Ordered $p$-median problems with neighbourhoods

Víctor Blanco$^{1,2}$

Received: 23 October 2017 / Published online: 18 February 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
In this paper, we introduce a new variant of the $p$-median facility location problem in which it is assumed that the exact location of the potential facilities is unknown. Instead, each of the facilities must be located in a region around their initially assigned location (the neighborhood). In this problem, two main decisions have to be made simultaneously: the determination of the potential facilities that must be open to serve the customers’ demand and the location of the open facilities in their neighborhoods, at global minimum cost. We present several mixed integer non-linear programming formulations for a wide family of objective functions which are common in Location Analysis: ordered median functions. We also develop two math-heuristic approaches for solving the problem. We report the results of extensive computational experiments.

Keywords Facility location · Mixed integer second order cone programming · Neighborhoods · Ordered median

Mathematics Subject Classification 90B85 · 90C22 · 90C30 · 47A30

1 Introduction

Facility location concerns with the optimal placement of one or several new facilities/plants to satisfy the customers’ demands. In discrete facility location problems, the positions of both the customers and the potential new facilities are part of the input as well as the travel costs between them. On the other hand, in continuous facility location problems, although the (geographical) coordinates (in a given $d$-dimensional space) of the customers are provided, the information about the potential location of the facilities is unknown, in the sense that the facilities can be located at any place of
the given space. Both the discrete and the continuous versions of these facility location problems have been widely studied in the literature (see the monographs [15,33] and the references therein). Several versions of these facility location problems have been analyzed, by considering different objective functions [45], by fixing either the number of facilities to be located (as in the \( p \)-median or \( p \)-center problems) [20] or the maximum capacities for the facilities (capacitated facility location) [25,38], or assuming uncertainty in the demands of the customers (see [2,9,10] for a recent review), amongst many others.

In this paper, we propose a unified framework for facility location problems in which the underlying problem is a discrete facility location problem. However, because of locational imprecision or inaccuracy, the new facilities are allowed to be located not only in the exact location of the potential facilities, but in certain regions around each of them, the \textit{neighborhoods}. In case the initial placements of the potential facilities are exact enough, that is, their neighborhoods are singletons (with a single element which coincides with the initial placement of the potential facilities), the problem becomes the discrete location version of the problem. On the other hand, if the neighborhoods are large enough, the problem turns into the continuous location version of the problem, allowing the facilities to be located in the entire space. Otherwise, different shapes and sizes for the neighborhoods allow one to model how imprecise the provided locational information is. The goal is, apart from the discrete location decision of the problem (placement of facilities among the given set and allocations customers-plants), to find the optimal location of the open facilities in the neighborhoods. The main difference between this problem and its underlying discrete facility location problem, is that in the former, the travel distances between facilities and customers are assumed to be known, while in the neighborhood version of the problem, as in the continuous case, those distances depend on the place where the facility is located in the neighborhood. Hence, in this problem, the matrix of travel costs is not provided, but a distance measure to compute the travel costs between customers and facilities is given. This problem, as far as we know, has not been fully investigated in Location Analysis, although some attempts have been presented in [8] and [21] where sensitivity analyses were performed by allowing the customers to \textit{move} around disk-shaped neighborhoods on the plane. Also, this problem can be seen as a constrained version of the classical multifacility location problem, which have been only partially studied in the literature (see [6]). This framework will be called \textit{Facility Location with Neighborhoods}, a terminology borrowed from the neighborhood versions of the Minimum Spanning Tree problem [4,14] and the Traveling Salesman problem [11,16].

The importance of analyzing this family of problems comes from its wide range of applications. It is well known that discrete facility location problems are useful in many real-world applications (see [27,32], amongst many others). However, in many situations, as for instance in the design of telecommunication networks, where a set of servers must be located to supply connection to a set of customers, the exact location of a server may not be exactly provided. In contrast, a region where the decision maker wishes to locate each of the facilities (a corridor, a room, or any other bounded shape) can be \textit{easily} given. In such a case, a robust worst-case decision would not reflect reality, since the decision maker does not explicitly provide the location of the facility because a lack of certainty but because it allows locational flexibility to the decision.
An optimal design may be obtained if the new facilities are allowed to be located in adequately chosen neighborhoods.

In this paper, we provide suitable mathematical programming formulations for the neighborhood versions of a widely studied family of objective functions in facility location problems: ordered median (OM) functions. In these problems, $p$ facilities are to be located by minimizing a flexible objective function that allows one to model different classical location problems. For instance, OM problems allow modeling location problems in which the customers support the median ($p$-median) or the maximum ($p$-center) travel costs, among many other robust alternatives. OM problems were introduced in Location Analysis by Puerto and Fernández [39] and several papers have analyzed this family of objective functions in facility location: discrete problems [23,26,37], continuous problems [3,5], network/tree location problems [22,44,47], hub location problems [41], stochastic facility location problems [49], multiobjective location [19], etc (see [43] for a recent overview on the recent developments on ordered median location problems). In particular, we analyze the neighborhood version of OM location problems for the so-called monotone case. We study the still general case in which the neighborhoods are second-order cone representable regions. These sets allow one to model as particular cases polyhedral neighborhoods or $\ell_\tau$-norm balls. The distance measure to represent travel costs between customers and facilities are assumed to be $\ell_\nu$-norm based distances. Within this framework we present four different mixed integer second order cone optimization (MISOCO) models.

The current limitations of the on-the-shelf solvers to solve mixed integer nonlinear problems, and the difficulty of solving even the underlying problem (the classical $p$-median problem is NP-hard), makes the resolution of the problem under study a hard challenge. For that reason, we also develop two math-heuristic algorithms based on different location-allocation schemes, which are able to solve larger problems.

Our paper is organized in five sections. In Sect. 2 we introduce the problem and some general properties are stated. Sect. 3 is devoted to provide four different mixed integer non linear programming formulations of the problem. At the end of the section, we run some computational experiments in order to compare the four formulations. In Sect. 4 the two math-heuristic approaches are described, and the results of some computational experiments are reported. Finally, some conclusions are presented in Sect. 6.

### 2 DOMP with neighborhoods

In this section we introduce the Ordered Median Problem with Neighborhoods (OMPN) in which the underlying discrete facility location problem is the Discrete Ordered $p$-Median Problem (DOMP).

For the sake of presentation, we first describe the DOMP. The input data for the problem is:

- $A = \{a_1, \ldots, a_n\} \subseteq \mathbb{R}^d$: set of coordinates of the customers. We assume, as usual in the location literature, that the coordinates of the potential facilities coincides with $A$. 
\[ D = \left( d(a_i, a_j) \right)_{i,j=1}^{n} \in \mathbb{R}^{n \times n} : \text{Travel cost matrix between facilities.} \]

- \( \lambda_1, \ldots, \lambda_n \geq 0 \): Ordered median function weights.

The goal of the DOMP is to select, from the elements of \( \mathcal{A} \), a subset of \( p \) facilities, \( \mathcal{B} \subset \mathcal{A} \) with \( |\mathcal{B}| = p \), that minimizes the ordered median objective function:

\[
\sum_{i=1}^{n} \lambda_i D_{(i)},
\]

where \( D_{i} = \min_{a \in \mathcal{B}} d(a_i, a) \) (the smallest travel cost to supply customer \( i \) from the open facilities), and \( D_{(i)} \) represent the \( i \)-th largest element in the set \( \{D_1, \ldots, D_n\} \), i.e. \( D_{(i)} \in \{D_1, \ldots, D_n\} \) with \( D_{(1)} \geq \cdots \geq D_{(n)} \).

The DOMP can be stated as the following optimization problem:

\[
\min_{\mathcal{B} \subset \mathcal{A} : |\mathcal{B}| = p} \sum_{i=1}^{n} \lambda_i D_{(i)} \quad \text{(DOMP)}
\]

We will assume that the \( \lambda \)-weights verify \( \lambda_1 \geq \cdots \geq \lambda_n \geq 0 \), dealing with the so-called convex ordered median problem. Most of the main well-known objective functions in Locational Analysis are part of this family, as for instance:

- Median (\( \lambda = (1, \ldots, 1) \)): \( \sum_{i=1}^{n} D_i \).
- Center (\( \lambda = (1, 0, \ldots, 0) \)): \( \max_{i=1,\ldots,n} D_i \).
- \( K \)-Center (\( \lambda = (1, K, 1, 0, \ldots, 0) \)): \( \sum_{i=1}^{K} D_{(i)} \).
- Cent-Dian\( \alpha \) (\( \lambda = (1, 1 - \alpha, \cdots, 1 - \alpha) \)): \( \alpha \max_{i=1,\ldots,n} D_i + (1 - \alpha) \sum_{i=1}^{n} D_i \), for \( 0 \leq \alpha \leq 1 \).

Ordered median functions are continuous and symmetric (in the sense that they are invariant under permutations). Furthermore, if \( \lambda_1 \geq \cdots \geq \lambda_n \geq 0 \), ordered median functions are convex, fact that will be exploited throughout this paper. The interested reader is referred to [43] for a complete description of the properties of ordered median functions.

A few formulations and exact solution approaches for DOMP have been developed since the problem was introduced. In particular Boland et. al [7] formulated the problem as a (non convex) quadratic problem with quadratic constraints. A suitable three index (pure) binary programming reformulation with \( O(n^3) \) variables and \( O(n^2) \) linear constraints was provided by linearizing the bilinear terms. A second formulation, reducing to two the indices of the variables in the formulation, was also presented in the same paper by using a different linearization strategy, and that allows reducing the number of binary variables to \( O(n^2) \). Puerto [38], Marin et al. [30], Marin et al. [29] and Labbé et al. [26] provided alternative formulations for the problem with two and three indices, that need, in a preprocessing phase, sorting the elements in the matrix \( D \) (and removing duplicates). All the above mentioned formulations are valid for general ordered median problems. Concerning the convex case, Ogryzack and Tamir [36]...
presented a different formulation which exploits the monotonicity of the \( \lambda \)-weights by applying a \( k \)-sum representation of the ordered median function (see also the recent paper [42] for further details on the power of this representation in a wide variety of optimization problems). Finally, in Blanco et al. [5] the authors derived a formulation that also avoids using the binary variables for sorting the distances. Also, a few heuristic approaches are available in the literature for the DOMP (see [13,40,46]).

Observe also that in the DOMP, once the travel costs matrix is provided, the locational coordinates of the customers are no longer needed, and then, the problem does not depend on the dimension of the space where the customers live.

For the OMPN framework, instead of providing a travel cost matrix between customers, we consider a travel distance measure \( d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+ \) induced by a norm \( \| \cdot \| \), i.e. \( d(a,b) = \|a - b\| \), for \( a, b \in \mathcal{A} \).

Also, each potential facility, \( a \in \mathcal{A} \), is associated to a convex set, \( \mathcal{N}(a) \subset \mathbb{R}^d \), with \( a \in \mathcal{N}(a) \), its neighbourhood. We denote by \( \overline{\mathcal{N}} = \prod_{a \in \mathcal{A}} \mathcal{N}(a) \), the space of neighborhoods. We also consider in this case, set-up costs for opening facilities (which may be neighborhood-dependent), which we denote by \( f(a) \) for each \( a \in \mathcal{A} \).

The goals of the Ordered Median Problem with Neighborhoods are:

- to find the indices of the \( p \) facilities to open, \( \{a_{i_1}, \ldots, a_{i_p}\} \subseteq \mathcal{A} \),
- to locate the facilities into their neighbourhoods: \( \bar{a}_{i_1}, \ldots, \bar{a}_{i_p} \) with \( \bar{a}_{i_j} \in \mathcal{N}(a_{i_j}) \), \( j = 1, \ldots, p \), and
- to allocate customers to their closest open facilities \( \bar{a}_{i_1}, \ldots, \bar{a}_{i_p} \),

by minimizing an ordered median function of the travel distances plus set-up costs.

Observe that the optimization problem to solve for the OMPN is similar to (DOMP):

\[
\min_{\mathcal{B} \subseteq \mathcal{A}, |\mathcal{B}|=p} \ C(\mathcal{B}) := \sum_{i=1}^{n} \lambda_i D_i(i) + \sum_{a \in \mathcal{B}} f(a) \quad \text{(OMP\textsubscript{N})}
\]

where now, \( D_i = \min_{a \in \mathcal{B}} d(a_i, \bar{a}) \), i.e. the travel distance from the \( i \)-th demand point to its closest facility depends not only on the open facilities, but on their positions into their neighborhoods, \( \bar{a} \) (recall that \( \bar{a} \) stands for the optimal position of an open facility \( a \in \mathcal{B} \) within its neighborhood \( \mathcal{N}(a) \)). So both the discrete location (open facilities and allocation scheme) and the continuous location decisions (coordinates of the new facilities) are involved in the problem.

**Remark 2.1** Observe that the OMPN is a family of facility location problems for which the discrete and the continuous versions belong to.

- If \( \mathcal{N}(b) = \{b\} \) for all \( b \in \mathcal{B} \), then, OMPN coincides with the DOMP. This is straightforward since the only option for each of the potential facilities is the facility itself.
- Let \( \text{ch}(\mathcal{A}) \) the convex hull of the demand points (which is polyhedral because of the finiteness of \( \mathcal{A} \)). It is known (see [33]) that at least one solution of the continuous multifacility ordered median problem belong to the convex hull of the demand points. Thus, if \( \text{ch}(\mathcal{A}) \subseteq \mathcal{N}(b) \) for all \( b \in \mathcal{B} \), the solutions of OMPN and the continuous version of the problem coincide.
Without loss of generality, we assume that each open facility is allocated to itself. Otherwise, one may duplicate the set of demand points/potential facilities to account for different allocations. Observe that once a facility is open, one could consider that no longer acts as a demand point but as a service point, whose opening cost is accounted in the set-up costs but not in the transportation costs.

We use the classical notation for the variables in $p$-median problem:

$$x_{ij} = \begin{cases} 1 & \text{if client } i \text{ is allocated to facility } j (i \neq j) \text{ or if facility } j \text{ is open } (i = j), \\ 0 & \text{otherwise} \end{cases}$$

for $i, j = 1, \ldots, n$.

Note that, using the above family of variables, the set of open facilities and assignments between customers and $p$ facilities can be represented by the set $\mathcal{X} = \mathcal{X}_R \cap \{0, 1\}^{n \times n}$, where

$$\mathcal{X}_R = \left\{ x \in [0, 1]^{n \times n} : \sum_{j=1}^{n} x_{ij} = 1, \forall i = 1, \ldots, n, \right. \\
\left. \sum_{j=1}^{n} x_{jj} = p, x_{ij} \leq x_{jj}, \forall i, j = 1, \ldots, n \right\},$$

is the so-called $p$-median polytope.

Observe also that, the above settings easily extend to the case in which the possible connections between demand points and facilities is induced by a graph.

On the other hand, the set of distances, will be represented by the following set:

$$\mathcal{D} = \left\{ (d, \tilde{a}) \in \mathbb{R}_+^{n \times n} \times \mathbb{N} : d_{ij} \geq \|a_i - \tilde{a}_j\|, i, j = 1, \ldots, n, i \neq j \right\},$$

where $d_{ij}$ (when one tries to minimize some aggregating function of the travel-costs) represents the distance between the customer located at $a_i$ and the facility located at $\tilde{a}_j$, for all $i, j = 1, \ldots, n$. Observe that the distances $d_{ii}$ are not taken into account in $\mathcal{D}$ since self-allocation costs will be accounted not in the transportation costs but in the set-up costs.

Note that the set $\mathcal{D}$ can be easily adapted to the case in which each customer uses a different travel distance measure (norm), and the structure of $\mathcal{D}$ remains the same.

With the above notation, the general OMPN can be compactly formulated as:

$$\min \sum_{i=1}^{n} \lambda_i z_{(i)} + \sum_{j=1}^{n} f_j x_{jj}$$

s.t. $z_i = \sum_{j=1}^{n} d_{ij} x_{ij}$, for $i = 1, \ldots, n$, \hspace{1cm} (2.2)

$$x \in \mathcal{X}, (d, \tilde{a}) \in \mathcal{D}. \hspace{1cm} (2.3)$$
where $f_j$ denotes the set-up cost of the facility initially located at $a_j$, for $j = 1, \ldots, n$, and $z_i$ represents the minimum distance between customer located at $a_i$ and the open facilities, for $i = 1, \ldots, n$.

