Received: 23 October 2021 | Revised: 23 June 2022 | Accepted: 30 October 2022 | Published on: 22 November 2022

DOI: 10.1002/net.22132

RESEARCH ARTICLE

WILEY

Formulations and valid inequalities for the capacitated dispersion problem

Mercedes Landete[1] | Juanjo Peiró[2] | Hande Yaman[3]

[1]Departamento de Estadística, Matemáticas e Informática, Instituto Centro de Investigación Operativa, Universidad Miguel Hernández de Elche, Elche, Spain
[2]Departament d'Estadística i Investigació Operativa, Facultat de Ciències Matemàtiques, Universitat de València, Valencia, Spain
[3]Research Centre for Operations Research and Business Statistics (ORSTAT), Faculty of Economics and Business, KU Leuven, Leuven, Belgium

Correspondence
Juanjo Peiró, Departament d'Estadística i Investigació Operativa, Facultat de Ciències Matemàtiques, Universitat de València, Valencia, Spain.
Email: juanjo.peiro@uv.es

Funding information
Generalitat Valenciana, Grant/Award Numbers: PROMETEO/2021/063, CIGE/2022/57; Ministerio de Ciencia, Innovación y Universidades del Gobierno de España, Grant/Award Numbers: PGC2018-099428-B-100, PID2021-122344NB-I00, RED2018-102363-T; Ministerio de Educación y Formación Profesional del Gobierno de España, Grant/Award Number: CAS19/00076.

Abstract

This work focuses on the capacitated dispersion problem for which we study several mathematical formulations in different spaces using variables associated with nodes, edges, and costs. The relationships among the presented formulations are investigated by comparing the projections of the feasible sets of the LP relaxations onto the subspace of natural variables. These formulations are then strengthened with families of valid inequalities and variable-fixing procedures. The separation problems associated with the valid inequalities that are exponential in number are shown to be polynomially solvable by reducing them to longest path problems in acyclic graphs. The dual bounds obtained from stronger but larger formulations are used to improve the strength of weaker but smaller formulations. Several sets of computational experiments are conducted to illustrate the usefulness of the findings, as well as the aptness of the formulations for different types of instances.

KEYWORDS

dispersion problem, extended formulation, location science, polyhedral combinatorics, separation, telescopic sums, valid inequalities

1 | INTRODUCTION

Dispersion problems, also known as *diversity* problems, are a family of $\mathcal{NP}$-hard optimization problems that consist in selecting a subset of elements from a given set in such a way that a distance measure between the selected elements is maximized. These problems are found in different application fields, specially when diversity is a constituent factor: biology, biodiversity and ecology, genetics, ethnicity and gender diversity, heterogeneous formation of work teams and committees, academic curriculum design, public facility location, market planning, financial portfolio design, and so forth.

Several versions of dispersion problems can be identified in the literature. From the perspective of their objective function, the most common is the maximization of the sum of distances among the selected elements (MaxSum). It is also frequent to find objectives aiming at maximizing the minimum distance (MaxMin). Less frequent are those objectives that maximize the mean (MaxMean) or the minimum sum (MaxMinSum) of such distances. In [12, 32], a variety of different-nature objective functions in the context of dispersion are covered, and in [30] a detailed study of the solution structure obtained with four mathematical objectives is conducted. From these papers, we can conclude that the MaxMin objective function is particularly useful when it is desirable to obtain elements in a set that are far from one to another and, at the same time, as equidistant as possible. From the perspective of the size of the set of selected elements that constitute a feasible solution, two types of problems have drawn the attention of researchers: on the one hand, the so-called p-dispersion variants for which that size, say p, is exogenously fixed; on the other hand, we find models subject to economic or physical resource constraints but not explicitly restricted to obtain solution sets with cardinality p, known as *constrained* dispersion variants. Briefly, we can say that the former ones directly address their applicability requirements by considering a prefixed number of elements to select. However, this simplification might sometimes be unrealistic and, in those cases, the latter may be preferred.

As reviewed in [25], the first results on maximizing dispersion started to be published for the p-dispersion version in [6] where this problem is studied on a tree-graph structure. In [20], the study was extended to more general graphs and an integer linear formulation was proposed and applied to small examples. Later, advances in solution techniques for this version were proposed: the first design of a heuristic and an exact method to solve small instances of the problem was proposed in [11], followed by the simulated annealing and tabu search proposals in [19, 21], respectively. A proof that its MaxMin version is $\mathcal{NP}$-hard was provided in [17], where a proposal of a greedy randomized heuristic for its solution was also given. Since [18], several heuristic algorithms for the MaxSum model have been proposed. They were reviewed and compared to a scatter search based technique in [15]. Later, an application of the GRASP methodology for solving the MaxMin version in a very efficient manner was proposed in [34]. More recently, an exact procedure for solving the p-dispersion problem based on the relationship between the p-dispersion problem and a collection of node packing problems was devised in [37], which allowed to solve instances with up to 1000 nodes in less than one hour of computing time.

In the context of constrained dispersion models, the work [35] expanded the horizons of dispersion problems with the inclusion of capacity and cost constraints, originally motivated by applications in location science. Lately, several papers have tackled these constrained problems: [26, 28, 31] studied the dispersion problem with a capacity constraint, and [27] extended these models in several directions by the addition of different features related to costs and capacity issues, generalized some of these models and proposed methodologies to solve them.

In this work, we focus on a version of the constrained dispersion problem known as the capacitated dispersion problem (CDP). In it, given a set V of n elements that we call nodes or facilities, a positive capacity c_i for each node i, a nonnegative distance d_{ij} between any pair of distinct nodes i and j, and a positive demand B to cover, we would like to find a subset V' of nodes such that their sum of capacities is large enough to cover the demand, that is, $\sum_{i\in V'\subseteq V} c_i \geq B$, and the nodes in V' are as distant as possible from one another, that is, $\min_{i,j\in V'\subseteq V : i\neq j} d_{ij}$ is maximum. This version was originally proposed in [35] and coined as the MAX-DIST/CAP problem. Some applications of this problem version arise in the context of:

- Public urban facility planning, when one looks to determine the location of facilities to cover a general demand for a service, and seeks for the facilities to be as distant as possible, for example, in the case of power generation plants to decrease the risk of damage from accidents at other facilities. Also, when determining the location of *similar-type* facilities like hospitals, schools or airports, to avoid overlapping geographical areas of services since locating them close to each other may deteriorate the ease of access by a large extent of the population (see, e.g., [36]).
- Franchise business schemes, to avoid local competition and cannibalization effects within a chain of retail franchises [27].
- Indoor layout planning when social distance is required, for example, during the COVID-19 pandemic [7, 29].

In [31], the CDP was first formulated as the following nonlinear integer program:

$$\begin{aligned}
\max\ & \min_{i,j\in V\,:\,i<j} d_{ij}x_i x_j,\\
\text{s.t.:}\ & \sum_{i\in V} c_i x_i \geq B,\\
& x_i \in \{0,1\}, \quad \forall i \in V,
\end{aligned}$$

where variable x_i takes value 1 if node $i \in V$ is selected (facility i is open), and 0 otherwise. The developments in [26, 28, 31] pay special attention to the adaptation of powerful metaheuristic methodologies to solve this dispersion variant. However, as pointed out in [25], no work about theoretical aspects of its formulations has been presented so far.

In this work, we try to cover this research gap. To do so, we first present some general results that are based on the x-variables defined above (Section 2). Then, three formulations and their associated valid inequalities are introduced, and methods to solve the related separation problems when the valid inequalities are exponential in number are provided. We compare the bounds of the linear programming relaxations of these formulations theoretically (Section 3). We conduct a computational study to see the performances of such formulations and valid inequalities, as well as the scope of applicability of each model. Based on the results, we outline an approach that makes use of stronger and larger formulations to improve the performance of weaker and smaller formulations (Section 4). We finish this work with some conclusions and motivate future research directions.

2 | GENERAL RESULTS

To eliminate trivial cases, we assume that at least two facilities should be open to cover the demand, that is, $c_i < B$ for all $i \in V$, and that we do not need to open all facilities, that is, there exists $j \in V$ such that $\sum_{i \in V \setminus \{j\}} c_i \geq B$. If all distances are equal to each other, then the optimal value is equal to this distance if it is possible to cover the demand, and the problem is infeasible otherwise. So we also assume that there are at least two different distance values. Finally, we assume, without loss of generality, that $d_{ij} = d_{ji}$ for all $i, j \in V$ with $i < j$ since, otherwise, we can consider the larger distance and ignore the other.

Let $X = \{x \in \{0, 1\}^n : \sum_{i \in V} c_i x_i \geq B\}$. Set X is a 0-1 knapsack cover set (see, e.g., [3], and the references therein) and any valid inequality for this set is also valid for the feasible sets of all our formulations. The aim of our study is to propose other valid inequalities.

First we have an optimality condition to break symmetry:

Proposition 1. *Let $i, j \in V$ be two distinct nodes such that $d_{it} \leq d_{jt}$ for all $t \in V \setminus \{i, j\}$ and $\max_{t \in V} d_{it} \leq d_{ij}$. Then there exists an optimal solution to CDP such that $x_i \leq x_j$.*

Proof. Suppose that i and j are such that $i \neq j$, $d_{it} \leq d_{jt}$ for all $t \in V \setminus \{i, j\}$ and $\max_{t \in V} d_{it} \leq d_{ij}$. Let x be an optimal solution with $x_i = 1$ and $x_j = 0$, and $V' = \{k \in V : x_k = 1\}$ be the set of open facilities in the solution. Observe that the optimal value is less than or equal to $\min_{k \in V' \setminus \{i\}} d_{ik}$. Setting $x_j = 1$ gives an alternative optimal solution since $\min_{k \in V' \setminus \{i\}} d_{ik} \leq d_{it} \leq d_{jt}$ for all $t \in V' \setminus \{i\}$ and $\min_{k \in V' \setminus \{i\}} d_{ik} \leq \max_{t \in V} d_{it} \leq d_{ij}$. ■

Next, we propose an approach to compute lower and upper bounds on the optimal value of CDP.

For each $j \in V$, we first compute an upper bound on the optimal value of the problem in which we enforce to open facility j by maximizing the minimum distance between j and other facilities to open to be able to cover the demand.

Proposition 2. *Let $x \in X$ and $j \in V$. If $x_j = 1$, then the objective function value for x is at most $\theta_j = \max_{V' \subseteq V \setminus \{j\} : \sum_{t \in V'} c_t + c_j \geq B} \min_{i \in V'} d_{ji}$.*

Proof. For the problem restricted with $x_j = 1$, θ_j is the optimal value of the relaxation obtained by replacing all distances d_{it} with $i, t \neq j$ with a very large number. ■

To compute θ_j, we sort the facilities in $V \setminus \{j\}$ in nonincreasing order of d_{jt}'s (if ties occur, we favor nodes with larger capacity) and find the smallest integer l such that $\sum_{t=1}^{l} c_t \geq B - c_j$. Then $\theta_j = d_{jl}$.

Remark 1. $\text{UB} = \max_{V' \subseteq V : \sum_{t \in V'} c_t \geq B} \min_{j \in V'} \theta_j$ is an upper bound on the optimal value of CDP.

In the sequel, we assume that V is sorted by θ_j such that $\theta_1 \geq \cdots \geq \theta_{|V|}$ favoring nodes with larger capacities. Given this order, it is straightforward to compute *UB* by finding the smallest integer l such that $\sum_{j=1}^{l} c_j \geq B$. Then $\text{UB} = \theta_l$.

In some cases, we may improve this upper bound as follows.

Proposition 3. *Let l be the smallest integer with $\sum_{j=1}^{l} c_j \geq B$. If $C = \{1, \ldots, l\}$ is such that $\sum_{j \in C \setminus \{t\}} c_j < B$ for all $t \in C$ and if $\min_{i,j \in C : i<j} d_{ij} < UB$ then we update UB as $\text{UB} = \max\{\min_{i,j \in C : i<j} d_{ij}, \theta_{l+1}\}$.*

Moreover, if $\text{UB} = \min_{i,j \in C : i<j} d_{ij}$, then x^* with $x_i^* = 1$ if $i \in C$ and $x_i^* = 0$ otherwise is an optimal solution and UB is the optimal value of CDP.

