A fresh view on the Discrete Ordered Median Problem based on partial monotonicity

Alfredo Marín
Department of Statistics and Operations Research, Faculty of Mathematics,
Universidad de Murcia, Spain
amarin@um.es

Diego Ponce
Instituto de Matemáticas de la Universidad de Sevilla (IMUS), Spain
Centre interuniversitaire de recherche sur les réseaux d'enterprise, la logistique et le transport (CIRRELT), Montreal, Canada
Department of Mechanical, Industrial and Aerospace Engineering,
Gina Cody School of Engineering and Computer Science, Concordia University, Montreal, Canada
dponce@us.es

Justo Puerto *
Instituto de Matemáticas de la Universidad de Sevilla (IMUS), Spain
puerto@us.es

Abstract

This paper presents new results for the Discrete Ordered Median Problem (DOMP). It exploits properties of k-sum optimization to derive specific formulations for the monotone DOMP (MDOMP), that arises when the λ weights are non-decreasing monotone, and new formulations for the general non-monotone DOMP. The main idea in our approach is to express ordered weighted averages as telescopic sums whose terms are k-sums, with positive and negative coefficients. Formulations of k-sums with positive coefficients derive from the linear programming representations obtained by [Ogryczack and Tamir, 2003] and [Blanco et al., 2014]. Valid formulations for k-sums with negative coefficients are more elaborated and we present 4 different approaches, all of them based on mixed integer programming formulations. An extensive computational experience based on a collection of well-known instances shows the usefulness of the new formulations to solve difficult problems such as trimmed and anti-trimmed mean.

Key words: Location – Combinatorial optimization – Logistics – Ordered Median Problem

*Corresponding author

1 Introduction

In a remarkable paper by [Ogryczack and Tamir, 2003] these authors introduce a novel linear time algorithm to compute the sum of the k-largest entries ($k \leq n$) of an arbitrary vector of n components. This approach gives rise to a linear programming representation of such an evaluation. More recently, a different linear programming representation of the sum of the k largest components of a vector has also been obtained [Blanco et al., 2014]. The earlier idea in [Ogryczack and Tamir, 2003] was later exploited by [Kalcsics et al., 2002] to develop an efficient representation for the k-centrum location problem and more generally extended to deal with some classes of Discrete Ordered Median Problems (DOMP) (the reader is referred to [Nickel and Puerto, 2005, Puerto et al., 2009, Kalcsics et al., 2010] for further details on this problem). More recently, the same approach has been used in [Puerto et al., 2017] to provide new algorithms to the general k-sum optimization problem. The interested reader is referred to [Aouad and Segev, 2019], [Blanco, 2019], [Delaplanque et al., 2020], [Labbé et al., 2017], [Olender and Ogryczak, 2019] or [Puerto, 2019] for the most recent results on DOMP appeared in the literature.

k-sum optimization problems are well-understood and solved in a number of important cases (see [Puerto et al., 2017]). In particular, it has been observed that DOMP, under the hypothesis that the λ modeling weights are monotone, can be easily reformulated using as building blocks k-sum terms.

The main goal of this paper is to show how to exploit the powerfulness of k-sum optimization within the framework of the monotone and general non-monotone DOMP. Our contribution is to present a range of formulations for the monotone and general version of DOMP using different forms of representing k-sums. In this journey, we resort to express ordered weighted averages as telescopic sums whose terms are k-sums, with positive and negative coefficients. Next, to get valid mathematical programming formulations we distinguish between reformulations for k-sums with positive and negative coefficients in the objective function. Formulations of k-sums with positive coefficients derive from the linear programming representations obtained by [Ogryczack and Tamir, 2003] and [Blanco et al., 2014]. Valid formulations for k-sums with negative coefficients are more cumbersome and we present 4 different approaches, all of them mixed integer programming formulations. The first two are based on radius formulations in the spirit of [Elloumi et al., 2004, Marín et al., 2009, Marín et al., 2010, García et al., 2011, Puerto et al., 2011]. The third one uses a folk approach including big-M constraints to represent the necessary sorting, and the fourth one is a new three-index formulation with a highly competitive performance. We compare these formulations among them and with another one presented in [Nickel and Velten, 2017] showing their differences and the superiority of our new three-index formulation. This formulation is able to solve instances with more than 150 demand points in few seconds.

The paper is organized in five sections. The first one is the introduction. There, we motivate our research and describe the contributions. In addition, there is a subsection devoted to state the considered problems and the relationship between k-sums functions and the ordered median function. Section 3 introduces the monotone version of DOMP and presents two valid mathematical programming formulations to handle it. There it is also proved that both are equivalent in terms of the linear programming relaxation, although, as we will see, their performance is not equal when applied to different combinatorial objects. Section 4 analyzes the general DOMP, i.e., with no hypothesis of monotonicity in its λ weights. We present four different formulations for the problem and analyze some theoretical relationships among them. In Section 5, we give alternative valid formulations for the general DOMP using a different rationale based on the results in [Blanco et al., 2014] rather than in those by [Ogryczack and Tamir, 2003] used in Section 4. Computational results are reported in Section 6. The paper ends with some conclusions and remarks for future research.

2 DOMP basics

2.1 The k-sum and the Ordered Median functions

Let $N = \{1, \ldots, n\}$ be a set of indices. Assume that we are given a real n-vector $d = (d_1, \ldots, d_n)$ and let us denote by $d_{()} = (d_{(1)}, \ldots, d_{(n)})$ the vector with the components of d sorted in non-decreasing sequence, i.e., $d_{(1)} \leq \ldots \leq d_{(n)}$. It is easy to check that the following two dual linear programs return as its optimal value

$$S_k(d) = \sum_{\ell=k}^{n} d_{(\ell)},$$

the sum of the $n-k+1$ largest values (i.e., from $d_{(k)}$ to $d_{(n)}$) out of the n components of the given vector d (see e.g. [Ogryczack and Tamir, 2003] and [Kalcsics et al., 2002]):

$$\left.\begin{array}{lll} \max & \sum_{i \in N} d_i x_i & \\ \text{s.t.} & \sum_{i \in N} x_i = n-k+1 & \\ & x_i \leq 1, & \forall i \in N \\ & x_i \geq 0, & \forall i \in N \end{array}\right\} = \left\{\begin{array}{llllll} \text{(OT)} & \min & (n-k+1)t & +\sum_{i \in N} z_i & & \\ & \text{s.t.} & t & +z_i & \geq d_i & \forall i \in N \\ & & & z_i & \geq 0 & \forall i \in N. \end{array}\right. \tag{1}$$

In the following we shall denote the formulation in the right hand side as OT. It is clear that, under the assumption of non-negative d-values, the first constraint in the maximization problem can be replaced by an inequality and then the associated dual variable t can be further strengthened to be non-negative.

A second way to recover the value $S_k(d)$ is based on two results that exploit the non-negativity and monotonicity of the λ-vectors used in ordered median functions (omf), namely $\sum_{k \in N} \lambda_k d_{(k)}$ (see [Nickel and Puerto, 2005] for further details). We reproduce them in Lemma 1 for the sake of completeness:

Lemma 1. *Let $\mathcal{P}(n)$ be the set of permutations of N.*

1. *Let $d \in \mathbb{R}^n$ and $\lambda \geq \mathbf{0}$. Then $\sum_{k \in N} \lambda_k d_{(k)}$ is a monotonically non-decreasing function of d.*
2. *If $0 \leq \lambda_1 \leq \cdots \leq \lambda_n$, then $\sum_{k \in N} \lambda_k d_{(k)} = \max_{\sigma \in \mathcal{P}(n)} \sum_{i \in N} \lambda_i d_{\sigma(i)}$.*

The proof can be found, for instance, in [Nickel and Puerto, 2005]. The reader may observe that using Lemma 1 one can write down the evaluation of the function $\sum_{k \in N} \lambda_k d_{(k)}$ by means of the optimal value of the following Integer Linear Problem:

$$\sum_{k \in N} \lambda_k d_{(k)} = \left\{\begin{array}{llll} \max & \sum_{k \in N} \sum_{i \in N} (\lambda_k d_i) p_{ik} & & \\ \text{s.t.} & \sum_{i \in N} p_{ik} & = 1 & \forall k \in N \\ & \sum_{k \in N} p_{ik} & = 1 & \forall i \in N \\ & p_{ik} & \in \{0,1\} & \forall i, k \in N. \end{array}\right. \tag{2}$$

We observe that the constraints of the problem above are those that model permutations, i.e., assignment constraints. Thus, variables p_{ik} are enforced to be binary by total unimodularity and could be relaxed to be non-negative. This fact allows us to compute the dual problem and its value will also be a valid form of evaluation for $\sum\limits_{k\in N} \lambda_k d_{(k)}$:

$$\sum_{k\in N} \lambda_k d_{(k)} = \begin{cases} \text{(BHP)} \quad \min & \sum\limits_{\ell\in N} u_\ell + \sum\limits_{i\in N} v_i \\ \qquad\qquad \text{s.t.} & u_\ell + v_i \geq \lambda_\ell d_i \quad \forall i,\ell \in N. \end{cases} \tag{3}$$

In the following we will denote the formulation in the right hand side of (3) as BHP because it appears in [Blanco et al., 2014] for the first time.

