

A branch-and-price approach to p -median location problems

Edson L. F. Senne*

FAX: +55 (12) 3123-2868

elfsenne@feg.unesp.br

FEG/UNESP – São Paulo State University
Engineering School – Mathematics Department
12.500-000 Guaratinguetá, SP - Brazil

Luiz A. N. Lorena

lorena@lac.inpe.br

LAC/INPE – Brazilian Space Research Institute
PO Box 515
12.201-970 São José dos Campos, SP – Brazil

Marcos A. Pereira

marcos@lac.inpe.br

LAC/INPE – Brazilian Space Research Institute
PO Box 515
12.201-970 São José dos Campos, SP – Brazil

* *corresponding author*

Abstract

This paper describes a branch-and-price algorithm for the p -median location problem. The objective is to locate p facilities (medians) such as the sum of the distances from each demand point to its nearest facility is minimized. The traditional column generation process is compared with a stabilized approach that combines the column generation and Lagrangean/surrogate relaxation. The Lagrangean/surrogate multiplier modifies the reduced cost criterion, providing the selection of new productive columns at the search tree. Computational experiments are conducted considering especially difficult instances to the traditional column generation and also with some large-scale instances.

Keywords: p -median, column generation, Lagrangean/surrogate relaxation, branch-and-price.

1 Introduction

This paper describes a branch-and price algorithm for the p -median location problem. The search for p -median nodes on a network is a classical location problem. The objective is to locate p facilities (medians) such that the sum of the distances from each demand point to its nearest facility is minimized. The problem is well known to be NP-hard and several heuristics have been developed for p -median problems.

The combined use of Lagrangean/surrogate relaxation and subgradient optimization in a primal-dual viewpoint was found to be a good solution approach to the problem [19]. The Lagrangean/surrogate generalizes the standard Lagrangean relaxation using the local surrogate information of constraints relaxed in Lagrangean relaxation, in order to accelerate subgradient-like methods. A local search is conducted at some initial iteration of the subgradient algorithm, correcting wrong step sizes. The gain in computational times can be substantial for large-scale problems [17, 19].

Column generation is a powerful tool for solving large-scale linear programming problems. Such linear programming problems may arise when the columns in the problem are not known in advance and a complete enumeration of all columns is not an option, or the problem is rewritten using Dantzig-Wolfe decomposition (the columns correspond to all extreme points of a certain constraint set) [5]. Column generation is explored in several applications, such as the well-known cutting-stock problem, vehicle routing and crew scheduling [6, 7, 8, 12, 13, 22, 23, 24]. In classical implementations of column generation, the algorithm iterates between a restricted

master problem and a column generation subproblem. Solving the master problem yields a certain dual solution, which is used in the subproblem to determine whether there is any column that might be an incoming column.

In many cases a straightforward application of column generation may result in slow convergence [18]. Senne and Lorena [20] recently presented a stabilized algorithm to p -median problems (see also [15]). The Lagrangean/surrogate relaxation performs as an acceleration process to column generation, generating new productive sets of columns. The known stability problems of column generation are reduced, mainly for large-scale problems. Other attempts to stabilize the dual appeared before, like the Boxstep method [16], the Bundle methods and the Analytic Center Cutting Plane method [9]. Neame [18] describes these and other recent alternative methods to stabilize the dual [10].

The branch-and-price [1] method was initially proposed to solve large-scale combinatorial optimization problems. It is implemented as a search tree algorithm employing column generation at each search node. We find applications of this method in generalized assignment, crew scheduling [1] and capacitated p -median problem [3], among others.

In this work, some aspects, like the branching rule and the tree search, are readily determined considering the uncapacitated p -median problem as a clustering problem. The Lagrangean/surrogate multiplier modifies the reduced cost criterion, providing the selection of new productive columns at the root node and also at the search tree

nodes. Computational experiments are conducted comparing the Lagrangean/surrogate and Lagrangean relaxations in a branch-and-price approach, considering especially difficult instances to the traditional column generation and also with some large-scale instances.

The paper is organized as follows. Section 2 summarizes the stabilized column generation approach to p -median presented in [20]. Section 3 presents the relevant aspects considered in our branch-and-price implementation. Section 4 presents the algorithms and Section 5 gives computational results.

