When centers can fail: A close second opportunity

Maria Albareda-Sambola[1], Yolanda Hinojosa[2], Alfredo Marín[3], Justo Puerto[4]

[1] Departamento de Estadística e Investigación Operativa, Universitat Politècnica de Catalunya. Barcelona Tech, Spain
[2] Departamento de Economía Aplicada I, Universidad de Sevilla, Spain
[3] Departamento de Estadística e Investigación Operativa, Universidad de Murcia, Spain
[4] Departamento de Estadística e Investigación Operativa, Universidad de Sevilla, Spain

Abstract

This paper presents the p-next center problem, which aims to locate p out of n centers so as to minimize the maximum cost of allocating customers to backup centers. In this problem it is assumed that centers can fail and customers only realize that their closest (reference) center has failed upon arrival. When this happens, they move to their backup center, i.e., to the center that is closest to the reference center. Hence, minimizing the maximum travel distance from a customer to its backup center can be seen as an alternative approach to handle humanitarian logistics, that hedges customers against severe scenario deteriorations when a center fails.

For this extension of the p-center problem we have developed several different integer programming formulations with their corresponding strengthenings based on valid inequalities and variable fixing. The suitability of these formulations for solving the p-next center problem using standard software is analyzed in a series of computational experiments. These experiments were carried out using instances taken from the previous discrete location literature.

Keywords: Integer Programming, Discrete Location, p-center.

1 Introduction: New location model for humanitarian logistics

The p-center problem (pCP) is a very well-known discrete optimization problem which consists in locating p out of n centers and assigning users to them so as to minimize the maximum distance (cost) between a user and the corresponding (reference) center. It was shown in [19] that pCP is an NP-hard problem. Some applications of pCP are the location of emergency services like ambulances, hospitals or fire stations, since all users should be within a small radius from some emergency center. As a consequence, pCP has been extensively studied, and both exact and heuristic algorithms have been proposed. Recent articles on the field are [10], [14] and [27]. We also refer the interested reader to Chapter 5 of [11]. Capacitated versions of pCP have also received some attention in the literature: A local search heuristic for a capacitated version of pCP was developed in [30], and a special case where all users have equal demands was studied in [2] and [20]; the case of an underlying tree network was approached in [18], whereas in [1] and [28] the authors propose exact algorithms.

Emergency services can, in practice, fail. In particular, after a large disaster, an emergency center located in the affected area is likely to be destroyed or seriously damaged. This kind of situation was given, for example, in May 11, 2011, in the Spanish town of Lorca. As a consequence of two consecutive earthquakes nine people were killed, three hundred were injured and the local hospital had to be evacuated due to the risk of collapse.

This fact has motivated us to extend the p-centerproblem in a new dirctin and define the p-next center problem (pNCP). We are interested in studying the situation where an unpredictable incident can occur in any of the p centers, forcing this center to be closed and all users allocated to it to be reassigned/derived to another center. We assume that these users only realize that their first center (from now on, *reference center*) is unavailable after arriving to it, i.e., no *a priori* information is known. Then they will be redirected to the center closest to the unavailable center (from now on, *backup center*). In the unlikely case of tie between two or more reference centers, we assume that the user chooses her best option, that is to say, the reference center with a closest backup center. It is further assumed that the possibility that both, the reference and the backup centers fail is negligible.

The pNCP can be seen as an extension of the so called *messenger-boy* problem [15] which aims at finding a 1-center in the plane, but where the cost associated with each customer is the sum of the distance between the customer and the center, plus a constant that depends on the customer. In the pNCP, instead of that constant we incorporate the distance from the reference center to its closest other center. Two other related problems that are pretty similar to each other are the *collection depot facility location problem* and the *one-way facility location problem*. In both cases, center objectives are considered, with assignment costs that are also associated with paths that visit three different locations (including the customer). However, in contrast to the p-next center problem, the path to serve a customer visits one of the located centers, and one of the already available depots. Indeed, the pNCP can be regarded as the extension of these problems where depots are restricted to be among the opened centers. Planar and network versions of these problems have also been studied in the literature [4, 5, 13, 32].

From a geometrical point of view this problem has an appealing meaning that is different from the meaning of the p-center. We can consider that, in pCP, we are looking for the centers of p spheres which cover the demand points and such that the largest radius is as small as possible. In our case, the problem is to find p points such that, if each demand point is covered by a sphere centered at the closest point and that center is covered by another sphere centered at its closest point, then the maximal sum of the two radii be minimal. It is clear that, in this regard, our problem adds a redundancy effect to hedge against failures in emergency systems.

The literature on location problems with facility failures is quite extensive. It starts with the seminal paper [12], where the possibility that several facilities in the p-center model may become inactive is studied. A probability that a facility becomes inactive is then introduced and a heuristic procedure is presented. Since then, a large variety of models have been proposed in the literature to incorporate the issue of disruption into facility planning decisions. Broadly speaking, these models can be classified according to two criteria. On the one hand, differences exist depending on the moment when customers realize that their reference facility is not available; most references assume that customers know it in advance and, thus, they travel directly to their backup facility, but other papers, as it is the case in this work, pay attention to situations where they do not know it in advance. For instance, in [7], a model where the customers may have to visit several facilities until finding an available one is analyzed to conclude that centralization and co-location are adequate strategies to locate centers in this case. More recently, the same authors try to explain in [6] how correlated failure probabilities and problem objective (median and center) affect the location decisions when information is available, and when it is not. On the other hand, using or not failure probabilities to include the possibility that facilities become unavailable defines a second classification. Again, the literature is unbalanced. Lots of papers exist where these probabilities play a crucial role in the studied models, as shown in [23], while only a few references concentrate on models that do not require the knowledge of these probabilities to identify sets of facilities that are robust enough to guarantee a reasonable service level even in the case of failures (see, for instance, [17]). In this regard, the present work is basically in the second category, since failure probabilities are not included in the model. However, as it will be seen,

they can be easily incorporated to the model if they were available.

In this paper we introduce and analyze different (mixed) linear integer programming formulations of pNCP, developed from different points of view. We first present a natural formulation that uses binary variables modeling the paths from a site to its backup center, through its reference center. This yields quite a strong formulation at the cost of using a large number of variables (variables with three indices). For this reason, we next explore the capability of smaller sets of variables to model pNCP. In particular, two alternative sets of variables with only two indices are considered, giving rise to two different formulations. One of them, using binary variables to identify pairs of sites where centers are established, is relegated to an appendix in order to focus the reader's attention on the most promising models. In the second case, binary variables indicate either links between a customer and its reference center, or links between a center and its backup center. Finally, as it has been successfully done for pCP (for instance, in [14]), a fourth formulation is studied that combines covering variables with the use of ordered distances and a telescopic objective function.

All formulations are studied and strengthened using valid inequalities and some variable fixing criteria that can be applied when valid upper bounds are available. Finally, a computational experience has been performed to compare their utility to solve pNCP using branch and bound with standard software.

The rest of the paper is organized as follows. First, the formal statement of the problem is given in Section 2 together with the notation that will be used throughout the paper, and a proof of the NP-hardness of the problem. The natural formulation, with three-indexed variables, is presented and analyzed in Section 3. Then, in Section 4, the formulation that uses two-indexed variables is developed and strengthened. Finally, Section 5 contains the formulation with covering variables together with its improvements. The description and the results of the computational experiments carried out are given in Section 6 and the paper is completed with some conclusions and remarks exposed in Section 7.

2 Problem definition, notation and NP-hardness

Let $A = \{1, \ldots, n\}$ be a given set of sites (points) which represent users and also candidate sites for locating the centers. For each pair (i, j), $i, j \in A$, let $d_{ij} \geqslant 0$ be the distance (cost) from i to j, which must satisfy $d_{ii} = 0$, $i \in A$, $d_{ij} > 0$, $i, j \in A$, $i \neq j$, and $d_{ij} \leqslant d_{ik} + d_{kj}$, $i, j, k \in A$. Finally, let $p \geqslant 2$ be the number of centers to install among the n possibilities.

The *p-next center problem* is defined as

$$\min_{\substack{Q \subset A \\ |Q| = p}} \max_{i \in A} \left\{ \min_{j \in Q} \{d_{ij}\} + \min_{j' \in \arg\min_{j \in Q}\{d_{ij}\}} \min_{\substack{k \in Q \\ k \neq j'}} \{d_{j'k}\} \right\},$$

i.e., to choose p elements of A so as to minimize the maximum, for all sites $i \in A$, of the sum of two distances, the distance between point i and its closest center(s) and the distance between the best closest center and its closest center. Throughout the paper we will use the notation $\delta_{ijk} := d_{ij} + d_{jk}$, $i, j, k \in A$, to denote the *total assignment cost* of site i if j is its reference center and k is its backup center.