Remark 1. *Under nonnegativity conditions of the vectors d and λ, the linear constraints of the above assignment problem* (2) *can be relaxed to be less than or equal to* 1*. This results in non-negative dual multipliers u and v. The implication is that the representation of $\sum_{k\in N} \lambda_k d_{(k)}$ is modeled by a simplified problem*

$$\sum_{k\in N} \lambda_k d_{(k)} = \begin{cases} \min & \sum\limits_{\substack{\ell\in N:\\ \lambda_\ell>0}} u_\ell + \sum\limits_{i\in N} v_i \\ \text{s.t.} & u_\ell + v_i \geq \lambda_\ell d_i \quad \forall i,\ell\in N: \lambda_\ell > 0 \\ & u_\ell \geq 0 \quad \forall \ell \in N: \lambda_\ell > 0 \\ & v_i \geq 0 \quad \forall i \in N. \end{cases}$$

In particular, the above discussion leads to an alternative representation to the one in (1) for $S_k(d)$. We state that result in the following lemma.

Lemma 2. *Assuming that $d \geq \mathbf{0}$, the following is a valid representation of $S_k(d)$:*

$$\begin{aligned} S_k(d) = \min \quad & \sum_{\ell\geq k} u_\ell + \sum_{i\in N} v_i \\ \text{s.t.} \quad & u_\ell + v_i \geq d_i \quad \forall i \in N, \forall \ell \geq k \\ & u_\ell \geq 0 \quad \forall \ell \geq k \\ & v_i \geq 0 \quad \forall i \in N. \end{aligned}$$

Now, to represent the ordered median value of d for any $\lambda \in \mathbb{R}^n$, namely $\sum_{k\in N} \lambda_k d_{(k)}$, we can resort to the following observation whose proof is direct and left to the reader.

Lemma 3. *It holds*

$$\sum_{k\in N} \lambda_k d_{(k)} = \Delta_1 \sum_{i\in N} d_i + \sum_{k\geq 2} \Delta_k S_k(d) = \sum_{k\geq 1} \Delta_k S_k(d) \tag{4}$$

where $\lambda_0 := 0$ and $\Delta_k := \lambda_k - \lambda_{k-1}$ for all $k \in N$.

2.2 The p-median problem

Based on Lemma 3 we can exploit the different representations of $S_k(d)$ to enforce new formulations of the ordered median objective with applications to different combinatorial objects. In particular, this paper focuses on the application to the so-called p-median polytope. Consider variables $y_j = 1$ if the facility at position j is open and zero otherwise; and $x_{ij} = 1$ if client placed at position i is allocated to the facility at j. Then, the p-median polytope is:

$$X = \{x \in \mathbb{R}^{n\times n}, y \in \mathbb{R}^n : \sum_{j\in N} y_j = p, \sum_{j\in N} x_{ij} = 1 \ \forall i \in N,$$

$$x_{ij} \le y_j \ \forall i, j \in N, \ 0 \le y_j \le 1 \ \forall j \in N, \ x_{ij} \ge 0 \ \forall i, j \in N\}.$$

In the following, for the sake of readability, we will also refer to the integer lattice points within X as X_I.

Based on this idea we can formulate the Discrete Ordered Median Location Problem by means of several mixed integer linear formulations. Next, we present the simplest cases, those that correspond to the choice of nondecreasing lambda weights.

Assume that we are given a cost matrix $C = (c_{ij})$, where c_{ij} represents the cost of allocating the client placed at position i to the facility opened at j, $c_{ij} \ge 0$. The goal of DOMP is to find a set of p facilities to open so as to minimize the allocation costs of clients to facilities evaluated with an ordered median objective function.

It is straightforward to realize that for a given feasible point $(x, y) \in X_I$, its assignment costs are

$$c(x) := (\sum_{j\in N} c_{1j}x_{1j}, \dots, \sum_{j\in N} c_{nj}x_{nj}).$$

Therefore, the evaluation of the ordered median function for a given vector $(\lambda_1, \dots, \lambda_n)$ results in:

$$\sum_{i\in N} \lambda_i c_{(i)}(x),$$

where $c_{()}(x) = (c_{(1)}(x), \dots, c_{(n)}(x))$ is a reordering of $c(x)$ that satisfies $c_{(1)}(x) \le c_{(2)}(x) \le \cdots \le c_{(n)}(x)$.

3 The Monotone Discrete Ordered Median Problem

Let us assume, in this section, that $0 \le \lambda_1 \le \cdots \le \lambda_n$, and denote this as the *monotonicity assumption.* Then, using (4) replacing $S_k(c(x))$ by its valid formulation given by (1), one obtains the following formulation for the Monotone Discrete Ordered Median Problem (MDOMP).

$$
\begin{array}{llrl}
(\mathrm{MDOMP}_{OT_0}) & \min & \displaystyle\sum_{k\in N} \Delta_k \left((n-k+1)t_k + \sum_{i\in N} z_{ik} \right) & \\
& \text{s.t.} & \displaystyle\sum_{j\in N} c_{ij}x_{ij} - t_k \le z_{ik} & \forall i, k \in N \\
& & (x, y) \in X_I & \\
& & z_{ik} \ge 0 & \forall i, k \in N.
\end{array}
$$

Once again, under the assumption of non-negative c_{ij} costs, we can assume that $t_k \ge 0$ for all $k \in N$. It is also well-known that if the costs are non-negative, in an optimal solution of the DOMP, allocations of clients are always done to the closest facility. This, in turns, implies that the p-median polytope does not need to include extra constraints to enforce that property. Nevertheless, if we consider some negative Δ, that property is lost and it must be enforced with *closest assignment constraints* [Espejo et al., 2012] added to the corresponding formulation.

Remark 2. *The above formulation can be strengthened taking advantage of the ties in the consecutive values of the λ-vector. Let us assume that there are Q different "blocks", each one with q_k replications, $k = 1, \dots, Q$:*

$$\lambda = (\underbrace{\lambda_1, \dots, \lambda_1}_{q_1}, \underbrace{\lambda_2, \dots, \lambda_2}_{q_2}, \dots, \underbrace{\lambda_Q, \dots, \lambda_Q}_{q_Q}).$$

Hence, taking $\Delta_k = \lambda_k - \lambda_{k-1}$ for $k = 2, \dots, Q$ and $\Delta_1 = \lambda_1$, the following is also a valid formulation for the MDOMP*:*

$$
\begin{array}{lllr}
(\mathrm{MDOMP}_{OT}) & \min & \displaystyle\sum_{k=1}^{Q} \Delta_k \left(\left(\sum_{k'=k}^{Q} q_{k'} \right) t_k + \sum_{i \in N} z_{ik} \right) & \\
& \text{s.t.} & \displaystyle\sum_{j \in N} c_{ij} x_{ij} - t_k \leq z_{ik} & \forall i \in N, k = 1, \dots, Q \\
& & (x, y) \in X_I & \\
& & z_{ik} \geq 0 & \forall i \in N, k = 1, \dots, Q.
\end{array}
$$

Analogously, we also obtain a valid formulation of MDOMP for general nondecreasing λ-weights using Lemma 2. We will enrich that formulation forcing the combination of Lemmas 2 and 3. Obviously, this results in using a larger number of variables and constraints, since we wish to have explicitly all k-sums in the objective function. Our goal will become transparent in the next sections. Indeed, the formulation that we announce is:

$$
\begin{array}{lllr}
(\mathrm{MDOMP}_{BHP}) & \min & \displaystyle\sum_{\substack{k \in N: \\ \Delta_k > 0}} \Delta_k \left(\sum_{\ell \geq k} u_{k\ell} + \sum_{i \in N} v_{ki} \right) & \\
& \text{s.t.} & \displaystyle u_{k\ell} + v_{ki} \geq \sum_{j \in N} c_{ij} x_{ij} & \forall i, k \in N: \ \Delta_k > 0, \ \forall \ell \geq k \\
& & (x, y) \in X_I & \\
& & u_{k\ell} \geq 0 & \forall k \in N: \ \Delta_k > 0, \ \forall \ell \geq k \\
& & v_{ik} \geq 0 & \forall k \in N: \ \Delta_k > 0, \ \forall i \in N.
\end{array}
$$

For the sake of readability, let us denote by z_A^{LP} the optimal value of the LP-relaxation of (MDOMP_A). Next, we analyze the relation between the two models for the MDOMP previously developed.

