Contents lists available at ScienceDirect

Computers and Operations Research

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

The max-out min-in problem: A tool for data analysis

Jorge Orestes Cerdeira [a,*], Maria João Martins [b], Marcos Raydan [c]

[a] Center for Mathematics and Applications (NovaMath) and Department of Mathematics, FCT NOVA, 2829-516, Caparica, Portugal
[b] Forest Research Center, School of Agriculture, University of Lisbon, Tapada da Ajuda 1349-017, Lisboa, Portugal
[c] Center for Mathematics and Applications (NovaMath), FCT NOVA, 2829-516, Caparica, Portugal

ARTICLE INFO

MSC:
90C27
05C22
90C20
68Q25
62H30

Keywords:
Combinatorial optimization
Weighted graphs
Quadratic programming
Computational complexity
Variable selection
Cluster analysis

ABSTRACT

Consider a graph with vertex set V and non-negative weights on the edges. For every subset of vertices S, define $\phi(S)$ to be the sum of the weights of edges with one vertex in S and the other in $V \setminus S$, minus the sum of the weights of the edges with both vertices in S. We consider the problem of finding $S \subseteq V$ for which $\phi(S)$ is maximized. We call this combinatorial optimization problem the max-out min-in problem (MOMIP). In this paper we (i) present a linear 0/1 formulation and a quadratic unconstrained binary optimization formulation for MOMIP; (ii) prove that the problem is NP-hard; (iii) report results of computational experiments on simulated data to compare the performances of the two models; (iv) illustrate the applicability of MOMIP for two different topics in the context of data analysis, namely in the selection of variables in exploratory data analysis and in the identification of clusters in the context of cluster analysis; and (v) introduce a generalization of MOMIP that includes, as particular cases, the well-known weighted maximum cut problem and a novel problem related to independent dominant sets in graphs.

1. Introduction

Variable selection, also known as feature selection, and cluster analysis are two major topics in data analysis. Variable selection consists in reducing the dimensionality of data sets while minimizing the loss of information (Cadima and Jolliffe, 2001; Jolliffe, 2002; Jolliffe and Cadima, 2016). More specifically, given a dataset consisting on the measurements of a number of variables, the purpose is to identify a subset of variables that adequately approximate the complete dataset. Several criteria have been proposed to quantify how well each subset approximates the whole dataset (Cadima et al., 2004; Cadima and Jolliffe, 2001). The goal of variable selection is to find a subset of variables that optimizes a particular criterion. This is a difficult optimization problem and, in general, these approaches have in common that the cardinality of the desired subset must be preset. Choosing the cardinality in advance is not always easy or convenient, and can lead to inadequate solutions.

Cluster analysis aims at identifying groups of observations such that the observations within each group are as similar as possible, while observations belonging to different groups are as different as possible. A number of different similarity/dissimilarity measures and different methods to find optimal clusters have been proposed (see, e.g., Pandove et al., 2018 for a recent survey). In general, the methods can be divided into hierarchical and partitioning, and usually rely on prior knowledge of the number of clusters for each data set to be clustered.

In this paper we introduce a combinatorial optimization problem that, given a graph with non-negative weights on the edges, looks for a set of vertices S that maximizes the sum of the weights of the edges connecting vertices in S with vertices outside S, while minimizes the sum of the edges having both vertices in S. We call this problem the *max-out min-in problem* (MOMIP). Some related graph-partition problems are the well-known weighted maximum cut (see, e.g., Shylo and Shylo, 2010), and the min–max cut (Ding et al., 2001). We discuss the similarities and differences between MOMIP and each of these problems in Section 2. We also show how MOMIP can be used as an approach for variable selection, without the need of knowing the cardinality of the subset of variables in advance, and for the identification of a cluster of observations that are similar to each other and dissimilar from observations outside the cluster.

The rest of this document is organized as follows. In Section 2, MOMIP is formalized, moving from a 0/1 linear model to a quadratic unconstraint binary optimization model. Section 3 focus on the computational complexity of MOMIP, and it is established that it is NP-hard. Section 4 reports computational results regarding the performances of the two models on simulated data. In Section 5, we show how MOMIP

[*] Corresponding author.
E-mail addresses: jo.cerdeira@fct.unl.pt (J.O. Cerdeira), mjmartins@isa.ulisboa.pt (M.J. Martins), m.raydan@fct.unl.pt (M. Raydan).

https://doi.org/10.1016/j.cor.2023.106218
Received 31 March 2022; Received in revised form 17 February 2023; Accepted 9 March 2023
Available online 11 March 2023

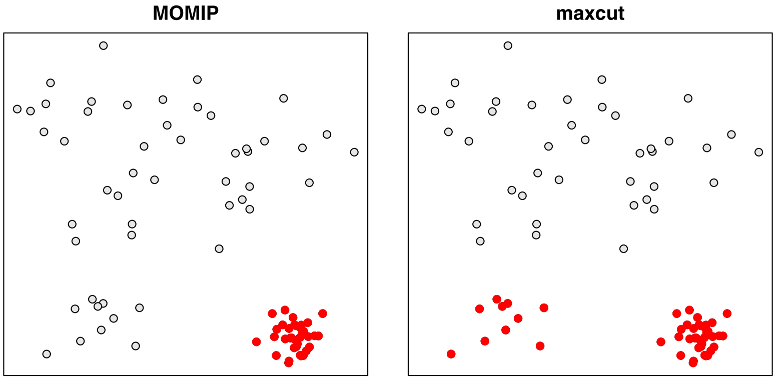

Fig. 1. Optimal partitions $(S, \bar{S})$ of MOMIP (left) and of the weighted maximum cut (right) w.r.t. the weighted graph using the Euclidean distances between all pairs of 75 points. The set S is represented by the (red) filled circles.

can be used in the context of variable selection and cluster analysis, and illustrate its properties on some real data sets. Finally, in Section 6, we present conclusions, remarks, and perspectives.

2. Problem formulation

Let $G = (V, E, w)$ be a weighted graph, where V is the vertex set, E the edge set and w_{uv} is a non-negative weight associated to every edge $uv \in E$. If $u, v \in V$ are not linked by an edge, we set $w_{uv} := 0$. For every subset of vertices $S \subseteq V$, let $\phi(S)$ be the sum of the weights of edges with one vertex in S and the other in $\bar{S} = V \setminus S$ minus the sum of the weights of the edges with both vertices in the set S, i.e.,

$$\phi(S) = \sum_{u \in S, v \in \bar{S}} w_{uv} - \sum_{u,v \in S} w_{uv}.$$

We consider the problem of finding S that maximizes $\phi(S)$. We call this combinatorial optimization problem the max-out min-in problem (MOMIP). Thus, MOMIP searches for a subset of vertices S such that the summation of the weights of edges with one vertex in S and the other in $\bar{S}$ is maximized, while the summation of the weights of the edges with both vertices in the set S is minimized. If S is an optimal solution we say that $(S, \bar{S})$ is an optimal partition of MOMIP.

As mentioned above, MOMIP is related to the weighted maximum cut and to the min–max cut. In the weighted maximum cut the objective function to be maximized is

$$\phi_{\text{maxcut}}(S) = \sum_{u \in S, v \in \bar{S}} w_{uv}.$$

In the (weighted) min–max cut the objective function to be minimized is

$$\phi_{\text{minmaxcut}}(S) = \frac{\sum_{u \in S, v \in \bar{S}} w_{uv}}{\sum_{u,v \in S} w_{uv}} + \frac{\sum_{u \in S, v \in \bar{S}} w_{uv}}{\sum_{u,v \in \bar{S}} w_{uv}}.$$

Regarding the weighted maximum cut, note that maximizing $\phi_{\text{maxcut}}(S)$ is equivalent to minimize $\sum_{u,v \in S} w_{uv} + \sum_{u,v \in \bar{S}} w_{uv}$. I.e., the objective function of the weighted maximum cut identically weighs the weights of the edges between pairs of vertices in set S and the weights of the edges between pairs of vertices in $\bar{S}$. It thus differs from MOMIP objective function that accounts differently these two quantities. Fig. 1 illustrates the different behavior of the two models on the complete graph with weights on the edges given by the Euclidean distances between all pairs of 75 points.

