results show that the Lagrangean heuristic suggested here outperforms results obtained using a classic Lagrangean heuristic. The variable fixing tests eliminate 90% of arcs on average. After reductions, it only remains a core that contains the hardness of each instances. Instances for which optimality was not proved, duality gaps are lower than 2%. Some remaining gaps are closed by using the branch-and-cut tree of a commercial solver.

This paper is organized as follows: a mathematical formulation is presented in Section 2. A Lagrangean relaxation based on the referred formulation is described in Section 3. In Section 4, the Lagrangean heuristic is described in detail. Section 5 is dedicated to the reduction tests. Computational results are reported in Section 6 and some concluding remarks are given in Section 7.

2 Problem formulation

Binary variables x_{ii} are associated with every node $i \in V$ to identify if node i is chosen as a median. Variables x_{ij} are associated with every arc $(i, j) \in A$ to state whether node j is connected to the median i or not. Then, the p-median problem can be formulated as follows:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} \qquad st. \tag{1}$$

$$\sum_{i:(i,j)\in A} x_{ij} + x_{jj} = 1 \quad \forall j \in V \tag{2}$$

$$\sum_{i\in V} x_{ii} = p \tag{3}$$

$$x_{ij} \leq x_{ii} \quad \forall (i,j) \in A \tag{4}$$

$$x_{ii} \in \{0,1\} \quad \forall i \in V \tag{5}$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in A \tag{6}$$

The objective function (1) minimizes the total cost of assigning each node j to a median i. Equations (2) state that either a node j is connected to one median or j is a median. Constraint (3) ensures that p nodes are chosen as median. Inequalities (4) state that a node j can be connected to a node i only if i is a median. Variables x are defined in Constraints (5) and (6).

A feasible solution S for the p-median problem can be represented as a set of p clusters: a cluster q_i is a set of nodes with one median i $(x_{ii} = 1)$ and with nodes $\{j_1, j_2, ..., j_l\}$ $(l \leq |V| - p)$ connected to i by an arc, that is $(x_{ij} = 1)$. Thus, a solution $S = \{q_1, q_2, ..., q_p\}$, with no arcs between the clusters. Figure 1 illustrates a feasible solution S for a graph with $|V| = 25$, $p = 3$, and a total cost equal to 546.

3 Lagrangean relaxation

Different Lagrangean relaxations based on the formulation (1)-(6) can be found in the literature: Narula et al. (1977), Christophides and Beasley (1982), Hanjoul and Peeters (1985), and Mirchandani et al. (1985). In this work, the Lagrangean relaxation proposed by Narula et al.

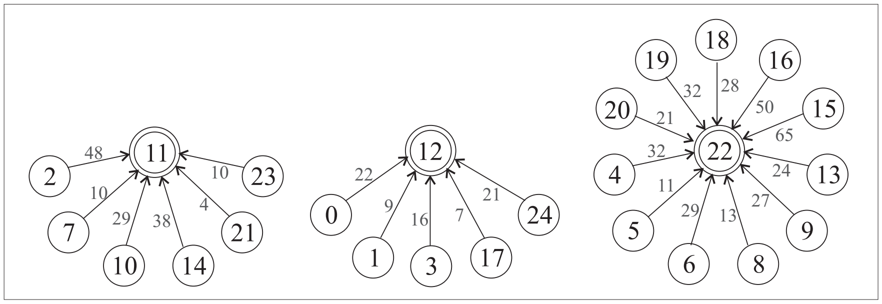

Figure 1: Example of a solution with three clusters.

(1977) is used since it produces very good lower bounds. Lagrange multipliers $\lambda_j \in \Re$ are associated to the Constraints (2), which are dualized. The corresponding Lagrangean dual problem is as follows:

$$L_1(\lambda) = \min \sum_{i \in V} \sum_{j \in V} (c_{ij} - \lambda_j) \cdot x_{ij} + \sum_{j \in V} \lambda_j \quad st. \tag{7}$$

Constraints (3) - (6).

Let $d_{ij} = c_{ij} - \lambda_j$ be the *Lagrangean costs*. For each node $i \in V$, its associated *auxiliary cost* is defined as $f_i = d_{ii} + \sum_{j \in V \setminus \{i\}} \min\{0, \ d_{ij}\}$. Thus, $L_1(\lambda)$ can be reformulated as:

$$L_1'(\lambda) = \min \sum_{i \in V} f_i x_{ii} + \sum_{j \in V} \lambda_j \quad st. \tag{8}$$

$$\sum_{i \in V} x_{ii} = p \tag{9}$$

$$x_{ii} \in \{0, 1\} \quad \forall i \in V \tag{10}$$

The Lagrangean subproblem (8)-(10) corresponds to a simple enumeration of the p facilities with the smallest costs f_i. This Lagrangean relaxation has the integrality property. Thus, the best lower bound attained using this Lagrangean relaxation is equal to the lower bound of the linear relaxation for the problem (1)-(6). The subgradient method of Held and Karp (1970) is used to compute Lagrangean multipliers λ_j. At each subgradient iteration k, multipliers λ_j^k are calculated as well as the *auxiliary costs* f_i^k. Then, since the p-median is a minimization problem, medians with the smallest *auxiliary cost* are selected for the dual solution. Moreover, each node j not chosen as median is connected to the median i ($x_{ij} = 1$), whenever $d_{ij}^k < 0$ and $x_{ii} = 1$. Arcs included in the solution have an associated negative *Lagrangean cost*. They have contributed in the computation of the *auxiliary costs* f_i^k, $\forall i \in V$, at iteration k.