Theorem 1. $z_{OT_0}^{LP} = z_{BHP}^{LP}$.

Proof. Let us assume a vector λ that satisfies the monotonicity assumption. We observe that for a given feasible point $(x, y) \in X$ the evaluation of $\sum_{k \in N} \lambda_k c_{(k)}(x)$, i.e., the omf for the assignment cost $c(x)$, is equivalently given as

$$\sum_{k\in N}\lambda_k c_{(k)}(x) = \left\{\begin{array}{llll} \min & \displaystyle\sum_{k\in N} u_k + \sum_{i\in N} v_i & & \\ \text{s.t.} & u_k + v_i & \geq \lambda_k \displaystyle\sum_{j\in N} c_{ij}x_{ij} & \\ & u_k & \geq 0 & \forall k \in N \\ & v_i & \geq 0 & \forall i \in N \end{array}\right.$$

$$= \left\{\begin{array}{llll} \min & \displaystyle\sum_{k\in N} \Delta_k((n-k+1)t_k + \sum_{i\in N} z_{ik}) & & \\ \text{s.t.} & \displaystyle\sum_{j\in N} c_{ij}x_{ij} - t_k & \leq z_{ik} & \forall i, \forall k \in N \\ & z_{ik} & \geq 0 & \forall i, \forall k \in N. \end{array}\right.$$

Therefore, the optimal values of the linear relaxations of (MDOMP$_{OT_0}$) and (MDOMP$_{BHP}$) coincide.

4 Some families of new formulations for DOMP based on OT

The extraordinary performance shown by the above formulations in the *monotone* cases (see Section 6.1) leads us to extend the rationale behind the k-sum representations to a more general framework where monotonicity of λ is lost.

Recall that we have already assumed in Section 3 the nonnegativity of the costs c_{ij}, $i, j \in N$, and parameters λ_k, $k \in N$. Consider again $S_k(c(x)) = \sum_{j\geq k} c_{(j)}(x)$. Based on Lemma 3 we have the validity of the following equation:

$$\min_{x\in X_I} \sum_{k\in N} \lambda_k c_{(k)}(x) = \min_{x\in X_I} \sum_{k\in N} \Delta_k S_k(c(x)). \tag{5}$$

The problem in the right hand side of (5) can be re-written as:

$$\min_{x\in X_I} \sum_{\substack{k\in N:\\ \Delta_k>0}} \Delta_k S_k(c(x)) + \sum_{\substack{k\in N:\\ \Delta_k<0}} \Delta_k S_k(c(x)). \tag{6}$$

The representations of the k-sum problem by means of one of the linear programs given in (MDOMP$_{OT}$) and (MDOMP$_{BHP}$) can be plugged into (6) provided that the coefficient Δ_k is positive. Otherwise, the validity is lost. This fact was already observed in [Puerto et al., 2017] although the goal of that paper was mainly theoretical and not focused on developing formulations.

Clearly, for all $k \in N$ with $\Delta_k > 0$, $S_k(c(x))$ can be replaced in (6) by the corresponding optimization problem (MDOMP$_{OT_0}$) or (MDOMP$_{BHP}$). However, if $\Delta_k < 0$, some alternative representations, most likely given by a mixed integer linear program, are required. We pursue this goal in the rest of the paper.

4.1 New radius formulation for the general DOMP (DOMP$_{OTr1}$)

Consider a fixed $(x, y) \in X_I$. A first valid representation for the k-sum, $S_k(c(x))$, as a mixed integer linear program, regardless of the sign of the Δ-coefficients, can be obtained via the radius r_{kh} variables that were applied for instance in [Elloumi et al., 2004, Puerto, 2008, García et al., 2011, Puerto et al., 2011]. By sorting the different nonzero values in the cost matrix C with g the number of different values and $G = \{1, \ldots, g\}$, we obtain

$$0 =: c_{(0)} < c_{(1)} < \ldots < c_{(g)} := \max_{i,j \in N} c_{ij}.$$

Then the radius variables are defined for $\ell \in N$ and $h \in G$ as follows:

$$r_{\ell h} = \begin{cases} 1 & \text{if the } \ell\text{-th allocation cost is at least } c_{(h)}, \\ 0 & \text{otherwise.} \end{cases}$$

Using these variables, the value of $S_k(c(x))$ can be obtained by solving the following problem:

$$\begin{aligned} \min \; & \sum_{\ell \geq k} \left(r_{\ell g} c_{(g)} + \sum_{h=1}^{g-1} (r_{\ell h} - r_{\ell,h+1}) c_{(h)} \right) & \\ \text{s.t. } & r_{\ell-1,h} \leq r_{\ell h} & \forall \ell \geq 2, \; \forall h \in G \\ & \sum_{\ell \in N} r_{\ell h} = \sum_{\substack{i,j \in N: \\ c_{ij} \geq c_{(h)}}} x_{ij} & \forall h \in G \\ & r_{\ell h} \in \{0, 1\} & \forall \ell \in N, \; \forall h \in G. \end{aligned} \tag{7}$$

Next, we apply this formulation with the assignment costs coming from variables $(x, y) \in X_I$. However, if its objective function appears with a negative coefficient within another problem, it is not true anymore that the location variables x in the p-median polytope will take the assignment given by the smallest possible costs. This must be enforced by the formulation by adding the so-called closest assignment constraints, see e.g. [Espejo et al., 2012]. In this case, we can use the following set of inequalities:

$$\sum_{\substack{j \in N: \\ c_{ij} > c_{im}}} x_{ij} + y_m \leq 1 \quad \forall i, m \in N. \tag{8}$$

Let $\alpha := \min\{k \in N : \Delta_k < 0\}$. The first valid formulation of DOMP with non-monotone λ-vector can be obtained combining (4), (MDOMP$_{OT_0}$), (7) and (8):

$$\begin{aligned} (\text{DOMP}_{OTr1}) \quad \min \; & \sum_{\substack{k \in N: \\ \Delta_k > 0}} \Delta_k \left((n - k + 1) t_k + \sum_{i \in N} z_{ik} \right) & \\ & + \sum_{\substack{k \in N: \\ \Delta_k < 0}} \Delta_k \sum_{\ell \geq k} \left(r_{\ell g} c_{(g)} + \sum_{h=1}^{g-1} (r_{\ell h} - r_{\ell,h+1}) c_{(h)} \right) & (9) \\ \text{s.t. } & t_k + z_{ik} \geq \sum_{j \in N} c_{ij} x_{ij} & \forall i, k \in N : \; \Delta_k > 0 \quad (10) \\ & \sum_{\substack{j \in N: \\ c_{ij} > c_{im}}} x_{ij} + y_m \leq 1 & \forall i, m \in N \quad (11) \end{aligned}$$

$$r_{\ell-1,h} \leq r_{\ell h} \qquad \forall \ell \geq \alpha,\ \forall h \in G \tag{12}$$

$$\sum_{\ell \geq \alpha} r_{\ell h} = \sum_{\substack{i,j \in N: \\ c_{ij} \geq c_{(h)}}} x_{ij} \qquad \forall h \in G \tag{13}$$

$$(x, y) \in X_I \tag{14}$$

$$z_{ik}, t_k \geq 0 \qquad \forall i, k \in N:\ \Delta_k > 0 \tag{15}$$

$$r_{\ell h} \in \{0, 1\} \qquad \forall \ell \geq \alpha,\ \forall h \in G. \tag{16}$$

The objective function (9) has two parts. The first part computes the sum of the k-sums terms ($S_k(c(x))$) for positive Δ_k, whereas the second part evaluates, according to (7), the terms with negative Δ_k. Constraints (10) are the same that appear in formulation (1) but applied to the allocation costs in the p-median polytope. Therefore, these constraints allow for a correct representation of $S_k(c(x))$ whenever $\Delta_k > 0$. Constraints (11) are closest assignment constraints. Constraints (12) and (13) are those that ensure the correct sorting of allocation costs in positions k with $\Delta_k < 0$. Finally, (14)-(16) define the range of the variables in the model.

