J Comb Optim (2010) 19: 69–83
DOI 10.1007/s10878-008-9162-0

A tighter formulation of the p-median problem

Sourour Elloumi

Published online: 23 May 2008

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. It can be viewed as a variation of the uncapacitated facility location problem. We propose a new formulation of this problem by a mixed integer linear problem. We show that this formulation, while it has the same value by LP-relaxation, can be much more efficient than two previous formulations. The computational experiment performed on two sets of benchmark instances has showed that the efficiency of the standard branch-and-cut algorithm has been significantly improved. Finally, we explore the structure of the new formulation in order to derive reduction rules and to accelerate the LP-relaxation resolution.

Keywords Facility location · p-median · Integer programming · Formulation

1 Introduction

We consider the *p-median* problem which has been extensively studied in the literature. Let N be the number of clients, called $C_1, C_2, \ldots, C_N$, let M be the number of potential sites or facilities, called $F_1, F_2, \ldots, F_M$, 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 (Briant and Naddef 2004; Fung and Mangasarian 2001) for recent applications. A recent review on solution methods can be found in (Reese 2006).

S. Elloumi (✉)
CEDRIC-CNAM, 292 Rue Saint-Martin, 75141 Paris cedex 03, France
e-mail: sourour.elloumi@cnam.fr

1.1 The classical formulation

The p-median problem is usually formulated by (1)–(6) (ReVelle and Swain 1970). For any feasible solution (x, y), $y_j = 1$ if and only if facility F_j is open, $x_{ij} = 1$ if and only if client C_i is assigned to facility F_j.

Formulation(CF)

$$\min \ cf(x, y) = \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij} x_{ij} \tag{1}$$

s.t.

$$\sum_{j=1}^{M} y_j = p, \tag{2}$$

$$\sum_{j=1}^{M} x_{ij} = 1, \quad i = 1, \ldots, N, \tag{3}$$

$$x_{ij} \le y_j, \quad i = 1, \ldots, N; \ j = 1, \ldots, M, \tag{4}$$

$$x_{ij} \ge 0, \quad i = 1, \ldots, N; \ j = 1, \ldots, M, \tag{5}$$

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

Constraint (2) fixes the number of open facilities to p, constraints (3) assign each client to exactly one facility, constraints (4) forbid the assignment of a client to a closed facility.

Formulation CF is the most commonly used in the literature. In several recent publications, the authors start from formulation CF and apply sophisticated techniques such as branch-and-cut-and-price (Avella et al. 2007) or semi-Lagrangian relaxation (Beltran et al. 2006). In (Church 2003), improvements of this formulation are proposed by a series of variables and constraints reductions. In (Avella and Sassano 2001) CF is studied from a polyhedral point of view and new facet-defining inequalities are introduced.

1.2 The alternative formulation

This formulation was introduced in (Cornuejols et al. 1980, Theorem 5.1) and is also presented in (Cornuejols et al. 1990). For any client i, let K_i be the number of different distances from i to any facility. It follows that $K_i \le M$. Let $D_i^1 < D_i^2 < \cdots < D_i^{K_i}$ be these distances, sorted. For each client i, let us define a hierarchy of neighborhoods in the following manner: For k in $1..K_i$, denote by V_i^k the kth neighborhood of client i, i.e. the set of facility sites located within distance D_i^k from i, or, more formally, $V_i^k = \{j : d_{ij} \le D_i^k\}$. In this definition, V_i^1 is the set of sites located at distance D_i^1 from i, and $V_i^{K_i}$ is the set of all facility sites. Hence, in an optimal solution to the p-median problem, a client i is assigned to its smallest neighborhood containing an open facility.

Formulation AF uses the same y variables as in formulation CF and introduces new variables z. For any client i and k in $1, \ldots, K_i$, $z_i^k = 1$ if and only if all the sites in V_i^k are closed and 0 otherwise.

In any feasible solution (z, y) and for any client i, the z_i^k values constitute a non-increasing geometrical progression which first member z_i^1 is 0 if client 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.

Formulation(AF)

$$\min\ af(z, y) = \sum_{i=1}^{N}\left(D_i^1 + \sum_{k=1}^{K_i-1}(D_i^{k+1} - D_i^k)z_i^k\right) \tag{7}$$

s.t.

$$\sum_{j=1}^{M} y_j = p, \tag{8}$$

$$z_i^k + \sum_{j:d_{ij}\le D_i^k} y_j \ge 1, \quad i = 1, \ldots, N;\ k = 1, \ldots, K_i, \tag{9}$$

$$z_i^{K_i} = 0, \quad i = 1, \ldots, N, \tag{10}$$

$$z_i^k \ge 0, \quad i = 1, \ldots, N;\ k = 1, \ldots, K_i, \tag{11}$$

$$y_j \in \{0, 1\}, \quad j = 1, \ldots, M. \tag{12}$$

Constraint (8) is the same as (2). Constraints (9) say that for any client i and for any of its neighborhoods V_i^k, either at least one facility is open in V_i^k or $z_i^k = 1$. Constraints (10) ensure that for every client i, at least one facility is open in $V_i^{K_i}$ that is precisely the set of all facilities. We can observe that Constraints (10) are actually variable fixing constraints. One can build a different formulation by taking into account these variable fixing but we prefer to keep the above formulation. Last, the objective function (7) sums up the allocation distances of all clients. For any client i, that distance is equal to D_i^1 if $z_i^1 = 0$, D_i^2 if $z_i^1 = 1$ and $z_i^2 = 0$, etc.

