

Models for the Discrete Ordered Median Problem based on DC decomposition

Stefan Nickel ^{a,1}, Justo Puerto ^{b,c,1}, Simon Ramoser ^{a,1,*}, Alberto Torrejon ^{b,c}

^a Institute for Operations Research, Karlsruhe Institute of Technology, Kaiserstraße 89, 76133 Karlsruhe, Germany

^b Department of Statistics and Operational Research, University of Seville, Calle Tarfia s/n, 41012 Sevilla, Spain

^c Institute of Mathematics, University of Seville, Avenida Reina Mercedes s/n, 41012 Sevilla, Spain

ARTICLE INFO

Dataset link: [Instance and solution data for 'Models for the Discrete Ordered Median Problem based on DC decomposition' \(Original data\)](#)

Keywords:

Combinatorial optimization
Discrete Ordered Median Problem
DC decomposition
Facility location
Ordered optimization

ABSTRACT

The Discrete Ordered Median Problem is a flexible model for a large variety of location problems. It relies on sorting allocation costs non-decreasingly before applying a weight vector. Being a non-linear operation, the cost-sorting is particularly challenging from a modeling point of view. Promising approaches use radius variables or reformulations of the ordered median objective function leading to efficient mixed-integer linear programming models. This study introduces an alternative modeling paradigm that exploits a known Difference of Convex functions (DC) decomposition of ordered median functions. The proposed framework offers a systematic way to combine existing formulations and to derive new ones. To illustrate its potential, we develop several new models for the Discrete Ordered Median Problem based on this concept. An extensive computational study is conducted to evaluate their performance against current state-of-the-art formulations. The results show that the proposed approach can lead to highly competitive and previously undiscovered models.

1. Introduction

Looking for more flexible modeling approaches in Location Analysis, ordered median objective functions were introduced in the field. First on continuous and network models (Puerto et al., 1997; Nickel and Puerto, 1999; Rodríguez-Chía et al., 2000; Puerto and Fernández, 2000) and lately in the discrete setting. Since its formulation by Nickel (2001), the Discrete Ordered Median Problem (DOMP) has become a well-established combinatorial optimization problem in the location literature. The goal is to determine which subset of potential facilities should be opened so that customers are served optimally. Solutions are evaluated by a weighted sum of ordered customer allocation costs, that is, the costs are sorted in non-decreasing sequence and weighted according to their position in that sequence. By choosing different weight vectors, the DOMP generalizes classical location objectives. For instance, setting all weights equal to one recovers the classic median problem (minimizing the total sum of costs), whereas assigning a weight of one to the largest cost and zero to all others yields the center problem (minimizing the maximum cost). Additionally, the DOMP captures combinations of classical objectives, such as the centdian problem, where a weight of one is assigned to the largest cost and some weight in $[0, 1]$ is applied to all others. Beyond these standard objectives, appropriately adjusting the weight vector allows the DOMP to

model more sophisticated criteria related to fairness, equity, dispersion, or other preferences.

The main difficulty in modeling the DOMP lies in the non-linearity of the sorting operation. Nevertheless, a variety of mixed-integer linear programming (MILP) formulations have been proposed over the years. Early contributions focused on formulations based on assignment variables and their refinements (Domínguez-Marín, 2003; Boland et al., 2006; Nickel and Puerto, 2005). A major improvement was the introduction of covering-based formulations using radius variables (Puerto, 2008; Marín et al., 2009). More recently, it was exploited that ordered median functions can be expressed via sums of the k largest costs, which led to a new class of formulations building on k -sum optimization techniques (Nickel and Velten, 2017; Puerto et al., 2017). Subsequent studies have further refined this approach (Marín et al., 2020; Ljubić et al., 2024). Additionally, advanced methods to resolve the DOMP have been applied successfully (e.g., Deleplanque et al., 2020; Cherkesly et al., 2025) and extensions have been considered, including hub location problems (e.g., Puerto et al., 2011, 2016, 2013; Pozo et al., 2021), connected facility problems (e.g., Pozo et al., 2024), formulations incorporating neighborhood structures (e.g., Blanco, 2019), and models that account for envy minimization among customers (e.g., Blanco et al., 2024).

* Corresponding author.

E-mail addresses: stefan.nickel@kit.edu (S. Nickel), puerto@us.es (J. Puerto), simon.ramoser@kit.edu (S. Ramoser), atorrejon@us.es (A. Torrejon).

¹ Part of this work was carried out while visiting the Institute for Operations Research at the Karlsruhe Institute of Technology.

In parallel to the growing DOMP literature, ordered median functions – the type of objective function used in the DOMP – were studied independently from a more theoretical perspective. Basic results such as continuity, symmetry, or closedness under linear combinations were collected in Nickel and Puerto (2005). Drezner and Nickel (2009) found a constructive way to express any ordered median function f as a Difference of Convex functions (DC), meaning it can be written as $f(x) = g(x) - h(x)$ for all inputs x , with both g and h being convex. Grzybowski et al. (2011) characterized linear and convex ordered median functions. Further representations and properties were established by Grzybowski et al. (2015a,b, 2016, 2018).

With this work, we take a step toward bridging the gap between the practical and theoretical study of the DOMP by proposing a modeling framework based on the DC decomposition of ordered median functions constructed by Drezner and Nickel (2009). The key motivation is that formulating the problem becomes simpler when the objective function has the additional exploitable property of being convex or concave. Because any ordered median function can be split into a convex component $g(x)$ and a concave component $-h(x)$, we seek to exploit these structural properties independently and recombine them into a single formulation valid for the general case. More precisely, we show that two specialized models – one for the convex case and one for the concave case – can be merged into a single general model by decoupling the ordering logic while sharing the location-specific logic across both.

This generic approach is not limited to the DOMP; it can be applied to any optimization problem with a suitable separable structure. The area where it is particularly relevant is ordered optimization, whose literature now extends well beyond location analysis. For example, the Ordered Median Traveling Salesman Problem has recently been introduced to generalize the classical Traveling Salesman Problem by evaluating the travel cost of a tour using an ordered median objective function (Ljubić et al., 2026a). Other domains in which ordered objective functions have been successfully incorporated include vehicle routing (Ljubić et al., 2026b), scheduling (Nickel and Velten, 2017), portfolio selection (Benati et al., 2025; Cesarone and Puerto, 2025), and statistics and machine learning (Blanco et al., 2018; Marín et al., 2022; Puerto and Torrejon, 2025), among many others. For a more comprehensive overview of these and other applications, the interested reader is referred to Blanco et al. (2026). In any of these settings, provided the underlying problem constraints (such as tour construction in routing or task assignments in scheduling) can be decoupled from the ordering logic, our DC framework can be readily applied to exploit specialized models for the convex and concave case of the respective ordered problem.

Our work offers the following contributions to the literature.

- We propose and justify a novel high-level framework for modeling optimization problems with ordered median objective functions based on a DC decomposition of such functions. The key insight is that, under suitable assumptions, specialized formulations for the convex and concave components that share underlying problem constraints can be combined into a single, more general model.
- We apply this framework to the DOMP and develop several new MILP formulations by combining specialized models that exploit either convex or concave structures.
- We present an extensive computational study that compares these new formulations with current state-of-the-art DOMP models. The results show that the proposed approaches achieve similar or better computational performance than existing formulations across a variety of complex λ vectors.

The rest of this paper is structured as follows. Section 2 recalls the definition of the DOMP and reviews the previously mentioned k -sum and radius modeling approaches in greater detail. Our central idea is described in Section 3, where we derive both compact and exponential DOMP formulations. Section 4 presents the computational study. Final conclusions and future perspectives are given in Section 5.

2. The discrete ordered median problem

2.1. Problem definition

Consider a number $n \in \mathbb{Z}_{\geq 1}$ of client demand points, which we assume to simultaneously serve as candidate locations for facilities. This standard assumption can be made without loss of generality since it is possible to introduce dummy clients with zero service costs if a potential location is not at one of the actual demand points. The goal is to select a subset of exactly $p \in \mathbb{Z}_{\geq 1}$ locations, with $p \leq n$, at which facilities are opened.

Let $c \in \mathbb{R}_{\geq 0}^{n \times n}$ denote the matrix of service costs, where c_{ij} represents the cost incurred when client i is served from a facility located at site j . Given a non-empty set of open facilities $Y \subseteq [n]^2$ each client i is supposed to be assigned to its closest facility in Y , inducing the allocation cost $c_i(Y) := \min_{j \in Y} c_{ij}$ and the corresponding cost vector $c(Y) := (c_1(Y), \dots, c_n(Y))$. Throughout, facilities are assumed to be uncapacitated, that is, all demand assigned to a facility can be served without restrictions.

Then, the DOMP is concerned with finding a subset $Y \subseteq [n]$ with $|Y| = p$ so that the scalar product of the non-decreasingly sorted allocation cost vector $c(Y)$ and a given weight vector $\lambda \in \mathbb{R}^n$ is minimized.

The concatenation of sorting and scalar product defines what is commonly referred to as ordered median function.

Definition 1 (Ordered Median Function (Nickel and Puerto, 2005, Def. 1.1)). For a vector $x \in \mathbb{R}^n$, let $x_{\leq} := (x_{(1)}, x_{(2)}, \dots, x_{(n)})$ be a re-ordering of x such that $x_{(1)} \leq \dots \leq x_{(n)}$. Given $\lambda \in \mathbb{R}^n$, the associated ordered median $f_{\lambda} : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined by

$$f_{\lambda}(x) := \langle \lambda, x_{\leq} \rangle = \sum_{i=1}^n \lambda_i x_{(i)} \quad \forall x \in \mathbb{R}^n.$$

To summarize, using Definition 1, the DOMP can be formally expressed as the optimization task

$$\min_{\substack{Y \subseteq [n] \\ |Y|=p}} f_{\lambda}(c(Y)). \quad (1)$$

2.2. State-of-the-art modeling approaches

In this section, we review two of the most competitive DOMP modeling approaches: k -sum and radius formulations. These will also serve as benchmarks for our numerical experiments.

Both k -sum and radius models use two-index variables for modeling the location decisions and additional variables to handle the cost-sorting logic. The variables $x_{ij} \in \{0, 1\}$ are supposed to take the value 1 if and only if client i is served by facility j . Furthermore, $y_j \in \{0, 1\}$ should be equal to 1 if and only if facility j is opened. Then, the p -median location polytope is defined as

$$X := \left\{ \begin{array}{l} x \in [0, 1]^{n \times n}, y \in [0, 1]^n : \sum_{j=1}^n y_j = p; \\ \sum_{j=1}^n x_{ij} = 1 \quad \forall i \in [n]; \quad x_{ij} \leq y_j \quad \forall i, j \in [n]; \\ \sum_{\substack{j \in [n] \\ c_{ij} > c_{im}}} x_{ij} + y_m \leq 1 \quad \forall i, m \in [n] \end{array} \right\}.$$

Moreover, we denote the intersection of the polytope X with the set of integer lattice points by $X_I := X \cap (\mathbb{Z}^{n \times n} \times \mathbb{Z}^n)$. Every solution $(x, y) \in X_I$ represents a selection of exactly p opened facilities $Y = \{j \in [n] \mid y_j = 1\}$ as well as an assignment of each client to a facility $j^* \in Y$ – indicated by $x_{ij^*} = 1$ – that minimizes the client's distance to any facility $j \in Y$.

² For $n \in \mathbb{Z}_{\geq 1}$, we use the notation $[n] := \{1, \dots, n\}$.

Notice that the last family of inequalities defining X enforces closest assignment by disallowing the allocation of customer i to facility j if another facility m with $c_{im} < c_{ij}$ is opened. These constraints are redundant when the objective weights λ are non-negative, as in many classical problems such as the p -median and the p -center problem. However, since we consider general $\lambda \in \mathbb{R}^n$, we include these inequalities, following previous studies in the DOMP literature (e.g., [Marín et al., 2020](#); [Ljubić et al., 2024](#)).

All DOMP models in this paper rely on the definition of X_I . Therefore, any feasible DOMP solution must lie inside X_I . The additional sorting logic implemented by the k -sum and radius approaches is explained in Sections 2.2.1 and 2.2.2, respectively.

2.2.1. Formulations based on k -sums

As already mentioned, it is possible to exploit the fact that any ordered median function can be rewritten in terms of sums of the $n-k+1$ largest entries of the input vector. To begin with, we have the following lemma that can be proved by means of a short calculation (e.g., [Nickel and Velten, 2017](#)).

Lemma 1 (*k -Sum Representation of Ordered Median Functions*). *Let $\lambda \in \mathbb{R}^n$ and $f_\lambda : \mathbb{R}^n \rightarrow \mathbb{R}$ be the corresponding ordered median function. Then, with $\lambda_0 := 0$, it holds for every $x \in \mathbb{R}^n$:*³

$$f_\lambda(x) = \sum_{k=1}^n \lambda_k x_{(k)} = \sum_{k=1}^n (\lambda_k - \lambda_{k-1}) \sum_{j=k}^n x_{(j)} =: \sum_{k=1}^n \Delta_k^\lambda S_k(x).$$

Lemma 1 is useful since it opens the possibility to view the DOMP from the perspective of k -sum optimization which is a well-studied topic ([Papini and Puerto, 2004](#)). Particularly, [Ogryczak and Tamir \(2003\)](#) gave the following linear program to compute the sum of the $n-k+1$ largest values of a given vector $d \in \mathbb{R}^n$.

$$\begin{aligned} \min \quad & (n-k+1)t + \sum_{i=1}^n z_i \\ \text{s.t.} \quad & t + z_i \geq d_i \quad \forall i \in [n] \\ & z_i \geq 0 \quad \forall i \in [n] \\ & t \in \mathbb{R} \end{aligned} \quad (2)$$

Since d is not multiplied with any decision variable, this model can be conveniently used to compute $S_k(d)$ for any $k \in [n]$ even if d itself becomes variable, provided that the coefficient $\Delta_k^\lambda = (\lambda_k - \lambda_{k-1})$ is non-negative. However, for general λ , this is not always the case. Therefore, the objective function must be split into two parts according to the sign of Δ_k^λ :

$$f_\lambda(d) = \sum_{\substack{k=1 \\ \Delta_k^\lambda > 0}}^n \Delta_k^\lambda S_k(d) + \sum_{\substack{k=1 \\ \Delta_k^\lambda < 0}}^n \Delta_k^\lambda S_k(d).$$

A promising compact model that makes use of this approach has been presented by [Marín et al. \(2020\)](#). For $\Delta_k^\lambda < 0$, the authors suggest using three-index binary variables θ_{ij}^k indicating whether the cost of allocating customer i to facility j is in position k or higher of the ordered cost vector. They show that these variables can be relaxed without compromising correctness of the model.