4.2 Another radius formulation (DOMP$_{OTr2}$)

The above formulation can be strengthened replacing the definition of the radius variables following [Marín et al., 2009]. Define for each row $i \in N$ of the cost matrix C, g_i as the number of different non-zero elements in row i and $G_i := \{1, \ldots, g_i\}$. Thus, we obtain the ordering

$$c^i_{(0)} := 0 < c^i_{(1)} < \ldots < c^i_{(g_i)} := \max_{j \in N} c_{ij}$$

of the row. In addition, let

$$\ell^i_h = \begin{cases} \min\{s : c^i_{(s)} \geq c_{(h)}\} & \text{if } c_{(h)} \leq c^i_{(g_i)}, \\ g_i + 1 & \text{otherwise.} \end{cases}$$

Then we can define the variables w_{ih}, for all $i \in N$, $h \in G_i$:

$$w_{ih} = \begin{cases} 1 & \text{if the allocation cost for } i \text{ is at least } c^i_{(h)}, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

We add for convenience $w_{i,g_i+1} := 0$, for all $i \in N$. Next, we also consider the new set of variables ρ_{kh}, for all k such that $\Delta_k < 0$ and $h \in G$, defined as the number of times that an allocation cost greater than or equal to $c_{(h)}$ is used in the computation of $S_k(c(x))$.

Now, we obtain, as in (DOMP$_{OTr1}$), the following formulation:

$$(\text{DOMP}_{OTr2}) \quad \min \sum_{\substack{k \in N: \\ \Delta_k > 0}} \Delta_k((n-k+1)t_k + \sum_{i \in N} z_{ik}) + \sum_{\substack{k \in N: \\ \Delta_k < 0}} \Delta_k \sum_{h \in G} (c_{(h)} - c_{(h-1)})\rho_{kh} \tag{18}$$

$$
\begin{aligned}
\text{s.t. } & w_{ih} \geq 1 - \sum_{\substack{j \in N: \\ c_{ij} < c^i_{(h)}}} y_j && \forall i \in N, \ \forall h \in G_i && (19)\\
& \sum_{j \in N} y_j = p && && (20)\\
& w_{ih} + y_j \leq 1 && \forall i, j \in N, \ \forall h \in G_i : \ c_{ij} < c^i_{(h)} && (21)\\
& t_k + z_{ik} \geq \sum_{h \in G_i} (c^i_{(h)} - c^i_{(h-1)}) w_{ih} && \forall i \in N, \forall k \in N : \ \Delta_k > 0 && (22)\\
& \rho_{kh} \leq \sum_{\substack{i \in N: \\ \ell_{ih} \leq g_i}} w_{i\ell_{ih}} && \forall k \in N, \forall h \in G : \Delta_k < 0 && (23)\\
& \rho_{kh} \leq n - k + 1 && \forall k \in N, \forall h \in G : \Delta_k < 0 && (24)\\
& y_j \in \{0, 1\} && \forall j \in N && (25)\\
& z_{ik}, t_k \geq 0 && \forall i, k \in N : \ \Delta_k > 0 && (26)\\
& \rho_{kh} \geq 0 && \forall k, \ \forall h \in G : \ \Delta_k < 0 && (27)\\
& w_{ih} \geq 0 && \forall i \in N, \ h \in G_i. && (28)
\end{aligned}
$$

The objective function (18) has two parts: the first one that accounts for the sum of the terms $S_k(c(x))$ for $\Delta_k > 0$, and the second one does the same for $\Delta_k < 0$. Constraints (19) and (21) enforce closest assignment of clients to facilities since w_{ih} assumes value 1 whenever all the facilities that give allocation costs to client i less than $c_{(h)}$ are closed and client i can be assigned at a cost at least $c_{(h)}$ only if any plant j satisfying $c_{ij} < c_{(h)}$ is closed. The next constraints, namely (20), assure that p facilities are opened. Constraints (21) together with (19) allow one to define the w variables as continuous. Constraints (22) define the terms $S_k(c(x))$ for $\Delta_k > 0$. Finally, constraints (23) and (24) ensure the correct definition of the ρ and w variables. Indeed, (23) limit the number of addends greater than or equal to $c_{(h)}$ in the k-sum $S_k(c(x))$, with $\Delta_k < 0$, whereas (24) state an upper bound on that number of addends. The range of the variables of the problem is given in (25)-(28).

Observe that the variables w_{ih} are defined as continuous although by the combination of constraints (19) and (21) they will assume binary values in any feasible solution. As a consequence, variables ρ will also assume integer values since they are bounded above by integer values and they appear in the objective function (18) with a negative coefficient.

4.3 A big M formulation for the general DOMP (DOMP$_{OT\gamma}$)

As mentioned above, one can obtain different formulations using different representations of $S_k(c(x))$ for those k such that $\Delta_k < 0$. In this section, we propose another formulation based on the same rationale that in the papers by [Labbé et al., 2017] and [Nickel and Velten, 2017]. To this end, we define variables γ_{ik} that assume value one if the allocation cost of client i goes sorted in position k and zero otherwise; and variables ω_{ik} that take the value of the allocation cost of client i if it goes in position k and zero otherwise. Then it follows that, for fixed $(x, y) \in X$ (regardless whether (x, y) are integer or not), we can compute $S_k(c(x))$ as the following maximization problem:

$$
\begin{array}{rll}
S_k(c(x)) = \max & \displaystyle\sum_{i\in N} \omega_{ik} & \\
\text{s.t.} & \displaystyle\sum_{i\in N} \gamma_{ik} = n-k+1 & \\
& \omega_{ik} \le M\gamma_{ik} & \forall i \in N \\
& \omega_{ik} \le \displaystyle\sum_{j\in N} c_{ij}x_{ij} & \forall i \in N \\
& \gamma_{ik} \in \{0,1\} & \forall i \in N \\
& \omega_{ik} \ge 0 & \forall i \in N.
\end{array} \tag{29}
$$

Next, combining (10), (11) and (29) results in the following valid formulation for DOMP:

$$
(\text{DOMP}_{OT\gamma}) \quad \min \sum_{\substack{k\in N:\\ \Delta_k>0}} \Delta_k \left((n-k+1)t_k + \sum_{i\in N} z_{ik} \right) + \sum_{\substack{k\in N:\\ \Delta_k<0}} \Delta_k \sum_{i\in N} \omega_{ik} \tag{30}
$$

$$
\begin{array}{lll}
\text{s.t.} & (10),\ (11) & \\
& \displaystyle\sum_{i\in N} \gamma_{ik} = n-k+1 & \forall k \in N:\ \Delta_k < 0 \quad (31) \\
& \omega_{ik} \le M_i\gamma_{ik} & \forall i,k \in N:\ \Delta_k < 0 \quad (32) \\
& \omega_{ik} \le \displaystyle\sum_{j\in N} c_{ij}x_{ij} & \forall i,k \in N:\ \Delta_k < 0 \quad (33) \\
& \gamma_{ik} \in \{0,1\} & \forall i,k \in N:\ \Delta_k < 0 \quad (34) \\
& \omega_{ik} \ge 0 & \forall i,k \in N:\ \Delta_k < 0 \quad (35) \\
& z_{ik}, t_k \ge 0 & \forall i,k \in N:\ \Delta_k > 0 \quad (36) \\
& (x,y) \in X_I, &
\end{array}
$$

where we can use as M_i the k-th biggest element in $\{c_{ij}\}_{j\in N}$.

The objective function (30) has two parts: the first one that accounts for the sum of the terms $S_k(c(x))$ for $\Delta_k > 0$, and the second one that, according to (29), does the same for $\Delta_k < 0$. Constraints (10) are borrowed from (DOMP_{OTr1}) and allow for a valid representation of $S_k(c(x))$ for $\Delta_k > 0$. Constraints (11) enforce closest assignment of clients to facilities. The rest of the constraints come from (29) and are used to get a valid representation of $S_k(c(x))$ for negative Δ_k. Indeed, (31) ensure that the $S_k(c(x))$ term will have only $n-k+1$ elements, namely those corresponding to $c_{(k)}(x), \ldots, c_{(n)}(x)$. With (32) and (33), it is enforced that the allocation cost of client i only goes sorted in position k if $\gamma_{ik} \neq 0$. Finally, (34)-(36) define the range of the variables.