In Section 6 we propose a generalization of MOMIP that includes the weighted maximum cut as a particular case.

Concerning the min–max cut, notice that in the expression of $\phi_{\text{minmaxcut}}(S)$, the first term on the right hand side mimics the objective function of MOMIP, in the sense that it maximizes (or minimizes) the numerator while minimizing (or maximizing) the denominator, whereas MOMIP uses plus and minus sign to add those two expressions instead of using a quotient. The main difference with MOMIP is that min–max cut has an additional quotient to the right hand side of the objective function. The summation of both quotients has the tendency to create a balanced partition, while MOMIP tends to produce unbalanced partitions. This difference is illustrated in Fig. 2 for a set of 20 points (the largest dimension for which we could solve min–max cut to optimality in a reasonable amount of computational time).

A 0/1 linear formulation

Consider 0/1 variables x_u associated to every $u \in V$, where $x_u = 1$ if $u \in S$, and $x_u = 0$ if $u \in \bar{S}$, and non-negative variables y_{uv} and z_{uv} associated to every edge $uv \in E$. Using these variables MOMIP can be formulated as

$$\text{maximize } \phi(S) = \frac{1}{2} \sum_{uv \in E} w_{uv}(y_{uv} - z_{uv}) \tag{1}$$

$$\text{subject to } y_{uv} \le x_u + x_v, \qquad uv \in E \tag{2}$$

$$y_{uv} \le 2 - (x_u + x_v), \qquad uv \in E \tag{3}$$

$$z_{uv} \ge x_u + x_v - 1, \qquad uv \in E \tag{4}$$

$$x_u \in \{0, 1\}, \qquad u \in V \tag{5}$$

$$y_{uv}, z_{uv} \ge 0, \qquad uv \in E \tag{6}$$

Constraints (5) and (6) define the ranges of variables x_u and y_{uv}, z_{uv}, respectively. Let $S = \{u \in V : x_u = 1\}$ and $\bar{S} = \{u \in V : x_u = 0\}$. Inequalities (2) and (3), together with (6), force y_{uv} to be zero when either $u, v \in S$ or $u, v \in \bar{S}$. Otherwise, $y_{uv} \le 1$. Inequalities (4) ensure that $z_{uv} \ge 1$ whenever $u, v \in S$. Otherwise, the only constraint on the value of z_{uv} is (6), i.e., $z_{uv} \ge 0$. Given that $w_{uv} \ge 0$, to maximize the objective function (1) the values of y_{uv} will be as large as possible, i.e., $y_{uv} = 1$, if $u \in S$ and $v \in \bar{S}$, or $u \in \bar{S}$ and $v \in S$, and the values of z_{uv} will be as low as possible, i.e., only when $u, v \in S$, $z_{uv} = 1$, otherwise $z_{uv} = 0$. Therefore, as expected, if S maximizes ϕ, the value of $\phi(S)$ is the sum of the weights of edges with one vertex in S and the other in $\bar{S}$, minus the sum of the weights of the edges with both vertices in the set S. Note that in (1) the summation accounts the weight of each edge twice.

In the next section we give an alternative formulation for MOMIP that only uses the variables x_u on the vertices.

A quadratic unconstrained binary optimization formulation

Using the variables x_u defined above, MOMIP can be reformulated as a 0/1 nonlinear optimization problem:

$$\text{maximize } \phi(S) = \frac{1}{2} \left(\sum_{u,v \in V} w_{uv} + \sum_{u,v \in V} w_{uv}(x_u + x_v - 1) - 3 \sum_{u,v \in V} w_{uv} x_u x_v \right) \tag{7}$$

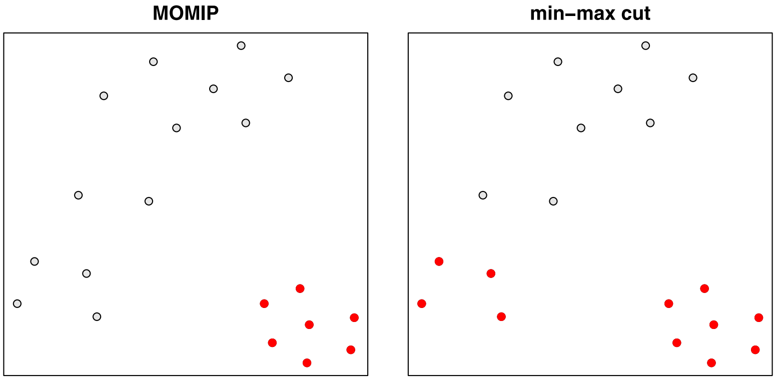

Fig. 2. Optimal partitions $(S, \bar{S})$ of MOMIP (left) and of the min–max cut (right) w.r.t. the weighted graph using the Euclidean distances between all pairs of 20 points. The set S is represented by the (red) filled circles.

subject to $\quad x_u \in \{0,1\}, \qquad u \in V.$

Let us recall that $S = \{u \in V : x_u = 1\}$ and $\bar{S} = \{u \in V : x_u = 0\}$, and let us interpret the objective function (7):

- $\phi_1(S) = \sum_{u,v\in V} w_{uv}$ is twice the sum the weights of all edges.
- $\phi_2(S) = \sum_{u,v\in V} w_{uv}(x_u + x_v - 1)$ is twice the sum of the weights of the edges with both vertices in S minus twice the sum of the weights of the edges with both vertices in $\bar{S}$.

Thus, $\phi_1(S) + \phi_2(S)$ equals four times the sum of the weights of the edges with both vertices in S, plus twice the sum of the weights of the edges with one vertex in S and the other in $\bar{S}$.

- $\phi_3(S) = -3\sum_{u,v\in V} w_{uv}x_u x_v$ is equal to minus six times the sum of the weights of the edges with both vertices in S.

Hence, $\phi(S) = \frac{1}{2}\big(\phi_1(S) + \phi_2(S) + \phi_3(S)\big)$ is the sum of the weights of the edges with one vertex in S and the other in $\bar{S}$, minus the sum of the weights of the edges with both vertices in the set S.

Using now that $x_u = x_u^2$ for every $u \in V$, and that $\sum_{u,v\in V} w_{uv}(x_u + x_v) = 2\sum_{u,v\in V} w_{uv}x_u$, we may work on the expression of $\phi(S)$,

$$\begin{aligned}\phi(S) &= \frac{1}{2}\left(\sum_{u,v\in V} w_{uv} + 2\sum_{u,v\in V} w_{uv}x_u - \sum_{u,v\in V} w_{u,v} - 3\sum_{u,v\in V} w_{uv}x_u x_v\right)\\ &= \frac{1}{2}\left(2\sum_{u,v\in V} w_{uv}x_u - 3\sum_{u,v\in V} w_{uv}x_u x_v\right),\end{aligned}$$

to obtain

$$\phi(S) = \sum_{u,v\in V} w_{uv}\, x_u^2 - \frac{3}{2}\sum_{u,v\in V} w_{uv}\, x_u x_v,$$

which allows us to formulate MOMIP as the following quadratic unconstrained binary optimization (QUBO) problem:

$$\text{maximize} \quad \phi(S) = x^\top \widehat{W} x \tag{8}$$

$$\text{subject to} \quad x \in \{0,1\}^{|V|}, \tag{9}$$

where the matrix $\widehat{W} \equiv [\hat{w}_{uv}]$, for $u, v \in V$ has the following entries:

$$\hat{w}_{uv} = \begin{cases} \sum_{s\in V} w_{us} & \text{if } u = v \\ -\frac{3}{2} w_{uv} & \text{if } u \neq v. \end{cases} \tag{10}$$

The matrix $\widehat{W}$ is symmetric and indefinite with positive diagonal. (We assume that, for every vertex $u \in V$, $\sum_{s\in V} w_{us} > 0$, since otherwise, i.e., if $\sum_{s\in V} w_{us} = 0$, vertex u could be removed.) To show that $\widehat{W}$ is indefinite, note that if $x_u = 1$, for some vertex $u \in V$, and $x_v = 0$ for every other vertex $v \in V \setminus \{u\}$, i.e., $S = \{u\}$, then

$$\phi(S) = x^\top \widehat{W} x = \sum_{s\in V} w_{us} > 0.$$

On the other hand, if $x_u = 1$, for every $u \in V$, i.e., $S = V$, then

$$\phi(S) = x^\top \widehat{W} x = \sum_{u,v\in V} \hat{w}_{u,v} = -\frac{1}{2}\sum_{u,v\in V} w_{u,v} < 0.$$

3. Computational complexity

We use the NP-hardness of Maximum 2-Satisfiability (Max 2-SAT) to prove that MOMIP is NP-hard. Given a collection of clauses, where each clause consists of two literals, Max 2-SAT asks for the maximum number of clauses satisfied by a true assignment to the variables. (The decision version of) Max 2-SAT is problem [LO5] in Garey and Johnson (1979) that has been proved to be NP-hard in Garey et al. (1976).