A substantial advantage of the alternative formulation AF is that it can be much smaller than formulation CF. Both of them have the same number of 0-1 variables y. In formulation CF, the number of linear variables x is $N \times M$ and the number of constraints is $(1 + N + N \times M)$. In formulation AF, the number of linear variables z is equal to $K = \sum_{i=1}^{N} K_i$. The number of constraints is equal to $(1 + N + K)$. Formulation AF has roughly $N \times M - K$ less variables and constraints than formulation CF. If the distance matrix is sparse or if many facilities are equidistant from a given client then K can be significantly smaller than $N \times M$. Take the extreme case where for all i, $K_i = 2$ which corresponds to the situation where, for any client, there is a set of close facilities and a set of far facilities. In that case, $K = 2N$ and the size of formulation AF is independent from M. Another extreme case is when the distance matrix is full with $K_i = M$ different distances per client i. In this case, the two formulations have the same sizes.

In real life instances of the p-median problem, K_i can be much smaller than M. Indeed, it often occurs that many facilities are equidistant from a given site. Further, formulation AF can also be built from an approximation of the distances. Considering that any distances that are similar in a certain sense are identical can lead to a more tractable MIP which resolution provides an approximate solution.

However, as we will see in Sect. 3, formulation AF is not always better than formulation CF if compared on the criterion of the running time needed by a standard MIP solver. In spite of a gain in size and an identical lower bound provided by LP-relaxation, AF can need much more time. We prove in this paper that this drawback of formulation AF can be eliminated in a similar but tighter formulation NF.

The remainder of this paper is organized as follows. In Sect. 2, we present formulation NF and show its theoretical properties compared to formulations AF and CF. A computational experience comparing the three formulations is outlined in Sect. 3. In Sect. 4, we give a few ideas for exploiting the structure of NF in order to reduce the problem size or to accelerate the resolution of its LP-relaxation.

2 The new formulation

The following nonlinear equations provide a definition of the z variables of formulation AF:

$$z_i^k = \prod_{j \in V_i^k} (1 - y_j), \quad i = 1, \ldots, N;\ k = 1, \ldots, K_i \tag{13}$$

from which we can deduce the following recursive definition of the z variables:

$$z_i^1 = \prod_{j: d_{ij} = D_i^1} (1 - y_j), \quad i = 1, \ldots, N, \tag{14}$$

$$z_i^k = z_i^{k-1} \prod_{j: d_{ij} = D_i^k} (1 - y_j), \quad i = 1, \ldots, N;\ k = 2, \ldots, K_i \tag{15}$$

and the following new formulation:

Formulation(NF)

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

s.t.

$$\sum_{j=1}^{M} y_j = p, \tag{16}$$

$$z_i^1 + \sum_{j: d_{ij} = D_i^1} y_j \geq 1, \quad i = 1, \ldots, N, \tag{17}$$

$$z_i^k + \sum_{j:d_{ij}=D_i^k} y_j \geq z_i^{k-1}, \quad i=1,\ldots,N;\ k=2,\ldots,K_i, \tag{18}$$

$$z_i^{K_i} = 0, \quad i=1,\ldots,N, \tag{19}$$

$$z_i^k \geq 0, \quad i=1,\ldots,N;\ k=1,\ldots,K_i, \tag{20}$$

$$y_j \in \{0,1\}, \quad j=1,\ldots,M. \tag{21}$$

Formulations AF and NF use the same set of variables y and z, have exactly the same objective function and describe the same set of integer points. However, we show in the following proposition that the polytopes corresponding to their LP-relaxations are different.

Proposition 1 *Let $\underline{AF}$ and $\underline{NF}$ be the LP-relaxations of formulations AF and NF respectively. The feasible solution set of $\underline{NF}$ is included within the feasible solution set of $\underline{AF}$. This inclusion can be strict.*

Proof Let (z, y) be a feasible solution to $\underline{NF}$. In order to show that (z, y) is a feasible solution to $\underline{AF}$, we only have to check that Constraints (9) are satisfied for any i and k. For $i = 1,\ldots,N$:

- for $k = 1$, Constraints (9) are identical to Constraints (17)
- for $k = 2,\ldots,K_i$, sum up Constraints (17) and Constraints (18) for $\ell = 2,\ldots,k$ and then simplify to get Constraints (9).