4.4 A three-index formulation for the general DOMP ($\text{DOMP}_{OT\theta}$)

Yet another formulation can be developed, based on ($\text{DOMP}_{OT\gamma}$) exploiting its structure to avoid defining the $O(n^2)$ γ binary variables and the use of big-M constraints. Instead, we define the variables θ_{ij}^k as 1 if the cost of allocating client i to facility j is sorted in position k and 0 otherwise. The reader should observe that although the variables are defined as integer, actually they can be considered as continuous since, as we will see, there always exists an optimal solution in integer values. This fact gives rise to the rather efficient formulation for the problem given below.

$$
\begin{aligned}
(\text{DOMP}_{OT\theta}) \quad \min \; & \sum_{\substack{k\in N:\\ \Delta_k>0}} \Delta_k \left((n-k+1)t_k + \sum_{i\in N} z_{ik} \right) \\
& + \sum_{\substack{k\in N:\\ \Delta_k<0}} \Delta_k \sum_{i\in N}\sum_{j\in N} c_{ij}\theta_{ij}^k && (37)\\
\text{s.t.} \quad & (10),\ (11) \\
& \sum_{i\in N}\sum_{j\in N} \theta_{ij}^k = n-k+1 && \forall k\in N:\ \Delta_k<0 \quad (38)\\
& \theta_{ij}^k \le x_{ij} && \forall i,j,k\in N:\ \Delta_k<0 \quad (39)\\
& \theta_{ij}^k \ge 0 && \forall i,j,k\in N:\ \Delta_k<0 \quad (40)\\
& z_{ik}, t_k \ge 0 && \forall i,k\in N:\ \Delta_k>0 \quad (41)\\
& (x,y)\in X_I.
\end{aligned}
$$

Once again, the objective function (37) has two parts: the first one accounts for the sum of the terms $S_k(c(x))$ for $\Delta_k > 0$, and the second one does the same for $\Delta_k < 0$. Constraints (10) allow for a valid representation of $S_k(c(x))$ for $\Delta_k > 0$. Constraints (11) enforce closest assignment of clients to facilities. The remaining constraints, namely (38) and (39), are used to get a valid representation of $S_k(c(x))$ for $\Delta_k < 0$. Finally, (40) define the range of the variables θ_{ij}^k. The reader may observe that since x_{ij} are binary and θ_{ij}^k appears in the objective function with a negative coefficient ($\Delta_k < 0$), if $x_{ij} = 1$ then $\theta_{ij}^k = 1$, thus in the optimal solution $\theta_{ij}^k \in \{0,1\}$.

Finally, we conclude this section with a relationship that holds between the two previous formulations ($\text{DOMP}_{OT\gamma}$) and ($\text{DOMP}_{OT\theta}$):

$$z_{OT\gamma}^{LP} \ge z_{OT\theta}^{LP}.$$

Indeed, the inequality can be verified checking that any feasible solution (t,z,x,y,γ,ω) to ($\text{DOMP}_{OT\gamma}$) induces a feasible solution to ($\text{DOMP}_{OT\theta}$) (t,z,x,y,θ) by taking $\theta_{ij}^k = x_{ij}\gamma_{ik}$ for all $i,j,k \in N$.

4.4.1 Valid inequalities for this formulation

For any of the above formulations, the relationship $S_k(c(x)) \le S_{k-1}(c(x))$ for all $k \ge 2$ translated into the adequate space of variables results in a valid inequality. In this case, it can be written for those k such that $\Delta_k > 0$, $\Delta_{k+1} < 0$ as

$$(n-k+1)t_k + \sum_{i\in N} z_{ik} \le \sum_{i\in N}\sum_{j\in N} c_{ij}\theta_{ij}^{k-1} \quad \forall k\in N:\ \Delta_k>0 \text{ and } \Delta_{k+1}<0.$$

Another observation that leads to derive valid inequalities comes from the fact that, by its own definition, the variables θ_{ij}^k are non-increasing in k for a given pair (i,j). This results in

$$\theta_{ij}^k \le \theta_{ij}^{\bar{k}} \quad \forall i,j\in N,\ \forall k>\bar{k}\ge 2, \quad \text{and } \Delta_k, \Delta_{\bar{k}} < 0.$$

4.5 Comparison

This subsection is devoted to compare our formulation (DOMP$_{OT\theta}$) with another one that, in addition, also uses the set of variables α_i^k, for $i, k \in N$, $\Delta_k < 0$ and corresponds to (29)-(35) in [Nickel and Velten, 2017]. We recall it for the sake of completeness.

$$
\begin{aligned}
(\text{DOMP}_{OT-NV}) \quad \min \; & \sum_{\substack{k\in N:\\ \Delta_k>0}} \Delta_k \left((n-k+1)t_k + \sum_{i\in N} z_{ik} \right) \\
& + \sum_{\substack{k\in N:\\ \Delta_k<0}} \Delta_k \sum_{i\in N}\sum_{j\in N} c_{ij}\theta_{ij}^k && (42)\\
\text{s.t.} \; & (10),\ (11) \\
& \sum_{i\in N} \alpha_i^k = n-k+1 && \forall k \in N:\ \Delta_k < 0 \quad (43)\\
& \theta_{ij}^k \le x_{ij} && \forall i,j,k \in N:\ \Delta_k < 0 \quad (44)\\
& \theta_{ij}^k \le \alpha_i^k && \forall i,j,k \in N:\ \Delta_k < 0 \quad (45)\\
& \theta_{ij}^k \ge 0 && \forall i,j,k \in N:\ \Delta_k < 0 \quad (46)\\
& z_{ik}, t_k \ge 0 && \forall i,k \in N:\ \Delta_k > 0 \quad (47)\\
& (x,y) \in X_I.
\end{aligned}
$$

One can easily check that (DOMP$_{OT\theta}$) is stronger than (DOMP$_{OT-NV}$). Indeed, we prove that any feasible solution (x, y, z, t, θ) of (DOMP$_{OT\theta}$) can be lifted to a feasible solution $(x, y, z, t, \theta, \alpha)$ of (DOMP$_{OT-NV}$). Define $\alpha_i^k = \sum_{j\in N} \theta_{ij}^k$ for all $i, k \in N$, $\Delta_k < 0$. Clearly, α satisfies (43) and α and θ satisfy (45). Moreover, one can also prove that (DOMP$_{OT\theta}$) is equivalent to a strict strengthening of (DOMP$_{OT-NV}$) replacing (45) by $\sum_{j\in N} \theta_{ij}^k \le \alpha_i^k$ for all $i, k \in N$, $\Delta_k < 0$ which follows since $\sum_{i\in N} x_{ij} = 1$.

The above mentioned strengthening is not a little change since it allows one to reduce the number of binary variables in $O(n^2)$. Moreover, as can be seen in the following example, it has an enormous impact in the quality of the lower bound given by the linear relaxations. We also note in passing that in the original formulation to model the general DOMP in [Nickel and Velten, 2017], there is a missing family of constraints. In DOMP, as in any other location problem, it is required that demand points are allocated to their closest service facility. This is automatically implied by formulations with positive coefficients in the objective function. However, if some coefficients are negative in the objective function this requirement must be imposed explicitly by some constraints. These constraints, at times called *closest assignment constraints* are missing in formulation (29)-(35) in [Nickel and Velten, 2017].

Example 1. *Consider the following network with $n = 6$ nodes and distance matrix C taken from Beasley library[Beasley, 2012]. We want to solve DOMP with a randomly generated* $\lambda = (0.62, 0.17, 0.54, 0.55, 0.02, 0.91)$ *and* $p = 2$.

$$
C = \begin{bmatrix}
143 & 127 & 185 & 171 & 78 & 115 \\
145 & 129 & 188 & 180 & 108 & 145 \\
99 & 83 & 142 & 134 & 154 & 134 \\
98 & 82 & 141 & 133 & 155 & 133 \\
70 & 54 & 113 & 105 & 160 & 123 \\
101 & 85 & 144 & 136 & 191 & 154
\end{bmatrix}
$$

If one runs formulations ($DOMP_{OT\theta}$) the linear relaxation of the problem is 236.358 *with a gap (*100*(optimal− linear relaxation bound)/optimal) at the root node of* 0.1%*. On the other hand, the correction of formulation ($DOMP_{OT-NV}$) carried out adding the required closest assignment constraints results in a linear relaxation of 37.0 with a gap at the root node of* 84.4%*. Extensive mumerical comparisons can be found in Section 6.*

5 Formulations based on BHP

In this section, our goal is to obtain new formulations of DOMP based on a different representation for the k-sums based on Lemma 2. Recall that we are interested in combinatorial objects defined by the points (x, y) belonging to the p-median polytope. Therefore, to model the DOMP we can apply the representations above to the assignment costs $c(x)$ induced by the points $(x, y) \in X_I$.

The new formulations follow easily from the corresponding ones in the previous section replacing, in each one of them, the representation of $S_k(c(x))$ given previously in Lemma 1 by the representation derived from Lemma 2. We include them for the sake of completeness. Variables and constraints are the same as those presented before and thus, we do not describe them again.