4 Lagrangean heuristics

Lagrangean heuristics have been successfully applied to a large number of problems in the literature, see for example Andrade et al. (2006); Belloni and Lucena (2003). In particular, for the p-median problem and its variants, Lagrangean heuristics are presented in Beasley (1993a); Senne and Lorena (2000). A basic characteristic of these strategies is the use of dual Lagrangean information to produce primal solutions.

Let us assume that, at each subgradient iteration k, the vector f^k contains the *auxiliary costs* f_i in increasing order : $f^k = [\ f_1^k \leq f_2^k \leq ... \leq f_p^k \leq f_{p+1}^k \leq ... \leq f_{|V|}^k\ ]$. The p first nodes are the set of medians for the dual problem. Let UB^k and LB^k be respectively the current values of the upper and of the lower bounds, at iteration k. Moreover, UB^* and LB^* are respectively the best values of the upper and of the lower bounds found so far.

The classic primal heuristic for relaxation (8)-(10), denoted here of *Classic Lagrangean heuristic* (*Classic LH*), consists in generating a primal solution at each subgradient iteration k as follows: the p nodes with the smallest associated *auxiliary costs* in the vector f^k are taken as medians. The remaining $|V| - p$ nodes are connected to the closest median. This is a straightforward way to generate primal solutions based on Lagrangean relaxation (8)-(10). The *Classic LH* worst case complexity is $O(p \cdot |V|)$. Unfortunately, results are not very good. This fact has motivated the investigation of better ways to improve primal solutions using dual information.

In the Lagrangean dual problem $L_1'(\lambda)$, the dual solution for $L_1'(\lambda)$ may contains clusters connected by arcs. Figure 2 illustrates a dual solution with violations in an iteration of the subgradient method using $L_1'(\lambda)$. The dotted lines can be interpreted as an indecision about the connection of those nodes. Moreover, the second type of violation corresponds to the nodes $2, 7, 10, 11, 14, 21$ and 23 which are not medians, neither they are connected to a median. These kind of violations occurs quite often and it leads to the following questions: how does it impact the primal solutions? is it possible to take advantage of those violations to improve a primal heuristic such as the *Classic LH* ? These violations can be reduced once the subgradient method converges. However, earlier good upper bounds are met, better for the convergence of the subgradient method.

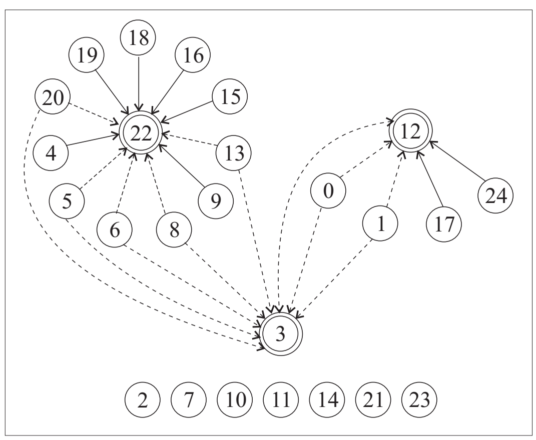

Figure 2: Example of violations in a dual solution.

To avoid the transfer of those indecisions (see Figure 2) from a dual solution to a primal solution, the heuristic proposed here, denoted *Collecting medians*, consists in taking the p first nodes with the smallest *auxiliary costs* as median, if there are no arcs between the medians in a dual solution. Thus, the *Collecting medians* considers the nodes in ascending order of f^k, and it selects a node as a median if it is not connected to the previous medians. After evaluating f^k until the position $|V|$, if less than p medians are selected, the procedure performs an additional step. It consists in considering the nodes with the smallest *auxiliary cost* from $p + 1$ to $|V|$, then from 1 to p as a median (this step is seldom performed). In doing so, feasibility is ensured. In the additional step, the test of connection with the previous medians is not performed. The $|V| - p$ nodes are connected to the closest median. For example, using the *Collecting medians* based on the dual solution shown in Figure 2, node 3 is not chosen as a median since it has *auxiliary cost* f_3 bigger than f_{22} and f_{12}, and it has a direct connection with the median 12. The first step to select the medians takes $O(p \cdot |V|)$ in the worst case. The additional step consumes up to $O(|V|)$ time in the worst case. Connecting the remaining nodes still remains $O(p \cdot |V|)$. Thus, the *Collecting medians* worst case complexity is $O(p \cdot |V|)$.

