Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 36 (2010) 455–462
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

A computational study for the p-median Problem

Sourour Elloumi [1]

CEDRIC ENSIIE
Evry, France

Agnès Plateau [2]

CEDRIC CNAM
Paris, France

Abstract

Given a set of clients and a set of potential sites for facilities, the p-median problem consists of opening a set of p sites and assigning each client to the closest open facility to it. In [3], a new formulation of this problem was proposed that takes benefit from identical values in the distance matrix. This formulation, when directly used in a mixed integer linear programming software, was proved to perform better than other known formulations, on a large number of instances. Here, we propose to improve the performances of the new formulation by taking benefit from its structure in the solution of its LP-relaxation. Rows and columns are gradually added to the linear program until a condition on the optimal values of the variables is reached. A computational comparison is carried out on many classes of instances.

Keywords: Facility Location, Integer Programming, Column and Row generation

[1] Email: sourour.elloumi@ensiie.fr
[2] Email: agnes.plateau_alfandari@cnam.fr

1571-0653/$ – see front matter
doi:10.1016/j.endm.2010.05.058

1 Introduction

Let N be the number of clients, called $C_1, C_2, ..., C_N$, let M be the number of potential sites or facilities, called $F_1, F_2, ..., F_M$, and let d_{ij} be the distance from C_i to F_j. The p-median problem consists of opening p facilities and assigning each client to its closest open facility, in order to minimize the total distance. Many applications of this problem arise in different sectors; see [2], [4] and [5] for some recent applications. A recent review on solution methods can be found in [6].

In order to solve the p-median problem, it is usually formulated as a mixed-integer linear programming model. While a classical formulation was known in the literature and extensively used, a new formulation was recently introduced in [3]. In this paper, we implement an algorithm that is based on the structure of this new formulation in order to improve the running time of its solution.

The remainder of this paper is organized as follows. In Section 2, we recall the new formulation NF. In Section 3, we present the generic procedure that will allow to speed up the resolution of the LP relaxation of NF. This procedure is based on a column-and-row generation strategy. Finally, Section 4 sums up our numerical results.

2 The new formulation

For any client C_i, let K_i be the number of different distances from C_i to any facility. It follows that $K_i \leq M$. Let $D_i^1 < D_i^2 < ... < D_i^{K_i}$ be these distances, sorted. For each client C_i, let us define a hierarchy of neighborhoods in the following manner: For k in $1..K_i$, denote by V_i^k the k^{th} neighborhood of C_i, i.e. the set of facility sites located within distance D_i^k from i, or, more formally, $V_i^k = \{F_j : d_{ij} \leq D_i^k\}$. In this definition, V_i^1 is the set of sites located at distance D_i^1 from C_i, and $V_i^{K_i}$ is the set of all facility sites. The main idea of the new formulation is that, in an optimal solution to the p-median problem, a client C_i is assigned to its smallest neighborhood containing an open facility.

The new formulation NF uses the binary variables y_j for any facility F_j, where $y_j = 1$ if and only if facility F_j is open and the binary variables z_i^k for any client C_i and k in $1...K_i$, with $z_i^k = 1$ if and only if all the sites in V_i^k are closed.

In any feasible solution (z, y) and for any client C_i, the z_i^k values constitute a non increasing geometrical progression which first member z_i^1 is 0 if client C_i is assigned to a facility within V_i^1 and is 1 otherwise. The last member of that geometrical progression is equal to 0 since the $V_i^{K_i}$ neighborhood is precisely

the set of all facilities. The new formulation NF is:

$$\min \quad nf(z,y) = \sum_{i=1}^{N}\left(D_i^1 + \sum_{k=1}^{K_i-1}(D_i^{k+1}-D_i^k)z_i^k\right)$$

$$\text{s.t.} \quad \sum_{j=1}^{M} y_j = p \tag{1}$$

$$z_i^1 + \sum_{j:d_{ij}=D_i^1} y_j \geq 1 \quad i=1,...,N \tag{2}$$

$$z_i^k + \sum_{j:d_{ij}=D_i^k} y_j \geq z_i^{k-1} \quad i=1,...,N; k=2,...,K_i \tag{3}$$