The following four formulations correspond, in the given order, with formulations (DOMP$_{OTr1}$), (DOMP$_{OTr2}$), (DOMP$_{OT\gamma}$) and (DOMP$_{OT\theta}$) after the substitution mentioned above.

5.1 BHP radius formulation obtained from (DOMP$_{OTr1}$)

We denote by (DOMP$_{BHPr1}$) the formulation that comes from (DOMP$_{OTr1}$) replacing the representation of $S_k(c(x))$ using Lemma 2. It results in the following.

$$
\begin{aligned}
(\mathrm{DOMP}_{BHPr1}) \quad \min & \sum_{k:\Delta_k>0}^{n} \Delta_k \left(\sum_{\ell=k}^{n} u_{k\ell} + \sum_{i\in N} v_{ki} \right) \\
& + \sum_{k:\Delta_k<0}^{n} \Delta_k \sum_{\ell=k}^{n} \left(r_{\ell g} c_{(g)} + \sum_{h=1}^{g-1} (r_{\ell h} - r_{\ell h+1}) c_{(h)} \right) && (48) \\
\text{s.t. } & (11), (12), (13) \\
& u_{k\ell} + v_{ki} \geq \sum_{j\in N} c_{ij} x_{ij} && i, k \in N,\ \ell = k, \ldots, n : \Delta_k > 0 \quad (49) \\
& u_{kl}, v_{ki} \geq 0 && i, \ell = k, \ldots, n : \Delta_k > 0 \quad (50) \\
& r_{\ell h} \in \{0, 1\} && \forall \ell \geq \alpha,\ \forall h \in G \\
& (x, y) \in X_I.
\end{aligned}
$$

5.2 BHP radius formulation obtained from (DOMP$_{OTr2}$)

Formulation (DOMP$_{BHPr2}$) is the one that comes from (DOMP$_{OTr2}$) replacing the representation of $S_k(c(x))$ using Lemma 2. It results in the following.

$$
\begin{aligned}
(\mathrm{DOMP}_{BHPr2}) \quad \min & \sum_{\substack{k\in N:\\ \Delta_k>0}} \Delta_k \left(\sum_{\ell=k}^{n} u_{k\ell} + \sum_{i\in N} v_{ki}\right) \\
& + \sum_{\substack{k\in N:\\ \Delta_k<0}} \Delta_k \sum_{h\in G}(c_{(h)} - c_{(h-1)})\rho_{kh} && (51)\\
\text{s.t. } & (19),(20),(21),(23),\ (24) \\
& u_{k\ell} + v_{ki} \geq \sum_{h\in G_i}(c^i_{(h)} - c^i_{(h-1)})w_{ih} \qquad \forall i,k,\ell \in N:\ \Delta_k>0, \ell\geq k && (52)\\
& y_j \in \{0,1\} \qquad \forall j\in N \\
& u_{k\ell}, v_{ki} \geq 0 \qquad i,\ell = k,\dots,n : \Delta_k > 0 && (53)\\
& \rho_{kh} \geq 0 \qquad \forall k,\ \forall h\in G:\ \Delta_k<0 \\
& w_{ih} \geq 0 \qquad \forall i\in N,\ h\in G_i.
\end{aligned}
$$

5.3 BHP formulation obtained from (DOMP$_{OT\gamma}$)

Formulation (DOMP$_{BHP\gamma}$) is the one that comes from (DOMP$_{OT\gamma}$) replacing the representation of $S_k(c(x))$ using Lemma 2. It results in the following.

$$
\begin{aligned}
(\mathrm{DOMP}_{BHP\gamma}) \quad \min & \sum_{k:\Delta_k>0}^{n} \Delta_k \left(\sum_{\ell=k}^{n} u_{k\ell} + \sum_{i\in N} v_{ki}\right) + \sum_{k:\Delta_k<0}\sum_{i\in N}\Delta_k \omega_{ik} \\
\text{s.t. } & (11),(31),(32),(33),(49) \\
& \gamma_{ik} \in \{0,1\} \qquad \forall i,k\in N:\ \Delta_k<0 \\
& \omega_{ik} \geq 0 \qquad \forall i,k\in N:\ \Delta_k<0 \\
& u_{k\ell}, v_{ki} \geq 0 \qquad i,\ell = k,\dots,n : \Delta_k>0 \\
& (x,y)\in X_I.
\end{aligned}
$$

5.4 BHP formulation obtained from (DOMP$_{OT\theta}$)

Formulation (DOMP$_{OT\theta}$) is the one that comes from (DOMP$_{OT\theta}$) replacing the representation of $S_k(c(x))$ using Lemma 2. It results in the following.

$$
\begin{aligned}
(\mathrm{DOMP}_{BHP\theta}) \quad \min & \sum_{k:\Delta_k>0}^{n} \Delta_k \left(\sum_{\ell=k}^{n} u_{k\ell} + \sum_{i\in N} v_{ki}\right) + \sum_{k:\Delta_k<0}^{n}\Delta_k \sum_{i\in N}\sum_{j\in N} c_{ij}\theta^k_{ij} \\
\text{s.t. } & (11),(38),(39),(49) \\
& \theta^k_{ij} \geq 0 \qquad \forall i,j,k\in N:\ \Delta_k<0
\end{aligned}
$$

Notation	λ-vector	Name
T1	$(1,1,\dots,1,1)$	p-median
T2	$(0,0,\dots,0,1)$	p-center
T3	$(0,0,\dots,0,0,\underbrace{1,1,\dots,1,1}_{k})$	k-centrum, $k=n/2$
T4	$(\underbrace{0,\dots,0}_{k_1},1,\dots,1,\underbrace{0,\dots,0}_{k_2})$	k_1-k_2-Trimmed mean, $k_1=k_2=n/10$
T7	λ random	Random
T8	$(\alpha,\alpha,\dots,\alpha,\alpha,1)$	Centdian, $\alpha=0.5$
T9	$(\alpha,0,\dots,0,1-\alpha)$	Hurwitz criteria, $\alpha=0.5$
T10	$(-1,0,\dots,0,1)$	Range
T11	$(\underbrace{1,\dots,1}_{k_1},0,\dots,0,\underbrace{1,\dots,1}_{k_2})$	k_1-k_2-Antitrimmed mean, $k_1=k_2=n/10$

Table 1: Types of λ-vectors used in experiments (T1, T2, T3 and T8 correspond to monotone vectors)

$$
\begin{aligned}
& u_{k\ell}, v_{ki} \geq 0 && i,\ell=k,\dots,n : \Delta_k > 0\\
& (x,y) \in X_I.
\end{aligned}
$$

6 Computational experiments

This section reports the computational experiments done to compare the performance of our new formulations for MDOMP and DOMP. All our experiments have been carried out on a PC with two Intel Xeon processors with 3.46 GHz and 48 GB of RAM. The models were written in Mosel and solved using Xpress 7.7. The results of the experiments, split by formulation and instance by instance, are available in `https://github.com/DiegoPonceIMUS/MDOMP`.

6.1 Comparing specialized formulation for monotone λ

This section is devoted to test whether it is advisable to apply specialized formulations for the MDOMP as compared with the general ones already available in the literature [Marín et al., 2009, Labbé et al., 2017].

First of all, we would like to compare the performance of our new formulations (MDOMP$_{OT}$) and (MDOMP$_{BHP}$), valid only for monotone λ-weights, with the results in [Labbé et al., 2017]. In order to compare the results, we use the same cost matrices (i.e., random instances in which the elements of the cost matrix are integer numbers randomly generated between 10,000 and 100,000) and family of lambda vectors (see Table 1) considered in the paper [Labbé et al., 2017]. The reader may note that we are also using the same notation as in that paper to refer to previous formulations of DOMP. We have tested five different configurations of monotone λ-vector (T1, T2, T3, T7-monotone and T8) per size. For each considered size (number of clients) in this section we report on the average of 5 instances, thus the overall number of solved instances is 405 (3 different sizes of instances, 3 different number of servers, 5 cost matrices and 9 λ-vectors[1]).

