Computers & Operations Research 168 (2024) 106682

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Solving Euclidean Max-Sum problems exactly with cutting planes

Hoa T. Bui *, Sandy Spiers, Ryan Loxton

ARC Training Centre for Transforming Maintenance through Data Science, Curtin University, Perth, Australia
Curtin Centre for Optimisation and Decision Science, Curtin University, Perth, Australia

ARTICLE INFO

Keywords:
Euclidean distance matrix
Cutting plane methods
Constrained diversity sum
Exact algorithms
Nonlinear binary optimisation

ABSTRACT

This paper studies binary quadratic programs in which the objective is defined by the maximisation of a Euclidean distance matrix, subject to a general polyhedral constraint set. This class of nonconcave maximisation problems, which we refer to as the Euclidean Max-Sum problem, includes the capacitated, generalised and max-sum diversity problems as special cases. Due to the nonconcave objective, traditional cutting plane algorithms are not guaranteed to converge globally. In this paper, we introduce two exact cutting plane algorithms to address this limitation. The new algorithms remove the need for a concave reformulation, which is known to significantly slow down convergence. We establish exactness of the new algorithms by examining the concavity of the quadratic objective in a given direction, a concept we refer to as *directional concavity*. Numerical results show that the algorithms outperform other exact methods for benchmark diversity problems (capacitated, generalised and max-sum), and can easily solve problems of up to three thousand variables.

1. Introduction

In this paper, we show how cutting planes can be used to generate exact solutions for the problem of maximising the sum of pairwise Euclidean distances between selected points, subject to general polyhedral constraints, hereafter referred to as the *Euclidean Max-Sum problem* (EMSP). The (EMSP) is a generalisation of the Euclidean Max-Sum diversity problem (Spiers et al., 2023), in which the cardinality constraint is replaced by a general polyhedral set. More precisely, given a set of locations $v_1, \ldots, v_n \in \mathbb{R}^s$ ($s \geq 1$), the (EMSP) is defined as the following nonconcave binary maximisation problem,

$$\begin{aligned} \max \quad & f(x) := \tfrac{1}{2}\langle Qx, x\rangle, \\ \text{s.t.} \quad & x \in P \cap \{0,1\}^n, \end{aligned} \tag{EMSP}$$

where $Q = [q_{ij}]_{i,j=1,\ldots,n}$ is an $n \times n$ Euclidean distance matrix defined by $q_{ij} := \left\|v_i - v_j\right\|$, and where $P \subset \mathbb{R}^n$ is a polyhedral set defined by

$$P = \{x \in \mathbb{R}^n : Ax \leq a\},$$

where $A \in \mathbb{R}^{m\times n}$ and $a \in \mathbb{R}^m$. Here, the definition of x can be easily generalised to include both integer and continuous variables. The matrix Q is symmetric, hollow and has positive off-diagonal entries. By a result from Schoenberg (1937), given $v_1, \ldots, v_n \in \mathbb{R}^s$, we can construct another set of n points $u_1, \ldots, u_n \in \mathbb{R}^n$ such that $\left\|v_i - v_j\right\| = \left\|u_i - u_j\right\|^2$ for $i, j = 1, \ldots, n$. As such, the distance matrix Q is also a *squared* Euclidean distance matrix. Furthermore, it is well-known that squared Euclidean distance matrices are *conditionally negative definite*, i.e., $\langle Qx, x\rangle \leq 0$ if $\sum_{i=1}^n x_i = 0$, and have exactly one positive eigenvalue (Bapat and Raghavan, 1997, Corollary 4.1.5, Theorem 4.1.7). In this work, we exploit this property to show how the cutting plane methodology, which is normally restricted to concave maximisation problems, can be applied to find an optimal solution of (EMSP).

The Euclidean max-sum problem has various important practical applications. In machine learning and statistical analysis, Euclidean distance is often used as a measure of dissimilarity between data points in clustering algorithms (Madhulatha, 2012; Shirkhorshidi et al., 2015). By maximising the Euclidean distance between points, clusters can be formed based on their dissimilarity, allowing for effective grouping and classification of data. An example of this is the well-known k-means clustering problem (MacQueen et al., 1967; Lloyd, 1982). Furthermore, in various practical applications such as urban planning or network design, there is a need to strategically locate unwanted facilities such as waste disposal sites or polluting industries (Kuby, 1987; Erkut and Neuman, 1989). Maximising the distance between these unwanted (but necessary) facilities and sensitive areas such as residential zones or environmental conservation areas helps minimise the negative impact on the surrounding communities or ecosystems. Lastly, maximising Euclidean distances allows for the selection of points that capture diverse characteristics or represent different regions of interest, thereby enhancing the coverage and diversity of the chosen set.

* Corresponding author at: Curtin Centre for Optimisation and Decision Science, Curtin University, Perth, Australia.
E-mail addresses: hoa.bui@curtin.edu.au (H.T. Bui), sandy.spiers@postgrad.curtin.edu.au (S. Spiers), r.loxton@curtin.edu.au (R. Loxton).

https://doi.org/10.1016/j.cor.2024.106682
Received 17 September 2023; Received in revised form 19 February 2024; Accepted 26 April 2024
Available online 6 May 2024
0305-0548/

This is seen in the Euclidean max-sum diversity problem (Spiers et al., 2023), which is a special case of the (EMSP) where the polyhedral set P is defined by a single cardinality constraint. For a recent review of this and other diversity models and their associated solution algorithms, we direct the reader to the comprehensive reviews in Martí et al. (2022) and Parreño et al. (2021). Among other applications, the maximum diversity problem has gained recent attention for its use in forming teams with diverse skill sets (Hochbaum et al., 2023).

Recently, in Spiers et al. (2023), we formulated a cutting plane algorithm for the Euclidean max-sum diversity problem by establishing the concavity of the objective function on the hyperplane $\sum_{i=1}^{n} x_i = p$, which ensures that tangent planes of feasible solutions serve as valid upper planes. As such, our cutting plane algorithm is globally convergent for the Euclidean max-sum diversity problem. The resultant exact algorithm is competitive with heuristic and meta-heuristic methods and can solve two-coordinate instances of up to eighty thousand variables. However, without the cardinality constraint, the objective function is not concave over the feasible set, and hence tangent planes do not always form valid cuts. The purpose of the current paper is to develop a new cutting plane methodology that still converges for this more general case, where concavity is not guaranteed.

To the best of our knowledge, outside of the Euclidean max-sum diversity problem, quadratic maximisation problems defined by Euclidean distance matrices have never been explored in isolation. One reason for this is that these maximisation problems are, in general, nonlinear and nonconcave. Mixed-integer nonlinear programming is one of the most challenging classes of optimisation problems. Although there are several exact methods that provide general frameworks to tackle *concave* maximisation problems, including outer approximation (Duran and Grossmann, 1986; Leyffer, 1993; Lubin et al., 2018; Kronqvist et al., 2020), branch and bound (Gupta and Ravindran, 1983; Vielma et al., 2008; Bonami et al., 2013), and cutting plane methods (Westerlund and Pettersson, 1995; Kronqvist et al., 2016; Lundell et al., 2022), advancements in exact algorithms for *nonconcave* problems are still modest. The most common way to handle binary nonconcave maximisation is to reformulate the problem into an equivalent *concave problem* by using a penalty approach, before applying exact methods to the new concave problem (Androulakis et al., 1995). In particular, thanks to the property $x_i^2 = x_i$ $(i = 1, \ldots, n)$ for $x \in \{0,1\}^n$, the nonconcave objective $f(x) = \frac{1}{2}\langle Qx, x\rangle$ can be replaced by a concave function $f_\rho(x) = \frac{1}{2}\langle (Q-\rho I_n)x, x\rangle + \frac{1}{2}\rho\sum_{i=1}^{n} x_i$, where ρ is greater than the largest eigenvalue of Q. This technique is one of the ways modern solvers such as CPLEX and Gurobi solve binary quadratic programming problems (Bliek et al., 2014; Lima and Grossmann, 2017). However, computational studies have shown that the convergence of this approach is often slow, especially when ρ is large (Bliek et al., 2014; Lima and Grossmann, 2017; Bonami et al., 2022). This is explained mathematically in our recent work (Proposition 6, Spiers et al. (2023)). In the case of the (EMSP), where Q is a Euclidean distance matrix, the Perron–Frobenius Theorem implies that the largest eigenvalue of Q is bounded by the minimum and maximum row sums, and hence the concave reformulation requires choosing $\rho > 0$ to ensure concavity. However, we proved in Spiers et al. (2023) that if the polyhedral set P is defined by a single cardinality constraint, then the concave reformulation is actually unnecessary, since it is possible to develop a globally convergent cutting plane methodology that directly tackles the original problem.

This paper extends the results in Spiers et al. (2023) to general Euclidean distance maximisation by relaxing the requirement for a cardinality constraint. This is achieved by exploiting the property that Euclidean distance matrices have exactly one positive eigenvalue. To provide intuition on the key idea, consider a full eigenvalue decomposition of the objective function,

$$f(x) = \frac{1}{2}\langle Qx, x\rangle = \frac{1}{2}x^T\left(\sum_{i=1}^{n}\lambda_i v_i v_i^T\right)x = \frac{1}{2}\sum_{i=1}^{n}\lambda_i x^T\left(v_i v_i^T\right)x,$$

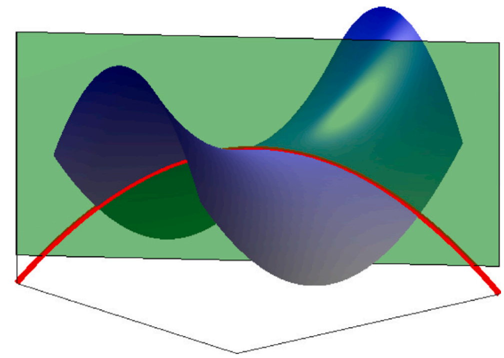

Fig. 1. The intersection of a paraboloid and a hyperplane is either convex or concave.

where $\{v_1, \ldots, v_n\}$ and $\lambda_1 \geq \cdots \geq \lambda_n$ are eigenvectors and eigenvalues of Q. This expresses the quadratic objective as a sum of functional components, which are either convex or concave depending on the sign of the respective eigenvalues. However, as Q is a Euclidean distance matrix, it is known to contain exactly one positive eigenvalue, and therefore $f(x)$ has only one convex component. By restricting our search domain to exclude directions that traverse this convex component, our objective function can effectively be treated as a concave function (see Lemma 1).