$$z_i^{K_i} = 0 \quad i=1,...,N \tag{4}$$

$$z_i^k \geq 0 \quad i=1,...,N; k=1,...,K_i \tag{5}$$

$$y_j \in \{0,1\} \quad j=1,...,M \tag{6}$$

3 Improving the resolution of NF

Let $\underline{NF}$ be the LP-relaxation of NF. This section presents a generic column-and-row procedure which takes benefit from the structure of $\underline{NF}$ to speed up its resolution. Then, we propose to use this generic algorithm to solve $\underline{NF}$ at each node of a branch and bound scheme. Let H be an N-vector such that $1 \leq H_i \leq K_i$ and let P_H be the following linear program:

$$\min \sum_{i=1}^{N}\left(D_i^1 + \sum_{k=1}^{H_i-1}(D_i^{k+1}-D_i^k)z_i^k\right)$$

$$\text{s.t.} \quad (1),(2)$$

$$z_i^k + \sum_{j:d_{ij}=D_i^k} y_j \geq z_i^{k-1} \quad i=1,...,N; k=2,...,H_i$$

$$z_i^k \geq 0 \quad i=1,...,N; k=1,...,H_i$$

$$y_j \in [0,1] \quad j=1,...,M$$

Problem P_H is obtained from $\underline{NF}$ by eliminating columns z_i^k for $i=1,...,N$ and $k=H_i+1,...,K_i$, constraints (4), and constraints (3) for $i=1,...,N$ and $k=H_i+1,...,K_i$. We define μ as the operator that takes a feasible solution (z,y) for $\underline{NF}$ and eliminates from z as many coordinates as necessary to fit the dimension of the solutions of P_H. Hence, by construction, if (z,y) is a feasible solution to $\underline{NF}$, $(y,\mu(z))$ is a feasible solution for P_H, with an equal

Choose initial values for the N-vector H such that $1 \leq H_i \leq K_i$

Repeat

Compute an optimal solution $(\tilde{z}, \tilde{y})$ to problem P_H ;

For $i = 1, \ldots, N$

If $\tilde{z}_i^{H_i} > \epsilon$ **then** $H_i \leftarrow H_i + 1$ **EndIf**

EndFor

Until : for all $i \quad \tilde{z}_i^{H_i} < \epsilon$

Return$(\nu(\tilde{z}), \tilde{y})$ { an optimal solution to $\underline{NF}$ }

Fig. 1. Generic algorithm based on column-and-row generation

to or smaller objective value: P_H is a relaxation for $\underline{NF}$. We define ν as the opposite operator to μ. It takes a feasible solution (z, y) for P_H and adds to z as many zeros as necessary to fit the dimension of the solutions of $\underline{NF}$. This time, if (z, y) is a feasible solution for P_H, $(y, \nu(z))$ is not always a feasible solution for $\underline{NF}$.

Proposition 3.1 *(see [3]) Let H be an N-vector such that $1 \leq H_i \leq K_i$, and let $(\tilde{z}, \tilde{y})$ be an optimal solution to P_H. If $\tilde{z}_1^{H_1} = \tilde{z}_2^{H_2} = ... = \tilde{z}_N^{H_N} = 0$ then $(\tilde{y}, \nu(\tilde{z}))$ is an optimal solution to $\underline{NF}$ and the two problems $\underline{NF}$ and P_H have the same optimal values.*

An algorithm for solving $\underline{NF}$ at the root of a branch-and-bound procedure can be deduced from this proposition. This algorithm consists in solving a series of problems P_H, the last P_H having the same optimal value as $\underline{NF}$. It is described in Figure 1.

For any i in $\{1, \ldots, N\}$, increasing H_i at an iteration of the algorithm of Figure 1 amounts to modifying P_H as follows (i) add the column $z_i^{H_i+1}$, (ii) add one row corresponding to constraint $z_i^{H_i+1} + \sum_{j:d_{ij}=D_i^k} y_j \geq z_i^{H_i}$.