Observe that (2.1)–(2.3) is a mixed integer non linear programming problem (MINLP), whose continuous relaxation is neither convex nor concave due to the bilinear constraint (2.2) and to the constraints in $D$. In case the neighborhoods are convex, the set $D$ is also convex (because of the convexity of the norm). Hence, if the discrete location variables $x$ were known, the problem (also because of the convexity of the ordered median function) becomes a continuous convex problem. On the other hand, if the distances were known, the problem becomes a DOMP, so several formulations can be applied to solve the problem. In the OMPN, both $X$ and $D$ are part of the final decision. Thus, both the difficulties of handling the DOMP and the continuous problem are inherited to the OMPN. In particular, since the $p$-median problem (or the $p$-center problem) is known to be NP-hard [24] which is a particular case of OMPN, the OMPN is also NP-hard.

The simplest OMPN problem, apart from the DOMP case (where the neighborhoods can be seen as singletons), is obtained when the set $D$ is a polyhedron (and then, defined by a set of linear inequalities). Since the geometry of $D$ depends on the distance measure induced by $\| \cdot \|$ and the shapes of the neighborhoods, $D$ will be a polyhedron when these two features can be linearly represented. The norms which are polyhedrally-representable are called block (or polyhedral) norms (see [33]) which are characterized by the fact that their unit balls are polytopes, i.e. $P = \{ z \in \mathbb{R}^d : \|z\| \leq 1 \}$ is a bounded polyhedron. On the other hand, the polyhedral representability of the neighborhoods, because they are assumed to be compact and convex sets, is assured if and only if they are also polytopes. In those cases, both sets $D$ and $X$ are identified with sets of linear inequalities and equations and by adding a set of $O(n^2)$ binary variables to our model. The above observations are summarized in the following result.

**Theorem 2.1** Let $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$, $\| \cdot \|$ a block norm and $N(a)$ a polyhedron, for each $a \in A$. Then, OMPN can be formulated as a mixed-integer linear programming problem.

**Proof** The proof follows noting that constraints in the form $Z \geq \|X - Y\|$, as those that appear in the description of $D$, can reformulated as:

$$Z \geq e' (X - Y), \forall e \in \text{Ext}(P^*),$$

where $\text{Ext}(P^*)$ the set of extreme points of $P^* = \{ v \in \mathbb{R}^d : v' b_g \leq 1, g = 1, \ldots, |\text{Ext}(P)| \}$, the unit ball of the dual norm of $P$ (see [33,48]).

The following example illustrates the new framework under study.

**Example 2.1** Let us consider a set of customers/potential facilities with coordinates in the plane $A = \{(0, 5), (1, 1), (1, 6), (1, 4), (5, 3), (10, 4), (6.5, 0), (8, 6)\}$, and travel distances measured with the Euclidean norm. The solutions for the 2-median,
the 2-center and the 2-4-center (\(K\)-center with \(K = 4\)) are drawn in Fig. 1 (there, stars represent open facilities, customers are identified with dots and lines are the allocation patterns).

Note that, as expected, the solutions for the three DOMPs highly depend on the \(\lambda\)-weights, being the optimal set of open facilities different for the three cases.

Let us now consider, for each demand point, a neighbourhood defined as the Euclidean disk with radii \(r \in \{1, 0.6, 1, 0.6, 2.4, 2.4, 0.8, 1.6\}\) (see Fig. 2).

The new facilities, now, are not restricted to be exactly located in the given coordinates but in a disk around them. We consider the radius of its neighborhood (disk) as a measure of the set-up cost for each facility, that is \(f(a_i) = r_i\). The solutions of the neighbourhood version of the 2-median, 2-center and 2-4-center problems are shown in Fig. 3.

In what follows, we derive some structural properties of DOMP that are inherited to the OMPN.

For each \(i, j = 1, \ldots, n\) with \(i \neq j\), we denote by \(\hat{D}_{ij} = \max\{\|a_i - \bar{a}_j\| : \bar{a}_j \in \mathcal{N}_j\}\), and \(\hat{d}_{ij} = \min\{\|a_i - \bar{a}_j\| : \bar{a}_j \in \mathcal{N}_j\}\), upper and lower bounds for the distances between the \(i\)th customer and the \(j\)th potential facility, respectively.
Properties 2.1 The following properties are satisfied:

1. There exists an optimal solution of \((\text{OMP})\) in which the \(p\) smaller travel distances equal 0.

2. Let \(\mathcal{B} \subseteq \mathcal{A}\) a set of \(p\) facilities such that its ordered cost \(c(\mathcal{B}) \leq UB\) and such that \(\min_{j \neq i} \hat{d}_{ij} > \frac{UB}{n-m+1}\) for some \(m = 2, \ldots, n\), then the \(i\)-th client is sorted at most in position \(m - 1\) in the whole sorted set of optimal distances.

Proof 1. The result follows from the assumption that if the facility \(a \in \mathcal{A}\) is an open facility, the travel cost between \(a\) and \(\tilde{a}\) is zero.

2. Assume that the \(i\)th customer is sorted in position \(r \geq m\) in the sorting sequence of distances, i.e. \(D^{(1)} \geq \ldots \geq D^{(m)} \geq D^{(r)} = D_i\). Then, we have that:

\[
c(\mathcal{B}) = \sum_{l=1}^{n} \lambda_l D^{(l)} + \sum_{b \in \mathcal{B}} f(b) \geq \sum_{l=1}^{n-m+1} \lambda_l D^{(l)} \geq \sum_{l=1}^{n-m+1} \lambda_l D_i > UB
\]

which contradicts the hypothesis.

3 MINLP formulations for the OMPN

In this section, we describe different mathematical programming formulations for solving general OMPN. In particular, we extend the formulations presented in \([5,7,36]\) to our problem. As mentioned above, the main difference between the DOMP and the OMPN problem is that in the OMPN the distances are not part of the input, but part of the decision. Hence, the formulations for the DOMP based on preprocessing the travel distances matrix (as those proposed in \([26,29,30,38]\)) cannot be applied to our framework.

Observe that, in the OMPN, an adequate representation of \(D\) is crucial for the development of efficient solution approaches for the problem. We assume that the neighborhoods belong to a family of convex sets that allows us to represent most of the convex shapes which are useful in practice, and that can be efficiently handled by commercial optimization solvers: second order cone (SOC)-representable sets \([28]\). SOC-representable sets are convex sets defined by second-order cone constraints in the form:

\[
\|A_i x - b_i\|_2 \leq c_i^T x + d_i, \forall i = 1, \ldots, M, x \in \mathbb{R}^N,
\]

where \(A_1, \ldots, A_i \in \mathbb{R}^{M_i \times N}, b_i \in \mathbb{R}^{M_i}, c_i \in \mathbb{R}^N, d_i \in \mathbb{R}, \) for \(i = 1, \ldots, M\), and \(\| \cdot \|_2\) is the Euclidean norm. Most of the state-of-the-art solvers are capable
to efficiently solve optimization problems involving SOC constraints by means of quadratic constraints with positive definite matrices, second order cone constraints (in the form $x'^T x \leq y^2$, for $y \geq 0$) or rotated second order cone constraints ($x'^T x \leq yz$ with $y, z \geq 0$). SOC constraints allow one to represent, not only Euclidean balls, but any $\ell_\tau$-norm ball (see [5] for further details on the explicit representation of $\ell_\tau$-norm based distance constraints as a set of SOC constraints for any $\tau \in \mathbb{Q}$ with $\tau \geq 1$). Clearly, any polyhedron is SOC-representable (setting $A$ and $b$ equal to zero) so any intersection of $\ell_\tau$-norm balls and polyhedra is suitable to be represented as a set of second order cone constraints. Hence, both the neighborhoods and the distances involved in our problem will be defined as SOC-constraints, being then $D$ a SOC-representable set.

For the sake of simplicity, and without loss of generality, we assume that the neighborhood of each $a \in \mathcal{A}$ is a $\ell_\tau$-norm ball, i.e. $N(a) = \{z \in \mathbb{R}^d : \|z - a\|_\tau \leq r_a\}$, for some $r_a \in \mathbb{R}_+$ and $\tau \in \mathbb{Q}$ with $\tau \geq 1$.

Also, we consider that the travel distances are induced by a $\ell_\nu$-norm with $\nu \in \mathbb{Q}$ and $\nu \geq 1$. With these settings, we explicitly describe $D$ as follows:

$$D = \{(d, \bar{a}) \in \mathbb{R}_+^{n \times n} \times \mathbb{R}^{n \times d} : d_{ij} \geq \|a_i - \bar{a}_j\|_\nu, r_j \geq \|a_j - \bar{a}_j\|_\tau, i, j = 1, \ldots, n, i \neq j\}$$

where $r_j$ denotes the radius of the neighborhood $N(a_j)$.

The following result, whose proof is straightforward from [5, Theorem 2], allows us to efficiently represent the set $D$ when the involved norms are $\ell_\tau$-based norms.

**Properties 3.1** Let $\tau = \frac{r_\tau}{s_\tau} \geq 1$ and $\nu = \frac{r_\nu}{s_\nu} \geq 1$ with $r_\tau, s_\tau, r_\nu, s_\nu \in \mathbb{Z}_+$ and $\gcd(r_\tau, s_\tau) = \gcd(r_\nu, s_\nu) = 1$. Then, $D$ is representable as a set of $(n^2 + n)(2d + 1)$ linear inequalities and $(n \log r_\nu + \log r_\tau) \cdot n d$ second order cone contraints.

### 3.1 The three index formulation

The first formulation is based on the one proposed in [7], which uses, apart from the $x_{jj}$-variables described above, the following set of sorting/allocation binary variables for the DOMP:

$$w_{ij}^k = \begin{cases} 1 & \text{if customer } i \text{ is allocated to facility } j \text{ and its distance, } \|a_i - \bar{a}_j\|, \\ \text{is sorted in the } k\text{th position.} & \\ 0 & \text{otherwise.} \end{cases}$$

for $i, j = 1, \ldots, n$ with $i \neq j$ and $k = 1, \ldots, n - p$.

Observe that by Proposition 2.1, only $n - p$ travel distances are taken into account in the formulations (the other $p$ correspond to self-allocation of open facilities).

This formulation reads as follows:

$$\min \sum_{i,j,k=1}^{n} \sum_{i \neq j}^{n-p} \lambda_k d_{ij} w_{ij}^k + \sum_{j=1}^{n} f_j x_{jj} \quad \text{(OMPNI3I)}$$
Ordered $p$-median problems with neighbourhoods

\[ \text{s.t. } \sum_{j=1}^{n} \sum_{k=1}^{n-p} w_{ij}^k = 1, \forall i = 1, \ldots, n(i \neq j), \quad (3.1) \]

\[ \sum_{i,j=1: i \neq j}^{n} w_{ij}^k = 1, \forall k = 1, \ldots, n, \quad (3.2) \]

\[ \sum_{k=1}^{n-p} w_{ij}^k \leq x_{jj}, \forall i, j = 1, \ldots, n, i \neq j \quad (3.3) \]

\[ \sum_{j=1}^{n} x_{jj} = p, \quad (3.4) \]

\[ \sum_{i,j=1: i \neq j}^{n} d_{ij}w_{ij}^{k-1} \geq \sum_{i,j=1: i \neq j}^{n} d_{ij}w_{ij}^{k}, \forall k = 2, \ldots, n - p, \quad (3.5) \]

The objective function assigns to each sorted distance its adequate weight $\lambda$. (3.1) (resp. (3.2)) ensures that each demand point (resp. each position) is assigned to a unique facility and a unique position (resp. demand point). (3.3) assures that allocation is not allowed unless the plant is open, and (3.4) restrict the problem to open exactly $p$ facilities. Constraints (3.5) allows us a correct definition of the $w$-variables in which the sorting of the distances is imposed [the $(k - 1)$th distance is at least as larger as the $k$-th distance].

Although the above formulation is valid for OMPN, both the objective function and the set of constraints (3.5) are quadratic and non-convex. We introduce a new set of variables to account for the non linear terms in the above formulation:

\[ \theta_{ij}^k = d_{ij}w_{ij}^k, \quad i, j = 1, \ldots, n(i \neq j), k = 1, \ldots, n - p. \]

Using the $\theta$-variables, the objective function can be reformulated as:

\[ \sum_{i,j=1: i \neq j}^{n} \sum_{k=1}^{n-p} \lambda_k \theta_{ij}^k + \sum_{j=1}^{n} f_j x_{jj}. \]

The correct definition of the new variables and the satisfaction of constraints (3.5) is assured by the following sets of linear constraints:

\[ \theta_{ij}^k \geq d_{ij} - \hat{D}_{ij} (1 - w_{ij}^k), \quad \forall i, j = 1, \ldots, n(i \neq j), k = 1, \ldots, n - p, \]
\[
\sum_{i,j=1: i \neq j}^{n} \theta_{ij}^{k-1} \geq \sum_{i,j=1}^{n} \theta_{ij}^{k}, \quad \forall k = 2, \ldots, n - p,
\]

where the first set of constraints comes from the McCormick linear reformulation \cite{31} of the bilinear terms defining the \( \theta \)-variables, and the second is the reformulation of (3.5) with the new variables.

The formulation above, has \( O(n^3) \) variables and \( O(n^2) \) constraints. Properties 2.1 allow us to strengthen the formulation (OMPN\(_{31}\)).

**Corollary 3.1** If \( UB \) is a known upper bound for the optimal value of OMPN, then:

\[
w_{ij}^k = 0, \forall i, j = 1, \ldots, n, (i \neq j), k = 1, \ldots, n - p, \text{ such that } \min_{j' \neq i} \hat{d}_{ij'} > UB \frac{n - k + 1}{\lambda_l}.\]

**Proof** Assume that there exist \((i, j, k)\) such that \( \min_{j' \neq i} \hat{d}_{ij'} > UB \frac{n - k + 1}{\lambda_l} \) and \( w_{ij}^k = 1 \). Hence, by definition of \( w \), facility \( j \) is open and the \( i \)th customer is ranked in position \( k \). Furthermore, customer \( i \) is allocated to \( j \), thus \( \min_{j' \neq i} d_{ij'} = d_{ij} \geq \hat{d}_{ij} > UB \frac{n - k + 1}{\lambda_l} \). However, by Proposition 2.1, customer \( i \) is sorted at most in position \( k - 1 \), contradicting the assumption. \( \square \)

### 3.2 The 2-index formulation

The second formulation, also based on the one presented in \cite{7}, considers an alternative representation of the sorting variables. It uses two different sets of variables. The first one allows us to sort the distances of supplying each of the customers:

\[
s_{ik} = \begin{cases} 1 & \text{if the distance supported by the } i \text{th customer is sorted in the } k \text{th position.} \\ 0 & \text{otherwise.} \end{cases}
\]

for \( i = 1, \ldots, n \) and \( k = 1, \ldots, n - p \), while the second represents the sorting (non decreasing) sequence of distances:

\[
\xi_k = \sum_{i=1}^{n} s_{ik} \sum_{j=1: j \neq i}^{n} d_{ij} x_{ij}, \quad k = 1, \ldots, n - p.
\]
This representation allows us to simplify the formulation to the following with $O(n^2)$ variables and $O(n^2)$ constraints.

\[
\min \sum_{k=1}^{n} \lambda_k \xi_k + \sum_{j=1}^{n} f_j x_{jj} \quad \text{(OMPN2)}
\]

\[
\text{s.t. } \xi_{k-1} \geq \xi_k, \forall k = 2, \ldots, n - p,
\]

\[
\sum_{k=1}^{n-p} \xi_k = \sum_{i,j=1: i \neq j}^{n} d_{ij} x_{ij}, \quad \text{(3.7)}
\]

\[
\xi_k \geq \sum_{j=1: j \neq i}^{n} d_{ij} x_{ij} - \hat{D}_{ij} (1 - s_{ik}), \forall i = 1, \ldots, n, k = 1, \ldots, n - p, \quad \text{(3.8)}
\]

\[
\sum_{i=1}^{n} s_{ik} = 1, \forall k = 1, \ldots, n - p, \quad \text{(3.9)}
\]

\[
\sum_{k=1}^{n-p} s_{ik} = 1, \forall i = 1, \ldots, n, \quad \text{(3.10)}
\]

\[
\xi_k \geq 0, \forall k = 1, \ldots, n - p,
\]

\[
s_{ik} \in \{0, 1\}, \forall i, k = 1, \ldots, n - p,
\]

\[
x \in \mathcal{X}, (d, \bar{a}) \in \mathcal{D}.
\]

The correct definition of the $\xi$-variables is assured by constraints (3.6)–(3.8), while constraints (3.9) and (3.10) allows us the adequate modeling of the $s$-variables.