Table 2 reports on the average of the 405 instances for formulation (DOMP$_4$(B&C-3)), described in [Labbé et al., 2017], (MDOMP$_{OT}$) and (MDOMP$_{BHP}$). In this table the reader can find the CPU time

[1] T7 has been replicated 5 times.

required to solve the problems whenever the optimal solution is found. In those cases in which some instances are not solved to optimality, we report between parenthesis the number of instances not solved to optimality. Further, we also report the maximum time required to solve all the instances and the integrality gap in percentage. The computational experiment illustrates the theoretical result obtained in Theorem 1: integrality gap of (MDOMP$_{OT}$) and (MDOMP$_{BHP}$) is the same. We observe that the specialized formulations (MDOMP$_{OT}$) and (MDOMP$_{BHP}$) take advantage of the monotone structure of the λ-vectors, which results in an important reduction of the CPU time as compared with the general formulations. The comparison between (MDOMP$_{OT}$) and (MDOMP$_{BHP}$) does not show a significant difference, at least in medium size instances. For this reason we have designed another computational experiment to compare these two formulations on bigger instances of MDOMP.

Solution approach	**Time (s)**	$\mathbf{T_{max}}$**(s)**	**GAP(%)**
$MDOMP_{BHP}(B\&B)$	0.13	0.93	2.62
$MDOMP_{OT}(B\&B)$	0.10	0.96	2.62
$DOMP_4(B\&C-3)$	338.97(11)	7200.00	5.10

Table 2: Results for different formulations for $n = 10, 20, 30$

In the following, we wish to test the size limit that can be handled with the monotone formulations. In this analysis, we fix a CPU time limit of 7200 seconds and consider instances with a number of clients ranging from 100 to 180 because even for this last size there are some cases that we could not solve to optimality.

We have solved 250 instances in this second computational experiment. The combinations of parameters were the following: 5 different number of clients, 5 cost matrices per each client size, 2 different number of servers and 5 different λ-vectors, as before.

We report the results in three tables (3-5). All the tables show the same information. First, they include the CPU time to solve the problems whenever the optimal solution is found and the number of unsolved instances between parenthesis. Second, they report the average number of nodes explored in the B&B tree.

Table 3 shows the behavior of the formulations depending on the number of clients (n). We can observe that, as the number of clients n increases, the problem becomes more difficult for both formulations, although (MDOMP$_{OT}$) outperforms (MDOMP$_{BHP}$) in CPU time and number of problems solved for $n \geq 160$.

	(**MDOMP**$_{\mathbf{OT}}$)		(**MDOMP**$_{\mathbf{BHP}}$)	
n	**Time(s)**	**Nodes**	**Time(s)**	**Nodes**
100	11.76	363	12.28	448
120	28.52	599	34.38	850
140	72.58	1880	103.52	2655
160	262.41	13722	622.29(1)	15572
180	518.95	13364	901.86(2)	15486

Table 3: Average CPU-Time and number of nodes by number of clients

We also wish to know the dependence of formulations with respect to the number of facilities to be opened. Table 4 compares the behavior of the formulations (MDOMP$_{OT}$) and (MDOMP$_{BHP}$), for two different number of facilities to be opened. We observe that once again (MDOMP$_{OT}$) has better performance, although in this case (MDOMP$_{BHP}$) outperforms (MDOMP$_{OT}$) for larger values of p. We note in passing that in both formulations we have applied the same preprocessing defined by Claim 1 in [Labbé et al., 2017].

	(MDOMP$_{\mathbf{OT}}$)		**(MDOMP$_{\mathbf{BHP}}$)**	
	Time(s)	**Nodes**	**Time(s)**	**Nodes**
$p = \lfloor \frac{n}{2} \rfloor$	47.95	951	29.18	1655
$p = \lfloor \frac{n}{3} \rfloor$	309.74	11020	640.55(3)	12349

Table 4: Average CPU-Time and number of nodes by available servers

In the final part of our first computational study, we compare the results in Table 5, classifying the problems by the corresponding λ-vectors. Here we should distinguish two different patterns: 1) we observe that the resolution times are rather competitive for the cases of the median ($T1$), center ($T2$) and cent-dian ($T8$); nevertheless, this is not the case for the k-centrum ($T3$) and the monotone random λ ($T7$) where the problems get more difficult and require very large branch-and-bound trees. At the moment, we do not have any clear explanation for this different behavior concerning the alternative λ-vectors.

	(MDOMP$_{\mathbf{OT}}$)		**(MDOMP$_{\mathbf{BHP}}$)**	
	Time(s)	**Nodes**	**Time(s)**	**Nodes**
T1	0.59	1	6.89	1
T2	1.26	13	1.26	13
T3	204.44	47657	946.37(2)	56429
T7	280.54	1240	410.49(1)	1316
T8	0.64	1	6.85	1

Table 5: Average CPU-Time and number of nodes by type of λ-vector

As a general conclusion of our computational experiments we can state that it is advisable to use the formulations that exploit the monotonicity in the λ-vector. In general, formulation (MDOMP$_{OT}$) performs better than (MDOMP$_{BHP}$) since it requires shorter CPU times, being the only exception the case where the number of servers is large (around $p = n/2$). In this last case one should use (MDOMP$_{BHP}$) as shown in Table 4.

6.2 New formulations for general DOMP

The goal of this section is to compare the performance of new formulations presented in Sections 4 and 5 for the general DOMP. We analyze four different classes of non-monotone λ-vectors, namely T4, T9, T10 and T11. We use a dataset consisting in $p - median$ instances from OR_Lib [Beasley, 2012]. In order to obtain the cost matrix C we use the procedure explained in [Labbé et al., 2017] which gives rise to instances of sizes $n = 50 - 200$ without symmetry and with very few repeated costs. The structure of the tables in this section is similar to those in Section 6.1, namely we report for each formulation the CPU time, the number of unsolved instances, the gap at termination, the integrality gap and the number of explored nodes of the B&B tree.

We first perform a preliminary study with smaller instances using the data of problems pmed1-pmed5 in Beasley's library, to discriminate among the most promising formulations in Section 4. Then, based on this pilot study we move forward to compare the four best formulations from Section 4 and Section 5 on larger size instances $pmed1 - pmed20$.

The results of our pilot analysis are summarized in Tables 6 and 7. There, we compare formulations (DOMP$_{OT\gamma}$), (DOMP$_{OT\theta}$), (DOMP$_{OTr1}$), (DOMP$_{OTr2}$), (DOMP$_{OT-NV}$), (DOMP$_{BHP\gamma}$), (DOMP$_{BHP\theta}$), (DOMP$_{BHPr1}$), and (DOMP$_{BHPr2}$). Note that formulations (DOMP$_{OT\gamma}$), (DOMP$_{OT\theta}$), (DOMP$_{BHP\gamma}$), and (DOMP$_{BHP\theta}$) differ from {(DOMP$_{OTri}$), (DOMP$_{BHPri}$)}, $i = 1, 2$ in the set of decision variables. The former are based on γ and θ variables whereas the latter are based on radius variables and their corresponding constraints. One can easily observe that formulations (DOMP$_{OT\theta}$) and (DOMP$_{OTr2}$) outperform (DOMP$_{OT\gamma}$) and (DOMP$_{OTr1}$). The same behavior is also observed for the corresponding formulation based on BHP.

These results lead us to include in our complete comparison only formulations (DOMP$_{OT\theta}$), (DOMP$_{OTr2}$), (DOMP$_{BHP\theta}$) and (DOMP$_{BHPr2}$), as they are the ones with the best performance in the pilot study. Table 8 summarizes the results of our comparative analysis.

For T4, T9 and T11 all the formulations are able to solve almost all the considered Beasley's instances ($pmed1 - pmed20$). In fact, the formulations proposed in this paper are very efficient for some particular λ structures as shown by the small gap at termination and computing time required to solve them, even for relatively large sizes ($n > 150$). It is remarkable that for T4, (DOMP$_{OT\theta}$) requires an average time of only 36.10 seconds and a very small number of nodes (8 in average) of the B&B trees over the entire set of instances. Recall that, as introduced for Table 2, numbers between parenthesis report the number of instances not solved to optimality within the time limit. In the trimmed mean λ structure, the formulations based on θ variables clearly outperform the radius formulations. Although in general (DOMP$_{OT\theta}$) requires less time to certify optimality, it is for this particular λ where the difference is the biggest one. We would also like to emphasize that instances with lambda type T10 are well-known to be very difficult (see [Marín et al., 2010]). However, formulations based on θ variables are able to solve 14 out of 20 instances within the time limit, what is a remarkable behavior for this particularly difficult λ structure.