As an example, consider the hyperbolic paraboloid defined by

$$g(x, y) = xy = \frac{1}{4}(x+y)^2 - \frac{1}{4}(x-y)^2.$$

Clearly, whenever $ax + by = 0$ $(ab > 0)$, the function reduces to $g(x, y) = x(-\frac{ax}{b}) = -\frac{a}{b}x^2$, which is concave. Hence, while $g(x, y)$ is nonconcave for $x, y \in \mathbb{R}$, it is concave on the $ax+by = 0$ plane. The resultant concave parabola is shown in red in Fig. 1. This is essentially the technique used in Spiers et al. (2023), where the Euclidean distance matrix is known to contain exactly one positive eigenvalue, and hence the objective has one convex functional component. The cardinality constraint then ensures that the feasible set excludes this convex component, and the quadratic function can be treated as concave. For the general problem (EMSP), which may not include a cardinality constraint, the main idea of our approach is to only generate the tangent planes on concave directions. By doing so, the cutting planes are valid, and the algorithm always converges to an optimal solution.

The remainder of this paper is organised as follows. In Section 2, we formalise the concept of directional concavity and, based on this, formulate two key sufficient conditions for valid tangent planes, as detailed in Theorem 3. These conditions then form the basis of two exact cutting plane algorithms, which vary in their approach to generating new cuts. Finally, in Section 3 we conduct extensive numerical experiments to evaluate the effectiveness of the proposed solution approaches.

2. Cutting plane algorithms

We denote the feasible set of (EMSP) as $\mathcal{K} := \{x \in \{0,1\}^n : x \in P\} \setminus \{0\}$, where $x = 0$ is excluded because $f(x) \geq 0 = f(0)$ for every $x \in \mathcal{K}$. Let $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ be the tangent plane of the function f, defined as:

$$h(x, y) := \langle Qy, x - y\rangle + \frac{1}{2}\langle Qy, y\rangle.$$

We say that the tangent plane at a feasible solution $y \in \mathcal{K}$ forms a *valid cut* if it provides an upper approximation for the optimal value of (EMSP), i.e, $f(x^*) \leq h(x^*, y)$, where x^* is an optimal solution of (EMSP). This differs from the majority of the existing literature, where valid cuts provide an upper approximation of the objective function at *all* feasible solutions (not just at an optimal solution).

Since the function f in (EMSP) is not concave, not every feasible solution $y \in \mathcal{K}$ generates a valid cut. In this section, we establish sufficient conditions for when the tangent plane $h(x, y)$ is valid. The key to our approach is to study the concavity of the function f when restricted to a given direction, exploiting the observation that the restriction of a quadratic function to a line is either concave or convex.

2.1. Directional concavity

Given a vector $u \in \mathbb{R}^n \setminus \{0\}$, we say that u is a concave direction of Q if $\langle Qu, u\rangle \le 0$; this means that $f(x)$ is concave along the line with direction u emanating from the origin. Conversely, a vector $v \in \mathbb{R}^n \setminus \{0\}$ is a convex direction of Q if $\langle Qv, v\rangle \ge 0$. Note that $x - y$ is a concave direction of Q if and only if

$$\begin{aligned} h(x, y) - f(x) &= \langle Qy, x - y\rangle + \tfrac{1}{2}\langle Qy, y\rangle - \tfrac{1}{2}\langle Qx, x\rangle \\ &= \langle Qy, x - y\rangle - \tfrac{1}{2}\langle Q(x + y), x - y\rangle \\ &= -\tfrac{1}{2}\langle Q(x - y), x - y\rangle \ge 0. \end{aligned}$$

Thus, the tangent plane at y is an upper approximation of $f(x)$ when the line from y to x is a concave direction of Q. We now show that $u = x - y$ is a concave direction of the matrix Q if u is orthogonal to Qz, where z is a convex direction of Q.

Lemma 1. *Suppose $x, y \in \mathbb{R}^n$, and there is vector $z \in \mathbb{R}^n \setminus \{0\}$ such that*

a. $\langle Qz, z\rangle \ge 0$, and
b. $\langle Qz, x - y\rangle = 0$.

Then, $h(x, y) \ge f(x)$.

Proof. The inequality $h(x, y) \ge f(x)$ is equivalent to

$$\langle Q(x - y), x - y\rangle \le 0.$$

We suppose to the contrary that $\langle Q(x - y), x - y\rangle > 0$. Because Q is a Euclidean distance matrix, by Bapat and Raghavan (1997, Corollary 4.1.5, Theorem 4.1.7), matrix Q has exactly one positive eigenvalue. Furthermore, because Q is a real symmetric matrix, it is orthogonally diagonalisable. Let $\lambda_1 > 0 \ge \lambda_2 \ge \cdots \ge \lambda_n$ be the eigenvalues of Q, and let $v_1, \dots, v_n$ be the corresponding eigenvectors, which are normalised, and orthogonal. Then, we can express $x - y$ and z on the basis $\{v_1, \dots, v_n\}$ as follows:

$$x - y = \sum_{i=1}^{n} \alpha_i v_i, \quad z = \sum_{i=1}^{n} \beta_i v_i,$$

for some $\alpha_i, \beta_i \in \mathbb{R}$ $(i = 1, \dots, n)$. Then,

$$\langle Qz, z\rangle = \sum_{i=1}^{n} \lambda_i \beta_i^2 \ge 0, \tag{1}$$

$$\langle Qz, x - y\rangle = \sum_{i=1}^{n} \lambda_i \beta_i \alpha_i = 0, \tag{2}$$

$$\langle Q(x - y), x - y\rangle = \sum_{i=1}^{n} \lambda_i \alpha_i^2 > 0. \tag{3}$$

Because $\lambda_i \le 0$ $(i = 2, \dots, n)$, inequality (1) and $z \ne 0$ imply that $\beta_1 \ne 0$, and (3) implies that $\alpha_1 \ne 0$. Therefore, we can multiply both sides of (1) by $\alpha_1^2 > 0$, (2) by $-2\alpha_1\beta_1 \ne 0$, and (3) by $\beta_1^2 > 0$, and sum up to obtain

$$\begin{aligned} 0 &< \left(\lambda_1 \beta_1^2 \alpha_1^2 + \alpha_1^2 \sum_{i=2}^{n} \lambda_i \beta_i^2\right) - 2\left(\lambda_1 \beta_1^2 \alpha_1^2 + \alpha_1 \beta_1 \sum_{i=2}^{n} \lambda_i \beta_i \alpha_i\right) \\ &+ \left(\lambda_1 \beta_1^2 \alpha_1^2 + \beta_1^2 \sum_{i=2}^{n} \lambda_i \alpha_i^2\right) \\ &= \sum_{i=2}^{n} \lambda_i (\alpha_1^2 \beta_i^2 - 2\alpha_1 \beta_1 \alpha_i \beta_i + \beta_1^2 \alpha_i^2) = \sum_{i=2}^{n} \lambda_i (\alpha_1 \beta_i - \alpha_i \beta_1)^2. \end{aligned}$$

The inequality above only holds when there is at least one positive eigenvalue among $\lambda_2, \dots, \lambda_n$, which is a contradiction. Hence, it must hold that $\langle Q(x - y), x - y\rangle \le 0$. □

Recall that the Euclidean distance matrix Q is *conditionally negative definite*. The next result exploits this fact to replace condition (b) in Lemma 1 with two new conditions.

Lemma 2. *Suppose $x, y \in \mathbb{R}^n$, and there is $z \in \mathbb{R}^n \setminus \{0\}$ such that*

a. $\langle Qz, z\rangle \ge 0$,
b. $\langle Qz, x - y\rangle \le 0$, and
c. either $\frac{\sum_{i=1}^n (x_i - y_i)}{\sum_{i=1}^n z_i} \ge 0$, or $\sum_{i=1}^n (x_i - y_i) = \sum_{i=1}^n z_i = 0$.

Then, $h(x, y) \ge f(x)$.

Proof. Similar to Lemma 1, $f(x) \le h(x, y)$ is equivalent to $\langle Q(x - y), x - y\rangle \le 0$. Let $u := x - y$, and choose $w \in \mathbb{R}^n$ such that

$$w := \alpha z, \qquad \text{where } \alpha := \begin{cases} 1 & \text{if } \sum_{i=1}^n u_i = \sum_{i=1}^n z_i = 0, \\ \frac{\sum_{i=1}^n u_i}{\sum_{i=1}^n z_i} & \text{otherwise.} \end{cases}$$

From (c), $\alpha \ge 0$ and $\sum_{i=1}^n u_i = \sum_{i=1}^n w_i$, or equivalently $\sum_{i=1}^n (u_i - w_i) = 0$. Note that from (a) and (b), we have

$$\langle Qw, w\rangle = \alpha^2 \langle Qz, z\rangle \ge 0, \quad \langle Qw, u\rangle \le 0.$$

Because Q is conditionally negative definite, we have $\langle Q(w - u), w - u\rangle \le 0$. Combining this with $\langle Qw, w\rangle \ge 0$ and $\langle Qw, u\rangle \le 0$, we get

$$\langle Qu, u\rangle = \langle Q(w - u), w - u\rangle - \langle Qw, w\rangle + 2\langle Qw, u\rangle \le 0,$$

thus giving $f(x) \le h(x, y)$. □

Using Lemmas 1 and 2, we now establish conditions for when a tangent plane $h(x, y)$ provides an upper approximation for higher value solutions in $\mathcal{K}$, i.e., $h(x, y) \ge f(x)$ for all x such that $f(x) \ge f(y)$.