From a practitioner point of view, we are modeling the case where a person does not know beforehand that her reference center is not available but, in case of tie among several centers, this person will attend the one from which going to another center implies covering the smallest distance. Note that if it were not forced that the reference center for a site must be among its closest centers, solutions where the user goes first to a center which is not the closest one can give better objective values. This can be clearly seen in the following example.
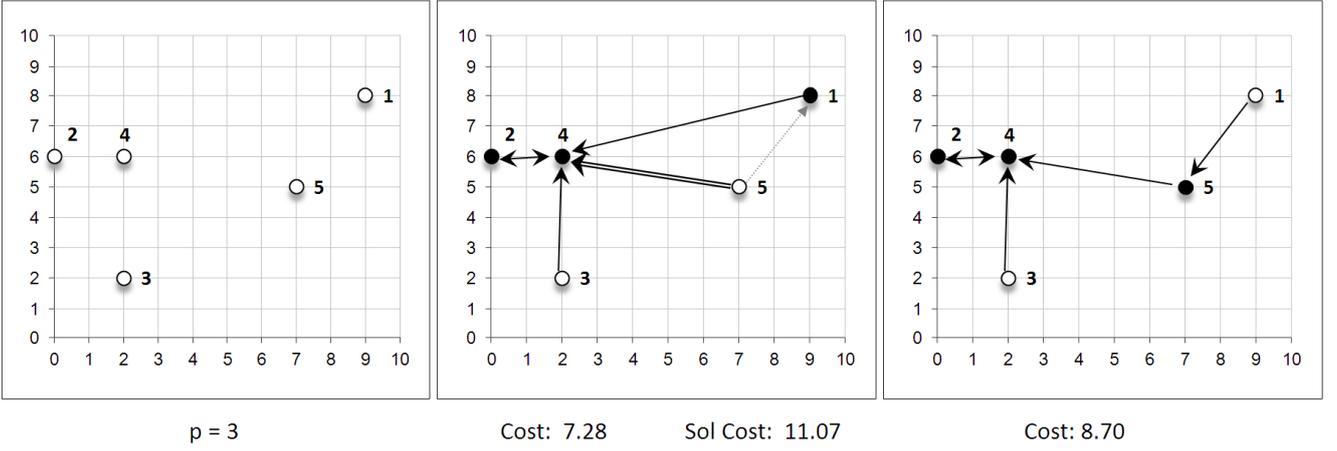
Figure 1: Problem solutions

Example 2.1 *Consider the problem defined by the five points in Figure 1, with Euclidean distances, and $p = 3$. If sites are not forced to choose their closest center as their reference center, then the optimal solution is the one given in the second sub-figure, with cost equal to 7.28, given by the total assignment cost of site 1 $(1 - 1 - 4)$. However, in this solution site 5 is using center 4 as its reference center whilst center 1 is much closer to it. Thus, if in this solution we force site 5 to visit first its closest center, then 4 will be its backup center giving rise to a solution cost of 11.07. In this case, it is optimal to locate a center at 5 instead of 1 (see the last sub-figure), and the solution cost 8.70 is given by site 1 which has 5 as its reference center, and 4 as its backup center.*

Note also that the solutions of this problem can be substantially different from the solutions of pCP*. In this example, if the optimal solution of* pNCP*, given by sites 2, 4 and 5, is seen as a p-center solution, it has an associated cost of 4, given by the assignment cost of site 3, while the optimal solution of* pCP *is given by sites 3, 4 and 5 and has a cost of 3.61 given by the assignment of site 1 to center 5.*

In what follows we show the complexity of pNCP.

Proposition 2.1 pNCP *is an NP-hard problem.*

Proof:

The proof follows by reduction from the Euclidean p-center problem EpCP (proven to be NP-hard in [26]) to pNCP. To this end, consider an instance of the decision version of EpCP given by n points in the plane with coordinates (x_i, y_i), $i = 1, \ldots, n$, a given number of centers q, Euclidean distances between the points, and a real $\alpha > 0$. We can assume without loss of generality that all Euclidean distances between pairs of points are different.

Let $0 < \varepsilon \leq \frac{1}{2} \min_{\substack{i,j=1 \\ i \neq j}}^{n} \{d_{ij}\}$, which can be computed in $O(n^2)$ time. We shall prove that there is a solution to EpCP with value less than or equal to α if and only if there is a solution to a specific pNCP instance with value less than or equal to $\alpha + \varepsilon$.

We build an associated instance of pNCP with $|A| = 2n$ and $p = 2q$, considering the n points (x_i, y_i) plus n *twin* points (x_{i+n}, y_{i+n}) with $x_{i+n} = x_i + \varepsilon$, $y_{i+n} = y_i$, $i = 1, \ldots, n$. Costs d_{ij} are given by Euclidean distances between points.

Consider any optimal solution to the second instance, represented by a set of $p = 2q$ centers C. Since each of the points in C is its own reference center and has as backup center the closest point in C (other than itself), we can build a directed network $N := (G, \ell) = (C, A_C, \ell)$ with $(u, v) \in A_C$ if v is the backup center of u and ℓ_{uv} the Euclidean distance between centers u and v.

If G contains a circuit with more than two arcs, say (u_1, u_2), (u_2, u_3),..., (u_{t-1}, u_t), (u_t, u_1), it must hold

$$\ell_{u_t u_1} \geq \ell_{u_1 u_2} \geq \ell_{u_2 u_3} \geq \cdots \geq \ell_{u_{t-1} u_t} \geq \ell_{u_t u_1}$$

implying

$$\ell_{u_t u_1} = \ell_{u_1 u_2} = \ell_{u_2 u_3} = \cdots = \ell_{u_{t-1} u_t} = \ell_{u_t u_1}.$$

In such a case the circuit can be destroyed by changing the backup center of u_1 to u_t, for instance, getting a new solution with the same value. Therefore, we can assume that all circuits in G have length 2.

Let now assume that there are points in network N with in-degree 0. Note that none of their twin points belongs to C since, otherwise, they would be part of a circuit of length two. For clarity we take these points to be (x_1, y_1), ..., (x_k, y_k), that is to say, none of the k first points are backup centers of any other center. We also assume that the backup centers of these points are (x_{k+1}, y_{k+1}), ..., (x_{k+r}, y_{k+r}) with $r \leq k$. Then, a different solution to this instance of pNCP is obtained by removing (x_i, y_i) $i = 1, \ldots, k$ from C to get the set of centers C', making $C = C' \cup \{(x_{i+k+n}, y_{i+k+n}), \ i = 1, \ldots, r\}$ and re-assigning the centers to their closest backup centers. If this change produces new points in N with in-degree 0, the transformation is repeated until the network reduces to a collection of circuits of length 2. Finally, if some of these circuits is not spanned by two twin points, we remove one of the points from C and add the twin point of the other one to C to obtain a network spanned by pairs of twin points.

Obviously, the assignment cost of any of the removed centers in the new solution does not increase in more than ε with respect to the initial solution, since now either (x_{i+k}, y_{i+k}) or (x_{i+k+n}, y_{i+k+n}) for some $i = 1, \ldots, r$ will be their backup center. Assignment of other points that are not centers, due to the triangle inequality, will neither increase in more than ε.

If, after the transformations, the network contains less than p centers (but in any case an even number of them), by adding pairs of twin centers to it the solution will never worsen.

If this last solution to pNCP is at most $\alpha + \varepsilon$ then we can build a solution to the original instance of pCP given by one center from each pair of twins and with objective value less than or equal to α, and vice versa. Q.E.D.

3 A three-indexed formulation using path variables

Two-stage location problems (see, e.g., [21]) and hub location problems ([8], [9]), among others, have been formulated by means of binary variables associated with the possible routes from each origin through the plants/hubs to the final destination. In our case, a route is given by three elements: site i, reference center j and backup center k. Then, for our first formulation we introduce binary variables

- y_j: to indicate whether there is a center in site $j \in A$,

- x_{ijk}: to indicate whether the reference center of i is j and the backup center of i (closest facility to j that is not j) is k (these variables are only defined for $i, j, k \in A$ with $j \neq k$, $i \neq k$ and $d_{ij} \leqslant d_{ik}$),

and the auxiliary continuous variable z, to account for the largest total assignment cost.

With these variables the problem is formulated as

$$\text{(F1)} \quad \min \quad z$$

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

$$\sum_{j \in A} \sum_{\substack{k \in A, k \neq i,j \\ d_{ij} \leqslant d_{ik}}} x_{ijk} = 1 \qquad i \in A, \tag{2}$$

$$\sum_{\substack{k \in A, k \neq i,j \\ d_{ik} \geqslant d_{ij}}} x_{ijk} + \sum_{\substack{k \in A, k \neq j \\ d_{ik} \leqslant d_{ij}}} x_{ikj} \leqslant y_j \qquad i, j \in A, \tag{3}$$

$$y_j + \sum_{\substack{h \in A \\ d_{ih} > d_{ij}}} \sum_{\substack{k \neq i,h \\ d_{ik} \geqslant d_{ih}}} x_{ihk} \leqslant 1 \qquad i, j \in A, i \neq j, \tag{4}$$

$$z \geqslant \sum_{j \in A} \sum_{\substack{k \in A, k \neq i,j \\ d_{ij} \leqslant d_{ik}}} \delta_{ijk} x_{ijk} \qquad i \in A, \tag{5}$$

$$y_j \in \{0, 1\} \qquad\qquad\qquad j \in A, \tag{6}$$

$$x_{ijk} \in \{0, 1\} \qquad\qquad i, j, k \in A, k \neq i, j, \ d_{ij} \leqslant d_{ik}. \tag{7}$$

Constraint (1) ensures that exactly p centers are set. Constraints (2) force each site to be assigned a two-leg trip, and (3) ensure that those paths go through two sites where centers have been set. Constraints (4) force each site to have its reference center among the closest ones. To ensure this, (4) forbid assigning site i to a center that is further from it than site j, if j is a center. Finally, constraints (5) are used to identify the length of the longest path.