Consider an arbitrary instance of Max 2-SAT, i.e., a collection $\{c_1, c_2, \ldots, c_m\}$ of m clauses over a set of Boolean variables $B = \{x_1, x_2, \ldots, x_n\}$, where each clause c_i consists of exactly two literals. From this instance we construct a weighted graph $G = (V, E, w)$ as follows. Each variable x gives rise to a pair of vertices x and $\bar{x}$ corresponding to the two literals associated to x. (If t is a true assignment for B, $x = true$ if $t(x) = true$ and $\bar{x} = true$ if $t(x) = false$.) We add an edge (a red edge) linking x to $\bar{x}$. Each clause $c_i = (l, l')$ gives rise to a vertex c_i and we connect by an edge (a black edge) c_i to each of the two vertices that correspond to literals l and l' (see Fig. 3).

For every one of the clauses $c_i = (l, l')$, we define two P_3 (black edges) paths: (l, v_{i1}, v_{i2}, c_i) and $(l', v'_{i1}, v'_{i2}, c_i)$, and add an edge (a red edge) linking v_{i2} to v'_{i2}. Finally, we append to vertex v_{i2} and to vertex v'_{i2} the triangles $(v_{i2}, v_{i3}, v_{i4}, v_{i2})$ and $(v'_{i2}, v'_{i3}, v'_{i4}, v'_{i2})$, respectively (edges $v_{i2}v_{i3}$, $v_{i2}v_{i4}$, $v'_{i2}v'_{i3}$, $v'_{i2}v'_{i4}$ are red, edges $v_{i3}v_{i4}$ and $v'_{i3}v'_{i4}$ are green). Let C_i denote this graph (see Fig. 4).

Call G the graph resulting from combining these structures. For the weighting of the edges of G, (i) assign the same value $L > 0$ to (the red) edges $x\bar{x}$, $v_{i2}v_{i3}$, $v_{i2}v_{i4}$, $v'_{i2}v'_{i3}$, $v'_{i2}v'_{i4}$; (ii) assign a weight 5 to (the green) edges $v_{i3}v_{i4}$ and $v'_{i3}v'_{i4}$, and (iii) set a weight equal to 1 to all other edges (the black edges). The value L is set to be large enough to ensure that the two vertices of every edge that has weight L (every red edge) belong to different sets of all optimal partitions of MOMIP with respect to the weighted graph G. We are now ready to establish the time complexity of MOMIP.

Proposition 1. *MOMIP is NP-hard.*

Proof. Let us consider the collection of clauses $\{c_1, c_2, \ldots, c_m\}$, over the set of Boolean variables $B = \{x_1, x_2, \ldots, x_n\}$, i.e., an instance of Max 2-SAT. Let G be the weighted graph constructed from that instance as

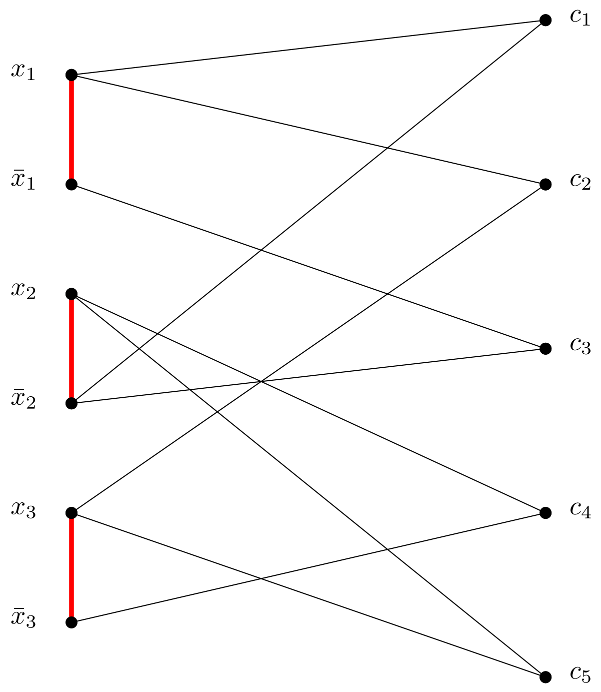

Fig. 3. The graph corresponding to the instance of Max 2-SAT consisting of clauses $c_1 = (x_1, \bar{x}_2)$, $c_2 = (x_1, x_3)$, $c_3 = (\bar{x}_1, \bar{x}_2)$, $c_4 = (x_2, \bar{x}_3)$, $c_5 = (x_2, x_3)$, over the set of Boolean variables $\{x_1, x_2, x_3\}$.

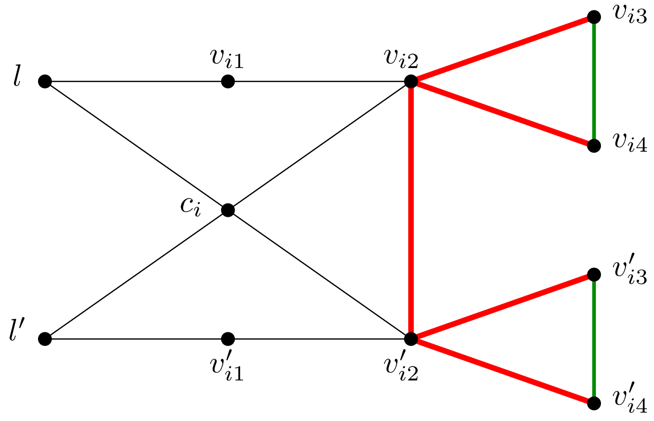

Fig. 4. The graph C_i corresponding to clause $c_i = (l, l')$.

described above, and let $(S, \bar{S})$ be a vertex optimal partition for MOMIP with respect to G.

Recall that a large weight $L > 0$ was assigned to the (red) edges $x\bar{x}$ so that in every optimal partition either $x \in S$ and $\bar{x} \in \bar{S}$, or else $x \in \bar{S}$ and $\bar{x} \in S$. This determines a one-to-one correspondence between true assignments t for B and the assignments of all pairs of vertices $x, \bar{x}$ of G to the two sets of the partition $S, \bar{S}$. We will assume that $\{x \in S \equiv t(x) = true\}$ ($\{x \in \bar{S} \equiv t(x) = false\}$).

Consider now the graph C_i associated to clause $c_i = (l, l')$ (see Fig. 4), and the three possibilities for the assignment of the vertices l, l' to $S, \bar{S}$: (i) $l, l' \in S$; (ii) $l \in S$, $l' \in \bar{S}$ (or $l \in \bar{S}$, $l' \in S$); and (iii) $l, l' \in \bar{S}$. If cases (i) and (ii) hold then clause c_i is satisfied, whereas case (iii) indicates that c_i is not satisfied.

Clearly, an optimal partition for G has to be optimal for every C_i. We claim that the optimal values for C_i are $1 + 5L$, if (i) or (ii) hold and $0 + 5L$, in case (iii) holds.

