Journal of Global Optimization (2023) 87:133–189
https://doi.org/10.1007/s10898-022-01267-4

Mixed-integer programming techniques for the minimum sum-of-squares clustering problem

Jan Pablo Burgard[1] · Carina Moreira Costa[2] · Christopher Hojny[3] · Thomas Kleinert[4] · Martin Schmidt[2]

Received: 9 March 2022 / Accepted: 13 December 2022 / Published online: 10 January 2023

Abstract
The minimum sum-of-squares clustering problem is a very important problem in data mining and machine learning with very many applications in, e.g., medicine or social sciences. However, it is known to be NP-hard in all relevant cases and to be notoriously hard to be solved to global optimality in practice. In this paper, we develop and test different tailored mixed-integer programming techniques to improve the performance of state-of-the-art MINLP solvers when applied to the problem—among them are cutting planes, propagation techniques, branching rules, or primal heuristics. Our extensive numerical study shows that our techniques significantly improve the performance of the open-source MINLP solver SCIP. Consequently, using our novel techniques, we can solve many instances that are not solvable with SCIP without our techniques and we obtain much smaller gaps for those instances that can still not be solved to global optimality.

Keywords Minimum sum-of-squares clustering · Mixed-integer nonlinear optimization · Global optimization · Computational techniques

✉ Martin Schmidt
martin.schmidt@uni-trier.de

Jan Pablo Burgard
burgardj@uni-trier.de

Carina Moreira Costa
carinamath5@gmail.com

Christopher Hojny
c.hojny@tue.nl

Thomas Kleinert
thomas.kleinert@quantagonia.com

1 Department of Economic and Social Statistics, Trier University, Universitätsring 15, 54296 Trier, Germany

2 Department of Mathematics, Trier University, Universitätsring 15, 54296 Trier, Germany

3 Combinatorial Optimization Group, Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

4 Quantagonia GmbH, Foellerweg 37, 61352 Bad Homburg, Germany

Mathematics Subject Classification 90C10 · 90C11 · 90C57 · 90-08

1 Introduction

Given a set of data points in a normed vector space and a number of clusters, the clustering problem consists in deciding which data point should be assigned to which cluster. Moreover, a representative point for each cluster needs to be determined. Clustering problems form a highly relevant sub-class of unsupervised learning in machine learning and computational statistics. Its relevance is supported by many applications, e.g., in functional data analysis [10, 53], image processing [9], bio-informatics [12], economics [32], and social sciences [31]. For a detailed survey of the history of clustering problems we refer to Steinley [58]. Depending on, e.g., the way how vicinity is measured and on whether the representative is an arbitrary point or one of the data points, different variants of clustering problems arise. In this paper we consider the minimum sum-of-squares clustering (MSSC) problem. Here, distance between data points is measured using the squared Euclidean norm and any point can be chosen as the representative for each cluster.

Modeling this problem leads to a nonconvex mixed-integer nonlinear optimization problem (MINLP) that is extremely hard to solve for high-dimensional real-world problems. Moreover, the problem is known to be NP-hard even in the case of two dimensions; see, e.g., Aloise et al. [11], Dasgupta [2], and Mahajan et al. [41]. This is why the problem is most frequently solved using heuristics out of which the k-means clustering method is the most prominent one; see, e.g., Lloyd [39], MacQueen [40]. However, solving such clustering problems only heuristically may come with severe disadvantages. Since solving the MSSC problem is an unsupervised learning problem, the outcome typically requires the interpretation of experts from the specific field of application such as medicine or social sciences. This interpretation, however, may be completely wrong in the case that the expert is confronted with a heuristic clustering solution of bad quality. Moreover, it is easy to imagine that such a misleading interpretation might have some severe, e.g., medical, consequences. Thus, there is a strong need for sophisticated optimization techniques to improve the process of solving clustering problems to global optimality and this is exactly the contribution of this paper: We take the MINLP solver SCIP and enhance its solution process by developing novel mixed-integer optimization techniques that enable us to solve MSSC problems to global optimality that cannot be solved with the plain version of SCIP.

Of course, we are not the first ones trying to solve the MSSC problem to global optimality. To the best of our knowledge, the earliest application of branch-and-bound methods is presented by Fukunaga et al. [25], which has been refined later on by Diehr [15]. A variant of a so-called repetitive branch-and-bound method has been devised by Brusco [7], where the authors conclude that their method is well-suited for a small number of clusters. Another recent branch-and-bound approach is presented by Sherali and Desai [55]. The authors use reformulation-linearization-techniques (RLT) embedded in a branch-and-bound method to solve the problem to global optimality. In their introduction, they also talk about a "limited number of optimization techniques" as opposed to a rather large number of heuristics that are used in practice. As an additional technique, the authors present further valid inequalities to tackle the inherent symmetry of the problem. Regarding symmetry breaking for clustering problems we also refer to Plastria [48], which is, generally speaking, a modeling tutorial paper but which also contains a discussion of symmetry breaking constraints for the cluster-

ing problem. Aloise and Hansen [4] also consider the MSSC problem and try to re-produce the results of Sherali and Desai [55]. However, the re-production failed since significantly longer running times have been observed. Consequently, the computational efficiency of the RLT-based branch-and-bound method may be taken with some care. Another algorithmic technique is the column generation approach presented first in Merle et al. [18] and which has been re-considered and improved by Aloise et al. [5]. Also other classic techniques of mixed-integer (non)linear optimization have been applied such as generalized Benders decomposition in Floudas et al. [21], and Tan et al. [59]. Alternatively, Peng and Xia [45] consider the MSSC problem as a concave minimization problem and adapt Tuy's cut method, see Horst and Tuy [34], for solving the problem. Further, Prasad and Hanasusanto [49] propose improved conic reformulations of the MSSC problem and also study some symmetry breaking techniques. Tîrnăucă et al. [60] follow a more geometric approach that is based on Voronoi diagrams. Finally, there is a rather large branch of literature towards the application of techniques from semi-definite programming (SDP). Peng and Wei [46], Peng and Xia [44] proved the equivalence between the MSSC problem and a 0-1 SDP reformulation. Based on this 0-1 SDP model, Aloise and Hansen [3] propose a branch-and-cut algorithm and solve instances with up to 202 data points to global optimality. More recently, Piccialli et al. [47] consider the same mixed-integer SDP for the MSSC problem and propose another branch-and-bound algorithm that is capable of solving real-world instances with up to 4000 data points. To the best of our knowledge, this is the most recent state-of-the-art branch-and-bound algorithm for the MSSC problem. SDP-like models have also been used in De Rosa and Khajavirad [13] to use $Z = XX^\top \in [0,1]^{n\times n}$ for encoding the clustering instead of $X \in \{0,1\}^{n\times k}$. The authors derive cutting planes and show a relation of their cutting planes to the cut polytope; see Deza and Laurent [14] for a survey on the latter. The presented numerical experiments show that these novel cutting planes can be strong but the authors only solve the initial LP relaxation and do not apply a complete branch-and-bound method. Finally, some recent ideas based on reduced-space techniques seem to be very promising, see Hua et al. [35] and Liberti and Manca [38]. Besides that, in Liberti and Manca [38], the authors discuss the MSSC problem with several side constraints. One of their base models is, in particular, the convex MINLP that we present in the next section.

In our contribution, we add to the literature on solving the MSSC problem to global optimality. To this end, we develop novel mixed-integer programming techniques that are mainly motivated by geometric insights and that improve the branch-and-cut solution process of an MINLP solver. To be more precise, we present two MINLP formulations of the problem (Sect. 2), develop cutting planes (Sect. 3), propagation methods (Sect. 4), as well as problem-specific branching rules (Sect. 5) and primal heuristics (Sect. 6). We implement and test all techniques in the open-source MINLP solver SCIP; see Gamrath et al. [26]. By doing so, we also automatically apply state-of-the-art symmetry breaking techniques to the problem. Our numerical results are presented and discussed in Sect. 8, where we show that our techniques significantly improve the solution process. We close the paper with some concluding remarks and some potential topics for future work in Sect. 9. Our code is publicly available at GitHub.[1] Although our numerical results clearly show that the solution process of an MINLP solver applied to the MSSC problem is significantly improved, we do not beat current state-of-the-art and SDP-based techniques as studied in Piccialli et al. [47]. Nevertheless, we are convinced that it is worth to also push MINLP-based approaches forward so that, in the end, different techniques for various approaches can be combined to lead to an even better and maybe hybrid solution approach.

[1] https://github.com/christopherhojny/globally-solving-MSSC.

2 MINLP models for the MSSC problem

We now model the minimum-sum-of-squares clustering (MSSC) problem as a mixed-integer nonlinear optimization problem (MINLP). To this end, we are given a set of data points $p \in P \subseteq \mathbb{R}^d$ and a positive integer $2 \leq k \leq |P|$, which is the number of clusters of the problem. The task is then to assign every data point $p \in P$ to a cluster (indexed by $j \in [k]{:=}\{1, \ldots, k\}$) so that the sum of the squared Euclidean distances between the data points and the corresponding centroids c^j is minimal. This problem is modeled via the following MINLP:

$$\min_{x,c} \quad \sum_{p\in P} \sum_{j\in[k]} x_{pj} \|p - c^j\|^2 \tag{1a}$$

$$\text{s.t.} \quad \sum_{j\in[k]} x_{pj} = 1, \quad p \in P, \tag{1b}$$

$$x_{pj} \in \{0, 1\}, \quad p \in P,\ j \in [k], \tag{1c}$$

$$c^j \in B, \quad j \in [k]. \tag{1d}$$

The binary variables x_{pj} are the assignment variables that model whether the data point p is assigned to cluster j ($x_{pj} = 1$) or not ($x_{pj} = 0$). Moreover, $B \subseteq \mathbb{R}^d$ is a set that contains all points in P. This can, e.g., be the bounding box of P. That is, if for each $i \in [d]$, $\ell_i = \min\{p_i : p \in P\}$ and $u_i = \max\{p_i : p \in P\}$, then $B = \{c \in \mathbb{R}^d : \ell_i \leq c_i \leq u_i,\ i \in [d]\}$ is a valid choice. Note that (1d) is not necessary for the correctness of Model (1). Nevertheless, we include it in our implementation, because Model (1) is a nonconvex MINLP, for which bounds on variables are usually beneficial. The objective function measures the sum of the squared Euclidean distances between the data points and the centroids of the clusters to which they belong. Finally, Constraint (1b) ensures that every point is assigned to exactly one cluster.