Some comments on (F1) follow.

- It is not needed to require explicitly in the formulation that the backup center of a given site is the closest center with respect to the reference center, since there is always an optimal solution satisfying it (p-center problem formulations also benefit of a similar property).

- Constraints forcing that the reference center of each site be its closest center need to be explicitly included. Otherwise, further reference centers could be preferred, if they allow for cheaper backup assignments (see Example 2.1). In this formulation this is done by means of Closest Assignment Constraints (CAC) (4). Recent studies of CAC for standard discrete location formulations can be consulted in [16, 22].

  Actually, constraints (4) can be replaced by the stronger constraints

$$y_j + \sum_{\substack{k \in A \\ d_{ik} > d_{ij}}} \sum_{\substack{h \neq i,k \\ d_{ik} \geqslant d_{ih}}} x_{ikh} + \sum_{\substack{k \in A \\ d_{ik} < d_{ij}}} \sum_{\substack{h \in A, h \neq i,k \\ d_{kh} > d_{kj}}} x_{ikh} \leqslant 1 \qquad i, j \in A, i \neq j. \tag{8}$$

  The new term added in the constraints forbids the second leg of the trip from each site i being longer than the distance from its reference center to site j, if j is a center. Note that the variables in this term correspond to assignments of i to centers that are closer to it than j, so that none of those variables can take value 1 if any of the variables of the original term takes value 1. Figure 2 shows the two situations that are forbidden through constraints (8). The situation in subfigure a) was already forbidden by the original constraints, but by adding the new term we also forbid the situation depicted in subfigure b).

- Note that if constraints (7) are relaxed to $0 \leqslant x_{ijk} \leqslant 1$ variables will still take integer values in some optimal solution to the problem.
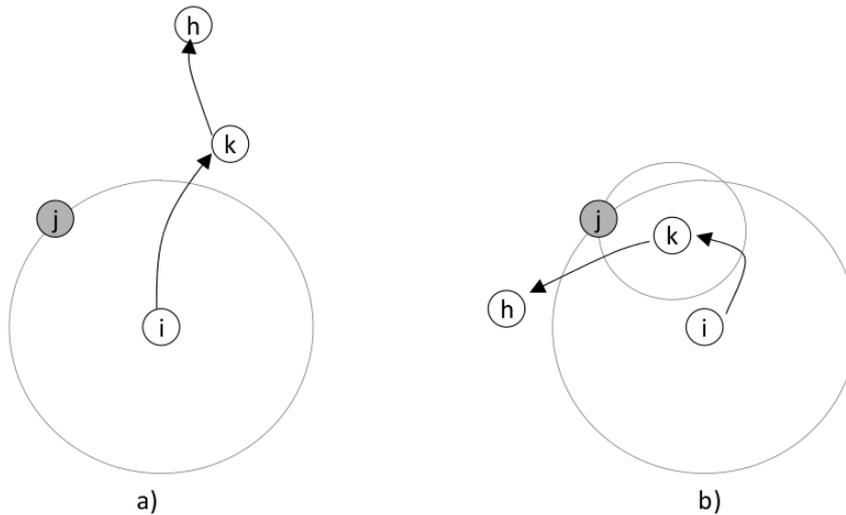
6

Figure 2: Inequalities (8): the drawn paths cannot be used if j is a center

- If costs take only integer values, then z will also take an integer value in the optimal solutions.

- If an upper bound z_{UB} on the optimal value is known, it can be used to fix variables to zero in this formulation. Indeed, whenever $\delta_{ijk} > z_{UB}$, it is sure that variable x_{ijk} will not be equal to 1 in any optimal solution, and can be thus removed from the formulation.

It is clear that, when a site i has its reference center in j, center j will be its own reference center. Then, the following result follows.

Proposition 3.1 *There exists an optimal solution to* (F1) *for which*

$$x_{ijk} \leqslant x_{jjk} \qquad \text{for all } i, j, k \in A, i \neq j \neq k, \ d_{ij} \leqslant d_{ik}. \tag{9}$$

Inequalities (9) might be incorporated to formulation (F1). However, including them all in the original formulation is rather inefficient, since this set of constraints has size $\mathrm{O}(n^3)$. We have considered the possibility of using these inequalities as optimality cuts.

4   A two-indexed formulation

Usually, three-indexed variables have produced good bounds in discrete location at the expenses of making the formulations too large and thus useless for medium sized instances. As previously observed in the literature of two-stage location problems, two-indexed formulations have a much smaller size and can be adequate to formulate larger instances. For this reason, in this section we present such a formulation, taking into account that the objective function is obtained from the sum of the two distances which join each site to its backup center, and also the specificities of how sites are allocated to both centers in the path. To this end, we have defined our first set of two-indexed variables in a very particular way. We will use the following binary variables:

- y_j: to indicate whether there is a center in location $j \in A$,

- x_{ij}: to indicate whether the closest center to i that is different from i is j (these variables are only defined for $i, j \in A$ with $j \neq i$),

and variable z with the same purpose as before.

Note that $x_{ij} = 1$ has a different meaning depending on whether i is a center or not. If i is not a center, $x_{ij} = 1$ implies that j is the reference center for i. Otherwise, if i is a center itself,

7

$x_{ij} = 1$ means that j is the backup center for i, and, therefore, for all sites having i as the reference center. Using these variables, and using the same idea as in Proposition 3.1, we have built the following formulation:

$$
\begin{aligned}
\text{(F2)} \qquad \min \quad & z \\
\text{s.t.} \quad & (1), (6) \\
& \sum_{\substack{j \in A \\ j \neq i}} x_{ij} = 1 & i \in A, & \qquad (10) \\
& x_{ij} \leqslant y_j & i, j \in A, i \neq j, & \qquad (11) \\
& y_j + \sum_{\substack{k \in A \\ d_{ik} > d_{ij}}} x_{ik} \leqslant 1 & i, j \in A, i \neq j, & \qquad (12) \\
& z \geqslant \sum_{\substack{k \in A \\ k \neq j}} d_{jk} x_{jk} & j \in A, & \qquad (13) \\
& z \geqslant d_{ij}(x_{ij} - y_i) + \sum_{\substack{k \in A \\ k \neq j}} d_{jk} x_{jk} & i, j \in A, i \neq j, & \qquad (14) \\
& x_{ij} \in \{0, 1\} & i, j \in A, i \neq j. & \qquad (15)
\end{aligned}
$$

Constraints (10) and (11) ensure that each point i is assigned to a center j different from i. Constraints (12) are CAC based on those proposed in [33] and, as in the previous formulation, they forbid the type of situations depicted in Figure 2, a). The main difference with respect to [33] is that the case $i = j$ must be removed, since in this case (12) would prevent against the allocation of i to its backup center.

To force z to take the correct value, we use constraints (13) and (14). With the first set, we make sure that z is larger than the travel cost of any site where a center is set. In this case, the first level assignment is the site itself (at a 0 cost) and the second assignment is to the closest center different from itself, which can be identified through the x-variables. In the second set of constraints, we use the fact that, according to Proposition 3.1, there is an optimal solution where all the sites that are primarily assigned to a given center j will have the same backup center, which coincides with the backup of j itself. Thus, if no center is set at site i and its reference center is j, its total assignment cost will be equal to the distance from i to j plus the distance from j to j's closest center, given by the sum in the second term of the inequality. Note that, in any other case, $x_{ij} - y_i \leqslant 0$ and the resulting constraint is weaker than the constraint in set (13) for index j.

Some comments on this formulation are the following.

- Removing CAC constraints (12) may result in general in solutions that do not fulfill the closest assignment requirement.

- In this formulation, relaxing constraints (15) to $0 \leqslant x_{ij} \leqslant 1$ may lead to fractional solutions, since constraint (13) may not be active in the optimal solution. However, relaxing constraints (6) does not have any effect on the set of feasible solutions of the problem since the binary character of y variables is enforced by (15) together with (11).

- Again, note that if distances only take integer values, then z will also take an integer value in the optimal solutions.

- Similarly to (F1), some x-variables can be fixed to zero in (F2) as follows:

   – If z_{UB} is an upper bound on the optimal value of (F2), then variable x_{ij} can be fixed to zero if $d_{ij} > z_{UB}$.

– For each site i, sort all locations: $j_1, j_2,\ldots, j_n = i$ so that $d_{ij_1} \geqslant d_{ij_2} \geqslant \cdots \geqslant d_{ij_n} = 0$. Then, all variables x_{ik} with $d_{ik} > d_{ij_{p-1}}$ can be fixed to zero since p centers will be located and, thus, at least $p-1$ of them will be established at sites different from i.

- The following CAC adapted from the literature can be also added as valid inequalities for (F2):

$$y_i - p \sum_{\substack{k \in A \\ d_{ik} > d_{ij}}} x_{ik} + \sum_{\substack{k \in A \\ d_{ik} > d_{ij}}} y_k \;\; \geqslant \;\; 0 \quad i,j \in A, i \neq j, \tag{16}$$

$$q_{ij} \sum_{\substack{a \in A \\ d_{ia} > d_{ij}}} x_{ia} + \sum_{\substack{a \in A \\ d_{ia} \leqslant d_{ij}}} y_a \;\; \leqslant \;\; q_{ij} + y_i \quad i,j \in A, \tag{17}$$

where $q_{ij} = \min\{p, |\{a \neq i : d_{ia} \leqslant d_{ij}\}|\}$.