To complete the proof, we present now an example where a feasible solution to $\underline{AF}$ is not feasible to $\underline{NF}$. Consider an instance where $V_1^1 = \{1,2\}$ and $V_1^2 = \{1,2,3\}$. The corresponding Constraints (9) of $\underline{AF}$ are $z_1^1 + y_1 + y_2 \geq 1$ and $z_1^2 + y_1 + y_2 + y_3 \geq 1$. The corresponding Constraints (17) and (18) of $\underline{NF}$ are $z_1^1 + y_1 + y_2 \geq 1$ and $z_1^2 + y_3 \geq z_1^1$. The fractional solution $z_1^1 = 1$, $z_1^2 = 0.6$, $y_1 = y_2 = 0.1$, and $y_3 = 0.2$ is feasible for $\underline{AF}$ but not for $\underline{NF}$. □

Although formulations AF and NF lie on different polytopes, the following proposition shows that their LP-relaxation optimal values are identical.

Proposition 2 *Let $\underline{AF}$ and $\underline{NF}$ be the LP-relaxations of formulations AF and NF respectively. Problems $\underline{AF}$ and $\underline{NF}$ have the same optimal values.*

Proof Let v_{AF} and v_{NF} be the optimal values of problems $\underline{AF}$ and $\underline{NF}$ respectively. The inequality $v_{AF} \leq v_{NF}$ follows from Proposition 1 and from the fact that the two problems have the same objective function. We now show that $v_{NF} \leq v_{AF}$.

Let $(\tilde{z}, \tilde{y})$ be an optimal solution to $\underline{AF}$. Let us show that $(\tilde{z}, \tilde{y})$ is a feasible solution to $\underline{NF}$, i.e. it satisfies Constraints (17) and (18). For $i = 1,\ldots,N$:

- As $(\tilde{z}, \tilde{y})$ is an optimal solution to $\underline{AF}$, and since the coefficient of z_i^1 in the objective function is nonnegative, $\tilde{z}_i^1 = \max(0, 1 - \sum_{j:d_{ij}\leq D_i^1} \tilde{y}_j)$ and hence Constraints (17) are satisfied.

– Now, for $k = 2, \ldots, K_i$, by the same reasons as above, $\tilde{z}_i^{k-1} = \max(0, 1 - \sum_{j:d_{ij} \le D_i^{k-1}} \tilde{y}_j)$. If $\tilde{z}_i^{k-1} = 0$ then Constraint (18) is trivially satisfied. If $\tilde{z}_i^{k-1} = 1 - \sum_{j:d_{ij} \le D_i^{k-1}} \tilde{y}_j$ then $\tilde{z}_i^{k-1} - \sum_{j:d_{ij}=D_i^k} \tilde{y}_j = 1 - \sum_{j:d_{ij} \le D_i^k} \tilde{y}_j \le \tilde{z}_i^k$ by Constraints (9). Again, Constraint (18) is satisfied. □

Now, we prove that the LP-relaxation optimal values of NF and CF are identical. This result could also be deduced from Proposition 2 and from (Cornuejols et al. 1980) where formulations AF and CF are proved to have the same LP-relaxation optimal values.

Proposition 3 *Let $\underline{CF}$ and $\underline{NF}$ be the LP-relaxations of formulations CF and NF respectively. The linear programs $\underline{CF}$ and $\underline{NF}$ are equivalent in the sense that from an optimal solution of the one, we can deduce a solution to the other one, with the same objective value.*

Proof Let (x^*, y^*) be an optimal solution to $\underline{CF}$. We compute a solution $(\tilde{z}, \tilde{y})$ to $\underline{NF}$ by:

◇ $\tilde{y} = y^*$

◇ $\tilde{z}_i^1 = 1 - \sum_{j:d_{ij}=D_i^1} x_{ij}^*$ for $i = 1, \ldots, N$

◇ $\tilde{z}_i^k = 1 - \sum_{j:d_{ij} \le D_i^k} x_{ij}^* = \tilde{z}_i^{k-1} - \sum_{j:d_{ij}=D_i^k} x_{ij}^*$ for $i = 1, \ldots, N; k = 2, \ldots, K_i$.

The fact that $(\tilde{z}, \tilde{y})$ satisfy (16)–(20) follows from the fact that (x^*, y^*) satisfy (2)–(5). Let us now compare the objective values:

$$\begin{aligned}
nf(\tilde{z}, \tilde{y}) &= \sum_{i=1}^{N} \left(D_i^1 + \sum_{k=1}^{K_i-1} (D_i^{k+1} - D_i^k) \tilde{z}_i^k \right) \\
&= \sum_{i=1}^{N} \left(D_i^1 (1 - \tilde{z}_i^1) + \sum_{k=2}^{K_i-1} D_i^k (\tilde{z}_i^{k-1} - \tilde{z}_i^k) + D_i^{K_i} \tilde{z}_i^{K_i-1} \right) \\
&= \sum_{i=1}^{N} \left(D_i^1 \left(\sum_{j:d_{ij}=D_i^1} x_{ij}^* \right) + \sum_{k=2}^{K_i-1} D_i^k \left(\sum_{j:d_{ij}=D_i^k} x_{ij}^* \right) \right. \\
&\qquad \left. + D_i^{K_i} \left(\tilde{z}_i^{K_i} + \sum_{j:d_{ij}=D_i^{K_i}} x_{ij}^* \right) \right) \\
&= \sum_{i=1}^{N} \left(\sum_{k=1}^{K_i} D_i^k \left(\sum_{j:d_{ij}=D_i^k} x_{ij}^* \right) \right) = \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij} x_{ij}^* = cf(x^*, y^*).
\end{aligned}$$