For the ease of presentation, we make use of the notation $\Delta_+^\lambda := \{k \in [n] \mid \Delta_k^\lambda > 0\}$ and $\Delta_-^\lambda := \{k \in [n] \mid \Delta_k^\lambda < 0\}$ throughout the rest of this paper.

The final MILP, denoted as DOMP_θ , is then as follows.

$$\text{DOMP}_\theta : \min \sum_{k \in \Delta_+^\lambda} \Delta_k^\lambda \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \sum_{k \in \Delta_-^\lambda} \Delta_k^\lambda \sum_{i=1}^n \sum_{j=1}^n c_{ij} \theta_{ij}^k \quad (3a)$$

$$\text{s.t.} \quad t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\lambda \quad (3b)$$

$$\sum_{i=1}^n \sum_{j=1}^n \theta_{ij}^k = n - k + 1 \quad \forall k \in \Delta_-^\lambda \quad (3c)$$

$$\theta_{ij}^k \leq x_{ij} \quad \forall i, j \in [n], k \in \Delta_-^\lambda \quad (3d)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\lambda \quad (3e)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^\lambda \quad (3f)$$

$$\theta_{ij}^k \geq 0 \quad \forall i, j \in [n], k \in \Delta_-^\lambda \quad (3g)$$

$$(x, y) \in X_I \quad (3h)$$

Some of the most advanced DOMP formulations available in the literature are obtained by applying Benders reformulation to model DOMP_θ . For instance, one may “project out” the θ -variables, to obtain model DOMP_{θ_B} , as [Ljubić et al. \(2024, Prop. 2\)](#) show.

$$\text{DOMP}_{\theta_B} : \min \sum_{k \in \Delta_+^\lambda} \Delta_k^\lambda \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \sum_{k \in \Delta_-^\lambda} \Delta_k^\lambda \varphi_k \quad (4a)$$

$$\text{s.t.} \quad t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\lambda \quad (4b)$$

$$\begin{aligned} \varphi_k \leq & (n-k+1)c_{ij(h)} + \\ & \sum_{\substack{i,j \in [n] \\ c_{ij} > c_{ij(h)}}} (c_{ij} - c_{ij(h)})x_{ij} \quad \forall h \in [n^2], k \in \Delta_-^\lambda \end{aligned} \quad (4c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\lambda \quad (4d)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^\lambda \quad (4e)$$

$$\varphi_k \geq 0 \quad \forall k \in \Delta_-^\lambda \quad (4f)$$

$$(x, y) \in X_I \quad (4g)$$

Here, $ij(h)$ denotes the index-pair ij whose cost c_{ij} is in position h of the lexicographically ordered list of all n^2 cost entries, with ties broken first by index i and then by index j . In practice, constraints (4c) are generated and added through callbacks as needed ([Ljubić et al., 2024](#)). DOMP_{θ_B} is the first benchmark model in our numerical experiments.

In Section 3 of this work, we explain how the decomposition according to Δ_+^λ and Δ_-^λ is generalized by the DC decomposition of f_λ , which allows for the construction of many additional formulations.

2.2.2. Formulations based on radius variables

Typically, radius variables indicate how many clients can be covered within a given cost radius. An advantage of this approach (that also the k -sum models share) is that it is not necessary to explicitly assign clients to ranks in the ordered cost vector. Moreover, radius variables are only defined for the possible different allocation costs, so it becomes beneficial to have repeated entries in the cost matrix. The following well-known radius model DOMP_r dates back to the works of [Puerto \(2008\)](#) and [Marín et al. \(2009\)](#).

Let $0 < c_{(1)} < \dots < c_{(g)} = \max_{i,j \in [n]} c_{ij}$ be the ordered sequence of distinct positive entries in the allocation cost matrix c . Moreover, define $c_{(0)} := 0$. The model relies on binary radius variables r_{lh} indicating whether the allocation cost in position l of the ordered cost vector is greater than or equal to $c_{(h)}$.

$$\text{DOMP}_r : \min \sum_{h=1}^g \sum_{l=1}^n \lambda_l r_{lh} (c_{(h)} - c_{(h-1)}) \quad (5a)$$

$$\text{s.t.} \quad \sum_{l=1}^n r_{lh} = \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij} \quad \forall h \in [g] \quad (5b)$$

$$r_{lh} \leq r_{l+1h} \quad \forall l \in [n-1], h \in [g] \quad (5c)$$

$$r_{lh} \geq r_{lh+1} \quad \forall l \in [n], h \in [g-1] \quad (5d)$$

³ For $x \in \mathbb{R}^d$ and $k \in [d]$, we use the notation $\Delta_k^x := x_k - x_{k-1}$ with $x_0 := 0$ throughout the rest of this paper.

$$r_{lh} \in \{0, 1\} \quad \forall l \in [n], h \in [g] \quad (5e)$$

$$(x, y) \in X_I \quad (5f)$$

Constraints (5b) properly relates the variables r_{lh} with the assignment variables x_{ij} according to the definition of the radius variables. The sorting of the radius variables is enforced by constraints (5c). Lastly, constraints (5d) are technically redundant, but useful valid inequalities that can improve the solver's performance (Marín et al., 2009; Labbé et al., 2017). Since DOMP_r is one of the most competitive compact DOMP formulations (see e.g., Labbé et al., 2017; Ljubić et al., 2024), we chose it as our second benchmark model for the numerical experiments.

3. DC-based MILP models for the DOMP

We now turn to the main part of this work, namely our new modeling approach that makes use of the fact that every ordered median function can be written as the difference of two convex functions. This result was constructively proved by Drezner and Nickel (2009).

Lemma 2 (Drezner and Nickel, 2009, Th. 3). *Let $\lambda \in \mathbb{R}^n$ and $f_\lambda : \mathbb{R}^n \rightarrow \mathbb{R}$ be the corresponding ordered median function. Then, there exist $\alpha, \beta \in \mathbb{R}^n$ with associated ordered median functions f_α, f_β and $\alpha_1 \leq \dots \leq \alpha_n, \beta_1 \leq \dots \leq \beta_n$, such that $f_\lambda = f_\alpha - f_\beta$.*

Due to another result by Grzybowski et al. (2011, Th. 3.6), it is well-known that ordered median functions are convex if and only if the associated objective vector is sorted non-decreasingly. This is the case for α and β in the above lemma, so f_α and f_β are indeed convex functions that can be obtained by Algorithm 1.

Algorithm 1: DC decomposition for ordered median functions (Drezner and Nickel, 2009)

Input : $\lambda \in \mathbb{R}^n$

Output: non-decreasing vectors $\alpha, \beta \in \mathbb{R}_{\geq 0}^n$ such that $f_\lambda = f_\alpha - f_\beta$

```

1  $\alpha_1 \leftarrow \max\{0, \lambda_1\}$ 
2  $\beta_1 \leftarrow \max\{0, -\lambda_1\}$ 
3 for  $k \leftarrow 2$  to  $n$  do
4    $\alpha_k \leftarrow \max\{\alpha_{k-1}, \lambda_k + \beta_{k-1}\}$ 
5    $\beta_k \leftarrow \max\{\beta_{k-1}, -\lambda_k + \alpha_{k-1}\}$ 
6 return  $\alpha, \beta$ 

```

Example 3. Table 1 provides the vectors α and β output by Algorithm 1 for selected λ vectors.

To visualize the DC decomposition in three dimensions, we fix the third input component x_3 to zero and plot the functions f_λ, f_α , and $f_{-\beta}$ for $x \in [-2, 2] \times [-2, 2] \times \{0\}$. The resulting plots are shown in Fig. 1.

Note that the DC decomposition given by (α, β) is not unique. In fact, any constant shift $\delta \in \mathbb{R}$ yields an alternative valid DC decomposition (α', β') where $\alpha'_i = \alpha_i + \delta$ and $\beta'_i = \beta_i + \delta$ for all $i \in [n]$. However, as shown by Drezner and Nickel (2009), the vectors returned by Algorithm 1 possess desirable properties compared to shifted ones. First, they are non-negative, which immediately follows from the initialization of α_1 and β_1 (to zero or $|\lambda_1|$) together with the fact that α and β are non-decreasing. Any negative shift ($\delta < 0$) destroys this property. Second, they are componentwise minimal, that is, for any two non-decreasing vectors $\tilde{\alpha}, \tilde{\beta} \in \mathbb{R}_{\geq 0}^n$ with $f_\lambda = f_{\tilde{\alpha}} - f_{\tilde{\beta}}$ and $i \in [n]$, we have $\alpha_i \leq \tilde{\alpha}_i$ and $\beta_i \leq \tilde{\beta}_i$. Trivially, shifting by $\delta > 0$ destroys minimality.

Since it is always possible to find a DC decomposition with non-negative vectors (e.g., by applying Algorithm 1), we will assume $\alpha, \beta \geq 0$ throughout the remainder of this paper. As will be seen later, this assumption benefits the modeling and facilitates the derivation of several results.

Table 1

Some λ -vectors and the corresponding output α, β of Algorithm 1.

λ	α	β
(1, 2, 3)	(1, 2, 3)	(0, 0, 0)
(3, 2, 1)	(3, 3, 3)	(0, 1, 2)
(-1, 0, 1)	(0, 1, 2)	(1, 1, 1)
(0, 1, 0)	(0, 1, 1)	(0, 0, 1)
(-1, 1, 0)	(0, 2, 2)	(1, 1, 2)
(0, -1, 0, 0, 1, 0)	(0, 0, 1, 1, 2, 2)	(0, 1, 1, 1, 1, 2)
(0, 0, 1, 0, 0, 1)	(0, 0, 1, 1, 1, 2)	(0, 0, 0, 1, 1, 1)

Furthermore, it directly follows from Definition 1 that $-f_\beta = f_{-\beta}$, hence, $f_\lambda = f_\alpha - f_\beta = f_\alpha + f_{-\beta}$.

The central idea of exploiting the DC representation $f_\lambda = f_\alpha + f_{-\beta}$ is to combine DOMP models for f_α with DOMP models for $f_{-\beta}$. Validity of such combined models is justified by the following simple result.

Lemma 4. *Let $X_1 \subseteq \mathbb{R}^{n_1}$, $X_2 \subseteq \mathbb{R}^{n_1+n_2}$, $X'_2 \subseteq \mathbb{R}^{n_1+n'_2}$, $n_1, n_2, n'_2 \in \mathbb{Z}_{\geq 1}$. Let $G : X_1 \times X_2 \rightarrow \mathbb{R}$ and $H : X_1 \times X'_2 \rightarrow \mathbb{R}$. Define for every $x_1 \in X_1$ the functions $g(x_1) := \min_{(x_1, x_2) \in X_2} G(x_1, x_2)$, $h(x_1) := \min_{(x_1, x'_2) \in X'_2} H(x_1, x'_2)$ and assume the minima are always attained. Further, let $f : X_1 \rightarrow \mathbb{R}$ such that $f(x_1) = g(x_1) + h(x_1)$ for every $x_1 \in X_1$. Then,*

$$\min_{x_1 \in X_1} f(x_1) = \min_{\substack{x_1 \in X_1 \\ (x_1, x_2) \in X_2, (x_1, x'_2) \in X'_2}} G(x_1, x_2) + H(x_1, x'_2).$$

Proof. Define for $x_1 \in X_1$ the function $F(x_1) := \min_{(x_1, x_2) \in X_2, (x_1, x'_2) \in X'_2} G(x_1, x_2) + H(x_1, x'_2)$. It suffices to show that $f = F$ on X_1 as this immediately implies $\min_{x_1 \in X_1} f(x_1) = \min_{x_1 \in X_1} F(x_1)$.

Fix an arbitrary $x_1 \in X_1$. For every $(x_1, x_2) \in X_2$, we have by definition $g(x_1) \leq G(x_1, x_2)$ and similarly for $(x_1, x'_2) \in X'_2$, it holds $h(x_1) \leq H(x_1, x'_2)$. Therefore, $f(x_1) = g(x_1) + h(x_1) \leq F(x_1)$.

Conversely, take minimizer $(x_1, \bar{x}_2) \in X_2$, $(x_1, \bar{x}'_2) \in X'_2$ with $g(x_1) = G(x_1, \bar{x}_2)$ and $h(x_1) = H(x_1, \bar{x}'_2)$, respectively. By definition of F , we obtain $F(x_1) \leq g(x_1) + h(x_1) = f(x_1)$.

Since x_1 was arbitrary, it follows that $f(x_1) = F(x_1)$ for every $x_1 \in X_1$, as required. $\square$

So, according to Lemma 4, we can obtain a valid DOMP minimization model for f_λ by simply combining two valid DOMP minimization models, one for f_α , and one for $f_{-\beta}$. Particularly, we need to add the two objective functions together, use a common feasible set X_1 from which the costs are derived (in our case, this is the p -median polytope X_I), and include all remaining constraints (defining X_2 and X'_2) from the two models. Note that introducing additional constraints that couple X_2 and X'_2 may invalidate the combined model, as Lemma 4 relies on the separability of these two sets. Such coupling would, in general, prevent the existence of the minimizers $(x_1, \bar{x}_2)$ and $(x_1, \bar{x}'_2)$ used in the second part of the proof.

The advantage of a combined model is that f_α and $f_{-\beta}$ generally have nicer structure than f_λ . In particular, f_α is convex and $f_{-\beta}$ is concave, properties that are very useful in modeling the DOMP. For instance, the DOMP with a convex objective can be efficiently formulated by exploiting the LP model (2) from k -sum optimization, which requires relatively few variables and constraints to correctly evaluate the sorted and weighted objective. Conversely, many DOMP formulations reduce to a simpler form if the ordered median objective is concave (see e.g., model (11) discussed later in this study), as certain ordering constraints naturally become redundant in this case.