This iterative algorithm can be applied at each node of a branch-and-bound procedure. The last P_H model built at the root of the search tree is then continuously enriched with new columns and rows until an optimal integer solution is reached. At the end of the branch and bound procedure, the obtained problem P_H contains a number of variables and constraints, far inferior to the original problem $\underline{NF}$. This is observed in our experiments on several instances, as detailed in the next section.

4 Computational results

We test our algorithm on three classes of instances: *ORLIB*, *GR* and *RW*. In all of them, $N = M$. Instances of class ORLIB (*PMED* and *SL* instances) were introduced by Beasley [1]. In the most difficult ones, N varies from 600 to 900 and p from 5 to 300. In the *GR* class proposed in [8] N equals 100 or 150 and p varies from 5 to 50. The third class, originally proposed by [7], is *RW*. We consider four different values of N: 100, 250, 500 and 1000 ; and several values of p.

All the computations reported in this section have been carried out on a DELL LATITUDE D610 Personal Computer with a Pentium M-1.73 Ghz processor and 512 Mo RAM. The code has been written in C and the solver is Cplex 10.0.

We compare two solution methods for the NF formulation of the p-median problem. In the first method, the NF model is directly solved by use of the Cplex MILP solver with all default parameters. In the second method, a P_H model is solved at each node of the Cplex branch-and-bound algorithm by our generic column-and-row generation as described in the previous section. In this second method, our generic algorithm has been coded in a user-written callback function and all Cplex cuts options have been turned off in order to guarantee the particular structure of our model.

Furthermore, to improve the performance of Cplex solver by stronger pruning, we compute a high quality upper bound via a hybrid heuristic combining GRASP and path relinking, called POPSTAR, reported as one of the most effective methods for the p-median. The code is publicly available by its authors [7][9].

Tables 1-2 report on computational results for the *RW*, *ORLIB* and *GR* instances described above. The following notation is used: *Name* is the problem name, N is the number of clients (and sites), p is the number of sites to be opened, OPT is the optimal integer value, LB_{LP} is the LP-bound, $T(s)LP$ (resp. $T(s)0-1$) represent computation time in seconds for LP-solving (resp. 0-1 solving), columns *#nodes* contain the number of nodes developed in the branch-and-bound algorithm and finally, columns *% cuts* show the percentage of constraints added by the generic column-and-row generation compared to the total number of constraints of the original NF model.

The computational results lead to three major observations underlining the significant interest of using our generic column-and-row generation. First of all, on average, computation times are divided by 4 for *PMED* instances, by 8 for *SL* instances, by 19 for *GR* instances in comparison with the classical

approach. For the *RW* instances solved to optimality, computation times are divided by 8 for rw100, by 33 for rw250, by 4 for rw500 and by 2.6 for rw1000. Secondly, the size of the last P_H problem obtained at the end of the branch and bound varies from 2% (for *RW* instances) to 22% (for *PMED* instances) of the original model's size. This second remark emphasizes the fact that a few number of constraints allow to quickly reach an extreme point of the integer convex hull. Finally, our generic algorithm allows to provide LP-bound for most of rw1000 instances whereas the classical method failed.