Note that this model is cubic since the objective function uses multiplications of the assignment variables x with the norms that depend on the centroids c, which are variables of the problem as well. In particular, Model (1) is a nonconvex MINLP. However, it can also be re-written as a convex MINLP in a lifted space by using its epigraph formulation. To this end, we model each term in the objective function using a separate variable and bound it in a newly introduced constraint. The resulting problem then reads

$$\min_{x,c,\eta} \quad \sum_{p\in P} \sum_{j\in[k]} \eta_{pj} \tag{2a}$$

$$\text{s.t.} \quad \eta_{pj} \geq \|p - c^j\|^2 - M_p(1 - x_{pj}), \quad p \in P,\ j \in [k], \tag{2b}$$

$$\sum_{j\in[k]} x_{pj} = 1, \quad p \in P, \tag{2c}$$

$$x_{pj} \in \{0, 1\}, \quad p \in P,\ j \in [k], \tag{2d}$$

$$c^j \in B, \quad j \in [k], \tag{2e}$$

$$\eta_{pj} \geq 0, \quad p \in P,\ j \in [k], \tag{2f}$$

where M_p are sufficiently large numbers. The objective function is linear now and we obtain the additional quadratic and convex constraints in (2b).

For every $p \in P$, M_p can be chosen to be the maximum distance of p to any other point $\tilde{p} \in P$. An overestimation can be easily computed via

$$M_p = M = (u_1 - \ell_1)^2 + \cdots + (u_d - \ell_d)^2,$$

where ℓ_i and u_i are the componentwise bounds of the bounding box given above. Moreover, for a given cluster assignment x, an optimal choice for the cluster centroids is immediate, an observation that we will exploit frequently.

Observation 2.1 *For a given assignment of x-variables adhering to (1b) or (2c), respectively, the optimal choice for c^j, $j \in [k]$, is*

$$\frac{\sum_{p \in P} p\, x_{pj}}{\sum_{p \in P} x_{pj}},$$

i.e., the barycenter of all points assigned to cluster j.

3 Cutting planes

Without doubt, cutting planes are among the most powerful techniques to enhance the solution process for mixed-integer problems. Modern MI(N)LP solvers have many general-purpose cutting planes built-in. However, it is very often beneficial to derive problem-specific cutting planes. This is particularly important for the MSSC problem, since it is well-known that one of the most challenging issues for developing an efficient branch-and-bound algorithm for the MSSC problem is the computation of good lower bounds in a reasonable amount of time. In this section, we state two tailored cutting planes. The first one is applicable to Model (1) and (2) whereas the second one is only applicable to Model (2).

3.1 Cardinality cuts

We first briefly discuss cardinality cuts, which are already mentioned in Aloise et al. [5] as well as Sherali and Desai [55]. Consider an optimal solution of Model (1). In this optimal solution, there cannot be any empty cluster, because otherwise the corresponding objective value can be decreased by assigning a point that is not a centroid to that empty cluster. On the other extreme, a cluster contains $|P| - k + 1$ data points if every other cluster consists of only a single data point. Thus, to tighten the formulation (1), the following cardinality cuts can be added to Model (1):

$$1 \leq \sum_{p \in P} x_{pj} \leq |P| - k + 1, \quad j \in [k].$$

Obviously, the same cuts are also valid for Model (2). Moreover, note that the upper bound is implied by the model's constraints and the lower bound of the previous inequalities as $\sum_{p \in P} \sum_{j=1}^{k} x_{pj} = |P|$ implies for a fixed $j \in [k]$ that $\sum_{p \in P} x_{pj} = |P| - \sum_{p \in P} \sum_{j' \in [k] \setminus \{j\}} x_{pj'} \leq |P| - (k-1)$.

The idea of cardinality cuts can also be localized, i.e., the cardinality bounds can be adapted to take local variable bounds at a node of the branch-and-bound tree into account. To this end, we introduce, for each $j \in [k]$ the integral variable κ_j with range $\{k-1, \ldots, |P|-1\}$ and link it with the x-variables via the linear constraint $\kappa_j + \sum_{p \in P} x_{pj} = |P|$, $j \in [k]$.

That is, κ_j describes the number of data points that are not assigned to cluster j. If a lower bound $\underline{\kappa}_j$ and an upper bound $\bar{\kappa}_j$ on κ_j is given, this equation implies the inequalities

$$1 \leq |P| - \bar{\kappa}_j \leq \sum_{p \in P} x_{pj} \leq |P| - \underline{\kappa}_j \leq |P| - k + 1, \quad j \in [k].$$

That is, they describe localized versions of cardinality cuts that get stronger if x-variables get fixed. Another side effect of the auxiliary variables κ_j is that a solver might decide to branch on these variables. In doing so, it imposes bounds on the size of cluster $j \in [k]$.

3.2 Outer approximation cuts

We now focus on Model (2). The only nonlinear constraints (2b) in this problem are convex. Thus, their first-order Taylor approximations are global underestimators at any point $(\bar{\eta}, \bar{c}, \bar{x})$ and thus provide valid inequalities that are linear in (η, c, x):

$$\sum_{i=1}^{d} \left(2\bar{c}_i^j c_i^j - 2p_i c_i^j + (p_i)^2 - (\bar{c}_i^j)^2 \right) - \eta_{pj} - M_p(1 - x_{pj}) \leq 0, \quad p \in P, \ j \in [k]. \tag{3}$$

This allows to solve Model (2) in an outer approximation or LP/NLP-based branch-and-bound fashion; see Duran and Grossmann [17], Fletcher and Leyffer [20] or Quesada and Grossmann [50], respectively. We start by relaxing the constraint set (2b). Next, we assume that $(\bar{\eta}, \bar{c}, \bar{x})$ is a solution of this relaxation, i.e., it particularly fulfills the binary conditions (2d). If the relaxation's solution is feasible for the nonlinear constraints (2b), it is also a solution for Model (2). If not, we can compute a feasible point $(\hat{\eta}, \hat{c}, \bar{x})$ of Model (2). In the original outer approximation method, this is done by fixing the binary variables $\bar{x}$ in Model (2) and solving the resulting convex NLP subproblem; see Duran and Grossmann [17]. The benefit in our specific application is that solving the subproblem boils down to a simple computation of the barycenters $\hat{c}$, see Observation 2.1, followed by an evaluation of the distances $\hat{\eta}$ according to Constraints (2b) and (2f).

From the theory of outer approximation, it is well-known that when adding the inequalities (3) at the solution $(\hat{\eta}, \hat{c}, \bar{x})$ of the subproblem, it holds

$$\sum_{p \in P} \sum_{j \in [k]} \eta_{pj} \geq \sum_{p \in P} \sum_{j \in [k]} \hat{\eta}_{pj}$$

for all feasible points $(\eta, c, \bar{x})$ of the updated relaxation. In other words, adding the outer-approximation cuts bounds the optimal objective value of the relaxation with fixed binaries $x = \bar{x}$ from below by $\sum_{p \in P} \sum_{j \in [k]} \hat{\eta}_{pj}$. Consequently, the updated relaxation yields a solution with a new, previously unseen, cluster assignment x or the optimality gap is closed. Thus, iterating this process terminates after a finite number of steps; see Duran and Grossmann [17] or Duran and Grossmann [20] for more details. We note that the number of inequalities (3) does not depend on the dimension d, which might be beneficial for problems with higher dimensions.

Instead of implementing an LP/NLP-based branch-and-bound from scratch, we can use solvers such as SCIP to solve Model (2). In this setting, we can separate and add cuts (3) to tighten the LP relaxations. Since for larger $|P|$ and k, adding all inequalities (3) might be impracticable, we may also add only a certain amount of cuts. In particular, in our implementation in SCIP, we add only 10 cuts per separation round.

4 Propagation

Suppose we are at a node of the branch-and-bound tree. Due to branching decisions and further reductions, some variables might have been fixed or their bounds have been tightened in comparison to the original problem formulation. The aim of propagation is to find further variable fixings or bound tightenings that are valid at the current node. That is, one tries to apply further reductions based on local variable bound information. According to Observation 2.1, every assignment of x-variables that satisfies (1b) or (2c) can be extended to a feasible solution of (1) or (2), respectively. Thus, it is crucial to derive propagation mechanisms that exclude assignments of x-variables that cannot be optimal. Moreover, we develop algorithms to strengthen bounds of the c-variables and the objective variables. Before we discuss our propagation algorithms, we fix the following notation and terminology.

For every $j \in [k]$, we denote by $P_j \subseteq P$ the set of all data points $p \in P$ whose corresponding variable x_{pj} has been fixed to 1 at the current node of the branch-and-bound tree. That is, we have already decided to assign p to cluster j. Moreover, we denote by $P'_j \subseteq P$ all data points p such that x_{pj} has not been fixed to 0 yet, i.e., p is already or can still be assigned to cluster j. Note that $P_j \subseteq P'_j$. For a continuous variable z, i.e., for the c- and η-variables, we denote by $\underline{z}$ and $\bar{z}$ the lower and upper bound on z at the current node, respectively.

4.1 Barycenter propagation

Given a non-empty set of data points $Q \subseteq P$ defining a cluster, the optimal choice for its centroid is the barycenter

$$\mathcal{C}(Q) := \frac{1}{|Q|} \sum_{p \in Q} p$$

of all data points in Q. The respective sum of all squared distances thus is $\mathcal{D}(Q) = \sum_{p \in Q} \|p - \mathcal{C}(Q)\|^2$. The idea of the barycenter propagation is to use this observation to find lower bounds on the objective and to strengthen the bounds for the c-variables.

4.1.1 Bound tightening for the objective function values

To find a lower bound on the objective in Model (1), note that for sets $Q \subseteq Q' \subseteq P$, we have $\mathcal{D}(Q) \leq \mathcal{D}(Q')$. Consequently, a lower bound on the objective is given by $\sum_{j \in [k]} \mathcal{D}(P_j)$. The barycenter propagator uses this value to possibly tighten the lower bound on the objective at the current node of the branch-and-bound tree. Computing this lower bound for all clusters can be done in $O(kd|P|)$ time and it has also been used by Brusco [7], see also Guns et al. [30], in a repetitive branch-and-bound framework.

For Model (2), no immediate lower bound on the objective can be enforced because the objective is decoupled via the η-variables. Nevertheless, for each $(p, j) \in P \times [k]$, the following steps can be done. We can prune a node of the branch-and-bound tree if $p \notin P'_j$ and $\underline{\eta}_{pj} > 0$, because an optimal solution has $\eta_{pj} = 0$ as data points not assigned to a cluster do not contribute to $\mathcal{D}(P_j)$. Otherwise, if $p \notin P'_j$ and $\underline{\eta}_{pj} = 0$, we can fix η_{pj} to 0. The first step is thus a pruning operation based on sub-optimal bounds in the subproblem, whereas the second step is a bound tightening operation.

4.1.2 Bound tightening for the centroids

Recall that $[k] = \{1, \ldots, k\}$. Besides strengthening bounds on the objective, barycenter information can also be used to tighten bounds on centroid variables c_i^j with $(i, j) \in [d] \times [k]$. Suppose $P_j' \setminus P_j = \{p^1, \ldots, p^s\}$ such that $p_i^1 \leq p_i^2 \leq \cdots \leq p_i^s$. For each $r \in [s]_0 := [s] \cup \{0\}$, we compute $\gamma^{j,r} = \mathcal{C}(P_j \cup \{p^1, \ldots, p^r\})$, i.e., the barycenter of the data points contained in P_j and the data points with the r smallest ith coordinates that are not contained in P_j. As we show next, the ith coordinates of these barycenters can be used to compute a lower bound on c_i^j.