We remark that decomposing the DOMP into convex and concave parts generalizes the decomposition used for the k -sum model in Section 2.2.1, which is based on Δ_+^k and Δ_-^k . Indeed, for $\lambda \in \mathbb{R}^n$ there is always a DC decomposition (α, β) such that $(\Delta_+^\alpha, \Delta_-^\beta) = (\Delta_+^\lambda, \Delta_-^\lambda)$. In that case, for any vector $d \in \mathbb{R}^n$, it holds that

$$f_\alpha(d) = \sum_{k \in \Delta_+^\alpha} \Delta_k^\alpha S_k(d) = \sum_{k \in \Delta_+^\lambda} \Delta_k^\lambda S_k(d) \quad \text{and}$$

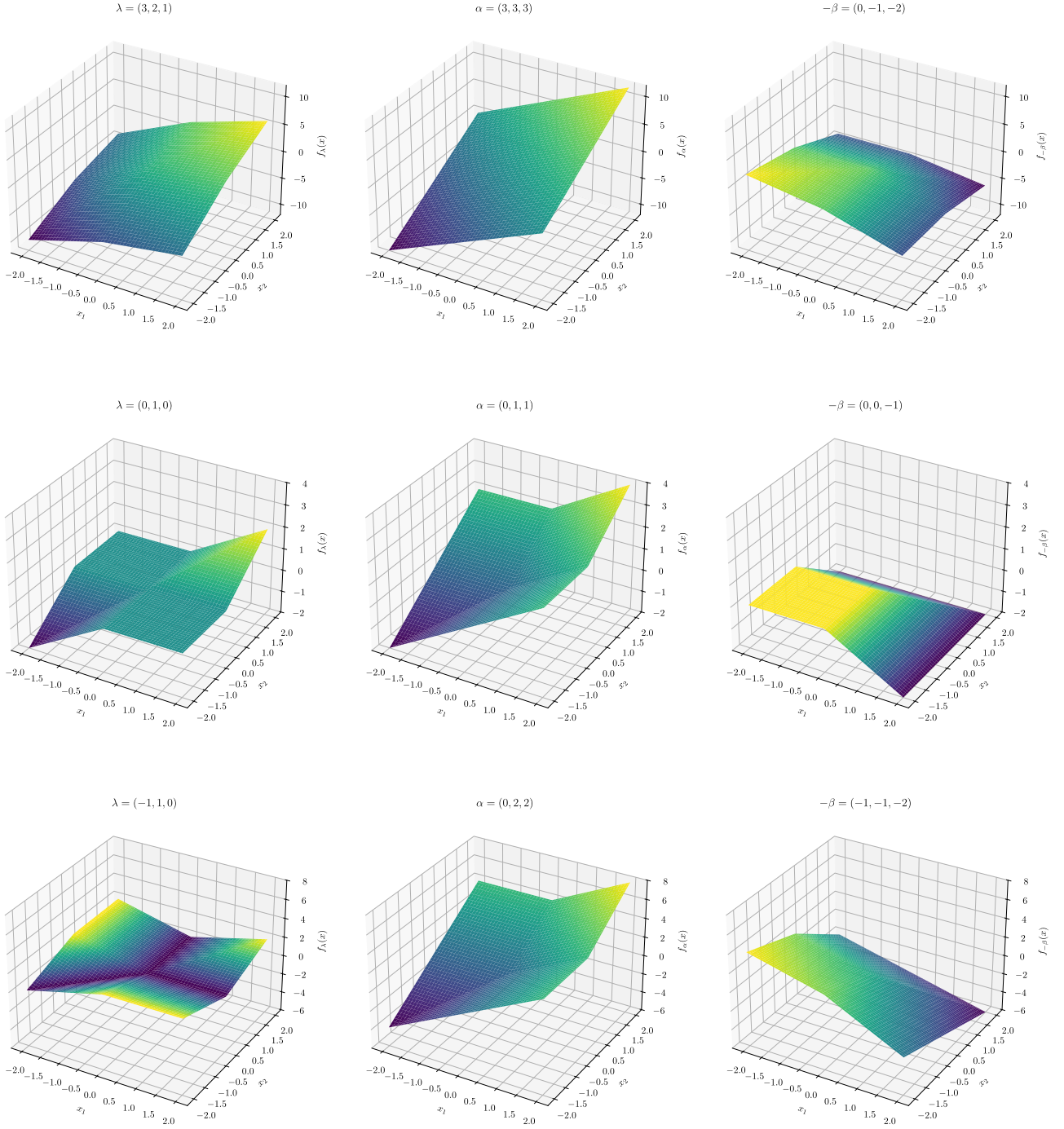


Fig. 1. Visualization of ordered median functions and their respective DC decomposition given by Algorithm 1 for $\lambda = (3, 2, 1)$, $\lambda = (0, 1, 0)$, and $\lambda = (-1, 1, 0)$. The third component x_3 is fixed to 0 in all plots.

$$f_{-\beta}(d) = \sum_{k \in \Delta_{-}^{-\beta}} \Delta_k^{-\beta} S_k(d) = \sum_{k \in \Delta_{-}^{\lambda}} \Delta_k^{\lambda} S_k(d).$$

However, representations of f_{α} and $f_{-\beta}$ are not limited to k -sum formulations, and alternative constructions are possible.

In the following subsections, we present several promising examples derived within our new framework. All of them assume the availability of a DC decomposition $f_{\lambda} = f_{\alpha} + f_{-\beta}$ with $\alpha, \beta \geq 0$. In practice, we compute α and β using Algorithm 1. Moreover, all models presented in Sections 3.1 and 3.2 use the k -sum based model DOMP_{θ} for f_{α} and differ in the modeling approach for $f_{-\beta}$. We note that DOMP_{θ} simplifies drastically for f_{α} since α is non-negative and non-decreasing, and therefore $\Delta_{-}^{\alpha} = \emptyset$. As a result, all θ -variables and constraints

associated with Δ_{-}^{α} can be dropped, leaving only (3a), (3b), (3e), (3f), and (3h). In Section 3.3, we briefly comment on alternative ways of representing f_{α} .

3.1. Compact models

3.1.1. Radius model for $f_{-\beta}$

The first idea combines the k -sum based model DOMP_{θ} with the radius model DOMP_r .

Ljubić et al. (2024) proved that for non-increasing λ , there always exists an optimal integral solution to model (5) if constraints (5e) are

relaxed. Moreover, they argued that, in this particular setting, inequalities (5c) and (5d) are redundant. This leaves us with the following DOMP formulation that is valid for any non-increasing λ .

$$\min \sum_{h=1}^g \sum_{l=1}^n \lambda_l r_{lh} (c_{(h)} - c_{(h-1)}) \quad (6a)$$

$$\text{s.t.} \quad \sum_{l=1}^n r_{lh} = \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij} \quad \forall h \in [g] \quad (6b)$$

$$r_{lh} \in [0, 1] \quad \forall l \in [n], h \in [g] \quad (6c)$$

$$(x, y) \in X_I \quad (6d)$$

Recall that β from the DC decomposition of f_λ is non-decreasing. Therefore, $-\beta$ is non-increasing, so model (6) can be used to determine the value of $f_{-\beta}$. Combining (6) for $f_{-\beta}$ with (3) for f_α according to Lemma 4 results in our first DC-based model $\text{DOMP}_{DC:r}$ which is valid for any general $\lambda \in \mathbb{R}^n$ with DC decomposition $f_\lambda = f_\alpha + f_{-\beta}$.

$$\text{DOMP}_{DC:r} : \min \sum_{k \in \mathcal{D}_+^a} \Delta_k^a \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \sum_{h=1}^g \sum_{l=1}^n -\beta_l r_{lh} (c_{(h)} - c_{(h-1)}) \quad (7a)$$

$$\text{s.t.} \quad t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \mathcal{D}_+^a \quad (7b)$$

$$\sum_{l=1}^n r_{lh} = \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij} \quad \forall h \in [g] \quad (7c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \mathcal{D}_+^a \quad (7d)$$

$$t_k \geq 0 \quad \forall k \in \mathcal{D}_+^a \quad (7e)$$

$$r_{lh} \in [0, 1] \quad \forall l \in [n], h \in [g] \quad (7f)$$

$$(x, y) \in X_I \quad (7g)$$

Note that the combined model $\text{DOMP}_{DC:r}$ reduces to DOMP_θ if $\beta_l = 0 \forall i \in [n]$ and to DOMP_r if $\Delta_k^a = 0 \forall k \in [n]$. Similar simplifications apply for all DC-based DOMP models.

We remark that $\text{DOMP}_{DC:r}$ is very similar to the model (DOMP_{OTr1}) presented in Marín et al. (2020). The key differences are that $\text{DOMP}_{DC:r}$ does not include ordering constraints for the r -variables and additionally relaxes their integrality. These distinctions may explain the poor performance reported for (DOMP_{OTr1}) and motivate a thorough evaluation of $\text{DOMP}_{DC:r}$.

3.1.2. Benders reformulation of radius model for $f_{-\beta}$

Instead of directly using the radius formulation DOMP_r to model $f_{-\beta}$, we will proceed by using the Benders reformulation of DOMP_r developed by Ljubić et al. (2024), which we recall first.

Proposition 5 (Ljubić et al., 2024, Prop. 1). *For any $\lambda \in \mathbb{R}^n$ satisfying $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, the following compact model is a valid DOMP formulation.*

$$\min \sum_{h=1}^g \theta_h (c_{(h)} - c_{(h-1)}) \quad (8a)$$

$$\text{s.t.} \quad \theta_h \geq \lambda_l \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij} - \sum_{l' > l} (\lambda_l - \lambda_{l'}) \quad \forall h \in [g], l \in [n] \quad (8b)$$

$$\theta_h \in \mathbb{R} \quad \forall h \in [g] \quad (8c)$$

$$(x, y) \in X_I \quad (8d)$$

Since $-\beta$ is non-increasing, it satisfies the assumption made in Proposition 5. Using (8) to model $f_{-\beta}$ and combining it with (3) for f_α results in the following valid DOMP formulation.

$$\text{DOMP}_{DC:rB} : \min \sum_{k \in \mathcal{D}_+^a} \Delta_k^a \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \sum_{h=1}^g \theta_h (c_{(h)} - c_{(h-1)}) \quad (9a)$$

$$\text{s.t.} \quad t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \mathcal{D}_+^a \quad (9b)$$

$$\theta_h \geq -\beta_l \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij} - \sum_{l' > l} (\beta_{l'} - \beta_l) \quad \forall h \in [g], l \in [n] \quad (9c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \mathcal{D}_+^a \quad (9d)$$

$$t_k \geq 0 \quad \forall k \in \mathcal{D}_+^a \quad (9e)$$

$$\theta_h \in \mathbb{R} \quad \forall h \in [g] \quad (9f)$$

$$(x, y) \in X_I \quad (9g)$$

There are $\mathcal{O}(n^2)$ variables θ_h to represent $f_{-\beta}$ which is less than the $\mathcal{O}(n^3)$ variables θ_{ij}^k used in DOMP_θ to model this part. However, with $\mathcal{O}(n^3)$ constraints of type (9c), there is a relatively high number of such constraints. Fortunately, the separation procedure proposed by Ljubić et al. (2024) for (DOMP_{r0B}) can be used to add them dynamically whenever needed. In an implementation, the model must be initialized with a valid lower bound on $\theta_h, h \in [g]$ in order for it to be bounded from the beginning of the computations. A close inspection of (9c) yields the following result.

Lemma 6. *Let $h \in [g]$ and define $N_h := |\{i \in [n] \mid \max_{j \in [n]} c_{ij} \geq c_{(h)}\}|$. Then, for model (9), it holds $\theta_h \geq \sum_{l=n-N_h+1}^n -\beta_l$.*

Proof. Observe that $N_h \geq \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij}$ for every feasible x . Using (9c) and the assumption $\beta \geq 0$, we thus obtain for each $h \in [g]$

$$\theta_h \geq -\beta_l N_h - \sum_{l' > l} (\beta_{l'} - \beta_l) = (n-l-N_h)\beta_l - \sum_{l'=l+1}^n \beta_{l'} =: f_h(l) \quad \forall l \in [n]. \quad (10)$$

We now claim that $l^* := n - N_h$ satisfies $f_h(l^*) \geq f_h(l)$ for all $l \in [n]$.

On the one hand, if $l < l^*$, we see that

$$\begin{aligned} f_h(l) - f_h(l^*) &= (n-l-N_h)\beta_l - \sum_{l'=l+1}^n \beta_{l'} + \sum_{l'=l^*+1}^n \beta_{l'} = (l^*-l)\beta_l - \underbrace{\sum_{l'=l+1}^{l^*} \beta_{l'}}_{\leq -(l^*-l)\beta_l} \\ &\leq (l^*-l)\beta_l - (l^*-l)\beta_l = 0. \end{aligned}$$

For the inequality, we used that β is monotonically increasing, so $\beta_{l'} \geq \beta_l$ for $l' \geq l+1$.

On the other hand, if $l > l^*$, it holds

$$\begin{aligned} f_h(l) - f_h(l^*) &= (n-l-N_h)\beta_l - \sum_{l'=l+1}^n \beta_{l'} + \sum_{l'=l^*+1}^n \beta_{l'} = -(l-l^*)\beta_l + \underbrace{\sum_{l'=l^*+1}^l \beta_{l'}}_{\leq (l-l^*)\beta_l} \\ &\leq -(l-l^*)\beta_l + (l-l^*)\beta_l = 0. \end{aligned}$$

From (10), we can therefore deduce $\theta_h \geq f_h(l^*) = -\sum_{l=n-N_h+1}^n \beta_l$. $\square$

3.1.3. Staircase model for $f_{-\beta}$

In their paper, Labbé et al. (2017) propose an alternative DOMP formulation that is similar to the radius formulation DOMP_r but differs slightly in the definition of the used variables. Particularly, the binary variables v_{lh} indicate whether the allocation cost in position l is equal to $c_{(h)}$ or not for each $l \in [n]$ and $h \in [g] \cup \{0\}$. For convenience, define $[g]_0 := [g] \cup \{0\}$.

$$\text{DOMP}_s : \min \sum_{h=0}^g \sum_{l=1}^n \lambda_l c_{(h)} v_{lh} \quad (11a)$$

$$\text{s.t.} \quad \sum_{h=0}^g v_{lh} = 1 \quad \forall l \in [n] \quad (11b)$$

$$\sum_{l=1}^n v_{lh} = \sum_{\substack{i,j \in [n] \\ c_{ij} = c_{(h)}}} x_{ij} \quad \forall h \in [g]_0 \quad (11c)$$

$$\sum_{h' < h} v_{l+1,h'} + \sum_{h' \geq h} v_{l,h'} \leq 1 \quad \forall l \in [n-1], h \in [g] \quad (11d)$$

$$v_{lh} \in \{0, 1\} \quad \forall l \in [n], h \in [g]_0 \quad (11e)$$

$$(x, y) \in X_I \quad (11f)$$

Just like before, we intend to make use of this for the non-increasing $-\beta$ objective vector. In that case, model (11) can be simplified. The following has been proved recently by Nickel et al. (2026b).