Since the weights of (red) edges $v_{i2}v_{i3}$, $v_{i2}v_{i4}$, $v_{i2}v'_{i2}$, $v'_{i2}v'_{i3}$, $v'_{i2}v'_{i4}$ are all equal to L, in every optimal partition, then exactly one vertex of each of these edges will belong to S. Hence, either $v_{i2}, v'_{i3}, v'_{i4} \in S$ and $v'_{i2}, v_{i3}, v_{i4} \in \bar{S}$, or $v'_{i2}, v_{i3}, v_{i4} \in S$ and $v_{i2}, v'_{i3}, v'_{i4} \in \bar{S}$. In both cases an edge with weight equal to 5 (a green edge) will have both vertices belonging to S, thus counting -5 to the optimal value.

In case (i) holds, there are two alternative optimal partitions for C_i: l, l', v_{i2}, $v'_{i3}, v'_{i4} \in S$ and the other vertices in $\bar{S}$, or $l, l', v'_{i2}, v_{i3}, v_{i4} \in S$ and the other vertices in $\bar{S}$. Since the (black) edges $c_i l$, $l v_{i1}$, $v_{i1}v_{i2}$, $v_{i2}c_i$, $c_i l'$, $l'v'_{i1}$, $v'_{i1}v'_{i2}$, $v'_{i2}c_i$ have all weights equal to 1, the optimal value for C_i will be $1 + 5L$.

In case (ii) holds, if $l \in S$ (and $l' \in \bar{S}$), the optimal partition of C_i is $l, v'_{i1}, v_{i2}, v'_{i3}, v'_{i4} \in S$ and the other vertices in $\bar{S}$. If $l \in \bar{S}$ (and $l' \in S$), the optimal partition is $l', v_{i1}, v'_{i2}, v_{i3}, v_{i4} \in S$ and the other vertices in $\bar{S}$. Here, as in the previous case, the optimal value for C_i is $1 + 5L$.

In case (iii) holds, there are two alternative optimal partitions for C_i: c_i, v'_{i1}, v_{i2}, $v'_{i3}, v'_{i4} \in S$ and the other vertices in $\bar{S}$, or $c_i, v_{i1}, v'_{i2}, v_{i3}, v_{i4} \in S$ and the other vertices in $\bar{S}$. Either way, the optimal value for C_i is $5L$.

Therefore, the value of an optimal partition of the vertices of G is $nL + s + 5mL$, where n is the number of variables, m is the number of clauses, and s is the maximum number of clauses satisfied by a true assignment, i.e., s is the correct answer to the Max 2-SAT problem, and the result follows. □

4. Performance comparison between the two models

This section reports computational experiments to compare the performances of the 0/1 linear programming (0/1 LP) formulation (1)–(6) and the quadratic unconstrained binary optimization (QUBO) formulation (8)–(9).

All computations were performed in R (R Core Team, 2020) on a desktop computer with CPU Intel Core i7–9700, 3.00 GHz, 8 Cores, RAM memory of 16 GB, running Windows 10 Enterprise 64–bits. For solving MOMIP we use the `gurobi` function from package `gurobi` (Gurobi Optimization, LLC, 2022).

We generated data in two different ways. One by assigning weights uniformly in $[0, 1]$ to the edges of random graphs with n vertices, where there is an edge linking each pair of vertices with probability p. We considered $n = 50, 75, 100, 150$ and $p = 0.1, 0.2, 0.5$. For each n and each p, we generated three weighted graphs. Thus, a total of 36 instances were obtained. The main results for this *graphs dataset* are presented in Table 1.

In the other dataset the weights are the Euclidean distances between pairs of n points (vertices) in $\mathbb{R}^2$ generated as follows. Two points c_1 and c_2 were located at distance d from each other. Then, $\theta \times n$ $(0 < \theta < 1)$ points were generated from a bivariate normal distribution centered at c_1 with standard deviation 0.05, and the remaining $(1-\theta) \times n$ points were generated from a bivariate normal distribution centered at c_2 with standard deviation 0.1. We considered $\theta = 0.1, 0.3, 0.5$ and $d = 0.25, 0.5, 1$. We first considered $n = 50, 100, 150$ and, for each combination (n, θ, d), we generated three configurations, thus producing a total of 81 weight matrices. Results obtained using this *points dataset* are displayed in Table 2.

In Tables 1 and 2, columns "objval" indicate the objective values of the computed solutions. Columns "time" give the execution CPU times, in seconds, when the search for a solution stopped before the imposed time limit of 10 min was reached by gurobi (≈ 4800 s since 8 cores are used), and "T" otherwise. Columns "gap (%)" indicate the percentage of optimality gap given by gurobi (i.e., $gap(\%) = \frac{ub - objvalue}{objvalue} \times 100$, where ub is an upper bound on the optimal value). When a solution is produced within the time limit, the solution is optimal and the gap is zero.

In general the QUBO formulation performed better than the 0/1 LP. Whenever 0/1 LP reached a solution within the time limit, QUBO also

Table 1
Computational execution times (in seconds), objective values and optimality gaps for the two models over the graphs dataset.

n	*p*	run	QUBO			0/1 LP		
			time	objval	gap (%)	time	objval	gap (%)
50	0.1	1	0.14	47.96	0	0.26	47.96	0
		2	0.24	51.98	0	0.25	51.98	0
		3	0.11	48.62	0	0.34	48.62	0
	0.2	1	1.11	77.18	0	2.23	77.18	0
		2	0.87	83.89	0	2.50	83.89	0
		3	0.76	75.00	0	2.80	75.00	0
	0.5	1	26.18	160.25	0	1348.56	160.25	0
		2	46.49	157.79	0	1868.34	157.79	0
		3	47.81	150.35	0	1559.07	150.35	0
75	0.1	1	0.99	92.39	0	3.22	92.39	0
		2	1.12	101.40	0	3.49	101.40	0
		3	1.61	102.29	0	5.21	102.29	0
	0.2	1	11.62	167.80	0	127.20	167.80	0
		2	22.97	159.01	0	103.96	159.01	0
		3	17.75	167.86	0	129.58	167.86	0
	0.5	1	1555.58	346.06	0	T	336.78	28.40
		2	3783.27	329.57	0	T	327.87	24.26
		3	2359.46	333.88	0	T	324.27	28.11
100	0.1	1	52.60	171.53	0	1177.75	171.53	0
		2	58.95	164.54	0	215.71	164.54	0
		3	84.40	167.46	0	260.21	167.46	0
	0.2	1	1630.80	256.54	0	T	256.54	5.18
		2	4315.78	288.78	0	T	286.60	8.97
		3	1328.14	277.02	0	T	277.02	6.27
	0.5	1	T	566.39	18.35	T	551.67	39.19
		2	T	574.64	20.60	T	566.97	37.10
		3	T	580.21	21.16	T	570.04	36.42
150	0.1	1	T	334.63	7.58	T	332.90	15.92
		2	T	350.86	8.14	T	350.06	14.89
		3	T	361.05	5.76	T	351.25	15.88
	0.2	1	T	576.34	15.35	T	583.58	23.58
		2	T	579.47	12.51	T	559.22	27.13
		3	T	605.20	13.39	T	607.68	22.61
	0.5	1	T	1231.14	18.35	T	1216.79	65.86
		2	T	1222.90	18.43	T	1201.48	65.90
		3	T	1240.12	18.14	T	1210.32	66.40

found a solution and has been faster than the 0/1 LP. When both 0/1 LP and QUBO reached the 10 min time limit, only in two cases (Table 1, n = 150) the objective values of the solutions produced from the 0/1 LP are greater than those found with QUBO.

For the graphs dataset, the computation times increase with the number of vertices (n) and also as graphs become more dense (when p increases). Both models were unable to find a solution within the time limit when $n = 150$.

For the points dataset, the CPU times increase as the distance between the centers of the two groups (d) decreases. This is an indication that when a group is well separated, then the computation times are lower, and when the borderlines are less clear, the computation times are heavier.

From the three considered proportions of group size (θ), the computations were lighter when 30% of the points are in one group, and were heavier when a group has only 10% of points.

For all sets with 150 points and for most sets of 100 points, computation time limit were attained when using the 0/1 LP model. The performance on these sets was significantly better with QUBO. We set aside the 0/1 LP model, and proceeded assessing the QUBO behavior on larger points dataset. The main results for $n = 200, 500, 1000$ are given in Table 3. The columns of Table 3 have the same meaning as the corresponding columns in Table 2 (w.r.t. QUBO).