2 A stabilized column generation for p -median problems

The p -median problem considered in this paper can be formulated as the following binary integer-programming problem:

$$\begin{aligned}
 \text{(Pmed):} \quad & v(\text{Pmed}) = \text{Min} \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \\
 \text{subject to} \quad & \sum_{i=1}^n x_{ij} = 1 \quad j \in N \quad (1) \\
 & \sum_{i=1}^n x_{ii} = p \quad (2) \\
 & x_{ij} \leq x_{ii} \quad i, j \in N \quad (3) \\
 & x_{ij} \in \{0,1\} \quad i, j \in N \quad (4)
 \end{aligned}$$

where:

$[d_{ij}]_{n \times n}$ is a symmetric cost (distance) matrix, with $d_{ii} = 0, \forall i \in N$;

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

n is the number of nodes in the network, $N = \{1, \dots, n\}$;

$[x_{ij}]_{n \times n}$ is the allocation matrix, with $x_{ij} = 1$ if node j is allocated to median i ,

and $x_{ij} = 0$, otherwise; $x_{ii} = 1$ if node i is a median and $x_{ii} = 0$, otherwise.

Constraints (1) and (3) ensure that each node j is allocated to only one node i , which must be a median. Constraint (2) determines the exact number p of medians to be located, and (4) gives the integer conditions.

The problem (*Pmed*) is a classical formulation explored in many papers. Garfinkel et al. [11] and Swain [21] proposed a set partition formulation to (*Pmed*), given by:

$$\text{(SP-Pmed):} \quad v(SP - Pmed) = \text{Min} \sum_{k=1}^m c_k x_k$$

$$\text{subject to} \quad \sum_{k=1}^m A_k x_k = 1 \quad (5)$$

$$\sum_{k=1}^m x_k = p \quad (6)$$

$$x_k \in \{0,1\},$$

where:

$S = \{S_1, \dots, S_m\}$, is the set of all subsets of N ;

$A_k = [a_{ik}]_{n \times 1}$, with $a_{ik} = 1$, if $i \in S_k$, and $a_{ik} = 0$ otherwise;

$[x_k]_{m \times 1}$ indicates if subset S_k is in the solution ($x_k = 1$), or not ($x_k = 0$); and

$$c_k = \text{Min}_{j \in S_k} \left(\sum_{i \in S_k} d_{ij} \right).$$

In this formulation (set partitioning problem with cardinality constraint), each subset S_k corresponds to a column A_k in constraint set (5), representing a cluster in which the median node is decided when the cost c_k is calculated, and so the columns of (*SP-Pmed*) implicitly consider the constraint set (3) in (*Pmed*). Constraints (1) and (2) are conserved and respectively updated to (5) and (6).

The Dantzig-Wolfe decomposition process is applied to (*SP-Pmed*) by relaxing the integrality requirements ($x_k \in [0,1]$). A restricted master problem (*LP-Pmed*) is then defined, in the column generation context [1], by dropping columns in the formulation of (*SP-Pmed*).

Senne and Lorena [19] presented the Lagrangean/surrogate relaxation for the p -median problem. A general description of the Lagrangean/surrogate relaxation appeared in the work of Narciso and Lorena [17]. For a given $t \in R$ and $\mathbf{p} \in R^n$, the *Lagrangean/surrogate* relaxation of problem (*Pmed*) is given by:

$$(\mathbf{LS}_i\mathbf{Pmed}^p): v(\mathbf{LS}_i\mathbf{Pmed}^p) = \text{Min} \sum_{i=1}^n \sum_{j=1}^n (d_{ij} - t\mathbf{p}_j)x_{ij} + t \sum_{j=1}^n \mathbf{p}_j$$

subject to (2), (3) and (4)

Problem $(\mathbf{LS}_i\mathbf{Pmed}^p)$ is solved considering constraint (2) implicitly and decomposing the problem for index i , obtaining the following n subproblems:

$$\text{Min} \sum_{j=1}^n (d_{ij} - t\mathbf{p}_j)x_{ij}$$

subject to (3) and (4).

Each problem is easily solved letting $\mathbf{b}_i = \sum_{j=1}^n [\text{Min}\{0, d_{ij} - t\mathbf{p}_j\}]$, and choosing I

as the index set of the p smallest \mathbf{b}_i (here constraint (2) is implicitly considered).

Then, a solution x_{ij}^p to problem $(\mathbf{LS}_i\mathbf{Pmed}^p)$ is:

$$x_{ii}^p = \begin{cases} 1, & \text{if } i \in I \\ 0, & \text{otherwise} \end{cases}$$

and for all $i \neq j$,

$$x_{ij}^p = \begin{cases} 1, & \text{if } i \in I \text{ and } d_{ij} - t\mathbf{p}_j < 0 \\ 0, & \text{otherwise} \end{cases}$$

The Lagrangean/surrogate solution is given by $v(LS_tPmed^P) = \sum_{i=1}^n \mathbf{b}_i x_{ii} + t \sum_{j=1}^n \mathbf{p}_j$.