Lemma 7 (Nickel et al. (2026b, Lem. 3.2)). *For any $\lambda \in \mathbb{R}^n$ satisfying $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, the following compact model is a valid DOMP formulation.*

$$\min \sum_{h=0}^g \sum_{l=1}^n \lambda_l c_{(h)} v_{lh} \quad (12a)$$

$$\text{s.t.} \quad \sum_{h=0}^g v_{lh} = 1 \quad \forall l \in [n] \quad (12b)$$

$$\sum_{l=1}^n v_{lh} = \sum_{\substack{i,j \in [n] \\ c_{ij} = c_{(h)}}} x_{ij} \quad \forall h \in [g]_0 \quad (12c)$$

$$v_{lh} \geq 0 \quad \forall l \in [n], h \in [g]_0 \quad (12d)$$

$$(x, y) \in X_I \quad (12e)$$

Analogously to the previous DC-based DOMP models, we can now put together a combined model using (3) for f_α and (12) for $f_{-\beta}$ to obtain the next valid DOMP formulation $\text{DOMP}_{DC:s}$.

$$\text{DOMP}_{DC:s} : \min \sum_{k \in \Delta_+^\alpha} A_k^\alpha \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \sum_{h=0}^g \sum_{l=1}^n -\beta_l c_{(h)} v_{lh} \quad (13a)$$

$$\text{s.t.} \quad t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (13b)$$

$$\sum_{h=0}^g v_{lh} = 1 \quad \forall l \in [n] \quad (13c)$$

$$\sum_{l=1}^n v_{lh} = \sum_{\substack{i,j \in [n] \\ c_{ij} = c_{(h)}}} x_{ij} \quad \forall h \in [g]_0 \quad (13d)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (13e)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^\alpha \quad (13f)$$

$$v_{lh} \geq 0 \quad \forall l \in [n], h \in [g]_0 \quad (13g)$$

$$(x, y) \in X_I \quad (13h)$$

3.1.4. Decomposition of $f_{-\beta}$

Similar to decomposing $f_\lambda = f_\alpha + f_{-\beta}$ into its DC components, one can further decompose $f_{-\beta}$ into simpler functions. Recall that $-\beta$ is a non-increasingly sorted vector. Moreover, since $\beta \geq 0$, we have $0 \geq -\beta_1 \geq \dots \geq -\beta_n$. This structure allows us to express $-\beta$ using d vectors of the form

$$\gamma_r = a_r(0, \dots, 0, \underbrace{-1, \dots, -1}_{k_r}) \in \mathbb{R}^n, \quad a_r \geq 0, k_r \geq 1,$$

so that $-\beta = \sum_{r=1}^d \gamma_r$.

Example 8. Let $-\beta = (0, -1, -3, -3, -5)$. Then, with

$$\gamma_1 = 1(0, -1, -1, -1, -1),$$

$$\gamma_2 = 2(0, 0, -1, -1, -1),$$

$$\gamma_3 = 2(0, 0, 0, 0, -1),$$

we obtain $-\beta = \gamma_1 + \gamma_2 + \gamma_3$. Note that for every decrement in $-\beta$, there is exactly one γ_r .

According to Lemma 4, we can now piece together DOMP formulations for objective vectors of the type γ_r to obtain DOMP models for $f_{-\beta}$ and, ultimately, for general $f_\lambda = f_\alpha + \sum_{r=1}^d f_{\gamma_r}$.

Benders reformulation of radius model for decomposed $f_{-\beta}$

To make use of this refined decomposition, we again use the model from Proposition 5, but now applied to each f_{γ_r} (see $\text{DOMP}_{DC:\beta B}$).

$$\text{DOMP}_{DC:\beta B} : \min \sum_{k \in \Delta_+^\alpha} A_k^\alpha \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \sum_{r=1}^d \sum_{h=1}^g \theta_{rh} (c_{(h)} - c_{(h-1)}) \quad (14a)$$

$$\text{s.t.} \quad t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (14b)$$

$$\theta_{rh} \geq (\gamma_r)_l \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij} - \sum_{l' > l} ((\gamma_r)_l - (\gamma_r)_{l'}) \quad \forall r \in [d], h \in [g], l \in [n] \quad (14c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (14d)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^\alpha \quad (14e)$$

$$\theta_{rh} \in \mathbb{R} \quad \forall r \in [d], h \in [g] \quad (14f)$$

$$(x, y) \in X_I \quad (14g)$$

This model has a higher number of variables and constraints compared to model (9), since we need to apply the Benders model d times. However, constraints (14c) reduce to a slightly simpler form. In particular, for every $r \in [d]$, we have

$$\sum_{l' > l} ((\gamma_r)_l - (\gamma_r)_{l'}) = \begin{cases} k_r a_r, & \text{if } (\gamma_r)_l = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, constraints (14c) reduce to

$$\theta_{rh} \geq \begin{cases} -k_r a_r, & \text{if } (\gamma_r)_l = 0, \\ -a_r \sum_{\substack{i,j \in [n] \\ c_{ij} \geq c_{(h)}}} x_{ij}, & \text{otherwise.} \end{cases}$$

The separation of these constraints can be done in the same fashion as proposed by Ljubić et al. (2024), but the violation checks must be done for each $r \in [d]$. Note that the general lower bounds from Lemma 6 specialize to $\theta_{rh} \geq -a_r \min\{N_h, k_r\}$ for all $r \in [d]$, $h \in [g]$.

Notice that for both the radius formulation (6) and the staircase formulation (12), objective vectors of the form γ_r do not lead to obvious simplifications, similar to those of model (8) described above. Consequently, the refined decomposition of $f_{-\beta}$ is unlikely to yield improved models compared to $\text{DOMP}_{DC:r}$ and $\text{DOMP}_{DC:s}$, respectively, as it increases model size. We therefore did not further consider these formulations in combination with the decomposition of $f_{-\beta}$.

3.2. Exponential-sized models

3.2.1. Benders reformulation of staircase model for $f_{-\beta}$

By applying Benders decomposition to the model in Lemma 7, one can obtain the following result (Nickel et al., 2026b).

Theorem 9 (Nickel et al. (2026b, Th. 3.3)). *For any $\lambda \in \mathbb{R}^n$ satisfying $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, the following exponential-sized model is a valid DOMP formulation.*

$$\min \quad \theta \quad (15a)$$

$$\text{s.t.} \quad \theta \geq \sum_{k=1}^n A_k^\lambda \left((n-k+1)c_{(f(k-1))} + \sum_{\substack{i,j \in [n] \\ c_{ij} > c_{(f(k-1))}}} (c_{ij} - c_{(f(k-1))})x_{ij} \right) \quad \forall f : [n]_0 \rightarrow [g]_0 \text{ mon. incr.} \\ \text{with } f(0) = 0 \quad (15b)$$

$$\theta \in \mathbb{R} \quad (15c)$$

$$(x, y) \in X_I \quad (15d)$$

Using model (15) for $f_{-\beta}$ and combining it with the k -sum model DOMP_θ for f_α , we obtain the following DOMP formulation that is valid for any objective vector λ .

$$\text{DOMP}_{DC:sB} : \min \sum_{k \in \Delta_+^\alpha} \Delta_k^\alpha \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \theta \quad (16a)$$

$$\text{s.t. } t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (16b)$$

$$\theta \geq \sum_{k=1}^n \Delta_k^{-\beta} \left((n-k+1)c_{(f(k-1))} + \sum_{\substack{i,j \in [n] \\ c_{ij} > c_{(f(k-1))}}} (c_{ij} - c_{(f(k-1))})x_{ij} \right) \quad \forall f : [n]_0 \rightarrow [g]_0 \text{ mon. incr. with } f(0) = 0 \quad (16c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (16d)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^\alpha \quad (16e)$$

$$\theta \in \mathbb{R} \quad (16f)$$

$$(x, y) \in X_I \quad (16g)$$

Separation of the constraints (16b) can be performed in linear time by solving a specially structured transportation problem, as explained by Nickel et al. (2026b).

A general lower bound on θ is given by the lowest possible value for $f_{-\beta}$. Since $-\beta_i \leq 0$ for $i \in [n]$, the largest possible sorted allocation cost vector will yield the lowest possible value for $f_{-\beta}$. In particular, it always holds $\theta \geq \sum_{i=1}^n -\beta_i c_{(i)}^{\max}$ where $c_i^{\max} = \max\{c_{ij} \mid j \in [n]\}$ for $i \in [n]$.

We note that model (16) has a similar structure to model $\text{DOMP}_{\theta 1a}$ presented by Ljubić et al. (2024). The latter is an aggregated reformulation of the state-of-the-art model $\text{DOMP}_{\theta B}$. Due to the similar performance reported for $\text{DOMP}_{\theta 1a}$ and $\text{DOMP}_{\theta B}$ (Ljubić et al., 2024), one may also expect $\text{DOMP}_{DC:sB}$ to perform comparably to the benchmark in the numerical experiments.

Furthermore, notice that another formulation equivalent to $\text{DOMP}_{DC:sB}$ can be obtained by replacing (15) with the second DOMP model for non-increasing λ proposed by Nickel et al. (2026b, Th. 3.4).

3.2.2. Big-M formulations

Let $c(x)$ denote the vector of allocation costs, that is, $c_i(x) := \sum_{j=1}^n c_{ij} x_{ij}$ for each $i \in [n]$. Then, each of the previously shown DC-based DOMP models is a linearization of the following concave minimization program.

$$\min \sum_{k \in \Delta_+^\alpha} \Delta_k^\alpha \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + f_{-\beta}(c(x)) \quad (17a)$$

$$\text{s.t. } t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (17b)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (17c)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^\alpha \quad (17d)$$

$$(x, y) \in X_I \quad (17e)$$

An equivalent reformulation of problem (17) can be obtained by introducing a suitably lower-bounded variable θ .

$$\min \sum_{k \in \Delta_+^\alpha} \Delta_k^\alpha \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \theta \quad (18a)$$

$$\text{s.t. } \theta \geq f_{-\beta}(c(x)) \quad (18b)$$

$$t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (18c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (18d)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^\alpha \quad (18e)$$

$$\theta \in \mathbb{R} \quad (18f)$$

$$(x, y) \in X_I \quad (18g)$$

Note that model (18) resembles $\text{DOMP}_{DC:sB}$ from Section 3.2.1. Particularly, constraints (16c) linearize the reverse convex constraint (18b). In the remainder of this subsection, we present two further ways of formulating constraint (18b) by means of linear inequalities. Both ideas result in exponential-sized big-M models for DOMP that can be solved through lazy cut generation during the branch-and-bound procedure.

No-good cut formulations

The easiest way to correctly enforce constraint (18b) is by means of no-good cuts that lower bound the value of θ for every possible solution. More precisely, let $(\bar{x}, \bar{y}) \in X_I$ be given (at some node of the branch-and-bound tree). It is easy to compute the value $\bar{\theta} := f_{-\beta}(c(\bar{x}))$ by simply sorting the fixed vector of allocation costs and computing the scalar product with $-\beta$. Further, let $\underline{\theta}$ be any valid lower bound on θ (in our implementation, we use $\underline{\theta} = \sum_{i=1}^n -\beta_i c_{(i)}^{\max}$ as proposed in Section 3.2.1). Then, we obtain the following no-good cut.

$$\theta \geq \bar{\theta} - (\bar{\theta} - \underline{\theta}) \left(\sum_{\substack{i,j \in [n] \\ \bar{x}_{ij}=1}} (1 - x_{ij}) + \sum_{\substack{i,j \in [n] \\ \bar{x}_{ij}=0}} x_{ij} \right) \quad (19)$$

A similar type of no-good cut based on the same idea can be considered for the y -variables.

$$\theta \geq \bar{\theta} - (\bar{\theta} - \underline{\theta}) \left(\sum_{\substack{j \in [n] \\ \bar{y}_j=1}} (1 - y_j) + \sum_{\substack{j \in [n] \\ \bar{y}_j=0}} y_j \right) \quad (20)$$

Adding these at every integer solution found by the branch-and-bound algorithm will obviously force the correct behavior. In order to strengthen the no-good cuts presented above, one may remove the part where $\bar{x}_{ij} = 0$ and $\bar{y}_j = 0$, respectively.

$$\theta \geq \bar{\theta} - (\bar{\theta} - \underline{\theta}) \sum_{\substack{i,j \in [n] \\ \bar{x}_{ij}=1}} (1 - x_{ij}) \quad (21)$$

$$\theta \geq \bar{\theta} - (\bar{\theta} - \underline{\theta}) \sum_{\substack{j \in [n] \\ \bar{y}_j=1}} (1 - y_j) \quad (22)$$

Validity of (21) is justified because the assignment constraints of the p -median polytope require $\sum_{j=1}^n x_{ij} = 1$ for all $i \in [n]$. Hence, any integer assignment $x \neq \bar{x}$ must differ in at least one variable where $x_{ij} = 0 \neq 1 = \bar{x}_{ij}$, ensuring $\sum_{\bar{x}_{ij}=1} (1 - x_{ij}) \geq 1$. Moreover, it holds for any $(x, y) \in X$ that

$$\bar{\theta} - (\bar{\theta} - \underline{\theta}) \sum_{\substack{i,j \in [n] \\ \bar{x}_{ij}=1}} (1 - x_{ij}) \geq \bar{\theta} - (\bar{\theta} - \underline{\theta}) \left(\sum_{\substack{i,j \in [n] \\ \bar{x}_{ij}=1}} (1 - x_{ij}) + \sum_{\substack{i,j \in [n] \\ \bar{x}_{ij}=0}} x_{ij} \right),$$

since we have $\bar{\theta} - \underline{\theta} \geq 0$ and $\sum_{\substack{i,j \in [n] \\ \bar{x}_{ij}=0}} x_{ij} \geq 0$. So, (21) indeed dominates (19). With similar arguments, one can verify that (22) is valid and stronger than (20).