To see how (16) works, assume there exists a site k such that $d_{ik} > d_{ij}$ with $x_{ik} = 1$. This means site i is allocated to a center further from i than j, and then all p centers (apart from i itself) must be also further from i than j. In the case of constraints (17), if a center exists whose distance to i (which is not a center itself) does not exceed d_{ij}, i cannot be allocated to any center at a distance greater than d_{ij}. In any case, the left hand side of (17) is bounded by $p + y_i$, but q_{ij} also takes into account the case in which there are less than p elements in the set $\{a \neq i : d_{ia} \leqslant d_{ij}\}$.

A second formulation with two-indexed variables, named (F3), is presented in the Appendix. In (F3) we define the objective function in terms of the location variables only, and assignment variables need not be considered. Preliminary computational results showed that none of the versions of this formulation was efficient in terms of computational times.

5 A formulation using covering Variables

Covering variables have been successfully used in discrete location problems, see e.g. [24, 25, 29]. In particular, a formulation based on these variables was used in [14] for the p-center problem, providing very good lower bounds. Introducing covering variables in our model has a greater degree of difficulty, since closest allocation to the reference center must be guaranteed. In this section we give the details of such a formulation and several possibilities of improvement.

First of all, some previous operations with the data are needed. In this case we sort the values δ_{ijk}, for those triplets (i,j,k) with $i,j,k \in A$, $k \neq i,j$ such that $d_{ij} \leqslant d_{ik}$, in increasing order and ignoring ties. Let

$$\Delta_1 < \Delta_2 < \ldots < \Delta_h$$

denote the strictly increasing sequence of such values, H be the set $\{1,\ldots,h\}$, $H(i,j)$ denote the set of indices $\ell \in H$ corresponding to $\delta_{ij\cdot}$ values and $H(i)$ denote the set of indices $\ell \in H$ corresponding to $\delta_{i\cdot\cdot}$ values.

We need the following binary variables:

- y_j: to indicate whether there is a center in site $j \in A$,

- y_{jk}: to indicate whether two centers are set in locations j and k (these variables are only defined for $j < k$). Note that variable y_{jk} can be seen as the product $y_j \times y_k$.

- z_ℓ: to indicate whether the assignment cost of some site is at least Δ_ℓ, $\ell \in H$.

The proposed formulation is

$$\text{(F4)} \quad \min \quad \Delta_1 z_1 + \sum_{\ell=2}^{h} (\Delta_\ell - \Delta_{\ell-1}) z_\ell \tag{18}$$

$$\text{s.t.} \quad (1), (6)$$

$$z_\ell + \sum_{\substack{a \in A \\ d_{ia} \leqslant d_{ij}}} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ \delta_{iab} < \Delta_\ell}} y_{a'b'} \geqslant y_j \quad i, j \in A, \ \ell \in H(i,j), \tag{19}$$

$$z_\ell \geqslant z_{\ell+1} \qquad\qquad\qquad \ell = 1, \ldots, h-1, \tag{20}$$

$$y_{jk} \leqslant y_j \qquad\qquad\qquad j, k \in A, j < k, \tag{21}$$

$$y_{jk} \leqslant y_k \qquad\qquad\qquad j, k \in A, j < k, \tag{22}$$

$$y_{jk} \in \{0,1\} \qquad\qquad\quad j, k \in A, j < k. \tag{23}$$

$$z_\ell \in \{0,1\} \qquad\qquad\quad \ell \in H. \tag{24}$$

For brevity, in (19), a' has been taken to represent $\min\{a,b\}$, and b' to represent $\max\{a,b\}$. Here, the objective function (18) measures the maximum assignment cost of a site, given by the Δ-value associated to the last z-variable taking value 1. To force that the primary assignment of site i is made to its closest center, we use constraints (19). Each such constraint forces a z-variable to take value 1 if a center is established at site j and there is no pair (a,b) of centers with a not further from i than j, a closer to i than b and giving a total cost δ_{iab} strictly smaller than Δ_ℓ. Finally, constraints (21) and (22) force variables to take consistent values.

As in previous formulations we next collect some useful observations:

- One can derive from [31] that the set of binary solutions to $\sum_j y_j = p$ and $y_{jk} = y_j y_k$, $k > j$ coincides with the set of binary solutions to (21), (22), $y_{jk} \geqslant y_j + y_k - 1$, $k > j$ and

$$\sum_{k>j} y_{jk} + \sum_{k<j} y_{kj} = (p-1) y_j, \quad j \in A. \tag{25}$$

  Due to the form of our formulation, constraints $y_{jk} \geqslant y_j + y_k - 1$ are always satisfied in any optimal solution and we do not need to impose them explicitly. Constraints (25) will be added to the formulation in order to get better lower bounds.

- Based on the above, in this formulation only the $y_.$ variables are required to be binary to enforce $y_{..}$ to be binary as well.

- Additionally, even if constraints (24) are removed from the formulation, z-variables will take binary values in any optimal solution.

- The following are valid inequalities for (F4):

$$z_\ell + \sum_{\substack{a \in A}} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ \delta_{iab} < \Delta_\ell}} y_{a'b'} \geqslant 1 \quad i \in A, \ \ell \in H(i). \tag{26}$$

  Each constraint in family (26) forces a z-variable to take value 1 if all the y-variables in the constraint take value 0, i.e., if there is not any pair of centers (a,b) with a closer to i than b and with a total assignment cost δ_{iab} less than Δ_ℓ. However, these constraints cannot be used instead of (19) since they do not imply closest allocation.

On the other hand, as it happened in previous formulations, (F4) can also be reinforced by using an upper bound on its optimal value, z_{UB}. Let ℓ_U be the largest index ℓ for which $\Delta_\ell \leqslant z_{UB}$. An improved version of (F4) based on this bound is

$$\text{(F4b)} \quad \min \quad \Delta_1 + \sum_{\ell=2}^{\ell_U} (\Delta_\ell - \Delta_{\ell-1}) z_\ell$$

$$\text{s.t.} \quad (1), (6), (21), (22)$$

$$z_\ell + \sum_{\substack{a \in A \\ d_{ia} \leqslant d_{ij}}} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ \delta_{iab} < \Delta_\ell}} y_{a'b'} \geqslant y_j \quad i, j \in A, \ \ell \in H(i,j) \cap \{1, \ldots, \ell_U\}, \tag{27}$$

$$\sum_{\substack{a \in A \\ d_{ia} \leqslant d_{ij}}} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ \delta_{iab} < \Delta_{\ell_U+1}}} y_{a'b'} \geqslant y_j \quad i, j \in A : H(i,j) \cap \{\ell_U + 1, \ldots, h\} \neq \emptyset, \tag{28}$$

$$\sum_{a \in A} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ \delta_{iab} < \Delta_{\ell_U+1}}} y_{a'b'} \geqslant 1 \quad i \in A, \tag{29}$$

$$z_\ell \geqslant z_{\ell+1} \quad \ell = 1, \ldots, \ell_U - 1, \tag{30}$$

$$z_\ell + \sum_{a \in A} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ \delta_{iab} < \Delta_\ell}} y_{a'b'} \geqslant 1 \quad i \in A, \ \ell = 1, \ldots, \ell_U. \tag{31}$$

Since the value of z_ℓ is fixed for $\ell \notin \{1, \ldots, \ell_U\}$, not all constraints (19) are needed. However, not all constraints from that set with indices $\ell \notin \{1, \ldots, \ell_U\}$ can be dropped. Otherwise, the assignment cost of some of the sites might not be accounted correctly. For this reason, in this case constraints (28) need to be added, to make sure that sites that might have an assignment cost larger than z_{UB} have available eligible centers that lead to smaller assignment cost and satisfy the closest assignment condition. In the same spirit, constraints (29) make sure that the set of selected centers lead to a cost smaller than the upper bound.

6 Computational experience

A series of computational experiments have been carried out to assess the usefulness of the proposed formulations to solve pNCP with standard software, and to compare their performances. These experiments and their results are reported in this section. Additionally, an analysis of the relationship between the solutions to pCP and pNCP is included in the last subsection.

6.1 Comparison of the formulations

The computational experiments carried out in order to evaluate the suitability of the different formulations to solve pNCP using standard MIP solvers are split into two parts. In a preliminary study, with a set of small instances, the best variant of each formulation has been identified. Then, the best versions of the different formulations are compared in a second part, using two sets of larger instances.

6.1.1 Preliminary study

Since the number of proposed formulations and improvements is large, a preliminary study with most of the variants on a small amount of instances (40) was carried out in first place. The

dimensions (n, p) of the instances considered in this initial experiment are depicted in Table 1. For each considered dimension (n, p), four instances were generated taking the first n points of

	$p = 5$	$p = 10$	$p = 20$
$n = 10$	✓		
$n = 20$	✓	✓	
$n = 30$	✓	✓	
$n = 40$	✓	✓	✓
$n = 50$		✓	✓

Table 1: Dimensions considered in the first experiment

each of the four well known instances `pmed1-pmed4` from the OR-library [3]. All formulations were implemented using Xpress Optimizer Version 23.01.05 on a Dell Computer with two Inter(R) Xeon(R) X5690 processors with 3.47GHz and 3.46GHz and 48 GB of RAM memory. A limit of two hours of CPU time was set.