4.1 Local search procedure

Two local search procedures are used. The first is called optimizing clusters local search and it is used at each iteration of the subgradient method. The second one is the swap local search introduced by Teitz and Bart (1968). Efficients implementations for the swap local search is proposed by Whitaker (1983); Resende and Werneck (2003). The neighbourhood of the swap local search is bigger than the optimizing clusters local search, and it consumes a larger running time at each iteration. As a consequence, it is applied when a best upper bound is attained or after performing a number of iterations without improving the best upper bound found so far.

The optimizing cluster local search procedure is as follows: given a feasible solution S, for each cluster q_i, a local search move consists in building a new cluster q_i' having the same set of vertices in q_i, such that one of the vertices in this cluster $\{j_1, j_2, j_3, ..., j_l\}$, $(l \leq |V| - p)$, say j_k, will act as its new median. A move is accepted under two conditions: first, it reduces the cost of S and second, it is the best move in a cluster q_i. This step consumes $p \cdot (|V| - p)$ time in the worst case.

An iteration of this local search consists in investigating every cluster $q_i \in S$. After that, if at least one move is done in a cluster, an additional test is performed. Given a node j connected to the median i, the test checks if there is a median $l \neq i$ such that $c_{lj} \leq c_{ij}$. This test is done for all node j that are not median and that do not belong to the cluster q_i. This step requires $(|V| - p) \cdot p$ time in the worst case. Thus, an iteration of this local search consumes $p \cdot (|V| - p)$ time in the worst case. The local search procedure stops when no improving move is found. An example is shown in Figure 3. A cluster q_i is shown in Figure 3-(a) where i is its median. All possible neighbour clusters are depicted in Figures 3-(b) to 3-(d). If at least a move is done in one cluster, the additional step mentioned above is performed.

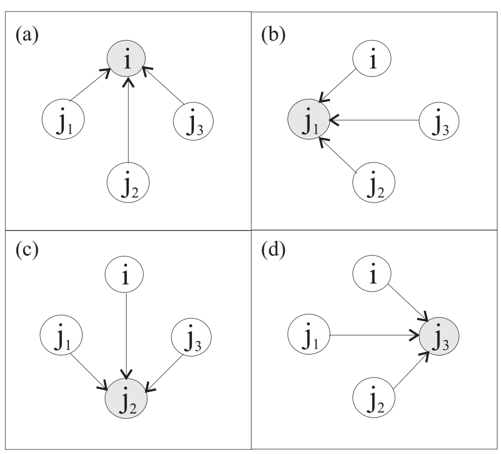

Figure 3: Example of the optimizing cluster local search move.

A typical local search for the p-median problem (Teitz and Bart, 1968) consists in finding the best swap between a median and a non-median node. Given a solution S and a candidate to be median j, the procedure computes the highest cost gain of introducing j as a median in S and the smallest cost loss of removing a current median i in S. Whenever, the total cost of S is improved and it corresponds to the best move in the neighbourhood, the move is performed. The procedure stops when no improving move is found. Whitaker (1983) and Resende and Werneck (2003) implementations for this local search consume at each iteration, $O(|V|^2)$ time in the worst case. The difference relies on updating the data structures which are speed up though the use of extra memory in (Resende and Werneck, 2003). Whitaker (1983) implementation is used in this work.

5 Problem reduction tests

Reduction tests allow variables to be fixed and thus the problem size is reduced. For the sake of clarity, fixing a variable associated to a node (x_{ii}) and to an arc (x_{ij}) is respectively referred to here as *node fixing* and as *arc fixing*. For the p-median problem, fixing a node to one implies that only $p-1$ medians need to be chosen afterward. Fixing a node to zero implies that the number of nodes able to be a median is reduced. Decisions can also be taken to fix arcs to zero or to one. The tests used here are adapted from Briant and Naddef (2000).

For every variable fixing test described in this section, the vector f^k contains the auxiliary costs f_i^k in increasing order as mentioned in Section 4. Moreover, let V_0 and V_1 be respectively the set of medians fixed to 0 and fixed to 1, as well as A_0 and A_1 represent respectively the set of arcs fixed to 0 and fixed to 1.