Only for $n = 200$ and for one instance with $n = 500$, the computations finished within the time limit. The pattern relating computational times with parameters θ and d are similar to that observed with less points. Except for three instances, the optimality gaps were less than 10%. Gaps were lower when one group has 30% of points and higher when one group has only 10% of points. In general, gaps decrease when a group is well separated, however this was not observed for $n = 1000$ and $\theta = 0.1$.

From these set of experiments we can conclude that the performance of QUBO clearly overcomes that of 0/1 LP, although QUBO failed to produce guaranteed optimal solutions (i.e., with zero gaps) for n = 500. The gaps indicate that the obtained solutions could be considered as good heuristic solutions. Furthermore, the gurobi program we are using is not specialized to solve QUBO problems. Specialized QUBO solvers have evolved in the last years (Kochenberger et al., 2014). If, as expected, QUBO problems can be solved in the near future with up to a million variables (see, e.g., Şeker et al., 2022), then our quadratic formulation could also be a convenient option for large-scale real data sets.

5. Two applications in data analysis

To give further insight into MOMIP, we illustrate its applicability on two different areas of data analysis: variable selection and clustering. In what follows, X is a numeric matrix representing a data set consisting on the measurements of p variables (columns) in n individuals or objects (rows).

Variable selection

The first application consists of identifying a subset S of the p original variables that is optimal for a given criterion of adequate approximation to the complete data set. The goal is the reduction of dimensionality in the original data set, by discarding the *redundant* variables and retaining a set of *independent* variables.

Table 2
Computational execution times (in seconds) and objective values for the two models over the points dataset with number of points $n \leq 150$.

n	θ	d	run	QUBO			0/1 LP		
				time	objval	gap (%)	time	objval	gap (%)
50	0.1	1	1	2.39	234.98	0	34.88	234.98	0
			2	1.24	227.97	0	40.57	227.97	0
			3	0.87	226.51	0	33.08	226.51	0
		0.5	1	13.91	165.61	0	T	165.61	12.40
			2	1.49	143.13	0	2191.18	143.13	0
			3	2.62	151.01	0	T	151.01	6.89
		0.25	1	3.03	117.79	0	T	116.89	19.33
			2	2.01	119.43	0	T	119.43	16.50
			3	1.97	128.02	0	2304.05	128.02	0
	0.3	1	1	0.16	519.04	0	6 9.03	519.04	0
			2	0.05	513.86	0	6 24.17	513.86	0
			3	0.14	537.89	0	6 11.68	537.89	0
		0.5	1	0.11	272.95	0	6 161.28	272.95	0
			2	0.11	249.97	0	6 161.98	249.97	0
			3	0.11	286.22	0	37.02	286.22	0
		0.25	1	0.40	150.90	0	2847.4	150.90	0
			2	0.36	146.93	0	2913.23	146.93	0
			3	0.27	138.19	0	297.39	138.19	0
	0.5	1	1	0.08	565.05	0	11.14	565.05	0
			2	0.14	613.51	0	4.08	613.51	0
			3	0.06	591.09	0	11.22	591.09	0
		0.5	1	0.14	298.20	0	83.77	298.20	0
			2	0.22	288.48	0	48.16	288.48	0
			3	0.22	284.12	0	30.56	284.12	0
		0.25	1	0.98	140.69	0	245.72	140.69	0
			2	0.52	144.06	0	203.12	144.06	0
			3	0.40	149.23	0	214.86	149.23	0
100	0.1	1	1	7.99	920.36	0	T	584.46	128.89
			2	2.78	951.88	0	T	607.46	129.06
			3	4.90	962.49	0	T	700.07	108.89
		0.5	1	T	602.43	7.28	T	508.63	111.72
			2	T	583.23	6.39	T	551.35	79.61
			3	T	585.69	3.55	T	486.74	90.34
		0.25	1	T	493.17	6.72	T	352.17	138.83
			2	T	475.63	5.52	T	410.75	114.69
			3	T	486.22	5.81	T	327.18	149.39
	0.3	1	1	0.19	2101.93	0	3836.93	2101.93	0
			2	0.28	2052.02	0	2796.78	2052.02	0
			3	0.17	2147.71	0	2743.86	2147.71	0
		0.5	1	0.30	1038.40	0	T	1038.40	24.80
			2	0.55	1080.08	0	T	1080.08	28.26
			3	0.28	1050.16	0	T	873.06	49.13
		0.25	1	2424.37	570.36	0	T	532.79	61.74
			2	T	550.17	4.62	T	412.20	105.37
			3	3371.55	590.66	0	T	430.12	107.97
	0.5	1	1	0.86	2404.83	0	T	2404.83	4.22
			2	0.81	2367.40	0	2715.56	2367.40	0
			3	0.73	2476.18	0	4247.41	2476.18	0
		0.5	1	1.25	1197.36	0	T	1197.36	18.56
			2	1.39	1178.10	0	T	1178.10	18.94
			3	1.97	1155.23	0	T	1155.23	19.44
		0.25	1	T	624.03	4.98	T	620.39	44.27
			2	T	664.31	5.71	T	588.24	57.79
			3	3815.91	589.30	0	T	544.64	59.23
150	0.1	1	1	453.96	2089.85	0	T	1306.80	140.10
			2	411.43	2095.19	0	T	1249.77	143.72
			3	60.47	2109.87	0	T	1294.93	148.17
		0.5	1	T	1323.79	4.77	T	923.37	162.65
			2	T	1350.57	3.60	T	917.70	158.41
			3	T	1324.94	3.96	T	923.78	148.28
		0.25	1	T	1026.00	3.74	T	734.13	165.36
			2	T	1071.19	3.85	T	750.29	158.03
			3	T	1047.25	2.51	T	722.60	166.12
	0.3	1	1	0.46	4679.95	0	T	3169.10	72.32
			2	0.35	4538.99	0	T	2499.17	114.59
			3	0.28	4780.95	0	T	3091.86	77.88
		0.5	1	9.28	2330.01	0	T	1383.45	127.60
			2	2.58	2355.35	0	T	1503.34	120.42
			3	1.64	2428.40	0	T	1479.40	125.40

(*continued on next page*)

Table 2 (*continued*).

n	θ	d	run	QUBO			0/1 LP		
				time	objval	gap (%)	time	objval	gap (%)
		0.25	1	40.74	1280.65	0	T	1066.85	95.66
			2	T	1322.75	1.92	T	877.81	144.67
			3	T	1346.93	1.26	T	901.86	147.40
	0.5	1	1	6.52	5402.92	0	T	3764.95	58.05
			2	5.50	5538.78	0	T	3822.87	59.67
			3	5.34	5380.35	0	T	3701.91	61.43
		0.5	1	17.56	2576.17	0	T	1887.86	69.81
			2	143.09	2653.98	0	T	1963.08	76.17
			3	159.08	2639.57	0	T	1937.78	74.95
		0.25	1	T	1228.96	3.11	T	978.00	100.72
			2	T	1389.40	1.08	T	1081.08	94.93
			3	T	1257.07	4.19	T	926.84	116.32

Although the problem is multivariate in nature, we propose a criterion based on pairs of variables. Let us consider MOMIP with the graph whose vertices are the p variables and the weight associated to each edge uv is the square of the Pearson correlation between the variables u and v. The squared correlation quantifies the proportion of variability of one variable that is explained by a linear regression on the other. This corresponds to consider the weight matrix $W = [w_{uv}]$, defined in Section 2, as the $p \times p$ matrix with zero entries on the diagonal, and the squared Pearson correlation between columns u and v of matrix X as the off-diagonal elements.

An optimal solution of MOMIP provides a set S of variables that maximizes the sum of the squared correlations between each variable in S and each variable outside S and that minimizes the sum of the squared correlations between all pairs of variables in S. Let k be cardinality of the solution set S.