Two alternative settings were tested for the three-indexed formulation: formulation (F1) presented in Section 3 but replacing constraints (4) by the stronger constraints (8) (for brevity, we will still denote this formulation (F1)), and the same formulation, but reinforced with valid inequalities (9), that, from now on we will denote by (F1)'. In the case of formulation (F2), we have compared the plain formulation with the formulation reinforced with valid inequalities (16) and (17) (from now on, (F2)'). Regarding the formulations presented in Section 5 (the formulation using covering variables) we have always used in our computational experiments its improved form (F4b) and considered two settings: the plain formulation (F4b) with upper bound, and the reinforcement of (F4b) which incorporates constraints (25), that we will denote by (F4b)'. Previous

Name	Formulation	
(F1)	min	z
	s.t.	$(1), (2), (3), (5), (6), (8)$
(F1)'	min	z
	s.t.	$(1), (2), (3), (5), (6), (8), (9)$
(F2)	min	z
	s.t.	$(1), (6), (10), (11), (12), (13), (14), (15)$
(F2)'	min	z
	s.t.	$(1), (6), (10), (11), (12), (13), (14), (15), (16), (17)$
(F4b)	min	$\Delta_1 + \sum_{\ell=2}^{\ell_U}(\Delta_\ell - \Delta_{\ell-1})z_\ell$
	s.t.	$(1), (6), (21), (22), (27), (28), (29), (30), (31)$
(F4b)'	min	$\Delta_1 + \sum_{\ell=2}^{\ell_U}(\Delta_\ell - \Delta_{\ell-1})z_\ell$
	s.t.	$(1), (6), (21), (22), (25), (27), (28), (29), (30), (31)$

Table 2: Naming convention and constraints for each formulation

tests discouraged us from studying other alternatives. This information together with the sizes of all formulations considered are summarized in Tables 2 and 3, respectively. Regarding Table

3, the number of variables in (F1) depends on the number of ties between distances d_i, whereas parameter h depends on the ties between δ-values and is bounded by n^3.

Formulation	Variables			Linear constraints	
	Minimum	Maximum	Binary	Minimum	Maximum
(F1)	$\frac{n^3-n^2+2n+2}{2}$	n^3-2n^2+2n+1	n	$2n^2+n+1$	
(F1)'	$\frac{n^3-n^2+2n+2}{2}$	n^3-2n^2+2n+1	n	$\frac{n^3+3n^2+2n+2}{2}$	n^3+2n+1
(F2)	n^2+1		n^2-n	$3n^2-n+1$	
(F2)'	n^2+1		n^2-n	$5n^2-2n+1$	
(F4b)	$\frac{n^2-n}{2}+n+h$		n	$2n^2-n+h$	n^3+n^2-n+h
(F4b)'	$\frac{n^2-n}{2}+n+h$		n	$2n^2+h$	n^3+n^2+h

Table 3: Number of variables and constrains of all formulations

All formulations benefit from the knowledge of z_{UB}, an upper bound on the optimal value of the problem. In order to get a value for z_{UB} without excessive computational effort, five feasible solutions were generated by randomly choosing p centers. For each of these candidate initial solutions, we performed an interchange phase trying to reduce the value of the largest assignment cost. The objective value given by the best among these five solutions was used as an upper bound for the problem. Since this is a randomized procedure, to make fair comparisons of the different formulations, for each considered instance the bounding procedure was run only once, and the same upper bound was used in all the formulations. Although this upper bound may be seen as *naïve*, it is rather effective as confirmed by the computational experiments.

Most of the 40 instances could be solved within the two hours of CPU time with the majority of formulations and variants. Indeed, there are only 4 instances, all with $n = 50$, that could not be solved with some of the formulations. The percent gaps between the best solution found and

source	n	p	(F1)	(F1)'	(F4b)	(F4b)'
pmed1	50	10	-	-	17.36	-
pmed2	50	10	-	-	7.89	-
		20	1.02	1.02	1.87	2.96
pmed3	50	10	-	-	19.58	3.85

Table 4: Percent gaps at termination, for the instances not solved with all formulations

the current lower bound after the two hours of CPU time obtained in those four cases are shown in Table 4. Alternatives (F2) and (F2)' are not included in this table since they could solve all instances in much shorter times (less than 20 minutes, and than one hour, respectively). From Table 4 it can be observed that all instances but one could be optimally solved both by (F1) and (F1)'. Moreover, in this single instance, the percent gap was only slightly over 1%. In fact, none of the other 3 instances required more than 1.5 minutes with either variant. The two variants of (F4) (the formulation with covering variables) show a different behavior. When using alternative (F4b) two instances remained unsolved, while this happened in four instances with (F4b)'. The percent gaps ranged from about 2% to almost 20%. With respect to the observed gaps, the reinforced alternative (F4b)' seemed to be the best one; it only left unsolved two of the instances, and the final percent gaps never reached 4%.

Additionally, Table 5 shows the average (among the four instances of the same dimensions n and p) percent gaps between the LP bounds and the optimal values of the proposed formulations. As it can be seen the two variants of (F1) provide the smallest LP gaps, although gaps are very similar for all the formulations considered. In order to make a deeper comparison of the variants of

n	p	(F1) %LP gap	(F1)' %LP gap	(F2) %LP gap	(F2)' %LP gap	(F4b) %LP gap	(F4b)' %LP gap
10	5	20.00	19.34	26.64	26.64	21.83	21.79
20	5	25.85	25.42	34.61	34.61	31.26	30.79
	10	8.16	8.16	8.16	8.16	8.14	8.14
30	5	30.03	29.92	42.12	42.12	38.18	37.65
	10	19.81	19.42	21.21	21.21	20.43	20.43
40	5	28.29	28.28	44.56	44.56	39.49	38.44
	10	21.28	21.20	24.48	24.48	22.73	22.70
	20	2.41	2.41	2.41	2.41	2.41	2.41
	10	26.16	25.98	29.56	29.56	27.78	27.78

a) F1 alternatives b) F2 alternatives c) F4 alternatives
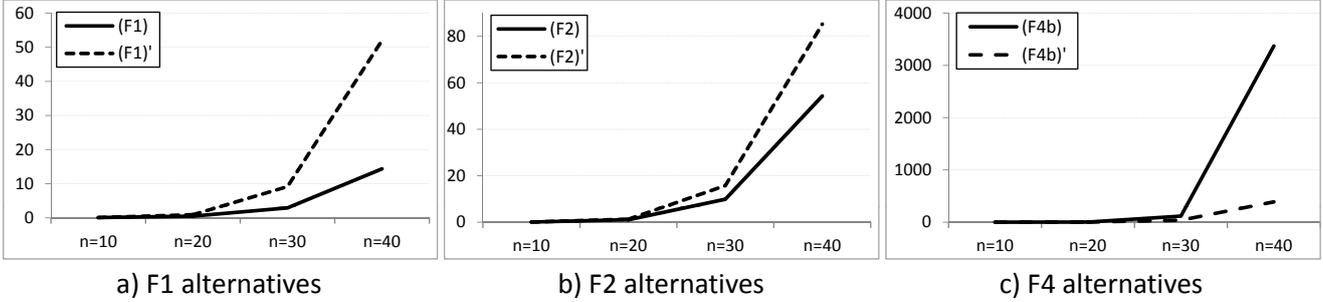
Figure 3: Average CPU times (s) for the variants of formulations (F1), (F2) and (F4)

Figure 3 shows, the plain version of formulation (F1) clearly outperforms the reinforced version, and the same happens in the case of (F2). Note that the number of additional constraints is $O(n^3)$ in the case of (F1) and $O(n^2)$ in the case of (F2). This increase of the formulation size may explain the faster growth of the time required to solve the instances by the reinforced formulations. So, in these cases, the plain version has been chosen for next experiments. Regarding sub-figure c), the observed behavior has led us to keep formulation (F4b)' for further analysis.
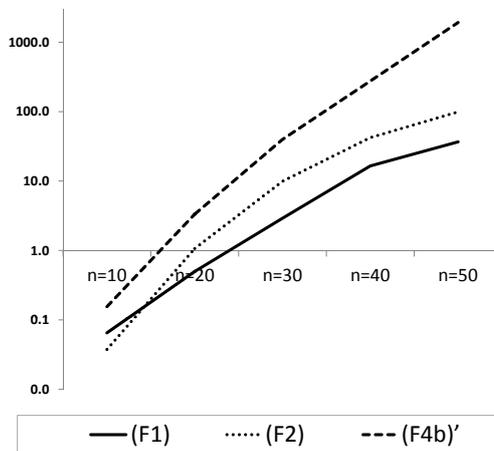

Figure 4: Average CPU times (s) for the small instances (logarithmic scale)

14

We next analyze the CPU times required to solve all the instances with the remaining for-mulations ((F1), (F2) and (F4b)'). Figure 4 displays (in logarithmic scale) the averages of these times (in seconds) taken among instances of the same dimension n. The CPU time limit 7200 has been used for the unsolved instances. Here, formulations (F1) and (F2) show their superiority over (F4b)'. Not only CPU times required to solve pNCP using formulation (F4b)' are larger, but they also tend to increase faster.