Theorem 3. *Suppose $x, y \in \mathbb{R}^n_+ \setminus \{0\}$, such that $f(x) \ge f(y)$. Then, $h(x, y) \ge f(x)$ if either*

a. $\sum_{i=1}^n x_i \le \sum_{i=1}^n y_i$, or
b. there is $w \in \mathbb{R}^n_+ \setminus \{0\}$ such that $\langle Qw, x - y\rangle \le 0$.

Proof. Because $f(x) \ge f(y)$, we have

$$\langle Q(x + y), x - y\rangle \ge 0. \tag{4}$$

a. Suppose $\sum_{i=1}^n x_i \le \sum_{i=1}^n y_i$. Choose $z := -(x + y)$. Because Q has positive off-diagonal entries, then

$$\langle Qz, z\rangle = \langle -Q(x + y), -(x + y)\rangle = \langle Q(x + y), x + y\rangle \ge 0,$$

and $\langle Qz, x - y\rangle \le 0$. Taking into account that $\sum_{i=1}^n x_i \le \sum_{i=1}^n y_i$ and $x + y \in \mathbb{R}^n_+$, condition (c) in Lemma 2 is fulfilled. Hence, by Lemma 2, the inequality $h(x, y) \ge f(x)$ holds.

b. Suppose there is $w \in \mathbb{R}^n_+ \setminus \{0\}$ such that $\langle Qw, x - y\rangle \le 0$. Then, given (4), there exists a $z \in \mathbb{R}^n_+ \setminus \{0\}$ on the line between w and $x + y$ such that $\langle Qz, x - y\rangle = 0$. Note that Q has zero diagonal and positive off-diagonal entries, hence $\langle Qz, z\rangle > 0$. Therefore by Lemma 1, we have that the inequality $h(x, y) \ge f(x)$ holds. □

2.2. Cutting plane algorithms

We now introduce two cutting plane algorithms designed to solve the nonconcave quadratic problem (EMSP). Let $A \subset \mathbb{R}^n_+$ denote an arbitrary finite set of points that generate valid tangent planes, such that for all $y \in A$ we have $f(x^*) \le h(x^*, y)$ for an optimal solution x^* where x^* is an optimal solution. Then, we define

$$\Gamma_A := \left\{(x, \theta) \in \mathbb{R}^{n+1} \;:\; x \in \mathcal{K},\ \theta \le h(x, y),\ \forall y \in A\right\}.$$

The cutting plane model of the (EMSP) is then given as the following mixed-integer linear program,

$$\max_{(x,\theta)\in\Gamma_A} \theta. \tag{ILP$_A$}$$

Since the points in A generate valid tangents we have $h(x^*, y) \geq f(x^*)$ for all $y \in A$ and hence $(x^*, f(x^*))$ is feasible for (ILP$_A$), meaning the optimal value of (ILP$_A$) is a valid upper bound for EMSP. We now present two algorithms for solving the (EMSP) that iteratively generate new, valid tangent planes, thereby tightening the approximation of (ILP$_A$). Provided the first cut added is valid, both methods are guaranteed to converge to an optimal solution of the (EMSP). Note that from Theorem 3.a, we can always choose the first cut to be the solution to the maximum cardinality problem. Let y be the solution to $\max_{x\in\mathcal{K}} \sum_{i=1}^n x_i$, then $y \in K$, $f(x^*) \geq f(y)$ and $\sum_{i=1}^n x_i^* \leq \sum_{i=1}^n y_i$. Therefore, by Theorem 3.a, y generates a valid tangent.

The first algorithm makes use of the following proposition, which asserts that the tangent plane at the optimal solution of (ILP$_A$) is always valid.

Proposition 4. *Given $A \subset \mathbb{R}^n_+$ is a set of points that generate valid tangents, let (x, θ) be an optimal solution of the cutting plane problem (ILP$_A$). Then $f(x^*) \leq h(x^*, x)$, where x^* is an optimal solution of (EMSP).*

Proof. We begin by proving that there is a $y \in A$ such that $\langle Qy, x^* - x\rangle \leq 0$. Suppose, for a contradiction, that for all $y \in A$ we have $\langle Qy, x^* - x\rangle > 0$, or equivalently, $\langle Qy, x^*\rangle > \langle Qy, x\rangle$. Then,

$$\theta \leq h(x, y) < h(x^*, y)$$

holds for all $y \in A$. Let $\hat{\theta}$ be such that,

$$\hat{\theta} := \min_{y\in A} h(x^*, y) > \theta.$$

However, $(x^*, \hat{\theta}) \in \Gamma_A$, and $\hat{\theta} > \theta$, which contradicts (x, θ) being an optimal solution. Hence, the first assertion is settled. The second assertion is a direct consequence of Theorem 3.b, where $w = y \neq 0$ (since otherwise y would not generate a valid cut), and noting that $f(x) \leq f(x^*)$. Hence, $f(x^*) \leq h(x^*, x)$. □

Using this result, we can now solve the (EMSP) by repeatedly solving (ILP$_A$) to optimality, and using the solutions as new valid tangent planes. An implementation of this approach is shown in Algorithm 1, and its convergence is established in Proposition 5.

Algorithm 1: Repeated (ILP$_A$) method for solving (EMSP).

function *RepeatedILP(f,$\mathcal{K}$,ϵ)*
  $k \leftarrow 0$, $UB_k \leftarrow +\infty$
  Take $x^0 \in \arg\max_{x\in\mathcal{K}} \sum_{i=1}^n x_i$
  Set $A_1 \leftarrow \{x^0\}$, $LB_k \leftarrow f(x^k)$
  while $\frac{UB_k - LB_k}{LB_k} > \epsilon$ **do**
    $k \leftarrow k + 1$
    Solve (ILP$_{A_k}$) to obtain (x^k, θ^k)
    $UB_k \leftarrow \theta_k$, $LB_k \leftarrow \max\{LB_{k-1}, f(x^k)\}$
    $A_{k+1} \leftarrow A_k \cup \{x^k\}$
  end
  return LB_k
end

Proposition 5. *Algorithm 1 converges to an optimal solution of the (EMSP) in a finite number of steps.*

Proof. As every (ILP$_A$) is solved to optimality, we have from Proposition 4 that the tangent plane of every x^k is valid. This implies that $(x^*, f(x^*))$ is always feasible at every step k, i.e., $(x^*, f(x^*)) \in \Gamma_{A_k}$ for all $k \geq 0$. Thus,

$$UB_k = \max_{(x,\theta)\in\Gamma_{A_k}} \theta \geq f(x^*) = \max_{x\in\mathcal{K}} f(x) \geq LB_k.$$

Because the feasible region $\mathcal{K}$ is finite (variables x are discrete, and the polyhedral set P is bounded), there is a step k such that the optimal solution (x^k, θ^k) of (ILP$_{A_k}$) satisfies $x^k \in A_k$. In this case, we have $\text{UB}_k = \theta^k \leq h(x^k, x^k) = f(x^k) \leq \text{LB}_k$, and hence, $\text{UB}_k = \text{LB}_k$. When $\text{UB}_k = \text{LB}_k$, we have $\theta^k = \max_{x\in\mathcal{K}} f(x)$, and therefore Algorithm 1 has converged to an optimal solution. □

We show a worked example outlining the steps of Algorithm 1 in Appendix. Note that the repeated (ILP$_A$) algorithm is similar to the extended cutting plane method presented in Westerlund and Pettersson (1995), with a modification on the first cut added $h(x, x^0)$.

Although Algorithm 1 is globally convergent, it requires solving (ILP$_A$) to optimality at every iteration, since only the optimal solution is guaranteed to generate a valid cut. Depending on $\mathcal{K}$, this potentially represents a difficult mixed-integer programming problem. We now describe an alternative algorithm in which cuts are added at intermediate feasible points to accelerate convergence.

Recall from Theorem 3.a that $f(x) \leq h(x, y)$ whenever $f(x) \geq f(y)$ and $\sum_{i=1}^n x_i \leq \sum_{i=1}^n y_i$. As such, consider the restriction of the (EMSP) to points with cardinality $c \in \mathbb{N}$. The feasible region of this problem is then given by $\mathcal{K}_c := \{x \in \mathcal{K} : \sum_{i=1}^n x_i = c\}$. We can solve $\max_{x\in\mathcal{K}_c} f(x)$ exactly by instead solving the following linear cutting plane problem,

$$\max \quad \theta \tag{5}$$

$$\text{s.t.} \quad \theta \leq h(x, y), \quad \forall y \in \mathcal{K}_c, \tag{6}$$

$$x \in \mathcal{K}_c.$$

Although there are an exponential number of constraints in (6), we can accelerate the solution process by using a branch and cut methodology, whereby cuts are added on the fly during the search procedure.

Given we can solve $\max_{x\in\mathcal{K}_c} f(x)$ by the cutting plane problem (5), we can therefore decompose (EMSP) such that

$$\max_{x\in\mathcal{K}} f(x) = \max_{c=1,\dots,C} \max_{x\in\mathcal{K}_c} f(x)$$
$$= \max_{c=1,\dots,C} \max_{x,\theta} \left\{\theta : x \in \mathcal{K}_c, \theta \leq h(x, y), \forall y \in \mathcal{K}_c\right\}$$

where C is the maximum cardinality achievable in $\mathcal{K}$. Algorithm 2 below executes this decomposition by solving the inner maximisation problem repeatedly for decreasing cardinality c. At each step k (corresponding to cardinality c_k), the algorithm imposes not only the cuts corresponding to points in $\mathcal{K}_{c_k}$, but also additional cuts generated by feasible points with higher cardinality that were encountered in previous iterations. By Theorem 3.a, for such feasible points y, the cut $\theta \leq h(x, y)$ does not exclude improved solutions with cardinality less than or equal to y, and hence the decomposition remains valid. The process repeats until the optimal solution is found.