The usual Lagrangean relaxation results from (LS_tPmed^P) if t is set to 1. For a fixed multiplier \mathbf{p} , the best value for t can be found solving approximately a local Lagrangean dual $\underset{t \in R}{Max} v(LS_tPmed^P)$. A dichotomous search used to find an approximate value for t is presented in [19].

The Lagrangean/surrogate is integrated to the column generation process transferring the multipliers \mathbf{p}_j ($j = 1, \dots, n$) of the restricted master problem $(LP-Pmed)$ to the problem $\underset{t \in R}{Max} v(LS_tPmed^P)$. The median (and its allocated non-median nodes) with smallest contribution on $v[\underset{t \in R}{Max} v(LS_tPmed^P)]$ results to be the one selected to produce the incoming column on the subproblem:

$$\mathbf{(Sub}_t\mathbf{Pmed)}: v(\mathbf{Sub}_t - \mathbf{Pmed}) = \underset{i \in N}{Min} \left[\underset{a_{ij} \in \{0,1\}}{Min} \sum_{j=1}^n (d_{ij} - t\mathbf{p}_j) a_{ij} \right]$$

Let \mathbf{a} be the dual variable corresponding to constraint (6) and j^* be the vertex index reaching the overall minimum on $v(\mathbf{Sub}_t\mathbf{Pmed})$. The new sets S_j are $\{i : a_{ij} = 1 \text{ in } (\mathbf{Sub}_t\mathbf{Pmed})\}$ and the column $\begin{bmatrix} A_{j^*} \\ 1 \end{bmatrix}$ is added to $(LP-Pmed)$ if

$\left[\sum_{j=1}^n (d_{ij} - \mathbf{p}_{j^*}) a_{ij^*} \right] < |\mathbf{a}|$. In effect, all the corresponding columns $\left[\frac{A_j}{1} \right]$ satisfying the

expression:

$$\left[\underset{a_{ij} \in \{0,1\}}{\text{Min}} \sum_{j=1}^n (d_{ij} - \mathbf{p}_j) a_{ij} \right] < |\mathbf{a}|, \quad (7)$$

can be added to the pool of columns. For $t = 1$ (Lagrangian case) this is also known as multi-pricing in column generation context [1].

3 The branch-and-price steps

The branch-and-price is detailed in this section, examining separately, the root node, the branching rule, and the search tree and pruning conditions.

3.1 The root node

The following algorithm is used on the root node:

Algorithm CG(t)

- (i) Set an initial pool of columns to (*LP-Pmed*);
- (ii) Solve (*LP-Pmed*) using CPLEX [14] and return the duals prices \mathbf{p}_j ,
 $j = 1, \dots, n$ and \mathbf{a} ;

- (iii) Solve approximately (by a dichotomous search) a local Lagrangean/surrogate dual $\text{Max}_{t \in R} v(LS_t Pmed^P)$, returning the corresponding columns of $(Sub_t Pmed)$;
- (iv) Append to $(SP-Pmed)$ the columns $\begin{bmatrix} A_j \\ 1 \end{bmatrix}$ satisfying expression (7);
- (v) If no columns are found in step (iv) or $\left[v(SP - Pmed) - v(LS_t Pmed^P) \right] < 1$, then stop;
- (vi) Perform tests to remove columns and return to (ii).

Assuming $t = 1$, the algorithm $CG(1)$ gives the traditional column generation process. In this case, the dichotomous search in step (iii) is not necessary and the usual Lagrangean bound $(LS_1 Pmed^P)$ implicitly solves the $(Sub_1 Pmed)$ problem. In any case, the bounds $v(LP-Pmed)$ and $v(LS_t Pmed^P)$ are calculated at each iteration.

The following algorithm is used to set an initial pool of columns to $(LP-Pmed)$:

Algorithm IC

Let

Max_Cols be the maximum number of columns for the initial pool of columns.

$ncols = 0$;

While ($ncols < Max_Cols$) do

Let $M = \{ n_1, \dots, n_p \} \subset N$ be a randomly generated set of nodes.