Lemma 4.1 *A valid lower bound on c_i^j is given by $\min_{r \in [s]_0} \gamma_i^{j,r}$.*

Proof Let $Q \subseteq P_j' \setminus P_j$ and assume $|Q| = r$. Then, $\mathcal{C}(P_j \cup Q)_i \geq \mathcal{C}(P_j \cup \{p^1, \ldots, p^r\})_i$, because the points $p^1, \ldots, p^r$ are points with the r smallest ith coordinates. Consequently, to find a lower bound on the centroids, it is sufficient to consider $\mathcal{C}(P_j \cup \{p^1, \ldots, p^r\})$ for each $r \in [s]_0$. □

Analogously, an upper bound is given by $\max_{r \in [s]_0} \mathcal{C}(P_j \cup \{p^{s-r}, \ldots, p^s\})_i$. Since computing an iterative sequence of barycenters can be done using the formula

$$\gamma^{j,r+1} = \frac{(|P_j| + r)\gamma^{j,r} + p^{r+1}}{|P_j| + r + 1},$$

we can compute the minimum and maximum values for all coordinates and clusters in $O(kd|P|)$ time.

4.2 Convexity and cone propagation

Based on optimality arguments, we can also derive rules to assign data points $p \in P_j' \setminus P_j$ to cluster $j \in [k]$. The key idea of the convexity propagator is the following simple observation.

Lemma 4.2 *There exists an optimal solution of MSSC with clusters $P_1, \ldots, P_k$ such that, for each $j \in [k]$, we have $\operatorname{conv}(P_j) \cap P = P_j$.*

Proof Given an optimal allocation of the k centroids, the Voronoi cells

$$C_j = \{x \in \mathbb{R}^d : \|x - c^j\| \leq \|x - c^{j'}\|, \ j' \in [k]\}$$

for $j \in [k]$ cover the entire $\mathbb{R}^d$ and only intersect at their boundaries. Since Voronoi cells are full-dimensional polyhedra, we can use the following mechanism to prove the assertion. We start with cluster 1 and observe that $P_1 \subseteq C_1$ in any optimal solution. If there exist $p \in P \setminus P_1$ that are contained in C_1, they are necessarily contained in the boundary of C_1. Hence, if we change the assignment of these points to P_1, this does not change the objective of MSSC. The assertion thus holds for P_1, and we can use the same arguments iteratively to conclude the proof. □

As a consequence, the convexity propagator computes $\operatorname{conv}(P_j)$ for each $j \in [k]$. If there exists $p \in P \cap \operatorname{conv}(P_j)$ it performs the following steps: If $p \notin P_j'$ holds, then we can prune the current node of the branch-and-bound tree, because the local variable bounds cannot lead to an optimal solution adhering to Lemma 4.2. Otherwise, x_{pj} can be fixed to 1.

Besides pruning nodes and fixing variables to 1, Lemma 4.2 has another consequence that allows us to fix some variables to 0, which is illustrated in Fig. 1.

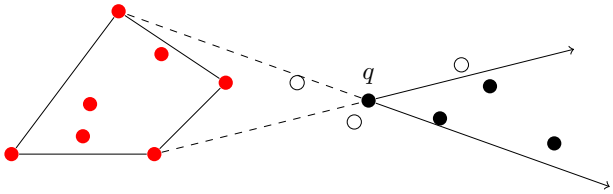

Fig. 1 Illustration of cone-based propagation. If q is not contained in the red cluster, none of the black points can be contained in the red cluster. Assigning the white points to the red cluster is still possible

Lemma 4.3 *Let $P_1 \cup \cdots \cup P_k$ be a partition of a finite set $P \subseteq \mathbb{R}^d$. Suppose* $\text{conv}(P_j) \cap P = P_j$ *for each $j \in [k]$. Then, for every $q \in P \setminus P_j$,*

$$q + \text{cone}\{-(p-q) \colon p \in P_j\} \supseteq P \setminus P_j.$$

Proof Note that $q + \text{cone}\{p - q \colon p \in P_j\}$ is the smallest cone with apex q that contains $\text{conv}(P_j)$, because $q \notin P_j$ and we shoot rays from q through each of the finitely many points in P_j. Hence, negating these rays leads to a cone that cannot contain any point from P_j as it does not contain $\text{conv}(P_j)$. □

We can use this observation as follows. If there is $q \in P \setminus P'_j$, i.e., x_{qj} is fixed to 0, then all points in $q + \text{cone}\{-(p-q) \colon p \in P_j\}$ can be fixed to 0 as well.

In arbitrary dimensions, the convexity propagator cannot be implemented efficiently, because $\text{conv}(P_j)$ might have $\Omega(2^d)$ many facets. In small dimensions, computing convex hulls can be done rather quickly and, as our numerical results will indicate, have a very positive impact on the time needed to solve MSSC.

4.3 Distance propagation

The distance propagator provides another set of rules to fix variables x_{pj}, $(p, j) \in P \times [k]$, to 0. To this end, it defines for each $j \in [k]$ the bounding box $B_j = \{y \in \mathbb{R}^d \colon \underline{c}_i^j \leq y_i \leq \bar{c}_i^j,\ i \in [d]\}$ for the centroids. That is, the smallest box that contains the centroid for cluster j based on local variable bound information. Afterward, for each $p \in P$ and $j \in [k]$, it computes the minimum and maximum distances $D_{j,p}^{\min}$ and $D_{j,p}^{\max}$ to the bounding box B_j, i.e.,

$$D_{j,p}^{\min} = \min\left\{\|p - x\| \colon x \in B_j\right\}, \quad D_{j,p}^{\max} = \max\left\{\|p - x\| \colon x \in B_j\right\}.$$

Since a data point is assigned to a centroid of minimum distance in an optimal solution, p cannot be assigned to cluster $j \in [k]$ if there is $j' \in [k]$ with $D_{j',p}^{\max} < D_{j,p}^{\min}$. Consequently, x_{pj} can be fixed to 0 in this case.

Finding B_j and computing the minimum and maximum distance of a point to a box can be done in $O(d)$ time. Hence, the distance propagator runs in $O(kd|P|)$ time.

5 Branching rules

After a node of the branch-and-bound tree has been processed (adding cutting planes, propagation), branching rules split the current subproblem into further subproblems to, e.g., tighten the problem formulation or enforce integrality of variables. The decision on how to split the

current subproblem is typically guided by the solution of the subproblem's LP relaxation. To enforce integrality of variables (in the notation of the MSSC problem), one typically selects a variable x_{pj} whose value in the LP solution is non-integral. Then, two subproblems are created that fix x_{pj} to 0 and 1, respectively. Despite the existence of many branching rules that perform well for generic problems, see Achterberg et al. [1], there also exist branching rules tailored to a specific problem. This becomes relevant, because they might allow to derive further reductions based on problem structure or further components of a solver such as propagation mechanisms.

For integer programs, Gilpin and Sandholm [27] proposed four families of branching rules motivated from an information-theoretic perspective. The common ground of all their rules is to interpret the values x_{pj} as the probability that a point $p \in P$ is assigned to cluster $j \in [k]$. Using their rules, they aim at reducing the assignment uncertainty in the current subtree. The first and second rule use a look-ahead approach, similar to strong branching. For the MSSC problem with a large number of clusters or points, look-ahead branching rules may become computationally prohibitive when applied to the full problem. Hence, we do not use them. The third family is called entropic look-ahead-free variable selection and the fourth is its extension for a multi-variable branching version. Since we think that the third family might be helpful for the MSSC problem, we describe it in the following. Afterward, two novel branching rules for the MSSC problem will be presented.

5.1 Entropy branching

Suppose that the optimal LP solution of the current subproblem does not satisfy the integrality constraints, which means that the relaxed solution $\bar{x}$ is non-integral. Let $\bar{X}$ be the set of all branching candidates that are non-integral in this LP solution, i.e.,

$$\bar{X} := \left\{ \bar{x}_{pj} : \bar{x}_{pj} \in (0,1),\ p \in P,\ j \in [k] \right\}.$$

Since each of these $\bar{x}_{pj}$ is non-integral, the cluster assignment of point p is not fixed. For the assignment to be fixed, $\bar{x}_{pj}$ has to be one. Due to Constraints (1b) or (2c), $\bar{x}_{pj}$ can be seen as a kind of posterior probability of point p to belong to cluster j [61]. A good strategy for branching would be to select the point p for which the probabilities for each cluster assignment are almost the same.

The most unclear situation is where $\bar{x}_{pj} = 1/k$ holds for all $j \in [k]$. Here, each cluster assignment is equally probable for point p. This can be seen as a homogeneous information setting. The level of homogeneity can be measured via the Shannon entropy of point p [54]. More precisely, for each variable $\bar{x}_{pj} \in \bar{X}$ the entropy of point p with probabilities $\bar{x}_{pj}$, $j \in [k]$, is

$$H_p = -\sum_{j \in [k]} \bar{x}_{pj} \log_2(\bar{x}_{pj}).$$

The maximum entropy ($H_p = \log_2 k$) occurs in the above mentioned extreme case. That is, the current LP solution does not provide any information on the best (or most probable) cluster assignment of point p. The minimum entropy is obtained when there is a clear cluster assignment. In this situation, let the point p be already assigned to a cluster, e.g., to cluster $j = 1$ and hence $\bar{x}_{p1} = 1$. Due to (1b) (or (2c)), $\bar{x}_{pj'} = 0$ for $j' \in [k] \setminus \{1\}$. The entropy of p is then

$$H_p = -1 \log_2 1 - 0 \log_2 0 - \cdots - 0 \log_2 0 = 0,$$

where $0 \log_2 0$ is taken to be zero.

We are interested in finding the point corresponding to the fractional variable $\bar{x}_{p^*j^*} \in \bar{X}$ such that the entropy of point p^* is the maximal over all points with fractional variables, i.e., we search for the most uncertain assignment. The point is formally given by

$$p^* \in \underset{\{p \in P\colon \bar{x}_{pj} \in \bar{X},\, j \in [k]\}}{\arg\max} H_p.$$

For this point p^*, we select the cluster index j to branch on arbitrarily, i.e., we create two subproblems by adding either $\bar{x}_{p^*j} = 1$ or $\bar{x}_{p^*j} = 0$.

5.2 Distance branching

Next, we describe three branching rules with a geometric motivation. The first one, called distance branching, is based on the intuition that clusters should be rather compact (opposed to being spread out). Given the current LP solution with its suggestion for the centroids c^j, $j \in [k]$, the variable $x_{pj} \in \bar{X}$ selected for branching is the one corresponding to the data point p and cluster j that are most apart from each other, i.e., we find

$$(p^*, j^*) \in \underset{\{(p,j) \in P \times [k]\colon \bar{x}_{pj} \in \bar{X}\}}{\arg\max} \|p - c^j\|.$$

Then, we branch on the fractional variable $\bar{x}_{p^*j^*}$, creating two subproblems by adding either $\bar{x}_{p^*j^*} = 1$ or $\bar{x}_{p^*j^*} = 0$. If an optimal cluster is indeed compact, then the 0-subproblem contains an optimal solution. Otherwise, the convexity propagator has the potential to also fix additional variables that lie between p^* and the remaining points of cluster j^* in the 1-subproblem.