Algorithm 2: Forced cardinality method for solving (EMSP).

function *ForcedCardinality(f,$\mathcal{K}$,ϵ)*
  $k \leftarrow 0$, $UB_0 \leftarrow +\infty$
  Take $x^0 \in \arg\max_{x\in\mathcal{K}} \sum_{i=1}^n x_i$
  $LB_k \leftarrow f(x^k)$
  $c_1 \leftarrow \sum_{i=1}^n x_i^0$, $A_1 \leftarrow \{x^0\}$
  while $\frac{UB_k - LB_k}{LB_k} > \epsilon$ *and* $c_k > 0$ **do**
    $k \leftarrow k + 1$
    Solve $\max_{x,\theta} \left\{\theta : x \in \mathcal{K}_{c_k}, \theta \leq h(x, y), \forall y \in \mathcal{K}_{c_k} \cup A_k\right\}$ for (x^k, θ^k) using branch and cut, saving all cuts found and adding their corresponding points to A_{k+1}
    $LB_k \leftarrow \max\{LB_{k-1}, f(x^k)\}$
    Solve $\max_{(x,\theta)\in\Gamma_{A_{k+1}}} \{\theta : \sum_{i=1}^n x_i \leq c_k - 1\}$ for UB_k
    $c_{k+1} \leftarrow c_k - 1$
  end
  return LB_k
end

Proposition 6. *Algorithm 2 converges to an optimal solution of the (EMSP) in a finite number of steps.*

Proof. Let x^* be an optimal solution of the (EMSP) and suppose $\sum_{i=1}^n x_i^* = c_k$. We will prove two things: at step k, x^k is a solution of (EMSP), and prior to step k, UB_k is an upper bound for the optimal value of (EMSP), ensuring the algorithm does not terminate early. For the first assertion, observe that $\sum_{i=1}^n x_i^* \le \sum_{i=1}^n y_i$ and $f(x^*) \ge f(y)$ for all $y \in \mathcal{K}_{c_k} \cup A_k$. Therefore, by Theorem 3.a, $f(x^*) \le h(x^*, y)$ for all $y \in \mathcal{K}_{c_k} \cup A_k$. Hence, $(x^*, f(x^*))$ is a feasible solution to the subproblem on line 8 of Algorithm 2. Suppose that (x^k, θ^k) is optimal for this subproblem. Then we have that

$$f(x^*) \le \theta^k \le h(x^k, x^k) = f(x^k).$$

Therefore x^k obtained at step k must be optimal for (EMSP), and $LB_k = f(x^*)$.

We now prove that the upper bound determined on line 10 of Algorithm 2 is always valid if the optimal solution has not yet been reached, i.e., at iterations $l < k$ where $\sum_{i=1}^n x_i^* = c_k$. At step $l < k$, the set A_{l+1} contains only solutions with cardinality at least c_l. Therefore, $\sum_{i=1}^n x_i^* = c_k \le c_l - 1 < \sum_{i=1}^n y_i$ and $f(x^*) \ge f(y)$ for all $y \in A_{k+1}$. By Theorem 3.a, this ensures that $f(x^*) \le h(x^*, y)$, meaning $(x^*, f(x^*))$ is a feasible solution for the subproblem on line 10 of Algorithm 2. Hence, the optimal value of this subproblem (UB_l) is an upper bound of the globally optimal solution, i.e.,

$$UB_l = \max_{(x,\theta)\in \Gamma_{A_{l+1}}} \left\{ \theta : \sum_{i=1}^n x_i \le c_l - 1 \right\} \ge f(x^*).$$

As such, the algorithm does not terminate until a globally optimal solution has been found. □

Note that, the lower bound at each iteration k of Algorithm 2 satisfies

$$LB_k = \max_{c=c_k,\dots,C} \max_{x\in\mathcal{K}_c} f(x) = \max_{x\in\mathcal{K}} \left\{ f(x) : \sum_{i=1}^n x_i \ge c_k \right\}. \quad (7)$$

In other words, it is the best function value achievable for the current and higher cardinalities. We can show this by induction. For $k = 1$, this statement is clearly true since $A_1 = \{x^0\} \subset \mathcal{K}_{c_1}$ meaning that the subproblem on line 8 of Algorithm 2 reduces to problem (5), and hence by the arguments given above, x^1 is optimal for $\max_{x\in\mathcal{K}_{c_1}} f(x)$ and consequently $LB_1 = \max_{x\in\mathcal{K}_{c_1}} f(x)$. Suppose the assertion holds at step $k-1$. Then, for step k, there are two cases to consider;

(i) $\max_{x\in\mathcal{K}_{c_k}} f(x) < LB_{k-1}$, or
(ii) $\max_{x\in\mathcal{K}_{c_k}} f(x) \ge LB_{k-1}$.

For case (i), given (7) holds for $k-1$ we have that

$$\begin{aligned} LB_k &= \max\{LB_{k-1}, f(x^k)\} \\ &= LB_{k-1} \\ &= \max_{c=c_{k-1},\dots,C} \max_{x\in\mathcal{K}_c} f(x) \\ &= \max_{c=c_k,\dots,C} \max_{x\in\mathcal{K}_c} f(x) \end{aligned}$$

as required. For case (ii), suppose x' is optimal for $\max_{x\in\mathcal{K}_{c_k}} f(x)$. Then we must have $f(x') \ge f(y)$ and $\sum_{i=1}^n x_i' \le \sum_{i=1}^n y_i$ for all $y \in A_k$, and hence from Theorem 3.a, $f(x') \le h(x', y)$. Therefore, $(x', f(x'))$ is feasible for the linear subproblem on line 8 of Algorithm 2. This implies

$$f(x') \le \theta^k \le h(x^k, x^k) = f(x^k).$$

Thus, given (7) holds for $k-1$, we have

$$LB_k = \max\left\{LB_{k-1}, f(x^k)\right\} = f(x^k) = \max_{c=c_k,\dots,C} \max_{x\in\mathcal{K}_c} f(x)$$

as required.

In difficult instances of the (EMSP), a large number of tangent planes are potentially required to sufficiently approximate the objective function (such as with high-coordinate instances in Spiers et al. (2023)). To accelerate cut generation, recall that the Euclidean distance matrix is conditionally negative definite, and therefore $\langle Q(x-y), x-y\rangle \le 0$ holds for all $x, y \in \mathbb{R}^n$ with $\sum_{i=1}^n (x_i - y_i) = 0$. This implies that we can generate valid cuts even for non-integer y; specifically, $f(x) \le h(x, y)$ holds for any y (integer or continuous) that has the same cardinality as x. Therefore, Proposition 6 still holds if we modify the subproblem on line 8 of Algorithm 2 to include additional cuts generated by non-integer points with cardinality c_k. These tangents can be generated by solving the linear relaxation, and are therefore computationally cheap to generate and may improve the approximation.

However, using the linear relaxation potentially introduces a large integrality gap, possibly reducing the effectiveness of these cuts. A good strategy to reduce this gap is to ensure additional cuts are generated close to the best-known solution. This is achieved by employing a trust-region methodology, where the continuous solutions are constrained to the region defined by $\|x - y^k\|_1 \le \gamma$, where y^k is the current best-known integer solution up to step k and $\gamma \ge 0$ is a given parameter. When $x \in [0,1]^n$, this is easily enforced by the following constraint

$$\sum_{\substack{i=1 \\ y_i^k=0}}^n x_i + \sum_{\substack{i=1 \\ y_i^k=1}}^n (1 - x_i) \le \gamma. \quad (8)$$

Algorithm 3 outlines the process for generating LP tangents, where ϵ is the maximum allowable relative tolerance between bounds and M is the maximum number of cuts to be added. This algorithm is called during each iteration k, before solving the subproblem on line 8, and its main inputs are A_k and the current best-known solution y^k. The cuts from Algorithm 3 can then be introduced by replacing the subproblem in line 8 with

$$\max_{x,\theta} \left\{ \theta : x \in \mathcal{K}_{c_k}, \theta \le h(x,y), \forall y \in \mathcal{K}_{c_k} \cup A_k \cup L_k \right\}$$

where L_k comes from Algorithm 3. Note that as the points $y \in L_k$ are not necessarily integer feasible, it is not always true that $f(x^*) \ge f(y)$ and hence they do not satisfy the requirements for Theorem 3. As such, the tangents of these solutions may only be used for the current iteration.

Algorithm 3: LP-relaxation cuts for (EMSP).

```
function GetLPTangents(f, K, A_k, c_k, γ, y^k, ϵ, M)
    p ← 0, UB_0 ← +∞, LB_0 ← f(y^k)
    L ← ∅.
    while (UB_p − LB_p)/LB_p > ϵ and p ≤ M do
        p ← p + 1
        Solve the continuous relaxation of
          max_{x,θ} { θ : x ∈ K_{c_k}, θ ≤ h(x, y), ∀y ∈ A_k ∪ L } with
          trust-region constraint (8) to obtain (x^p, θ^p)
        UB_p ← θ_p, LB_p ← max{LB_{p−1}, f(x^p)}, L ← L ∪ {x^p}
    end
    return L
end
```

3. Numerical results

We now present numerical results for Algorithms 1 and 2. These algorithms were implemented in Julia 1.10 using the JuMP mathematical programming package (Lubin et al., 2023) and Gurobi version 11.0 as the mixed-integer linear solver. The branch and cut method in Algorithm 2 utilised the *lazy constraint callback* function, enabling the addition of tangent planes as constraints during the branch and bound procedure. For Algorithm 2, we add LP-tangents by Algorithm 3 with

$\gamma = 0$, $0.5n$, or n and with a maximum iteration limit of $M = 100$. This results in four distinct solver configurations.

Our implementation's source code, including the raw results data, can be accessed at https://github.com/sandyspiers/EuclideanMaximisation/tree/v1.0-julia.[1] The results data also include tests using CPLEX 22.1.1 as the MIP solver, which produce similar results to those reported here. A relative termination tolerance of $\epsilon = 10^{-6}$ was used for the main iterations of Algorithms 1–3, as well as for any mixed-integer subproblems solved by Gurobi. All other mixed-integer programming parameters were set to their defaults. All tests were conducted on a machine with a 2.3 GHz AMD EPYC processor with 64 GB RAM, using a single thread.