Now, let $(\tilde{z}, \tilde{y})$ be an optimal solution to $\underline{NF}$. We compute a solution (x^*, y^*) to $\underline{CF}$ by:

◇ $y^* = \tilde{y}$

◇ For $i = 1, \dots, N$, vector $x_{i.}^*$ is obtained as any solution of the following linear system with M variables denoted by r_j:

$$\begin{cases} \sum_{j:d_{ij}=D_i^1} r_j = 1 - \tilde{z}_i^1, \\ \sum_{j:d_{ij}=D_i^k} r_j = \tilde{z}_i^{k-1} - \tilde{z}_i^k, \quad k = 2, \dots, K_i, \\ 0 \le r_j \le \tilde{y}_j, \quad j = 1, \dots, M. \end{cases} \tag{22}$$

System (22) has a particular structure. It can be decomposed into the K_i following bin-packing problems:

$$\begin{cases} \sum_{j:d_{ij}=D_i^1} r_j = 1 - \tilde{z}_i^1, \\ 0 \le r_j \le \tilde{y}_j, \quad j : d_{ij} = D_i^1 \end{cases} \tag{23}$$

and, for $k = 2, \dots, K_i$:

$$\begin{cases} \sum_{j:d_{ij}=D_i^k} r_j = \tilde{z}_i^{k-1} - \tilde{z}_i^k, \\ 0 \le r_j \le \tilde{y}_j, \quad j : d_{ij} = D_i^k. \end{cases} \tag{24}$$

Systems (23) and (24) are feasible since constraints (17) and (18) are satisfied by $(\tilde{z}, \tilde{y})$. Furthermore, $1 - \tilde{z}_i^1 \ge 0$ and $\tilde{z}_i^{k-1} - \tilde{z}_i^k \ge 0$ are ensured by the optimality of solution $(\tilde{z}, \tilde{y})$ where $\tilde{z}_i^1 = \max(0, 1 - \sum_{j:d_{ij}=D_i^1} y_j)$ and, for $k \ge 2$, $\tilde{z}_i^k = \max(0, \tilde{z}_i^{k-1} - \sum_{j:d_{ij}=D_i^k} y_j)$. Hence, for a given $i = 1, \dots, N$, one can solve (23) and (24) and then fix $x_{ij}^* = r_j$ for $j = 1, \dots, M$.

Let us now compare the objective values:

$$\begin{aligned} cf(x^*, y^*) &= \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij} x_{ij}^* \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K_i} D_i^k \sum_{j:d_{ij}=D_i^k} x_{ij}^* \\ &= \sum_{i=1}^{N} \left(D_i^1 (1 - \tilde{z}_i^1) + \sum_{k=2}^{K_i} D_i^k (\tilde{z}_i^{k-1} - \tilde{z}_i^k) \right) = nf(\tilde{z}, \tilde{y}). \end{aligned}$$

□

The three MILPs CF, AF, and NF formulate the same optimization problem and have the same lower bound by linear relaxation. Nevertheless, we show in Sect. 3 that it can be of much interest to use formulation NF rather than formulations CF or AF.

3 Computational comparison

Our objective here is to compare the three formulations CF, AF, and NF from the computational time point of view. We already know that all of them have the same

LP-relaxation optimal value, that AF and NF have the same size, and have at worst the same size as CF.

We use AMPL as a modeling language to build the MIP corresponding either to formulation CF, AF, or NF. We then solve the obtained MIP by use of the branch-and-cut algorithm of CPLEX 8.1. All the software parameters are set to their default values except that we stop the branch-and-cut algorithm after one hour of CPU time. Our experiments are carried out on a portable PC with a Pentium IV of 1600 MHz and 1024 MB of RAM.

We first consider the 15 largest instances from the popular 40 pmedian instances of ORLIB (Beasley 1990). Each of these instances is a graph with a given value for p. A node in the graph represents both a client and a potential facility site, and the distance d_{ij} between two nodes is the length of a shortest path linking them. For the considered instances, the number $N = M$ of clients and facility sites varies from 600 to 900 and p varies from 5 to 200. The second class of instances we study is rw. Originally proposed in (Resende and Werneck 2004), it corresponds to completely random distance matrices. In every case, the number of potential facilities (M) is equal to the number of clients (N). The distance between each facility and each client has an integer value taken uniformly at random from the interval $[1, N]$. Here, we consider three values of N: 100, 250, and 500. Instances names are rw100, rw250, and rw500 respectively. For each value of N and an identical distance matrix, several values of p are tested.