5.3 Centrality branching

Since the distance branching rule is tailored towards the extremes of compact vs. far spread-out clusters, the centrality branching rule takes a more balanced approach by selecting a point whose distance to a cluster is not too big. Given the current LP solution with its suggestion for the centroids c^j, $j \in [k]$, we would like to branch on the non-integral variable x_{pj} corresponding to the data point p and cluster j that is lying in the center of the cloud of unassigned data points. To obtain a cheap evaluation, we take the point p^* that is in the center of all centroids, i.e.,

$$p^* \in \underset{\{p \in P\colon \bar{x}_{pj} \in \bar{X},\, j \in [k]\}}{\arg\min} \sum_{j \in [k]} \|p - c^j\|.$$

From this point p^*, we select an arbitrary variable $\bar{x}_{p^*j^*}$, creating two subproblems by adding either $\bar{x}_{p^*j^*} = 1$ or $\bar{x}_{p^*j^*} = 0$. If the distance of p^* to cluster j^* is not too small, then there is a chance that the convexity propagator can fix further data points to be contained in cluster j^* in the 1-subproblem. Opposed to the distance branching rule, however, also the 0-subproblem becomes relevant for the convexity propagator as it might fix some variables to 0 based on its cone propagation.

5.4 Pairs-in-the-middle-of-pairs branching

The next branching rule is a variation of the last one, but now we branch on more general linear inequalities. Given the current LP solution with its suggestion for the centroids c^j,

$j \in [k]$, we would like to branch on the sum of a pair of non-integral variables corresponding to the two data points located in the middle of a pair of clusters.

First assume that there are only two clusters with corresponding centroids c^1 and c^2. We want to find the points that are nearest to the point lying half way from centroid c^1 to centroid c^2. Any point $p \in P$ that lies on the line segment between c^1 and c^2, minimizes the sum $\|p - c^1\| + \|p - c^2\|$ by the triangle inequality of the Euclidean distance. With the same reasoning, the smaller this sum is, the nearer is point p to the line segment between the two clusters. However, there can be multiple points p with the same value for $\|p - c^1\| + \|p - c^2\|$. We are interested in the ones that are nearest to the middle. Thus, we penalize longer distances by minimizing the sum $\|p - c^1\|^2 + \|p - c^2\|^2$ of squares instead.

Now assume that there are more than two clusters which have pairwise the exact same distance of centroids to each other. Then, still looking for the two points $p \in P$ that minimize $\|p - c^j\|^2 + \|p - c^{j'}\|^2$ for $j, j' \in [k]$ with $j \neq j'$ gives us the desired points. Let p and q be the selected points and j and j' be the selected clusters. We then compute the sums in the current LP solution, $\bar{x}_{pj} + \bar{x}_{qj}$ and $\bar{x}_{pj'} + \bar{x}_{qj'}$, and select the sum that is most fractional or least fractional. Both versions are tested in our numerical experiments. Suppose that the first sum is selected, which means that the selected cluster is j. Then, we branch on

$$\bar{x}_{pj} + \bar{x}_{qj} \leq \lfloor \bar{x}_{pj} + \bar{x}_{qj} \rfloor$$

and

$$\bar{x}_{pj} + \bar{x}_{qj} \geq \lceil \bar{x}_{pj} + \bar{x}_{qj} \rceil.$$

6 Primal heuristics

Primal heuristics try to find feasible solutions of good quality in a short amount of time. Having good feasible solutions at hand early in the solving process is crucial. Feasible solutions help to prune branch-and-bound nodes based on bounding as well as to perform further fixings and reductions. Moreover, a user may already be satisfied with the quality of the heuristic solution, such that the solving process can be stopped at an early stage. In this section we present three primal heuristics for the MSSC problem.

6.1 A root-node heuristic

To obtain a first feasible point, i.e., a point for warm-starting, we use the k-means algorithm, which is the most popular heuristic for finding a feasible solution for the MSSC problem; see, e.g., Lloyd [39] and MacQueen [40]. It consists of two main steps. First, given an initial guess for the location of the centroids, each data point is assigned to the nearest centroid. Afterward, each centroid is updated by calculating the mean of the data points assigned to this centroid. This process is repeated until the centroids no longer change. To obtain an initial guess for the location of the centroids, we use the "furthest point heuristic", also known as "Maxmin" [28]. The idea is to select the first centroid randomly within the respective bounding box and then obtain new centroids one by one. In each iteration, the next centroid is the point that is the furthest (max) from its nearest (min) existing centroid. Here, we choose the first data point as the first centroid. For a comparison of several initialization heuristics, see, e.g., Fränti and Sieranoja [23].

6.2 A rounding heuristic

Feasible solutions can be obtained at each node by applying a rounding scheme to the LP solution. We use the rounding heuristic proposed by Sherali and Desai [55]. For completeness, we also describe it here.

Given a non-integral LP solution $(\tilde{x}, \tilde{c})$, or $(\tilde{x}, \tilde{c}, \tilde{\eta})$ for Model (2), at a node of the branch-and-bound tree, we round the non-integral $\tilde{x}$-solution to the closest feasible binary solution $\bar{x}$, while respecting the decisions that have already been made, i.e., if a data point is already assigned to a cluster it will remain in that cluster. First, we ensure that there are no empty clusters by finding, for each $j \in [k]$ with $P_j = \emptyset$, the point $\bar{p} \in P \setminus \cup_{j\in[k]} P_j$ such that $\bar{p} \in \arg\max \left\{\tilde{x}_{pj} \colon p \in P \setminus \cup_{j\in[k]} P_j\right\}$, and setting $\bar{x}_{\bar{p}j} = 1$. To break a tie, the point with smallest index is chosen. Now, to ensure that the point $\bar{p}$ is only in one cluster, we set $\bar{x}_{\bar{p}j'} = 0$, for all $j' \in [k] \setminus \{j\}$.

Furthermore, for each data point $p \in P$ such that $\tilde{x}_{pj}$, $j \in [k]$, is not yet rounded, we find a cluster j^* such that $\tilde{x}_{pj^*} = \max\left\{\tilde{x}_{pj} \colon j \in [k]\right\}$. Again, we break ties by selecting the cluster with the smallest index. Then, we set $\bar{x}_{pj^*} = 1$ and $\bar{x}_{pj} = 0$ for all $j \in [k]\setminus\{j^*\}$. With $\bar{x}$ at hand, we can then compute the centroids for each cluster, see Observation 2.1, and obtain a feasible solution for the MSSC problem.

6.3 An improvement heuristic

Given a feasible solution $(\bar{x}, \bar{c})$ of Model (1) or $(\bar{x}, \bar{c}, \bar{\eta})$ of Model (2), we try to improve this solution by evaluating the loss function (i.e., the intra-variance) within each cluster. For that, consider the weighted value of the loss function restricted to cluster C_j as

$$F_j = \frac{1}{|C_j|} \sum_{p\in C_j} \|p - \bar{c}^j\|^2.$$

It may happen that some clusters have a large loss function value, while some other clusters may have a very small one. Thus, we may find a better solution—regarding the sum of all losses—by splitting a cluster into two smaller clusters and joining two other clusters. This heuristic has been proposed in Burgard et al. [8], where the motivation for its development is explained in more details.

The procedure is described as follows. For each pair of clusters (C_{j_1}, C_{j_2}), we compute their joint centroid and the corresponding total loss via

$$c^{j_1 j_2} = \frac{1}{|C_{j_1}| + |C_{j_2}|} \sum_{p\in C_{j_1}\cup C_{j_2}} p$$

and

$$F_{j_1 j_2} = \frac{1}{|C_{j_1}| + |C_{j_2}|} \sum_{p\in C_{j_1}\cup C_{j_2}} \|p - c^{j_1 j_2}\|^2.$$

Now, consider the set

$$\Psi := \left\{(C_{j_1}, C_{j_2}, C_{j_3}) \colon F_{j_1 j_2} < F_{j_3}\right\},$$

which is the set of all possible combinations of three clusters such that the total loss within two joined clusters is smaller than the total loss within a third cluster. Note that the set Ψ can be empty. If so, this means that we cannot obtain a better solution by joining two clusters and splitting another one. On the other hand, i.e., if there exists $(C_{j_1}, C_{j_2}, C_{j_3}) \in \Psi$, then the

total loss of the joined clusters C_{j_1} and C_{j_2} is smaller than the total loss within cluster C_{j_3}. Thus, we obtain a better solution by joining C_{j_1} and C_{j_2} and by splitting cluster C_{j_3} into two smaller clusters. To this end, we update the centroids in such a way that the clusters C_{j_1} and C_{j_2} are now one cluster with centroid $c^{j_1 j_2}$, cluster C_{j_3} receives two new centroids, and the other centroids remain the same, i.e.,

$$\hat{c}^{j_1} \leftarrow c^{j_1 j_2}, \quad \hat{c}^{j_2} \leftarrow \tilde{c}, \quad \hat{c}^{j_3} \leftarrow \tilde{c}',$$
$$\hat{c}^{j} \leftarrow \bar{c}^{j} \quad \text{for all} \quad j \notin \{j_1, j_2, j_3\},$$

where $\tilde{c}$ and $\tilde{c}'$ are obtained as follows. First we find the two furthest points in C_{j_3} to be the initial guesses for the location of the centroids, i.e.,

$$(\tilde{c}, \tilde{c}') \in \underset{p, p' \in C_{j_3}}{\arg\max} \left\{ \|p - p'\|^2 \right\}.$$

Next, each point in C_{j_3} is assigned to the closest centroid, either $\tilde{c}$ or $\tilde{c}'$. Then, the centroids $\tilde{c}$ and $\tilde{c}'$ are updated based on this assignment. Now, the update of the assignments and centroids is repeated until they do not change anymore. This way we obtain the new centroids $\tilde{c}$ and $\tilde{c}'$ that give us the desired splitting of cluster C_{j_3}.

Finally, if the set Ψ has more than one element, then we repeat the process starting with the element $(C_{j_1}, C_{j_2}, C_{j_3})$ that gives the minimum ratio $F_{j_1 j_2}/F_{j_3}$. Each time an element $(C_{j_1}, C_{j_2}, C_{j_3})$ is used, we exclude all the elements that contain C_{j_1}, C_{j_2}, or C_{j_3}, because these clusters have already been modified.

With $\hat{c}$ at hand, we can easily compute $\hat{x}$ and, thus, a new feasible solution $(\hat{x}, \hat{c})$ is obtained. If the objective function value is better at this new solution, then we have found an improved solution out of $(\bar{x}, \bar{c})$.