To evaluate the effect of the number of centers to locate, p, on these CPU times, we have concentrated on the instances with $n = 40$, since all p values are considered in this set. Figure 5 gathers the obtained results. This figure shows how formulations (F1) and (F2) behave similarly,
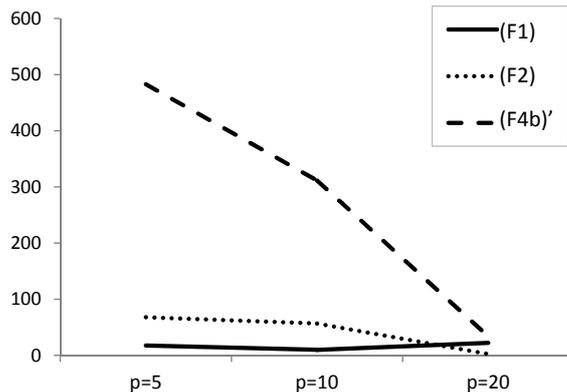

Figure 5: Effect of p on the CPU times (s) ($n = 40$)

and their computational times do not vary much as p increases. Instead, with formulation (F4b)' instances with small values of p are harder to solve. A similar effect has already been observed when using covering formulations for other discrete location problems. When p is large, the number of z-variables that are candidates to take value 1 in the optimum becomes small.

6.1.2 Analysis with larger instances

With the three remaining formulations ((F1), (F2) and (F4b)') a second experiment was carried out on larger instances, also generated from the well known p-median instances `pmed1-pmed4`. Now, we generated instances with n taken from $\{60, 70, 80, 90, 100\}$ and $p \in \{10, 20, 30\}$. Additionally, the value $p = 50$ was also considered for the instances with $n = 90$ and $n = 100$. This gave rise to a set of 68 instances. Again, a CPU time limit of two hours was set.

There were many instances that could not be solved within this time limit. Indeed, formulation (F1) allowed to solve 37 of the instances, which represents 54% of the set, whereas (F2) allowed to solve 26 (38% of the total) and only 16 instances (24% of the total) could be solved using (F4b)'. In total, the optimal solution was found for 41 of the instances. 15 instances could only be solved using (F1), for 3 instances (F2) was the only formulation that succeeded, whereas for other 7 instances the optimal solution was found by both, (F1) and (F2), but not by (F4b)'. One single instance (with $n = 60$ and $p = 20$) could be solved using (F2) and (F4b)' but not with (F1). Finally, there is a set of 15 instances with very diverse dimensions that could be solved with all three formulations. Among the instances that could not be solved with any formulation we find all the instances with $n = 90$ and $p = 20$, and all the instances with $n = 100$ and $p \leqslant 20$.

The results obtained with this set of larger instances are summarized in Table 6. Each row corresponds to a different combination of n and p. Columns under heading '#' represent the number of instances solved up to optimality. For the three formulations, the average percent gaps after two hours of computing time are depicted under heading '%gap', whereas the average computing times, in seconds, are given under heading 'time'. Notice that, here, a CPU time equal

		(F1)				(F2)				(F4b)'		
n	p	%gap	time	#	%var	%gap	time	#	%var	%gap	time	#
	10	0.0	399	4	18.3	1.3	5031	3	73.65	20.0	6292	1
60	20	1.3	3856	2	9.6	2.2	2434	3	54.96	9.4	5617	1
	30	0.0	112	4	4.1	0.0	46	4	33.72	0.0	1172	4
	10	0.0	1537	4	16.5	21.9	7200	0	74.32	33.9	7200	0
70	20	1.8	4297	2	7.5	8.5	7200	0	49.66	14.9	7200	0
	30	0.9	2388	3	4.6	2.3	1919	3	38.17	4.5	3661	2
	10	1.2	3957	3	21.1	29.1	7200	0	78.78	51.5	7200	0
80	20	7.5	5752	1	11.0	25.8	7200	0	61.57	37.5	7200	0
	30	3.5	6147	1	7.0	3.6	4834	2	46.97	13.2	7200	0
	10	2.8	5770	2	23.3	38.7	7200	0	82.71	71.9	7200	0
90	20	16.3	7200	0	12.5	28.3	7200	0	66.36	59.5	7200	0
	30	6.9	3643	2	7.6	7.8	3681	2	50.26	17.5	7200	0
	50	0.0	39	4	4.5	0.0	5	4	34.01	0.0	1471	4
	10	17.5	7200	0	31.1	49.2	7200	0	87.80	105.6	7200	0
100	20	22.3	7200	0	13.5	29.8	7200	0	69.47	77.3	7200	0
	30	8.5	5594	1	10.0	13.4	6216	1	53.85	43.0	7200	0
	50	0.0	71	4	3.6	0.0	227	4	32.38	0.0	3263	4
Overall		5.3	3833	37	12.1	15.4	4823	26	58.2	32.9	5922	16

Table 6: Average results and number of solved instances for the set of larger instances

to two hours has been attributed for the instances that could not be solved within this limit, therefore times are underestimated. Additionally, for formulations (F1) and (F2), the percentage of the variables not fixed to zero value, after applying the criteria exposed in Sections 3 and 4 is given under heading '%var'.
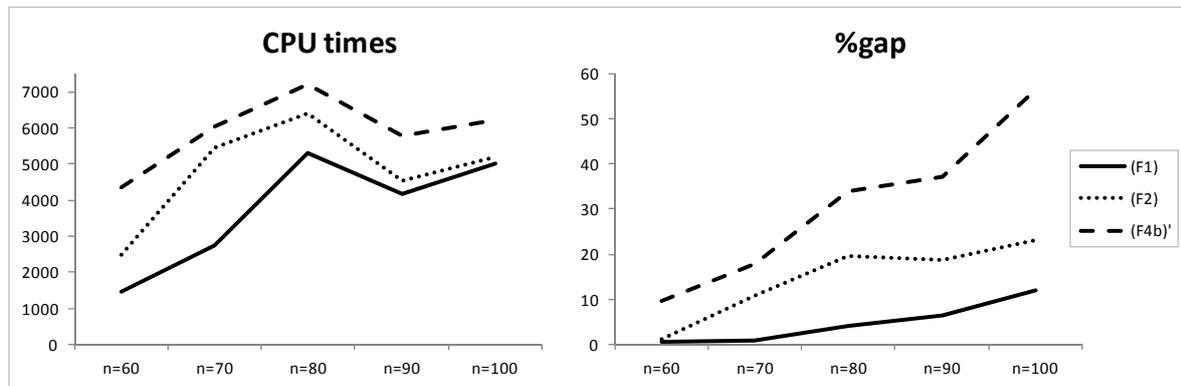

Figure 6: Summary of results for large instances

Most of the conclusions drawn in the first experiment scaled to these larger instances. For instance, (F1) and (F2) are superior to (F4b)' in terms of both, % gaps after two hours, and CPU times. This can be better appreciated in Figure 6, which displays average times and percent gaps for the different values of n. Also, the computational times for formulations (F1) and (F2) are very similar, although formulation (F1) is, again, slightly faster. This behavior might seem surprising, taking into account that formulation (F2) is much smaller than (F1) due to the number of variables. Nevertheless, as we can see in Table 6, the number of variables fixed to zero by using the criteria exposed in Sections 3 and 4 is much higher in (F1) than in (F2). This fact may partially explain the better behavior of (F1).

Table 6 also shows that, again, for a fixed dimension n formulation (F4b)' becomes more effective as p increases. This is not the case for formulations (F1) and (F2), where the effect of p on the times and percent gaps depends pretty much on the size of the instance.

n	p	#	(F1)	(F2)	(F4b)'
60	10	1	57.2	2052.6	3569.7
60	30	4	111.6	45.5	1172.5
70	30	2	19.4	8.4	121.0
90	50	4	38.6	5.0	1471.4
100	50	4	71.4	226.7	3263.2

Table 7: CPU times for large instances solved with all formulations

Recall that the results shown in Table 6 and in Figure 6 are biased, since a CPU time of two hours has been considered for all unsolved instances. To avoid this bias, in Table 7 we display the solution times, in seconds, for the 15 instances that could be solved with all formulations. Each row corresponds to a different combination of n and p. Under heading '#' we give the number of instances that could be solved by all formulations. The three next columns give the corresponding average CPU times. Now, the difference between (F4b)' and formulations (F1) and (F2) is clearer, with CPU times that can be about two orders of magnitude larger. It is also worth mentioning that all instances with $p = 50$ could be solved with all formulations, despite having 90 and 100 nodes, while no instance with 80 nodes appears in the table; i.e., each 80 nodes instance was left unsolved by at least one formulation.

On the other hand, it becomes clear that, although (F1) seems to be the most effective formulation, (F2) is very competitive, and it is the best alternative for larger instances, when the size of (F1) makes it useless. To confirm this, a third experiment was carried out on larger instances. Now we generated 24 instances from the three p-median instances pmed6, pmed7 and pmed8. We considered all combinations of $n \in \{150, 200\}$ with $p \in \{20, 30, 50, 80, 100\}$ except $(200, 20)$ and $(150, 100)$. For these experiments, a CPU time limit of four hours was established. The results

		(F1)		(F2)	
n	p	%gap	CPU	%gap	CPU
	20	28.16	14571.40	47.17	14428.40
150	30	19.36	14576.37	16.91	14422.90
	50	0	866.52	0	893.17
	80	0	271.96	0	19.37
	30	*	*	53.67	14477.90
200	50	*	*	19.03	10959.50
	80	*	*	0	376.39
	100	*	*	0	81.37

Table 8: Average results for large instances solved with (F1) and (F2)

obtained with these larger instances are summarized in Table 8. As can be seen, all the instances with $n = 200$ are intractable with formulation (F1), whereas formulation (F2) is able to solve to optimality instances with $n = 200$ and $p = 80, 100$, although for the remaining instances the gap is also large. From the above, we conclude that (F2) is the natural alternative to solve the pNCP whenever the instances have a medium to large size, i.e. $n \geqslant 150$.