As in (OMPN3), to avoid nonconvex terms in the formulation, the bilinear terms $d_{ij} x_{ij}$ can be linearized by introducing a new variable $\theta_{ij} = d_{ij} x_{ij}$ and replacing (3.7) and (3.8) by:

\[
\sum_{k=1}^{n} \xi_k = \sum_{i,j=1}^{n} \theta_{ij}, \quad \text{(3.11)}
\]

\[
\xi_k \geq \sum_{j=1: j \neq i}^{n} \theta_{ij} - \hat{D}_{ij} (1 - s_{ik}), \forall i = 1, \ldots, n, k = 1, \ldots, n - p, \quad \text{(3.12)}
\]

\[
\theta_{ij} \geq d_{ij} - \hat{D}_{ij} (1 - x_{ij}), \forall i, j = 1, \ldots, n(i \neq j), k = 1, \ldots, n - p. \quad \text{(3.13)}
\]

### 3.3 The $K$-sum formulation

Ogryczak and Tamir presented in [36] some linear programming formulations for the problem of minimizing the sum of the $K$ largest linear functions (which is a particular
case of ordered median function). In the same paper, the approach is extended to the minimization of convex ordered median functions by means of a telescopic sum of $K$-sum functions. In the next formulation, we apply this idea to formulate the OMPN. For the sake of readability, we first formulate the $K$-center problem.

Let $\lambda = (1, \ldots, 1, 0, \ldots, 0)$. The ordered median function associated to this particular choice of $\lambda$ is known as the $K$-center function. With our notation, provided the set of distances $D_1, \ldots, D_n$, the $K$-center problem consists of minimizing $\Theta_K(D) = \sum_{i=1}^{K} D(i)$. Such an objective function, is proved in [36] to be equivalent to the following expression

$$\Theta_K(D) = \frac{1}{n} \left( K \sum_{i=1}^{n} D_i + \min_{t \in \mathbb{R}} \sum_{i=1}^{n} (K (t - D_i)_+ + (n - K) (D_i - t)_+) \right)$$

where $z_+ = \max\{0, z\}$ for $z \in \mathbb{R}$, and the optimal value $t^*$ into the above expression coincides with $D(K)$ (the $K$-th largest distance). Hence, to minimize $\Theta_K(D)$ one may proceed by solving:

$$\min_K t + \sum_{i=1}^{n} z_i$$

s.t. $z_i \geq D_i - t$, $\forall i = 1, \ldots, n$,

$z_i \geq 0$, $\forall i = 1, \ldots, n$,

$t \in \mathbb{R}$.

where the variable $z_i$ is identified with $(D_i - t)_+$ in the above formulation, for $i = 1, \ldots, n$. Thus, incorporating the whole information to define the distances in our location problem, the $K$-center location problem with neighborhoods can be formulated as:

$$\min_K t + \sum_{i=1}^{n} z_i + \sum_{j=1}^{n} f_j x_{ij}$$

s.t. $z_i \geq D_i - t$, $\forall i = 1, \ldots, n$,

$D_i \geq d_{ij} - \hat{D}_{ij}(1 - x_{ij})$, $\forall i, j = 1, \ldots, n (i \neq j)$,

$z_i, D_i \geq 0$, $\forall i = 1, \ldots, n$,

$t \in \mathbb{R}$,

$x \in \mathcal{X}$, $(d, \bar{a}) \in \mathcal{D}$.

The above formulation can be extended to general convex ordered median functions. Observe that if $\lambda_1 \geq \cdots \geq \lambda_n \geq \lambda_{n+1} := 0$ one may represent the ordered median function of the distances $D_1, \ldots, D_n$ by using a telescopic sum:
\[
\sum_{i=1}^{n-1} \lambda_i D_{(i)} = \sum_{k=1}^{n-1} (\lambda_k - \lambda_{k+1}) \sum_{i=1}^{K} D_{(i)} = \sum_{K=1}^{n} \Delta_k \Theta_k(D)
\]

where \(\Delta_k = \lambda_k - \lambda_{k+1} \geq 0\) for \(k = 1, \ldots, n - 1\).

Thus, the convex ordered objective functions can be equivalently rewritten as a weighted sum of \(K\)-sums, being then suitable to be represented as in the \(K\)-center problem. With such an observation and introducing new \(t\)-variables (one for each of the \(K\)-sums involved) and \(z\)-variables, in this case with two indices to account not only for the customer \((i)\) but also for the \(K\)-sum representation, one obtains the following valid formulation for the OMPN:

\[
\text{min} \quad \sum_{k=1}^{n-1} \Delta_k (k t_k + \sum_{i=1}^{n} z_{ik}) + \sum_{j=1}^{n} f_j x_{jj} \\
\text{s.t.} \quad z_{ik} \geq D_i - t_k, \forall i, k = 1, \ldots, n, \\
D_i \geq d_{ij} - \hat{D}_{ij} (1 - x_{ij}), \forall i, j = 1, \ldots, n (i \neq j), \quad (3.18) \\
z_{ik}, D_i \geq 0, \forall i = 1, \ldots, n, \quad (3.19) \\
t_k \in \mathbb{R}, k = 1, \ldots, n, \quad (3.20) \\
x \in X, (d, \bar{a}) \in D.
\]

Observe that this formulation has also \(O(n^2)\) variables and \(O(n^2)\) constraints, but, as will be shown in the computational results, it has a better performance than \((\text{OMP}N_{2f})\) since it uses, intrinsically, the special structure of the convex ordered median objective.

### 3.4 The BEP formulation

Finally, we present a formulation, based on the one provided in [5] and that, as the one in the previous subsection, is only valid for the convex case. The idea under the formulation comes from the observation that, because \(\lambda_1 \geq \cdots \geq \lambda_n \geq 0\), the evaluation of the ordered median function on a set of distances, is reached when choosing, among all possible permutations of the indices, \(\mathcal{P}_n\), the one that maximizes the weighted sum, that is:

\[
\sum_{i=1}^{n} \lambda_i D_{(i)} = \max_{\sigma \in \mathcal{P}_n} \sum_{i=1}^{n} \lambda_i D_{\sigma(i)}.
\]

Then, if the permutations of \(\{1, \ldots, n\}\) are represented by using the set of binary variables

\[
p_{ik} = \begin{cases} 
1 & \text{if the permutation assigns index } i \text{ to index } k, \\
0 & \text{otherwise}
\end{cases}
\]

Observe that this formulation has also \(O(n^2)\) variables and \(O(n^2)\) constraints, but, as will be shown in the computational results, it has a better performance than \((\text{OMP}N_{2f})\) since it uses, intrinsically, the special structure of the convex ordered median objective.
verifying that \( \sum_{i=1}^{n} p_{ik} = 1 \) (for all \( k = 1, \ldots, n \)) and \( \sum_{k=1}^{n} p_{ik} = 1 \) (for all \( i = 1, \ldots, n \)).

Then, using these variables, the ordered median sum of a given set of values \( D_1, \ldots, D_n \) is equivalent to:

\[
\sum_{i=1}^{n} \lambda_i D_i = \max_{p \in \{0, 1\}^{n \times n}} \sum_{i,k=1}^{n} \lambda_k D_i p_{ik}
\]

s.t. \( \sum_{i=1}^{n} p_{ik} = 1, \forall k = 1, \ldots, n, \)

\( \sum_{k=1}^{n} p_{ik} = 1, \forall i = 1, \ldots, n. \)

The optimization problem above is an assignment problem, then, the total unimodularity of the constraints matrix assures that its optimal value coincides with the one of its dual problem which reads:

\[
\min \sum_{k=1}^{n} u_k + \sum_{i=1}^{n} v_i
\]

s.t. \( u_i + v_k \geq \lambda_k D_i, \forall i, k = 1, \ldots, n, \)

\( u, v \in \mathbb{R}^n. \)

Merging the above representation of the ordered median function with the location problem, the OMPN can be reformulated as:

\[
\min \sum_{k=1}^{n} u_k + \sum_{i=1}^{n} v_i + \sum_{j=1}^{n} f_j x_{jj}\quad (\text{OMP}N_{BEP})
\]

s.t. \( u_i + v_k \geq \lambda_k D_i, \forall i, k = 1, \ldots, n, \)

\( D_i \geq d_{ij} - \hat{D}_{ij}(1 - x_{ij}), \forall i, j = 1, \ldots, n(i \neq j), \)

\( x \in \mathcal{X}, \)

\( (d, \bar{a}) \in \mathcal{D}. \)

### 3.5 Comparison of formulations

In this section we provide some theoretical results that allow us to compare the tightening of the convex relaxations of each of the provided formulations. Let us denote by \( z_{\text{OMP}N_{3I}}^R, z_{\text{OMP}N_{2I}}^R, z_{\text{OMP}N_{OT}}^R \) and \( z_{\text{OMP}N_{BEP}}^R \) the optimal values of the continuous relaxations of formulations \( (\text{OMP}N_{3I}), (\text{OMP}N_{2I}), (\text{OMP}N_{OT}) \) and \( (\text{OMP}N_{BEP}) \), respectively.

**Proposition 3.1** The following relations are verified:
1. $z_{3I}^R \leq z_{2I}^R$.
2. $z_{3I} \leq z_{BEP}^R$.
3. $z_{BEP}^R = z_{OT}^R$.

**Proof** See “Appendix A”.

In the above result, the relationship between $z_{3I}^R$ and $z_{BEP}^R$ (or $z_{OT}^R$) is not stated. One may think that the continuous relaxation of (OMP$_{BEP}$) is tighter than (OMP$_{2I}$), because the first exploits the monotonicity of the $\lambda$-weights. However, that is not always true as illustrated in the following example.

**Example 3.1** Let us consider five points in $\mathbb{R}^2$, $A = \{(2, 92), (33, 70), (65, 50), (73, 69), (40, 63)\}$ and neighborhoods defined as Euclidean disks with radii $\{2, 1, 0.05, 5, 1\}$. If the distance measure is the Euclidean norm and we choose as ordered median function the one with $\lambda = (1, 1, 1, 1, 1)$ and $p = 2$, we get that:

$z_{3I}^R = 2.8348 < z_{OT}^R = 24.4140 < z_{BEP}^R = 34.2145 < z_{OPT}^* = 68.4751$

where $z_{OPT}^*$ is the optimal value of the OMPN.

### 3.6 Computational comparison of relaxations

We have run a series of experiments to study the computational performance of the formulations (OMP$_{3I}$), (OMP$_{2I}$), (OMP$_{OT}$) and (OMP$_{BEP}$) and also to know the computational limits of the OMPN. We have randomly generated instances of $n$ demand points in $[0, 100]^2$ and $[0, 100]^3$ with $n$ ranging in $\{5, 6, 7, 8, 9, 10, 20, 30\}$. Five random instances were generated for each number of points. We solved OMPN problems $p$ (number of new facilities to be located) ranging in $\{2, 3, 5\}$ (provided that $p < n$). Euclidean distances were considered to measure the distances between points. We considered neighborhoods defined as disks (for the planar instances) or 3-dimensional Euclidean balls (for the 3-dimensional instances) with randomly generated radii. The sizes of the neighborhoods were generated under four scenarios:

- **Scenario 1.** Radii generated in $[0, 5]$.
- **Scenario 2.** Radii generated in $[5, 10]$.
- **Scenario 3.** Radii generated in $[10, 15]$.
- **Scenario 4.** Radii generated in $[15, 20]$.

In Fig. 4 we show the neighborhoods for one of our 8-points instances under the four scenarios. In Scenario 1, the neighborhoods are small disks around the potential facilities while in Scenario 4 the neighborhoods are large ball that even overlap. Thus, the solutions obtained for the first scenario are expected to be similar (if not the same) to the classical DOMP. In contrast, in Scenario 4, the facilities are allowed to be positioned in almost the entire convex hull of the demand points. Hence, the solutions under this scenario should be similar to those of the continuous ordered median problem.

The set-up cost of each facility was assumed to be the radius of its neighborhood. It can be seen as the cost of locating the whole neighborhood (larger as $r$ increases).
The four formulations were coded in C, and solved using Gurobi 7.01 in a Mac OS X El Capitan with an Intel Core i7 processor at 3.3 GHz and 16GB of RAM. A time limit of 1 h was set in all the experiments.

Also, four different convex ordered median problems were solved for each of the instances:

- $p$-Median (M): $\lambda = (1, \ldots, 1)$.
- $p$-Center (C): $\lambda = (1, 0, \ldots, 0)$.
- $p$-KCenter (K): $\lambda = \left(\left\lfloor \frac{n}{2} \right\rfloor, \ldots, 1, 0, \ldots, 0\right)$.
- $p$-Cent-Dian0.5 (D): $\lambda = (1, 0.5, \ldots, 0.5)$.

For the small instances ($n \leq 10$) we report here only the results obtained for $n = 10$ (the interested reader may check the complete result of the experiments for $n = 5, \ldots, 10$ in bit.ly/resultsDOMPN).

First, we run the continuous relaxation of the four formulations to analyze their integrality gaps. The average results are shown in Table 1. We report the integrality gaps ($IG = \frac{z^*}{z^R}$) for each of the formulations [(OMP$3_{OT}$), (OMP$2_{OT}$) and (OMP$BE_{OT}$)] (obviating (OMP$OT$) whose integrality gaps coincide with those of (OMP$BE_{OT}$) by Theorem 3.1]. The table summarizes the average integrality gaps for the each of the scenarios. As remarked above, there is no order relation between $z^R_{21}$ and $z^R_{BE_{OT}}$. We have boldfaced those values in which the average integrality gaps of (OMP$2_{OT}$) is smaller than the one for (OMP$BE_{OT}$). Note that it only happens for a few combinations of $n$, $p$, problem types and scenarios. In particular, it mostly occurs for small values.
of \( p \) and only for Scenario 1. In Table 2 we show the percentage of instances (of all of those with fixed \( n \) and \( p \)) for which \( z_{21}^R \geq z_{BE}^R \). In the total instances, this percentage is 9.97\%, while among the instances of Scenario 1 is 26.15\%.

Tables 3 and 4 show the results for the planar and 3-dimensional problems, respectively. For each of the formulations, we provide the average consumed CPU times (time), the number of nodes explored in the branch-and-bound tree (#nodes) and the deviation with respect to the solution obtained at the end of the root node of the search tree (%gapR).

As can be observed from the results, formulations (OMPNO) and (OMPNB) are much less time consuming than (OMPNO3) and (OMPN2) in all the cases. Also, the solutions obtained after exploring the root node of (OMPNO) and (OMPNB) are tighten than the rest. Consequently, the number of explored nodes to find the optimal solution or to certify optimality is higher in the two first formulations. Observe that the results are as expected since the first two formulations do not exploit the convexity of monotone ordered median problems. Observe that the sorting constraints in the first two formulations involve binary variables while in the two last formulations no need of new binary variables are needed for this task.

Since (OMPNO) and (OMPNB) seems to have a similar computational behavior for the small-size instances, we have performed a series of experiments for medium-size instances to compare these two formulations. The results are shown in Table 5, where now \( n \) ranges in \{20, 30\} and \( p \) in \{2, 5, 10\}. Apart from the average CPU times (Times) and their standard deviations (StDev), we report the average percentage of instances which were not solved within the time limit of 1 h (%NS) for each of the formulations. As can be observed, the performance (in terms of CPU time) of both formulation is similar, but (OMPNB) seems to need, in average, less CPU time to solve the problems in most of the problems, and the standard deviations (StDev) of the consuming times for (OMPNB) are smaller than those for (OMPNO). Furthermore, we were able to solve all the instances before the 1 h time limit, but 2.56\% of them by using (OMPNB), while (OMPNO) was not able to solve 11.34\% of the the problems. Moreover, in all the instances, (OMPNB) obtained better upper bounds for the optimal value of the problems in all the instances (the deviation of the best upper bounds obtained with the OT formulation with respect to the best solution obtained with the BEP formulation is shown in column %DevBest).