7 Symmetry breaking

Note that both Model (1) and (2) are symmetric with respect to cluster assignments. That is, once a feasible solution has been found, one can generate equivalent (symmetric) solutions by exchanging the labels of the clusters. Such symmetries are known to deteriorate the performance of search-based approaches like branch-and-bound, because symmetric subproblems are created repeatedly without providing the solver with new information. Such cluster symmetries can be handled in both models by imposing additional restrictions on the x-variables. If we interpret x as a binary matrix whose columns are labeled by clusters, then we can handle symmetries by enforcing that the columns of x need to be sorted lexicographically non-increasing. Since each row of matrix x has exactly one 1-entry due to (1b) and (2c), the lexicographic sorting can be imposed by orbitopal fixing, see Kaibel et al. [37], and separating the symmetry handling inequalities developed by Kaibel and Pfetsch [36].

8 Numerical experiments

In this section, we report extensive computational results that show the benefits of the techniques proposed in Sects. 3–6. To this end, we have incorporated all these techniques into the state-of-the-art solver SCIP; see Gamrath et al. [26]. As a reference for comparison, we use plain SCIP for both problem formulations (1) and (2). That is, we solve the two formulations without our problem-specific enhancements but with enabled symmetry handling.

To conduct the experiments, we use different test sets from the literature, which contain both real-world as well as synthetic instances. The test sets and the general computational setup are described in Sects. 8.1 and 8.2, respectively. Then, in Sect. 8.3, we start the discussion of the numerical results for the case $k = 2$. We evaluate the benefits of each particular technique and indicate which setting performs best. Next, in Sect. 8.4, we repeat the discussion but for the case $k = 3$. Finally, in Sect. 8.5, we present results on a larger test set in order to draw solid and comprehensive conclusions about the performance of the novel techniques.

8.1 Test sets

We evaluate the impact of the presented algorithmic ideas for solving the MSSC problem using both synthetic and real-world test sets. To be able to draw conclusions on a reliable basis, we have collected all publicly available instances that have been used in the related literature for solving the MSSC problem to global optimality. Thus, to the best of our knowledge, our results are based on the largest publicly available test set for the MSSC problem consisting of realistic instances. Specifically, we use the instances that have been used in Aloise and Hansen [4], Sherali and Desai [55], as well as in Aloise et al. [5]. Since these instances come from different sources, we provide the source for every instance in Table 1. The synthetic test set has been proposed in Fränti and Sieranoja [22]. The authors show that these synthetic instances cover a wide range of classic MSSC instances. In particular, the test set contains instances with different degrees of overlap, density, and sparsity of data points.

Note that some of the synthetic instances contain data points with very large coordinate values. In preliminary experiments, we have observed that this leads to very large big-M values in Model (2), which in turn causes numerical instabilities. To avoid numerical issues, we therefore re-scale these instances as follows. First, for each coordinate, we shift the data points such that their coordinate-wise minimum and maximum value is the same to have a "symmetric" distribution. Then, we re-scale the data points if they do not fit into $[-10^3, 10^3]^d$. More precisely, for each dimension $i \in [d]$, we compute the maximum and minimum coordinate value obtaining $\bar{v}_i$ and $\underline{v}_i$, respectively. Then, we take $u_i = 0.5(\bar{v}_i + \underline{v}_i)$ and shift each data point p obtaining $\hat{p}_i = p_i - u_i$ for all $i \in [d]$. If we do this for all data points, they get centered around the origin. Now, if $\underline{w}_i = \underline{v}_i - u_i < -10^3$ or $\bar{w}_i = \bar{v}_i - u_i > 10^3$ holds, we re-scale the data. The desired new bounds then are $\underline{z}_i = -10^3$ and $\bar{z}_i = 10^3$. Thus, the re-scaled data point $\tilde{p}$ is

$$\tilde{p}_i = \frac{\hat{p}_i - \underline{w}_i}{\bar{w}_i - \underline{w}_i} \cdot (\bar{z}_i - \underline{z}_i) + \underline{z}_i, \quad i \in [d].$$

The corresponding instances that needed to be re-scaled are s1, s2, s3, s4, and unbalance.

8.2 Computational setup

To conduct our experiments, we use SCIP 7.0.3 as a branch-and-bound framework. All LP relaxations are solved using CPLEX 12.8. Our novel techniques discussed in Sects. 3–6 are implemented as SCIP plugins written in C/C++ and our code is publicly available at GitHub[2] (git hash 19003a37). To handle symmetries, we use the orbitope constraint handler plugin of SCIP, which implements orbitopal fixing and the symmetry handling inequalities as mentioned in Sect. 7. To compute convex hulls and cones in the convexity propagator proposed in

[2] https://github.com/christopherhojny/globally-solving-MSSC.

Table 1 Information about the test sets

ID	Instance	Reference	n	d
1	Fisher150iris	Dua and Graff [16] and Fisher [19]	150	4
2	German22	Späth [57]	22	2
3	German59	Späth [57]	59	2
4	body-measurements	Heinz et al. [33]	507	5
5	cities-coord-202	Grötschel [29]	202	2
6	cities-coord-666	Grötschel [29]	666	2
7	concrete-compressive	Dua and Graff [16]	1030	8
8	glass-identification	Dua and Graff [16]	214	9
9	image-segmentation	Dua and Graff [16]	2310	19
10	padberg-rinaldi-hole-dri	Padberg and Rinaldi [42]	2392	2
11	reinelt-hole-drilling	Reinelt [51]	1060	2
12	ruspini	Ruspini [52]	75	2
13	telugu-indian-vowel	Pal and Majumder [43]	871	3
14	a1	Fränti and Sieranoja [22]	3000	2
15	a2	Fränti and Sieranoja [22]	5250	2
16	a3	Fränti and Sieranoja [22]	7500	2
17	dim	Fränti and Sieranoja [22]	1024	32
18	g2-2-30	Fränti and Sieranoja [22]	2048	2
19	g2-2-50	Fränti and Sieranoja [22]	2048	2
20	g2-2-70	Fränti and Sieranoja [22]	2048	2
21	s1	Fränti and Sieranoja [22]	5000	2
22	s2	Fränti and Sieranoja [22]	5000	2
23	s3	Fränti and Sieranoja [22]	5000	2
24	s4	Fränti and Sieranoja [22]	5000	2
25	unbalance	Fränti and Sieranoja [22]	6500	2

The first part corresponds to real-world test sets whose instances come from different sources. The second part corresponds to the synthetic test set

Sect. 4.2, we use the Qhull[3] C++ interface proposed by Barber et al. [6]. We have also conducted experiments using the CDD library [24] for computing convex hulls and cones, but due to numerical instabilities therein, we decided to use Qhull. Moreover, since we observed that many of SCIP's internal heuristics require a lot of running time without generating a feasible solution, we disabled these heuristics. A list of disabled heuristics can be found in "Appendix A". All computations were performed on a computer with two Intel Xeon CPU E5-2699 v4 at 2.20 GHz (2×44 threads) and 756 GB RAM. The time limit of all computations is 1 h per instance.

In the following, we discuss the impact of our techniques on solving the MSSC problem for $k = 2$ and $k = 3$ clusters. We only report on aggregated results in the discussion and refer the reader to "Appendix B" for results per instance. The tables that we present show for both the quadratic and epigraph formulation the mean number of nodes in the branch-and-bound tree (column #nodes), the mean running time per instance in seconds (time), and the number of solved instances (#solved). Instances that cannot be solved within the time limit contribute

[3] http://www.qhull.org.

Table 2 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different heuristics for 2 clusters

Setting			Quadratic model			Epigraph model		
round	impr	init	#nodes	time	#solved	#nodes	time	#solved
0	0	0	14,218.4	3406.45	1	23,275.1	2959.08	1
1	1	1	17,106.6	3273.37	1	19,884.9	2945.94	1

3600 s to the mean time value. Moreover, we report on the used setting, where each of the following subsections describes how the settings are encoded in the tables. All mean numbers of measurements $t_1, \ldots, t_n$ are provided as shifted geometric means $\prod_{i=1}^{n}(t_i + s)^{1/n} - s$ to reduce the impact of outliers. For time we use a shift of $s = 10$ and for nodes a shift of $s = 100$.

8.3 Discussion of the numerical results for 2 clusters

We start with the discussion of the numerical results for the case when there are 2 clusters. First, we apply plain SCIP to all the 25 instances presented in Table 1. Afterward, we gradually enable our techniques in SCIP and evaluate the benefits of each particular technique as well as the benefits of different combinations of techniques. To allow for a concise encoding, we abbreviate the different techniques as described below. Whether a technique is enabled (resp. disabled) is encoded by 1 (resp. 0) in the corresponding tables.

8.3.1 Primal heuristics

We start by evaluating the impact of primal heuristics. A summary of the obtained results is presented in Table 2, where "round.", "impr.", and "init", serve as abbreviations for the rounding, improvement, and root-node heuristic, respectively. Recall that the improvement heuristic is only active for $k > 2$, i.e., it has no effect in the experiments discussed next. The first row shows the results obtained by plain SCIP. It can be directly seen that the MSSC problem is extremely hard to solve. Note that SCIP is able to solve only 1 instance to global optimality, regardless of which model is used. Enabling all primal heuristics still does not allow to solve more instances. However, we can see that the mean running time decreases in both models, where the impact is larger for the quadratic model. That is, the single instance that can be solved is solved in approximately 3.9% faster using heuristics.

Let us stress that, based on preliminary experiments, the main difficulty of solving the MSSC problem to global optimality is to obtain good dual bounds in a reasonable amount of time. For this reason, the impact of heuristics on the solving process is expected to be minor in comparison to the impact of techniques that improve the dual bound. However, since we needed to disable many of SCIP's internal heuristics as described above, we enable all our heuristics in the following experiments as their running time is low and they produce good solutions.

8.3.2 Propagators

The results of our experiments regarding propagators are summarized in Table 3, where "bary.", "conv.", "cone", and "dist." abbreviate the barycenter, convexity, cone, and distance

Table 3 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different propagators for 2 clusters

Setting				Quadratic model			Epigraph model		
bary	conv	cone	dist	#nodes	time	#solved	#nodes	time	#solved
0	0	0	0	17,106.6	3273.37	1	19,884.9	2945.94	1
1	0	0	0	91,401.7	2853.27	1	18,868.4	2863.78	2
0	1	0	0	3607.3	1735.34	4	10,478.6	1902.80	4
0	1	1	0	3029.8	1565.24	5	7448.3	1529.15	6
1	1	1	0	5996.1	835.39	8	6784.9	1527.04	5

propagator, respectively. However, we do not include the results using the distance propagator here, since in preliminary numerical experiments we observed that this propagator is not able to derive many reductions if used alone. In later experiments, we will enable it again to investigate whether it is able to improve the solution process if also other components are enabled.

The first row of results in Table 3 corresponds to the setting where only primal heuristics are enabled. It can be directly seen that as more propagators are enabled, more instances are solved. Without propagators only 1 instance is solved to global optimality. Using all our propagators, we are able to solve 8 instances with the quadratic model. Thus, the geometric ideas incorporated into the propagators are an important component to solve the MSSC problem effectively. In particular, plain SCIP is not able to make use of the simple geometric observations on its own. In the following, we discuss the benefits of each particular propagator in more detail.