			New formulation		Column-and-row Generation			
name	$N = M$	P	T(s) LP	T(s) 0-1	T(s) LP	% cuts	T(s) 0-1	% cuts
rw100	100	10	1,12	116,10	0,31	6,24	14,80	12,65
	100	20	0,10	3,50	0,13	1,36	0,30	1,93
	100	30	0,05	0,12	0,08	0,23	0,10	0,34
	100	40	0,03	0,10	0,04	0,02	0,05	0,08
	100	50	0,02	0,09	0,04	0	0,04	0,00
average			**0,26**	**23,98**	**0,12**	**1,57**	**3,06**	**3**
rw250	250	10	64,72	-	10,02	7,9	-	27,66
	250	25	23,36	-	4,07	2,33	-	8,18
	250	50	8,12	1303,46	2,26	0,66	37,20	1,17
	250	75	1,02	4,00	1,00	0,08	1,50	0,10
	250	100	0,49	2,50	0,28	0,01	0,30	0,02
	250	125	0,17	2,10	0,24	0	0,20	0,00
average			**6,63**	**328,02**	**1,57**	**0,62**	**9,80**	**1,89**
rw500	500	10	3014,00	-	209,00	8,6	-	13,02
	500	25	2954,74	-	68,80	2,89	-	8,42
	500	50	676,54	-	37,50	1,1	-	3,50
	500	75	168,65	-	28,30	0,52	-	1,56
	500	100	110,84	-	24,30	0,29	-	0,63
	500	150	18,04	23,79	2,90	0,03	4,80	0,03
	500	200	2,00	5,60	1,59	0	1,80	0,00
	500	250	0,70	4,20	1,52	0	1,60	0,00
average			**60,05**	**11,20**	**11,72**	**0,17**	**2,73**	**0,44**
rw1000	1000	5	*	-	*	12,2	-	
	1000	10	*	-	*	8,65	-	
	1000	20	*	-	1767,43	4,21	-	
	1000	25	*	-	1412,2	3,28	-	
	1000	50	*	-	707,5	1,44	-	
	1000	75	*	-	827,39	0,86	-	
	1000	100	*	-	721,96	0,57	-	
	1000	150	*	-	579,92	0,28	-	
	1000	200	*	-	309,51	0,15	-	
	1000	300	135,68	421,00	103,09	0,02	225,80	0,02
	1000	400	32,55	124,20	11,1	0,00	12,70	0,00
	1000	500	12,00	107,20	10,3	0	10,30	0,00
average				**217.47**		**2,64**	**82.93**	