Enforcing constraint (18b) using no-good cuts results in the following valid DOMP model.

$$\text{DOMP}_{DC:ngc} : \min \sum_{k \in \Delta_+^\alpha} \Delta_k^\alpha \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \theta \quad (23a)$$

$$\text{s.t. } \text{Eq. (21), Eq. (22)} \quad \forall (\bar{x}, \bar{y}) \in X_I \quad (23b)$$

$$t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (23c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^\alpha \quad (23c)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^a \quad (23d)$$

$$\theta \in \mathbb{R} \quad (23e)$$

$$(x, y) \in X_I \quad (23f)$$

In our preliminary numerical experiments, we consider three different variants of the no-good cut model. In the first, $\text{DOMP}_{DC:ngc}$, both (21) and (22) are added during separation. In contrast, in variant $\text{DOMP}_{DC:ngc_x}$, only cuts of type (21) are added and, similarly, in variant $\text{DOMP}_{DC:ngc_y}$, we only apply cuts of type (22). The results of the comparison of these three models are reported in Section 4.1.

Exponential formulation for decomposed $f_{-\beta}$

As in Section 3.1.4, we decompose $f_{-\beta}(c(x)) = \sum_{r=1}^d f_{\gamma_r}(c(x))$. So, in model (18), we replace θ with $\sum_{r=1}^d \bar{\theta}_r$ and constraint (18b) with d constraints of the form $\bar{\theta}_r \geq f_{\gamma_r}(c(x))$. The value of $f_{\gamma_r}(c(x))$ is obtained by summing up the k_r largest cost values in $c(x)$ and multiplying with $-a_r$. In other words, we seek to compute the k -sum $S_{n-k_r+1}(c(x))$ to find $f_{\gamma_r}(c(x)) = -a_r S_{n-k_r+1}(c(x))$. If θ_r ($r \in [d]$) represents the value of $S_{n-k_r+1}(c(x))$, we have terms $\bar{\theta}_r = -a_r \theta_r$ in the objective and need constraints of the form $\theta_r \leq S_{n-k_r+1}(c(x))$.

For a given subset of opened facilities $J \subseteq [n]$, $|J| = p$, let $j_i \in \{j^* \mid c_{ij^*} \leq c_{ij} \forall j \in J\}$ be a closest open facility of client i . Further, we can sort the clients $i \in [n]$ according to the allocation costs from their closest open facilities. In particular, assume that $c_{i_1 j_{i_1}} \geq \dots \geq c_{i_n j_{i_n}}$. Then, we have $S_{n-k_r+1}(c(x)) = \sum_{j \in J} \sum_{l=1}^{k_r} c_{i_l j_{i_l}} x_{i_l j_{i_l}}$. This logic can be modeled using big-M constraints leading to formulation $\text{DOMP}_{DC:\beta e}$.

$$\text{DOMP}_{DC:\beta e} : \min \sum_{k \in \Delta_+^a} \Delta_k^a \left((n-k+1)t_k + \sum_{i=1}^n z_{ik} \right) + \sum_{r=1}^d -a_r \theta_r \quad (24a)$$

$$\text{s.t.} \quad \theta_r \leq \sum_{i=1}^{k_r} c_{i_l j_{i_l}} x_{i_l j_{i_l}} + M(p - \sum_{j \in J} y_j) \quad \forall r \in [d], J \subseteq [n] : |J| = p \quad (24b)$$

$$t_k + z_{ik} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall i \in [n], k \in \Delta_+^a \quad (24c)$$

$$z_{ik} \geq 0 \quad \forall i \in [n], k \in \Delta_+^a \quad (24d)$$

$$t_k \geq 0 \quad \forall k \in \Delta_+^a \quad (24e)$$

$$\theta_r \in \mathbb{R} \quad \forall r \in [d] \quad (24f)$$

$$(x, y) \in X_I \quad (24g)$$

The big-M constant in this formulation can be chosen individually for each $r \in [d]$ as $M_r = \sum_{i=n-k_r+1}^n c_{(i)}^{\max}$ where $c_{(i)}^{\max} = \max\{c_{ij} \mid j \in [n]\}$ for $i \in [n]$. Further, we have $\theta_r \leq M_r$ for all $r \in [d]$, so the big-M constants can also be used to bound the model.

A straightforward separation algorithm for constraints (24b) works as follows. Given a solution $(\bar{x}, \bar{y}, \bar{\theta})$, let $J := \{j \in [n] : \bar{y}_j = 1\}$ and sort the clients according to the allocation costs of their closest open facilities, starting with the highest. For each $r \in [d]$, check if $\bar{\theta}_r > \sum_{l=1}^{k_r} c_{i_l j_{i_l}} \bar{x}_{i_l j_{i_l}}$. If this is the case, add the constraint $\theta_r \leq \sum_{l=1}^{k_r} c_{i_l j_{i_l}} x_{i_l j_{i_l}} + M_r(p - \sum_{j \in J} y_j)$ to the problem.

3.3. Alternative representation of f_α

Recall that the DC-based models each combine two DOMP formulations, one for a non-decreasing objective vector $\alpha \geq 0$, and another for a non-increasing objective vector $-\beta \leq 0$. In this study, we focus on different ways of modeling the $-\beta$ -part. However, to enrich the discussion of the DC-approach, we briefly comment on some alternative ways of representing f_α in this section. The suggested ideas were not tested extensively and are therefore excluded from the numerical experiments in Section 4.

3.3.1. Compact alternatives

The k -sum model exploits the convexity of f_α , but does not benefit from repetitions in the cost matrix like DOMP_r or DOMP_s . It is possible to replace the k -sum model with one of these models to deal with f_α . However, as both formulations do not reduce to a simpler form for non-decreasing objective vectors, it is likely that a combined model does not outperform the plain versions of DOMP_r and DOMP_s , which are valid for arbitrary λ .

A more promising compact alternative is the formulation (25) below, given by Marín et al. (2020, model MDOMP_{BHP}), which is derived from the dual of an assignment formulation for the DOMP.

$$\min \sum_{k \in \Delta_+^a} \Delta_k^a \left(\sum_{l=k}^n u_{kl} + \sum_{i=1}^n v_{ki} \right) \quad (25a)$$

$$\text{s.t.} \quad u_{kl} + v_{ki} \geq \sum_{j=1}^n c_{ij} x_{ij} \quad \forall k \in \Delta_+^a, i \in [n], l \in [n] : l \geq k \quad (25b)$$

$$u_{kl} \geq 0 \quad \forall k \in \Delta_+^a, l \in [n] : l \geq k \quad (25c)$$

$$v_{ki} \geq 0 \quad \forall k \in \Delta_+^a, i \in [n] \quad (25d)$$

$$(x, y) \in X_I \quad (25e)$$

3.3.2. Exponential alternatives

Instead of the mentioned compact models, it could be worthwhile to consider an exponential-sized formulation for f_α . It is shown by Ljubić et al. (2024, Prop. 3) that the following is a valid DOMP model for non-decreasing $\alpha \geq 0$.

$$\min \sum_{k \in \Delta_+^a} \Delta_k^a \varphi_k \quad (26a)$$

$$\text{s.t.} \quad \varphi_k \geq \sum_{i \in N_L} \sum_{j \in N} c_{ij} x_{ij} \quad \forall k \in \Delta_+^a, N_L \subseteq [n] : |N_L| = n - k + 1 \quad (26b)$$

$$\varphi_k \in \mathbb{R} \quad \forall k \in \Delta_+^a \quad (26c)$$

$$(x, y) \in X_I \quad (26d)$$

Note that the sum of any $n - k + 1$ client allocation costs provides a lower bound on $S_k(c(x))$, with equality for some subset $\bar{N}_L \subseteq [n]$ of size $n - k + 1$. Therefore, the variables φ_k represent $S_k(c(x))$ in any optimal solution, as desired. Constraints (26b) can simply be separated by sorting the costs induced by a given integral solution and identifying the $n - k + 1$ largest of them. To avoid overloading the model, it can be helpful to combine this procedure with heuristics and cut pool management techniques.

4. Numerical experiments

This section presents the numerical experiments conducted to test the DC-based DOMP formulations from Sections 3.1 and 3.2, which are summarized in Table 2.

All models were implemented in Python 3.12 and solved using Gurobi 12.0.3 (Gurobi Optimization, LLC, 2026) on a single thread. Moreover, we deactivated presolve, heuristics, and solver generated cuts. The solver was given three termination criteria for each run: a time limit of two hours, a MIP Gap of 10^{-6} , as well as a soft memory limit of 100 GB (GB). Each of these criteria is explained in Gurobi's reference manual (Gurobi Optimization, LLC, 2026). The separation procedures were implemented, if applicable, using callbacks and triggered only at integral solutions; constraints were added whenever an absolute cut violation threshold of 0.01 was surpassed by the given solution, a choice confirmed to be suitable for our setting through preliminary experiments and post-verification of the results. Depending on the data, the tolerance may need to be adjusted to ensure validity of the solutions. No model enhancements were implemented beyond the necessary initial variable bounding for the models that incorporate separation. We ran the experiments on AMD EPYC 9454 processors on the bwUniCluster system.

Table 2

Summary of the state-of-the-art and DC-based DOMP formulations presented in this paper. The table lists, for each model, the reference section, the variable count, the constraint count (excluding variable bounds), the classification as compact (polynomial-sized) or exponential, and whether a separation procedure is implemented for the numerical experiments.

Formulation	Section	#Variables	#Constraints	Classification	Separation
DOMP _{BB}	2.2.1	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$	Compact	Yes
DOMP _r	2.2.2	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	Compact	No
DOMP _{DC:r}	3.1.1	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	Compact	No
DOMP _{DC:rB}	3.1.2	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$	Compact	Yes
DOMP _{DC:s}	3.1.3	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	Compact	No
DOMP _{DC:βB}	3.1.4	$\mathcal{O}(n^3)$	$\mathcal{O}(n^4)$	Compact	Yes
DOMP _{DC:sB}	3.2.1	$\mathcal{O}(n^2)$	$\mathcal{O}(n^n)$	Exponential	Yes
DOMP _{DC:ngc}	3.2.2	$\mathcal{O}(n^2)$	$\mathcal{O}(n^n)$	Exponential	Yes
DOMP _{DC:βe}	3.2.2	$\mathcal{O}(n^2)$	$\mathcal{O}(2^n)$	Exponential	Yes

Table 3

Summary of tested λ -vectors.

Notation	λ -vector	Name
λ_1	$(0, \dots, 0, -1)$	obnoxious p -center
λ_2	$(0, \dots, 0, -1, \dots, -1)$	obnoxious k -centrum, $k = \lfloor \frac{n}{2} \rfloor$
λ_3	$(n, n-1, \dots, 2, 1)$	reverse
λ_4	$(0, \dots, 0, \underbrace{1}_{k\text{-th position}}, 0, \dots, 0)$	k -max, $k = \lfloor \frac{n}{2} \rfloor$
λ_5	$(0, -1, 0, \dots, 0, 1, 0)$	second range
λ_6	$(\dots, 0, 0, 1, 0, 0, 1)$	001
λ_7	$\lambda \in \{-5, -4, \dots, 4, 5\}^n$	random

In order to obtain a broad picture of the performance of our DC-based models, we tested them on a multitude of different instances. For this, we follow the usual approach in the DOMP literature and consider two datasets differing by the cost structure. The first consists of randomly generated $n \times n$ cost matrices for $n \in \{20, 30, 50, 100, 150, 200\}$, where each matrix entry is a number in $[100, 1000]$ with two decimal places. The second comprises structured cost matrices of sizes $n \in \{50, 100, 150, 200\}$ derived from the test set used by Beasley (1985). The latter were for example also used by Labbé et al. (2017), Marín et al. (2020), and Ljubić et al. (2024). Per n , we always consider five different cost matrices. Further, we choose $p \in \{\lfloor \frac{n}{4} \rfloor, \lfloor \frac{n}{3} \rfloor, \lfloor \frac{n}{2} \rfloor\}$.

To complete the test sets of DOMP instances, the λ -vectors need to be chosen. Recall that all presented DC-based models use the k -sum representation for f_α and differ in the modeling of $f_{-\beta}$. We expect to observe more differences between these models when $f_{-\beta}$ is not a linear function, or equivalently, β is not a multiple of $(1, \dots, 1)$ (Grzybowski et al., 2011, Th. 3.6). This is the case whenever λ is not monotonically increasing. Table 3 summarizes our selection of such vectors on which we decided to test our models.

All tested λ -vectors have appeared in the DOMP literature before. Obnoxious criteria (λ_1, λ_2) are known to be challenging and gained interest recently (Ljubić et al., 2024; Nickel et al., 2026b). Reverse (λ_3), k -max (λ_4), and second range (λ_5) were all included in the numerical experiments by Ljubić et al. (2024). The 001 (λ_6) and random (λ_7) objectives have already been used much earlier in Boland et al. (2006) and Labbé et al. (2017), respectively.

In total, we considered 630 random and 420 structured instances. All of them, along with the detailed solution statistics are available online (Nickel et al., 2026a).

As already announced in Section 2.2, the benchmark models from the literature are DOMP_{BB} and DOMP_r. While the former outperforms other models on a large variety of difficult settings including obnoxious and fairness criteria, the latter excels on structured instances by exploiting repetitions in the cost matrix.

To compare the models, we primarily resort to the standard metrics runtime and optimality gap. Due to the presence of negative coefficients in the objective function and potentially negative optimal values, we define the optimality gap as follows. Let $\bar{z}$ denote the objective value

of the best incumbent solution and $\underline{z}$ be the best found lower bound, then the gap is computed as

$$\text{gap} := \frac{\bar{z} - \underline{z}}{\max\{1, |\bar{z}|, |\underline{z}|\}}.$$

If no solution has been found, the gap is defined to be infinite.

The rest of this section is organized as follows. In Section 4.1, we explain how we determined the best representative of the no-good cut model variants that should undergo further testing. Next, Section 4.2 discusses the results on randomly generated instances, whereas the results on the structured instances derived from Beasley's dataset are analyzed in Section 4.3. Finally, Section 4.4 provides practical recommendations for selecting among all tested models.

4.1. Preliminary experiments to determine best no-good cut model

Before proceeding with larger instances, the best version of the no-good cut model should be determined on the small instances with $n = 20$, $p \in \{\lfloor \frac{n}{4} \rfloor, \lfloor \frac{n}{3} \rfloor, \lfloor \frac{n}{2} \rfloor\} = \{5, 6, 10\}$, and $\lambda \in \{\lambda_1, \lambda_3, \lambda_6\}$. Recall that the three candidate models DOMP_{DC:ngc_x}, DOMP_{DC:ngc_y}, DOMP_{DC:ngc} are distinguished by whether they add (21), (22), or both in the separation callback. Table 4 captures average runtimes and gaps on these small instances.