8.3.3 Barycenter propagator

By using the barycenter propagator and the quadratic model, much more nodes can be processed if compared with the previous setting and, more importantly, in significantly less time. The reason for this is that the barycenter propagator is able to perform many reductions, which in turn simplifies the LP relaxations. As a consequence, the dual bounds obtained with the quadratic model drastically improve by using the barycenter propagator. This can be clearly seen in Fig. 2, where we plot the instances vs. the corresponding gap between the primal and dual bounds.

This already demonstrates the great benefit that the barycenter propagator adds to the solution process. As discussed in Sect. 4.1.1, the barycenter propagator is less powerful for the epigraph model as it is for the quadratic model, which is also reflected in the results. Nevertheless, it allows 1 more instance to be solved to global optimality and it slightly reduces the running times. Looking at Fig. 2 again, we also see that for many instances the gaps improve.

8.3.4 Convexity+Cone propagator

The convexity propagator is based on geometric ideas and is extremely powerful. Using only this propagator alone and heuristics, we can already solve 3 more instances if the quadratic model is used, and 2 more instances if the epigraph model is used. Without the convexity propagator, these instances cannot be solved. This technique drastically helps in the solution

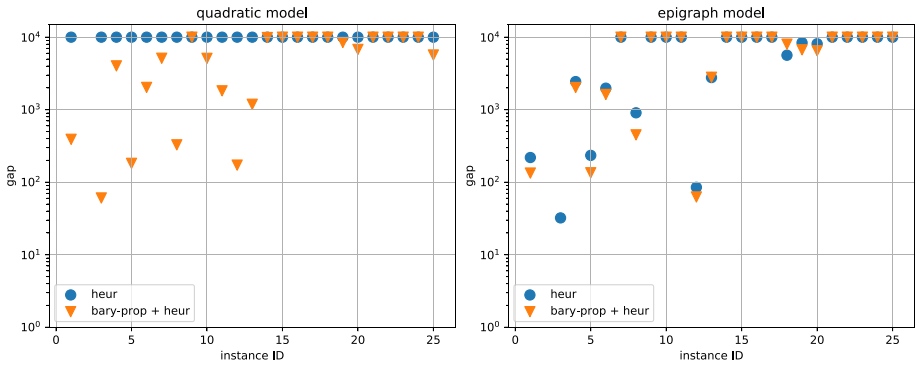

Fig. 2 Instance ID vs. gap (in percentage and log-scale) for 2 clusters. Since the y-axis is in log-scale, if a particular instance is solved to global optimality by a particular method, then the gap is zero and hence it does not appear in the plot. Whereas, if the gap is equal or larger than 10^4, then it assumes the gap limit of 10^4 in the plot

process of the MSSC problem. Besides allowing more instances to be solved, it also requires half of the time that was needed before. Moreover, the number of nodes that need to be processed to solve the instances also reduces significantly.

Using the cone propagation in combination with the convexity propagator, this effect is even more pronounced. It allows 1 more instance to be solved if the quadratic model is used, and 2 more instances if the epigraph model is used and results in much lower mean running times.

8.3.5 Barycenter+Convexity+Cone propagators

Although the barycenter and convexity-cone propagators alone already significantly improved SCIP's performance, their combination allows to solve three further instances in the quadratic model. This results in a significant reduction of running time by approximately 46%. Interestingly, the mean number of nodes in the combined setting is roughly twice as large as if just the convexity-cone propagator is used. This again shows that the reductions found by the propagators simplify the structure of relaxations drastically, e.g., because fixed x-variables remove non-convex expressions from the quadratic model. These reductions allow SCIP to process more nodes, which in turn allows to solve more instances. In the epigraph model, the combination of the three propagators does not qualitatively change the results. Although 1 less instance can be solved, for many instances the gaps improved; see Fig. 3.

We conclude that, for both the quadratic and epigraph model, our propagation algorithms are an important component to solve the MSSC problem to global optimality. In particular, using combinations of these propagators creates synergies that allow to solve more instances in comparison with just using a single propagator, where the effect is more prominent for the quadratic model.

8.3.6 Cutting planes

Next, we evaluate the impact of cutting planes on SCIP. As before, we also enable all heuristics and, due to the positive effect of propagators, also the convexity-cone and barycenter propagator. Preliminary numerical results showed that by localizing the cardinality cuts, no

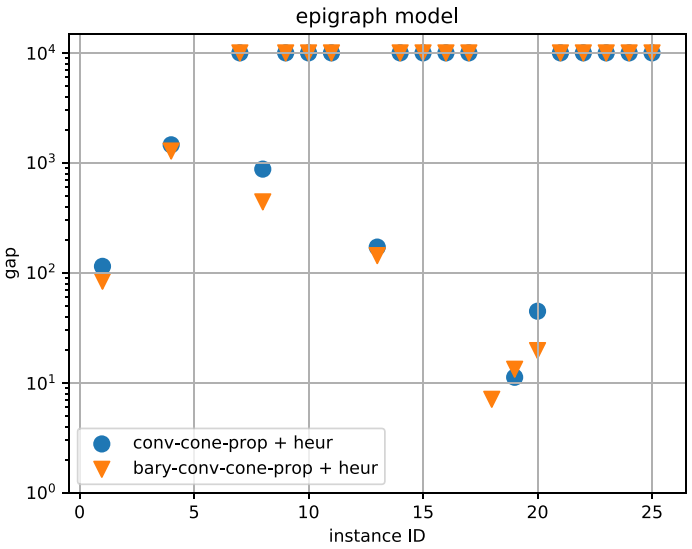

Fig. 3 Comparison of gaps using different propagators for 2 clusters

Table 4 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using or not the OA cuts for 2 clusters

Setting	Epigraph model		
	#nodes	time	#solved
w/o OA cuts	6784.9	1527.04	5
w/ OA cuts	1526.8	1552.97	5

positive impact on the solution process can be achieved in general. Therefore, we focus only on the outer-approximation (OA) cuts. Since these cuts are only applicable for the epigraph model, we concentrate only on the epigraph model in the following discussion. Table 4 summarizes our results.

At first glance, it seems that OA cuts only have a minor impact on SCIP's performance as the number of solved instances does not change. Comparing the gaps with and without OA cuts, however, reveals a clear impact; see Fig. 4. For 8 instances, we observe a change in the gap if cuts are enabled. In three cases, the gaps slightly degrade when OA cuts are enabled. For the remaining five instances, however, OA cuts either reduce or drastically reduce the gap. Thus, although no clear trend is visible, we may conclude that OA cuts are helpful when solving MSSC problems. The effect of OA cuts is less pronounced compared to the effect of propagators, which might be explained by the fact that OA cuts do not exploit the specific problem structure of MSSC. In contrast to this, our novel propagator techniques are tailored to the MSSC problem and thus allow stronger reductions.

8.3.7 Distance propagator

As reported above, the distance propagator alone is not able to significantly improve SCIP's performance. For this reason, we test its effect if also further components are enabled. From Table 5, we can see that using the distance propagator together with our other techniques has a slightly positive effect. Therefore, it is also enabled in the experiments discussed next.

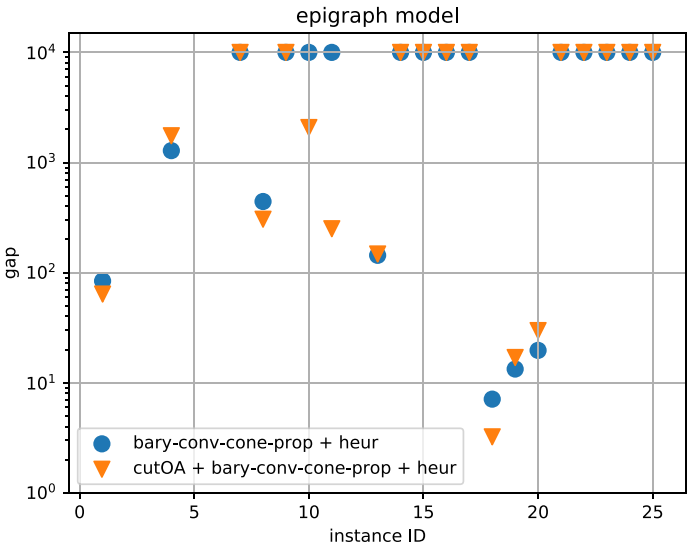

Fig. 4 Comparison of gaps using propagators with the OA cuts enabled or not for 2 clusters

Table 5 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using or not distance propagator for 2 clusters

Setting	Quadratic model			Epigraph model		
	#nodes	time	#solved	#nodes	time	#solved
w/o dist. propagator	5996.1	835.39	8	1526.8	1552.97	5
w/ dist. propagator	5976.4	813.00	8	1809.9	1497.49	6

8.3.8 Branching rules

The last components to be tested are branching rules. We have implemented all branching rules described in Sect. 5. Preliminary numerical experiments, however, revealed that only the entropy and distance branching rules may be beneficial for some instances. In contrast, the centrality and pairs-in-the-middle-of-pairs harm the solution process leading to a less well-performing code. Therefore, we focus only on the branching rules that have a positive impact on some instances in the following discussion. We present the summary results in Table 6. By "standard" we refer to SCIP's default branching rule.

Our experiments show that no branching rule dominates the others. On the one hand, by using the distance branching rule and the quadratic model, 1 additional instance can be solved. On the other hand, the number of nodes and the time required increases. If the epigraph model is used, then the entropy branching rule is performing best: The number of solved instances remains the same but the running times are slightly lower. The overall impact of branching rules, however, seems to heavily depend on the underlying instance to solve, which does not allow us to provide a clear winner.