The performance of the algorithms was evaluated against the well-known Glover linearisation of the objective function. This reformulation was first introduced in Glover (1975) and is given as

$$\begin{aligned} \max \quad & \sum_{i=1}^{n-1} w_i, && (9)\\ \text{s.t.} \quad & x \in P \cap \{0,1\}^n, \\ & w_i \le x_i \sum_{j=i+1}^{n} q_{ij}, \quad 1 \le i \le n-1, \\ & w_i \le \sum_{j=i+1}^{n} q_{ij} x_j, \quad 1 \le i \le n-1, \\ & w_i \ge 0, \qquad 1 \le i \le n-1. \end{aligned}$$

This formulation was shown in Martí et al. (2010) to be effective for diversity-sum problems and was later used as the exact solver for the comprehensive empirical analysis presented in Parreño et al. (2021) and Martí et al. (2022). In addition to (9), we solve (EMSP) using the mixed-integer quadratic programming solver available within Gurobi.

3.1. Capacitated diversity problem

We begin by evaluating the performance of the different solution methods for solving the capacitated diversity problem. In this problem, the constraint set P contains only the following knapsack constraint,

$$\sum_{i=1}^{n} c_i x_i \le b,$$

where $c_i \in \mathbb{R}_+$ $(i = 1,\dots,n)$, and $\min_{i=1,\dots,n} c_i \le b < \sum_{i=1}^n c_i$. As such, (EMSP) then becomes the problem of selecting a subset of pre-defined locations, each with a weight, to maximise the sum of the pairwise distances while keeping the total weight less than or equal to a given limit. The capacitated diversity problem belongs to the family of diversity problems, which have a wide variety of practical applications, including facility location, social network analysis, and ecological conservation (Lu et al., 2023; Lai et al., 2018; Peiró et al., 2021).

The test instances used are derived from the publicly available MDPLIB 2.0[2] test library (Martí et al., 2021). Within this test library, we use the Euclidean instances of the capacitated diversity problem. This includes 10 instances each of sizes 50, 150, and 500. These instances were generated such that the weight of each node was randomly generated in the range $[1, 1000]$, with the capacity set to $b = 0.2\sum_{i=1}^n c_i$ and $b = 0.3\sum_{i=1}^n c_i$, making 60 instances in total.

In addition to the previous publicly available test sets, we randomly generated some larger instances of the capacitated diversity problem. These instances are made up of either 1000, 1500, 2000, 2500, or 3000 nodes, where each node contains either 2, 10, or 20 coordinates. Each coordinate of a location was uniformly randomly generated in the range $[0, 100]$. The weight of each node was uniformly randomly generated

Table 1
Average solve time in seconds of the various solver setups, broken down by test set and test size. Each problem is solved with a time limit of 600 s, using a single thread.

Type	n	Repeated	Forced cardinality			Glover	Quadratic
		(ILP_A)	$\gamma = 0$	$\gamma = 0.5n$	$\gamma = n$	linearisation	programming
CDP	50	0.21	0.05	0.16	0.15	0.40	196.52
	150	0.08	0.04	0.10	0.10	12.59	600.00
	500	0.15	0.09	0.14	0.14	245.49	600.01
RCDP	1000	0.58	0.64	0.63	0.58	–	–
	1500	0.68	1.93	1.75	1.71	–	–
	2000	0.80	3.88	3.36	3.37	–	–
	2500	0.54	12.48	7.09	6.94	–	–
	3000	0.59	12.55	9.35	9.16	–	–
GDP	50	0.25	0.19	0.14	0.05	0.21	0.13
	150	0.39	0.08	0.11	0.11	4.01	3.67
	500	0.36	0.32	0.39	0.39	36.43	600.01
RGDP	1000	0.80	1.83	1.73	1.69	–	–
	1500	0.93	4.58	4.32	4.29	–	–
	2000	1.22	10.15	7.72	7.62	–	–

in the range $[1, 1000]$, and the capacity was set to $b = 0.2\sum_{i=1}^n c_i$ or $b = 0.3\sum_{i=1}^n c_i$. For every combination of the number of nodes and the number of coordinates, we generated 5 instances, comprising a total of 150 test instances in total.

The performance of various solvers for the benchmark problem instances (labeled CDP) and randomised problem instances (labeled RCDP) over a 600-s time limit is displayed in Figs. 2 and 3, respectively. For CDP test instances, Algorithms 1 and 2 exhibit similar performance, both efficiently solving the entire test set within a maximum of 3.14 s. This represents a substantial improvement compared to the other exact methods. While Glover linearisation can solve some of the smaller instances within 5 s of run time, it fails to solve the entire problem set within the 600-s time limit. Solving the problem in its original quadratic form proved to be the worst method by a significant margin, solving fewer than 20 instances. The use of LP-tangents did not appear to improve Algorithm 2 and, in fact, worsened the runtime slightly.

On the larger RCDP test instances, we continue to see impressive performance from both Algorithms 1 and 2. Remarkably, even with the immense size of these instances, the repeated (ILP_A) method was still able to solve all instances in under 5 s. We also begin to see a difference in performance between the two algorithms, with the forced cardinality method beginning to perform worse for these very large problem sizes. That said, it is still able to solve all instances within 60 s of run time.

The results from these tests are further summarised in Table 1, which shows the average solve time for each test set, broken down by problem size. It clearly shows how, on the capacitated diversity problem, the performance of Algorithm 1 remains stable for increasing problem size.

3.2. Generalised diversity problem

The generalised diversity problem (GDP) represents a fundamental optimisation problem in the fields of facility location, supply chain management, and network design (Martinez-Gavara et al., 2021). At its core, the GDP seeks to strategically position a set of facilities on a network to efficiently serve a given demand distribution. This entails optimising not only the allocation of facilities to locations but also considering the spread of these facilities. The max-sum GDP is given as

$$\begin{aligned} \max \quad & f(x) && \text{(GDP-f)}\\ \text{s.t.} \quad & \sum_{i=1}^{n} c_i x_i \ge B, \\ & \sum_{i=1}^{n} a_i x_i \le K, \end{aligned}$$

[1] Commit reference b921170.
[2] Available at https://www.uv.es/rmarti/paper/mdp.html.

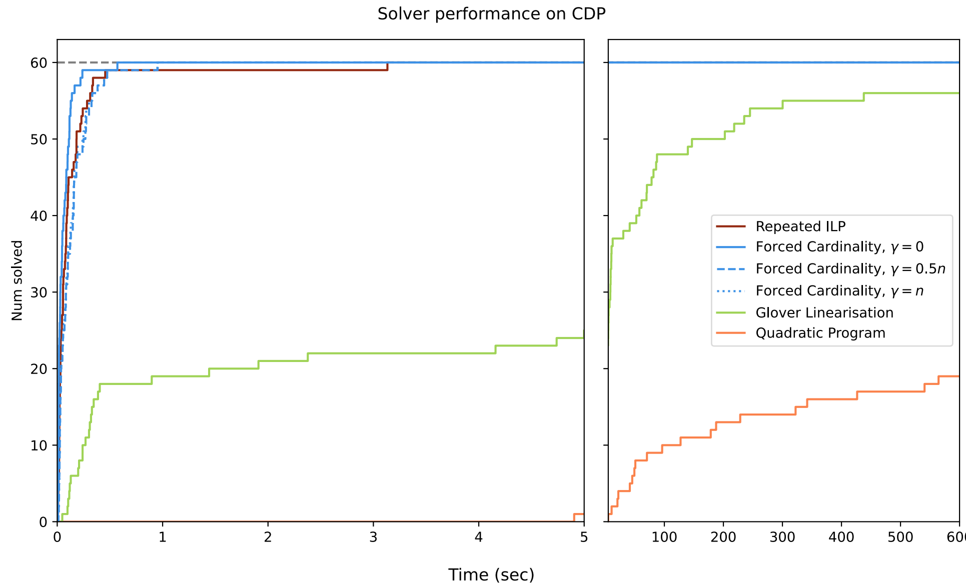

Fig. 2. Solver performance on the 60 capacitated diversity problem instances available within the `MDPLIB 2.0` test library, using a single thread for computation. The time axis is split at 5 s due to marked differences in solver performance.

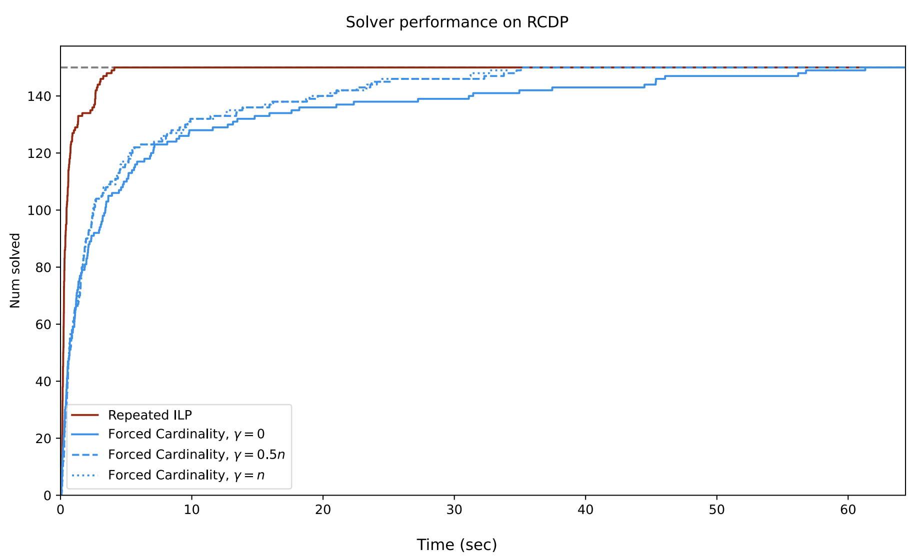

Fig. 3. Solver performance on the 150 randomly generated capacitated diversity problem instances.

$$x_i \in \{0,1\}, \quad i = 1, \dots, n,$$

where c_i and a_i represent the capacity and cost of site i. Sites must be chosen such that the minimum demand B is met, and setup cost is kept below the maximum K. The formulation in (GDP-f) considers the capacity to be fixed once a facility is open. A more realistic model considers variable setup costs, where extra capacity can be achieved at a given cost, once the facility is open. The variable cost version of the GDP is given as

$$\begin{aligned} \max \quad & f(x) && \text{(GDP-v)} \\ \text{s.t.} \quad & \sum_{i=1}^{n} t_i \geq B, \\ & \sum_{i=1}^{n} \left(a_i x_i + b_i t_i\right) \leq K, \\ & t_i \leq c_i x_i, \quad i = 1, \dots, n, \\ & t_i \in \mathbb{Z}, x_i \in \{0,1\}, \quad i = 1, \dots, n. \end{aligned}$$

We note that (GDP-f) and (GDP-v) were first introduced in Martinez-Gavara et al. (2021) where the objective was to maximise the minimum distance, however for our purposes we have changed this objective to maximise the sum of pairwise distances.