Proof. If $C = \{1, \ldots, l\}$ is such that $\sum_{j \in C \setminus \{t\}} c_j < B$ for all $t \in C$, then a feasible solution either uses exactly the nodes in set C or it includes at least one node j^* from $V \setminus C$. If $\min_{i,j \in C : i<j} d_{ij} < UB$, then if the solution uses exactly the nodes in set C then the objective function value of this solution is $\min_{i,j \in C : i<j} d_{ij}$. If the solution includes at least one node j^* from $V \setminus C$, then as V is sorted, we know that $\theta_{j^*} \leq \theta_{l+1}$. Hence $\max\{\min_{i,j \in C : i<j} d_{ij}, \theta_{l+1}\}$ is a valid upper bound.

If $\text{UB} = \min_{i,j \in C : i<j} d_{ij}$, then x^* with $x_i^* = 1$ if $i \in C$ and $x_i^* = 0$ otherwise is feasible and $\min_{i,j \in C : i<j} d_{ij}$ is also a lower bound. This proves that x^* is an optimal solution. ■

Remark 2. By computing θ_j's and UB, we find subsets of V with sufficient capacity to cover the demand. The minimum distance among the nodes in each subset is a lower bound (*LB*) for CDP.

Example 1. To illustrate the above results, consider the next example with five nodes and the following distances:

$i \setminus j$	1	2	3	4	5
1	–	4	3	3	4
2	4	–	4	2	2
3	3	4	–	6	6
4	3	2	6	–	2
5	4	2	6	2	–

The capacities are $c = (2, 2, 1, 1, 1)$ and $B = 5$. First, we compute the θ's. To compute θ_1, we sort V as $2 - 5 - 3 - 4$. Then we open facilities using this order until the capacity is at least $B - c_1 = 3$. We need to open facilities 2 and 5 and thus $\theta_1 = d_{15} = 4$. The lower bound we obtain from the feasible solution in which we open facilities 1, 2 and 5 is 2 since $d_{25} = 2$. Similarly, we have $\theta_2 = 4$ (we open 1 and 3 in addition to 2 and the lower bound is 3), $\theta_3 = 4$ (we open 4, 5 and 2 in addition to 3 and the lower bound is 2), $\theta_4 = 2$ (we open 3, 1 and 2 in addition to 4 and the lower bound is 2), $\theta_5 = 2$ (we open 3, 1 and 2 in addition to 5 and the lower bound is 2). So $\theta = (4, 4, 4, 2, 2)$. As set V is already sorted with respect to θ's, we can immediately calculate $\text{UB} = 4$. Now, since $C = \{1, 2, 3\}$ is such that the demand cannot be covered if any node in C is not used and $\min_{i,j \in C : i<j} d_{ij} = 3$, we can update the upper bound as $\text{UB} = \max\{\min_{i,j \in C : i<j} d_{ij}, \theta_{l+1}\} = \max\{3, 2\} = 3$. As we also have a lower bound of 3, the problem is solved.

3 | FORMULATIONS AND VALID INEQUALITIES

We present three mathematical formulations to tackle the problem, along with families of valid inequalities and variable-fixing processes to strengthen each of them. In addition to this, we connect these formulations by comparing the projections of the LP relaxations' feasible sets by taking advantage of the fact that all have in common the x-variables previously defined. Their main differences arise from the point of view we take when modeling the problem: nodes, edges, or costs—a priori information that could be exploited in order to speed up the solution process.

The goal of this section is then twofold: on the one hand, to provide possible formulations for the problem; on the other hand, to theoretically compare these formulations in order to identify which of them are more promising for computation.

3.1 | Kuby formulation

Our first formulation is a direct modification of the p-dispersion formulation from [20] (see also [36]) and consists of:

$$\max m, \tag{1}$$

$$\text{s.t.: } m \leq d_{ij} + D_{\max}(2 - x_i - x_j), \quad \forall i, j \in V : i < j, \tag{2}$$

$$x \in X, \tag{3}$$

where $D_{\max}$ is the largest distance, that is, $D_{\max} = \max_{i,j \in V : i<j} d_{ij}$. Let $K0$ be this formulation and X_K be its feasible set. The projection of X_K on to the space of x, denoted by $\text{Proj}_x X_K$, is the same as set X.

In the sequel, $\overline{D}$ is an upper bound and $\underline{D}$ is a lower bound onto the optimal value of CDP. Without loss of generality, we assume that each one is equal to one of the distances. An obvious upper bound is $D_{\max}$ and an obvious lower bound is $D_{\min} = \min_{i,j \in V : i<j} d_{ij}$.

Proposition 4. *The formulation obtained by replacing constraints (2) with*

$$m \leq d_{ij} + (\overline{D} - d_{ij})(2 - x_i - x_j), \quad \forall i, j \in V : i < j, d_{ij} < \overline{D} \tag{4}$$

and by adding the constraint

$$m \leq \overline{D}, \tag{5}$$

is a correct formulation. The smaller $\overline{D}$ is, the stronger the formulation is.

Proof. As the upper bound $\overline{D}$ corresponds to a distance value, either $m = \overline{D}$ or the optimal solution uses at least two nodes i and j whose distance $d_{ij} < \overline{D}$. In the latter case, constraints (4) for such pairs of i and j ensure that $m \leq d_{ij}$. For a pair i, j such that $d_{ij} < \overline{D}$ and $x_i + x_j \leq 1$, constraint (4) is redundant since $d_{ij} + (\overline{D} - d_{ij})(2 - x_i - x_j) = \overline{D} + (\overline{D} - d_{ij})(1 - x_i - x_j) \geq \overline{D}$. This shows that the formulation is correct.

Let D_1 and D_2 be two upper bounds with $D_1 > D_2$. For a pair $i, j \in V$ with $i < j$,

- if $d_{ij} < D_2$, then (4) with $\overline{D} = D_2$ implies (4) with $\overline{D} = D_1$,
- if $D_1 > d_{ij} \geq D_2$, then (5) with $\overline{D} = D_2$ implies (4) with $\overline{D} = D_1$ since $d_{ij} + (D_1 - d_{ij})(2 - x_i - x_j) \geq d_{ij} \geq D_2$,
- and if $d_{ij} \geq D_1$, then (5) with $\overline{D} = D_2$ implies (5) with $\overline{D} = D_1$.

Hence the formulation with $\overline{D} = D_2$ is stronger than the formulation with $\overline{D} = D_1$. ■

Corollary 1. *If $\overline{D} = D_{\max}$, then constraints (4) and (5) become*

$$m \leq d_{ij} + (D_{\max} - d_{ij})(2 - x_i - x_j), \quad \forall i, j \in V : i < j. \tag{6}$$

It is easy to see that this formulation is stronger than $K0$.

Proposition 5. *There exists an optimal solution to CDP that satisfies*

$$x_j = 0, \quad \forall j \in V : \theta_j < \underline{D}, \tag{7}$$

$$x_i + x_j \leq 1, \quad \forall i, j \in V : i < j, d_{ij} < \underline{D}. \tag{8}$$

Corollary 2. *If $\underline{D} = D_{\min}$, then we do not get any inequalities (7) and (8).*

Let $K(\overline{D}, \underline{D})$ be the formulation obtained by replacing constraints (2) with (4) and (5) and adding (7) and (8).

Proposition 6. *Let $S \subseteq \{j \in V : \theta_j < \overline{D}\}$. Without loss of generality, assume that $S = \{j_1, \ldots, j_{|S|}\}$ and is sorted in the same way as V. Let $\theta_{j_0} = \overline{D}$. Then,*

$$m \leq \overline{D} - \sum_{s=1}^{|S|} (\theta_{j_{s-1}} - \theta_{j_s}) x_{j_s} \tag{9}$$

is a valid inequality for set X_K.

Proof. Given a solution $(x, m) \in X_K$, let $S^* = \{j \in S : x_j = 1\}$. If $S^* = \emptyset$, then $m \leq \overline{D}$ is satisfied. Otherwise, let $S^* = \{j_1^*, \ldots, j_{|S^*|}^*\}$ be sorted in the same manner as V. We know that $\theta_{j^*_{|S^*|}}$ is an upper bound on m. Let $\theta_{j_0^*} = \overline{D}$. The right-hand side of the inequality is $\overline{D} - \sum_{s=1}^{|S|} (\theta_{j_{s-1}} - \theta_{j_s}) x_{j_s} \geq \overline{D} - \sum_{t=1}^{|S^*|} (\theta_{j^*_{t-1}} - \theta_{j^*_t})$ since for each $j_t^* \in S^*$ if s is such that $j_t^* = j_s$, then $\theta_{j^*_{t-1}} \geq \theta_{j_{s-1}}$. Since $\overline{D} - \sum_{t=1}^{|S^*|} (\theta_{j^*_{t-1}} - \theta_{j^*_t}) = \theta_{j^*_{|S^*|}}$, we have that $m \leq \theta_{j^*_{|S^*|}} = \overline{D} - \sum_{t=1}^{|S^*|} (\theta_{j^*_{t-1}} - \theta_{j^*_t}) \leq \overline{D} - \sum_{s=1}^{|S|} (\theta_{j_{s-1}} - \theta_{j_s}) x_{j_s}$. Hence the inequality is again satisfied. As all solutions $(x, m) \in X_K$ satisfy inequality (9), this inequality is valid for set X_K. ■

Example 2. Consider an instance with five nodes, $c = (3, 4, 4, 3, 2)$ and $B = 9$. The distance matrix is:

$i \setminus j$	**1**	**2**	**3**	**4**	**5**
1	–	8	3	9	1
2	8	–	7	1	3
3	3	7	–	6	9
4	9	1	6	–	1
5	1	3	9	1	–

In this example, $\theta = (8, 7, 7, 6, 3)$. If we take $\overline{D} = 7$, nodes 4 and 5 have θ values below the upper bound and we obtain the following inequalities (9) for different choices of S:

$$S = \{4\}, \quad m \leq 7 - (7 - 6)x_4,$$
$$S = \{4, 5\}, \quad m \leq 7 - (7 - 6)x_4 - (6 - 3)x_5,$$
$$S = \{5\}, \quad m \leq 7 - (7 - 3)x_5.$$

In case we take $\overline{D} = 6$, then node 5 is the only node with a θ value below the upper bound: inequalities (9) reduce to the inequality $m \leq 6 - (6 - 3)x_5$.

Separation: Let $(\overline{m}, \overline{x})$ be a feasible solution for the LP relaxation of $K(\overline{D}, \underline{D})$. We would like to check whether this solution satisfies inequalities (9). Let $V^< = \{j \in V : \theta_j < \overline{D}\}$. Inequalities (9) can be separated in polynomial time by solving a longest path problem in an acyclic network. For this purpose, we define a directed graph $G = (N, A)$ where

$N = \{0, n+1\} \cup \{j \in V^< : \bar{x}_j > 0\}$ and $A = \{(i,j) : i,j \in N, i < j\}$. For $(i,j) \in A$ with $j = n+1$, we set the length of the arc to 0 and for others we set the length of the arc to $(\theta_i - \theta_j)\bar{x}_j$ with $\theta_0 = \overline{D}$. A directed path from 0 to $n+1$ that visits a set of nodes $S = \{j_1, \dots, j_{|S|}\}$ can visit them in the order $0 \to j_1 \to \cdots \to j_{|S|} \to n+1$ with $j_1 < \cdots < j_{|S|}$. Such a path has length $\sum_{s=1}^{|S|}(\theta_{j_{s-1}} - \theta_{j_s})\bar{x}_{j_s}$. To minimize the right-hand side of inequality (9), we maximize $\sum_{s=1}^{|S|}(\theta_{j_{s-1}} - \theta_{j_s})\bar{x}_{j_s}$ and hence we find a longest path. The longest path problem is easy to solve since the graph is acyclic.

Let $KV(\overline{D}, \underline{D})$ be the formulation obtained after adding all inequalities (9) to $K(\overline{D}, \underline{D})$.