Tables 1 and 2 sum up the results obtained for the two sets of instances. Columns 1–4 give the instance characteristics: the instance name, the number $N = M$ of clients and facility sites, p, and the ratio NM/K (where $K = \sum_i K_i$) that gives the ratio of the sizes of the classical formulation CF by the other formulations AF and NF. Column 5 (resp. 6) gives the CPU time needed to solve the LP-relaxation (resp. the MIP) obtained by formulation CF. The CPU times needed by formulation AF (resp. NF) are given in the following (resp. last) two columns. All CPU times are reported in seconds.

In Tables 1, we can observe that the ratio NM/K varies from 12 to 24 and is equal to about 16 in average. This means that the MIPs associated to formulations AF and NF have about 16 times less variables and constraints than formulation CF. However, formulation AF does not always have a better performance than formulation CF from the CPU time point of view. Over the 15 tested instances, 12 (resp. 10) are solved within one hour by formulation CF (resp. NF). Formulation NF has a much better performance since the LP-relaxation solution time is in average 16 (resp. 13) times smaller than by formulation CF (resp. AF). Formulation NF allows to solve the 15 instances to optimality within one hour and the MIP resolution needs only 57 seconds in average. Trying to understand why formulation AF takes more time than formulation CF for many instances, we observed for instance pmed33 that formulation AF (resp. CF) has 31852 (resp. 640800) variables, 31853 (resp. 640801) constraints, and 10439822 (resp. 1470700) non zero elements in the row matrix. Hence, the row matrix of formulation AF is much more dense than the row matrix of formulation CF. It is well known that linear programming algorithms perform better with sparse row matrices. The gain in size of formulation AF is lost by its denser row matrix. This is not the case with formulation NF. For the pmed33 instance, the

Table 1 Results for pmedian instances

	$N=M$	p	NM/K	CF		AF		NF	
				t LP	t 0-1	t LP	t 0-1	t LP	t 0-1
pmed26	600	5	12	340	1473	116	–	26	124
pmed27	600	10	12	321	781	1858	3187	23	77
pmed28	600	60	12	86	90	226	274	12	14
pmed29	600	120	12	54	58	74	102	6	8
pmed30	600	200	12	46	51	40	68	4	6
pmed31	700	5	16	460	2132	99	–	31	102
pmed32	700	10	15	610	1062	196	330	38	102
pmed33	700	70	16	124	129	296	328	15	16
pmed34	700	140	15	83	88	104	156	8	10
pmed35	800	5	19	524	3503	135	–	40	185
pmed36	800	10	18	1139	–	372	–	53	3057
pmed37	800	80	18	167	174	594	638	18	21
pmed38	900	5	23	1264	–	234	–	49	500
pmed39	900	10	24	910	–	111	–	41	436
pmed40	900	90	22	264	274	593	649	22	25

–: the MIP resolution was stopped after 1 hour

size is of course identical to the size of formulation AF but the number of non zero elements is 553004.

Table 2 presents the results for the rw instances. Here, the ratio NM/K is uniformly equal to about 1.6, which is not surprising in view of the instances generation. We can observe that, as in Table 1, formulation NF has the best performance among the three formulations. From the two tables, it appears that, compared to CF, formulation NF provides a speedup that is of the same order as the ratio NM/K, or, equivalently, as the gain in size. However, the gain in size is not so profitable for formulation AF that has a dramatically bad performance in this case. Recall that formulations NF and AF have the same sizes and the same objective function. The difference comes from the row matrices. For the rw500 instances, both formulations AF and NF have about 158500 variables and constraints, in formulation NF, the row matrix has 566514 non zero elements while it has 39747131 non zero elements in formulation AF. This much larger density of the row matrix seems to explain why these instances cannot be handled by our experimental environment: an "unrecoverable failure" error message occurs. Last, let us observe that, for the rw500 instances, formulation CF has about 250500 variables and constraints, and the row matrix has 750500 non zero elements.

As a matter of fact, we can show that NF has no more non zero elements in its row matrix than CF. In CF there are $N+3NM$ non zero elements while in NF there are $M+NM+2K$ non zero elements. Recall that $K \leq NM$ to conclude. From all this, it appears that NF is definitely a more interesting formulation than CF.