Table 6 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different branching rules for 2 clusters

Setting	Quadratic model			Epigraph model		
	#nodes	time	#solved	#nodes	time	#solved
Standard	5976.4	813.00	8	1809.9	1497.49	6
Entropy	6613.7	863.79	8	2541.1	1446.77	6
Distance	6942.0	845.73	9	1878.8	1488.07	6

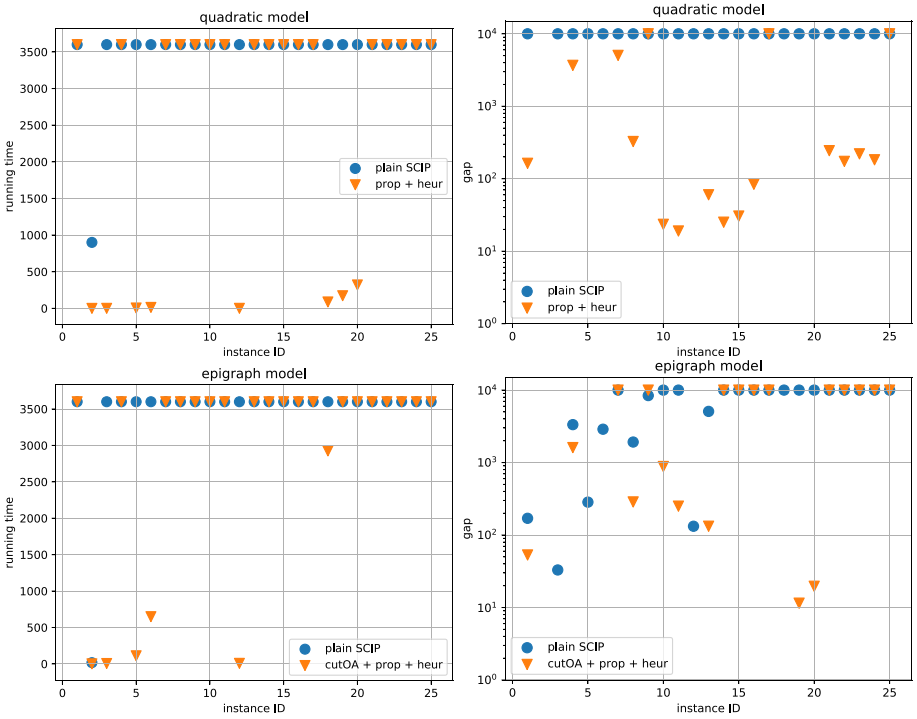

Fig. 5 Running times and gaps comparison between plain SCIP and SCIP enabled with the best setting for 2 clusters

8.3.9 Best setting

To conclude the discussion of the numerical results for $k = 2$, we show a comparison of plain SCIP with the best combination of the techniques proposed in this paper. The latter comprises primal heuristics, propagators, OA cuts (for the epigraph model), and the standard branching rules of SCIP, since our branching rules and standard branching rules are performing equally good on average. This comparison in shown in Fig. 5.

Regarding the performance of plain SCIP, we emphasize that the dual bounds found by SCIP in the quadratic model are very weak which leads to very large gaps. In contrast to this, if the epigraph formulation is used, better dual bounds can be obtained by using plain SCIP. Despite their simplicity, the plots show that our novel geometric ideas drastically improve

Table 7 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different heuristics for 3 clusters

Setting			Quadratic model			Epigraph model		
round	impr	init	#nodes	time	#solved	#nodes	time	#solved
0	0	0	7365.1	3600.00	0	5756.0	3094.27	1
1	1	1	8400.1	3600.00	0	6998.6	3113.39	1

on the performance of SCIP, thus adding powerful methods to the toolbox for solving the MSSC problem to global optimality if $k = 2$.

These methods work particularly well for instances with 2-dimensional data and the number of data points not exceeding 2048 as almost all such instances from our test set, see Table 1, can be solved to global optimality within the time limit by the quadratic model. The only exception is instance 11, which terminates after one hour with a gap of 19.02%. For more detailed results of the best setting we refer the reader to Table 21 in the "Appendix B".

8.4 Discussion of the numerical results for 3 clusters

We now turn our attention to the experiments for $k = 3$. The MSSC problem is much harder to solve to global optimality in this setting. We proceed as in the last section.

8.4.1 Primal heuristics

The summary results of plain SCIP and SCIP enabled with our primal heuristics are presented in Table 7.

By using plain SCIP and the epigraph model, we can solve only 1 instance to global optimality. Enabling heuristics does not change the number of solved instances. This is in line with the observations made above: the main difficulty in solving the MSSC problem to global optimality is to provide tight dual bounds, which are not provided by primal heuristics. However, we observe that by enabling the heuristics in the epigraph model, the primal-dual gap improves for many instances substantially; see Fig. 6. For this reason, we enable heuristics in the following experiments.

8.4.2 Propagators

Next, we evaluate the impact of our propagation techniques for $k = 3$. In Table 8, we show the summarized results obtained by activating our primal heuristics and the propagators. Taking a general look at the results and comparing the first row (SCIP + heuristics) with the last row, we can see that 2 more instances can be solved to global optimality, using either the quadratic or the epigraph model. Thus, although the MSSC problem for $k = 3$ is much harder to solve than for $k = 2$, the propagation techniques are still helpful; see also Fig. 7 where we compare the gaps obtained by using propagators. Therefore, in what follows, we discuss the benefits of the separate propagators in turn.

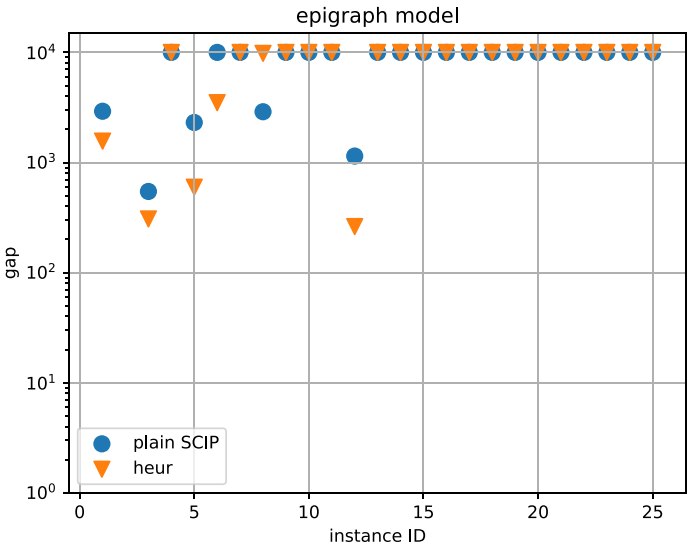

Fig. 6 Comparison of gaps using or not the heuristics for 3 clusters

Table 8 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different propagators for 3 clusters

Setting				Quadratic model			Epigraph model		
bary	conv	cone	dist	#nodes	time	#solved	#nodes	time	#solved
0	0	0	0	7365.1	3600.00	0	5756.0	3094.27	1
1	0	0	0	20,902.8	2983.99	1	7409.5	2977.89	1
0	1	0	0	8066.6	3250.65	1	9169.4	2939.98	1
0	1	1	0	11,815.9	3385.49	1	7196.0	2743.45	3
1	1	1	0	26,530.7	2856.07	2	6057.0	2760.58	3

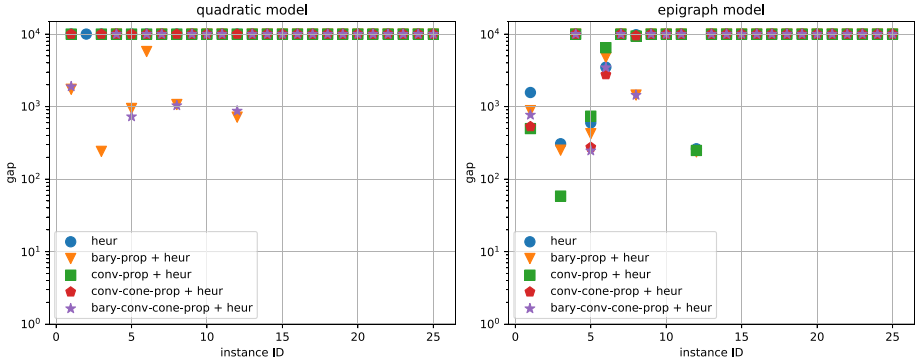

Fig. 7 Comparison of gaps using different propagators for 3 clusters

8.4.3 Barycenter propagator

By enabling only the barycenter propagator, we see that more nodes can be explored in the branch-and-bound tree. Particularly, when the quadratic model is used, there is a huge difference in the number of nodes when comparing the first two rows of Table 8. Moreover, the barycenter propagator allows one more instance to be solved, which could not be solved before by the quadratic model, resulting in a much lower mean running time. Finally, although the number of solved instances increases only slightly, we can see that the barycenter propagator has also a very positive effect on the other instances: the primal-dual gap improves significantly, in particular, for the quadratic model; see Fig. 7.

Besides the improvement in the gaps for the epigraph model, the barycenter propagator also allows more nodes to be explored while reducing the running times considerably. Therefore, also for $k = 3$, the barycenter propagator allows to simplify the relaxations used in branch-and-bound.

8.4.4 Convexity+Cone propagator

If we use only the convexity propagator and compare the results with plain SCIP and enabled heuristics, then we see that more nodes can be processed in significantly less time. Additionally enabling the cone propagator also allows to solve two more instances in the epigraph model; see Fig. 7 again. Regarding the quadratic model, using the convexity and cone propagators allows to process more nodes in the branch-and-bound tree. This, however, comes at the price that the solvable instance requires more time. We conclude that the convexity and cone propagators enhance the solution process, where the effect is more dominant for the epigraph model.

8.4.5 Barycenter+Convexity+Cone propagator

The quadratic model clearly benefits from using the barycenter propagator together with convexity and cone propagation as one more instance can be solved and in less time. For the epigraph model, no significant change regarding running times can be observed, however it has a positive effect on some gaps; see Fig. 7.

8.4.6 Cutting planes

Regarding the cutting planes, we again focus only on the OA cuts, since the localized version of the cardinality cuts does not positively affect the solution process in general. In Table 9 we present the aggregated results for the epigraph model with and without OA cuts. Note that by using the OA cuts we can solve the same number of instances, while requiring less time and much less nodes. The OA cuts help mainly in terms of dual bounds; see also Fig. 8. Therefore, we conclude that the OA cuts are also beneficial for the epigraph model in the harder case of $k = 3$.

8.4.7 Distance propagator

The distance propagator does not impact the solution process if used standalone. The reason is that, for most of the instances, the propagator does not find any reduction, which is to be expected because fixing x-variables to 0 is less powerful than fixing them to 1 (as the

Table 9 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using OA cuts for 3 clusters

Setting	Epigraph model		
	#nodes	time	#solved
w/o OA cuts	6057.0	2760.58	3
w/ OA cuts	3601.0	2686.29	3

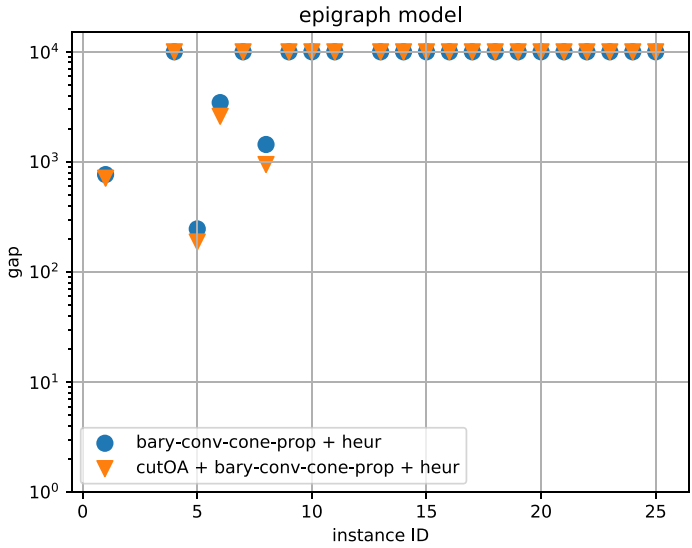

Fig. 8 Comparison of gaps using different propagators and using or not the OA cuts for 3 clusters

Table 10 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using or not the distance propagator for 3 clusters

Setting	Quadratic model			Epigraph model		
	#nodes	time	#solved	#nodes	time	#solved
w/o dist. propagator	26,530.7	2856.07	2	3601.0	2686.29	3
w/ dist. propagator	29,947.6	2821.97	2	4602.5	2629.80	3

convexity propagator does, for instance). However, as more reductions and fixings are being made by other components, the distance propagator slightly helps; see the summary results shown in Table 10.