For the GDP, we again use the Euclidean test instances available within the `MDPLIB 2.0` test library on the (GDP-v) model. All parameters were uniformly randomly generated as follows. The capacity c_i was generated in the range $[1, 1000]$, the fixed cost a_i in the range $[c_i/2, 2c_i]$ and finally the variable cost b_i in the range $[\min\{1, a_i\}/100, \max\{1, a_i\}/100]$. The minimum capacity is set at either $B = 0.2 \sum_{i=1}^{n} c_i$ or $B = 0.3 \sum_{i=1}^{n} c_i$. Finally, the maximum budget is set as $K = \phi \sum_{i=1}^{n} \left(a_i + b_i c_i\right)$, where $\phi = 0.5$ or $\phi = 0.6$. As before, there are 10 instances each of size 50, 150 and 500, making a total of 120 test instances.

To test the solution algorithms on a larger scale, we generated several large instances of (GDP-v). These instances were generated similarly to the method described above; however, we increased the number of locations to 1000, 1500, and 2000 and generated locations with 2, 10, and 20 sets of coordinates. Furthermore, to reduce the

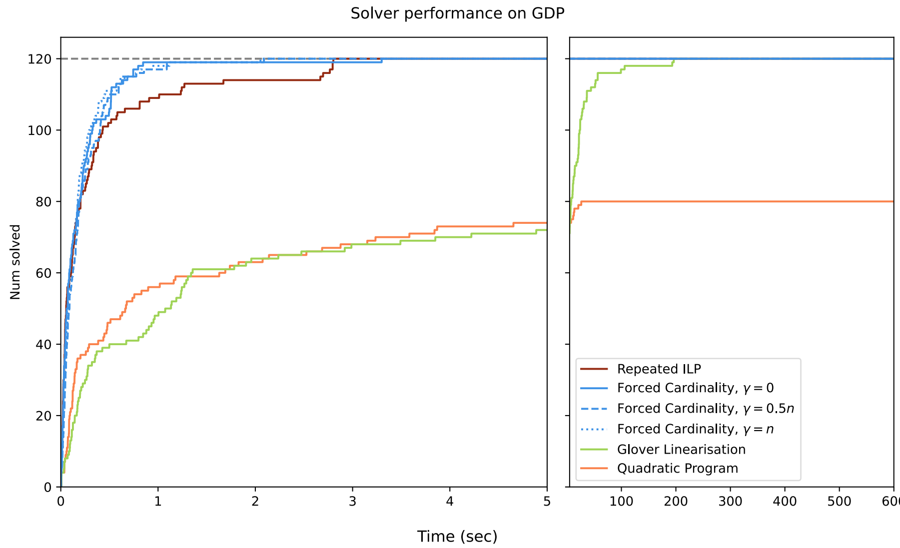

Fig. 4. Solver performance on the 120 variable cost generalised diversity problem instances within the `MDPLIB 2.0` test library. The time axis is split at 5 s due to marked differences in solver performance.

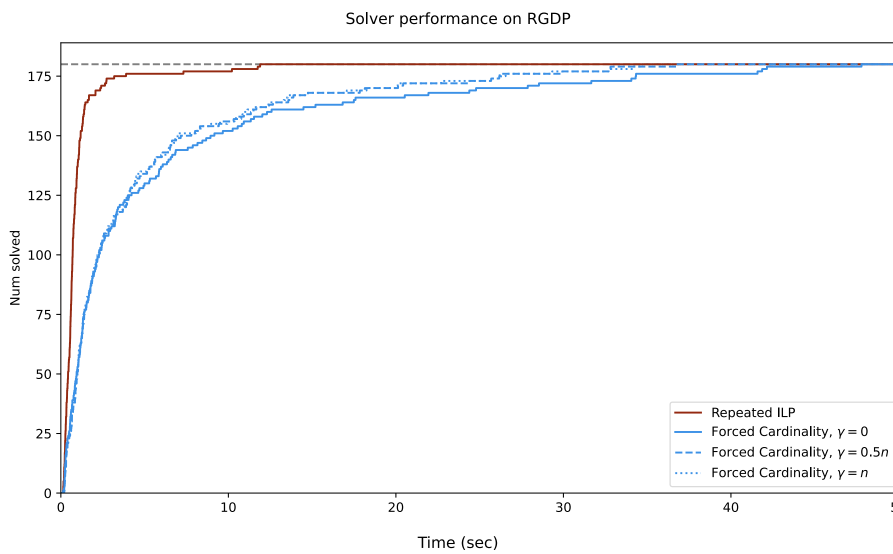

Fig. 5. Solver performance on the 180 randomised variable cost generalised diversity problem instances.

cardinality of the optimal solution and avoid the possibility of full solutions, we decreased the minimum capacity and maximum budget parameters such that $B = 0.05 \sum_{i=1}^{n} c_i$ or $B = 0.1 \sum_{i=1}^{n} c_i$ and $K = \phi \sum_{i=1}^{n} (a_i + b_i c_i)$, where $\phi = 0.1$ or $\phi = 0.2$. For every combination of the number of nodes, the number of coordinates, minimum capacity, and maximum budget, we generated 5 instances, comprising a total of 180 test instances in total.

The performance of different solver setups for the benchmark instances (labeled GDP) and random instances (labeled RGDP) over a 600-s time limit is displayed in Figs. 4 and 5, respectively. For GDP instances, Algorithms 1 and 2 exhibit similar performance, both efficiently solving nearly the entire set within 4 s. The incorporation of LP-tangent planes for Algorithm 2 has a negligible effect on its solve time. However, both Glover linearisation and the quadratic programming approach find this test set comparatively easier than the capacitated diversity problem, as the Glover linearisation model can solve over half the instances within five seconds and the full set in under 200 s. Turning to the results of the RGDP instances shown in Fig. 5, the repeated (ILP_A) method continues to outperform other solver setups. Moreover, the results suggest that introducing LP-tangent planes can marginally improve Algorithm 2 at large problem sizes. A summary of solve times for these larger instances is provided in Table 1.

3.3. Cardinality and cut strength

To gain a deeper insight into the strength of the cutting planes, we present a breakdown of the number of each type of cut added in Fig. 6. The figure shows the number of integer cuts (defined by points in A_k) and LP-tangents (defined by points in L_k) across the four solver setups for the CDP, RCDP, GDP, and RGDP test instances. Interestingly, the repeated (ILP_A) method consistently outperforms the forced cardinality method in almost all test sets, despite the latter introducing significantly more cutting planes since all intermediate feasible solutions are used to generate cuts. This suggests that by solving (ILP_A) to optimality, the cut generated provides a very tight approximation of the objective function at the optimal solution. Therefore, in many cases, it is worth taking the extra time to solve the (ILP_A) subproblem to optimality, as the cut generated is expected to be tight. This also explains why the addition of LP-tangent planes does not seem to provide much computational benefit to either approach. As these cuts are generated

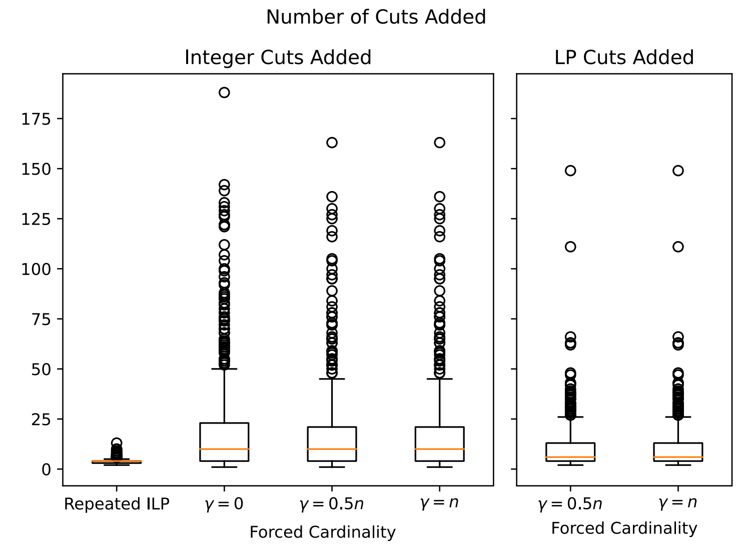

Fig. 6. Breakdown of the number of integer- and LP-tangents added across the CDP, RCDP, GDP and RGDP test instances.

on the continuous relaxation, they are expected to be even further away from the optimal solution than any integer solution, and hence provide a worse approximation. Tightening the trust region by decreasing γ also seems to have little effect. While LP-tangents are easy to generate and can therefore introduce a large number of cuts, they do not provide a good approximation of the objective function, and hence they do not substantially reduce the number of integer tangents required.