The main idea to fix arcs to zero is to check if: including an arc in the solution will lead to a lower bound greater than the best upper bound found so far. If so, the arc cannot belong to an optimal solution. The first test to fix an arc to zero is as follows: given a median i, a node j not median, and d_{ij}^k the *Lagrangean cost* of arc (i,j). If $(i,j) \notin A_0$, $d_{ij}^k > 0$ and Inequality (11) holds, arc (i,j) can be fixed to zero. Thus, $A_0 = A_0 \cup \{(i,j)\}$.

$$LB^k + d_{ij}^k > UB^* \tag{11}$$

The second test to fix an arc to zero is as follows: given two nodes j and l that are both not medians, thus, arc (j,l) is not in the current dual solution. Suppose that (j,l) is in an optimal solution. Then, j should be median and a median should be removed from the current dual solution. The dropped median is the one which contributes the least to the lower bound value. Its removal gives an under-estimation of the lower bound value with $p-1$ medians. If $(j,l) \notin A_0$, $d_{jl}^k > 0$ and Inequality (12) is satisfied, then $A_0 = A_0 \cup \{(j,l)\}$.

$$LB^k - f_p^k + f_j^k + d_{jl}^k > UB^* \tag{12}$$

The general idea to fix nodes is similar to the one to fix arcs: if forcing a node to be a median (it implies that one of the current medians is removed) or if removing a current median (it means a new node is selected as a median) from a dual solution leads to a lower bound greater than the best upper bound found so far, variables are fixed. Fixing nodes result in fixing arcs afterwards.

The first test used to fix a median to zero is as follows: given a node j that is not a median in the current dual solution, suppose that j is in an optimal solution. Forcing j in the solution implies removing one of the current medians. The selected dropped median is the one which contributes the least to the lower bound (f_p^k). If node $j \notin V_0$ and Inequality (13) is satisfied, j cannot be a median in an optimal solution and $V_0 = V_0 \cup \{j\}$. Consequently, if j cannot be a median in an optimal solution, j has no ingoing arcs. Thus, $A_0 = A_0 \cup \{(j,l)\}$ for all $(j,l) \in V$. This test starts with the node j with the highest *associated auxiliary* cost, that is $(f_{|V|}^k)$, and it is carried out until Inequality (13) does not hold anymore.

$$LB^k - f_p^k + f_j^k > UB^* \tag{13}$$

The second test is applied to fix medians to one. It consists in assuming a current median is not in an optimal solution. The median i with the smallest *auxiliary cost* is kept because it is the one which most contributes to the current lower bound value. By doing so, a node j not median with the smallest associated *auxiliary cost*, that is f_{p+1}^k, enters the solution. Whenever i is a median in the current dual solution, $i \notin V_1$ and Inequality (14) holds, node i is necessarily a median in an optimal solution. Thus, $V_1 = V_1 \cup \{i\}$. If i is necessarily a median in an optimal solution, then every outgoing arc of i can be fixed to zero. In this case, $A_0 = A_0 \cup \{(j,i)\}$ for all $(j,i) \in A$. The test starts with the median with the smallest *auxiliary cost* and it is applied until Inequality (14) does not hold anymore.

$$LB^k - f_i^k + f_{p+1}^k > UB^* \tag{14}$$

Another consequence of fixing a median i to one is: given an arc (i,j) that belongs to the current dual solution (that is $d_{ij} < 0$). If $(i,j) \notin A_1$ and Inequality (15) holds, this arc necessarily belongs to an optimal solution. Then, $A_1 = A_1 \cup \{(i,j)\}$. That means also node j is never a median in an optimal solution. Then, the consequences given above to fix a median to zero is applied to node j.

$$LB^k - d_{ij}^k > UB^* \tag{15}$$

Some simple logical implications are used, basically to try to fix variables x_{ii} to one. The aim is to take advantage of the large number of arcs eliminated from the problem. The first logical implication checks if a node is necessarily a median in an optimal solution: if $j \notin V_1$, and if every outgoing arc of j is in A_0, then $j \in V_1$.

The second logical implication is as follows: given a node $l \in V_0$ with only one outgoing arc, say (i,l), and with all other outgoing arcs already fixed to zero. Then, arc (i,l) can be fixed to one, $A_1 = A_1 \cup \{(i,l)\}$. Furthermore, in this case, i is necessarily a median in an optimal solution because l is assigned to node i. Thus, if $i \notin V$, $V_1 = V_1 \cup \{i\}$ and all consequences to fix a median to one are applied to node i.

6 Computational results

The experiments were carried out on a Core 2 Duo machine with 2.66 GHz of clock and 4 Gb of RAM memory. Cplex 11 is used to close the remaining gaps. The parameter *uppercut-off* is set to the UB^* found by the heuristic *Collecting medians*. All other parameters are left to default values. Results are reported for the OR-library (OR-LIB) and for the TSP-Library (TSP-LIB) instances. The OR-LIB Instances with a bold name are those for which Cplex has run out of memory. The TSP-LIB instances with a bold name are those classified as difficult in (Beltran et al., 2006). The Euclidean distances for the TSP-LIB instances are truncated as in (Avella et al., 2007; Beltran et al., 2006). The parameters of subgradient method are set as suggested in (Beasley, 1993b): 1000 subgradient iterations are run and the parameter π is divided by 2 each time 5% of the total number of iterations are attained.