The results were compared with those obtained by a classic optimization algorithm implemented using the `eleaps` function of the R package `subselect` (Cerdeira et al., 2020). This function identifies, for an arbitrary $1 \leq k \leq p$, a k-subset of variables which is optimal with respect to a given criterion that measures and quantifies how well each subset approximates the whole data set. We consider the three following multivariate criteria (Cadima et al., 2004; Cadima and Jolliffe, 2001)

- RM: measures the similarity of the spectral decompositions of the p-variable correlation matrix, and of the matrix which results from regressing all the variables on a subset of only k variables.
- RV: measures the similarity (after rotations, translations and global resizing) of two configurations of n points given by: (i) observations on each one of the p variables, and (ii) the regression of those p observed variables on a subset of the variables.
- GCD: computes Yanai's Generalized Coefficient of Determination for the similarity of the subspaces spanned by a subset of variables and a subset of the full Principal Components data set.

For the set of k variables selected by MOMIP, and the sets with the same cardinality, k, selected by `eleaps`, we quantify the four criteria: MOMIP, RM, RV, and GCD.

Cluster analysis

The second application is a particular case of cluster analysis. The goal of clustering is to identify pattern or groups of similar individuals or objects within a data set. Cluster analysis seeks to partition a given data set into groups based on specified features so that the observations within a group are more similar to each other than the observations in different groups; see, e.g., Everitt et al. (2011).

The partition of observations into groups requires the definition of a distance or dissimilarity between each pair of observations. Common dissimilarity measures for numerical data are Euclidean and Mahalanobis distances (Gan et al., 2007). Other dissimilarity measures, such as correlation-based dissimilarities, are often used. These measures consider two observations similar if their features are highly correlated, even though the observed values may be far apart in terms of Euclidean distance. They are useful when one aims to identify groups of observations with the same general patterns, regardless of their individual magnitudes of each feature. This is particularly the case in the analysis of gene expression data (Jiang et al., 2004). When using Pearson's correlation, one can easily prove that the squared Euclidean distance between two standardized vectors (after subtracting the mean and dividing by the standard deviation) is equal to one minus the Pearson correlation coefficient divided by twice the dimension.

Clustering methods differ also on the definition of the distance between an object and a cluster and between two clusters, and in the clustering algorithm. Many conventional clustering algorithms are hierarchical: agglomerative when the two most similar groups are merged to form a large cluster at each step, or divisive when the process is reversed by starting with all data observations in one cluster and subdividing into smaller clusters. In both cases, an *optimal* number of clusters is needed.

We propose the following *1-cluster* method. Consider MOMIP with the graph whose vertices are the n observations and where the weight associated to each edge is the dissimilarity between the observations, defined by one minus the Pearson correlation. Notice that each observation is a p-dimensional vector. This corresponds to consider the weight matrix $W = [w_{uv}]$, defined in Section 2, as the $n \times n$ matrix with entry (u, v) equal to one minus the Pearson correlation between rows u and v of matrix X. An optimal solution of MOMIP provides a set S of observations that maximizes the sum of the dissimilarities between each observation in S and each observation outside S, and that minimizes the sum of dissimilarities between observations in S.

We consider four data sets, chosen with respect to different properties like number of data points, number of features and complexity of the classification task. The two procedures described above (variable selection and clustering) are applied to each data set.

Next, a brief description of each of the four considered data sets will be given, and the results obtained will be presented and analyzed. All computations were performed in R (R Core Team, 2020), using the computer described in Section 4.

Iris data set

The first example is the Iris flower dataset, one of the most popular multivariate datasets in pattern recognition literature. This dataset was introduced by Fisher in 1936 (Fisher, 1936) as an example of application of linear discriminant analysis.

It consists on measurements of $p = 4$ morphometric variables (length and width of sepal and petal) in 50 iris flowers of each of three species: Iris setosa, Iris virginica and Iris versicolor, thus making $n = 150$ observations. For additional details see the web page https://en.wikipedia.org/wiki/Iris_flower_data_set.

Table 3
Computational execution times (in seconds) and optimality gaps for the QUBO model over the points dataset with number of points $n \geq 200$.

QUBO					
n	theta	d	run	CPU	gap (%)
200	0.1	1	1	697.22	0
			2	T	0.89
			3	401.25	0
		0.5	1	T	5.03
			2	T	5.84
			3	T	5.80
		0.25	1	T	5.09
			2	T	5.49
			3	T	5.13
	0.3	1	1	0.79	0
			2	1.25	0
			3	0.84	0
		0.5	1	3.50	0
			2	13.50	0
			3	4.57	0
		0.25	1	T	2.17
			2	T	3.55
			3	T	3.58
	0.5	1	1	67.76	0
			2	148.83	0
			3	212.68	0
		0.5	1	1955.30	0
			2	504.80	0
			3	3678.32	0
		0.25	1	T	4.89
			2	T	4.70
			3	T	3.86
500	0.1	1	1	T	7.13
			2	T	7.67
			3	T	6.58
		0.5	1	T	8.33
			2	T	8.64
			3	T	8.50
		0.25	1	T	8.28
			2	T	7.52
			3	T	8.02
	0.3	1	1	T	0.70
			2	T	0.73
			3	2001.85	0
		0.5	1	T	3.11
			2	T	2.53
			3	T	2.24
		0.25	1	T	6.70
			2	T	6.13
			3	T	6.07
	0.5	1	1	T	2.67
			2	T	2.71
			3	T	2.90
		0.5	1	T	3.90
			2	T	3.39
			3	T	3.83
		0.25	1	T	8.22
			2	T	8.04
			3	T	7.85
1000	0.1	1	1	T	10.24
			2	T	12.21
			3	T	11.13
		0.5	1	T	8.37
			2	T	8.42
			3	T	8.80
		0.25	1	T	8.03
			2	T	8.48
			3	T	8.49
	0.3	1	1	T	1.81
			2	T	1.82
			3	T	1.79
		0.5	1	T	3.41
			2	T	3.27
			3	T	3.63
		0.25	1	T	7.26
			2	T	6.58
			3	T	6.01
	0.5	1	1	T	3.41
			2	T	3.66
			3	T	3.65
		0.5	1	T	5.04
			2	T	4.62
			3	T	4.38
		0.25	1	T	8.51
			2	T	9.25
			3	T	9.39

(continued on next page)

Table 3 (*continued*).

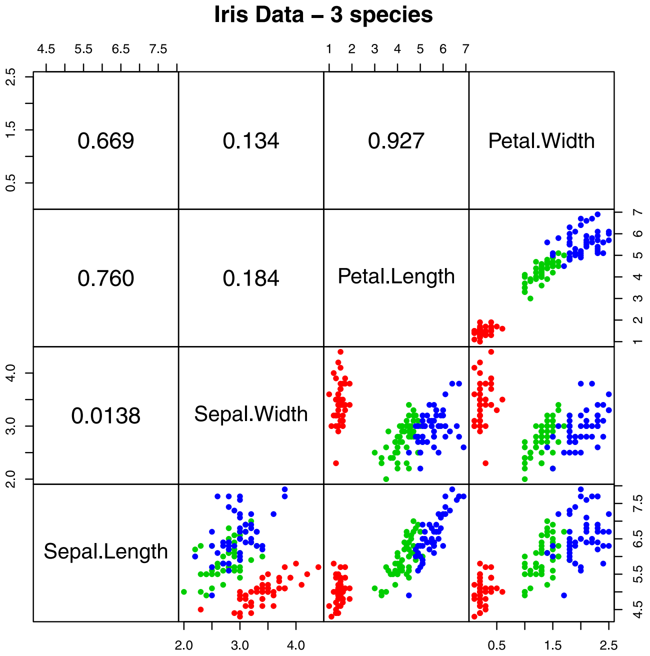

Fig. 5. Scatter plot of the Iris data set (red = Setosa, green = Versicolor, blue = Virginica) with the squared Pearson correlation coefficients.

Table 4
Values of each criterion for the common optimal solution with cardinality $k = 1$ for Iris data set.

MOMIP value	RM value	RV value	GCD value
1.8706	0.8471	0.9375	0.9832

In Fig. 5 we show the scatter plot of the data set, with the squared Pearson correlation coefficients between each pair of variables. When the squared correlation coefficients were used for the off-diagonal elements of W, the solution obtained by MOMIP selects $k = 1$ variable, the Petal Length. Now, using $k = 1$, the optimization algorithm implemented in function `eleaps` with each one of the three criterion (RM, TV, and GCD) selects exactly the same variable. The values of each criterion for the (common) optimal solution is presented in Table 4.