Table 1

Computational results for the *RW* instances

- : The MIP resolution was stopped after 1 hour

* : The LP was not solved after 1 hour

					New formulation			Column-and-row Generation			
Name	$N = M$	P	OPT	LB_{LP}	T(s) LP	T(s) 0-1	# nodes	T(s) LP	T(s) 0-1	# nodes	% cuts
pmed26	600	5	9917	9854	68,57	257,3	24	24,9	113,2	20	29,5
pmed27	600	10	8307	8302	59,03	206,9	7	19,1	56,3	4	23,19
pmed28	600	60	4498	4498*	20,51	51,9	0	9,4	21	0	12,51
pmed29	600	120	3033	3033*	13	44,7	0	5,4	13,4	0	8,81
pmed30	600	200	1989	1989*	10,48	42,1	0	3,3	11,5	0	6,95
pmed31	700	5	10086	10026	77,95	259,1	17	26,1	113,5	13	33,3
pmed32	700	10	9297	9293	84,8	260,2	0	25,4	66,8	2	26,93
pmed33	700	70	4700	4700*	26,42	74	0	13,2	33,5	0	13,42
pmed34	700	140	3013	3013*	18,16	65,9	0	5,1	16,4	0	9,34
pmed35	800	5	10400	10302	93,41	219,66	0	33,4	169,4	19	32,24
pmed36	800	10	9934	9834	110,78	1971,1	256	36,5	848	331	30,16
pmed37	800	80	5057	5057*	61,32	130,4	0	16,4	36,6	0	13,05
pmed38	900	5	11060	10948	123,98	752,3	37	43,2	326,4	63	33,86
		10	9431	9362	121,9	1891,9	107	36,7	308,3	68	28,81
		20	7839	7832	92,64	460,5	14	29,3	112,1	17	22,61
		50	5892	5889	373,88	1082,7	8	22,9	90,2	13	16,82
		100	4450	4450*	35,24	128,9	0	15,4	39,7	0	13,07
pmed39	900	5	11069	10938	107,5	864,59	60	45,5	253,8	30	33,6
		10	9423	9365	92,98	1371,2	78	34,3	239	51	29,11
		20	7894	7894	286,78	381	0	25,5	70,6	0	23,36
		50	5941	5937	213,24	1706,8	28	23,1	94,1	19	18,22
		100	4462	4461	43,75	150	0	16,5	49,8	0	13,67
pmed40	900	5	12305	12246	123,01	448,2	6	52,8	186,6	8	37,35
		10	10491	10439	114,93	895,5	18	42,8	189,7	19	31,87
		20	8717	8711	134,52	550,3	8	35,7	105,2	8	25,78
		50	6518	6505	317,64	1279,7	101	23,1	232,9	99	18,8
		90	5128	5128*	47,2	142,2	0	16,7	40,2	0	13,64
		100	4878	4878*	44,74	139,7	0	17,2	43,6	0	13,22
average					**104,23**	**565,31**		**24,96**	**138,64**		**21,9**
sl700	700	233	1847	1847*	3,23	12,8	0	2,3	1,7	0	7,66
sl800	800	267	2026	2026*	4,35	18,1	0	2,7	2,2	0	7,21
sl900	900	300	2106	2106*	7	26	0	3	3,2	0	8,74
average					**4,86**	**18,97**		**2,67**	**2,37**		**7,87**
gr100	100	5	5703	5686	0,54	2,3	0	0,4	0,3	0	20,76
		10	4426	4263	0,26	38	343	0,7	3,5	205	13,33
		15	3893	3859	0,16	13,3	221	0,4	1,8	201	7,33
		20	3565	3563	0,05	1,3	0	0,1	0,2	8	2,21
		25	3291	3290	0,07	1,5	0	0,07	0,1	1	1,52
		30	3032	3031	0,05	1,7	0	0,07	0,1	1	1,04
		40	2542	2542*	0,03	0,1	0	0,08	0,1	0	0,71
		50	2083	2083*	0,02	0,1	0	0	0	0	0,25
gr150	150	5	10839	10688	0,64	7,3	25	1,9	2,3	16	16,83
		10	8729	8456	1,51	95,8	398	1,8	10,9	317	14,26
		15	7390	7031	1,52	2328,01	7345	1,7	118,5	5851	12,84
		20	6454	6265	1,03	1288,51	6507	1,5	70,7	3838	8,55
		25	5875	5817	0,55	507,5	2347	1,1	17,57	1201	5,29
		30	5495	5485	0,4	15,5	28	1	1,1	25	2,57
		40	4907	4905	0,34	7,2	4	0,7	0,4	5	1,29
		50	4374	4372	0,15	3,1	5	0,4	0,4	9	0,84
average					**0,46**	**269,45**		**0,75**	**14,25**		**6,85**

Table 2

Computational results for *ORLIB* and *GR* instances

References

[1] Beasley, J. E., *OR-Library: distributing test problems by electronic mail*, Journal of the Operational Research Society **41** (1990), 1069–1072.

[2] Briant, O., and D. Naddef, *The optimal Diversity management problem*, Operations Research **52**(4) (2004), 515–526.

[3] Elloumi, S., *A tighter formulation of the p-median problem*, J. Comb. Optim., **1**9 (2004), 69–83.

[4] Fung, G., and O.L.Mangasarian, *Semi-supervised support vector machines for unlabeled data classification*, Optimization Methods and Software, **1**5 (2001), 29–44.

[5] Goldengorin, B., D. Krushinsky and V. Kuz'menko, *The problem of margin calculation and its reduction via the p-Median problem model*, AIC'09: Proceedings of the 9th WSEAS international conference on Applied informatics and communications, (2009), 444–449.

[6] Reese, J., *Solution methods for the p-median problem: an annotated bibliography*, Networks **4**8(3) (2006), 125-142.

[7] Resende, M. G. C., and R. F. Werneck, *A hybrid heuristic for the p-median problem*, Journal of Heuristics **1**0(1), (2004), 59–88.

[8] Senne, E. L. F., and L. A. N. Lorena, "Lagrangean/ surrogate heuristics for p-median problems," In M. Laguna and J. L. Gonzalez-Verde editors, Computing tools for modeling, Optimization and simulation: Interfaces in Computer Science and Operations Research, Kluwer, (2000), 115–130.

[9] http://www2.research.att.com/ mgcr/popstar/downloads.html.