In the first experiment, a comparison between the *Classic LH* and the *Collecting medians* is done. In this experiment, the variable fixing tests and the local search procedures are not used in the algorithm. The goal is to observe the real benefit of using the *Collecting medians* heuristic.

The *Classic LH* is carried out at each iteration of the subgradient method. The same strategy is used for the *Collecting medians*. Results for a subset of the OR-LIB instances are presented in Table 1. The remaining instances of this group have similar results. Each line corresponds to an instance. Columns (Instances), (p) and (O^*) indicate respectively the instance name, the amount of medians and the optimal values. The linear relaxation (LR) obtained using the Cplex is given. Moreover, for both the *Classic LH* and the *Collecting medians* heuristic, the lower bound (LB), the upper bound (UB), the relative gap (GAP) (computed as $(100 \cdot (\text{UB} - \text{LB})/\text{LB})$), and the time in seconds (time) are presented.

The *Collecting medians* performs better than the *Classic LH*, closing gaps or reducing them. Results obtained by the *Collecting medians* are particularly promising because neither local search or reduction tests are used in this experiment. Results for the complete version of the *Collecting medians* heuristic is given later in this paper.

For nine out of fifteen instances in this subset (values highlighted in bold on the column UB), the *Collecting medians* heuristic has met the optimal values. While, the *Classic LH* has found the optimal value for only one instance. For these nine instances, if there is a gap using the *Collecting medians* heuristic is due to the lower bound values. In terms of lower bounds, the Lagrangean relaxation attains the theoretical limit for almost all instances, except for the pmed27 and pmed32. For these instances in particularly, it seems the theoretical limit was attained independent of the upper bound quality. However, this is not the case for the TSP-LIB instances.

6.1 Calibrating the variable fixing phase

Most work for the p-median problem does not check when the reduction tests are more likely to be efficient. Thus, a computational experiment was carried out to decide when to start the reduction tests. Variable fixing tests were introduced in the Lagrangean relaxation at each iteration after running the *Collecting medians* heuristic. Then, for each instance, the first iteration where the reduction tests start working was kept. Results are summarized in Table 2 for a subset of the OR-LIB instances. The remaining instances have similar results. Columns (Iteration), (LB), (UB) and (GAP) contain respectively the iteration number, the lower bound, the upper bound, and the relative gap which corresponds to the first iteration where the reduction tests start to work.

Results show that the reduction tests start to work when the gaps fall below 5%. Considering all the instances, the reduction tests begin to work earlier when the number of medians is smaller.

6.2 Results for the complete version of the Collecting medians heuristic

The complete version of the *Collecting medians* heuristic includes the two local search procedures and the variables fixing tests. At each iteration k of the subgradient method, a primal solution S^k is computed using the *Collecting medians* heuristic. Then, S^k is submitted to the optimizing cluster local search procedure. The swap local search is run whenever an upper bound is better than the best upper bound found so far, or when 10 iterations is performed without improving the best upper bound known. Finally, the reduction tests are applied when the gap falls down 5%. 1000 iterations of the subgradient method are performed. After running 1000 iterations, if at least 10% of the current variables are fixed, data structures are cleaned and new subgradient iterations are authorized.

The TSP-LIB instances are used to illustrate the performance of the complete version of the

					Cplex	*Classic LH*				*Collecting medians*			
Instances	$\|V\|$	$\|A\|$	p	O^*	LR	LB	UB	GAP	time	LB	UB	GAP	time
pmed26	600	359,400	5	9917	9854	9854	10008	1.56%	13.00	9854	9924	0.71%	13.03
pmed27			10	8307	8302	8301	8310	0.11%	12.81	8301	**8307**	0.07%	12.91
pmed28			60	4498	4498	4498	4520	0.49%	13.17	4498	**4498**	0.00%	4.75
pmed29			120	3033	3033	3033	3155	4.02%	13.69	3033	3034	0.03%	14.00
pmed30			200	1989	1989	1989	2246	12.92%	14.42	1989	**1989**	0.00%	9.61
pmed31	700	489,300	5	10086	10026	10026	**10086**	0.60%	18.11	10026	**10086**	0.60%	18.34
pmed32			10	9297	9293	9291	9333	0.45%	17.95	9291	**9297**	0.06%	18.22
pmed33			70	4700	4700	4700	4790	1.91%	18.53	4700	**4700**	0.00%	7.66
pmed34			140	3013	3013	3013	3205	6.37%	19.42	3013	3014	0.03%	19.66
pmed35	800	639,200	5	10400	10302	10302	10401	0.96%	23.72	10302	10401	0.96%	23.78
pmed36			10	9934	9833	9833	10309	4.84%	23.61	9833	9972	1.41%	23.69
pmed37			80	5057	5057	5057	5146	1.76%	24.25	5057	**5057**	0.00%	10.03
pmed38	900	809,100	5	11060	10947	10947	11316	3.37%	31.00	10947	11070	1.12%	31.19
pmed39			10	9423	9364	9364	9546	1.94%	30.67	9364	**9423**	0.63%	30.80
pmed40			90	5128	5128	5128	5159	0.60%	31.92	5128	**5128**	0.00%	13.34

Table 1: Comparative results between the *Classic LH* and the *Collecting medians* for the OR-Library instances.