**4 Math-heuristics for the OMPN**

In this section we describe two mathematical programming location-allocation based heuristic approaches for solving the OMPN for larger instances. Some heuristics have been proposed for solving ordered \( p \)-median problems (see [13]). However, most of them are based on the use of “fast” procedures to compute the overall cost of opening/closing certain sets of facilities. Note that when the travel cost matrix is provided and a set of open facilities is obtained, one can easily evaluate, for each customer, its cheapest facility (or its second cheapest facility), and once all of them are computed, evaluate an ordered median function can also be efficiently performed. In the OMPN, even if the open facilities are known, the allocation costs depend on
### Table 1  Integrality gaps for the formulations of DOMPN

| n | Pr. | p | Sc. 1 (%) | Sc. 2 (%) | Sc. 3 (%) | Sc. 4 (%) |
|---|-----|---|----------|----------|----------|----------|
|   |     |   | IG₃ | IG₂ | IGₛₑₚ | IG₃ | IG₂ | IGₛₑₚ | IG₃ | IG₂ | IGₛₑₚ | IG₃ | IG₂ | IGₛₑₚ |
| 10 | C   | 2  | 3204 | 442 | 631   | 3131| 605 | 569   | 998 | 644 | 383   | 1796| 538 | 371   |
|    | 3   | 1942 | 483 | 505 | 2103| 561 | 466 | 582 | 444 | 296 | 874 | 418 | 297   |
|    | 4   | 1115 | 448 | 408 | 1289| 472 | 387 | 358 | 323 | 234 | 458 | 311 | 233   |
|    | 5   | 773  | 404 | 340 | 740 | 404 | 328 | 258 | 249 | 195 | 310 | 247 | 194   |
| D  | 2   | 10, 294 | 381 | 629 | 9961| 1031| 613 | 3029| 1952| 528 | 5391| 1604| 503   |
|    | 3   | 5303 | 610 | 510 | 5410| 1315| 493 | 1430| 1089| 399 | 2246| 1062| 395   |
|    | 4   | 2780 | 742 | 430 | 3047| 1074| 420 | 752 | 679 | 314 | 995 | 671 | 313   |
|    | 5   | 1615 | 810 | 369 | 1497| 791 | 355 | 449 | 432 | 249 | 555 | 438 | 250   |
| K  | 2   | 12, 847 | 521 | 797 | 12, 350| 1335| 767 | 3705| 2390| 652 | 6558| 1953| 606   |
|    | 3   | 7154 | 860 | 697 | 7285| 1777| 651 | 1880| 1433| 509 | 2898| 1369| 475   |
|    | 4   | 4010 | 1098| 609 | 4333| 1524| 551 | 1032| 931 | 399 | 1336| 900 | 378   |
|    | 5   | 2439 | 1219| 527 | 2235| 1178| 463 | 634 | 610 | 319 | 759 | 599 | 297   |
| M  | 2   | 17, 160 | 371 | 638 | 16, 432| 1343| 630 | 4947| 3187| 580 | 8594| 2554| 551   |
|    | 3   | 8493 | 719 | 516 | 8610| 1989| 510 | 2217| 1688| 445 | 3436| 1622| 433   |
|    | 4   | 4362 | 1005| 448 | 4716| 1659| 440 | 1117| 1008| 355 | 1440| 970 | 345   |
|    | 5   | 2439 | 1209| 393 | 2235| 1178| 375 | 634 | 610 | 284 | 759 | 599 | 278   |
the final location of the facilities (which depends also on the customers allocated to each facility). Hence, the previously developed heuristics for the DOMP are no longer directly applicable to the OMPN problem. We propose two alternative local search math-heuristic which allows us to solve larger instances of the problem at smaller computational costs than the exact approaches, at the price of not warranting the optimality of the solution.

The two heuristics procedures that we propose are part of the well-known family of location-allocation procedures. These type of schemes based on performing changes over a set of $p$ facilities candidates for being opened, trying to improve the incumbent objective value. For the sake of that, a crucial step is to compute the overall cost of opening a given set of facilities $J \subseteq \{1, \ldots, n\}$, with $|J| = p$. Observe that if a set of open facilities is known, we have to compute the minimum (ordered weighted) cost of allocating the customers to those facilities. As mentioned above the computation of such a cost will involve the computation of the allocation customer-open facility, which also depends of the position of the open facilities inside their neighborhoods. Although different formulations can be used for such a task, we will use the one based on formulation (OMPNBEP).

**Proposition 4.1** Let $J \subset I = \{1, \ldots, n\}$ with $|J| = p$. Then, the cost of using $J$ as the set of open facilities of the OMPN problem can be computed by solving the following mixed integer non linear programming problem:

$$c(J) := \min \left\{ \sum_{i \in I \setminus J} u_i + \sum_{k \in I \setminus J} v_k + \sum_{j \in J} f_j \right\}$$

s.t. $u_i + v_k \geq \lambda_k z_i, \forall i, k \in I \setminus J,$
$$z_i \geq d_{ij} - \hat{D}_{ij}(1 - x_{ij}), \forall i \in I \setminus J, j \in J,$$
$$d_{ij} \geq \|a_i - \bar{a}_j\|, \forall i \in I \setminus J, j \in J,$$
$$r_j \geq \|a_j - \bar{a}_j\|, \forall j \in J,$$
$$\sum_{j \in J} x_{ij} = 1, \forall i \in I \setminus J,$$
$$x_{ij} \in \{0, 1\}, \forall i \in I \setminus J, j \in J,$$
$$z_i \geq 0, \forall i \in I \setminus J,$$
$$\bar{a}_j \in \mathbb{R}^d, \forall j \in J,$$
$$d_{ij} \in \mathbb{R}_+, \forall i \in N \setminus J, j \in J,$$
$$u_i, v_i \in \mathbb{R}, \forall i \in I.$$
| Sc. | Pr. | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) |
|-----|-----|------|--------|----------|------|--------|----------|------|--------|----------|------|--------|----------|
| 1 C | 2   | 217.19 | 118,505 | 96.71 | 26.73 | 51,583 | 85.01 | 0.39 | 79 | 32.97 | 0.21 | 56 | 33.88 |
| 3   | 766.09 | 391,417 | 93.85 | 139.82 | 268,504 | 84.06 | 0.37 | 159 | 38.90 | 0.26 | 103 | 36.97 |
| 4   | 3375.43 | 872,945 | 88.82 | 1088.98 | 16,24,482 | 78.99 | 0.31 | 180 | 39.78 | 0.19 | 54 | 36.34 |
| 5   | 3527.08 | 1,109,172 | 83.98 | 2225.30 | 4,064,066 | 75.20 | 0.29 | 217 | 45.23 | 0.18 | 87 | 39.07 |
| 2 D | 78.49 | 17,176 | 98.90 | 20.82 | 24,835 | 86.82 | 0.35 | 55 | 9.17 | 0.26 | 34 | 4.49 |
| 3   | 142.18 | 5,336 | 95 | 22.76 | 40,081 | 90.40 | 0.34 | 4 | 8.86 | 0.15 | 0 | 0 |
| 4   | 68.88 | 5336 | 95 | 22.76 | 40,081 | 90.40 | 0.34 | 4 | 8.86 | 0.15 | 0 | 0 |
| 5   | 79.83 | 10,692 | 91.43 | 147.91 | 304,842 | 86.54 | 0.43 | 45 | 15.07 | 0.27 | 18 | 8.26 |
| 2 K | 291.01 | 136,621 | 99.14 | 113 | 184,032 | 90.47 | 0.28 | 69 | 26.14 | 0.34 | 88 | 23.22 |
| 3   | 2239.44 | 961,277 | 98.20 | 457.96 | 580,536 | 95.25 | 0.32 | 135 | 31.15 | 0.30 | 103 | 26.99 |
| 4   | 3079.66 | 745,095 | 96.55 | 1538.17 | 1,946,764 | 93.38 | 0.36 | 169 | 37.92 | 0.31 | 123 | 32.72 |
| 5   | 613.05 | 246,909 | 94.09 | 4081.56 | 3,501,105 | 90.72 | 0.38 | 259 | 42.68 | 0.34 | 183 | 36.29 |
| 2 M | 45.98 | 9877 | 99.32 | 7.32 | 7,234 | 86.93 | 0.39 | 26 | 8.19 | 0.26 | 5 | 2.21 |
| 3   | 35.95 | 6140 | 98.43 | 10.44 | 6924 | 95.83 | 0.23 | 6 | 2.43 | 0.16 | 1 | 0.71 |
| 4   | 32.07 | 6413 | 96.76 | 3.45 | 4077 | 93.77 | 0.24 | 26 | 7.82 | 0.15 | 0 | 0 |
| 5   | 24.75 | 4168 | 94.09 | 4.42 | 4489 | 90.72 | 0.27 | 9 | 5.37 | 0.14 | 0 | 0 |
| 2 C | 185.79 | 243,222 | 97.06 | 35.65 | 160,015 | 87.11 | 0.18 | 100 | 33.21 | 0.12 | 62 | 33.54 |
| 3   | 3368.98 | 2,751,134 | 94.71 | 157.57 | 564,670 | 82.55 | 0.18 | 203 | 38.36 | 0.15 | 165 | 31.62 |
| 4   | 3056.51 | 152,278 | 90.92 | 557.25 | 1,636,689 | 77.63 | 0.18 | 255 | 38.03 | 0.12 | 114 | 28.83 |
| 5   | 312.58 | 319,355 | 87.29 | 334.03 | 1,082,684 | 73.49 | 0.18 | 234 | 39 | 0.12 | 140 | 26.26 |
| 2 D | 40.02 | 20,172 | 95 | 6.64 | 232,79 | 95.43 | 0.13 | 38 | 13.76 | 0.14 | 31 | 8.55 |
| 3   | 76.92 | 30,308 | 97.87 | 11.71 | 34,052 | 92.74 | 0.13 | 22 | 10.90 | 0.09 | 12 | 7.57 |
Table 3 continued

| Sc. | Pr. | p   | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) |
|-----|-----|-----|------|--------|-----------|------|--------|-----------|------|--------|-----------|------|--------|-----------|
|     |     |     |      |        |           |      |        |           |      |        |           |      |        |           |
| 4   |     | 93.42 | 15,935 | 95.85 | 10.59 | 39,267 | 89.51 | 0.14 | 11   | 7.97 | 0.13 | 0   | 4.02 |
| 5   |     | 66.53 | 15,954 | 90.74 | 14.53 | 70,825 | 85.35 | 0.16 | 40   | 14.52 | 0.13 | 0   | 2.12 |
| K   |     | 257.25 | 239,132 | 99.21 | 94.28 | 289,128 | 96.41 | 0.14 | 101 | 27.71 | 0.16 | 105 | 25.70 |
| 3   |     | 1804.51 | 1,410,405 | 98.44 | 320.74 | 836,508 | 94.71 | 0.13 | 116 | 32.21 | 0.21 | 177 | 29.38 |
| 4   |     | 2053.02 | 1,469,281 | 97.08 | 724.58 | 1,705,800 | 92.64 | 0.17 | 177 | 37.55 | 0.19 | 185 | 34.11 |
| 5   |     | 833.50 | 774,058 | 93.53 | 3164.10 | 6,317,876 | 89.75 | 0.16 | 190 | 38.59 | 0.18 | 245 | 34.51 |
| M   |     | 37.72 | 18,920 | 99.38 | 5.01 | 9075 | 97.17 | 0.16 | 42 | 11.33 | 0.15 | 28 | 7.21 |
| 3   |     | 27   | 11,529 | 98.63 | 4.66 | 7610 | 95.35 | 0.17 | 4 | 4.28 | 0.09 | 1 | 1.78 |
| 4   |     | 24.24 | 11,298 | 97.26 | 4.86 | 12168 | 93.08 | 0.19 | 59 | 4.15 | 0.12 | 1 | 0.23 |
| 5   |     | 15.07 | 5040 | 93.53 | 2.46 | 5522 | 89.75 | 0 | 46 | 8.47 | 0.12 | 5 | 3.31 |
| 3 C |     | 173.15 | 24,7260 | 88.93 | 10.57 | 31894 | 83.40 | 0.14 | 52 | 38.77 | 0.10 | 59 | 29.83 |
| 3   |     | 1319.33 | 1,565,352 | 79.43 | 87.54 | 246,224 | 74 | 0.13 | 97 | 39.18 | 0.10 | 56 | 24.94 |
| 4   |     | 1143.93 | 996,110 | 65.65 | 128.58 | 429,914 | 62.69 | 0.14 | 117 | 34.17 | 0.09 | 47 | 18.47 |
| 5   |     | 1035.39 | 1,506,863 | 53.49 | 190.32 | 751,796 | 52.24 | 0.17 | 206 | 32.27 | 0.14 | 150 | 17.85 |
| D   |     | 48.22 | 36,372 | 96.06 | 4.12 | 9171 | 94.09 | 0.13 | 62 | 33.80 | 0.09 | 42 | 15.04 |
| 3   |     | 36.70 | 20,929 | 90.86 | 3.89 | 10671 | 88.43 | 0.12 | 60 | 37.01 | 0.12 | 18 | 8.69 |
| 4   |     | 340.46 | 506,451 | 81.77 | 4.29 | 11,882 | 80.20 | 0.10 | 74 | 40.74 | 0.13 | 3 | 3.01 |
| 5   |     | 83.84 | 172,707 | 69.92 | 5.66 | 23,412 | 69.08 | 0.13 | 88 | 35.34 | 0.16 | 22 | 4.15 |
| K   |     | 558.29 | 466004 | 96.85 | 131.97 | 270,282 | 95.27 | 0.12 | 49 | 41.50 | 0.12 | 42 | 27.59 |
| 3   |     | 653.97 | 510,418 | 93.17 | 285.01 | 572,239 | 91.36 | 0.14 | 101 | 48.30 | 0.15 | 91 | 29.63 |
| 4   |     | 309.64 | 307,686 | 86.81 | 1573.06 | 2,854,817 | 85.69 | 0.16 | 163 | 54.60 | 0.14 | 105 | 31.47 |
| 5   |     | 248.32 | 247,774 | 77.51 | 3732.59 | 1,571,394 | 76.88 | 0.17 | 224 | 55.83 | 0.16 | 147 | 28.04 |
| Sc. | Pr. | p   | Time  | #nodes | %gapR (%) | Time  | #nodes | %gapR (%) | Time  | #nodes | %gapR (%) | Time  | #nodes | %gapR (%) |
|-----|-----|-----|-------|--------|-----------|-------|--------|-----------|-------|--------|-----------|-------|--------|-----------|
| M   | 2   | 50  | 37663 | 97.54  | 3.95      | 6822  | 96.30  | 0.16      | 80    | 37.33  | 0.11      | 56    | 16.12  |
|     | 3   | 35.68 | 21,078 | 93.93 | 3.25 | 4576 | 92.31 | 0.11 | 59 | 41.43 | 0.15 | 17 | 9.80 |
|     | 4   | 22.94 | 16,293 | 87.33 | 1.94 | 3479 | 86.24 | 0.14 | 58 | 48.34 | 0.14 | 3 | 5.70 |
|     | 5   | 167.96 | 230,957 | 77.51 | 1.50 | 1669 | 76.88 | 0.14 | 93 | 43.84 | 0.14 | 11 | 3.94 |
| 4   | C   | 2   | 203.77 | 245,152 | 94.31 | 49.20 | 127,865 | 82.70 | 0.14 | 139 | 33.46 | 0.10 | 82 | 26.93 |
|     | 3   | 1201.06 | 1,114,765 | 86.88 | 529.54 | 728,381 | 76.31 | 0.17 | 249 | 32.73 | 0.08 | 63 | 8.65 |
|     | 4   | 323.95 | 498,328 | 73.29 | 353.45 | 676,140 | 64.98 | 0.14 | 126 | 18.48 | 0.01 | 243 | 28.73 |
|     | 5   | 1542.56 | 110,782 | 62.29 | 54.46 | 151,331 | 56.15 | 0.13 | 217 | 23.09 | 0.07 | 51 | 6.05 |
| D   | 2   | 423.35 | 210,911 | 97.95 | 13.32 | 29,156 | 93.49 | 0.19 | 166 | 32.28 | 0.17 | 174 | 29.19 |
|     | 3   | 396.21 | 211,169 | 94.31 | 24.84 | 50,853 | 89.21 | 0.21 | 241 | 32.73 | 0.01 | 243 | 28.73 |
|     | 4   | 213.08 | 121,462 | 86.61 | 18.59 | 45,899 | 81.77 | 0.11 | 277 | 26.03 | 0.16 | 244 | 21.95 |
|     | 5   | 65.49  | 49704  | 76.53 | 20.74 | 54,514 | 72.06 | 0.17 | 160 | 18.81 | 0.14 | 168 | 14.17 |
| K   | 2   | 700.32 | 457,647 | 98.34 | 349.44 | 456,972 | 94.73 | 0.16 | 158 | 40.93 | 0.18 | 136 | 38.28 |
|     | 3   | 2865.25 | 1667,538 | 95.59 | 643.49 | 870,765 | 91.64 | 0.23 | 296 | 40.93 | 0.24 | 248 | 37.75 |
|     | 4   | 2825.25 | 736,735 | 90.02 | 3361.02 | 4,465,105 | 86.42 | 0.26 | 404 | 38.85 | 0.22 | 304 | 35.83 |
|     | 5   | 1753.15 | 257,832 | 82.12 | 5823 | 2,160,439 | 78.71 | 0.23 | 372 | 32.61 | 0.20 | 350 | 29.26 |
| M   | 2   | 576   | 281,412 | 98.67 | 6.96 | 12,096 | 95.78 | 0.17 | 150 | 38.42 | 0.16 | 133 | 33.23 |
|     | 3   | 699.11 | 296,994 | 96.20 | 4.90 | 7174 | 92.80 | 0.22 | 313 | 41.03 | 0.17 | 272 | 35.29 |
|     | 4   | 170.92 | 109,314 | 90.54 | 6.66 | 12,752 | 87.12 | 0.19 | 306 | 36.70 | 0.17 | 280 | 30.04 |
|     | 5   | 138.54 | 102,782 | 82.12 | 5.74 | 13,638 | 78.71 | 0.16 | 209 | 29.09 | 0.20 | 433 | 22.64 |
Table 4: Average results of comparing the formulations for the 3-dimensional instances with $n = 10$.