All models find optimal solutions consistently for instances with $p \in \{5, 6\}$. For $p = 10$, the lowest optimality gaps are, on average, achieved by DOMP_{DC:ngc_x}. The average runtime on the instances with $p \in \{5, 6\}$ is lowest for DOMP_{DC:ngc_y}, but this model always reaches the time limit for $p = 10$. In that case, DOMP_{DC:ngc_x} dominates the runtime comparison. Adding both types of no-good cuts apparently creates too much unnecessary overhead, which explains that DOMP_{DC:ngc} is outperformed by the other two variants.

For the remaining experiments, we will consider DOMP_{DC:ngc_x} as the representative no-good cut model.

4.2. Random instances

In the following, we report and discuss the results on the random instance set. We distinguish between small- to medium-sized instances (Section 4.2.1, $n \in \{20, 30, 50\}$) and large instances (Section 4.2.2, $n \in \{100, 150, 200\}$).

4.2.1. Small- to medium-sized random instances

For $n \in \{20, 30, 50\}$, Fig. 2 shows the number of solved instances together with the average runtime until termination for each of the tested models.

It is immediately apparent that the big-M formulations DOMP_{DC:βe} and DOMP_{DC:ngc_x} are the weakest models as they do not consistently find the optimal solution for $n = 20$ and even fail to solve any instance to optimality within the time limit for $n = 30$. As a consequence, we excluded these two from further consideration and did not test them at $n \geq 50$.

Table 4

Comparison of no-good cut models for runtime in seconds (cpu) and optimality gap at termination (gap). The reported values are averages across all five test instances with the given specification (λ, n, p) .

λ		DOMP _{DC:ngc}		DOMP _{DC:ngc_x}		DOMP _{DC:ngc_y}	
λ_1	n-p	cpu	gap	cpu	gap	cpu	gap
	20-5	399.59	0.00	325.86	0.00	161.22	0.00
	20-6	2328.86	0.00	1667.36	0.00	896.36	0.00
	20-10	6013.05	0.20	5484.47	0.16	7200.00	0.27
λ_3	n-p	cpu	gap	cpu	gap	cpu	gap
	20-5	673.14	0.00	315.00	0.00	232.97	0.00
	20-6	2421.32	0.00	1153.68	0.00	1073.01	0.00
	20-10	5974.00	0.99	5428.22	0.75	7200.00	1.23
λ_6	n-p	cpu	gap	cpu	gap	cpu	gap
	20-5	625.77	0.00	324.57	0.00	245.36	0.00
	20-6	3004.83	0.00	1282.68	0.00	1277.91	0.00
	20-10	6205.04	0.82	5468.49	0.61	7200.00	1.02

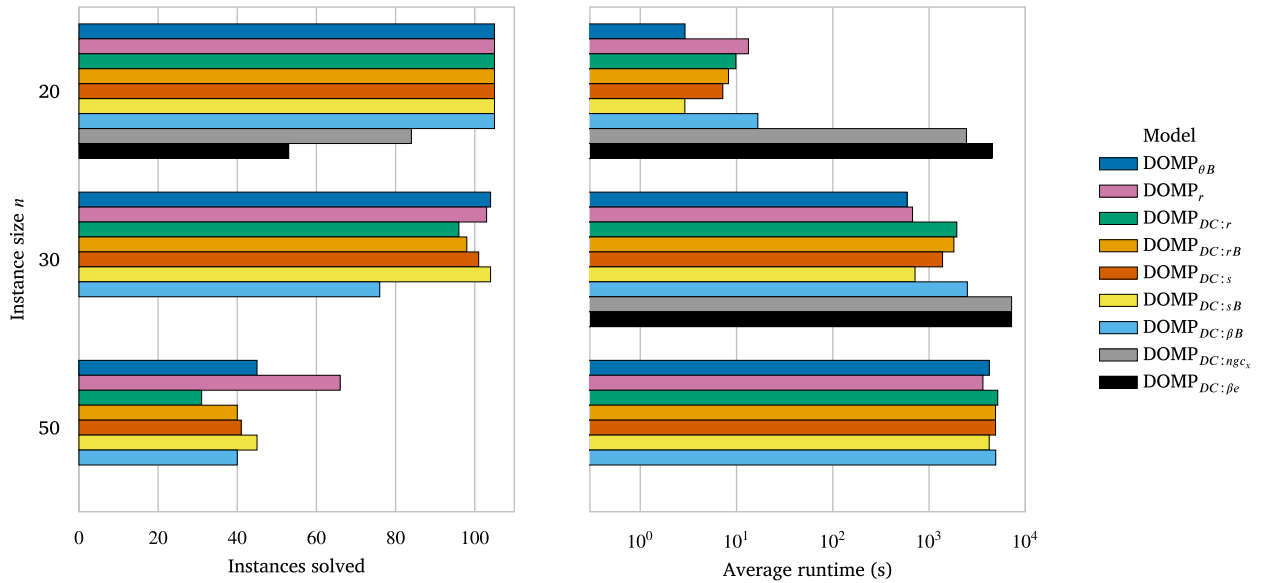


Fig. 2. Number of small- to medium-sized random instances solved to optimality and average runtime (in seconds) per instance size n .

The remaining models are all capable of consistently finding optimal solutions quickly for instances with $n = 20$. The average runtimes are mostly at 10 s or less with the exception of DOMP_r and $\text{DOMP}_{DC:\beta_B}$.

For $n = 30$, most instances can still be solved to optimality, but the average runtime increases for all models by two orders of magnitude (except for $\text{DOMP}_{DC:\beta_e}$ and $\text{DOMP}_{DC:ngc_x}$ that reached the time limit also in many cases at $n = 20$). Again, $\text{DOMP}_{DC:\beta_B}$ sticks out negatively: it solves less instances than its competitors and takes more runtime on average. Also, $\text{DOMP}_{DC:r}$ and $\text{DOMP}_{DC:rB}$ seem to be slightly less promising than the benchmark models and the models derived from the staircase formulation DOMP_s .

At $n = 50$, clearly all models start to fail in more cases. The radius benchmark model DOMP_r is the only one to solve more than half of the instances to optimality. The average runtimes are approaching the time limit.

To shed more light on the achieved solution quality, especially for instances in which optimality could not be proved within two hours, Fig. 3 plots for each model the fraction of instances that were solved within a given optimality gap.

Overall, the benchmark model DOMP_r shows the highest robustness in terms of solution quality. This success is reflected in its strong lower-bounding performance: for $n = 50$, DOMP_r terminates in 90 out of 105 instances with the best lower bound found by any of the tested models — at least double the count of any alternative formulation. DOMP_r is

followed by the model $\text{DOMP}_{DC:sB}$ and the second benchmark model DOMP_{θ_B} which achieve nearly identical solution quality overall.

Comparing $\text{DOMP}_{DC:s}$ with its Benders decomposition version $\text{DOMP}_{DC:sB}$, the latter performs slightly better. A similar observation can be made when comparing $\text{DOMP}_{DC:r}$ and $\text{DOMP}_{DC:rB}$; the Benders reformulation outperforms its counterpart.

We remark that none of the models ran out of memory for any small- to medium-sized instance.

4.2.2. Large random instances

Fig. 4 captures the optimality counts as well as the average runtimes on the large random instances with $n \in \{100, 150, 200\}$.

The number of solved instances mostly reduces to 20 or less across all n ; exceptions are $\text{DOMP}_{DC:s}$ and $\text{DOMP}_{DC:sB}$ that are capable of solving 29 and 22 out of 105 total instances for $n = 100$, respectively. Clearly, the average runtimes are therefore close to the time limit.

Note that for these instance sizes, some of the models start to reach the memory limit of 100 GB. In particular, Table 5 shows which models ran out of memory on which instance sizes.

As expected, $\text{DOMP}_{DC:\beta_B}$ with its $\mathcal{O}(n^4)$ constraints and $\mathcal{O}(n^3)$ variables in the worst case is the first to reach memory limits; already at $n = 100$, this occurs in 45 out of 105 instances. Consequently, we did not evaluate this model for instances with $n \geq 150$. Next, notice that DOMP_r does not appear in Fig. 4 for $n = 200$. The reason is that it ran out of memory on all of these instances. Furthermore, it is noticeable that

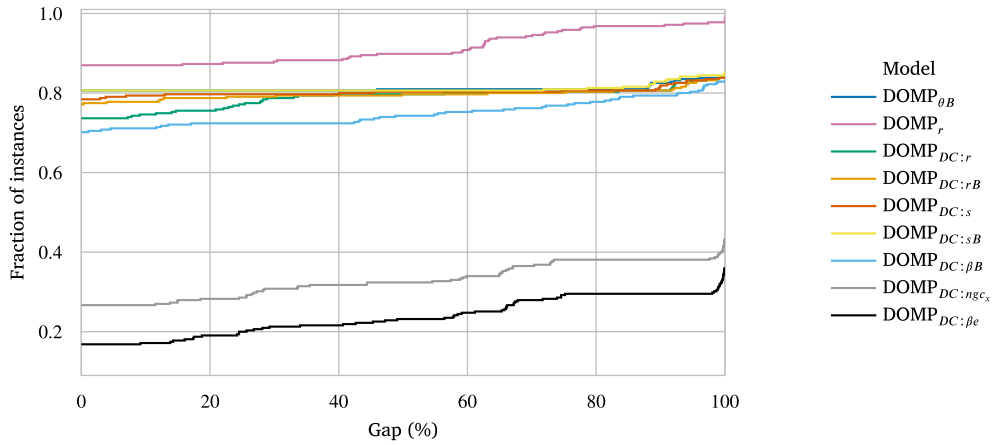


Fig. 3. Cumulative distribution of gaps (in %) at termination for small- to medium-sized random instances.

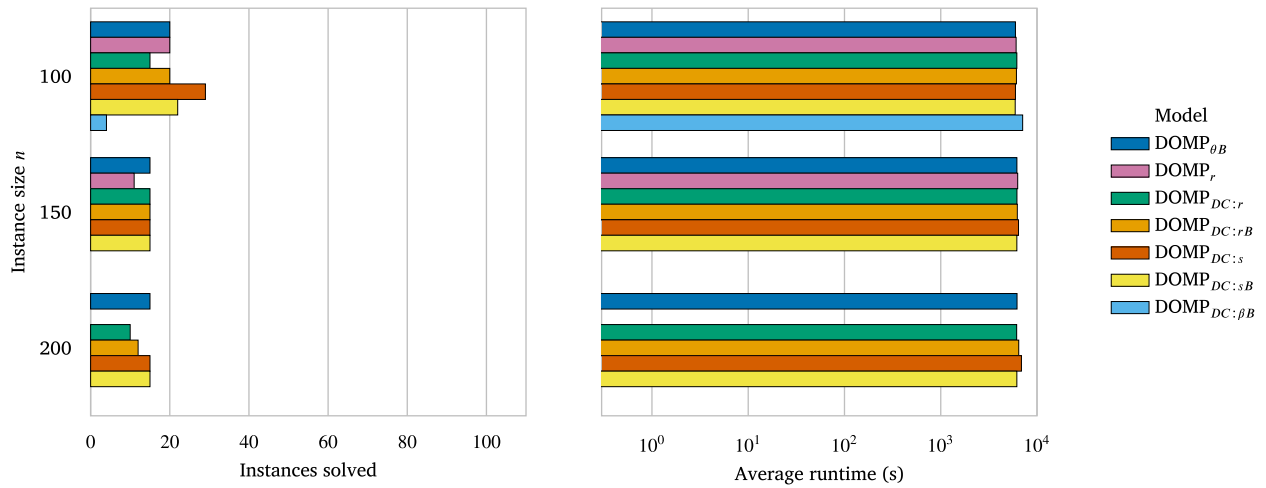


Fig. 4. Number of large random instances solved to optimality and average runtime (in seconds) per instance size n . The average runtimes are computed across all instances that did not terminate due to the memory limit.

Table 5

Out-of-memory runs per model, grouped by n . A dash indicates that no data is available.

Model	$n = 100$	$n = 150$	$n = 200$
DOMP_r	0	19	105
$\text{DOMP}_{DC:r}$	0	0	42
$\text{DOMP}_{DC:rB}$	0	0	23
$\text{DOMP}_{DC:\beta B}$	45	–	–

$\text{DOMP}_{DC:s}$ never runs out of memory while the similarly structured $\text{DOMP}_{DC:r}$ reaches the memory limit in 42 cases. The reason for this is that the accumulation of x_{ij} for $c_{ij} \geq c_{(h)}$ on the right-hand side of constraints (7c) creates a much denser coefficient matrix compared to constraints (13d) that involve summing x_{ij} for $c_{ij} = c_{(h)}$. A closer look at the memory requirements for large random instances confirms that all models that include radius variables tend to consume large amounts of memory. Fig. 5 shows the maximum memory allocation by the solver alongside the number of branch-and-bound nodes for all models, averaged across all large-scale instances of a given size n .

Indeed, $\text{DOMP}_{DC:\beta B}$, DOMP_r , $\text{DOMP}_{DC:r}$, and $\text{DOMP}_{DC:rB}$ exhibit the highest average memory requirements, in this order. Furthermore, we observe an inverse relationship for all models between memory consumption and the number of B&B nodes explored: while memory usage increases with increasing n , the number of B&B nodes decreases. On larger instances, individual node solves become computationally more

expensive. The static radius-type models DOMP_r and $\text{DOMP}_{DC:r}$ process an average of only one node for $n = 200$, indicating that the solver surpasses the memory limit at the root node before any branching occurs. Conversely, models for which constraints are added dynamically via a separation procedure – namely $\text{DOMP}_{\theta B}$, $\text{DOMP}_{DC:rB}$, $\text{DOMP}_{DC:sB}$, and $\text{DOMP}_{DC:\beta B}$ – maintain a smaller LP formulation. This reduces the time and memory required per node, allowing the solver to explore significantly more nodes compared to the static formulations. Notice that constraints (4c) and (16c), added dynamically to the memory-efficient models $\text{DOMP}_{\theta B}$ and $\text{DOMP}_{DC:sB}$, respectively, are also dense. Nevertheless, the number of generated cuts is much lower than for the memory-intensive models $\text{DOMP}_{DC:rB}$ and $\text{DOMP}_{DC:\beta B}$. Across the large random instances, $\text{DOMP}_{\theta B}$ and $\text{DOMP}_{DC:sB}$ add an average of only 203.7 and 43.5 lazy constraints, respectively. By contrast, $\text{DOMP}_{DC:rB}$ and $\text{DOMP}_{DC:\beta B}$ add 21,414.1 and 241,114.0 constraints on average.