Instances	$\|V\|$	$\|A\|$	p	Iteration	LB	UB	GAP
pmed32	700	489,300	10	158	9226	9422	2.12%
pmed33			70	223	4694	4753	1.26%
pmed34			140	137	2985	3060	2.51%
pmed35	800	639,200	5	105	10160	10538	3.72%
pmed36			10	154	9764	10012	2.54%
pmed37			80	162	5042	5089	0.93%
pmed38	900	809,100	5	103	10723	11170	4.17%
pmed39			10	156	9311	9480	1.82%
pmed40			90	204	5119	5164	0.88%

Table 2: Calibration results for the variable fixing phase using the OR-LIB instances.

Collecting medians because they are difficult instances containing a million to about fourteen millions of arcs. Cplex solver is not able to treat the TSP-LIB instances because it run out of memory. Results for a complete version of the *Collecting medians* heuristic are presented in Tables 3 and 4. The follow instances were tested: rl1304 with 10, 100, 300 and 500 medians, fl1400 with 100, 200, 300, 400 and 500 medians, ul1432 with 20, 50, 100, 200, 300 and 400 medians, vm1748 with 10, 20, 50, 100, 300, 400 and 500 medians, d2103 with 10, 20, 50, 100, 200 and 300 medians, pcb3038 with 5, 100, 150, 200, 300, 400 and 500 medians, and fl3795 with 150, 200, 300, 400 and 500 medians. Results for a subset of the TSP-LIB instances is shown. The remaining instances have similar results. Table 3 contains the amount of variables fixed for each problem. The instance name (Instance), the number of nodes ($|V|$), and the number of arcs ($|A|$), and the amount of medians (p) are given. Columns ($|V_0|$), ($|V_1|$), ($|A_0|$) and ($|A_1|$) correspond respectively to the number of nodes fixed to zero, the number of nodes fixed to one, the number of arcs fixed to zero, and the number of arcs fixed to one. Moreover, the total percentage of fixed nodes (% fixed nodes) and of fixed arcs (% fixed arcs) are also given.

Most of the instances have a strong reduction of arcs fixed to zero with millions of arcs eliminated. Considering all the forty instances tested for the TSP-LIB, on average, 31.1% of nodes and 90.79% of arcs are fixed. For the TSP-LIB instances, in particularly, the variables fixing phase is an important tool to help proving optimality. After fixing several variables, it remains only a difficult core for each problem.

Table 4 contains the results obtained for the TSP-LIB instances. Columns (Instances), (p), and (O^*) correspond respectively to the instance name, the amount of medians, and the optimal values. For the *Collecting medians* heuristics, the lower bound (LB), the upper bound (UB), the relative gap (GAP) (computed as $(100 \cdot (\text{UB} - \text{LB})/\text{LB})$), and the time in seconds (time) round up are presented. Columns (final GAP) and (+time) give respectively the final gap and the extra time required to close the remaining gaps by using the Cplex solver. The symbol (-) means the solver has ran out of memory. Whenever there is a gap on the column (final GAP), it corresponds to the gap before the Cplex solver ran out of memory. That is the case of the instance fl1400 and $p = 500$ medians.

For the forty TSP-LIB instances tested, the heuristic *Collecting medians* produces gaps really small. It allows to prove optimality for 30 out of 40 instances efficiently, specially considering its size. For some instances, even with a really small gap, cplex does not prove optimality because it ran out of memory. This happens for 10 out of 40 instances tested. They are fl1400 with 300, 400 and 500 medians, ul1432 with 400 medians, d2103 with 50 et 100 medians, and

Instances	$\|V\|$	$\|A\|$	p	$\|V_0\|$	$\|V_1\|$	$\|A_0\|$	$\|A_1\|$	% fixed nodes	% fixed arcs
rl1304	1,304	1,699,112	100	719	1	1,689,634	11	55.21	99.44
			300	879	187	1,697,639	796	81.75	99.96
			500	565	224	1,697,709	615	60.51	99.95
fl1400	1,400	1,958,600	100	696	11	1,930,917	4	50.50	98.59
			200	0	11	1,804,505	0	0.79	92.13
			500	0	0	1,908,805	0	0.00	97.46
u1432	1,432	2,049,192	**50**	128	0	1,836,028	0	8.94	89.60
			100	595	1	2,031,969	0	41.62	99.16
			200	140	0	2,006,989	0	9.78	97.94
vm1748	1,748	3,053,756	10	1464	0	2,964,753	0	83.75	97.09
			20	1121	0	2,936,963	0	64.13	96.18
			50	1411	1	3,041,545	39	80.78	99.60
d2103	2,103	4,420,506	**100**	0	0	1,662,465	0	0.00	37.61
			200	0	1	4,205,165	0	0.05	95.13
			300	0	1	3,978,526	0	0.05	90.00
pcb3038	3,038	9,226,406	200	37	0	8,938,597	0	1.22	96.88
			400	1515	13	9,177,383	78	50.30	99.47
			500	215	1	9,004,981	1	7.11	97.60
fl3795	3,795	14,398,230	**200**	0	1	13,501,977	0	0.03	93.78
			300	24	5	14,062,089	0	0.76	97.67
			400	0	5	13,707,447	0	0.13	95.20