One example of where LP-tangents become highly beneficial is in problems that have a large difference between the maximum cardinality and the cardinality of an optimal solution. In such cases, the forced cardinality approach must solve many iterations before reaching an iteration that contains an optimal solution. Fig. 7 shows the minimum, average, and maximum solve time at each difference between the maximum cardinality and the cardinality of the found solution across the CDP, RCDP, GDP, and RGDP test instances. The repeated (ILP$_A$) method is virtually unaffected by this metric, and its average runtime remains steady. However, the forced cardinality method performs substantially worse as this number increases. That said, LP-tangents appear to improve the performance by quickly solving earlier iterations, thereby reducing overall solve time. As such, they become fairly beneficial in these cases.

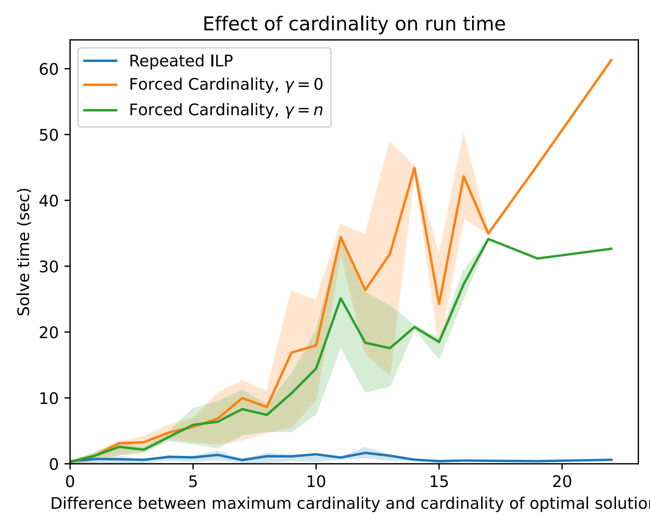

Fig. 7. Average solve time based on the difference between the maximum cardinality and cardinality of the optimal solution, across the CDP, RCDP, GDP and RGDP test instances. The shaded regions denote the range bounded by the minimum and maximum solve times.

3.4. Multi-threaded tests

While all tests mentioned thus far use a single thread for computation to provide a fair test setup, this is rarely required in practice. As such, we now revisit test sets CDP and GDP, allowing the solver to use all 16 available threads. Note that for Algorithms 1–3, the main loop iterations are still single-threaded, but the mixed-integer solver may now use all threads to solve the required subproblems.

The results on sets CDP and GDP are shown in Figs. 8 and 9 respectively. Given their already short runtimes, the performances of Algorithms 1 and 2 do not improve substantially. This is partly explained by the fact each subproblem is easy to solve, and hence do not benefit greatly from parallelism. For Glover linearisation, the performance difference within the first 5 s is marginal, solving only a few extra instances in each case. However, after this time frame, the solver benefits greatly and sees marked improvements, especially on the CDP instances. Finally, the quadratic programming approach benefits the most from parallelism, allowing it to perform comparably with Glover linearisation on the GDP instances. That said, the cutting plane algorithms remain the best performers on each test set.

3.5. Max-sum diversity problem

We finish this section by looking at difficult instances of the max-sum diversity problem. This is similar to the capacitated diversity problem visited earlier, except the knapsack constraint is replaced by the following cardinality constraint,

$$\sum_{i=1}^{n} x_i = p.$$

The problem has many real-world applications and fits the structure of (EMSP). While this problem can be efficiently solved using the cutting plane approach in Spiers et al. (2023), we can use this problem to test the robustness of Algorithms 1 and 2. Notably, we showed in Spiers et al. (2023) that instances with a large number of coordinates are particularly difficult to approximate by cutting planes, thereby resulting in poor performance. As such, we use the test instances in set `GKD-c` of `MDPLIB2.0`. These 20 instances each contain 500 locations with 20 coordinates, and where $p = 50$.

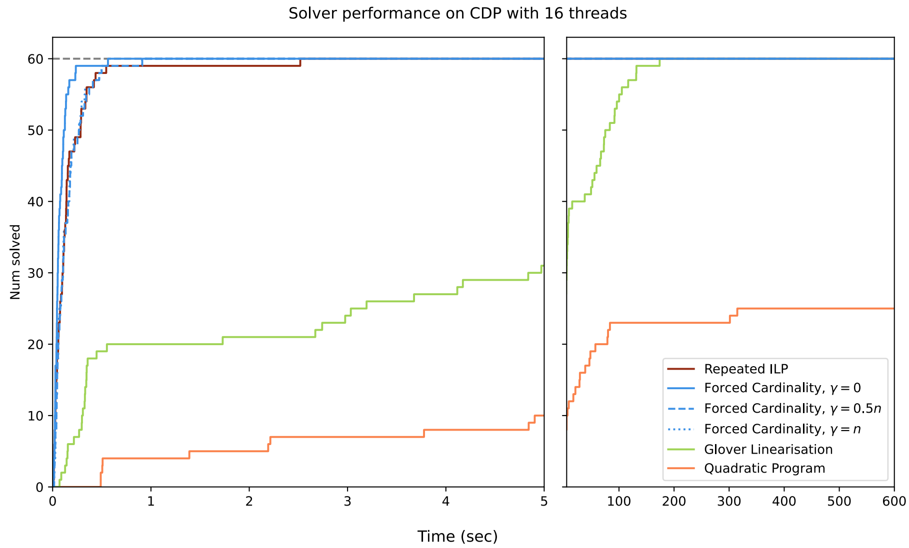

Fig. 8. Solver performance on the 60 capacitated diversity problem instances available within the MDPLIB 2.0 test library, using all 16 threads for computation.

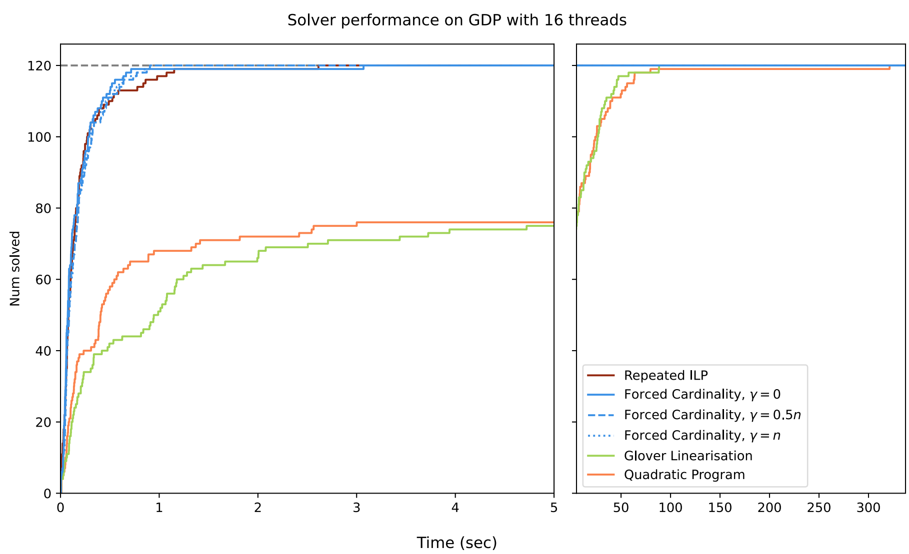

Fig. 9. Solver performance on the 120 generalised diversity problem instances available within the MDPLIB 2.0 test library, using all 16 threads for computation.

Table 2 shows the average gap as a percentage, average objective value, average number of integer cuts added, and the number of problems solved to optimality on the GKD−c test set over a 600-s time limit. While Algorithm 1 is only able to solve 3 out of 20 instances, the average final gap is very small at just 0.07%. Furthermore, it is able to achieve this gap with an average of only 141 cuts. In contrast, the forced cardinality approach is only able to solve a single instance to optimality. For the remaining instances, the algorithm is never able to reach its upper bounding subproblem (line 10, Algorithm 2) as it times out beforehand. Consequently, the algorithm never determines a valid upper bound. This represents a major shortcoming of this approach. That said, it can still achieve a decent lower bound, close to that of Algorithm 1.

4. Conclusion and future work

In this paper, we present two exact cutting plane algorithms for the general Euclidean distance maximisation problem. We establish the validity of tangents by introducing the concept of *directional concavity*. This notion led to the formulation of two important sufficient conditions for valid cuts, shown in Theorem 3. Two cutting plane solution algorithms were then introduced. The algorithms exploit Theorem 3 to ensure the search for the optimal solution always stays on a concave direction of the objective function, therefore ensuring all cuts are valid. This was achieved by either repeatedly solving the cutting plane subproblem to optimality, or by iteratively forcing and decreasing the cardinality of the problem.

Extensive numerical experiments were conducted to test the suggested solution algorithms. The results are very promising, with all proposed methods easily able to solve capacitated diversity problem instances with 3000 locations in under 60 s. This represents a significant improvement compared to other exact methods for the (EMSP). The repeated (ILP_A) method appeared to be the best overall performer as it generates tight cuts and remains fairly stable for increasing dimensions. Additionally, the approach is still able to provide a good upper bound even for very difficult instances, unlike the forced cardinality approach. Therefore, the choice of which approach to use should depend on the specific problem structure, especially the expected difference between the maximum cardinality and the cardinality of an optimal solution.

The identification of specific problem structures remains an important avenue for future research. We note a significant gap in the literature on the application of the (EMSP) to real-world problems.

Table 2
Solver performance on test set GKD-c using a single thread over a 600 s time limit.

	Repeated	Forced cardinality			Glover	Quadratic
	(ILP$_A$)	$\gamma = 0$	$\gamma = 0.5n$	$\gamma = n$	linearisation	programming
Ave gap (%)	0.07	∞	∞	∞	115.59	702.83
Ave objective value	19 500.83	19 490.72	19 482.78	19 482.78	18 985.52	19 501.70
Ave number integer cuts	141.45	3820.10	4015.80	4001.40	–	–
Number solved	3	1	0	0	0	0

The tests used here provide interesting conceptual frameworks and future work should expand on these results by applying the methods to real-world datasets and problems. In addition to identifying practical (EMSP) models, we should also attempt to identify difficult instances of these problems. In Spiers et al. (2023) we showed how the diversity problem becomes more challenging with a larger number of coordinates. That difficulty was not observed for the CDP or GDP problems, and hence more work is required to identify other difficult instances of the (EMSP). These problems can also help to understand and decide on which algorithm to use in which scenario.