For each $i = 1, \dots, p$ do

$$S_i = \{n_i\} \cup \{k \in N \setminus M : d_{kn_i} = \underset{j \in M}{\text{Min}}(d_{kj})\}$$

$$c_i = \underset{j \in S_i}{\text{Min}} \left(\sum_{k \in S_i} d_{jk} \right)$$

For $j = 1, \dots, n$ do

Set $a_{ij} = 1$ if $j \in S_i$

$a_{ij} = 0$, otherwise

End_for

Include column $\begin{bmatrix} A_j \\ 1 \end{bmatrix}$ in the initial pool of columns.

End_For;

$\text{ncols} = \text{ncols} + p$;

End_While;

The following algorithm is used in step (vi) of $CG(t)$:

Algorithm RC

Let

rc_mean be the average of the reduced costs for the initial pool of columns (after algorithm IC) of $(LP-Pmed)$

m be the total number of columns in the current $(LP-Pmed)$

rc_i be the reduced cost of the columns in the current $(LP-Pmed)$
 $(i = 1, \dots, m)$

For $i = 1, \dots, m$ do

Delete column i from the current ($LP-Pmed$) if $rc_i > rc_mean$.

End_For;

3.2 The branching rule

The branching rule considers the partitioning with identical subsets described in Wolsey [22]. Let q and r be the indices of rows presenting pairs of fractional columns like $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$ in the restricted master final solution. The pair (q, r) is identified as follows.

Assume that $|A|$ denotes the cardinality of a set A . Let $X = \{x_1, \dots, x_m\}$ be the set of fractional decision variables corresponding to the set $S = \{S_1, \dots, S_m\}$ of columns of ($LP-Pmed$). Let $QS(i) = \{S_j : i \in S_j, j = 1, \dots, m\}$ for each row index i ($i = 1, \dots, n$). Then, q is chosen as the row index such that $|QS(q)| > |QS(i)|, \forall i$ ($i = 1, \dots, n$). Note that, if $|QS(q)| = 1$, then X is a feasible solution of ($LP-Pmed$).

Let $RS(i) = \{S_j \in QS(q) : i \in S_j, j = 1, \dots, m\}$ for each row index i ($i = 1, \dots, n$). Let T be the set of row index i for which the set $RS(i)$ is non-empty, that is, $T = \{i : RS(i) \neq \emptyset, i = 1, \dots, n\}$. Then, r is chosen as the row index such that $|RS(r)| < |RS(i)|, \forall i \in T$.

Once determined the pair (q, r) of row indices, the following problem is solved on the left branch nodes:

$$v(\text{Sub}_tPmed) = \text{Min}_{j \in N} \left[\text{Min}_{a_{ij} \in \{0,1\}} \sum_{i=1}^n (d_{ij} - t\mathbf{p}_i) a_{ij} \right] \quad (8)$$

$$\text{subject to} \quad a_{qj} = a_{rj}, \quad j = 1, \dots, n.$$

And, on the right branch nodes, the following problem is solved:

$$v(\text{Sub}_tPmed) = \text{Min}_{j \in N} \left[\text{Min}_{a_{ij} \in \{0,1\}} \sum_{i=1}^n (d_{ij} - t\mathbf{p}_i) a_{ij} \right] \quad (9)$$

$$\text{subject to} \quad a_{qj} + a_{rj} \leq 1, \quad j = 1, \dots, n.$$

Observe that the integer binary problem at each node of the search tree is a combination of several problems such as (8) and (9), depending on the path from the root to the considered node.

3.3 The search tree and pruning conditions

The search tree is built in a depth first search and, at each node, the corresponding problems (8) and (9) are solved using CPLEX [14]. A tree node is pruned if the corresponding $v(LP\text{-}Pmed)$ or $v(LS_tPmed^P)$ are not less than the current best feasible solution.

4 The computational tests

The branch-and-price method described above was coded in C language and executed in a Pentium III 1.13 GHz microcomputer. We compared the computational results of the branch-and-price algorithms, when the variable t is fixed to 1, corresponding to the Lagrangean relaxation case, and for t calculated by dichotomous search, corresponding to the Lagrangean/surrogate relaxation. Table I shows the results for some problems obtained from OR-Library [2]. In this table:

n	is the number of nodes;
p	is the number of medians;
solution	is the calculated optimal solution value;
Num_Cols	is the maximum problem size, in terms of number of columns;
Tree size	is the number of nodes generated;
Time	is the total computational time (in seconds).

Table I – Results for OR-Library instances.

As can be observed from Table I, the branch-and-price approach using the Lagrangean/surrogate relaxation explores smaller trees to obtain new (fewer) columns to the restricted master problem, generally resulting in smaller computational times.