Table 2 Results for rw instances

	$N = M$	p	NM/K	CF		AF		NF	
				t LP	t 0-1	t LP	t 0-1	t LP	t 0-1
rw100	100	10	1.6	2.08	344	3.46	1654	1.12	152
	100	20	1.6	0.55	9.52	1.45	43.6	0.28	3.41
	100	30	1.6	0.19	0.29	0.55	1.33	0.05	0.12
	100	40	1.6	0.13	0.23	0.31	1.7	0.03	0.10
	100	50	1.6	0.12	0.22	0.25	1.03	0.02	0.09
rw250	250	10	1.6	230	–	839	–	95	–
	250	25	1.6	77	–	477	–	39	–
	250	50	1.6	20	2790	64	–	18	3196
	250	75	1.6	4	5.27	15.38	34.6	1.23	2.29
	250	100	1.6	3	3.7	8.44	24	0.68	1.36
	250	125	1.6	2.1	2.8	4	19.4	0.25	0.92
rw500	500	10	1.6	*	–	**	**	3286	–
	500	25	1.6	2360	–	**	**	1462	–
	500	50	1.6	658	–	**	**	622	–
	500	75	1.6	396	–	**	**	302	–
	500	100	1.6	312	–	**	**	224	–
	500	150	1.6	45	53	**	**	18	24
	500	200	1.6	27	29	**	**	3	9
	500	250	1.6	16	18	**	**	1	8

–: the MIP resolution was stopped after 1 hour
*: LP not solved after 1 hour
**: the instance couldn't be handled by Cplex 8.1

4 Exploiting the new formulation properties

In this section, we explore two axes for improving the solution process if one uses formulation NF. The first axe consists in a preprocessing algorithm which aims at reducing of the size of the mathematical program. The second axe takes benefit from the structure of the z variables and the associated constraints in order to reduce the solution time of the LP-relaxation of formulation NF.

4.1 Reducing the program size

Two reduction rules can be trivially deduced from the definition of the z variables in (13).

Rule R1: For any client i, if V_i^1 is a singleton $\{y_\alpha\}$ then $z_i^1 = 1 - y_\alpha$ holds for any feasible solution. Variable z_i^1 can be substituted by $1 - y_\alpha$ and constraint $z_i^1 + y_\alpha \geq 1$ that defines variable z_i^1 can be eliminated.

Rule R2: If, for any i, k, i', k', $V_i^k = V_{i'}^{k'}$ then $z_i^k = z_{i'}^{k'}$ holds for any feasible solution. Variable $z_{i'}^{k'}$ can be replaced by z_i^k and constraint $z_{i'}^{k'} + \sum_{j:d_{i'j}=D_{i'}^{k'}} y_j \geq z_{i'}^{k'-1}$ that defines variable $z_{i'}^{k'}$ can be eliminated.

Rule R3: If, for any i, i', $V_i^{K_i-1} = V_{i'}^{K_{i'}-1}$ then Rule R2 can be applied to deduce that $z_i^{K_i-1} = z_{i'}^{K_{i'}-1}$. Further, in this case, the set of facilities j such that $d_{ij} = D_i^{K_i}$ is equal to the set of facilities j such that $d_{i'j} = D_{i'}^{K_{i'}}$. Finally, as $z_i^{K_i} = z_{i'}^{K_{i'}} = 0$, we can eliminate constraint $z_{i'}^{K_{i'}} + \sum_{j:d_{i'j}=D_{i'}^{K_{i'}}} y_j \geq z_{i'}^{K_{i'}-1}$.

In the Appendix, the above reduction rules are applied to a small example.

4.2 Improving the resolution of the LP-relaxation

Let P be the problem obtained from NF by LP-relaxation, i.e. by replacing constraints (21) by $0 \leq y_j \leq 1$. Let H be an N-vector such that $1 \leq H_i \leq K_i$ and let P_H be the following linear program:

Problem(P_H)

$$\min \sum_{i=1}^{N} \left(D_i^1 + \sum_{k=1}^{H_i-1} (D_i^{k+1} - D_i^k) z_i^k \right) \tag{25}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} y_j = p, \tag{26}$$

$$z_i^1 + \sum_{j:d_{ij}=D_i^1} y_j \geq 1, \quad i = 1, \ldots, N, \tag{27}$$

$$z_i^k + \sum_{j:d_{ij}=D_i^k} y_j \geq z_i^{k-1}, \quad i = 1, \ldots, N;\ k = 2, \ldots, H_i, \tag{28}$$

$$z_i^k \geq 0, \quad i = 1, \ldots, N;\ k = 1, \ldots, H_i, \tag{29}$$

$$0 \leq y_j \leq 1, \quad j = 1, \ldots, M. \tag{30}$$

Problem P_H is obtained from P by eliminating columns z_i^k for $i = 1, \ldots, N$ and $k = H_i + 1, \ldots, K_i$, constraints (19), and constraints (18) for $i = 1, \ldots, N$ and $k = H_i + 1, \ldots, K_i$. We define μ as the operator that takes a feasible solution (z, y) for P 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 for P, $(y, \mu(z))$ is a feasible solution for P_H, with an equal to or smaller objective function value: P_H is a relaxation for P. 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 P. This time, if (z, y) is a feasible solution for P_H, $(y, \nu(z))$ is not always a feasible solution for P_H.