CRediT authorship contribution statement

Hoa T. Bui: Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing, Supervision. **Sandy Spiers:** Conceptualization, Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing. **Ryan Loxton:** Conceptualization, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

The authors are supported by the Australian Research Council through the Centre for Transforming Maintenance through Data Science (grant number IC180100030). This work was supported by resources provided by the Pawsey Supercomputing Research Centre with funding from the Australian Government and the Government of Western Australia.

Appendix. Worked example

We now show the steps of Algorithm 1 through a small worked example. Suppose we are asked to solve a Euclidean max-sum problem of the form,

$$\max \quad f(x) = \frac{1}{2}\langle Qx, x\rangle, \tag{A.1}$$

$$\text{s.t.} \quad x \in P \cap \{0,1\}^n, \tag{A.2}$$

where Q is a symmetric hollow (zero diagonal) matrix and P is a polyhedral set. We must first confirm that Q is in fact a Euclidean distance matrix by using the *Schoenberg Criterion*. Let the $(n-1)\times(n-1)$ Gram matrix $G = [g_{ij}]_{i,j=2,\dots,n}$ be given by

$$g_{ij} = \frac{1}{2}\left(q_{1i} + q_{1j} - q_{ij}\right).$$

Then from Schoenberg (1935) we have that Q is a Euclidean distance matrix if and only if its Gram matrix G is positive semidefinite.

Provided Q is a valid Euclidean distance matrix, both Algorithms 1 and 2 are known to converge to an optimal solution of (A.1). To begin either algorithm, one must first solve the maximum cardinality problem

$$\max \quad \sum_{i=1}^{n} x_i,$$
$$\text{s.t.} \quad x \in P \cap \{0,1\}^n,$$

to generate a valid starting cut.

Algorithm 1 then begins by solving the outer-approximation subproblem given by

$$\begin{aligned} UB = \max \quad & \theta \\ s.t. \quad & \theta \le f(x^0) + \left\langle \nabla f(x^0), x - x^0 \right\rangle \\ & \theta \ge 0, \\ & x \in P \cap \{0,1\}^n, \end{aligned} \tag{A.3}$$

where x^0 is a solution of maximum cardinality. Let θ^1, x^1 be the solution to (A.3). Then $UB = \theta^1$ and $LB = \max\{f(x^1), f(x^0)\}$ provide valid upper and lower bounds for (A.1). The tangent plane of x^1 is then generated and used to formulate the next iteration of outer-approximation subproblems, given by

$$\begin{aligned} \max \quad & \theta \\ s.t. \quad & \theta \le f(x^0) + \left\langle \nabla f(x^0), x - x^0 \right\rangle \\ & \theta \le f(x^1) + \left\langle \nabla f(x^1), x - x^1 \right\rangle \\ & \theta \ge 0, \\ & x \in P \cap \{0,1\}^n. \end{aligned}$$

This problem is once again solved to optimality to generate a new solution θ^2, x^2. Bounds are again updated such that $UB = \theta^2$ and $LB = \max\{LB, f(x^2)\}$. The algorithm continues to repeat these iterations until $LB = UB$, at which point an optimal solution to (A.1) has been found.

References

Androulakis, I.P., Maranas, C.D., Floudas, C.A., 1995. αBB: A global optimization method for general constrained nonconvex problems. J. Glob. Optim. 7, 337–363. http://dx.doi.org/10.1007/BF01099647, 1995 7:4, Publisher: Springer.

Bapat, R.B., Raghavan, T.E.S., 1997. Conditionally positive definite matrices. In: Nonnegative Matrices and Applications. In: Encyclopedia of Mathematics and its Applications, Cambridge University Press, pp. 161–195. http://dx.doi.org/10.1017/CBO9780511529979.005.

Bliek, C., Bonami, P., Lodi, A., 2014. Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report. In: Proceedings of the Twenty-Sixth RAMP Symposium. pp. 16–17.

Bonami, P., Lee, J., Leyffer, S., Wächter, A., 2013. On branching rules for convex mixed-integer nonlinear optimization. J. Exp. Algorithmics (JEA) 18, 2.6:2.1–2.6:2.31.

Bonami, P., Lodi, A., Zarpellon, G., 2022. A classifier to decide on the linearization of mixed-integer quadratic problems in CPLEX. Oper. Res. 70 (6), 3303–3320. http://dx.doi.org/10.1287/opre.2022.2267.

Duran, M.A., Grossmann, I.E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. 36 (3), 307–339.

Erkut, E., Neuman, S., 1989. Analytical models for locating undesirable facilities. European J. Oper. Res. 40 (3), 275–291. http://dx.doi.org/10.1016/0377-2217(89)90420-7.

Glover, F., 1975. Improved linear integer programming formulations of nonlinear integer problems. Manage. Sci. 22 (4), 455–460. http://dx.doi.org/10.1287/mnsc.22.4.455.

Gupta, O.K., Ravindran, A., 1983. Nonlinear integer programming and discrete optimization. Trans. ASME J. Mech. Transm. Autom. Des. 105, 160–164.

Hochbaum, D.S., Liu, Z., Goldschmidt, O., 2023. A breakpoints based method for the maximum diversity and dispersion problems. In: SIAM Conference on Applied and Computational Discrete Algorithms (ACDA23). In: Proceedings, Society for Industrial and Applied Mathematics, pp. 189–200. http://dx.doi.org/10.1137/1.9781611977714.17, URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977714.17.

Kronqvist, J., Bernal, D.E., Grossmann, I.E., 2020. Using regularization and second order information in outer approximation for convex MINLP. Math. Program. 180 (1–2), 285–310.

Kronqvist, J., Lundell, A., Westerlund, T., 2016. The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. J. Global Optim. 64, 249–272.

Kuby, M.J., 1987. Programming models for facility dispersion: The *p*-dispersion and maxisum dispersion problems. Geogr. Anal. 19, 315–329. http://dx.doi.org/10.1111/J.1538-4632.1987.TB00133.X.

Lai, X., Yue, D., Hao, J.-K., Glover, F., 2018. Solution-based tabu search for the maximum min-sum dispersion problem. Inform. Sci. 441, 79–94.

Leyffer, S., 1993. Deterministic methods for Mixed-Integer Nonlinear Programming (Ph.D. thesis). Citeseer.

Lima, R.M., Grossmann, I.E., 2017. On the solution of nonconvex cardinality boolean quadratic programming problems: a computational study. Comput. Optim. Appl. 66 (1), 1–37.

Lloyd, S., 1982. Least squares quantization in PCM. IEEE Trans. Inf. Theory 28 (2), 129–137.

Lu, Z., Martínez-Gavara, A., Hao, J.-K., Lai, X., 2023. Solution-based tabu search for the capacitated dispersion problem. Expert Syst. Appl. 223, 119856. http://dx.doi.org/10.1016/j.eswa.2023.119856.

Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., Vielma, J.P., 2023. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. Math. Program. Comput. 15, 581–589. http://dx.doi.org/10.1007/s12532-023-00239-3.

Lubin, M., Yamangil, E., Bent, R., Vielma, J.P., 2018. Polyhedral approximation in mixed-integer convex optimization. Math. Program. 172 (1), 139–168.

Lundell, A., Kronqvist, J., Westerlund, T., 2022. The supporting hyperplane optimization toolkit for convex MINLP. J. Global Optim. 84 (1), 1–41.

MacQueen, J., et al., 1967. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol. 1, Oakland, CA, USA, pp. 281–297.

Madhulatha, T.S., 2012. An overview on clustering methods. arXiv:1205.1117.

Martí, R., Duarte, A., Martínez-Gavara, A., Sánchez-Oro, J., 2021. The MDPLIB 2.0 library of benchmark instances for diversity problems. URL https://www.uv.es/rmarti/paper/mdp.html.

Martí, R., Gallego, M., Duarte, A., 2010. A branch and bound algorithm for the maximum diversity problem. European J. Oper. Res. 200, 36–44. http://dx.doi.org/10.1016/J.EJOR.2008.12.023.

Martí, R., Martínez-Gavara, A., Pérez-Peló, S., Sánchez-Oro, J., 2022. A review on discrete diversity and dispersion maximization from an OR perspective. European J. Oper. Res. 299, 795–813. http://dx.doi.org/10.1016/j.ejor.2021.07.044.

Martinez-Gavara, A., Corberan, T., Marti, R., 2021. GRASP and tabu search for the generalized dispersion problem. Expert Syst. Appl. 173, 114703.

Parreño, F., Álvarez-Valdés, R., Martí, R., 2021. Measuring diversity. A review and an empirical analysis. European J. Oper. Res. 289 (2), 515–532.

Peiró, J., Jiménez, I., Laguardia, J., Martí, R., 2021. Heuristics for the capacitated dispersion problem. Int. Trans. Oper. Res. 28 (1), 119–141.

Schoenberg, I.J., 1935. Remarks to maurice Frechet's article sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de Hilbert. Ann. of Math. 36, 724–732. http://dx.doi.org/10.2307/1968654, Publisher: Annals of Mathematics.

Schoenberg, I.J., 1937. On certain metric spaces arising from euclidean spaces by a change of metric and their imbedding in Hilbert space. Ann. Math. 787–793.

Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y., 2015. A comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS One 10 (12), e0144059.

Spiers, S., Bui, H.T., Loxton, R., 2023. An exact cutting plane method for the euclidean max-sum diversity problem. European J. Oper. Res. http://dx.doi.org/10.1016/j.ejor.2023.05.014.

Vielma, J.P., Ahmed, S., Nemhauser, G.L., 2008. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. INFORMS J. Comput. 20 (3), 438–450.

Westerlund, T., Pettersson, F., 1995. An extended cutting plane method for solving convex MINLP problems. Comput. Chem. Eng. 19, 131–136.