6.2 Solution Analysis

In view of the complexity of the pNCP as compared with pCP, one might wonder to what extent these problems yield different solutions and it is thus worth solving it instead of pCP in some

situations. In order to visualize the differences between the solutions of pCP and pNCP we have carried out a last experiment.

For the set of instances used in Section 6.1.1 we have evaluated the optimal solution to pCP as a solution to the pNCP, and vice-versa, and in both cases we have observed large deviations from the optimum of the corresponding problem. Figure 7 summarizes the obtained results. For each instance, pC(pNC) is the deviation of the p-next center solution from the p-center optimal value, while pNC(pC) is the deviation of the p-center solution from the p-next center optimal value. Averages of these deviations have been taken among all the instances of the same size. There were no instances were both deviations were small. That is, the solutions of these problems
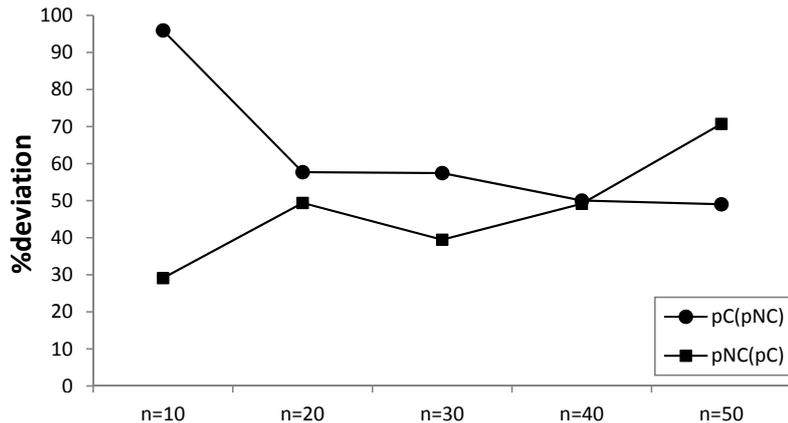
Figure 7: Comparison of p-center and p-next center: Percent deviations

are always quite different. In general terms, we observed deviations of similar magnitudes in both cases, with average values of about 50%. However, the two deviations corresponding to the same instance were most often very different. Indeed, there is a weak, albeit statistically significant, negative correlation between both deviations; in the instances where one of them is relatively small, the other one is quite large. As can be seen in the figure, the deviation of the p-next center solution from the p-center optimal value tends to decrease for larger instances, while the deviation of the p-center solution from the p-next center optimal value increases.

As mentioned in the introduction, failure probabilities are most often used in facility location models where facilities are considered to be prone to failure. This is not the case of the pNCP. However, these probabilities can be incorporated in a straightforward way in any of the proposed formulations. While the pNCP aims at minimizing the largest service distance in case there is one facility failure, one might decide to minimize the expected largest service distance when the failure probabilities are known. Assume all these failure probabilities all equal to q. Then, the expected service distance of customer i having j and k as its reference and backup centers would be $d_{ij} + q d_{jk}$. So, this extension of the pNCP can be directly tackled with the previous formulations just by redefining $\delta_{ijk} = d_{ij} + q d_{jk}$ and adding a factor q in the last term of constraints (14). Note that, for $q = 0$ and $q = 1$ we find as particular cases of this extension pCP and pNCP, respectively. Therefore, solving a series of instances of this new problem with increasing values of q, ranging from 0 to 1, we can obtain a sequence of solutions that evolve from the solution to pCP to the solution to pNCP. Figure 8 shows this sequence of solutions for an instance with $n = 20$, $p = 4$ and Euclidean distances. As can be seen, as q increases, the centers tend to be located in pairs, so that each center has a close backup. This same behavior has been observed in different instances.

This extension provides a suitable way to find tradeoff solutions that balance the two objectives: having small service costs when the system operates in regular conditions, and providing a reasonable service if one facility fails. The analysis of the formulations that has been carried out in the previous subsections applies also to this case.
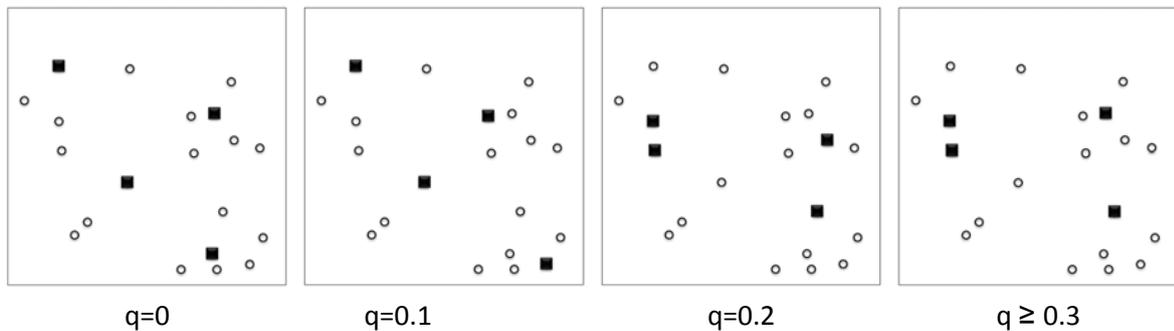
18

Figure 8: Evolution of the solution with q

7 Conclusions

This paper introduces a new location problem, namely the p-next center problem, which can be considered as a new attempt to handle humanitarian logistics. For this problem, different formulations are proposed. In order to evaluate their utility to solve this new problem, we have compared the formulations after some strengthening and preprocessing. To this end, we have solved several sets of instances using all formulations on an off the shelf solver. As a result of that comparison, we can compare the suitability of the different formulations and observe that we can systematically solve instances with up to 50 nodes within a reasonable computational effort.

The findings of this paper can be the basis of further research on this interesting problem. From a modeling point of view, it would be of interest to consider the existence of capacity limits in the facilities (at least in the backup facilities) since in real life systems work with limited capacity. In case of failure, it is not realistic to assume that the backup facility has enough spare capacity to serve all the demand previously served by a disrupted facility. The introduction of capacity constraints is possible at the price of some modifications in the formulations. In particular, closest assignment constraints would be needed as well to ensure closest allocations to the backup facilities whereas in the original models these constraints were only necessary for the first level allocations. Another possibility may be to relax the closest assignment hypothesis which would result in a different type of models. From a technical point of view, it would be also very interesting to address the separation of valid inequalities for the different formulations inside a branch-and-cut framework. Indeed, improving the lower bounds at a low computational cost is needed to be capable of solving this problem much faster and thus, be able to solve larger instances.

Acknowledgements

Appendix

A second formulation with two-indexed variables and several reinforcements are presented here. All versions of this formulation behaved badly in terms of computational times. Nevertheless, it has a certain theoretical interest and could be further studied to take advantage of its reduced

size.

We define the following binary variables

- y_j: to indicate whether a center is set in location $j \in A$,

- y_{jk}: to indicate whether there are two centers in sites $j, k \in A$, $k > j$,

that we will use again together with the auxiliary continuous variable z.

The corresponding formulation is:

$$(\text{F3}) \quad \min \quad z$$

$$\text{s.t.} \quad (1), (6), (21), (22), (23)$$

$$z + \sum_{\substack{a \in A \\ d_{ia} < d_{ij}}} \delta_{ijk} y_a + \sum_{\substack{b \in A \setminus \{j\} \\ d_{ib} > d_{ij} \\ d_{jb} < d_{jk}}} \delta_{ijk} y_b +$$

$$\sum_{\substack{a \in A \setminus \{j\} \\ d_{ia} = d_{ij}}} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ d_{ab} < d_{jk}}} \delta_{ijk} y_{a'b'} \geqslant \delta_{ijk} y_j \qquad i, j, k \in A, k \neq j, d_{ij} \leqslant d_{ik}, \qquad (32)$$

Constraints (32) force variable z to take the required value in the objective function. Like in (F4), a' has been taken to represent $\min\{a, b\}$, and b' to represent $\max\{a, b\}$. Given a site i and two sites j and k that are eligible for being, respectively, the first and second assignment of i, constraint (32) is non trivial only if a center is set at site j ($y_j = 1$). In this case, the constraint forces z to take at least the value δ_{ijk} that would correspond to i being assigned to j and having k as its backup center, unless any of the following occurs:

1. A center is set at some site a that is closer to i than j.

2. A center is set at some site b that is closer to j than k.

3. Two centers are set at sites (a, b) that allow a first assignment of i as close as j and a second assignment to b giving rise to a total assignment cost smaller than δ_{ijk}.

Finally, constraints (21) and (22) force variables to take consistent values.