Proposition 4 *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 $\tilde{z}_1^{H_1} = \tilde{z}_2^{H_2} = \cdots = \tilde{z}_N^{H_N} = 0$ then $(\nu(\tilde{z}), \tilde{y})$ is an optimal solution to P and the two problems P and P_H have the same optimal values.*

Proof The fact that $\tilde{z}_1^{H_1} = \tilde{z}_2^{H_2} = \cdots = \tilde{z}_N^{H_N} = 0$ implies that all constraints (18) and constraints (19) are satisfied by $(\nu(\tilde{z}), \tilde{y})$. The two solutions $(\tilde{z}, \tilde{y})$ and $(\nu(\tilde{z}), \tilde{y})$ have the same objective values since $\nu(\tilde{z})$ consists of adding zero coordinates to $\tilde{z}$. We can just recall that P_H is a relaxation of P to conclude. □

We can now deduce an algorithm for solving problem P by solving a series of problems P_H.

Algorithm **Solve** P

Step 1. *Choose* initial values for the N-vector H such that $1 \leq H_i \leq K_i$
Step 2. compute an optimal solution $(\tilde{z}, \tilde{y})$ to problem P_H
Step 3. if $\tilde{z}_1^{H_1} = \tilde{z}_2^{H_2} = \cdots = \tilde{z}_N^{H_N} = 0$ then goto Step 6
Step 4. *Increase* H
Step 5. goto Step 2
Step 6. STOP. $(\nu(\tilde{z}), \tilde{y})$ is an optimal solution to P

The above algorithm is a kind of generic algorithm where Step 1 and Step 4 need to be precised in order to get an ready-to-implement algorithm. Step 1 consists in choosing initial values for vector H. Different strategies can be elaborated for this. A simple strategy consists in fixing any H_i to a fraction of K_i. A more sophisticated strategy can be to compute a greedy solution y^* to the initial p-median problem and then, for any client i, compute H_i as the index of the smallest neighborhood client i is assigned to in solution y^*. Step 4 consists in increasing vector H. This amounts to adding a number of columns and rows to the previous P_H problem. Again, different strategies can be elaborated. One of them consists in increasing, by a fixed number, those H_i for which $\tilde{z}_i^{H_i} > 0$.

By a brute-force implementation of Step 1, based on a greedy solution, and Step 4, where H_i is increased by p at each iteration and whenever possible, we could observe that solving the LP-relaxation of instance rw250 (resp. rw500) with $p = 10$ took 30 seconds (resp. 1000 seconds) against 95 seconds (resp. 3286 seconds) by direct resolution.

We think that an appropriate choice for Steps 1 and 4 can be tuned after a specific computational experiment is carried out. The tuned algorithm should represent an important gain if compared to the direct resolution of P by an LP algorithm.

Acknowledgements The author wishes to thank two anonymous referees for the helpful comments.

Appendix A

A.1 A basic example

Let the following example in Table 3 with $N = 5$ clients, $M = 4$ potential facility sites, and $p = 2$.

Table 3 Data for the basic example with $N=5$, $M=4$, and $p=2$

	F_1	F_2	F_3	F_4
C_1	1	2	1	4
C_2	6	1	2	3
C_3	5	2	3	1
C_4	3	3	3	8
C_5	4	5	3	2

Table 4 Neighborhoods for the basic example of Table 3

C_1	$V_1^1=\{F_1,F_3\}$	$V_1^2=\{F_1,F_2,F_3\}$	$V_1^3=\{F_1,F_2,F_3,F_4\}$	
C_2	$V_2^1=\{F_2\}$	$V_2^2=\{F_2,F_3\}$	$V_2^3=\{F_2,F_3,F_4\}$	$V_2^4=\{F_1,F_2,F_3,F_4\}$
C_3	$V_3^1=\{F_4\}$	$V_3^2=\{F_2,F_4\}$	$V_3^3=\{F_2,F_3,F_4\}$	$V_3^4=\{F_1,F_2,F_3,F_4\}$
C_4	$V_4^1=\{F_1,F_2,F_3\}$	$V_4^2=\{F_1,F_2,F_3,F_4\}$		
C_5	$V_5^1=\{F_4\}$	$V_5^2=\{F_3,F_4\}$	$V_5^3=\{F_1,F_3,F_4\}$	$V_5^4=\{F_1,F_2,F_3,F_4\}$

The number of different distances per client are here $K_1=3$, $K_2=4$, $K_3=4$, $K_4=2$, and $K_5=4$. An extensive list of the neighborhoods is given by the following Table 4.

The formulation of this basic example by CF gives a mathematical program with 4 y variables, 20 x variables, and 26 constraints. By formulation NF, the obtained mathematical program contains 4 y variables, 17 z variables, and 23 constraints.