Christofides and Beasley [4] observed that the relation n/p determines the complexity of instances of p -median problems solved by the combined use of Lagrangean relaxation and subgradient optimization. In the column generation approach, the bigger the relation n/p , the harder the solution process. Table II presents the results of algorithms $CG(1)$ and $CG(t)$ for instances where $n/p \geq 10$.

Table II – Results for OR-Library harder instances.

The results confirm the superiority of Lagrangean/surrogate relaxation in the column generation process.

5 Conclusion

In this paper we present a branch-and-price method for p -median location problems. This method uses a column generation process that differs from the traditional because employs Lagrangean/surrogate relaxation. The Lagrangean/surrogate multiplier modifies the reduced cost criterion, providing the selection of more productive columns at each search tree node than the traditional column generation approach. The algorithm proposed is able to find the optimal solution of p -median problems exploring smaller search trees, and in shorter computational times.

Acknowledgments: The authors acknowledge Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (processes 302408/88-6, 300837/89-5 and 380758/02-4, respectively) for partial financial research support. The useful suggestions made by two anonymous referees are also greatly acknowledged.

References

1. Barnhart, C.; Johnson, E. L.; Nemhauser, G. L.; Savelsbergh, M. W. P. and Vance, P. H. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research* 1998; 46: 316-329.
2. Beasley, J. E. OR-Library: Distributing test problems by electronic mail. *Journal Operational Research Society* 1990; 41: 1069-1072.
3. Ceselli, A., Righini G. A branch-and-price algorithm for the capacitated p -median problem. Working paper 2002, University of Milano (<http://sansone.crema.unimi.it/~righini/Papers/CPMP.pdf>)
4. Christofides, N.; Beasley, J. E. A tree search algorithm for the p -median problems. *European Journal of Operational Research* 1982; 10: 196-204.
5. Dantzig, G. B. and Wolfe, P. Decomposition principle for linear programs. *Operations Research* 1960; 8: 101-111.
6. Day, P. R. and Ryan, D. M. Flight Attendant Rostering for Short-Haul Airline Operations. *Operations Research* 1997; 45 :649-661.

7. Desrochers, M. and Soumis, F. A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science* 1989; 23: 1-13.
8. Desrochers, M.; Desrosiers, J. and Solomon, M. A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research* 1992; 40: 342-354.
9. du Merle, O.; Goffin, J. L. and Vial, J. P. On Improvements to the Analytic Centre Cutting Plane Method. *Computational Optimization and Applications* 1998; 11: 37-52.
10. du Merle, O.; Villeneuve, D.; Desrosiers, J. and Hansen, P. Stabilized column generation. *Discrete Mathematics* 1999; 194: 229-237.
11. Garfinkel, R. S.; Neebe, W. and Rao, M. R. An Algorithm for the M-median Location Problem. *Transportation Science* 1974; 8: 217-236
12. Gilmore, P. C. and Gomory, R. E. A linear programming approach to the cutting stock problem. *Operations Research* 1961; 9: 849-859.
13. Gilmore, P. C. and Gomory, R. E. A linear programming approach to the cutting stock problem - Part II. *Operations Research* 1963; 11: 863-888.
14. ILOG Inc., Cplex Division. CPLEX 6.5, 1999.
15. Lorena, L. A. N.; Senne, E. L. F. A column generation approach to capacitated p-median problems. To appear in *Computers and Operations Research* 2003.
16. Marsten, R. M.; Hogan, W. and Blankenship, J. The Boxstep method for large-scale optimization. *Operations Research* 1975; 23: 389-405.
17. Narciso, M. G.; Lorena, L. A. N. Lagrangean/surrogate Relaxation for Generalized Assignment Problems. *European Journal of Operational Research* 1999; 114(1): 165-177.

18. Neame, P. J. Nonsmooth Dual Methods in Integer Programming. Phd Thesis - Department of Mathematics and Statistics, The University of Melbourne, March 1999.
19. Senne, E. L. F.; Lorena, L. A. N. 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. pp. 115-130.
20. Senne, E. L. F.; Lorena, L. A. N. Stabilizing column generation using Lagrangean/surrogate relaxation: an application to p -median location problems. Presented at EURO2001 Conference, Erasmus University, Rotterdam, July 2001 (<http://www.lac.inpe.br/~lorena/ejor/EURO2001.pdf>).
21. Swain, R. W. A Parametric Decomposition Approach for the Solution of Uncapacitated Location Problems. Management Science 1974; 21: 955-961.
22. Wolsey, L. A. Integer programming. New York: John Wiley & Sons, 1998.
23. Valério de Carvalho, J. M. Exact Solution of Bin-Packing Problems Using Column Generation and Branch-and-Bound, Universidade do Minho, Departamento Produção e Sistemas, Working Paper 1996.
24. Vance, P. H.; Barnhart, C.; Johnson, E. L. and Nemhauser, G. L. Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound. Computational Optimization and Applications 1994; 3: 111-130.