Proposition 7. *Let $V^{\geq} = V \setminus V^<$. If the following system*

$$\sum_{j \in V^{\geq}} c_j x'_j \geq B, \tag{10}$$

$$x'_i + x'_j \leq 1, \quad \forall i,j \in V^{\geq} : i < j, d_{ij} < \overline{D}, \tag{11}$$

$$0 \leq x'_j \leq 1, \quad \forall j \in V^{\geq} \tag{12}$$

has a solution, then the optimal value of the LP relaxation of $KV(\overline{D}, \underline{D})$ is $\overline{D}$. Otherwise this LP relaxation has an optimal value smaller than $\overline{D}$.

Proof. Suppose that the above system has a solution x^*. Let $x_j = x^*_j$ for $j \in V^{\geq}$, $x_j = 0$ for $j \in V^<$, and $m = \overline{D}$. Let $i,j \in V$ with $i < j$ and $d_{ij} < \overline{D}$. First observe that constraints (7) are satisfied since any j with $\theta_j < \underline{D} \leq \overline{D}$ is in $V^<$.

We know that $x_i + x_j \leq 1$ for $i,j \in V^{\geq}$ with $i < j$ and $d_{ij} < \overline{D}$. In addition if $i \in V^<$ then $x_i = 0$ and if $j \in V^<$ then $x_j = 0$. Thus $x_i + x_j \leq 1$ for all $i,j \in V$ with $i < j$ and $d_{ij} < \overline{D}$. Since $\underline{D} \leq \overline{D}$, constraints (8) are satisfied. Moreover, $(\overline{D} - d_{ij})(2 - x_i - x_j) \geq \overline{D} - d_{ij}$ and consequently the right-hand side of constraint (4) is at least $\overline{D}$ and the solution (m, x) satisfies constraints (4).

Since $x_j = 0$ for all $j \in V^<$, (m, x) also satisfies all inequalities (9) and is optimal for the LP relaxation of $KV(\overline{D}, \underline{D})$.

If (10)–(12) do not have a solution, then for all $(\hat{m}, \hat{x})$ that are feasible for the LP relaxation of $KV(\overline{D}, \underline{D})$, either

- there exists $i,j \in V^{\geq}$ with $i < j$, $d_{ij} < \overline{D}$ and $\hat{x}_i + \hat{x}_j > 1$ and the right-hand side of constraint (4) is less than $\overline{D}$,
- or there exists $j \in V^<$ with $\hat{x}_j > 0$ and the right-hand side of (9) is less than $\overline{D}$ for $S = \{j\}$.

In both cases, $\hat{m} < \overline{D}$. ■

Example 2 (continued). For $\overline{D} = 6$, system (10)–(12) is

$$3x'_1 + 4x'_2 + 4x'_3 + 3x'_4 \geq 9,$$
$$x'_1 + x'_4 \leq 1,$$
$$x'_2 + x'_3 \leq 1,$$
$$0 \leq x'_j \leq 1, \quad j = 1, 2, 3, 4.$$

This system has no solution. Indeed, the optimal values of the LP relaxations of $K0$, $K(9, 1)$, and $K(6, 1)$ are 10.6, 8.29, and 5.76, respectively ($D_{\max} = 9$ and $D_{\min} = 1$). The optimal solution for the LP relaxation of $K(6, 1)$ is $x = (0.07, 0.53, 1, 0.50, 0.53)$. Using $\underline{D} = 3$, we can add inequalities $x_1 + x_5 \leq 1$, $x_2 + x_4 \leq 1$ and $x_4 + x_5 \leq 1$ to this LP relaxation, obtaining $x = (0.11, 0.55, 1, 0.44, 0.55)$ with an LP bound of 5.67. Finally, we add $m + 3x_5 \leq 6$ to obtain the LP bound of 5.5.

3.2 | Edge formulation

A second formulation, proposed in [31], uses edge variables y_{ij} for all $i,j \in V$ with $i < j$ to obtain a linear model. This is the classical linearization for the product of binary variables: y_{ij} is 1 if both variables x_i and x_j are 1, and is 0 otherwise. The edge formulation is as follows:

$$\max\ m,$$

$$\text{s.t.: } y_{ij} \leq x_i, \quad \forall i,j \in V : i < j,$$

$$y_{ij} \leq x_j, \quad \forall i,j \in V : i < j,$$

$$x_i + x_j \leq y_{ij} + 1, \quad \forall i,j \in V : i < j, \tag{13}$$

$$m \leq d_{ij} + (D_{\max} - d_{ij})(1 - y_{ij}), \quad \forall i,j \in V : i < j, \tag{14}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i,j \in V : i < j, \tag{15}$$

$$x \in X. \tag{16}$$

Note that one can drop the first two sets of constraints. Let $E(D_{\max}, D_{\min})$ be this formulation and X_E be its feasible set, that is, the set of all (m, x, y) that satisfy (13)–(16). We have that $\text{Proj}_{m,x} X_E = X_K$. Hence, the edge formulation is an extended formulation for our problem.

It is also possible to relax the integrality of y variables and replace constraints (15) with $y_{ij} \geq 0$ for all $i, j \in V$ with $i < j$. Our preliminary computational results showed that this was not advantageous in terms of computation time. So we keep the y variables as binary variables in the edge formulation.

In the sequel, for any formulation F, P_F denotes the feasible set of the LP relaxation of F.

Proposition 8. $\text{Proj}_{m,x} P_{E(D_{\max},D_{\min})} = P_{K(D_{\max},D_{\min})}$.

Proof. Let $(m, x) \in \text{Proj}_{m,x} P_{E(D_{\max},D_{\min})}$. Then there exists y such that $(m, x, y) \in P_{E(D_{\max},D_{\min})}$. As all the constraints defining $P_{K(D_{\max},D_{\min})}$ except (6) are also part of the definition of $P_{E(D_{\max},D_{\min})}$, to prove that $(m, x) \in P_{K(D_{\max},D_{\min})}$, we only need to show that (m, x) satisfies (6). This is easy since for $i, j \in V$ with $i < j$, the right-hand side of constraint (6) is $d_{ij} + (D_{\max} - d_{ij})(2 - x_i - x_j) \geq d_{ij} + (D_{\max} - d_{ij})(1 - y_{ij})$ since $1 - x_i - x_j \geq -y_{ij}$ and $D_{\max} \geq d_{ij}$. As $d_{ij} + (D_{\max} - d_{ij})(1 - y_{ij}) \geq m$, constraint (6) is satisfied. So, $\text{Proj}_{m,x} P_{E(D_{\max},D_{\min})} \subseteq P_{K(D_{\max},D_{\min})}$.

Now let $(m, x) \in P_{K(D_{\max},D_{\min})}$ and define y as $y_{ij} = (x_i + x_j - 1)^+$ for all $i, j \in V$ with $i < j$. Clearly $0 \leq y_{ij} \leq 1$. If $x_i + x_j \geq 1$, then $y_{ij} = x_i + x_j - 1$ and the right-hand side of constraint (14) is $d_{ij} + (D_{\max} - d_{ij})(1 - y_{ij}) = d_{ij} + (D_{\max} - d_{ij})(2 - x_i - x_j)$. This is greater than or equal to m by (6). If $x_i + x_j < 1$, then $y_{ij} = 0$ and the right-hand side of constraint (14) is $D_{\max}$. We know that $D_{\max} \geq m$ for all $(m, x) \in P_{K(D_{\max},D_{\min})}$ since constraint (6) for the pair i, j with the largest distance becomes $m \leq d_{ij} = D_{\max}$. So, all constraints (14) are satisfied by (m, x, y) and $(m, x) \in \text{Proj}_{m,x} P_{E(D_{\max},D_{\min})}$. Hence, we proved that $P_{K(D_{\max},D_{\min})} \subseteq \text{Proj}_{m,x} P_{E(D_{\max},D_{\min})}$. ∎

As it was the case with the Kuby model, stronger models can be obtained by using better upper and lower bounds. Let $E(\overline{D}, \underline{D})$ be the model obtained by replacing constraints (14) in $E(D_{\max}, D_{\min})$ with

$$m \leq d_{ij} + (\overline{D} - d_{ij})(1 - y_{ij}), \quad \forall i, j \in V : i < j, d_{ij} < \overline{D} \tag{17}$$

and

$$m \leq \overline{D}$$

and by adding (7), (8) and $y_{ij} = 0$ for all $i, j \in V : i < j$ with $\theta_i < \underline{D}$ or $\theta_j < \underline{D}$ or $d_{ij} < \underline{D}$.

Since $\text{Proj}_{m,x} X_E = X_K$, any valid inequality for X_K is also valid for X_E. Let $EV(\overline{D}, \underline{D})$ be the formulation obtained by adding all inequalities (9) to $E(\overline{D}, \underline{D})$.

Proposition 9. $\text{Proj}_{m,x} P_{E(\overline{D},\underline{D})} = P_{K(\overline{D},\underline{D})}$ *and* $\text{Proj}_{m,x} P_{EV(\overline{D},\underline{D})} = P_{KV(\overline{D},\underline{D})}$.

Proof. Similar to the proof of Proposition 8. ∎

Additional valid inequalities for X_E can be derived using the y variables:

Proposition 10. *Let $E = \{e_1, \ldots, e_{|E|}\}$ be a subset of $V \times V$ such that $d_{e_1} > \cdots > d_{e_{|E|}}$. Let $d_{e_0} = \overline{D}$. The inequality*

$$m \leq \overline{D} - \sum_{e=1}^{|E|} (d_{e-1} - d_e) y_e \tag{18}$$

is a valid inequality for set X_E.

Proof. Similar to the proof of Proposition 6. ∎

Let $EV^+(\overline{D}, \underline{D})$ be the formulation obtained by adding all inequalities (18) to formulation $EV(\overline{D}, \underline{D})$.

Proposition 11. *The optimal values of the LP relaxations of formulations $EV(\overline{D}, \underline{D})$ and $EV^+(\overline{D}, \underline{D})$ are equal to $\overline{D}$ if and only if (10)–(12) has a solution.*

Proof. Suppose that (10)–(12) has a solution x'. As in the proof of Proposition 7, let $x_j = x'_j$ for $j \in V^{\geq}$, $x_j = 0$ for $j \in V^{<}$, $y_{ij} = (x_i + x_j - 1)^+$ for all $i, j \in V$ with $i < j$, and $m = \overline{D}$. For $i, j \in V$ with $i < j$ and $d_{ij} < \overline{D}$, we have $y_{ij} = 0$ either because $i, j \in V^{\geq}$ and $x_i + x_j \leq 1$ or at least one of i and j is in $V^{<}$ and the corresponding x variable is 0. Then it is easy to see that all constraints (17) and inequalities (18) are satisfied. From the proof of Proposition 7, we also know that all inequalities (7)–(9) are satisfied. Hence (m, x, y) is in $P_{EV(\overline{D},\underline{D})}$ and $P_{EV^+(\overline{D},\underline{D})}$ and is optimal for the LP relaxations of both $EV(\overline{D}, \underline{D})$ and $EV^+(\overline{D}, \underline{D})$.

If (10)–(12) does not have a solution, then for all (m, x, y) in $P_{EV(\overline{D},\underline{D})}$ and $P_{EV^+(\overline{D},\underline{D})}$ we are in one of the two cases: Either there exists a pair $i, j \in V^{\geq}$ with $i < j$, $d_{ij} < \overline{D}$, $x_i + x_j > 1$ and $y_{ij} > 0$. Then the right-hand side of constraint (17) is less than $\overline{D}$. Or there exists $j \in V^{<}$ with $x_j > 0$ and the right-hand side of (9) is less than $\overline{D}$ for $S = \{j\}$. In both cases, $m < \overline{D}$. ∎

3.3 | Telescopic formulation

We now model the capacitated dispersion problem by writing the objective function as a telescopic sum of terms (see [8]). Successful applications of this modeling framework in location science are, among others, those in [1, 2, 4, 5, 10, 13, 14, 16, 22, 33], which make use of the so-called *cumulative* binary variables and, depending on the work, are referred to as *covering*, *radius*, or *ordering* models.