	T4		**T9**		**T10**		**T11**		**Total**	
	Time(s)	**Gap**	**Time(s)**	**Gap**	**Time(s)**	**Gap**	**Time(s)**	**Gap**	**Time(s)**	**Gap**
DOMP$_{\mathbf{OT\gamma}}$	1624.88	0.00	81.76	0.00	20.73	0.00	442.57	0.00	**542.49**	**0.00**
DOMP$_{\mathbf{OT\theta}}$	0.82	0.00	**0.49**	0.00	**2.92**	0.00	**0.48**	0.00	**1.18**	**0.00**
DOMP$_{\mathbf{OTr1}}$	7.05	0.00	600.79	0.00	7200.26(5)	78.99	251.94	0.00	**2015.01(5)**	**19.75**
DOMP$_{\mathbf{OTr2}}$	4.32	0.00	1.74	0.00	26.30	0.00	2.31	0.00	**8.67**	**0.00**
DOMP$_{\mathbf{OT-NV}}$	6632.92(4)	60.61	5787.74(4)	4.54	23.87	0.00	318.41	0.00	**3190.74(8)**	**16.29**
DOMP$_{\mathbf{BHP\gamma}}$	1824.38	0.00	929.45	0.00	23.34	0.00	296.43	0.00	**768.40**	**0.00**
DOMP$_{\mathbf{BHP\theta}}$	**0.72**	0.00	1.14	0.00	4.99	0.00	1.15	0.00	**2.00**	**0.00**
DOMP$_{\mathbf{BHPr1}}$	9.77	0.00	3159.29(2)	8.89	5925.34(4)	79.43	223.66	0.00	**2329.51(6)**	**22.08**
DOMP$_{\mathbf{BHPr2}}$	6.82	0.00	1.89	0.00	38.62	0.00	3.20	0.00	**12.63**	**0.00**

Table 6: Average CPU-Time and gap at termination by different ordered median objective functions. Instances $pmed1 - pmed5$

	T4		T9		T10		T11		Total	
	GAP(%)	Nodes	GAP(%)	Nodes	GAP(%)	Nodes	GAP(%)	Nodes	GAP(%)	Nodes
$\mathbf{DOMP_{OT\gamma}}$	47.83	177044	11.10	29020	91.90	1223	8.84	243494	**39.92**	**112695**
$\mathbf{DOMP_{OT\theta}}$	0.07	1	2.66	23	73.90	870	1.66	29	**19.57**	**231**
$\mathbf{DOMP_{OTr1}}$	1.10	1	2.66	25150	100.00	236472	1.66	5932	**26.36**	**66889**
$\mathbf{DOMP_{OTr2}}$	2.97	5	2.66	13	91.21	583	1.66	27	**24.63**	**157**
$\mathbf{DOMP_{OT-NV}}$	118.51	156276	11.10	979528	73.90	924	8.84	27338	**53.09**	**291017**
$\mathbf{DOMP_{BHP\gamma}}$	47.83	198046	11.10	574939	91.90	949	8.84	138331	**39.92**	**228066**
$\mathbf{DOMP_{BHP\theta}}$	0.07	1	2.66	42	73.90	729	1.66	56	**19.57**	**207**
$\mathbf{DOMP_{BHPr1}}$	1.10	1	2.66	112794	100.00	130698	1.66	1708	**26.36**	**61300**
$\mathbf{DOMP_{BHPr2}}$	2.97	4	2.66	18	91.21	1102	1.66	23	**24.63**	**287**

Table 7: Average integrality gap and number of nodes of the branching tree by different ordered median objective functions. Instances $pmed1 - pmed5$

	T4		T9		T10		T11		Total	
	Time(s)	Gap	Time(s)	Gap	Time(s)	Gap	Time(s)	Gap	Time(s)	Gap
$\mathbf{DOMP_{OT\theta}}$	**36.10**	0.00	**384.81**(1)	0.13	**2699.55**(6)	7.59	**483.07**(1)	0.06	**900.88(8)**	**1.94**
$\mathbf{DOMP_{OTr2}}$	858.99(1)	0.13	406.74(1)	0.38	3810.06(9)	30.93	583.33(1)	0.09	**1414.78(12)**	**7.88**
$\mathbf{DOMP_{BHP\theta}}$	136.03	0.00	427.07(1)	0.38	3357.44(6)	14.41	699.20(1)	0.10	**1154.93(8)**	**3.72**
$\mathbf{DOMP_{BHPr2}}$	976.25	0.00	425.33(1)	0.38	4543.16(10)	42.07	682.61(1)	0.11	**1656.84(12)**	**10.64**

Table 8: Average CPU-Time and gap at termination, by different ordered median objective functions. Instances $pmed1 - pmed20$

7 Conclusions

This paper exploits new results on k-sum optimization to derive new formulations for the (MDOMP) and (DOMP). We present two different families of formulations that are defined by the rationale to represent k-sums, either with the OT formulation presented in (1) or with the BHP representation described in (3). Within these two families we compare formulations based on different sets of variables. Our computational results show that the best results come from the combination of the OT formulation with θ variables, namely formulation ($\mathrm{DOMP}_{OT\theta}$).

For the sake of readability, this paper has restricted itself to consider ordered median problems based on location problems, or in other words, based on allocation costs coming from the p-median polytope. Extensions of the same tools to other combinatorial objects such as shortest paths, matchings, spanning trees, etc, will be the subject of a follow up paper.

Acknowledgements

The first author was partially supported by the research project MTM2015-65915-R (MINECO, Spain). The second and third authors' research was supported by Spanish/FEDER grant number MTM2016-74983-C02-01.

References

Aouad, A. and Segev, D. (2019). The ordered k-median problem: surrogate models and approximation algorithms. *Mathematical Programming.*, 177:55 – 83.

Beasley, J. (2012). OR-Library. http://people.brunel.ac.uk/∼mastjjb/jeb/info.html.

Blanco, V. (2019). Ordered p-median problems with neighbourhoods. *Computational Optimization and Applications*, 73(2):603–645.

Blanco, V., Ali, S. E. H. B., and Puerto, J. (2014). Revisiting several problems and algorithms in continuous location wih $l_\tau-$norm. *Computational Optimization and Applications*, 58(3):563–595.

Deleplanque, S., Labbé, M., Ponce, D., and Puerto, J. (2020). A Branch-Price-and-Cut procedure for the Discrete Ordered Median Problem *INFORMS Journal on Computing*, https://doi.org/10.1287/ijoc.2019.0915

Elloumi, S., Labbé, M., and Pochet, Y. (2004). A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing*, 16(1):84–94.

Espejo, I., Marín, A., and Rodríguez-Chía, A. M. (2012). Closest assignment constraints in discrete location problems. *European Journal of Operational Research*, 219(1):49 – 58.

García, S., Labbé, M., and Marín, A. (2011). Solving large p-median problems with a radius formulation. *INFORMS Journal on Computing*, 23:546–556.

Kalcsics, J., Nickel, S., Puerto, J., and Tamir, A. (2002). Algorithmic results for ordered median problems. *Operations Research Letters*, 30:149–158.

Kalcsics, J., Nickel, S., Puerto, J., and Rodríguez-Chía, A.M. (2010). The ordered capacitated facility location problem. *TOP*, 18: 203–22.

Labbé, M., Ponce, D., and Puerto, J. (2017). A comparative study of formulations and solution methods for the discrete ordered p-median problem. *Computers & Operations Research*, 78:230–242.

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

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

Nickel, S. and Puerto, J. (2005). *Location Theory: A Unified Approach.* Springer Verlag.

Nickel, S. and Velten, S. (2017). Optimization problems with flexible objectives: A general modeling approach and applications. *European Journal of Operational Research*, 258:79–88.

Ogryczack, W. and Tamir, A. (2003). Minimizing the sum of the $k-$largest functions in linear time. *Information Processing Letters*, 85:117–122.

Olender, P. and Ogryczak, W. (2019). A revised variable neighborhood search for the discrete ordered median problem. *European Journal of Operational Research*, 274(2):445 – 465.

Puerto, J. (2008). A new formulation of the capacitated discrete ordered median problem with $\{0,1\}$-assignment. In *Operations Research Proceedings 2007*, volume 1, pages 165–170. Springer.

Puerto, J. (2019). An exact completely positive programming formulation for the discrete ordered median problem: an extended version. *Journal of Global Optimization*, https://doi.org/10.1007/s10898-019-00863-1:1–19.

Puerto, J., Ramos, A. B., and Rodríguez-Chía, A. M. (2011). Single-allocation ordered median hub location problems. *Computers & Operations Research.*, 38(2):559–570.

Puerto, J., Rodríguez-Chía, A. M., and Tamir, A. (2009). Minimax Regret Single-Facility Ordered Median Location Problems on Networks. *INFORMS Journal on Computing*, 21:77–89.

Puerto, J., Rodríguez-Chía, A. M., and Tamir, A. (2017). Revisiting k-sum optimization. *Mathematical Programming*, 165(2):579–604.