Table 3: Variables fixing results for the TSP-LIB instances.

fl3795 with 150, 200, 400 and 500 medians. However, some difficult instances were solved as fl3795 with 300 nodes. For those 10 difficult instances, its core is now available and it can be exploited in future work using, for example, a specialized branch-and-cut tree. Using the *Classic LH*, it is possible to prove optimality for only 10 out of 40 TSP-LIB instances because of the produced gaps. For example, the best, the worst and the average gaps for this subset of instances are respectively 0.11%, 5.30% and 1.95%. It leads also to a weaker variables fixing: about 10% of nodes and 50% of arcs.

7 Conclusions

A new Lagrangean heuristic is proposed in this work. The *Collecting medians* heuristic uses two dual information to build primal solutions: the auxiliary costs and the selection of medians which is composed independent clusters in a dual solution. By doing so, it somehow avoids the transfer of indecisions from a dual to a primal solution. Computational results show that the *Collecting medians* outperforms the *Classic LH*.

Additionally, variables fixing tests are used to reduce the size of the problems. In general, the amount of reductions is very significant. For the TSP-LIB instances, millions of arcs are eliminated (on average, 90% of arcs are eliminated). Moreover, it is also shown, the reduction tests start to work and to be more efficient when the gaps are within 5%. This is important because it avoids applying the reduction tests at every subgradient iteration, and consequently time is saved. Furthermore, two local search procedures are also used to improve the *Collecting medians* heuristic. Consequently, optimality is proved for difficult instances using the complete

			Collecting medians				Cplex	
Instances	p	O^*	LB	UB	GAP	time (s)	final GAP	+time (s)
rl1304	100	491,639	491,496	491,664	0.034	270	0.00	1.63
	300	177,326	177,318	177,326	0.005	423	0.00	0.01
	500	97,024	97,018	97,024	0.006	557	0.00	0.03
fl1400	100	15,962	15,960	15,970	0.063	323	0.00	3.55
	200	8,806	8,791	8,846	0.626	520	0.00	3,293.44
	500	3,764	3,756	3,766	0.266	433	0.15	13,195.05
u1432	**50**	362,072	361,597	362,427	0.230	378	0.00	2,601.75
	100	243,793	243,719	243,821	0.042	347	0.00	3.20
	200	159,887	159,838	159,934	0.060	370	0.00	11.69
vm1748	10	2,983,645	2,981,277	2,983,645	0.079	388	0.00	192.76
	20	1,899,680	1,897,839	1,899,680	0.097	505	0.00	124.72
	50	1,004,331	1,004,203	1,004,331	0.013	507	0.00	0.95
d2103	**100**	unknown	193,793	195,511	0.887	1364	-	-
	200	117,753	117,727	117,985	0.219	1361	0.00	78.14
	300	90,471	90,267	90,752	0.537	1393	0.00	443.95
pcb3038	200	237,399	237,276	237,540	0.111	2558	0.00	448.36
	400	156,276	156,267	156,281	0.009	2483	0.00	7.58
	500	134,798	134,774	134,817	0.032	2697	0.00	59.08
fl3795	**200**	53,928	53,866	54,041	0.325	4754	-	-
	300	39,586	39,570	39,611	0.104	5245	0.00	11,602.83
	400	31,354	31,331	31,452	0.386	6294	-	-

Table 4: Results for the complete version of the *Collecting medians* for the TSP-LIB instances.

version of the *Collecting medians* and Cplex solver.

A simple idea is used here to improve the lower and the upper bounds qualities. It reinforces the fact that it is well worth using the problem characteristics to improve primal and dual solutions. As a future work, the Lagrangean relaxation can be used as a base for a more sophisticated algorithm such as a branch-and-bound tree or a relax-and-cut algorithm (Escudero et al., 1994; Lucena, 2005). Another possibility is to use the *Collecting medians* heuristic as a preprocessing phase since it strongly reduces the size of the problems. Moreover, characteristics of the difficult core for the TSP-LIB instances could be also investigated.

References

Andrade, R., Lucena, A. P. and Maculan, N.: 2006, Using lagragian dual information to generate degree constrained spanning trees, *Discrete Applied Mathematics* **154**(5), 703–717.