Let $D_1 < \cdots < D_K$ represent all the distinct values in the distance matrix. Let $\mathcal{K} = \{1, \ldots, K-1\}$. We know that $K \geq 2$ and the optimal value is at most D_K. The capacitated dispersion problem can also be formulated as

$$\max D_K - \sum_{k=1}^{K-1} (D_{k+1} - D_k) u_k, \tag{19}$$

$$\begin{aligned} \text{s.t.: } & x_i + x_j \leq 1 + u_k, && \forall i, j \in V : i < j, \ \forall k \in \mathcal{K} : d_{ij} = D_k, \qquad (20) \\ & u_{k-1} \leq u_k, && \forall k \in \mathcal{K} \setminus \{1\}, \\ & u_k \in \{0, 1\}, && \forall k \in \mathcal{K}, \\ & x \in X, \end{aligned}$$

where $u_k = 0$ if the dispersion is greater than D_k, and 1 otherwise. This model has an optimal solution (x, u) in which u satisfies $u_k = 0$ for $k = 1, \ldots, k'-1$ and $u_k = 1$ for $k = k', \ldots, K-1$ for some $k' \in \{1, \ldots, K\}$, $x_i + x_j \leq 1$ for all pairs i and j such that $i < j$ and $d_{ij} < D_{k'}$, and there exists a pair $i, j \in V$ with $i < j$ such that $x_i = x_j = 1$ and $d_{ij} = D_{k'}$. The objective function value is $D_K - \sum_{k=1}^{K-1} (D_{k+1} - D_k) u_k = D_K - \sum_{k=k'}^{K-1} (D_{k+1} - D_k) = D_{k'}$.

Let $T(D_{\max}, D_{\min})$ be the model obtained by changing (19) to max m and by adding the constraint $m \leq D_K - \sum_{k=1}^{K-1} (D_{k+1} - D_k) u_k$. The above model has the same LP bound as $T(D_{\max}, D_{\min})$. The reformulation with variable m is done to enable comparison with the previous formulations. Let X_T be the feasible set of $T(D_{\max}, D_{\min})$. Again we have $\text{Proj}_{m,x} X_T = X_K$. So the telescopic formulation is also an extended formulation.

As before, stronger formulations can be obtained by using better upper and lower bounds. Let $T(\overline{D}, \underline{D})$ be the formulation obtained after fixing $u_k = 1$ for all k with $D_k \geq \overline{D}$, and adding $u_k = 0$ for all k with $D_k < \underline{D}$ and $x_j = 0$ for all $j \in V$ with $\theta_j < \underline{D}$ to formulation $T(D_{\max}, D_{\min})$.

Proposition 12. $\text{Proj}_{m,x} P_{T(\overline{D},\underline{D})} \subseteq P_{K(\overline{D},\underline{D})}$.

Proof. Let $(m, x) \in \text{Proj}_{m,x} P_{T(\overline{D},\underline{D})}$. Then there exists u such that $(m, x, u) \in T(\overline{D}, \underline{D})$. If $\overline{D} = D_{K'}$, then after fixing $u_k = 1$ for all $k = K', \ldots, K-1$, we get $D_K - \sum_{k=1}^{K-1} (D_{k+1} - D_k) u_k = \overline{D} - \sum_{k=1}^{K'-1} (D_{k+1} - D_k) u_k$. Since $\overline{D} - \sum_{k=1}^{K'-1} (D_{k+1} - D_k) u_k \leq \overline{D}$ we have $m \leq \overline{D}$. Let $i, j \in V$ with $i < j$ and $d_{ij} < \overline{D}$ and k' be the index such that $d_{ij} = D_{k'}$. Then

$$\begin{aligned} \overline{D} - \sum_{k=1}^{K'-1} (D_{k+1} - D_k) u_k &\leq \overline{D} - \sum_{k=k'}^{K'-1} (D_{k+1} - D_k) u_k \\ &\leq \overline{D} - \sum_{k=k'}^{K'-1} (D_{k+1} - D_k) u_{K'}, \end{aligned}$$

where the first inequality holds since $u_k \geq 0$ and $D_{k+1} - D_k \geq 0$ for all $k = 1, \ldots, k'-1$ and the last inequality holds since $u_k \geq u_{k'}$ and $D_{k+1} - D_k \geq 0$ for all $k = k', \ldots, K'-1$. Now $\overline{D} - \sum_{k=k'}^{K'-1} (D_{k+1} - D_k) u_{K'} = \overline{D} - (\overline{D} - D_{K'}) u_{K'}$. As $u_{k'} \geq x_i + x_j - 1$, we have $\overline{D} - (\overline{D} - D_{k'}) u_{k'} \leq \overline{D} - (\overline{D} - D_{k'})(x_i + x_j - 1)$. Using $D_{k'} = d_{ij}$, the latter is equal to $d_{ij} + (\overline{D} - d_{ij}) + (\overline{D} - d_{ij})(1 - x_i - x_j) = d_{ij} + (\overline{D} - d_{ij})(2 - x_i - x_j)$. Hence $m \leq d_{ij} + (\overline{D} - d_{ij})(2 - x_i - x_j)$ and (m, x) satisfies constraint (4).

Finally, observe that for $i, j \in V$ such that $i < j$ and $d_{ij} < \underline{D}$, if $d_{ij} = D_k$ then $u_k = 0$ together with (20) gives $x_i + x_j \leq 1$. ∎

In general, $\text{Proj}_{m,x}P_{T(\overline{D},\underline{D})} \neq P_{K(\overline{D},\underline{D})}$. We show this with a small example.

Example 3. Consider the following instance with three nodes, $c = \{10, 10, 10\}$, $B = 18$ and the following distance matrix:

$i \backslash j$	1	2	3
1	–	2	1
2	2	–	3
3	1	3	–

Solution $(m, x) = (2.8, 0.5, 0.7, 0.6)$ is in $P_{K(3,1)}$ although there is no u such that (m, x, u) is in $\text{Proj}_{m,x}P_{T(3,1)}$. It is in $P_{K(3,1)}$ because $d_{ij} + (D_{\max} - d_{ij})(2 - x_i - x_j) \in \{2.8, 3\}$ and $10 \times 0.5 + 10 \times 0.7 + 10 \times 0.6 = 18$. If u exists, it must satisfy $u_1 \geq 0.5 + 0.6 - 1 = 0.1$, $u_2 \geq 0.5 + 0.7 - 1 = 0.2$ and $u_1 \leq u_2$. Thus,

$$D_K - \sum_{k=1}^{K-1}(D_{k+1} - D_k)u_k = 3 - u_1 - u_2 \leq 3 - 0.1 - 0.2 = 2.7 < m.$$

Proposition 13. *For $j \in V$, let k_j be the index such that $D_{k_j} = \theta_j$. Then, inequality*

$$x_j \leq u_{k_j} \tag{21}$$

is valid for X_T.

Proof. Follows from the definition of θ_j. ■

Let $TV(\overline{D}, \underline{D})$ be the formulation obtained by adding inequalities (21) to $T(\overline{D}, \underline{D})$.

Proposition 14. $\text{Proj}_{m,x}P_{TV(\overline{D},\underline{D})} \subseteq P_{KV(\overline{D},\underline{D})}$.

Proof. Let $(m, x) \in \text{Proj}_{m,x}P_{TV(\overline{D},\underline{D})}$ and u be such that $(m, x, u) \in P_{TV(\overline{D},\underline{D})}$. We know that $(m, x) \in P_{K(\overline{D},\underline{D})}$. To prove that $(m, x) \in P_{KV(\overline{D},\underline{D})}$, we need to show that (m, x) satisfies inequalities (9). Let $S = \{j_1, \ldots, j_{|S|}\} \subseteq \{j \in V : \theta_j < \overline{D}\}$. Without loss of generality, we can assume that $\overline{D} = \theta_{j_0} > \cdots > \theta_{j_{|S|}}$. For each $j_s \in S$, θ_{j_s} is equal to some distance value not larger than $\overline{D}$. Let k_s be the index with $\theta_{j_s} = D_{k_s}$. Also let $k_0 = K'$. Then, $\overline{D} - \sum_{s=1}^{|S|}(\theta_{j_{s-1}} - \theta_{j_s})x_{j_s} = \overline{D} - \sum_{s=1}^{|S|}(D_{k_{s-1}} - D_{k_s})x_{j_s}$.
Since $x_{j_s} \leq u_{k_s}$, we have $\overline{D} - \sum_{s=1}^{|S|}(D_{k_{s-1}} - D_{k_s})x_{j_s} \geq \overline{D} - \sum_{s=1}^{|S|}(D_{k_{s-1}} - D_{k_s})u_{k_s}$. Observe that $k_0 > \cdots > k_{|S|}$. For $s \in \{1, \ldots, |S|\}$, we also have $(D_{k_{s-1}} - D_{k_s})u_{k_s} = \sum_{\kappa=k_s}^{k_{s-1}-1}(D_{\kappa+1} - D_\kappa)u_{k_s} \leq \sum_{\kappa=k_s}^{k_{s-1}-1}(D_{\kappa+1} - D_\kappa)u_\kappa$ since $u_{k_s} \leq u_\kappa$ for all $\kappa = k_s, \ldots, k_{s-1} - 1$. Hence, $\overline{D} - \sum_{s=1}^{|S|}(D_{k_s+1} - D_{k_s})u_{k_s} \geq \overline{D} - \sum_{s=1}^{|S|}\sum_{\kappa=k_s}^{k_{s-1}-1}(D_{\kappa+1} - D_\kappa)u_\kappa = \overline{D} - \sum_{k=k_{|S|}}^{K'-1}(D_{k+1} - D_k)u_k \geq \overline{D} - \sum_{k=1}^{K'-1}(D_{k+1} - D_k)u_k$. Now, since $m \leq \overline{D} - \sum_{k=1}^{K'-1}(D_{k+1} - D_k)u_k$ and $\overline{D} - \sum_{k=1}^{K'-1}(D_{k+1} - D_k)u_k \leq \overline{D} - \sum_{s=1}^{|S|}(\theta_{j_{s-1}} - \theta_{j_s})x_{j_s}$, inequality (9) for set S is satisfied. Hence $(m, x) \in P_{KV(\overline{D},\underline{D})}$. ■

Figure 1 summarizes the information in Propositions 8,9,12, and 14. All the feasible sets refer to $(\overline{D}, \underline{D})$. The figure states the inclusion relationships between the six feasible sets P_K, $\text{Proj}_{m,x}P_E$, $\text{Proj}_{m,x}P_T$, P_{KV}, $\text{Proj}_{mx}P_{EV}$, and $\text{Proj}_{m,x}P_{TV}$.

Next, we generalize the idea of inequalities (21).

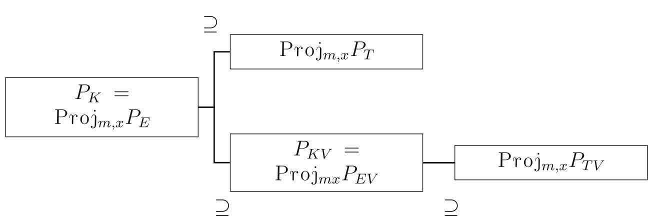


FIGURE 1 Inclusion relationships between feasible sets

Proposition 15. *Let $C \subset V$ with $|C| \geq 2$ and compute k_C as follows. Let l' be the index with $D_{l'} = \min_{i,j \in C : i<j} d_{ij}$.*

- *If $\sum_{j \in C} c_j \geq B$, then $k_C = l'$.*
- *If $\sum_{j \in C} c_j < B$, then let l'' be the index with*

$$D_{l''} = \max_{V' \subseteq V \setminus C : \sum_{\iota \in V' \cup C} c_\iota \geq B} \min_{i \in V'} \delta_i,$$

where $\delta_i = \min_{j \in C} d_{ij}$ *for* $i \in V \setminus C$. *Set* $k_C = \min\{l', l''\}$.

Inequality

$$\sum_{j \in C} x_j \leq \sum_{j \in C} u_{k_j} + u_{k_C} - u_{k'_C}, \tag{22}$$

where $k'_C = \min_{j \in C} k_j$, *is valid for* X_T.