| Sc. | Pr. | $p$ | (OMP$N_{31}$) | Time | #nodes | %gap(%) | (OMP$N_{21}$) | Time | #nodes | %gap(%) | (OMP$O T$) | Time | #nodes | %gap(%) | (OMP$R E P$) | Time | #nodes | %gap(%) |
|-----|-----|-----|--------------|------|--------|---------|--------------|------|--------|---------|------------|------|--------|---------|------------|------|--------|---------|
| 1   | C   | 2   | 473.84      | 368360 | 97.22  | 35.74   | 52967       | 74.37 | 0.24   | 82      | 24.03      | 0.15 | 79      | 25.17   | 3842.67    | 611,415 | 89.13  | 0.25   | 203     | 41.80    | 0.14 | 147    | 41.37   |
| 2   | D   | 2   | 61.91       | 18,930 | 99.19  | 88.55   | 43,946      | 68.97 | 0.13   | 6       | 2.99       | 0.15 | 2       | 0.91    | 371.31      | 47,028 | 98.49  | 0.18   | 29      | 6.27     | 0.14 | 31     | 3.96    |
| 3   | K   | 2   | 511.77      | 225,792 | 99.33 | 81.79   | 222,769     | 76.49 | 0.22   | 138     | 19.71      | 0.22 | 146     | 17.53   | 4807.08     | 995,386 | 98.87 | 0.23   | 176     | 31.58    | 0.26 | 235    | 29.14   |
| 4   | M   | 2   | 57.47       | 18,665 | 99.52 | 14.19   | 13,224      | 67.92 | 0.21   | 32      | 3.85       | 0.15 | 4       | 0.99    | 5913.30     | 116,629 | 98.05 | 0.27   | 261     | 39.21    | 0.27 | 226    | 36.34   |
| 5   |     |     | 4950.00     | 324,797 | 96.85 | 98.05   | 2,442,139   | 93.58 | 0.21   | 224     | 45.21      | 0.23 | 308     | 41.78   | 5913.30     | 116,629 | 98.05 | 0.27   | 261     | 39.21    | 0.27 | 226    | 36.34   |
| 2   | C   | 2   | 698.00      | 582674 | 97.05 | 52.27   | 98,390      | 84.39 | 0.15   | 50      | 23.94      | 0.13 | 67      | 25.04   | 6363.00     | 103,467 | 95.60 | 0.18   | 123     | 32.22    | 0.12 | 56     | 33.39   |
| 3   | D   | 2   | 26.97       | 13871  | 99.14 | 10.64   | 2,1167      | 91.26 | 0.13   | 19      | 7.30       | 0.16 | 10      | 5.64    | 5647.00     | 3688.955 | 82.04 | 0.20   | 176     | 36.36    | 0.17 | 130    | 35.83   |
| 4   |     |     | 26.09       | 16012  | 98.48 | 7.30    | 14,069      | 94.35 | 0.12   | 28      | 7.39       | 0.14 | 39      | 5.16    | 5778.93     | 20217 | 88.74 | 0.20   | 179     | 38       | 0.11 | 56     | 35.64   |
| 5   |     |     | 45.70       | 10844  | 97.38 | 18.50   | 30,920      | 92.60 | 0.13   | 55      | 8.75       | 0.13 | 38      | 7.08    | 23.17       | 4170   | 94.76 | 0.16   | 88      | 9.53     | 0.17 | 13     | 4.38    |
Table 4 continued

| Sc. | Pr. | p   | (OMP\textsubscript{N1}) | (OMP\textsubscript{N2}) | (OMP\textsubscript{OT}) | (OMP\textsubscript{REP}) |
|-----|-----|-----|-----------------|-----------------|-----------------|-----------------|
|     |     |     | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) | Time | #nodes | %gapR (%) |
| K   | 2   | 1785.21 | 167030 | 99.28 | 255.95 | 299,773 | 93.07 | 0.13 | 41 | 20.37 | 0.16 | 40 | 18.82 |
|     | 3   | 3734.82 | 368152 | 98.86 | 2184.15 | 792,684 | 95.79 | 0.16 | 86 | 32.39 | 0.16 | 78 | 30.66 |
|     | 4   | 2497.71 | 540841 | 98.16 | 4728.84 | 1,716,593 | 94.82 | 0.17 | 137 | 38.34 | 0.17 | 121 | 36.34 |
|     | 5   | 3099.23 | 160480 | 96.57 | 6860.23 | 3,165,215 | 93.49 | 0.19 | 188 | 43.53 | 0.18 | 163 | 41.23 |
| M   | 2   | 22.84 | 12370 | 99.49 | 5.46 | 7523 | 92.07 | 0.07 | 0 | 0 | 0 | 0 | 0 |
|     | 3   | 25.44 | 11016 | 99.06 | 4.84 | 7306 | 95.50 | 0.10 | 19 | 6.12 | 0.16 | 17 | 3.92 |
|     | 4   | 16.44 | 7086 | 98.33 | 2.74 | 4104 | 95.30 | 0.14 | 19 | 6.04 | 0.13 | 18 | 3.70 |
|     | 5   | 12.70 | 2887 | 96.57 | 2.24 | 2401 | 94.90 | 0.15 | 18 | 1.56 | 0.10 | 3 | 0.71 |
| 3   | C   | 420.03 | 251,532 | 90.68 | 44.81 | 67,897 | 84.83 | 0.21 | 75 | 28.92 | 0.12 | 63 | 22.81 |
|     | 3   | 4166.57 | 629,476 | 85.05 | 2144.77 | 627,453 | 79.66 | 0.24 | 161 | 36.97 | 0.18 | 172 | 29.87 |
|     | 4   | 3496.18 | 360,878 | 76.21 | 2907.10 | 1,019,207 | 73.11 | 0.30 | 273 | 37.35 | 0.18 | 183 | 27.55 |
|     | 5   | 4390.44 | 566,911 | 66.78 | 5770.53 | 1,238,536 | 64.98 | 0.28 | 307 | 37.85 | 0.16 | 181 | 26.25 |
| D   | 2   | 88.31 | 43,064 | 97.12 | 15.64 | 27,226 | 95.31 | 0.14 | 43 | 24.16 | 0.15 | 37 | 10.38 |
|     | 3   | 257.27 | 72,613 | 94.29 | 42.11 | 46,924 | 92.25 | 0.25 | 191 | 27.63 | 0.17 | 95 | 12.43 |
|     | 4   | 524.51 | 104,272 | 89.46 | 310.57 | 133,618 | 88.06 | 0.25 | 229 | 31.88 | 0.21 | 150 | 11.61 |
|     | 5   | 284.34 | 89,650 | 82.26 | 161.15 | 228,276 | 81.29 | 0.30 | 324 | 34.84 | 0.22 | 236 | 10.35 |
| K   | 2   | 2833.08 | 429,198 | 97.59 | 273.07 | 318,542 | 96.08 | 0.18 | 70 | 31.86 | 0.15 | 50 | 21.16 |
|     | 3   | 5614.46 | 389,241 | 95.60 | 2917.04 | 1,364,077 | 94.03 | 0.28 | 219 | 43.42 | 0.25 | 200 | 31.08 |
|     | 4   | 5881.77 | 60,182 | 92.30 | 6243.04 | 1,470,989 | 91.30 | 0.37 | 342 | 50.21 | 0.30 | 347 | 36.06 |
|     | 5   | 5833.66 | 40,320 | 87.79 | 7200.04 | 1 | 87.13 | 0.33 | 357 | 55.10 | 0.30 | 383 | 39.35 |
| Sc. | Pr. | p  | Time   | #nodes | %gap (%) | Time   | #nodes | %gap (%) | Time   | #nodes | %gap (%) | Time   | #nodes | %gap (%) |
|-----|-----|----|--------|--------|----------|--------|--------|----------|--------|--------|----------|--------|--------|----------|
| M   | 2   | 116.51 | 50,768 | 98.27  | 6.31     | 9523   | 97     | 0.14     | 55     | 28     | 10.32    | 0.14   | 28     | 10.32    |
|     | 3   | 174.75 | 52,089 | 96.39  | 50.50    | 19,088 | 95.10  | 0.23     | 196    | 71     | 8.60     | 0.17   | 71     | 8.60     |
|     | 4   | 151.05 | 53,351 | 93.05  | 11.74    | 13,976 | 92.15  | 0.23     | 232    | 94     | 8.53     | 0.14   | 94     | 8.53     |
|     | 5   | 74.87  | 32,522 | 87.79  | 9.07     | 12,475 | 87.13  | 0.26     | 281    | 158    | 8.97     | 0.14   | 158    | 8.97     |
| 4   | C   | 535.61 | 315,885| 94.76  | 64.32    | 98,609 | 83.19  | 0.17     | 101    | 90     | 21.68    | 0.16   | 90     | 21.68    |
|     | 3   | 3061.28| 171,458| 89.60  | 1463.17  | 148,985| 78.88  | 0.19     | 132    | 146    | 20.10    | 0.19   | 146    | 20.10    |
|     | 4   | 3104.27| 151,466| 81.17  | 1812.87  | 700,664| 73.08  | 0.21     | 181    | 213    | 19.90    | 0.26   | 213    | 19.90    |
|     | 5   | 3058.78| 128,605| 71.34  | 2007.26  | 496,927| 64.75  | 0.18     | 175    | 110    | 15.01    | 0.12   | 110    | 15.01    |
| K   | 2   | 688.65 | 156,903| 98.35  | 67.95    | 62,430 | 94.61  | 0.22     | 179    | 221    | 19.05    | 0.26   | 221    | 19.05    |
|     | 3   | 1375.45| 231,147| 96.31  | 47.77    | 45,659 | 92.40  | 0.28     | 278    | 258    | 18.76    | 0.25   | 258    | 18.76    |
|     | 4   | 683    | 149878 | 91.86  | 206.62   | 98,633 | 88.25  | 0.31     | 354    | 355    | 18.26    | 0.32   | 355    | 18.26    |
|     | 5   | 267.16 | 84,209 | 85.28  | 187.50   | 221,145| 81.69  | 0.26     | 334    | 338    | 15.05    | 0.25   | 338    | 15.05    |
| M   | 2   | 4411.18| 838,744| 98.63  | 772.68   | 641,139| 95.51  | 0.27     | 215    | 205    | 26.63    | 0.26   | 205    | 26.63    |
|     | 3   | 6083.69| 348,711| 97.14  | 5170.44  | 649,666| 94.11  | 0.27     | 248    | 363    | 34.72    | 0.41   | 363    | 34.72    |
|     | 4   | 6244.01| 434,725| 93.94  | 7200.13  | 1       | 91.25  | 0.39     | 384    | 410    | 37.34    | 0.35   | 410    | 37.34    |
|     | 5   | 6069.28| 142,560| 89.59  | 7200.06  | 1       | 86.93  | 0.39     | 493    | 484    | 37.04    | 0.37   | 484    | 37.04    |
| M   | 2   | 1010.07| 173,718| 99     | 19.32    | 23,381 | 96.71  | 0.21     | 183    | 172    | 23.34    | 0.22   | 172    | 23.34    |
|     | 3   | 2168.18| 133,674| 97.75  | 33.67    | 26,616 | 95.12  | 0.30     | 296    | 237    | 21.16    | 0.26   | 237    | 21.16    |
|     | 4   | 373.24 | 98,523 | 94.47  | 17.35    | 28,016 | 92.02  | 0.35     | 441    | 346    | 19.04    | 0.29   | 346    | 19.04    |
|     | 5   | 1567.87| 43,192 | 89.60  | 21.58    | 33,498 | 86.93  | 0.28     | 323    | 332    | 16.50    | 0.24   | 332    | 16.50    |
Table 5 Comparison of \((\text{OMPN}_{OT})\) and \((\text{OMPN}_{BEP})\) for instances with \(n = 20, 30\)

| Sc. | \(n\) | \(p\) | Time\(_{BEP}\) | StDev\(_{BEP}\) | Time\(_{OT}\) | StDev\(_{OT}\) | %NS\(_{BEP}\)(%) | %NS\(_{OT}\)(%) | %DevBest (%) |
|-----|------|-----|--------------|--------------|--------------|--------------|----------------|----------------|---------------|
| 1   | 20   | 2   | 1.16         | 0.32         | 1.05         | 0.51         | 0              | 0              | 0             |
|     | 5    | 2.37 | 1.38         | 4.70         | 6.01         | 0             | 0              | 0              | 0             |
|     | 10   | 3.31 | 4.85         | 18.32        | 65.64        | 0             | 0              | 0              | 0             |
| 30  | 2    | 14.41 | 8.13         | 8.36         | 9.94         | 0             | 0              | 0              | 0             |
|     | 5    | 52.78 | 61.72        | 146.09       | 567.29       | 0             | 2.5            | 0.03           | 0.16          |
|     | 10   | 285.12 | 782.56       | 519.45       | 1045.90      | 2.56          | 5.13           | 0.00           |               |
| 2   | 20   | 2    | 90.48        | 505.86       | 83.47        | 505.36        | 0              | 0              | 0             |
|     | 5    | 927.69 | 1458.47      | 1032.65      | 1539.87      | 0              | 15             | 1              |               |
|     | 10   | 16.52 | 28.31        | 228.26       | 795.27       | 0              | 5              | 0.20           |               |
| 30  | 2    | 38.05 | 23.53        | 61.03        | 68.34        | 0              | 0              | 0              |               |
|     | 5    | 796.67 | 949.04       | 1361.34      | 1348.09      | 5              | 17.50          | 1.26           |               |
|     | 10   | 1717.63 | 1664.82      | 1960.70      | 1608.83      | 42.50          | 42.50          | 3.82           |               |
| 3   | 20   | 2    | 9.73         | 2.71         | 11.36        | 3.58          | 0              | 0              | 0             |
|     | 5    | 253.35 | 18.65        | 449.32       | 28.15        | 0              | 2.56           | 0.01           |               |
|     | 10   | 46.97 | 13.85        | 77.52        | 17.47        | 0              | 0              | 0              |               |
| 30  | 2    | 59.64 | 6.48         | 148.23       | 14.70        | 0              | 0              | 0              |               |
|     | 5    | 2931.44 | 36.11        | 3099.25      | 32.91        | 75             | 77.5           | 1.63           |               |
|     | 10   | 2861.03 | 37.34        | 3070.86      | 34.40        | 77.5           | 80             | 3.75           |               |
| 4   | 20   | 2    | 26.45        | 4.40         | 30.03        | 4.95          | 0              | 0              | 0             |
|     | 5    | 1865.88 | 41.15        | 1874.01      | 41.33        | 40             | 40             | 0.30           |               |
|     | 10   | 9.51  | 2.44         | 22.13        | 6.17         | 0              | 0              | 0              |               |
| 30  | 2    | 735.58 | 36.69        | 849.51       | 37.10        | 15             | 15             | 0.17           |               |
|     | 5    | 2742.49 | 39.04        | 2836.95      | 36.91        | 75             | 75             | 1.15           |               |
|     | 10   | 2745.59 | 39.12        | 2789.27      | 38.12        | 75             | 75             | 3.28           |               |

**Proof** The proof easily follows noting that \((\text{ALLOC}(J))\) is nothing but the simplification of \((\text{OMPN}_{BEP})\) when the values of \(x_{jj}\) are known and fixed to \(x_{jj} = 1\) if \(j \in J\) and \(x_{jj} = 0\), otherwise.