Vitae

Edson Luiz França Senne teaches computer science in the Mathematics Department at São Paulo State University, Campus of Guaratinguetá. He received the Doctorate in Applied Computation in 1987 from INPE – Brazilian Space Research Institute. His research interest includes combinatorial optimization applications, decision support systems and geographical information systems. He has published in European Journal of Operational Research, Journal of Intelligent Systems and International Journal of Mathematical Algorithms.

Luiz Antonio Nogueira Lorena is a senior researcher of the Applied Mathematics and Computation Laboratory (LAC) at INPE (Brazilian Space Research Institute), Brazil. He received a Dr. Degree in System Engineering and Computation (1985) from COPPE at Federal University of Rio de Janeiro, Brazil. His main research activities are on combinatorial optimization applications with particular emphasis in heuristics. He has published in European Journal of Operational Research, Journal of the Operational Research Society, Evolutionary Computation, Geoinformatica, IEEE TCAD, International Journal of Industrial Engineering, and International Journal of Mathematical Algorithms.

Marcos Antonio Pereira is a Doctorate student at the Applied Mathematics and Computation Laboratory (LAC) at INPE (Brazilian Space Research Institute), Brazil, under supervision of Dr. Luiz A. N. Lorena. He received a Master Degree in Applied Mathematics from State University of Campinas – UNICAMP (1993). His research

interest includes combinatorial optimization and geographical information systems.

He published in International Journal of Industrial Engineering.

Table I – Computational results for OR-Library instances

<i>n</i>	<i>p</i>	solution	Branch-and-price Lagrangean			Branch-and-price Lagrangean/surrogate		
			Num_Cols	Tree size	Time	Num_Cols	Tree size	Time
100	20	3034	2172	1	0,28	2124	1	0,31
100	33	1355	1614	1	0,11	1603	1	0,12
200	40	2734	3344	9	39,23	3442	7	67,61
200	67	1255	2046	3	2,81	2013	1	0,49
300	60	2968	4051	15	113,95	3231	5	42,31
300	100	1729	2352	1	1,39	2384	1	1,38
400	80	2845	3669	1	24,29	3167	1	9,83
400	133	1789	2798	1	1,30	2708	1	1,57
500	100	2961	23191	9	3394,57	14417	7	1288,72
500	167	1828	3237	3	59,20	3620	9	223,94
600	120	3033	28357	25	9064,46	6453	5	304,56
600	200	1989	3499	7	204,45	3532	5	241,11
700	140	3013	48579	3	22688,06	7860	1	170,54
700	233	1847	3950	5	91,52	3966	5	159,10
800	267	2026	4966	21	787,65	3756	1	18,75
900	300	2106	3900	5	233,55	4643	15	1390,87
Average			8857,8	6,9	2294,2	4307,4	4,1	245,1

The columns contains:

- n* is the number of nodes;
- p* is the number of medians;
- solution is the calculated optimal solution value;
- Num_Cols is the maximum problem size, in terms of number of columns;
- Tree size is the number of nodes generated;
- Time is the total computational time (in seconds).

Table II – Results for OR-Library harder instances

<i>n</i>	<i>p</i>	solution	Branch-and-price Lagrangean			Branch-and-price Lagrangean/surrogate		
			Num_Cols	Tree size	Time	Num_Cols	Tree size	Time
100	5	5819	3230	1	12,98	3444	1	13,61
100	10	4093	5510	9	52,57	3830	7	42,80
100	10	4250	3626	31	78,65	3298	23	84,30
200	20	4445	7657	1	123,56	7409	1	94,75
300	30	4374	6686	1	458,56	8430	3	601,73
400	40	4809	54830	39	21694,66	14297	19	1742,25
500	50	4619	40243	3	19550,80	13911	3	1921,23
Average			17397,4	12,1	5996,0	7802,7	8,1	643,0

The columns contains:

- n* is the number of nodes;
- p* is the number of medians;
- solution is the calculated optimal solution value;
- Num_Cols is the maximum problem size, in terms of number of columns;
- Tree size is the number of nodes generated;
- Time is the total computational time (in seconds).