Proof. First observe that if $x_j = 1$ for all $j \in C$, then D_{k_C} is an upper bound on the optimal value and so $u_{k_C} = 1$. As we also have $u_{k_j} = 1$ for all $j \in C$, both sides of the inequality are equal to $|C|$. If $\sum_{j \in C} x_j \leq |C| - 1$, then if $u_{k_C} - u_{k'_C} \geq 0$ the inequality is satisfied since $x_j \leq u_{k_j}$ is satisfied for all $j \in C$. The remaining case is when $\sum_{j \in C} x_j \leq |C| - 1$ and $u_{k_C} - u_{k'_C} = -1$. In this case $u_{k_C} = 0$ and $u_{k'_C} = 1$. When $u_{k'_C} = 1$, $u_{k_j} = 1$ for all $j \in C$ since $k'_C = \min_{j \in C} k_j$. Hence, the right-hand side is equal to $|C| - 1$ and the inequality is again satisfied. ∎

As we did in computing θ_j's, here we sort the nodes in set $V \setminus C$ in nonincreasing order of δ_i's. Let π be the order, that is, $\delta_{\pi(1)} \geq \cdots \geq \delta_{\pi(|V \setminus C|)}$ and l be the smallest integer such that $\sum_{i=1}^{l} c_{\pi(i)} \geq B - \sum_{j \in C} c_j$. Then, k_C is the index with $D_{k_C} = \delta_{\pi(l)}$.

Remark 3. Inequality (22) with $k_C = l'$ is also valid.

Remark 4. For $C \subset V$ with $|C| \geq 2$, inequality (22) is implied by the sum of inequalities $x_j \leq u_{k_j}$ over $j \in C$ unless $k_C < k'_C$.

Our preliminary experiments showed that inequalities (22) for subsets C with $|C| \geq 3$ are not useful in reducing the computation times. So for further comparison, we let $TV^+(\overline{D}, \underline{D})$ be formulation $TV(\overline{D}, \underline{D})$ plus inequalities (22) for subsets C with $|C| = 2$.

Remark 5. The integrality of the u_k variables can be relaxed.

4 | COMPUTATIONAL EXPERIMENTS

In this section, we report some computational experiments performed to complement the contribution of the previous sections.

We start by providing details about the experimental setting. Then, we show results of a comprehensive numerical study performed using a subset of instances, aiming at investigating the computational efficacy of each formulation as well as their potential applications. We present later several solution procedures for taking advantage of the polyhedral study performed. We finish the section by extending the tests to larger instances, where we compare with the state-of-the-art algorithms devised for this problem [31], and we show the advantages of using the procedures proposed.

4.1 | Experimental design: Technology employed and test instances

Our computational experiments have been performed on a machine with an Intel i5-8500 processor, 3.6 GHz, and 64 GB of RAM, running Ubuntu 20.04.2-64 bits Linux operating system. The procedures have been implemented in the `C` language, making use of the IBM ILOG CPLEX solver version 12.10 through its Callable Library, and compiled with the `gcc` compiler version 7.5 using optimization flag `-O3`. This solver has been used with the exploitation of multithreading capabilities, that is, regardless of the solution method, all cores (six in our case) are available for use. Other than this, the default parameters are assumed unless otherwise stated.

Benchmark instances of diversity problems can be downloaded from the MDPLIB public domain [23]. Several datasets previously employed in different studies on many variants of such problems can be found on this website. Among them, we consider the following three to be of special relevance: the data proposed in [18] where the cost values were generated as the Euclidean distance from randomly generated points; the dataset generated in [9] in which the matrices were randomly generated following a uniform distribution either in the interval [0, 10] or [0, 1000], depending on the instance; and the SOM dataset, proposed in [24], where the distance matrices were generated with random numbers between 0 and 9 from an integer uniform distribution. These sets were reviewed and adapted to the capacitated version of the dispersion problem in [31] by randomly generating a capacity value in the range [1, 1000] for each node. To generate B, the authors computed the sum of all nodes' capacities and set B as this sum multiplied by a factor in the set $\{0.2, 0.3\}$; thus two different B values were created for each instance data.

On the other hand, [14] generated instances for the maximum dispersion problem. Among them, we focus on those in the *WEEE* set in which the coordinates of nodes are generated uniformly from the square $Q = ((0,0),(10,10))$ and the distances between nodes are computed using the ℓ_2-metric.

Following these papers, as well as the references therein, we motivate three major ways of creating distance matrices for the capacitated dispersion problem:

1. Euclidean distances between randomly generated points in $\mathbb{R}^2$.
2. Data randomly generated following a uniform distribution in a large given interval.
3. Data randomly generated following a uniform distribution for which the possible integer distance values are chosen from a small range.

Accordingly, we have generated two sets of instances that we have used in different experiments. The first set, which we named the *testing* set, consists of a collection of medium-size instances that we use for carrying out an in-depth analysis of the formulations proposed along the manuscript. The second, the *competitive* set, consists of larger-size instances that turn out to be more challenging and that can be used for competitive comparisons among the most prominent formulations identified in previous experiments. In all instances of both sets, each B value has been determined as the sum of the nodes' capacities multiplied by a factor as in [31]. To avoid instance duplicity, we note that when generating each instance data, the random seed has been set to a different value. The particular characteristics of the two sets follow.

- Testing set, consisting of three subsets of instances of different natures:
  - Euclidean type: this subset, labeled in the tables as "Euclid.," involves 15 instances in total, with $n = \{100,110,120\}$. The capacity value for each node is randomly obtained in the integer interval $[1,1000]$ following a uniform distribution.
  - Uniform type with large range of possible values (labeled "Uni_LR"): this set also involves 15 instances in total, with $n = \{120,125,140\}$. The distance value for each pair of nodes is uniformly generated in the integer range $[100,1000]$. The capacity value for each node is again randomly obtained in the integer interval $[1,1000]$ following a uniform distribution.
  - Uniform type with small range of possible values (labeled "Uni_SR"): this set involves seven instances in total, with $n = \{200,250\}$. This time, the distance value for each pair of nodes is uniformly generated in the integer range $[1,10]$, and the capacity value for each node is randomly obtained in the integer interval $[100,3000]$ following a uniform distribution.
- Competitive set, consisting of two types of instances:
  - Euclidean type: this set involves 50 instances in total, with $n = \{150,175,200,300\}$. The capacity value for each node is randomly obtained in the integer interval $[1000,4000]$ following a uniform distribution. Moreover, following the idea in [14], for each node the coordinates are generated uniformly distributed in the square $((0,0),(10,10))$.
  - Uniform with small range of values directly taken from the SOM dataset originally proposed in [24]. It consists of 11 large-size instances generated in [31], whose sizes are in the set $\{300,400,500\}$. As mentioned before, the distance matrices of the SOM set are generated with random numbers between 0 and 9 from an integer uniform distribution, and the capacity value for each node is randomly obtained in the integer interval $[1,1000]$ following a uniform distribution.

The instances from these sets can be downloaded from the following link.[1]

4.2 | Computational experiments

We show now a set of results of a comprehensive numerical study performed aiming at investigating the computational efficacy of each formulation and several designs of solution procedures.

4.2.1 | Comparison of the LP bounds

Using the testing set of instances, we perform a comparison of the LP bounds of our formulations. Table 1 summarizes this comparison in which the first four columns refer to the nature of the data, the instance identifiers, the number of nodes and the optimal values. Three sets of columns follow, one for each formulation type. For each of these sets, a column is devoted to a particular configuration of the formulation. We compute the upper bound $\overline{D}$ using Remark 1 and Proposition 3 and the lower bound $\underline{D}$ using Remark 2. We consider formulations $K0$, $K1 = K(D_{\max}, D_{\min})$, $K2 = K(\overline{D}, D_{\min})$, $K3 = K(\overline{D}, \underline{D})$ and $K4 = KV(\overline{D}, \underline{D})$ for those of the Kuby family, $E5 = EV^{+}(\overline{D}, \underline{D})$ for the edge family, and $T4 = TV(\overline{D}, \underline{D})$ and $T5 = TV^{+}(\overline{D}, \underline{D})$

[1] https://www.dropbox.com/sh/fk8rie01339ehme/AADLO4WlO1nobBTVP0VRC64Ja?dl=0

TABLE 1 Upper bounds on the instances of the testing set with each formulation

Family	Inst.	n	Opt. val	$K0$	$K1$	$K2$	$K3$	$K4$	$E5$	$T4$	$T5$
Euclid.	e01	100	269.8	1994.5	1196.3	829.7	829.7	829.7	829.7	801.8	774.5
	e02	100	256.7	2083.0	1254.0	845.9	845.9	818.7	818.5	817.6	789.5
	e03	100	256.1	2182.7	1319.2	869.7	869.7	869.7	869.7	842.8	816.7
	e04	100	256.0	2047.2	1235.0	864.2	864.2	864.2	864.2	836.6	809.2
	e05	100	253.6	2080.9	1256.0	847.8	847.8	824.1	824.0	823.2	798.9
	e06	120	224.6	2263.4	1377.2	891.6	891.6	864.6	864.5	863.6	836.1
	e07	110	264.5	2164.1	1304.5	858.7	858.7	858.7	858.7	830.2	802.5
	e08	120	233.1	2110.1	1281.2	803.5	803.5	775.4	775.1	774.0	745.0
	e09	110	260.6	2017.6	1217.4	889.2	889.2	889.2	889.2	867.6	846.6
	e10	110	233.4	2087.3	1262.8	874.9	874.9	874.9	874.9	850.1	826.6
	e11	100	248.5	2159.0	1307.3	888.6	888.6	860.6	860.4	859.5	830.8
	e12	100	235.8	2169.2	1310.7	817.1	817.1	817.1	817.1	785.6	757.7
	e13	100	264.6	2146.9	1295.0	864.1	864.1	838.8	838.6	837.1	810.9
	e14	100	283.3	2160.3	1299.0	779.9	779.9	779.9	779.9	750.3	721.6
	e15	100	275.7	2158.9	1301.6	869.4	869.4	869.4	869.4	841.6	814.5
Uni_LR	u01	120	297	1714.9	1000.0	856.0	856.0	856.0	856.0	846.4	837.1
	u02	120	292	1716.7	1000.0	853.0	853.0	853.0	853.0	845.5	838.2
	u03	120	284	1712.8	1000.0	851.0	851.0	851.0	851.0	842.0	833.1
	u04	120	297	1712.5	1000.0	865.0	865.0	865.0	865.0	857.4	849.8
	u05	120	320	1716.1	1000.0	853.0	853.0	844.7	844.7	844.6	836.2
	u06	125	285	1714.9	1000.0	856.0	856.0	848.7	848.7	848.6	841.3
	u07	125	297	1714.5	1000.0	847.0	847.0	847.0	847.0	842.5	838.1
	u08	125	289	1716.0	1000.0	852.0	852.0	852.0	852.0	844.1	836.4
	u09	125	294	1714.7	1000.0	845.0	845.0	845.0	845.0	837.9	830.8
	u10	125	287	1714.1	1000.0	843.0	843.0	835.5	835.5	835.4	827.8
	u11	140	270	1711.3	1000.0	845.0	845.0	845.0	845.0	838.1	831.4
	u12	140	278	1712.3	1000.0	863.0	863.0	854.2	854.2	854.1	845.3
	u13	140	252	1712.5	1000.0	856.0	856.0	846.6	846.6	846.5	837.4
	u14	140	286	1714.4	1000.0	860.0	860.0	853.0	853.0	852.9	845.8
	u15	140	261	1713.6	1000.0	846.0	846.0	838.9	838.9	838.9	831.8
	u_b_1	200	2	17	10	9	9	9	9	9	9
	u_b_2	200	2	17	10	9	9	9	9	9	9
	u_b_3	200	2	17	10	9	9	9	9	9	9
Uni_SR	u_b_4	200	2	17	10	9	9	9	9	9	9
	u_b_5	250	2	17	10	9	9	9	9	9	9
	u_b_6	250	2	17	10	9	9	9	9	9	9
	u_b_7	250	2	17	10	9	9	9	9	9	9

for the telescopic family. Note that results for $E1 = E(D_{\max}, D_{\min})$, $E2 = E(\overline{D}, D_{\min})$, $E3 = E(\overline{D}, \underline{D})$ and $E4 = EV(\overline{D}, \underline{D})$ are not reported since, according to Section 3, they are the same as those of $K1$, $K2$, $K3$, and $K4$, respectively. We have also omitted the results regarding $T1 = T(D_{\max}, D_{\min})$, $T2 = T(\overline{D}, D_{\min})$, $T3 = T(\overline{D}, \underline{D})$ as, in this experiment, their bounds are, respectively, equal to those of $K1$, $K2$, and $K3$.