The observed memory usage patterns and the B&B node counts directly affect the models' success in closing the optimality gap. In order to assess this aspect further, Fig. 6 captures the solution quality achieved by the models on the large random instances.

In contrast to the results on small- to medium-sized instances, the radius model DOMP_r no longer dominates. Instead, $\text{DOMP}_{DC:sB}$ stands out, again showing a similar cumulative gap distribution as $\text{DOMP}_{\theta B}$. Both of these models prove optimality gaps of less than 80% in 42% of the cases. As observed earlier, the models obtained via Benders decomposition tend to outperform their counterparts in cumulative gaps (see

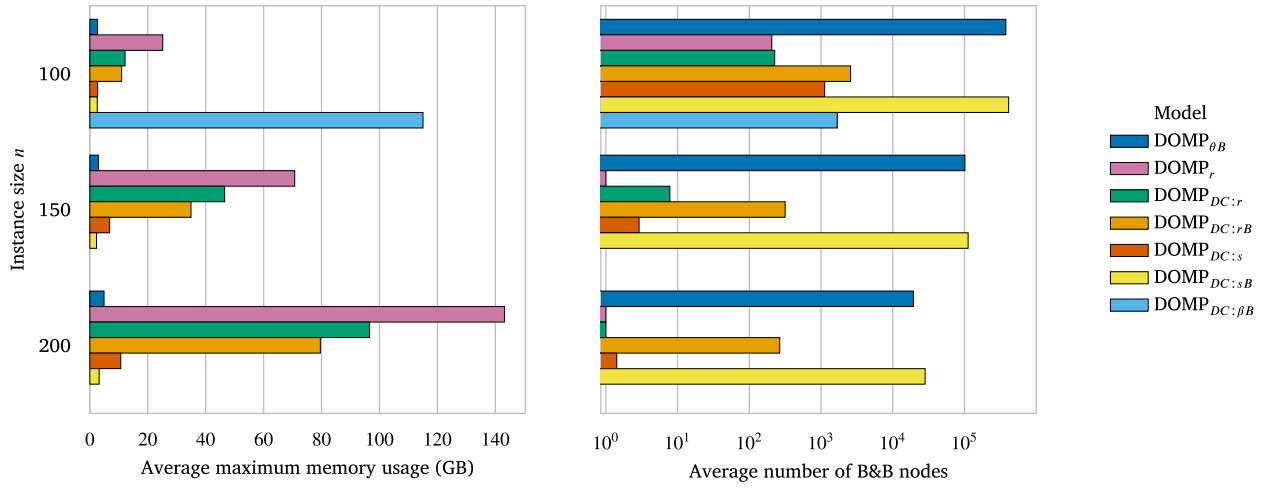


Fig. 5. Average maximum memory usage (in GB) and average number of branch-and-bound (B&B) nodes per instance size n . Memory consumption may exceed 100 GB by the definition of Gurobi's soft memory limit.

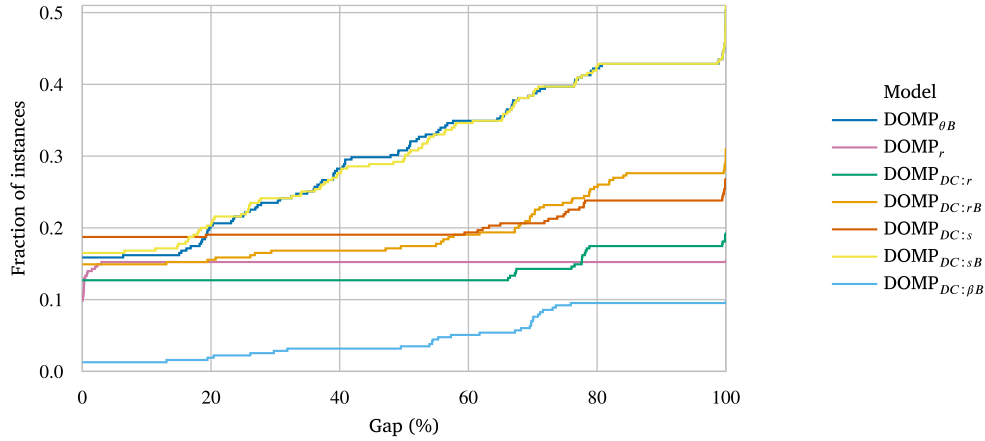


Fig. 6. Cumulative distribution of gaps (in %) at termination for large random instances.

$\text{DOMP}_{DC:rB}$ versus $\text{DOMP}_{DC:r}$, and $\text{DOMP}_{DC:sB}$ versus $\text{DOMP}_{DC:s}$). This aligns with the observation that these models explore significantly more nodes of the B&B tree (see Fig. 5), thereby reducing optimality gaps effectively.

4.3. Structured Beasley instances

Next, we turn to the structured instances based on Beasley's dataset. The number of instances solved to optimality (Fig. 7) shows that DOMP_r deals with the tested instances much better than all other models. The large number of repeated entries in the cost matrix results in a reduced model size, so that memory limits are not an issue for the radius model on these instances.

Regarding the other models, the number of solved instances as well as the average runtimes are on a similar level across the entire dataset.

This similarity becomes even more apparent from the cumulative gaps plot shown in Fig. 8.

It seems like the models based on decomposing the ordered median objective are not competitive with DOMP_r on this dataset. The truth however, is more nuanced and has to do with the considered λ . Indeed, if we restrict our view to the obnoxious k -centrum objective (λ_2), the radius model is outperformed by all other models in terms of cumulative gaps as Fig. 9 shows.

Several earlier studies have also observed that the performance of different modeling strategies depends on the instance data (e.g., Marín et al., 2009, 2020; Ljubić et al., 2024). Naturally, the weight vector

λ plays a key role, since it captures many different, even conflicting, objectives. The numerical experiments in Ljubić et al. (2024) provide further examples in which DOMP_r performs empirically worse overall than k -sum-based models. These examples include the p -center, k -centrum, range, obnoxious range, and sum of absolute differences problems (Ljubić et al., 2024), all of which are modeled by a monotone λ . Recall that, for a DC decomposition $f_\lambda = f_\alpha - f_\beta$ with monotone λ , either f_α or f_β is linear. Our intuition is that, in such cases, a model specialized to minimizing f_α or f_β will also perform well on the full problem with objective f_λ . Since the DC models are intended to combine two such specialized models, this may explain their strong performance on these instances. However, Ljubić et al. (2024) report similar results for obnoxious k -max and second obnoxious range problems, both of which are modeled by a non-monotone λ . At present, it is unclear whether one can decide a priori, based on λ , which modeling approach will perform best. This question requires further investigation.

4.4. Practical recommendations

Based on the computational study, we identify the following recommendations for selecting among the tested models. The exponential-sized big-M formulations $\text{DOMP}_{DC:ngc}$ and $\text{DOMP}_{DC:\beta e}$ are not computationally competitive and should therefore be avoided. Likewise, refining the decomposition of $f_{-\beta}$, as in $\text{DOMP}_{DC:\beta B}$, does not provide a computational advantage in our experiments.

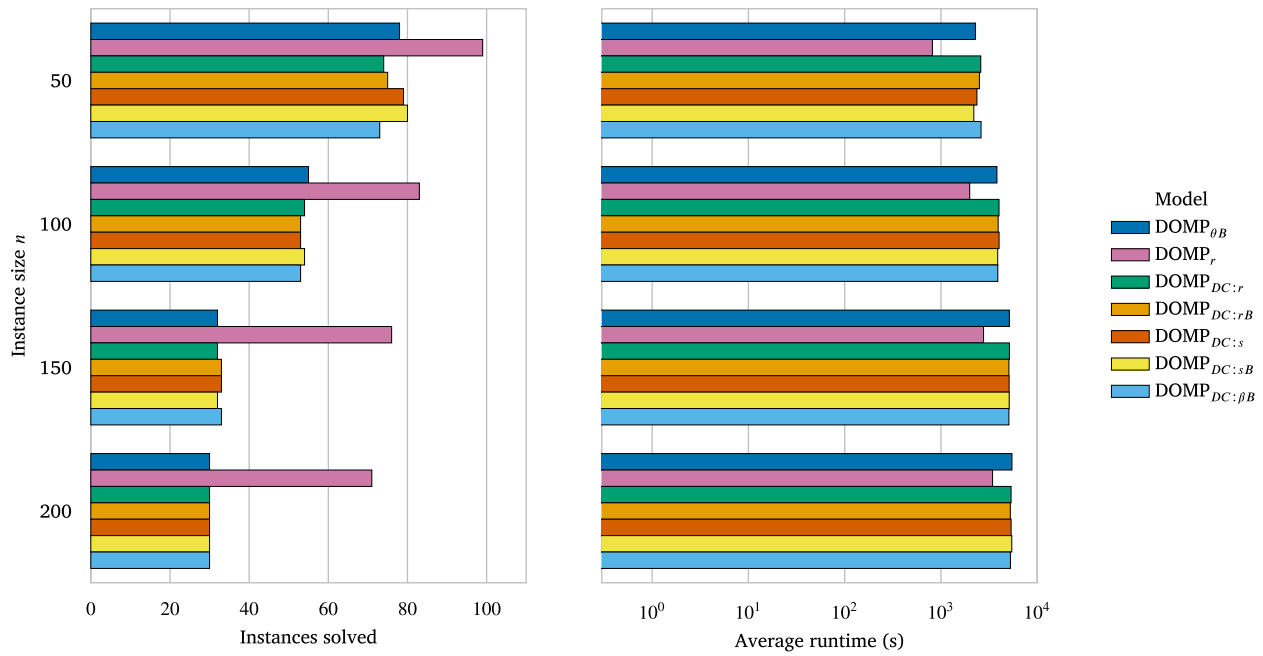


Fig. 7. Number of Beasley instances solved to optimality and average runtime (in seconds) per instance size n .

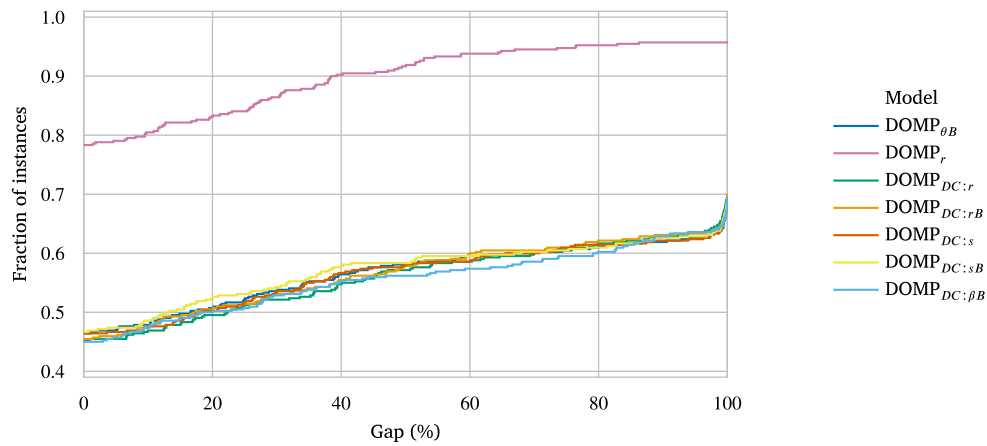


Fig. 8. Cumulative distribution of gaps (in %) at termination for Beasley instances.

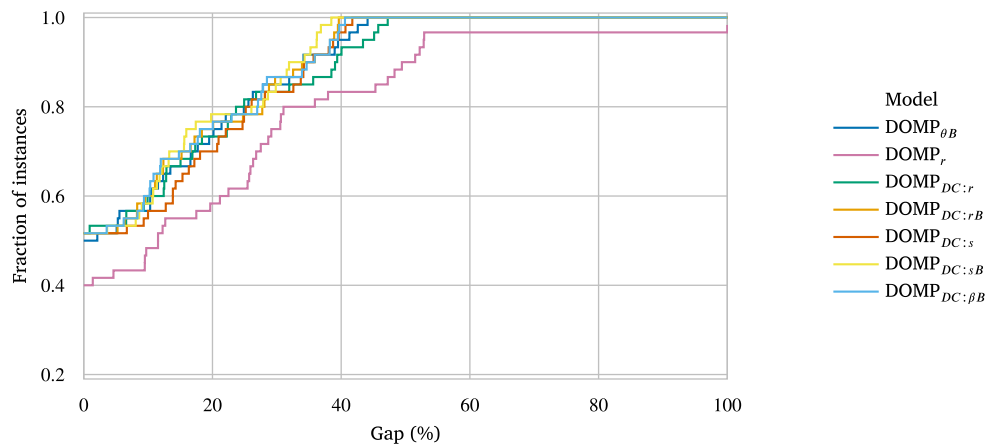


Fig. 9. Cumulative distribution of gaps (in %) at termination for Beasley instances with objective λ_2 .

Among the proposed DC formulations, those derived via Benders decomposition are more promising compared to their counterparts. Dynamic constraint generation contributes to this success: it allows the solver to start from small initial formulations, leading to fast node solves and reduced memory consumption. On large random instances, $\text{DOMP}_{DC:sB}$ achieves computational performance comparable to the benchmark formulation $\text{DOMP}_{\theta B}$, while being the most memory-efficient. On small- and medium-sized random instances and the structured Beasley instances, the radius formulation DOMP_r achieves the best average performance. However, its effectiveness depends on the optimization criterion defined by the weight vector λ . Since no single formulation consistently dominates across all instance classes, it is advisable to compare multiple promising formulations for each particular application.

Finally, we emphasize that the presented implementations intentionally omit several common model enhancements, including preprocessing, tailored valid inequalities, warm starts, and solver features such as presolve, heuristics, and cutting planes. Incorporating these enhancements is expected to further improve practical performance and may be necessary when solving large instances.