Avella, P., Sassano, A. and Vasil'ev, I.: 2007, Computational study of large-scale p-median problems, *Mathematical Programming: Series A and B* **109**(1), 89–114.

Beasley, J.: 1993a, Lagrangean heuristics for location problems, *European Journal of Operational Research* **65**(3), 383–399.

Beasley, J. E.: 1990, OR-Library: Distributing test problems by electronic mail, *Journal of the Operational Research Society* **41**, 1069–1072.

Beasley, J. E.: 1993b, Lagrangean relaxation, *in* C. Reeves (ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Wiley, New York, pp. 243–303.

Belloni, A. and Lucena, A.: 2003, A lagrangian heuristic for the linear ordering problem, *in* M. Resende and J. P. de Sousa (eds), *Metaheuristics: Computer Decision-Making*, Kluwer Academic Publishers, pp. 123–151.

Beltran, C., Tadonki, C. and Vial, J.-P.: 2006, Solving the p-median problem with a semi-lagrangean relaxation, *Computational Optimization and Applications* **35**(2), 239 – 260.

Briant, O. and Naddef, D.: 2000, The optimal diversity management problem, *Operations Research* **52**(4), 515–526.

Captivo, M. E.: 2007, The p -median problem: A survey of metaheuristic approaches, *European Journal of Operations Research* **179**(3), 65–74.

Christophides, N. and Beasley, J. E.: 1982, A tree search algorithm for the p-median problem, *European Journal of Operations Research* **10**(2), 196–204.

de Farias Jr., I. R.: 2001, A family of facets for the uncapacited p median polytope, *Operations Research Letters* **28**, 161–167.

Escudero, L., Guignard, M. and Malik, K.: 1994, A Lagrangean relax-and-cut approach for the sequential ordering problem with precedence relationships, *Annals of Operations Research* **50**, 219–237.

Garey, M. and Johnson, D.: 1979, *Computers and intractability: A guide to the theory of NP-Completeness*, W.H. Freeman, New York.

Hanjoul, P. and Peeters, D.: 1985, A comparasion of two dual-based procedures for solving the p-median problem, *European Journal of Operational Research* **20**(3), 387–396.

Hansen, P. and Jaumard, B.: 1997, Cluster analysis and mathematical programming, *Mathematical programming* **79**(1–3), 191–215.

Hansen, P. and Mladenović, N.: 2008, Complement to a comparative analysis of heuristics for the p-median problem, *Source, Statistics and Computing* **18**(1), 41–46.

Held, M. and Karp, R. M.: 1970, The travelling-salesman problem and minimum spanning trees, *Operations Research* **18**, 1138–1162.

Lucena, A.: 2005, Non delay relax-and-cut algorithms, *Annals of Operations Research* **140**, 375–410.

Mirchandani, P. B., Oudjit, A. and Wong, R.: 1985, Multidimensional extensions and a nested dual approach for the m-median problem, *European Journal of Operations Research* **21**(1), 121–137.

Narula, C., Ogbu, U. I. and Samuelsson, H. M.: 1977, An algorithm for the p-median problem, *Operations Research* **25**(4), 709–713.

Rao, M. R.: 1971, Cluster analysis and mathematical programming, *Journal of the American Statical Association* **6**(335), 622–626.

Reese, J.: 2006, Solutions methods for the p-median problem: an annotated bibliography, *Networks* **48**(3), 125–142.

Reinelt, G.: 1991, TSPLIB $-$ A traveling salesman problem library, *ORSA Journal on Computing* **3**, 376–384.

Resende, M. G. C. and Werneck, R. F.: 2003, On the implementation of a swap-based local search procedure for the p-median problem, *Proceedings of the fifth workshop on algorithm engineering and experiments*, SIAM, Philadelphia, pp. 119–127.

Resende, M. G. C. and Werneck, R. F.: 2004, A hybrid heuristic for the p-median problem, *Journal of Heuristics* **10**(1), 59–88.

Senne, E. L. F. and Lorena, L. A. N.: 2000, Lagrangean/surrogate heuristics for p-medians problems, *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research* pp. 115–130.

Tansel, B. C., Francis, R. L. and Lowe, T. J.: 1983a, Location on networks: a survey part i: the p-center and p-median problems, *Management Science* **29**(4), 482–497.

Tansel, B. C., Francis, R. L. and Lowe, T. J.: 1983b, Location on networks: a survey part ii: exploiting tree network structure, *Management Science* **29**(4), 498–511.

Teitz, M. B. and Bart, P.: 1968, Heuristic methods for estimating the generalized vertex median of a weighted graph, *Operations Research* **16**(5), 955–961.

Whitaker, R.: 1983, A fast algorithm for the greedy interchange of large-scale clustering and median location problems, *Information systems and operations research (INFOR)* **21**, 95–108.