Several interesting insights arise from Table 1. Regarding the Kuby formulation, the improvements obtained from $K0$ to $K1$, and also from $K1$ to $K2$, are quite abrupt in the three families of instances we are solving. However, we do not find any differences between $K2$ and $K3$. Nonetheless, we obtain again better bounds with $K4$ for those instances for which violated inequalities (9) were found (see, e.g., the results corresponding to e02, e05, u05, u06). As for the edge formulation, configuration $E5$ is able to slightly improve the bounds compared to the results for $K4$ in those instances where violated cuts are found. Concerning the results of the telescopic formulation, it is interesting to see that $T4$, when compared to $E5$, is able to improve all the bounds of the testing set except those of the Uni_SR subset. It can also be seen that $T5$ obtains the best upper bounds when compared to all other configurations in this table.

We also note that when solve the instances in the testing set, the average number of added inequalities (8) used in formulation $K3$ has been 14.6, and 141.3 and 0, respectively for the three families of instances. None of the inequalities (7) has been found in this set. Detailed information about these facts is provided in Table A1 of the Appendix.

4.2.2 | Two solution procedures

In order to solve instances of the capacitated dispersion problem we have devised two procedures. The first one, **P1**, consists of a straightforward application of the CPLEX solver to the different configurations of the three families of formulations. Recall that inequalities (9) and (18) are exponential in number, thus we need a row generation approach for solving models $K4$, $E4$ and $E5$. Our preliminary computational results showed that solving these models using a cut-and-branch technique led to poor computational performance. For this reason, we devised a second approach, **P2**, to take advantage of these models, in which we start by computing UB as explained in Remark 1 and Proposition 3, and by setting $\overline{D} = UB$. Then, we check if system (10)–(12) has a solution. By the application of Proposition 7, if none exists, we solve the LP relaxation of the models by separating inequalities exactly using the polynomial separation methods presented before. We repeat this process until such a system has a solution and, then, we use the MILP to solve either model $K3$ or model $E3$ with the best $\overline{D}$ found so far. A sketch of this second procedure is depicted in Algorithm 1.

Algorithm 1. Procedure **P2**

Compute UB as explained in Remark 1 and Proposition 3. Set $\bar{D} := UB$.
while System (10)–(12) with $\bar{D}$ has no solution **do**
  if We are solving $K4$ (resp. $E4$) **then**
    Separate inequalities (9) exactly.
    Solve the LP relaxation of $K4$ (resp. $E4$) including the most violated inequality that was separated. Let z^* be the optimal value of this LP relaxation.
  if We are solving $E5$ **then**
    Separate inequalities (9) and (18) exactly.
    Solve the LP relaxation of $E5$ including the most violated inequality obtained from each separation procedure. Let z^* be the optimal value of this LP relaxation.
  Set $\bar{D} := z^*$.
Solve $K3$ or $E3$ with $\bar{D}$ using the MILP solver.

We performed a comparison of these procedures with the aim of investigating their computational efficacy. We did so by registering, for each formulation and instance of the testing set, the CPU time needed, measured in seconds, for solving the given instance following the procedures described above. Table 2 summarizes this comparison. In this table, we have divided the information in three sets of columns, one for each formulation type. For each of these sets, a column is devoted for a particular configuration of the formulation: **P1** is applied to $K0, \dots, K3, E1, \dots, E3, T1, \dots, T5$; **P2** is applied with $K4, E4, E5$. Several interesting insights arise from this table, all of them regarding the time needed to solve the integer problems to optimality:

- Euclidean instances:
  - In general terms, the Kuby formulations perform better than the edge and telescopic formulations. This difference is significant, allowing us to discard as competitive options the edge formulations for solving these type of instances.
  - By specifically looking at the Kuby columns, we can see that the computing times get smaller with the successive refinements proposed for this formulation. This is especially remarkable when we compare these results to $K0$ (the direct adaptation of the Kuby model). Comparing the **P1** with $K3$ and **P2** with $K4$, we do not appreciate a clear winner. Despite this fact, we select **P2** with $K4$ as our preferred Kuby method for Euclidean instances since it provides a relaxation bound at least as good as $K3$.
- Uniform instances:
  - For the case of instances with large range distance values, we see that $K3$ and $T3$ both using **P1** obtain short computing times. These two options are significantly better than their corresponding related formulations, especially in the case of $T3$ that clearly outperforms $T4$ and $T5$.
  - The telescopic formulation clearly outperforms the two other formulations in the case of uniform instances with small range of possible distance values. Among the telescopic options, $T2$ and $T3$ seem to be the best options.

TABLE 2 CPU time (in seconds) for solving the instances of the testing set with each procedure and model

		Kuby formulation					Edge formulation					Telescopic formulation				
		P1	P1	P1	P1	P2	P1	P1	P1	P2	P2	P1	P1	P1	P1	P1
Family	Inst.	*K*0	*K*1	*K*2	*K*3	*K*4	*E*1	*E*2	*E*3	*E*4	*E*5	*T*1	*T*2	*T*3	*T*4	*T*5
Euclid.	e01	50	5	1	1	1	8	7	6	6	6	25	76	99	90	149
	e02	4	2	6	4	4	48	8	6	9	67	100	238	20	133	557
	e03	149	7	4	5	3	9	8	19	19	18	25	133	111	209	181
	e04	18	2	3	3	3	8	7	8	8	9	119	1599	129	231	279
	e05	46	4	3	2	8	9	8	7	10	24	28	131	64	108	164
	e06	7200	82	23	4	12	22	13	11	15	41	7200	4389	1463	1187	2112
	e07	209	5	3	3	3	14	7	9	9	9	107	109	109	222	162
	e08	171	18	4	5	8	17	20	12	20	36	657	263	190	584	662
	e09	721	4	3	9	3	18	34	9	9	9	2819	1281	102	584	734
	e10	1093	4	6	10	5	46	14	43	44	44	322	306	223	481	814
	e11	27	2	6	2	4	9	7	59	17	32	2697	193	173	156	939
	e12	443	5	2	2	2	13	10	7	7	7	213	889	139	161	361
	e13	31	5	4	2	5	10	8	8	11	42	640	109	106	58	184
	e14	15	3	3	2	3	9	37	6	6	7	82	234	149	58	83
	e15	70	5	11	2	11	8	7	7	7	7	524	96	75	233	96
Uni_LR	u01	11	17	77	23	77	90	44	61	62	62	42	67	38	143	340
	u02	111	101	53	15	53	190	102	68	69	69	59	45	40	277	443
	u03	14	15	101	9	101	138	165	87	88	88	47	30	52	174	428
	u04	62	42	24	18	23	106	108	45	45	46	49	46	38	163	234
	u05	63	12	43	27	26	53	39	44	49	64	47	63	25	81	158
	u06	378	172	86	49	19	263	111	91	150	162	77	37	59	264	520
	u07	420	65	20	39	20	221	185	124	125	126	75	42	41	714	265
	u08	202	17	44	19	43	145	99	82	82	82	47	35	38	273	329
	u09	424	76	171	45	171	133	136	88	89	89	72	44	48	95	335
	u10	55	318	178	30	26	190	128	81	118	132	65	53	39	399	312
	u11	962	1653	1387	30	1388	913	887	911	911	914	218	175	164	1759	1839
	u12	91	711	261	36	87	1042	164	390	319	306	85	147	84	1394	2283
	u13	166	85	46	33	97	2738	1266	868	1166	918	149	193	191	1045	4722
	u14	69	68	259	85	344	512	246	249	402	430	92	106	129	685	1714
	u15	125	139	37	196	80	892	967	852	791	804	138	194	174	1869	2746
	u_b_1	184	119	131	131	131	448	293	285	285	286	47	36	36	45	354
	u_b_2	189	132	100	100	101	1209	1237	1177	1180	1185	163	98	98	88	220
	u_b_3	282	24	131	130	130	516	291	274	274	274	46	46	47	43	782
Uni_SR	u_b_4	170	138	131	125	127	362	309	288	288	292	137	36	36	35	552
	u_b_5	599	533	346	346	346	4698	3921	3751	3753	3760	95	93	95	503	877
	u_b_6	748	336	463	462	464	7211	7210	7211	7210	7211	628	356	357	634	2029
	u_b_7	1196	527	426	426	426	7212	7211	7211	7210	7211	422	66	68	297	1074

4.2.3 | A result that leads to other solution procedures

Remark 6. If the optimal value of an LP relaxation, say z^*, is less than $\overline{D}$, we can update the latter to the largest distance value not greater than z^*.

Example 2 (continued)**.** In this example, the different distance values are $\{1, 3, 6, 7, 8, 9\}$. By the end of Section 3.2, we know that $\overline{D} = 6$ and $z^* = 5.5 < \overline{D}$. We can update $\overline{D}$ to 3 since we know that the closest distance value not greater than 5.5 is a valid upper bound.

Remark 6 inspires us to design other solution procedures for the CDP. In particular, a third procedure, **P3**, can be obtained with a minor modification of **P2**. Its difference lies in the application of Remark 6 after step 9 of Algorithm 1. As in the case of **P2**, **P3** can be applied with $K4$, $E4$, and $E5$. Our last proposal, procedure **P4**, uses the telescopic formulation: new $\overline{D}$ values

TABLE 3 Insights about the separation procedures in **P2** and **P3**

		P2 with $K4$			P3 with $K4$			P2 with $E5$				P3 with $E5$			
		$\overline{D}$	t.	Ineq.	$\overline{D}$	t.	Ineq.	$\overline{D}$	t.	Ineq.	Ineq.	$\overline{D}$	t.	Ineq.	Ineq.
Family	**Inst.**			(9)			(9)			(9)	(18)			(9)	(18)
Euclid.	e01	829.7	0	0	829.7	0	0	829.7	0	0	0	829.7	0	0	0
	e02	787.5	1	153	787.5	1	111	771.2	65	821	813	771.2	60	388	442
	e03	869.7	0	0	869.7	0	0	869.7	0	0	0	869.7	0	0	0
	e04	864.2	0	0	864.2	0	0	864.2	0	0	0	864.2	0	0	0
	e05	792.4	2	335	792.4	1	140	785.5	13	208	245	785.5	17	144	229
	e06	824.7	2	190	824.7	2	160	819.2	24	200	182	819.2	29	190	227
	e07	858.7	0	0	858.7	0	0	858.7	0	0	0	858.7	0	0	0
	e08	732.7	3	230	732.7	3	177	730.5	20	217	184	730.5	22	181	239
	e09	889.2	0	0	889.2	0	0	889.2	0	0	0	889.2	0	0	0
	e10	874.9	0	0	874.9	0	0	874.9	0	0	0	874.9	0	0	0
	e11	828.5	1	156	828.5	2	120	820.2	10	223	215	820.2	26	173	224
	e12	817.1	0	0	817.1	0	0	817.1	0	0	0	817.1	0	0	0
	e13	817.4	1	132	817.4	1	108	796.5	22	529	526	796.5	36	245	342
	e14	779.9	0	0	779.9	0	0	779.9	0	0	0	779.9	0	0	0
	e15	869.4	0	0	869.4	0	0	869.4	0	0	0	869.4	0	0	0
Uni_LR	u01	856	0	0	856	0	0	856	0	0	0	856	0	0	0
	u02	853	0	0	853	0	0	853	0	0	0	853	0	0	0
	u03	851	0	0	851	0	0	851	0	0	0	851	0	0	0
	u04	865	0	0	865	0	0	865	0	0	0	865	0	0	0
	u05	833	4	389	833	2	115	833	29	231	199	833	23	126	199
	u06	838	37	2887	838	4	180	838	30	377	333	838	21	154	131
	u07	847	0	0	847	0	0	847	0	0	0	847	0	0	0
	u08	852	0	0	852	0	0	852	0	0	0	852	0	0	0
	u09	845	0	0	845	0	0	845	0	0	0	845	0	0	0
	u10	824	3	248	824	3	128	824	25	228	186	824	20	144	174
	u11	845	0	0	845	0	0	845	0	0	0	845	0	0	0
	u12	840	5	393	840	5	145	839	63	425	385	839	27	162	177
	u13	831	20	1424	831	5	171	829	147	627	591	829	42	219	328
	u14	842	6	409	842	3	151	841	59	413	380	841	29	169	224
	u15	829	3	204	829	3	110	829	142	203	324	829	23	135	171

(after application of Remark 6) may lead to new u-variables that can be fixed to 1, so we can iterate over a **while** loop combining these two remarks until no new $\overline{D}$ is reached. After this loop, we can solve any integer telescopic formulation using the best $\overline{D}$ value found so far. Hence, it can be applied with $T1, \ldots, T5$. We compare in Table 3 some information about the longest-paths separation routines embedded in procedures **P2** and **P3**.