A.2 Formulation

NF applied to our basic example gives the following MIP:

$$(NFex)$$

$$\min \quad 8+z_1^1+2z_1^2+z_2^1+z_2^2+3z_2^3+z_3^1+z_3^2+2z_3^3+5z_4^1+z_5^1+z_5^2+z_5^3$$

s.t.

$$y_1+y_2+y_3+y_4=p,$$

$$z_1^1+y_1+y_3\geq 1, \tag{31}$$

$$z_1^2+y_2\geq z_1^1, \tag{32}$$

$$z_1^3+y_4\geq z_1^2, \tag{33}$$

$$z_1^3=0, \tag{34}$$

$$z_2^1+y_2\geq 1, \tag{35}$$

$$z_2^2+y_3\geq z_2^1, \tag{36}$$

$$z_2^3+y_4\geq z_2^2, \tag{37}$$

$$z_2^4+y_1\geq z_2^3, \tag{38}$$

$$z_2^4 = 0, \tag{39}$$

$$z_3^1 + y_4 \geq 1, \tag{40}$$

$$z_3^2 + y_2 \geq z_3^1, \tag{41}$$

$$z_3^3 + y_3 \geq z_3^2, \tag{42}$$

$$z_3^4 + y_1 \geq z_3^3, \tag{43}$$

$$z_3^4 = 0, \tag{44}$$

$$z_4^1 + y_1 + y_2 + y_3 \geq 1, \tag{45}$$

$$z_4^2 + y_4 \geq z_4^1, \tag{46}$$

$$z_4^2 = 0, \tag{47}$$

$$z_5^1 + y_4 \geq 1, \tag{48}$$

$$z_5^2 + y_3 \geq z_5^1, \tag{49}$$

$$z_5^3 + y_1 \geq z_5^2, \tag{50}$$

$$z_5^4 + y_2 \geq z_5^3, \tag{51}$$

$$z_5^4 = 0, \tag{52}$$

$$z_i^k \geq 0, \tag{53}$$

$$y_j \in \{0, 1\}, \quad j = 1, \ldots, 4. \tag{54}$$

A.3 Reduction of formulation NF

1. We eliminate the fixed variables and the corresponding constraints (34), (39), (44), (47), and (52).
2. By applying Rule R1, as V_2^1, V_3^1, and V_5^1 are singletons (see Table 4), we can deduce $z_2^1 = 1 - y_2$, $z_3^1 = 1 - y_4$, and $z_5^1 = 1 - y_4$. We can therefore eliminate constraints (35), (40), and (48).
3. By applying Rule R2, as $V_4^1 = V_1^2$ and $V_3^3 = V_2^3$ (see Table 4), we can deduce $z_4^1 = z_1^2$ and $z_3^3 = z_2^3$. We can therefore eliminate constraints (45) and (42).
4. By applying Rule R3, we can moreover eliminate constraints (46) and (43).

The final MIP for the basic example of Table 3 has 4 y variables, 7 z variables, and 11 constraints.

$$(NFexr)$$

$$\min \; 8 + (1 - y_2) + 2(1 - y_4) + z_1^1 + 7z_1^2 + z_2^2 + 5z_2^3 + z_3^2 + z_5^2 + z_5^3$$

s.t.

$$y_1 + y_2 + y_3 + y_4 = p,$$

$$z_1^1 + y_1 + y_3 \geq 1,$$

$$z_1^2 + y_2 \geq z_1^1,$$
$$y_4 \geq z_1^2,$$
$$z_2^2 + y_3 \geq 1 - y_2,$$
$$z_2^3 + y_4 \geq z_2^2,$$
$$y_1 \geq z_2^3,$$
$$z_3^2 + y_2 \geq 1 - y_4,$$
$$z_5^2 + y_3 \geq 1 - y_4,$$
$$z_5^3 + y_1 \geq z_5^2,$$
$$y_2 \geq z_5^3,$$
$$z_i^k \geq 0,$$
$$y_j \in \{0, 1\}, \quad j = 1, \ldots, 4.$$

References

Avella P, Sassano A (2001) On the p-median polytope. Math Program 89:395–411
Avella P, Sassano A, Vasil'ev I (2007) Computational study of large-scale p-median problems. Math Program 109(1):89–114
Beasley JE (1990) OR-Library: distributing test problems by electronic mail. J Oper Res Soc 41:1069–1072. http://mscmga.ms.ic.ac.uk/info.html
Beltran C, Tadonki C, Vial J-Ph (2006) Solving the p-median problem with a semi-Lagrangian relaxation. Comput Optim Appl 35(2):239–260
Briant O, Naddef D (2004) The optimal diversity management problem. Oper Res 52(4):515–526
Church RL (2003) COBRA: a new formulation of the classic p-median location problem. Ann Oper Res 122:103–120
Cornuejols G, Nemhauser G, Wolsey LA (1980) A canonical representation of simple plant location problems and its applications. SIMAX 1(3):261–272
Cornuejols G, Nemhauser G, Wolsey LA (1990) The uncapacitated facility location problem. In: Mirchandani PB, Francis RL (eds) Discrete location theory. Wiley, New York
Fung G, Mangasarian OL (2001) Semi-supervised support vector machines for unlabeled data classification. Optim Methods Softw 15:29–44
Reese J (2006) Solution methods for the p-median problem: an annotated bibliography. Networks 48(3):125–142
Resende MGC, Werneck RF (2004) A hybrid heuristic for the p-median problem. J Heuristics 10(1):59–88
ReVelle CS, Swain R (1970) Central facilities location. Geogr Anal 2:30–42