For each fixed set of open facilities, i.e. \(J \subset I = \{1, \ldots, n\}\) with \(|J| = p\), \((\text{ALLOC}(J))\) can be reformulated as a mixed integer second order cone constraint problem with \((n - p)p\) binary variables (instead of the \(n^2\) in \((\text{OMPN}_{BEP})\)). Furthermore, a variable fixing strategy can be applied to \((\text{ALLOC}(J))\) in order to reduce the number of binary variables of the problem as described in the following result:

**Properties 4.1** Let \(J \subseteq N\) with \(|J| = p\), \(i \in I \setminus J\), \(j \in J\) and \(x^* \in \mathcal{X}\) optimal allocation solution of \((\text{ALLOC}(J))\).

1. If \(\exists k \notin J\) such that \(\hat{D}_{ik} < \hat{d}_{ij}\), then \(x^*_{ij} = 0\).
2. If \(\min_{k \neq j} \hat{d}_{ik} > \hat{D}_{ij}\), then \(x^*_{ij} = 1\).
3. If \(\mathcal{J}_i = \{j' \in J : \hat{D}_{ik} \geq \hat{d}_{ij'}, \forall k \neq j'\} = \{j\}\), then \(x^*_{ij} = 1\).
Ordered \( p \)-median problems with neighbourhoods

\[
x_{ij}^* = 0 \quad \text{if} \quad i \neq j
\]

**Proof**

1. Let us assume that \( x_{ij}^* = 1 \). Then, \( d_{ij}^* = \| a_i - \bar{a}_j \| \geq \hat{d}_{ij} \). By hypothesis there exists \( k \in J \) \( (k \neq j) \) with \( \hat{D}_{ik} < \hat{d}_{ij} \). Hence, \( d_{ij}^* > \hat{D}_{ik} \geq d_{ik} = \| a_i - \bar{a}_k \| \), so \( d_{ij}^* \neq \min_{j' \in J} \| a_i - \bar{a}_k \| \) contradicting the optimality of the solution.

2. If \( x_{ij}^* = 0 \), then, there exists \( k \in J \), \( k \neq j \) such that \( d_{ij}^* \geq d_{ik}^* > \hat{D}_{ij} \geq d_{ij}^* \). Thus, \( x_{ij}^* = 1 \).

3. Let \( j' \in J \setminus \{ j \} \), by hypothesis, since \( j' \notin \mathcal{J}_i \), there exists \( k \in J \setminus \{ j' \} \) such that \( \hat{D}_{ik} < \hat{d}_{ij'} \). Thus, by 1 we get that \( x_{ij'}^* = 0 \) for all \( j' \in J \setminus \{ j \} \). Since all demand points must be allocated to a facility, we get that \( x_{ij} \) must be one. \( \square \)

The situations described in Proposition 4.1, and that allow us to fix some of the variables in (ALLOC(J)), are illustrated in Fig. 5. There, we assume that the three disk-shaped neighborhoods are the open facilities \((j, k \text{ and } k')\), while the dot \((i)\) is one of the demand points \((a_i)\). In the left picture, the smallest possible distance from \( i \) to the \( j \)th neighborhood is larger than the largest possible distance to the \( k \)th neighborhood. Thus, \( i \) will never be allocated to \( j \) \((x_{ij}^* = 0)\). In the picture in the center, the minimum of the smallest possible distances between \( i \) and both the neighborhoods \( k \) and \( k' \) is greater than the largest possible distance to the \( j \)th neighborhood. Then, \( j \) will always be closer to \( i \) than the rest of the open facilities, and \( i \) is allocated to \( j \) \((x_{ij}^* = 1)\). Finally, in the right picture, one can check that among the three open facilities \((j, k \text{ and } k')\), the only one that verifies that the smallest possible distance from \( i \) to the neighborhood is smaller than the larger possible distance to the rest of the neighborhoods, is \( j \), and then, \( i \) must be allocated to \( j \) \((x_{ij}^* = 1)\).

As we will show in our computational experiments, the above strategies for fixing variables allows us to fix an average of 80% of the binary variables in the test problems.

Using the above-described formulation, (ALLOC(J)), we implemented two different heuristic algorithms. Both algorithms will move through different feasible solutions in order to improve an initial feasible one. This initial solution is constructed by either solving the standard DOMP with a set of weights based on the distances between centers of the neighborhoods (a convex combination of \( \hat{D}_{ij} \) and \( \hat{d}_{ij} \)), or solving the OMPN problem for simpler neighborhoods (as polyhedral neighborhoods) and polyhedral distances (which may require less computational effort than general \( \ell_\tau \)-norm
based metrics or neighborhoods). Hence, we consider that an initial solution \( x^0 \in \mathcal{X} \) is known.

4.1 Math-heuristic Algorithm 1

Let us denote \( I = \{1, \ldots, n\} \) the index set of customers/facilities. Given a feasible solution \( \bar{x} \in \mathcal{X} \), the first algorithm searches, for each facility \( j_0 \) in \( J \), the best replacement by a facility in \( \mathcal{X} \). Two different options are possible here:

1. Solve \( \text{ALLOC}(J \cup \{j\} \setminus \{j_0\}) \) for each \( j \in I \setminus J \). It will find the cost of replacing \( j_0 \) by \( j \). Moving through all \( j \in I \setminus J \) and keeping the lower cost provides us the best replacement for \( j_0 \). It requires solving \((n-p)\) problems, or

2. Solve the following mixed integer non linear programming problem which decides, through the binary variable \( \rho_j = \begin{cases} 1 & \text{if } j \text{ replaces } j_0 \text{ in } J, \\ 0 & \text{otherwise.} \end{cases} \) for \( j \in I \setminus J \), whether \( j \) is the best replacement for \( j_0 \).

\[
\min_{i \in I} \sum_{k \in I} u_k + \sum_{k \in I \setminus \{j_0\}} v_k + \sum_{j \in j \setminus \{j_0\}} f_j + \sum_{i \in I \setminus J} f_i \rho_i, \quad \text{(BestRepl}(j_0))
\]

\[
\text{s.t. } u_{j_0} + v_k \geq \lambda_k z_{j_0}, \quad \forall i, k \in I \setminus J,
\]

\[
z_{j_0} \geq d_{j_0 j} - \bar{D}_{j_0 j}(1 - x_{j_0 j}), \quad \forall j \in J \setminus \{j_0\},
\]

\[
d_{j_0 j} \geq \|a_{j_0} - \bar{a}_j\|, \quad \forall j \in I \setminus \{j_0\},
\]

\[
\sum_{j \in I \setminus \{j_0\}} x_{j_0 j} = 1,
\]

\[
u_i + v_k \geq \lambda_k z_i - \bar{D}_i \rho_i, \quad \forall i, k \in I \setminus J,
\]

\[
z_i \geq d_{ij} - \bar{D}_{ij}(2 - x_{ij} - \rho_j), \quad \forall i \in \{j_0\} \cup I \setminus J, j \in I \setminus J,
\]

\[
d_{ij} \geq \|a_i - \bar{a}_j\|, \quad \forall i \in \{j_0\} \cup I \setminus J, j \in J \setminus \{j_0\},
\]

\[
\sum_{j \in I \setminus \{j_0\}} x_{ij} = 1 - \rho_i, \quad \forall i \in I \setminus J,
\]

\[
r_j \geq \|a_j - \bar{a}_j\|, \quad \forall j \in I \setminus \{j_0\},
\]

\[
x_{ii} \geq \rho_i, \quad \forall i \in I \setminus J,
\]

\[
x_{ij} \leq x_{jj}, \quad \forall i \in I \cup \{j_0\} \setminus J, j \in I \setminus \{j_0\}
\]

\[
\sum_{j \in I \setminus J} \rho_j = 1,
\]

\[
\bar{a}_i \in \mathbb{R}^d, \quad \forall i \in I,
\]

\[
d_{ij}, z_i \geq 0, \quad \forall i \in \{j_0\} \cup I \setminus J, j \in J \setminus \{j_0\},
\]

\[
x_{ij} \in \{0, 1\}, \quad \forall i \in \{j_0\} \cup I \setminus J, j \in I \setminus \{j_0\},
\]

\[
\rho_j \in \{0, 1\}, \quad \forall \in I \setminus J
\]
The objective function account for the travel costs (sum of the $u$ and $v$-variables) and the set-up costs, for which there is a constant term (the one that coincides with the previous solution) and the one accounting for the set-up cost of the facility replacing $j_0$.

Constraints (4.1)–(4.4) allow to model the sorting distances and the allocation of $j_0$ (which turns into a non-open facility). On the other hand, constraints (4.5)–(4.8) model the same but for any potential non-open facility, except $j_0$, i.e. those demand points in $I\setminus J$. (4.9) define the neighborhoods. Constraints (4.10) force to open the new facility replacing $j_0$. (4.11) is the usual requirement of allocating demand points only to open facilities. With constraint (4.12) a single replacement for $j_0$ is found among the non-open facilities.

Although (BestRepl($j_0$)) is similar to (OMPN$_{REP}$), the number of binary variables in the problem is $(n - p)n$ instead of $n^2$.

We have empirically checked that, in general, solving (BestRepl($j_0$)) requires more CPU time than solving the $n - p$ problems in the form (ALLOC($J$)). However, for problems in which $n - p \ll n$ (i.e. when $p$ is large), the compact formulation (BestRepl($j_0$)) is expected to consume less CPU time than loading and solving $n - p$ problems in the shape of (ALLOC($J$)).

Observe also that the problem of finding multiple replacements of sets of open facilities $\{j_1, \ldots, j_q\}$ (with $q \leq p$) simultaneously, could also be formulated as a MINLP similar to (BestRepl($j_0$)), but it would require a large number of binary variables, increasing the computational complexity of the problem.

The next ingredient to take into account in our math-heuristic approach, as in any local search heuristic, is to find a good enough initial feasible solution for the problem (in the $x$-variables). We compute the solution of the DOMP using as costs between facilities $i$ and $j$ a convex combination of the lower and upper bounds $\hat{d}_{ij}$ and $\hat{D}_{ij}$ which provide good results in practice. One may also consider solving the DOMP for a linear norm (as $\ell_1$) and polyhedral neighborhoods approximating the non-linear ones, in whose case, the problem turns into a Mixed Integer Linear Programming problem.

Now, we are ready to describe our first math-heuristic procedure, whose pseudocode is shown in Algorithm 1. We give as input an initial set of $p$ open facilities ($\hat{J}$) provided by the initial solution, as well as its overall cost $UB = c(\hat{J})$, which is computed by solving (ALLOC($\hat{J}$)). Then, for each open facility in $\hat{J}$, we find the best replacement, as described above. With such a replacement, if the overall cost is smaller than $UB$, the set of open facilities and the upper bound are updated. The procedure repeats the same scheme until a termination criterion is fulfilled. In our case, two stopping criteria are considered: maximum number of iterations and maximum number of iterations without improvement in the solution.

In order to reduce the computation times required for solving (ALLOC($J$)) or (BestRepl($j_0$)), we consider a randomized version of the algorithm in which instead of finding best replacements for all the open facilities, a random one is selected at that phase of the approach, i.e. $j_0$ is randomly chosen at each iteration of the algorithm.
Algorithm 1: Math-Heuristic 1 for solving OMPN.

Initialization: Let $\hat{J} \subset N$ with $|\hat{J}| = p$ an initial set of open facilities and $UB = c(\hat{J})$.

while $it < it_{max}$ do
  for $j_0 \in \hat{J}$ (or random) do
    Find the best replacement for $j_0$ (by solving (ALLOC($J)$) for $J = \hat{J} \cup \{i\}\{j_0\}$ (for $i = 1, \ldots, n$) or (BestRepl($j_0$)): $c^* = c(J)$.
    if $c^* < UB$ then
      Update $UB = c^*$
      $\hat{J} = J$
    end
  end
  $it = it + 1$
end

4.2 Math-heuristic Algorithm 2

The second heuristic is based on alternating the resolution of the discrete and the continuous problems involved in the DOMPN. This family of approaches has been called mountain climbing or alternate search algorithms in some papers analyzing biconvex problems (see [18]). The main idea of this type of procedures is to separate the variables of the model into two sets, $X$ and $Y$. Then, for some initial values of $Y$, the problem is solved in the $X$ variables. Then, the $X$-variables are fixed to these optimal values, and the problem is solved in the $Y$ variables. The procedure is repeated until there is no improvement in two consecutive solutions. This procedure is known to converge to a local optimum of the problem.

Let us describe this search for the DOMPN. The two sets of variables where the above general procedure is applied are the sets of discrete decisions, $X$ ($x$-variables), and the set of continuos decisions, $Y$ [variables $\bar{a}$ and $d$ in problem (OMPN_BEP)]. Note that the rest of the variables ($u, v, z$) are auxiliary variables used to model the Ordered Median function. First, the DOMP is solved by fixing $\bar{a} = a$ (which implies fixing the distances $d_{ij} = \|a_i - \bar{a}_j\|$ among each pair of customer/facilities). The resolution of this problem provides initial solutions in the $x$-variables ($X$). This set of optimal open facilities are kept and given as input to (ALLOC($J$)). Then, we obtain values for the continuous values ($Y$). With this solution, the DOMP is again solved to update $X$. The process is repeated, keeping the best solution found, until no improvement is obtained.

In order to escape from local optima, two different strategies were applied. Each time DOMP is solved, we add the constraint:

$$\sum_{j \in J} x_{jj} \leq p - 1$$

where $J$ is the set of open facilities obtained in the previous iteration. This constraint assures that the same set of open facilities is not going to be obtained. Also, a multi-start approach is applied, in which, instead of fixing $\bar{a} = a$ in the initial solution, a random point $\bar{a} \in N(a)$ is chosen.
The pseudocode for this approach is shown in Algorithm 2. There, we denote by OMN(\(\bar{a}\)) the DOMPN in which the \(\bar{a}\)-variables are fixed to \(\bar{a}\).