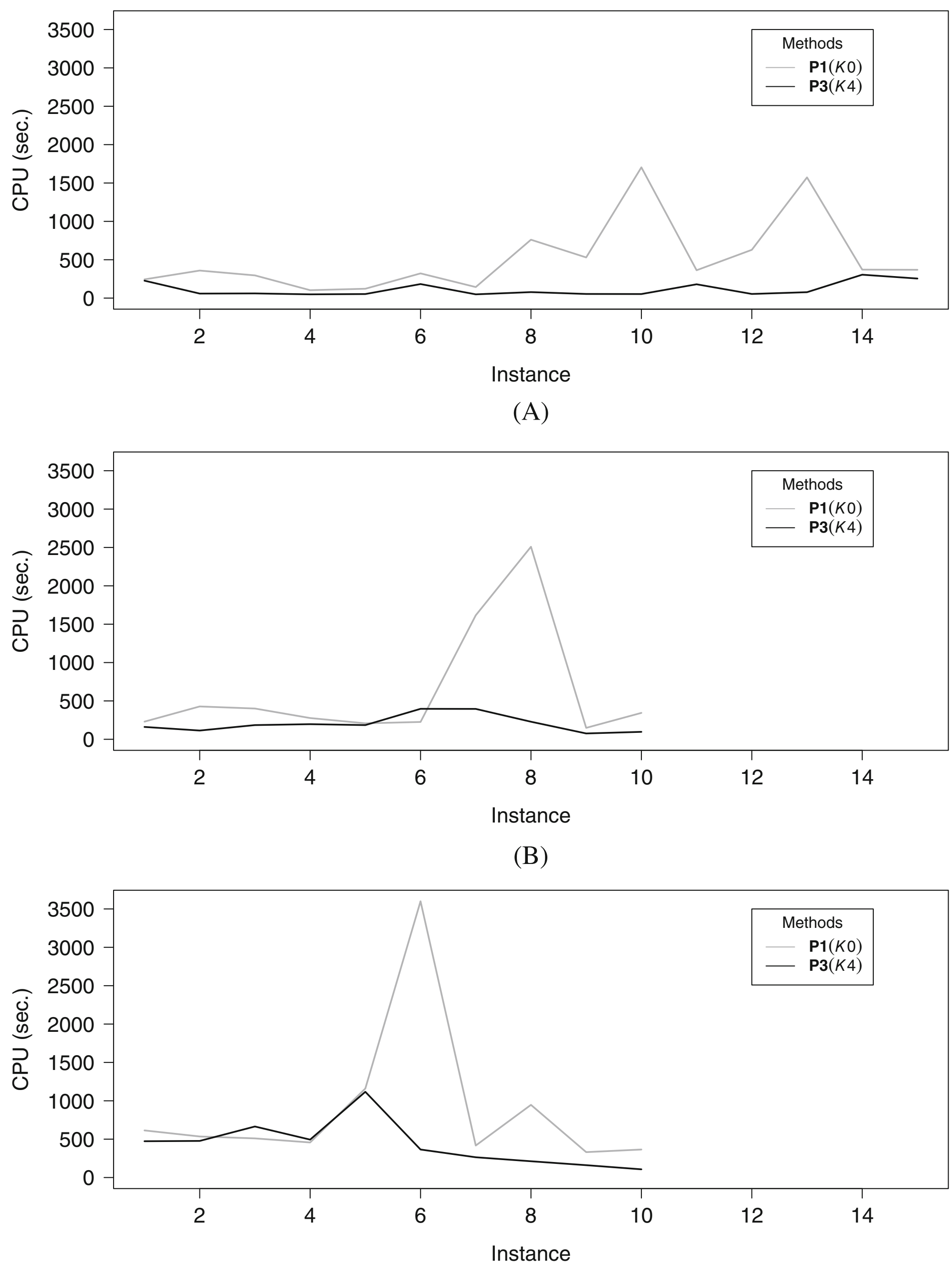
In particular, this table reports three types of measures: the best upper bound obtained (denoted by $\overline{D}$), the total computing time (denoted by "t.") after the overall separation procedures, and the number of violated cuts added from the two families of inequalities. From this table some observations arise. First, we observe that the separation procedures of inequalities (9) are performed in negligible computing time, despite the fact that the number of iterations in the procedure is significant, as reflected in column ineq. (9). The results of formulation $E5$ are very interesting since they show a significant increase—most of the cases, an order of magnitude—in the time spent by the separation routines. This could be due to the increment in size of the separation problem, as the number of variables in the edge formulation is an order of magnitude larger than the one in the Kuby formulation. Second, we see that the difference in the application of **P2** and **P3** to a given formulation affects only the number of times the separation procedures are applied. Generally, **P3** results in shorter separation times with fewer iterations (violated cuts) required. Notice that no effect is observed on the final value of $\overline{D}$. Third, we point out that in this table no information about the uniform instances with short range is displayed since no violated cuts were found for this subset of instances.

We perform a last experiment to test **P4**, whose detailed results are depicted in Table A2 of the Appendix and can be directly compared with those of the telescopic formulation in Tables 1 and 2. In our opinion, **P4** compares unfavorably to the results reported in Table 2. Hence, we discard it for the competitive testing that will be carried out later.

4.3 | Competitive comparison on larger instances

In this section, we perform a competitive comparison of the most prominent formulations using medium and large-size instances.

We begin with a set of 35 medium-size Euclidean instances that we divide into three subsets: the first is made of 15 instances with 150 nodes, the second contains 10 instances with 175 nodes, and the third contains also 10 instances with 200 nodes. In Figure 2, we depict a plot for each of the subsets. Each horizontal axis contains the instance number in the subset, and the vertical axis compares the computing time (in seconds) needed to solve each instance to optimality when we apply **P1** with $K0$ (gray line) and **P3** with $K4$ (black line). For this experiment, a time limit of 3600 s is imposed for both methods. As can be seen from this figure, the reported time is significantly shorter when **P3**($K4$) is applied compared to **P1**($K0$) in most instances. Only instance number 06 of the subset with 200 nodes reached the time limit with **P1**($K0$), while **P3**($K4$) is able to solve it in just 365 s. Detailed information of the output of this experiment is provided in Table A3 of the Appendix.



(C)

FIGURE 2 Comparison of CPU times for the two methods with Euclidean instances. (A) Instances with 150 nodes. (B) Instances with 175 nodes. (C) Instances with 200 nodes

Table 4 reports a similar experiment carried out also with **P1**($K0$) and **P3**($K4$) for large-size Euclidean instances, that is, those with 300 nodes, also imposing a time limit of 3600 s. From the table, we can see that **P1**($K0$) is able to solve 9 out of 15 instances within the time limit. For those instances that **P1**($K0$) has not been able to solve, we report the optimality gap. Regarding **P3**($K4$), all instances are solved before reaching the time limit and many of them are solved in a significantly shorter computing time compared to **P1**($K0$). In this experiment, **P3**($K4$) is able to find violated inequalities (9) in four instances, as reported in the last column of the table.

We conclude our competitive comparison by performing an experiment using a set of 10 large-size uniform instances. The results of this experiment are reported in Table 5. This table is divided into three blocks of different instance sizes that, this time, are of 300, 400, and 500 nodes, respectively. The instances used here are of our particular interest as they were the largest generated in [31] although, as reported in that paper, these authors were not able to use them in their experiments since the meta-heuristic they proposed was only able to obtain solutions with zero objective function value in all cases. In these instances, the distance values are $\{0, \dots, 9\}$. To tackle these instances we use **P3**($K4$) and **P1**($T3$), the best configurations we found for the Kuby and telescopic formulations, respectively, in our experiments of Section 4.2. In this experiment, we impose a time limit of 7200 s for both formulations. As we report in the columns referring to **P3**($K4$), none of the instances are solved with this formulation after two hours of computing time, and the gap between the lower (LB) and upper bounds (UB) seems far from being closed. However, we are able to solve to optimality all the instances of the experiment using **P1**($T3$) with relatively short computing times, as we report in the last two columns of the table. Moreover, we show that the optimal value is 1 for all instances, in contrast to the value of 0, which was obtained by the heuristic in [31].

TABLE 4 Computational results for large-size Euclidean instances

		P1($K0$)			P3($K4$)		
n	Inst.	CPU	$\overline{D}$	OptGap	CPU	$\overline{D}$	Ineq. (9)
300	01	3600*	91.0	115.1%	3581	–	–
	02	3600*	87.8	7.6%	1085	–	–
	03	565	92.1	–	491	87.0	351
	04	3600*	92.6	43.4%	952	88.9	240
	05	540	91.2	–	411	87.5	278
	06	1520	92.2	–	484	–	–
	07	2370	92.0	–	425	–	–
	08	3600*	90.1	30.8%	2788	–	–
	09	3600*	92.0	0.6%	2343	87.5	248
	10	2788	91.8	–	2230	–	–
	11	2440	91.6	–	1735	–	–
	12	1240	89.7	–	671	–	–
	13	3600*	92.7	5.2%	857	–	–
	14	1355	82.5	–	709	–	–
	15	985	92.9	–	379	–	–

TABLE 5 Computational results for large-size uniform discrete values in $\{0, \dots, 9\}$ instances

		P3($K4$)			P1($T3$)	
n	Inst.	$\underline{D}$	UB	CPU	Opt.	CPU
300	SOM-b-10	0	5	7215	1	159
	SOM-b-11	0	5	7218	1	128
	SOM-b-12	0	5	7215	1	127
400	SOM-b-13	0	6.77	7245	1	278
	SOM-b-14	0	6.90	7247	1	251
	SOM-b-15	0	6.96	7248	1	267
	SOM-b-16	0	6.65	7246	1	490
500	SOM-b-17	0	8	7300	1	589
	SOM-b-18	0	8	7303	1	1873
	SOM-b-19	0	8	7299	1	1269
	SOM-b-20	0	8	7305	1	552

Finally, we note that we also tested different procedures where we obtained the best upper bounds using $E5$ and $T5$ and gave them to the best performing formulations. However the time spent in computing the upper bounds was not compensated by the time saved due to the improvement in their quality.

5 | CONCLUDING REMARKS

In this article, we have conducted a study of three different families of formulations for the capacitated dispersion problem and derived valid inequalities for each of them. The theoretical developments have been complemented by a series of computational tests aiming at evaluating the potential of the findings. By a systematic analysis of all the families of inequalities proposed, we have been able to devise two solution procedures. The numerical results show that our study rendered two procedures that have a superior performance when compared to the standard use of a state-of-the-art solver in large instances: **P3**($K4$) for the case of the Euclidean instances, and **P1**($T3$) for the case of uniform instances.

Although we have considered a setting where the choice of nodes is subject to a knapsack cover constraint, the formulations we proposed can be adapted easily to incorporate further constraints on the nodes. Their comparison remains valid. The only parts of our study that are specific to the knapsack cover constraint are the algorithms we use to compute the upper bounds.

Several interesting future research directions can be foreseen: on the one hand, the development of a branch-and-bound algorithm based on combinatorial relaxations that can be solved without a solver; on the other hand, the analyzed problem has some similarities with the so-called p-center problem. In fact, the problem that we have dealt in this work is a max-min location problem whereas the p-center problem is a min-max location problem. In both cases, the objective function can be written as a variable that is less than or equal in the max-min case, or greater than or equal in the min-max case, a family of expressions. Therefore, all the structure of valid inequalities that are written as that variable greater than or equal to (or less than or equal to) an expression can sometimes be rewritten for both the max-min model and the min-max model. An example of inequalities of this type are (4) and (9).