Note that RM, RV, and GCD take values between zero and one, and that both RM and RV increase with cardinality k.

When considering each iris flower as a 4-dimensional vector, and the square matrix W, with dimension $n = 150$, whose entry (u, v) is $1 - r_{uv}$ where r_{uv} is the Pearson correlation coefficient between irises u and v, the optimal solution of MOMIP is the set of irises of Setosa

Table 5
Comparison between MOMIP and the other 3 criteria for the Crayfish data set.

	MOMIP	RM	RV	GCD
MOMIP value	**12.508**	11.804	12.004	10.766
RM value	0.882	**0.905**	0.893	0.884
RV value	0.965	0.960	**0.965**	0.901
GCD value	0.500	0.763	0.579	**0.866**

species. Thus, the proposed *1-cluster* methodology was able to identify Setosa and correctly distinguish this species from the other two.

Crayfish data set

This dataset is taken from Somers (1986) and consists of measurements (in millimeters) of $p = 13$ morphometric variables for $n = 63$ crayfish collected in Lake Opeongo, Ontario. The 13 variables are: carapace length, tail length, carapace width, carapace depth, tail width, areola length, areola width, rostrum length, rostrum width, postorbital width, propodus length, propodus width, and dactyl length. Out of the 63 crayfish, 21 are female and 42 are male.

When applying MOMIP to select a subset of variables, the solution is the set of $k = 4$ morphometric variables: carapace length, tail length, carapace width and propodus length. The solution of `eleaps` algorithm with the same cardinality depends on the quality criterion. Criterion RM selected variables: carapace width, areola width, rostrum length and propodus length; RV determined: carapace width, tail length, rostrum length and propodus length, and criterion GCD selected: areola length, areola width, rostrum length and propodus length.

In order to compare the solutions obtained by the 4 criteria, each solution was evaluated by each of the 4 criteria. Results are presented in Table 5.

Table 5 shows that, when comparing the solution of MOMIP with the solutions of `eleaps` with the same cardinality, the scores of MOMIP are quite similar to the scores of the other 3 criteria.

The solution of the *1-cluster* method is the set of the female crayfish (21 observations) together with 2 males.

Seeds data set

This data set was taken from Dua and Graff (2019) and consists of values of the characteristics of the internal structure of the kernel of wheat seeds, detected by means of a soft X-ray technique. The research that produced those values was conducted using combined harvested wheat grain, originating from experimental fields, explored at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin. Seventy seeds from each of three wheat varieties (Kama, Rosa and Canadian) were analyzed. For each seed, seven geometric parameters of wheat kernels were measured: area A, perimeter P, compactness $C = 4 * pi * A/P^2$, length of kernel, width of kernel, asymmetry coefficient and length of kernel groove. Thus, this data set has $n = 210$ observations and $p = 7$ features. For additional details see the web page http://archive.ics.uci.edu/ml/datasets/seeds.

The solution obtained by MOMIP, to the selection variable problem, is the set of the following $k = 3$ features: width of kernel, asymmetry coefficient and length of kernel groove. The comparison between the quality measures of the best subsets with $k = 3$ variables, given by each of the 4 criteria is presented in Table 6, showing that the quality of the solution obtained by MOMIP is similar to the remaining criteria. Criteria RM and RV selected the same three variables: area, compactness and asymmetry coefficient, while criterion GCD chose compactness, length of kernel groove and asymmetry coefficient.

The solution to the *1-cluster* problem is composed by 67 observations: 62 out of 70 seeds of the Canadian variety and 5 seeds out of 70 of the Kama variety. We note that no seed of the Rosa variety was selected.

Table 6
Comparison between MOMIP and the other 3 criteria for the Seeds data set.

	MOMIP	RM	RV	GCD
MOMIP value	**5.228**	4.237	4.237	4.632
RM value	0.981	**0.990**	0.990	0.988
RV value	0.990	0.995	**0.995**	0.989
GCD value	0.940	0.989	0.989	**0.989**

Table 7
Comparison between MOMIP and the other 3 criteria for the WDBC data set.

	MOMIP	RM	RV	GCD
MOMIP value	**40.763**	33.456	40.319	35.379
RM value	0.957	**0.968**	0.960	0.961
RV value	0.989	0.987	**0.991**	0.983
GCD value	0.781	0.866	0.785	**0.897**

Wisconsin diagnostic breast cancer data set

The Wisconsin Diagnostic Breast Cancer (WDBC) data set was taken from Dua and Graff (2019). It consists on numerical characteristics of the cell nuclei, computed from an image analysis of a breast mass. Ten real-valued features were computed for each cell nucleus: radius (mean of distances from center to points on the perimeter), texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter^2/area - 1), concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry and fractal dimension (“coastline approximation” - 1). The 30 variables of this dataset are the mean, standard error and mean of the three largest values of each feature, across all cell nucleus in an image. There are 569 instances, 357 corresponding to benign cases and 212 to malignant. For details see https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).

MOMIP selected a set 11 variables: radius (mean and largest values), mean texture, mean concavity, symmetry (mean and standard error), standard errors of area, concave points, and fractal dimensions, largest values of smoothness and compactness. Standard error of symmetry and of fractal dimension were also selected by the other three approaches, which also selected standard error of radius. Beside those choices, RM selected mean perimeter and smoothness, symmetry (mean and standard error), radius (standard error), texture (standard error and largest values) and compactness (largest values). RV selected mean area, mean compactness, standard error of concave points and largest values of texture, perimeter, smoothness, concavity and symmetry. GCD criterion selected mean radius, texture (mean and standard error), mean compactness, mean symmetry, standard error of texture, smoothness and concave points and largest values of concavity.

Table 7 reports the values of all criteria of the optimal selection obtained for each criterion. We can observe, once again, that the scores of the optimal solution of MOMIP are similar to the scores of the other 3 criteria.

The 1-cluster method determined an optimal partition $(S, \bar{S})$ of the 569 instances with $|S| = 164$. Table 8 reproduces the confusion matrix that relates the elements of S and $\bar{S}$ to the instances identified as benign and malignant. TN, FN, FP and TP mean true negative, false negative, false positive and true positive, respectively.

To assess the ability of MOMIP to distinguish between benign and malignant samples, we calculated the balanced accuracy; see, e.g., Guyon et al. (2015). The balanced accuracy is the average between sensitivity and specificity, where sensitivity is the ratio TP/(TP+TN) and specificity is TN/(FP+TN). The balanced accuracy of MOMIP gave 0.849, which can be considered as an acceptable accuracy.

Table 8
Comparison between the optimal partition $(S, \bar{S})$ of MOMIP and the diagnostic (benign and malignant) for the WDBC data set.

	$\bar{S}$	S	Total
Benign	347 (TN)	10 (FP)	357
Malignant	58 (FN)	154 (TP)	212
Total	405	164	$n = 569$

We close this section with some general comments. The results obtained for these four datasets seem promising. MOMIP was well succeeded in both the reduction of dimensionality, and the selection of a subset of observations that share similar feature relationships. Notice that MOMIP does not take advantage of the knowledge of the variable response, as it is usually done by the supervised algorithms for feature selection and classification.

Concerning the computational effort measured in CPU time, using the QUBO formulation, Gurobi obtained the solution of each of the three first variable selection problems in less than 0.22 s. Selecting variables for the WDBC data set required 158 s. For the *1-cluster* approach the execution CPU times in seconds were 2.06 for iris, 0.58 for crayfish, and 117.76 for seeds. For WDBC, gurobi stopped after the imposed time limit of 10 min producing a solution with an optimality gap of 6%. We note that, for this latter problem, we let gurobi work up to 24 h obtaining the same solution only reducing the optimality gap to 5.03%.

6. Conclusions and perspectives

We have developed a graph-based combinatorial optimization approach which is effective for variable selection and a particular type of clustering. A key feature of solving this novel max-out min-in problem (MOMIP) formulation is that the cardinality of the involved sets is not required in advance. We have established that MOMIP is NP-hard. In addition, we have illustrated its performance, for variable selection and clustering, on four different data sets with a variety of distinct properties.