**Algorithm 2:** Math-Heuristic 2 for solving OMPN.

```
Initialization: Random \(\bar{a}^*_j \in \mathcal{N}(a_j)\) for \(j \in I\), \(F_1 = -\infty\), \(F_2 = \infty\), \(F = \infty\), \(J = \emptyset\).

while \(|F_1 - F_2| > \varepsilon\) do
  Solve OMN(\(\bar{a}^*\)): \(F_1 = F_1^*\) and \(J = \{j \in I : x_{jj}^* = 1\}\).
  Solve (ALLOC(\(J\))): \(F_2 = F_2^*\) and \(\bar{a}^* = \bar{a}\).
  if \(F_2 < F\) then
    \(F = F_2\)
  end
end
output : \(F\)
```

Before giving some details about the experiments run on a real dataset to test the heuristic algorithms, we have checked their performance on the most difficult random instances generated in Sect. 3.6, i.e. those with \(n = 20, 30\) and \(p = 5\) (see Table 5). A summary of the average results obtained with the two heuristics is drawn in Fig. 6. There, we show in the left side, the comparisons of the average CPU times consumed by the exact BEP formulation (time\(_{BEP}\)), Math-Heuristic 1 (time\(_{H1}\)) and Math-Heuristic 2 (time\(_{H2}\)). As expected, the times needed by the heuristics are much smaller than those consumed by the best exact approach (OMP\(_{BEP}\)). On the other hand, in the right picture, we report the percent deviation of the solution obtained with the heuristics with respect to the exact solution. The obtained deviations are very small for both heuristics. The Math-heuristic 1 seems to provide better solutions at the price of consuming more CPU time than the Math-Heuristic 2.
Fig. 7 Basic neighborhoods for the 49 US states (radii \( r^0 \))

5 Experiments

In order to test the performance of the math-heuristic approaches, we have run some experiments over the real dataset instance of 2-dimensional coordinates (normalized longitude and latitude) of geographical centers of 49 states of the Unites States (we exclude Alaska, Hawaii and those outside Northamerica). We consider as neighborhoods Euclidean disks with radii based on the areas of each state. For each state (indexed by \( j \)), the area (in km\(^2\)), \( A_j \), was obtained and we construct the radius \( r^0_j = \sqrt{\frac{A_j}{\pi}} \). The coordinates and the disks built applying this strategy are drawn in Fig. 7. Then, three different scenarios were considered:

- **S1**: \( r_j = r^0_j \), for \( j = 1, \ldots, 49 \).
- **S2**: \( r_j = 2 \times r^0_j \), for \( j = 1, \ldots, 49 \).
- **S3**: \( r_j = 3 \times r^0_j \), for \( j = 1, \ldots, 49 \).

This dataset is new and constructed to test the OMPN problem. The interested reader may download the datasets at http://bit.ly/datasetUS.

We implemented, in Gurobi under the C API, the two math-heuristic approaches and we compare the running times and the best values obtained with these procedures and those obtained with the exact (OMPN\(_{BEP}\)) formulation (with a time limit of 1 h). We solved the \( p \)-Median (M), \( p \)-Center (C), \( p \)-25-center (K) and \( p \)-Centdian (D) with \( p \in \{2, 5, 10\} \). The results are reported in Table 6. In such a table, the first column indicates the scenario (1, 2 or 3), the second column indicates the number of facilities to be open, \( p \), and the third column (Pr.) shows the problem type. The values of the solutions obtained by using the different approaches as well as their CPU running times (in seconds) are reported:

- Initial solution obtained by solving the nominal DOMP and solving (ALLOC(\( J \))) for the obtained open facilities, and the consumed CPU time: \( H0 \) and \( t0 \).
- Best solution obtained by the math-heuristic approach 1 and the consumed CPU time: \( H1 \) and \( t1 \).
Table 6  Results of Math-Heuristic Approaches and (OMPΝ\textsubscript{BF}F) in the US dataset

| Sc. | p   | Pr. | H0     | H1     | H2     | BEP     | t0  | t1  | t2  | tBEP | %VarFixed (%) | G1 (%) | G2 (%) | G0 (%) |
|-----|-----|-----|--------|--------|--------|---------|-----|-----|-----|------|---------------|--------|--------|--------|
| SC1 | 2   | M   | 395.3482 | 394.8909 | 395.3482 | 394.891 | 2.5     | 31.52 | 13.28 | 32.44 | 89.15         | 0      | 0.12   | 0.12   |
|     | C   | 19.105 | 18.0278 | 18.0278 | 18.0278 | 0.08    | 0.28 | 4.2 | 8.18 | 90.41 | 0     | 0      | 5.64   |
|     | K   | 272.8855 | 270.8245 | 270.7348 | 270.7348 | 0.84    | 12.88 | 33.24 | 22.92 | 89.84 | 0.03 | 0      | 0.79   |
|     | D   | 207.2299 | 207.2299 | 207.2299 | 207.2298 | 2.99    | 18.14 | 12.41 | 22.12 | 89.88 | 0     | 0      | 0      |
| 5   | M   | 222.3974 | 221.607 | 221.5985 | 222.6594 | 2.68    | 56.03 | 160.26 | >3600 | 81.54 | -0.47 | -0.48 | -0.12 |
|     | C   | 18.619 | 18.2246 | 16.2491 | 16.2491 | 0.17    | 0.35 | 9.57 | 37.59 | 84.90 | 10.84 | 0      | 12.73  |
|     | K   | 167.6711 | 163.564 | 160.1906 | 160.1138 | 3.98    | 24.07 | 239.04 | >3600 | 83.05 | 2.11  | 0.05   | 4.51   |
|     | D   | 120.9466 | 120.0572 | 119.8601 | 119.7391 | 2.66    | 67.81 | 116.23 | >3600 | 83.02 | 0.27  | 0.10   | 1.00   |
| 10  | M   | 146.2992 | 144.6164 | 144.2134 | 141.8635 | 8.4     | 145.96 | 245.51 | >3600 | 85.33 | 1.90  | 1.63   | 3.03   |
|     | C   | 27.0357 | 22.9692 | 19.9944 | 19.9944 | 0.19    | 3.06 | 15.9 | 55.57 | 84.83 | 12.95 | 0      | 26.04  |
|     | K   | 118.7461 | 118.0626 | 117.7095 | 125.0503 | 6.34    | 70.23 | 1760.77 | >3600 | 85.80 | -5.92 | -6.24 | -5.31  |
|     | D   | 86.6926 | 85.2886 | 85.7954 | 85.5946 | 10.84   | 94.08 | 357.67 | >3600 | 86.32 | -0.36 | 0.23   | 1.27   |
| SC2 | 2   | M   | 399.6586 | 395.1866 | 398.2659 | 395.1789 | 2.58    | 96.14 | 27.78 | 462.06 | 77.53 | 0     | 0.78   | 1.12   |
|     | C   | 23.3894 | 21.8305 | 22.1818 | 21.7935 | 0.11    | 0.45 | 6.72 | 10.48 | 85.33 | 0.17  | 1.75   | 6.82   |
|     | K   | 275.9869 | 274.8279 | 274.6485 | 274.2601 | 1.4     | 25.67 | 32.68 | 365.95 | 78.30 | 0.21  | 0.14   | 0.63   |
|     | D   | 211.7095 | 209.7885 | 211.8238 | 209.7885 | 2.59    | 138.32 | 56.4 | 262.4 | 75.94 | 0     | 0.96   | 0.91   |
| 5   | M   | 228.6722 | 221.514 | 226.7708 | 223.3972 | 637.69  | >3600 | 689.41 | >3600 | 67.96 | -0.85 | 1.49   | 2.31   |
|     | C   | 29.0877 | 23.0926 | 22.0907 | 21.5931 | 0.55    | 1.68 | 15.67 | 46.04 | 71.11 | 6.49  | 2.25   | 25.77  |
|     | K   | 171.2099 | 168.7613 | 167.6214 | 166.7242 | 21.55   | 163.7 | 1025.78 | >3600 | 64.07 | 1.21  | 0.54   | 2.62   |
|     | D   | 129.7708 | 125.1129 | 125.4487 | 127.2981 | 266.3   | 600.94 | 641.76 | >3600 | 69.92 | -1.75 | -1.47  | 1.91   |
| 10  | M   | 153.013 | 153.013 | 155.285 | 151.5515 | 3600.1  | >3600 | 3208.72 | >3600 | 76.33 | 0.96  | 2.40   | 0.96   |
|     | C   | 44.5115 | 35.0981 | 29.8835 | 29.842 | 0.66    | 16.66 | 46.47 | 53.42 | 74.23 | 14.98 | 0.14   | 32.96  |
|     | K   | 132.6658 | 132.6658 | 131.9971 | 142.0924 | >3600   | >3600 | >3600 | >3600 | >3600 | -7.11 | -7.65  | -7.11  |
|     | D   | 100.474 | 100.474 | 97.6098 | 108.2469 | >3600   | >3600 | 3215.14 | >3600 | 76.33 | -7.74 | -10.90 | -7.74  |
| Sc. | p  | Pr. | H0   | H1   | H2   | BEP  | t0   | t1   | t2   | tBEP | %VarFixed (%) | G1 (%) | G2 (%) | G0 (%) |
|-----|----|-----|------|------|------|------|------|------|------|------|--------------|--------|--------|--------|
| SC3 | 2  | M   | 404.466 | 394.9373 | 398.7388 | 395.7311 | 2.94 | 108.55 | 49.53 | > 3600 | 62.96 | 0.20 | 0.75 | 2.16 |
|     |    | C   | 28.0977 | 24.1051 | 24.1924 | 24.1051 | 0.11 | 0.66  | 6.03  | 10.05  | 74.69 | 0  | 0.36 | 14.21 |
|     |    | K   | 280.5643 | 278.0229 | 278.9015 | 278.0228 | 1.42 | 32.54 | 41.81 | 1455.24 | 52.65 | 0  | 0.32 | 0.91 |
|     |    | D   | 216.5167 | 210.8067 | 214.7534 | 210.6248 | 2.97 | 74.29  | 83.65 | 1488.5 | 72.21 | 0.09 | 1.92 | 2.72 |
| 5   | M  | 232.5595 | 223.4184 | 235.2989 | 240.8416 | 915.52 | > 3600 | 702.09 | > 3600 | 60.38 | 0.36 | 2.36 | 3.56 |
|     | C  | 34.7835 | 28.4267 | 27.4501 | 26.5732 | 0.23 | 3.57  | 13.95 | 36.93  | 64.23 | 6.52 | 3.19 | 23.60 |
|     | K  | 183.8955 | 171.2615 | 178.0249 | 173.4423 | 128.12 | 191.26 | 1310.85 | > 3600 | 54.76 | −1.27 | 2.57 | 5.68 |
|     | D  | 134.1824 | 133.0089 | 132.521 | 138.3565 | 1069.6 | > 3600 | 1206.69 | > 3600 | 57.96 | −4.02 | −4.40 | −3.11 |
| 10  | M  | 167.5751 | 167.5751 | 171.1057 | 177.7685 | > 3600 | > 3600 | 3208.25 | > 3600 | 70.41 | −6.08 | −3.89 | −6.08 |
|     | C  | 50.7627 | 47.0082 | 39.3866 | 38.5431 | 1.1 | 4.02  | 25.13 | 32.2  | 70  | 18.01 | 2.14 | 24.07 |
|     | K  | 144.9262 | 144.9262 | 146.6831 | 161.303 | > 3600 | > 3600 | 2336.69 | > 3600 | 70.41 | −11.30 | −9.97 | −11.30 |
|     | D  | 110.2827 | 110.2827 | 110.5119 | 116.5976 | > 3600 | > 3600 | 3214.45 | > 3600 | 70.41 | −5.73 | −5.51 | −5.73 |
– Best solution obtained by the math-heuristic approach 2 and the consumed CPU time: H2 and t2.
– Best solution obtained by exact formulation (OMPN_{BEP}) within the time limit and the consumed CPU time: BEP and t_{BEP}.

We also report in the 12th column (%VarFixed) the average number of binary variables fixed in the first heuristic, and the percentage deviations of the obtained solutions with respect to the best solution found with the exact formulation within the time limit: G1, G2 and G0 for the first heuristic, the second heuristic and the initial solution, respectively.

One can observe from the results that, the CPU times needed to run the math-heuristic approaches are much smaller than those needed to solve the OMPN problem with the MINLP formulation. In those cases in which all the approaches were able to solve the problem before the time limit of 1 h, the highest deviation with respect to the optimal solutions was 15% for the first heuristic and 3.2% for the second one. In those cases in which the exact approach was not able to certify optimality in 1 h, the math-heuristic approaches found a better solution for the problem. In the first heuristic, we apply the fixing variables strategy each time (ALLOC(J)) is solved. The average percentage of binary variables that are fixed with this strategy, is at least 84% for scenario SC1, 75% for SC2 and 52% for SC3. Observe also that the initial solution based on fixing the open facilities to the solution of the DOMP and then compute the location on the neighborhoods and the allocation of the customers according to these positions, is in many instances, far from being a close-to-optimal choice, with percentage deviations up to 33%, which also indicates that OMPN and DOMP can be very different problems even for small-size neighborhoods.

Note that the two math-heuristic approaches are still very time consuming. One may not forget that both proposed approaches are based on solving mixed integer non linear programming problems which are known to be NP-hard. Recall also that the OMPN has as particular case the DOMP (whose resolution is highly exploited in the heuristics). As far as we know, the largest instances for which the DOMP has been exactly solved are of size $n = 100$, using a recent sophisticated Branch-Price-and-Cut procedure [12]. The advantage of the two math-heuristic approaches proposed in this paper is that they provide good quality solutions at the first iterations, which are competitive with the exact solutions (in terms of gap).

In Fig. 8 we show the best solutions obtained for the test problem for $p = 5$ under the center objective function for scenario SC1. The initial solution for this problem is drawn in Fig. 9 (the solutions for $p = 2, 5, 10$ can be found in bit.ly/resultsDOMPN). The reader can observe that small modification of the coordinates of the potential facilities (through neighborhoods) may produce different location-allocation solutions.

### 6 Conclusions

A unified version of the classical $p$-median problem is introduced in this paper which includes as particular cases the discrete and the continuous $p$-median and $p$-center problems. The problem considers that each facility can be located not only in the exact
given position but in a neighborhood around it. Also, ordered median objective functions are modeled for the problem. Several mathematical programming formulations are proposed based on formulations for the discrete ordered median problem obtained from different sources. Two location-allocation heuristic approaches for solving the problem are presented. Although the optimization problems needed to solve are still NP-hard, the reduced dimensionality of them allows us to provide good quality solutions in more reasonable times.

Several extensions are possible within this new framework. The first is the development of decomposition approaches for solving the OMPN. Lagrangean decomposition (relaxing the ordering constraints) combined with Benders decomposition (to separate the discrete and the continuous decisions) may produce exact solutions in better CPU times. Regarding the heuristic approaches, the incorporation of any of the approximated approaches described in [13,40,46] when the DOMP has to be solved in our math-heuristic procedures could considerably improve the CPU times needed to solve the OMPN.

 Springer
On the other hand, although we analyze the ordered \( p \)-median problem, the results in this paper can be extended to other discrete location problems. For instance, capacitated [38] or multiperiod [1,34] location problems can be embedded into the neighborhoods framework. Other interesting related problems which are left for further research are location-routing problems with neighborhoods. Those problems involve not only the discrete facility location problem with neighborhoods but also the TSP with neighborhoods, then, the combination of the methods proposed in this paper with those provided in [17] may be applicable to the problem. Also, the case in which the neighborhood of each facility is the union of convex sets would be an interesting next step within this framework. In particular, it would model the case in which two facilities may belong to the same neighborhood, which may imply reducing the cost. The extended MINLP formulations for such a problem will become disjunctive MINLP for which some techniques are available in the literature. Another approach that would extend the version introduced in this paper is the one in which \( k_j \) facilities are allowed to be located at the \( j \)-th neighborhood to allocate the demand points. In such a case, a nested multifacility \( p \)-median problem is considered for which more sophisticated strategies should be developed to solve even small-size instances.

Acknowledgements The author was partially supported by Project MTM2016-74983-C2-1-R (MINECO, Spain), the research group SEJ-534 (Junta de Andalucía) and the research project PP2016-PIP06 (Universidad de Granada). The author would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

A Appendix: Proof of Proposition 3.1

Denote by \( F_{3I} \), \( F_{2I} \), \( F_{OT} \) and \( F_{BEP} \) the feasible regions of \( (OMP_{3I}) \), \( (OMP_{2I}) \), \( (OMP_{OT}) \) and \( (OMP_{BEP}) \) obtained when relaxing the integrality conditions of the models.