ACKNOWLEDGMENTS

This work was supported by Gobierno de España through Ministerio de Ciencia, Innovación y Ministerio de Universidades and Fondo Europeo de Desarrollo Regional (FEDER) under projects PGC2018-099428-B-100, PID2021-122344NB-I00, and RED2018-102363-T, by Generalitat Valenciana under projects PROMETEO/2021/063 and CIGE/2022/57, and by Gobierno de España through Ministerio de Educación y Formación Profesional under grant CAS19/00076.

The authors want to thank the comments and suggestions provided by three anonymous reviewers that have contributed to improve the content and readability of the article.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in MDPLIB at http://grafo.etsii.urjc.es/optsicom/. These data were derived from the following resources available in the public domain: MDPLIB, http://grafo.etsii.urjc.es/optsicom/.

ORCID

Juanjo Peiró https://orcid.org/0000-0002-2944-599X
Hande Yaman https://orcid.org/0000-0002-3392-1127

REFERENCES

[1] M. Albareda-Sambola, Y. Hinojosa, A. Marín, and J. Puerto, *When centers can fail: A close second opportunity*, Comput. Oper. Res. **62** (2015), 145–156.
[2] M. Albareda-Sambola, L. I. Martínez-Merino, and A. M. Rodríguez-Chía, *The stratified p-center problem*, Comput. Oper. Res. **108** (2019), 213–225.
[3] A. Atamtürk, *Cover and pack inequalities for (mixed) integer programming*, Ann. Oper. Res. **139** (2005), 21–38.
[4] S. Benati, S. García, and J. Puerto, *Mixed integer linear programming and heuristic methods for feature selection in clustering*, J. Oper. Res. Soc. **69** (2018), no. 9, 1379–1395.
[5] H. Calik and B. C. Tansel, *Double bound method for solving the p-center location problem*, Comput. Oper. Res. **40** (2013), no. 12, 2991–2999.
[6] R. Chandrasekaran and A. Daughety, *Location on tree networks: P-centre and n-dispersion problems*, Math. Oper. Res. **6** (1981), 50–57.
[7] C. Contardo and L. Costa, *On the optimal layout of a dining room in the era of COVID-19 using mathematical optimization*, Int. Trans. Oper. Res. **29** (2022), no. 6, 3294–3315.

[8] G. Cornuéjols, G. Nemhauser, and L. Wolsey, *A canonical representation of simple plant location problems and its applications*, SIAM J. Algebr. Discr. Methods **1** (1980), no. 3, 261–272.
[9] A. Duarte and R. Martí, *Tabu search and GRASP for the MDP*, Eur. J. Oper. Res. **178** (2007), 71–84.
[10] S. Elloumi, M. Labbé, and Y. Pochet, *A new formulation and resolution method for the p-center problem*, INFORMS J. Comput. **16** (2004), no. 1, 84–94.
[11] E. Erkut, *The discrete p-dispersion problem*, Eur. J. Oper. Res. **46** (1990), no. 1, 48–60.
[12] E. Erkut and S. Neuman, *Comparison of four models for dispersing facilities*, INFOR Inform. Syst. Oper. Res. **29** (1991), no. 2, 68–86.
[13] I. Espejo, A. Marín, and A. M. Rodríguez-Chía, *Capacitated p-center problem with failure foresight*, Eur. J. Oper. Res. **247** (2015), no. 1, 229–244.
[14] E. Fernández, J. Kalcsics, and S. Nickel, *The maximum dispersion problem*, Omega **41** (2013), no. 4, 721–730.
[15] M. Gallego, A. Duarte, M. Laguna, and R. Martí, *Hybrid heuristics for the maximum diversity problem*, Comput. Optim. Appl. **44** (2009), no. 3, 441–426.
[16] S. García, M. Labbé, and A. Marín, *Solving large p-median problems with a radius formulation*, INFORMS J. Comput. **23** (2011), no. 4, 546–556.
[17] J. B. Ghosh, *Computational aspects of the maximum diversity problem*, Oper. Res. Lett. **19** (1996), no. 4, 175–181.
[18] F. Glover, C. C. Kuo, and K. S. Dhir, *Heuristic algorithms for the maximum diversity problem*, J. Inf. Optim. Sci. **19** (1998), 109–132.
[19] R. K. Kincaid, *Good solutions to discrete noxious location problems via metaheuristics*, Ann. Oper. Res. **40** (1992), no. 1, 265–281.
[20] M. J. Kuby, *Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems*, Geogr. Anal. **19** (1987), no. 4, 315–329.
[21] C.-C. Kuo, F. Glover, and K. S. Dhir, *Analyzing and modeling the maximum diversity problem by zero-one programming*, Decis. Sci. **24** (1993), no. 6, 1171–1185.
[22] A. Marín, S. Nickel, J. Puerto, and S. Velten, *A flexible model and efficient solution strategies for discrete location problems*, Discr. Appl. Math. **157** (2009), no. 5, 1128–1145.
[23] R. Martí and A. Duarte, *MPDLIB maximum diversity problem library*, 2010.
[24] R. Martí, M. Gallego, and A. Duarte, *A branch and bound algorithm for the maximum diversity problem*, Eur. J. Oper. Res. **200** (2010), no. 1, 36–44.
[25] R. Martí, A. Martínez-Gavara, S. Pérez-Peló, and J. Sánchez-Oro, *A review on discrete diversity and dispersion maximization from an OR perspective*, Eur. J. Oper. Res. **299** (2022), 795–813.
[26] R. Martí, A. Martínez-Gavara, and J. Sánchez-Oro, *The capacitated dispersion problem: An optimization model and a memetic algorithm*, Memetic Computing **13** (2021), 131–146.
[27] A. Martínez-Gavara, T. Corberán, and R. Martí, *GRASP and tabu search for the generalized dispersion problem*, Expert Syst. Appl. **173** (2021), 114703.
[28] N. Mladenović, R. Todosijević, D. Urošević, and M. Ratli, *Solving the capacitated dispersion problem with variable neighborhood search approaches: From basic to skewed VNS*, Comput. Oper. Res. **139** (2022), 105622.
[29] A. T. Murray, *Planning for classroom physical distancing to minimize the threat of COVID-19 disease spread*, PLoS One **15** (2020), no. 12, e0243345.
[30] F. Parreño, R. Alvarez-Valdés, and R. Martí, *Measuring diversity. A review and an empirical analysis*, Eur. J. Oper. Res. **289** (2021), 515–532.
[31] J. Peiró, I. Jiménez, J. Laguardia, and R. Martí, *Heuristics for the capacitated dispersion problem*, Int. Trans. Oper. Res. **28** (2021), no. 1, 119–141.
[32] O. A. Prokopyev, N. Kong, and D. L. Martinez-Torres, *The equitable dispersion problem*, Eur. J. Oper. Res. **197** (2009), 59–67.
[33] J. Puerto, A. B. Ramos, and A. M. Rodríguez-Chía, *Single-allocation ordered median hub location problems*, Comput. Oper. Res. **38** (2011), no. 2, 559–570.
[34] M. G. Resende, R. Martí, M. Gallego, and A. Duarte, *GRASP and path relinking for the max-min diversity problem*, Comput. Oper. Res. **37** (2010), no. 3, 498–508.
[35] D. Rosenkrantz, G. K. Tayi, and S. S. Ravi, *Facility dispersion problems under capacity and cost constraints*, J. Comb. Optim. **4** (2000), 7–33.
[36] D. Sayah and S. Irnich, *A new compact formulation for the discrete p-dispersion problem*, Eur. J. Oper. Res. **256** (2017), no. 1, 62–67.
[37] F. Sayyady and Y. Fathi, *An integer programming approach for solving the p-dispersion problem*, Eur. J. Oper. Res. **253** (2016), no. 1, 216–225.

How to cite this article: M. Landete, J. Peiró, and H. Yaman, *Formulations and valid inequalities for the capacitated dispersion problem*, Networks. **81** (2023), 294–315. https://doi.org/10.1002/net.22132

APPENDIX

Table A2 shows the results of procedure **P4** on $T4$ regarding the best bound ($\overline{D}$) obtained before solving $T4$, as well as the computing time needed to apply the *while* loop over these two remarks until no new $\overline{D}$ is reached. The last column of the table also reports the total computing time needed to solve afterwards the integer program with $T4$. These results can be directly compared with those of the telescopic formulation in Tables 1 and 2.

Table A3 is divided by rows in three blocks, one for each instance sizes tackled: 150, 175, and 200 nodes. By columns, we show some measures of the solver performance when applying **P1** with $K0$ (that we abbreviate as **P1**($K0$)) and **P3** with $K4$ (**P3**($K4$)). The CPU columns report the computing time (in seconds) needed to solve the instance to optimality. For this experiment, a time limit of 3600 s is imposed for both methods. We also report the value of $\overline{D}$ for the two methods and the number of violated inequalities (9) for **P3**($K4$).

TABLE A1 Valid inequalities (8) added to $K3$ for the instances in the testing set

Family	Inst.	Ineq. (8) added
Euclid.	e01	19
	e02	12
	e03	18
	e04	18
	e05	12
	e06	17
	e07	10
	e08	22
	e09	4
	e10	5
	e11	19
	e12	7
	e13	10
	e14	15
	e15	32
	Average	14.6
Uni_LR	u01	112
	u02	140
	u03	101
	u04	127
	u05	137
	u06	160
	u07	117
	u08	181
	u09	139
	u10	230
	u11	132
	u12	147
	u13	99
	u14	120
	u15	178
	Average	141.3
	u_b_1	0
	u_b_2	0
	u_b_3	0
Uni_SR	u_b_4	0
	u_b_5	0
	u_b_6	0
	u_b_7	0
	Average	0

TABLE A2 Numerical results with **P4** on $T4$

Family	Inst.	$\overline{D}$	t.	CPU
Euclid.	e01	762.45	0	117
	e02	783.01	2	271
	e03	808.46	2	154
	e04	806.28	1	494
	e05	789.96	1	501
	e06	819.96	2	2336
	e07	798.57	1	134
	e08	731.14	2	927
	e09	839.86	2	674
	e10	809.82	1	874
	e11	824.24	1	242
	e12	733.72	1	190
	e13	799.83	2	103
	e14	713.6	1	93
	e15	805.03	2	207
Uni_LR	u01	832.85	1	415
	u02	838	1	307
	u03	831	1	416
	u04	845	1	538
	u05	833	1	269
	u06	838	1	343
	u07	837.87	1	601
	u08	833	1	687
	u09	826	1	230
	u10	824	0	288
	u11	827	1	2041
	u12	839	1	1065
	u13	829	2	4398
	u14	841	2	1076
	u15	829	1	1487
	u_b_1	9	1	337
	u_b_2	9	1	40
	u_b_3	9	1	164
Uni_SR	u_b_4	9	1	273
	u_b_5	9	2	627
	u_b_6	9	2	407
	u_b_7	9	2	1296

TABLE A3 Computational results for medium-size Euclidean instances

		P1($K0$)		P3($K4$)		
n	Inst.	CPU	$\overline{D}$	CPU	$\overline{D}$	Ineq. (9)
150	01	244	10.2	227	–	0
	02	359	9.6	59	–	0
	03	296	9.9	61	–	0
	04	103	10.0	49	–	0
	05	122	9.5	53	–	0
	06	322	10.0	183	9–5	94
	07	144	10.0	49	–	0
	08	761	10.4	78	–	0
	09	530	10.1	54	–	0
	10	1704	9.9	53	–	0
	11	363	9.8	180	9–1	99
	12	629	10.2	54	–	0
	13	1573	9.5	77	–	0
	14	371	10.4	305	–	0
	15	369	9.7	255	9.1	104
175	01	230	10.0	161	9.3	113
	02	428	10.0	114	–	0
	03	401	10.2	185	–	0
	04	277	9.5	197	–	0
	05	208	10.0	185	–	0
	06	226	10.3	397	–	0
	07	1616	10.0	396	–	0
	08	2509	10.0	229	–	0
	09	149	10.2	76	9–6	140
	10	344	9.9	97	–	0
200	01	614	10.0	473	–	0
	02	535	10.0	477	–	0
	03	510	10.2	665	–	0
	04	457	10.0	494	–	0
	05	1160	9.8	1117	–	0
	06	3600*	10.1	365	–	0
	07	417	9.6	264	–	0
	08	947	10.1	212	–	0
	09	331	9.8	161	–	0
	10	365	10.1	107	–	0