5. Conclusion

We introduced a modeling framework for the DOMP that generalizes the earlier k -sum approach and extends beyond classical location problems. This framework exploits a DC decomposition $f_\lambda = f_\alpha - f_\beta$ of the ordered median objective function, where α and β are non-negative and non-decreasing vectors. The idea is to combine specialized formulations for the DOMP with monotone objective vectors. In such cases, the ordered median function is either convex (non-decreasing weights) or concave (non-increasing weights), which enables specially structured or simplified formulations. Within this framework, we derived several compact and exponential formulations for the DOMP. Our primary focus was on modeling the concave component $f_{-\beta}$; all proposed formulations share the same representation for f_α . Despite some similarities to previously proposed models, the change in perspective revealed improvements that have so far been overlooked (e.g., $\text{DOMP}_{DC:r}$ compared to DOMP_{OTr1} by Marín et al., 2020).

Numerical experiments indicate that the exponential-sized DC models based on big-M formulations are not computationally competitive. Also, refining the decomposition of $f_{-\beta}$ – as in the case of $\text{DOMP}_{DC:\beta B}$ – does not appear to be beneficial. In contrast, the DC models derived via Benders decomposition are more promising. By generating constraints dynamically, they maintain smaller linear programming relaxations, which enables the solver to explore significantly more branch-and-bound nodes and effectively reduce optimality gaps. In particular, $\text{DOMP}_{DC:sB}$ is competitive with the benchmark model $\text{DOMP}_{\theta B}$. Formulations that use the classic radius variables (DOMP_r , $\text{DOMP}_{DC:r}$, $\text{DOMP}_{DC:rB}$) are relatively memory-intensive due to the large number of nonzero coefficients in their constraint matrices, and they reach memory limits for several large-scale instances. By comparison, $\text{DOMP}_{DC:s}$ and $\text{DOMP}_{DC:sB}$ remain within memory limits across all tested instances. Although the radius model DOMP_r performs best on average for small- to medium-sized random instances and the structured Beasley instances, its effectiveness is sensitive to the choice of the weight vector λ , and thus to the specific problem type.

Since no single formulation consistently outperforms all others, a promising direction for future research is a deeper investigation of how problem structure – captured by λ and its DC components α and β – affects the performance of different models. Insights from such an analysis could, for example, serve as the basis for training a machine learning model that predicts the most suitable formulation for a given DOMP instance, thereby enabling robust performance across a wide range of problem types. Another avenue is to further explore the modeling potential of the DC approach. This includes investigating alternative representations for f_α , as well as enhancing the formulations through tailored cutting planes and preprocessing techniques. A polyhedral study could complement these modeling advances to provide deeper structural insights into the formulations.

CRedit authorship contribution statement

Stefan Nickel: Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Justo Puerto:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Simon Ramoser:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Alberto Torrejón:** Writing – review & editing, Validation, Software, Methodology, Formal analysis.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used ChatGPT (OpenAI) and Gemini (Google) in order to improve the readability and language of the manuscript and to assist in drafting selected code snippets. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) [grant number 541767835]; and the Agencia Estatal de Investigación (AEI, MICIU/AEI/10.13039/50110001 1033) [grant numbers PCI2024-155024-2, PID2020-114594 GB-C21, PID2024-156594NB-C21, CEX2024-001517-M].

The numerical experiments were performed on the computational resource bwUniCluster. The authors acknowledge support by the State of Baden-Württemberg through bwHPC.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The instances and solution files accompanying this study are publicly available at <https://doi.org/10.17632/chjvbjvfjk.1>. The code will be made available upon request.

Instance and solution data for 'Models for the Discrete Ordered Median Problem based on DC decomposition' (Original data) (Mendeley Data)

References

- Beasley, J.E., 1985. A note on solving large p -median problems. *European J. Oper. Res.* 21, 270–273. [http://dx.doi.org/10.1016/0377-2217\(85\)90040-2](http://dx.doi.org/10.1016/0377-2217(85)90040-2).
- Benati, S., Leal, M., Puerto, J., 2025. Bilevel portfolio optimization with ordered pricing models. *Omega* 133, 103274. <http://dx.doi.org/10.1016/j.omega.2024.103274>.
- Blanco, V., 2019. Ordered p -median problems with neighbourhoods. *Comput. Optim. Appl.* 73, 603–645. <http://dx.doi.org/10.1007/s10589-019-00077-x>.
- Blanco, V., Marín, A., Puerto, J., 2024. Intra-facility equity in discrete and continuous p -facility location problems. *Comput. Oper. Res.* 162, 106487. <http://dx.doi.org/10.1016/j.cor.2023.106487>.
- Blanco, V., Pozo, M.A., Puerto, J., Torrejón, A., 2026. Optimization-based computation and integration of ordered measures. *Ann. Oper. Res.* <http://dx.doi.org/10.1007/s10479-026-07314-7>.
- Blanco, V., Puerto, J., Salmerón, R., 2018. Locating hyperplanes to fitting set of points: A general framework. *Comput. Oper. Res.* 95, 172–193. <http://dx.doi.org/10.1016/j.cor.2018.03.009>.
- Boland, N., Domínguez-Marín, P., Nickel, S., Puerto, J., 2006. Exact procedures for solving the discrete ordered median problem. *Comput. Oper. Res.* 33, 3270–3300. <http://dx.doi.org/10.1016/j.cor.2005.03.025>.

- Cesarone, F., Puerto, J., 2025. Flexible enhanced indexation models through stochastic dominance and ordered weighted average optimization. *European J. Oper. Res.* 323, 657–670. <http://dx.doi.org/10.1016/j.ejor.2024.11.050>.
- Cherkesly, M., Contardo, C., Gruson, M., 2025. Ranking Decomposition for the Discrete Ordered Median Problem. *INFORMS J. Comput.* 37, 230–248. <http://dx.doi.org/10.1287/ijoc.2023.0059>.
- Deleplanque, S., Labbé, M., Ponce, D., Puerto, J., 2020. A Branch-Price-and-Cut Procedure for the Discrete Ordered Median Problem. *INFORMS J. Comput.* 32, 582–599. <http://dx.doi.org/10.1287/ijoc.2019.0915>.
- Domínguez-Marín, P., 2003. The Discrete Ordered Median Problem: Models and Solution Methods. *Comb. Optim.* <http://dx.doi.org/10.1007/978-1-4419-8511-8>.
- Drezner, Z., Nickel, S., 2009. Constructing a DC decomposition for ordered median problems. *J. Global Optim.* 45, 187–201. <http://dx.doi.org/10.1007/s10898-008-9326-6>.
- Grzybowski, J., Kalcsics, J., Nickel, S., Pallaschke, D., Urbański, R., 2015a. On maxim representations of ordered median functions. *Optim.* 64, 339–348. <http://dx.doi.org/10.1080/02331934.2012.750322>.
- Grzybowski, J., Kalcsics, J., Nickel, S., Pallaschke, D., Urbański, R., 2015b. On topological types of ordered median functions. *Optim.* 64, 149–160. <http://dx.doi.org/10.1080/02331934.2014.883512>.
- Grzybowski, J., Kalcsics, J., Nickel, S., Pallaschke, D., Urbański, R., 2018. Ascent and descent cones of ordered median block functions. *Optim.* 67, 507–522. <http://dx.doi.org/10.1080/02331934.2018.1429432>.
- Grzybowski, J., Nickel, S., Pallaschke, D., Urbański, R., 2011. Ordered median functions and symmetries. *Optim.* 60, 801–811. <http://dx.doi.org/10.1080/02331931003677772>.
- Grzybowski, J., Pallaschke, D., Urbański, R., 2016. Reduced Pairs of Compact Convex Sets and Ordered Median Functions. *J. Optim. Theory Appl.* 171, 354–364. <http://dx.doi.org/10.1007/s10957-015-0860-3>.
- Gurobi Optimization, LLC, 2026. Gurobi optimizer reference manual. <https://www.gurobi.com> (Accessed 15 April 2026).
- Labbe, M., Ponce, D., Puerto, J., 2017. A comparative study of formulations and solution methods for the discrete ordered p -median problem. *Comput. Oper. Res.* 78, 230–242. <http://dx.doi.org/10.1016/j.cor.2016.06.004>.
- Ljubić, I., Marín, A., Puerto, J., Temprano, F., 2026a. Ordered Median Traveling Salesman Problem. *Networks* <http://dx.doi.org/10.1002/net.70036>.
- Ljubić, I., Pozo, M.A., Puerto, J., Torrejon, A., 2024. Benders decomposition for the discrete ordered median problem. *European J. Oper. Res.* 317, 858–874. <http://dx.doi.org/10.1016/j.ejor.2024.04.030>.
- Ljubić, I., Puerto, J., Torrejon, A., 2026b. Fair Vehicle Routing via Bilevel Optimization. <https://optimization-online.org/?p=33847>, Optim. Online Preprint.
- Marín, A., Martínez-Merino, L.I., Puerto, J., Rodríguez-Chía, A.M., 2022. The soft-margin Support Vector Machine with ordered weighted average. *Knowl.-Based Syst.* 237, 107705. <http://dx.doi.org/10.1016/j.knsys.2021.107705>.
- Marín, A., Nickel, S., Puerto, J., Velten, S., 2009. A flexible model and efficient solution strategies for discrete location problems. *Discrete Appl. Math.* 157, 1128–1145. <http://dx.doi.org/10.1016/j.dam.2008.03.013>.
- Marín, A., Ponce, D., Puerto, J., 2020. A fresh view on the Discrete Ordered Median Problem based on partial monotonicity. *European J. Oper. Res.* 286, 839–848. <http://dx.doi.org/10.1016/j.ejor.2020.04.023>.
- Nickel, S., 2001. Discrete Ordered Weber Problems. In: Fleischmann, B., Lasch, R., Derigs, U., Domschke, W., Rieder, U. (Eds.), *Operations Research Proceedings*. Springer Berlin, Heidelberg, Berlin, Heidelberg, pp. 71–76. http://dx.doi.org/10.1007/978-3-642-56656-1_12.
- Nickel, S., Puerto, J., 1999. A unified approach to network location problems. *Networks* 34, 283–290. [http://dx.doi.org/10.1002/\(SICI\)1097-0037\(199912\)34:4<283::AID-NET8>3.0.CO;2-2](http://dx.doi.org/10.1002/(SICI)1097-0037(199912)34:4<283::AID-NET8>3.0.CO;2-2).
- Nickel, S., Puerto, J., 2005. *Location Theory*. Springer Berlin, Heidelberg, Berlin/Heidelberg, <http://dx.doi.org/10.1007/3-540-27640-8>.
- Nickel, S., Puerto, J., Ramoser, S., Torrejon, A., 2026a. Instance and solution data for 'Models for the Discrete Ordered Median Problem based on DC decomposition'. Mendeley Data, V1. <http://dx.doi.org/10.17632/chjvbjvfjk.1>.
- Nickel, S., Puerto, J., Ramoser, S., Torrejon, A., 2026b. Revisiting transportation problems under Monge costs with applications to the Discrete Ordered Median Problem. <http://dx.doi.org/10.48550/arXiv.2602.15151>, ArXiv Preprint.
- Nickel, S., Velten, S., 2017. Optimization problems with flexible objectives: A general modeling approach and applications. *European J. Oper. Res.* 258, 79–88. <http://dx.doi.org/10.1016/j.ejor.2016.08.045>.
- Ogryczak, W., Tamir, A., 2003. Minimizing the sum of the k largest functions in linear time. *Inform. Process. Lett.* 85, 117–122. [http://dx.doi.org/10.1016/S0020-0190\(02\)00370-8](http://dx.doi.org/10.1016/S0020-0190(02)00370-8).
- Papini, P.L., Puerto, J., 2004. Averaging the k largest distances among n : k -centra in Banach spaces. *J. Math. Anal. Appl.* 291, 477–487. <http://dx.doi.org/10.1016/j.jmaa.2003.11.011>.
- Pozo, M.A., Puerto, J., Rodríguez-Chía, A.M., 2021. The ordered median tree of hubs location problem. *TOP* 29, 78–105. <http://dx.doi.org/10.1007/s11750-020-00572-z>.
- Pozo, M.A., Puerto, J., Torrejon, A., 2024. The Ordered Median Tree Location Problem. *Comput. Oper. Res.* 169, 106746. <http://dx.doi.org/10.1016/j.cor.2024.106746>.
- Puerto, J., 2008. A New Formulation of the Capacitated Discrete Ordered Median Problems with $\{0, 1\}$ -Assignment. *Operations Research Proceedings 2007*. Springer, Berlin, Heidelberg, pp. 165–170. http://dx.doi.org/10.1007/978-3-540-77903-2_26.
- Puerto, J., Fernández, F.R., 2000. Geometrical properties of the symmetrical single facility location problem. *J. Nonlinear Convex Anal.* 1, 321–342.
- Puerto, J., Ramos, A., Rodríguez-Chía, A., 2011. Single-allocation ordered median hub location problems. *Comput. Oper. Res.* 38, 559–570. <http://dx.doi.org/10.1016/j.cor.2010.07.018>.
- Puerto, J., Ramos, A., Rodríguez-Chía, A., 2013. A specialized branch & bound & cut for Single-Allocation Ordered Median Hub Location problems. *Discrete Appl. Math.* 161, 2624–2646. <http://dx.doi.org/10.1016/j.dam.2013.05.035>.
- Puerto, J., Ramos, A., Rodríguez-Chía, A., Sánchez-Gil, M., 2016. Ordered median hub location problems with capacity constraints. *Transp. Res. Part C Emerg. Technol.* 70, 142–156. <http://dx.doi.org/10.1016/j.trc.2015.05.012>.
- Puerto, J., Rodríguez-Chía, A.M., Fernández-Palacín, F., 1997. Ordered Weber problems with attraction and repulsion. *Stud. Locat. Anal.* 11, 127–141.
- Puerto, J., Rodríguez-Chía, A.M., Tamir, A., 2017. Revisiting k -sum optimization. *Math. Program.* 165, 579–604. <http://dx.doi.org/10.1007/s10107-016-1096-1>.
- Puerto, J., Torrejon, A., 2025. A fresh view on Least Quantile of Squares Regression based on new optimization approaches. *Expert Syst. Appl.* 282, 127705. <http://dx.doi.org/10.1016/j.eswa.2025.127705>.
- Rodríguez-Chía, A.M., Nickel, S., Puerto, J., Fernández, F.R., 2000. A flexible approach to location problems. *Math. Methods Oper. Res.* 51, 69–89. <http://dx.doi.org/10.1007/s001860050003>.