It is worth mentioning that a generalization of MOMIP can be obtained by replacing the matrix $\widehat{W} \equiv [\hat{w}_{uv}]$, in the QUBO formulation (8)–(10), by the following one

$$\hat{w}_{uv}(\lambda) = \begin{cases} \sum\limits_{s \in V} w_{us} & \text{if } u = v \\ -\frac{3}{2}\,\lambda\, w_{uv} & \text{if } \; u \neq v, \end{cases} \tag{11}$$

where $\lambda > 0$ is a real parameter. We note that if $\lambda = 1$ then we recover the original MOMIP formulation (8)–(10), and if $\lambda = 2/3$ then we obtain the well-known weighted maximum cut problem; see, e.g., Shylo and Shylo (2010). Moreover, if W is the adjacency matrix of graph G and $\Delta(G)$ denotes the maximum degree among the vertices of G, then it can be seen that for any arbitrary $\lambda > \Delta(G)/3$ an optimal solution S cannot have an edge with both vertices in S, i.e., S is an independent set of G. Furthermore, if S is an optimal set, every vertex not in S is adjacent to some vertex in S, i.e., S is a maximal independent set or, equivalently, it is a dominating set. Hence, for whatever different values of $\lambda > \Delta(G)/3$, the optimal solutions will be the same and all of them (if more than one exists) will be an independent dominating set of G. Independent dominating sets have been largely studied in graph theory, see, e.g., Goddard and Henning (2013). Particular attention has been directed to the independent domination number of a graph, denoted by $i(G)$, which is the minimum size of an independent dominating set. Therefore, a related problem is the variant of MOMIP that can be obtained by setting $\lambda > \Delta(G)/3$ in (11): find an independent dominating set S of graph G having the largest number of edges with one end in S.

We also note that the *1-cluster* approach that we described and illustrated in Section 5, as an application of MOMIP, is not an archetypal clustering procedure. The *1-cluster* solution S maximizes dissimilarities from elements in S to the elements outside S and minimizes the dissimilarities between pairs of elements in S. No concern is explicitly given to dissimilarities among the elements which are not in S. The dissimilarities among the elements outside S is a consequence of the goals defined for finding S. So, it is expected that, if the sum of the dissimilarities between elements of a subset $S' \subseteq \bar{S} = V \setminus S$ to elements in $\bar{S} \setminus S'$ is large and the sum of dissimilarities between pairs of elements in S' is small, then the elements of S' should be very dissimilar from the elements of S. Taking this remark into account, the following iterative hierarchical clustering procedure could be considered. At each iteration MOMIP defines, on the current weighted graph G, an optimal *1-cluster* solution S, with the additional concern to add to S every isolated vertex; and then the graph G is updated removing S from the current G. The procedure stops when G has no edges. This hierarchical clustering algorithm has the advantage of having no need to know in advance the number of clusters. Continuous optimization models in which the number of clusters is not required, have also been recently proposed; see, e.g., Shah and Koltun (2017). We plan to explore our iterative hierarchical clustering procedure in a future work.

CRediT authorship contribution statement

Jorge Orestes Cerdeira: Study, Conception, Design, Data collection, Conceptualization, Methodology, Formal analysis, Writing of the manuscript. **Maria João Martins:** Study, Conception, Design, Data collection, Conceptualization, Methodology, Formal analysis, Writing of the manuscript. **Marcos Raydan:** Study, Conception, Design, Data collection, Conceptualization, Methodology, Formal analysis, Writing of the manuscript.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

We would like to thank the anonymous referees for pointing out some key references and for their constructive comments and suggestions that helped us to improve the final version of this paper. In particular, the suggestion of comparing the performance of the two different proposed formulations, and also the suggestion of considering a controlled test on simulated data that has a well-separated group of vertices from the rest of the graph. All authors read and approved the final manuscript.

Funding

The first and third authors were financially supported by the Fundação para a Ciência e a Tecnologia, Portugal (Portuguese Foundation for Science and Technology) through the projects UIDB/MAT/00297/2020, UIDP/MAT/00297/2020 (Centro de Matemática e Aplicações), and PTDC/CCI-BIO/4180/2020. The second author was financially supported by the Forest Research Center, a research unit funded by Fundação para a Ciência e a Tecnologia (FCT), Portugal, through the project (UIDB/00239/2020).

References

Cadima, J., Cerdeira, J.O., Minhoto, M., 2004. Computational aspects of algorithms for variable selection in the context of principal components. Comp. Stat. Data Anal. 47, 225–236.

Cadima, J., Jolliffe, I.T., 2001. Variable selection and the interpretation of principal subspaces. J. Agric. Biol. Environ. Stat. 6, 62–79.

Cerdeira, J.O., Silva, P.D., Cadima, J., Minhoto, M., 2020. subselect: Selecting variable subsets. R package version 0.15.2. https://cran.r-project.org/web/packages/subselect/.

Ding, C.H.Q., He, Xiaofeng, Zha, Hongyuan, Gu, Ming, Simon, H.D., 2001. A min–max cut algorithm for graph partitioning and data clustering. In: Proceedings 2001 IEEE International Conference on Data Mining. pp. 107–114.

Dua, D., Graff, C., 2019. UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA, http://archive.ics.uci.edu/ml.

Everitt, B.S., Landau, S., Leese, M., Stahl, D., 2011. Cluster Analysis. John Wiley & Sons Ltd, United Kingdom.

Fisher, R.A., 1936. The use multiple measurements in taxonomic problems. Ann. Eugenics 7, 179–188.

Gan, G., Ma, C., Wu, J., 2007. Data Clustering: Theory, Algorithms, and Applications, ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, ASA, Alexandria, VA.

Garey, M., Johnson, D.S., 1979. Computers and Intractability: A Guide To the Theory of NP- Completeness. W. H. Freeman & Co, San Francisco.

Garey, M., Johnson, D.S., Stockmeyer, L., 1976. Some simplified NP-complete graph problems. Theoret. Comput. Sci. 1, 237–267.

Goddard, W., Henning, M.A., 2013. Independent domination in graphs: A survey and recent results. Discrete Math. 313, 839–854.

Gurobi Optimization, LLC, 2022. Gurobi Optimizer Reference Manual. https://www.gurobi.com.

Guyon, I., Bennett, K., Cawley, G., Escalante, H.J., Escalera, S., Ho, T.K., Macià, N., Ray, B., Saeed, M., Statnikov, A., Viegas, E., 2015. Design of the 2015 chalearn automl challenge. In: International Joint Conference on Neural Networks. IJCNN, pp. 1–8.

Jiang, D., Tang, C., Zhang, A., 2004. Cluster analysis for gene expression data: A survey. In: IEEE Transactions on Knowledge and Data Engineering, vol. 16, pp. 1370–1386.

Jolliffe, I.T., 2002. Principal Component Analysis, second ed. Springer, New York.

Jolliffe, I.T., Cadima, J., 2016. Principal component analysis: A review and recent developments. Phil. Trans. R. Soc. A 374, 20150202.

Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y., 2014. The unconstrained binary quadratic programming problem: A survey. J. Comb. Optim. 28, 58–81.

Pandove, D., Goel, S., Rani, R., 2018. Systematic review of clustering high-dimensional and large datasets. ACM Trans. Knowl. Discov. Data. 12, 16.

R Core Team, 2020. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Şeker, Oylum, Tanoumand, Neda, Bodur, Merve, 2022. Digital Annealer for quadratic unconstrained binary optimization: A comparative performance analysis. Appl. Soft Comput. 127, 109367.

Shah, S.A., Koltun, V., 2017. Robust continuous clustering. Proc. Natl. Acad. Sci. U. S. A. 114 (37), 9814–9819.

Shylo, V.P., Shylo, O.V., 2010. Solving the max-cut problem by the global equilibrium search. Cybernet. Systems Anal. 46 (5), 744–754.

Somers, K.M., 1986. Allometry, isometry and shape in principal component analysis. Syst. Zool. 38, 169–173.