1. Consider the mapping \( \pi : \mathbb{R}_+^n \times \mathbb{R}_+^{n \times n} \times [0, 1]^{n \times n} \times \mathcal{X}_R \times \mathcal{D} \to \mathbb{R}_+^3 \times [0, 1]^3 \times [0, 1]^n \times \mathcal{D} \) defined as:

\[
\pi(\xi, \theta, s, x, (d, \bar{a})) = ((\xi_k s_{ik} x_{ij})_{i,j,k=1}^n, (s_{ik} x_{ij})_{i,j,k=1}^n, (x_{jj})_{j=1}^n, (d, \bar{a}))
\]

First, let us check that \( \pi(F_{2I}) \subseteq F_{3I} \), which would prove the first inequality. Let \((\theta, \xi, s, x, (d, \bar{a})) \in F_{2I}\), and define \((\bar{\theta}, \bar{x}, (d, \bar{a})) = \pi(\theta, \xi, s, x, (d, \bar{a}))\), i.e.:

\[
\bar{\theta}_{ij}^k = \xi_k s_{ik} x_{ij}, \quad \bar{w}_{ij}^k = s_{ik} x_{ij}, \quad \bar{x}_{jj} = x_{jj}, \quad \forall i, j, k = 1, \ldots, n.
\]

By construction, the constraints (3.1)–(3.4) are verified:

\[
- \sum_{j,k=1}^n \bar{w}_{ij}^k = \sum_{j,k=1}^n s_{ik} x_{ij} = \sum_{j=1}^n x_{ij} \sum_{k=1}^n s_{ik} \overset{(3.10)}{=} \sum_{j=1}^n x_{ij} = 1.
\]

\[
- \sum_{i,j=1}^n \bar{w}_{ij}^k = \sum_{i,j=1}^n s_{ik} x_{ij} = \sum_{i=1}^n s_{ik} \sum_{j=1}^n x_{ij} \overset{x \in \mathcal{X}_R}{=} \sum_{i=1}^n s_{ik} \overset{(3.9)}{=} 1.
\]
\[- \sum_{k=1}^{n} \bar{w}_{ij}^k = \sum_{k=1}^{n} s_{ik} x_{ij} = x_{ij} \sum_{k=1}^{n} s_{ik} = x_{ij} \in \mathcal{X}_R \]
\[- \sum_{j=1}^{n} x_{jj} = \sum_{j=1}^{n} x_{jj} = x_{jj} \in \mathcal{X}_R \]
\[- \bar{\theta}_{ij}^k = \xi_k s_{ik} x_{ij} = (d_{ij} - \hat{D}_{ij} (2 - s_{ik} - x_{ij}))(s_{ik} x_{ij} = d_{ij} - \hat{D}_{ij} (1 - s_{ik} x_{ij}) + \hat{D}_{ij} s_{ik} x_{ij} (s_{ik} + x_{ij})) \geq d_{ij} - \hat{D}_{ij} (1 - \bar{w}_{ij}^k).
\]
\[- \sum_{i,j=1}^{n} \bar{\theta}_{ij}^k = \sum_{i,j=1}^{n} \xi_k s_{ik} x_{ij} = \xi_k \sum_{i=1}^{n} s_{ik} \sum_{j=1}^{n} x_{ij} = \xi_k \sum_{i=1}^{n} s_{ik} = x_{jj} \in \mathcal{X}_R \]
\[- \sum_{j=1}^{n} f_j x_{jj} = \sum_{j=1}^{n} f_j x_{jj} \leq \sum_{k=1}^{n} \lambda_k \xi_k \sum_{i=1}^{n} s_{ik} + \sum_{j=1}^{n} f_j x_{jj} \]
\[- x_{ij} = \sum_{k=1}^{n} \lambda_k \xi_k s_{ik} x_{ij} + \sum_{j=1}^{n} f_j x_{jj} = \sum_{k=1}^{n} \lambda_k \xi_k s_{ik} + \sum_{j=1}^{n} f_j x_{jj} \]

Then, \(\pi(\theta, \xi, s, x, (d, \bar{a})) \in F_3^I\), so \(\pi(F_2 I) \subset F_3 I\), i.e. any solution of the convex relaxation of \((\text{OMPN}_2 I)\) induces a solution of the convex relaxation of \((\text{OMPN}_3 I)\). Furthermore, the objective values for \((\theta, \xi, s, x, (d, \bar{a})) \) in \((\text{OMPN}_2 I)\) and \(\pi(\theta, \xi, s, x, (d, \bar{a})) \) in \((\text{OMPN}_3 I)\) coincides:
\[
\sum_{i,j=1}^{n} \lambda_k \bar{\theta}_{ij}^k + \sum_{j=1}^{n} f_j x_{jj} = \sum_{i,j,k=1}^{n} \lambda_k \xi_k s_{ik} x_{ij} + \sum_{j=1}^{n} f_j x_{jj} = \sum_{k=1}^{n} \lambda_k \xi_k s_{ik} + \sum_{j=1}^{n} f_j x_{jj} \]

Thus, \(z_R^I \geq z_R^3 I\).

2. Let us check now that \(z_R^3 I \leq z_{BE P}^R\).

Let \((u, v, D, x, (d, \bar{a})) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathcal{X}_R \times \mathcal{D}\) be the optimal solution of the continuous relaxation of \((\text{OMPN}_{BE P})\). Let \(p_{ik}\) be the optimal dual variables associated to constraint \(3.21\). By optimality conditions they must verify:
\[
\sum_{i=1}^{n} p_{ik} = 1, \forall k = 1, \ldots, n,
\]
\[
\sum_{k=1}^{n} p_{ik} = 1, \forall i = 1, \ldots, n.
\]
Let us construct the following vector in \( \mathbb{R}^n_+ \times \mathbb{R}^{n \times n}_+ \times [0, 1]^{n \times n} \times X_R \times D \):

\[
(\tilde{\theta}, \tilde{w}, x, (d, \tilde{a})) := (\{d_{ij} p_{ik} x_{ij}\}, \{p_{ik} x_{ij}\}, x, (d, \tilde{x})).
\]

By the construction of the \( p_{ik} \)-values, it is straightforward to prove that the projected values verify the constraints of (OMP\(\text{N}_{2I}\)) and that the objective values of both solutions coincide. Thus, \( z^R_{\text{BEP}} \geq z^R_{\text{3I}} \).

Finally, we prove that \( z^R_{\text{BEP}} = z^R_{\text{OT}} \).

Let us consider \( x \in X_R, (d, a) \in D \). Observe that the difference between (OMP\(\text{N}_{\text{OT}}\)) and (OMP\(\text{N}_{\text{BEP}}\)) is the way of evaluating the ordered median cost, \( \text{om}(x, d, a) \). In the first case:

\[
\text{om}(x, d, a) = \min \sum_{k=1}^{n-1} \Delta_k (k t_k + \sum_{i=1}^{n} z_{ik})
\]

\[
s.t. \ z_{ik} \geq D_i - t_k, \forall \ i, k = 1, \ldots, n,
\]

\[
D_i \geq d_{ij} - \hat{D}_{ij} (1 - x_{ij}), \forall \ i, j = 1, \ldots, n, \ i \neq j,
\]

\[
z_{ik}, D_i \geq 0, \forall \ i, k = 1, \ldots, n,
\]

\[
t_k \in \mathbb{R}, \forall k = 1, \ldots, n.
\]

while in the BEP formulation is via:

\[
\text{om}(x, d, a) = \min \sum_{k=1}^{n} u_k + \sum_{i=1}^{n} v_i
\]

\[
s.t. \ u_i + v_k \geq \lambda_k D_i, \forall \ i, k = 1, \ldots, n,
\]

\[
D_i \geq d_{ij} - \hat{D}_{ij} (1 - x_{ij}), \forall \ i, j = 1, \ldots, n, \ i \neq j,
\]

but the values coincides with the ordered median function of the (relaxed) travel costs. Thus, \( z^R_{\text{BEP}} = z^R_{\text{OT}} \).

References

1. Albareda-Sambola, M., Fernández, E., Hinojosa, Y., Puerto, J.: The multi-period incremental service facility location problem. Comput. Oper. Res. 36(5), 1356–1375 (2009)
2. Albareda-Sambola, M., Fernández, E., Saldanha-da Gama, F.: The facility location problem with bernoulli demands. Omega 39(3), 335–345 (2011)
3. Blanco, V., El Haj Ben Ali, S., Puerto, J.: Minimizing ordered weighted averaging of rational functions with applications to continuous location. Omega 39(3), 335–345 (2011)
4. Blanco, V., Fernández, E., Puerto, J.: Mathematical programming formulations and solution approaches for minimum spanning trees with neighborhoods. Eur. J. Oper. Res. 262(3), 863–878 (2017)
5. Blanco, V., Puerto, J., El Haj Ben Ali, S.: Revisiting several problems and algorithms in continuous location with \( \ell_1 \)-norms. Comput. Optim. Appl. 58(3), 563–595 (2014)
6. Blanco, V., Puerto, J., El-Haj Ben-Ali, S.: Continuous multifacility ordered median location problems. Eur. J. Oper. Res. 250(1), 56–64 (2016)
7. Boland, N., Domínguez-Marín, P., Nickel, S., Puerto, J.: Exact procedures for solving the discrete ordered median problem. Comput. Oper. Res. 33(11), 3270–3300 (2006)
8. Cooper, L.: Bounds on the weber problem solution under conditions of uncertainty. J. Reg. Sci. 18(1), 87–92 (1978)
9. Correia, I., Nickel, S., Saldanha-da Gama, F.: A stochastic multi-period capacitated multiple allocation hub location problem: formulation and inequalities. Omega 74, 122–134 (2018)
10. Correia, I., Saldanha da Gama, F.: Facility location under uncertainty, pp. 177–203. Springer, Berlin (2015)
11. de Berg, M., Gudmundsson, J., Katz, M.J., Levcan, A.F.: TSP with neighborhoods of varying size. J. Algorithms 57(1), 22–36 (2005)
12. Deleplanque, S., Labbé, M., Ponce, D., Puerto, J.: An extended version of a branch-price-and-cut procedure for the discrete ordered median problem (2018). arXiv preprint arXiv:1802.03191
13. Domínguez-Marín, P., Nickel, S., Hansen, P., Mladenović, N.: Heuristic procedures for solving the discrete ordered median problem. Ann. Oper. Res. 136(1), 145–173 (2005)
14. Dorrigiv, R., Fraser, R., He, M., Kamali, S., Kawamura, A., López-Ortiz, A., Seco, D.: On minimum-and maximum-weight minimum spanning trees with neighborhoods. Theory Comput. Syst. 56(1), 220–250 (2015)
15. Drezner, Z., Hamacher, H.W. (eds.): Facility Location: Applications and Theory. Springer, Berlin (2002)
16. Dumitrescu, A., Mitchell, J.S.B.: Approximation algorithms for TSP with neighborhoods in the plane. J. Algorithms 48(1), 135–159 (2003)
17. Gentili, I., Margot, F., Shimada, K.: The travelling salesman problem with neighbourhoods: Mip solution. Optim. Methods Softw. 28(2), 364–378 (2013)
18. Garey, M.R., Johnson, D.S.: The traveling salesman problem and its variations. In: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
19. Hakimi, S.: Optimum location of switching centers and the absolute centers and medians of a graph. Oper. Res. 12, 450–459 (1964)
20. Juel, H.: Bounds in the generalized weber problem under locational uncertainty. Oper. Res. 29(6), 1219–1227 (1981)
21. Kalcsics, J., Nickel, S., Puerto, J.: Multifacility ordered median problems on networks: a further analysis. Networks 41(1), 1–12 (2003)
22. Kalcsics, J., Nickel, S., Puerto, J., Rodríguez-Chía, A.M.: The ordered capacitated facility location problem. TOP 18(1), 203–222 (2010)
23. Kariv, O., Hakimi, S.: An algorithmic approach to network location problems. ii: The p-median problem. SIAM J. Appl. Math. 37(3), 539–560 (1979)
24. Kuehn, A.A., Hamburguer, M.J.: A heuristic program for locating warehouses. Manag. Sci. 9, 643–666 (1963)
25. Labbé, M., Ponce, D., Puerto, J.: A comparative study of formulations and solution methods for the discrete ordered p-median problem. Comput. Oper. Res. 78, 230–242 (2017)
26. Ljubić, I., Gollwitzer, S.: Layered graph approaches to the hop constrained connected facility location problem. INFORMS J. Comput. 25(2), 256–270 (2012)
27. Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. Linear Algebra Appl. 284(1), 193–228 (1998)
28. Marín, A., Nickel, S., Puerto, J., Velten, S.: A flexible model and efficient solution strategies for discrete location problems. Discrete Appl. Math. 157(5), 1128–1145 (2009)
29. Marín, A., Nickel, S., Puerto, J., Velten, S.: An extended covering model for flexible discrete and equity location problems. Math. Methods Oper. Res. 71(1), 125–163 (2010)
30. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part i–convex underestimating problems. Math. Program. 10(1), 147–175 (1976)
31. Melo, M.T., Nickel, S., Saldanha da Gama, F.: Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. Comput. Oper. Res. 33(1), 181–208 (2006)
32. Nickel, S., Puerto, J.: Location Theory: A Unified Approach. Springer, Berlin (2005)
33. Nickel, S., Saldanha da Gama, F.: Multi-period facility location, pp. 289–310. Springer, Cham (2015)
35. Ogryczak, W., Ołender, P.: On milp models for the OWA optimization. J. Telecommun. Inf. Technol. 2, 5–12 (2012)
36. Ogryczak, W., Tamir, A.: Minimizing the sum of the $k$ largest functions in linear time. Inf. Process. Lett. 85(3), 117–122 (2003)
37. Ponce, D.: The discrete ordered median problem revisited: new formulations, properties and algorithms, Ph.D. thesis, Universidad de Sevilla (2016)
38. Puerto, J.: A new formulation of the capacitated discrete ordered median problems with $\{0, 1\}$-assignment, pp. 165–170. Springer, Berlin (2008)
39. Puerto, J., Fernandez, F.R.: Geometrical properties of the symmetrical single facility location problem. J. Nonlinear Convex Anal. 1(3), 321–342 (2000)
40. Puerto, J., Pérez-Brito, D., García-González, C.G.: A modified variable neighborhood search for the discrete ordered median problem. Eur. J. Oper. Res. 234(1), 61–76 (2014)
41. Puerto, J., Ramos, A.B., Rodríguez-Chía, A.M.: Single-allocation ordered median hub location problems. Comput. Oper. Res. 38(2), 559–570 (2011)
42. Puerto, J., Rodríguez-Chía, A.M., Tamir, A.: Revisiting $k$-sum optimization. Math. Program. 165, 1–26 (2016)
43. Puerto, J., Rodríguez-Chía, A.M.: Ordered Median Location Problems, pp. 249–288. Springer, Cham (2015)
44. Puerto, J., Tamir, A.: Locating tree-shaped facilities using the ordered median objective. Math. Program. 102(2), 313–338 (2005)
45. Rodríguez-Chía, A.M., Nickel, S., Puerto, J., Fernández, F.R.: A flexible approach to location problems. Math. Methods Oper. Res. 51(1), 69–89 (2000)
46. Stanimirović, Z., Kratica, J., Dugošija, D.: Genetic algorithms for solving the discrete ordered median problem. Eur. J. Oper. Res. 182(3), 983–1001 (2007)
47. Tang, H., Cheng, T.C.E., Ng, C.T.: Finite dominating sets for the multi-facility ordered median problem in networks and algorithmic applications. Comput. Ind. Eng. 57(3), 707–712 (2016)
48. Ward, J.E., Wendell, R.E.: Using block norms for location modeling. Oper. Res. 33(5), 1074–1090 (1985)
49. Yan, S., Lin, J.-R., Chen, Y.-C., Xie, F.-R.: Rental bike location and allocation under stochastic demands. Comput. Ind. Eng. 107, 1–11 (2017)
50. Yang, Y., Lin, M., Xu, J., Xie, Y.: Minimum spanning tree with neighborhoods, pp. 306–316. Springer, Berlin (2007)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.