Some comments on this formulation are the following:

- As in (F4), the shape of (F3) guarantees the satisfaction of the constraints $y_{jk} \geqslant y_j + y_k - 1$ in any optimal solution, making it possible to remove it.

- As it happens in formulation (F4), forcing variables $y_.$ to be binary implicitly forces variables $y_{..}$ to be binary as well, so that constraints (23) can be dropped from the formulation.

- If distances take only integer values, then z will also take an integer value in the optimal solution.

Note that variables y_{jk} with $j \neq k$ only appear in one term of constraints (32) that is required to cope correctly with situations where different centers can be set at the same distance from a given site i, and in the linking constraints (21) and (22) required to force them to take values consistent with those of variables y_j. Thus, if there are no ties in the distances from any site i to

the others, this formulation can be sensibly reduced. Indeed, variables y_{jk} are no longer needed, and the resulting formulation is

$$(\text{F3r}) \quad \min \quad z$$
$$\text{s.t.} \quad (1), (6)$$
$$z + \sum_{\substack{a \in A \\ d_{ia} < d_{ij}}} \delta_{ijk} y_a + \sum_{\substack{b \in A \setminus \{j\} \\ d_{ib} > d_{ij} \\ d_{jb} < d_{jk}}} \delta_{ijk} y_b \geqslant \delta_{ijk} y_j \quad i, j, k \in A; k \neq j, d_{ij} \leqslant d_{ik}.$$

The above two formulations for pNCP (the one with and the one without ties) can benefit from the knowledge of and upper bound z_{UB}. An improvement of (F3) is

$$(\text{F3b}) \quad \min \quad z$$
$$\text{s.t.} \quad (1), (6), (21), (22)$$
$$z + \sum_{\substack{a \in A \\ d_{ia} < d_{ij}}} \delta_{ijk} y_a + \sum_{\substack{b \in A \setminus \{j\} \\ d_{ib} > d_{ij} \\ d_{jb} < d_{jk}}} \delta_{ijk} y_b \sum_{\substack{a \in A \setminus \{j\} \\ d_{ia} = d_{ij}}} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ d_{ab} < d_{jk}}} \delta_{ijk} y_{a'b'} \geqslant \delta_{ijk} y_j \quad \begin{array}{l} i,j,k \in A; k \neq j, \ d_{ij} \leqslant d_{ik}, \\ \delta_{ijk} \leqslant z_{UB}, \end{array} \quad (33)$$

$$\sum_{\substack{a \in A \\ d_{ia} < d_{ij}}} y_a + \sum_{\substack{b \in A \setminus \{j\} \\ d_{ib} > d_{ij} \\ d_{jb} < d_{jk}}} y_b + \sum_{\substack{a \in A \\ d_{ia} = d_{ij}}} \sum_{\substack{b \in A \setminus \{a\} \\ d_{ia} \leqslant d_{ib} \\ d_{ab} < d_{jk}}} y_{a'b'} \geqslant y_j \quad \begin{array}{l} i,j,k \in A, \ k \neq j, \ d_{ij} \leqslant d_{ik}, \\ \delta_{ijk} > z_{UB}. \end{array} \quad (34)$$

Constraints (34) have to be incorporated to the formulation since there may be sites i with no constraint in family (33) associated, which would lead to an underestimation of the optimal value of the problem.

In those cases where formulation (F3) can be reduced to (F3r), i.e., when ties in the distances do not exist, a similar use of upper bounds can be carried out giving rise to the following formulation:

$$(\text{F3rb}) \quad \min \quad z$$
$$\text{s.t.} \quad (1), (6)$$
$$z + \sum_{\substack{a \in A \\ d_{ia} < d_{ij}}} \delta_{ijk} y_a + \sum_{\substack{b \in A \setminus \{j\} \\ d_{ib} > d_{ij} \\ d_{jb} < d_{jk}}} \delta_{ijk} y_b \geqslant \delta_{ijk} y_j \quad \begin{array}{l} i,j,k \in A; k \neq j, d_{ij} \leqslant d_{ik} \\ \delta_{ijk} \leqslant z_{UB}, \end{array}$$
$$\sum_{\substack{a \in A \\ d_{ia} < d_{ij}}} y_a + \sum_{\substack{b \in A \setminus \{j\} \\ d_{jb} < d_{jk}}} y_b \geqslant y_j \quad \begin{array}{l} i,j,k \in A; k \neq j, \\ d_{ij} \leqslant d_{ik}, \delta_{ijk} > z_{UB}. \end{array}$$

References

[1] Albareda-Sambola A, Díaz JA, Fernández E. Lagrangean duals and exact solution to the capacitated p-center problem. European Journal of Operational Research 2010;201:71-81.

[2] Bar-Ilan J, Kortsarz G, Peleg D. How to allocate network centers. Journal of Algorithms 1993;15:385-415.

[3] Beasly JE. OR-library. `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`

[4] Benkoczi R, Bhattacharya BK, Tamir A. Collection depots facility location problems in trees. Networks 2009;53:50-62.

[5] Berman O, Drezner Z, Wesolowsky G. On the collection depots location problem on networks. Naval Ressearch Logistics 2002;49:15-24.

[6] Berman O, Krass D, Menezes M. Locating facilities in the presence of disruptions and incomplete information. Decision Sciences 2009;40:845-868.

[7] Berman O, Krass D, Menezes M. Location and reliability problems on a line: Impact of objectives and correlated failures on optimal location patterns. Omega 2013;41:766-779.

[8] Cánovas L, García S, Marín A. Solving the uncapacitated multiple allocation hub location problem by means of a dual ascent technique. European Journal of Operational Research 2007;179:990-1007.

[9] Cánovas L, Landete M, Marín A. New formulations for the uncapacitated multiple allocation hub location problem. European Journal of Operational Research; 2006;172:274-292.

[10] Daskin MS. A new approach to solving the vertex p-center problem to optimality: algorithm and computational results. Communications of the Operations Research Society of Japan 2000;45:428-436.

[11] Daskin MS. Network and discrete location. NewYork:Wiley;1995.

[12] Drezner Z. Heuristic solution methods for two location problems with unreliable facilities. Journal of the Operational Research Society 1987;38:509-514.

[13] Drezner Z, Wesolowsky G. On the collection depots location problem. European Journal of Operational Research 2001;130:510-518.

[14] Elloumi S, Labbé M, Pochet Y. A new formulation and resolution method for the p-center problem. INFORMS Journal on Computing 2004;16:84-94.

[15] Elzinga J, Hearn D. Geometrical solutions for some minimax location problems. Transportation Science 1972;6:379-394.

[16] Espejo I, Marín A, Rodríguez-Chía AM. Closest assignment constraints in discrete location problems. European Journal of Operational Research 2012;219:49-58.

[17] Hernández I, Ramírez-Márquez JE, Rainwater C, Pohl E, Medal H. Robust facility location: Hedging against failures. Reliability Engineering and System Safety 2014;123:73-80.

[18] Jaeger M, Goldberg J. A polynomial algorithm for the equal capacity p-center problem on trees. Transportation Science 1994;28:167-175.

[19] Kariv O, Hakimi SL. An algorithmic approach to network location problems part I: the p-centers. SIAM Journal of Applied Mathematics 1979;37:513-538.

[20] Khuller S, Sussmann YJ. The capacitated k-center problem. SIAM Journal on Discrete Mathematics 2000;13:403-418.

[21] Landete M, Marín A. New facets for the two-stage uncapacitated facility location polytope. Computational Optimization and Applications 2009;44:487-519.

[22] Lei TL, Church RL. Constructs for Multilevel Closest Assignment in Location Modeling. International Regional Science Review 2011;34:339-367.

[23] Lim KM, Bassamboo A, Chopra S, Daskin MS. Facility Location Decisions with Random Disruptions and Imperfect Estimation. Manufacturing and Service Operations Management 2013;15:239-249.

[24] Marín A, Nickel S, Puerto J, Velten S. A flexible model and efficient solution strategies for discrete location problems. Discrete Applied Mathematics 2009;157(5):1128-1145.

[25] Marín A, Nickel S, Velten S. An extended covering model for flexible discrete and equity location problems. Mathematical Methods of Operations Research 2010;71(1):125-163.

[26] Meggido N, Supowit KJ. On the complexity of some common geometric location problems. SIAM Journal on Computing 1984;13(1):182-196.

[27] Mladenovic N, Labbé M, Hansen P. Solving the p-center problem with tabu search and variable neighborhood search. Networks 2003;42:48-64.

[28] Özsoy FA, Pinar MÇ. An exact algorithm for the capacitated vertex p-center problem. Computers and Operations Research 2006;33:1420-1436.

[29] Puerto J, Ramos AB, Rodríguez-Chía AM. Single-Allocation Ordered Median Hub Location Problems. Computers and Operations Research 2011;38:559-570.

[30] Scaparra MP, Pallotino S, Scutellà MG. Large-scale local search heuristics for the capacitated vertex p-center problem. Networks 2004;32:241-255.

[31] Sørensen M, New facets and a branch-and-cut algorithm for the weighted clique problem. European Journal of Operational Research 2004;154:57-70.

[32] Tamir A, Halman, A. One-way and round-trip location problems. Discrete Optimization 2005;2:168-184.

[33] Wagner JL, Falkson LM. The optimal nodal location of public facilities with price-sensitive demand. Geographical Analysis 1975;7:69-83.