The distance propagator improves the solution process of both models, since more nodes can be explored and in less time while solving the same number of instances. Therefore, we enable this propagator as well in the following experiments.

8.4.8 Branching rules

Finally, we report on the effect of branching rules. Again, the centrality and pairs-in-the-middle-of-pairs rule hinder the solution process. Therefore, we focus only on the entropy and distance branching rules, which are summarized in Table 11. The branching rules do not really change the results. In Fig. 9, we see that the entropy branching rule has a positive effect

Table 11 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different branching rules for 3 clusters

Setting	Quadratic model			Epigraph model		
	#nodes	time	#solved	#nodes	time	#solved
standard	29,947.6	2821.97	2	4602.5	2629.80	3
entropy	31,264.8	3059.02	1	5839.5	2843.52	2
distance	32,084.3	2833.34	2	7674.1	2628.51	3

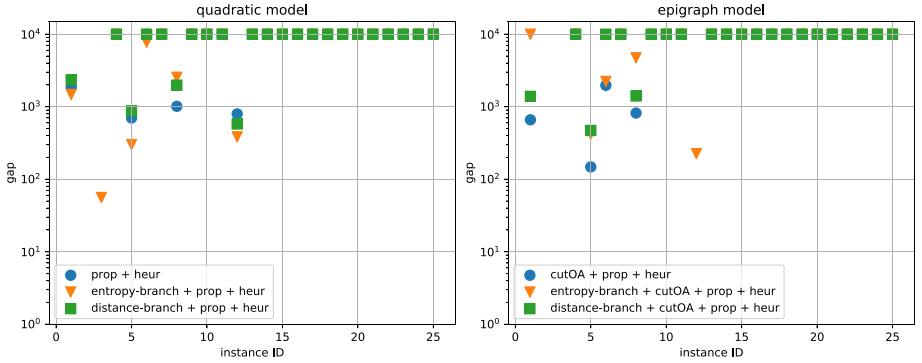

Fig. 9 Comparison of gaps using different branching rules for 3 clusters

on four gaps of the quadratic model, but harms the solution process of two other instances. Moreover, it requires more time. For the epigraph model, it is visibly bad. The standard and distance branching rules perform equally good for the quadratic model; see Fig. 9 again. While, for the epigraph model, the standard branching rule performs best.

8.4.9 Best setting

To finalize the discussion of the numerical results for the case $k = 3$, we again compare plain SCIP with the best setting so far. We consider as the best setting the following: the primal heuristics, the four propagators, the OA cuts (for the epigraph model), and the standard branching rules that are part of SCIP, all enabled. The comparison is presented in Fig. 10.

As for the case $k = 2$, the dual bounds obtained with the epigraph model are better than the ones obtained with the quadratic model if plain SCIP is used. Although not many instances can be solved by using our techniques, the improvement that they bring to the solution process is significant in terms of primal-dual gaps. The instances solved to global optimality within the time limit are the smallest instances in our test set in terms of data points, i.e., instances 2 and 3 can be solved by the quadratic model and the epigraph model additionally solves instance 12; see Table 1 for details about these instances and Table 31 in the "Appendix" for detailed results per instance.

8.5 Discussion of the numerical results for samples of instances

The results discussed in the last two sections indicate that our techniques are highly beneficial for SCIP when solving the MSSC problem with a number of data points that is not too large.

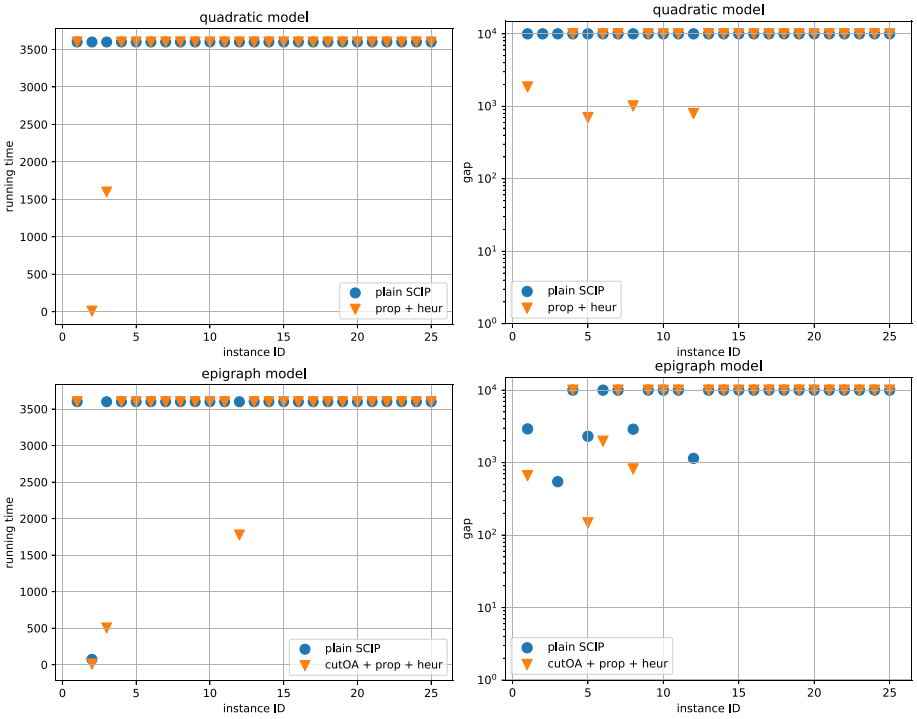

Fig. 10 Running times and gaps comparison between plain SCIP and SCIP enabled with the best setting for 3 clusters

On the tested instances, this roughly means $n < 1000$ and $d = 2$. To further support this hypothesis, we conducted experiments on a broader test set of smaller instances. We have formed 10 new sub-test sets by extracting samples of sizes $\{100, 200, \ldots, 1000\}$ from the two-dimensional and large instances a1, a2, a3, s1, s2, s3, s4. Each sub-test set is comprised of 7 instances, which in total yields additional 70 instances. To sample the data, we use the Python routine skopt.sampler.Sobol; for Sobol sequences see Sobol' [56]. In Fig. 11, we show two examples of these samples (or sub-instances). The blue points represent the original data points, while the red crosses represent the obtained sample. We believe that these samples extract meaningful information of the larger instances as the samples look similar to the full instances.

In the following, we investigate how our novel methods scale with increasing problem size. We focus on the case $k = 2$, because the instances for $k = 3$ are still very challenging to solve and drawing reliable conclusions is difficult.

8.5.1 Primal heuristics and propagators

Table 12 shows aggregated results obtained by enabling our primal heuristics and propagators in SCIP. We can immediately see that SCIP's performance clearly improves if more of our components are enabled as more instances can be solved and the running times decrease drastically. In particular, enabling all our techniques allows us to solve using the quadratic model all 7 instances per sub-test set for up to 500 data points. Using the epigraph model, all instances with up to 900 data points can be solved to global optimality within the time limit.

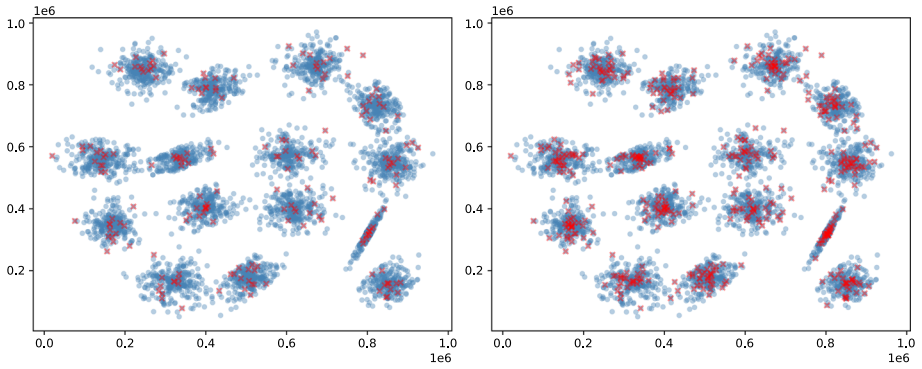

Fig. 11 Example of samples extracted from the s1 instance. The sample sizes are: 200 (left) and 500 (right). The blue points represent the original data points, while the red crosses represent the sampled data points

Moreover, for both models, we can solve almost all instances with 1000 data points if all our methods are enabled. Without our techniques, SCIP cannot solve a single instance even if the number of data points is 100.

8.5.2 Cutting planes

We again only focus on the OA cuts and on the epigraph model. The summarized results are presented in Table 13. Interestingly, the OA cuts do more harm than good in this experiment. We can thus conclude that since these cuts are not based on problem-specific clustering ideas, they do not help as much as the propagators do for solving the MSSC problem effectively.

8.5.3 Branching rules

Now we evaluate the performance of our branching rules. Again, we focus only on the two branching rules that yield some improvement in the solution process of the MSSC problem. The comparison results are displayed in Table 14.

The entropy branching rule performs better if the epigraph model is used, whereas both the distance and the standard branching rules of SCIP are equally good if the quadratic model is used.

8.5.4 Best setting

As in the last sections, we conclude the analysis by comparing plain SCIP with the best setting, see Fig. 12. One can clearly see the significant improvement achieved by using our techniques. Moreover, it is also visible that as the sizes of the instances get larger, the harder the instances become. Therefore, if the size of the MSSC problem at hand is not too large, i.e., the number of data points n is around 1000, the dimension d is 2, and the number of clusters k is 2, then our techniques can be efficiently used to solve the problem to global optimality in just a few seconds. However, it is important to note that this is the case only for the instances we consider because the difficulty of a MSSC problem does not solely depend on the size of the instance but also depends on the structure of the given data points.

Table 12 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different propagators for the sampled instances and 2 clusters

Setting				Quadratic model			Epigraph model		
bary	conv	cone	dist	#nodes	time	#solved	#nodes	time	#solved
Sample (100)									
0	0	0	0	1,48,998.2	3600.00	0	145,954.6	3600.00	0
1	0	0	0	1,430,149.6	3600.00	0	124,688.3	3600.00	0
0	1	0	0	11,516.5	55.03	7	8304.1	129.77	7
0	1	1	0	5084.5	24.22	7	2026.4	39.89	7
1	1	1	0	1715.9	1.88	7	2171.7	43.00	7
1	1	1	1	1715.9	1.90	7	2171.7	42.65	7
Sample (200)									
0	0	0	0	66,858.5	3600.00	0	75 996.9	3600.00	0
1	0	0	0	613,716.4	3600.00	0	62 344.1	3600.00	0
0	1	0	0	38,690.2	512.18	7	39 877.6	1291.27	7
0	1	1	0	9265.9	115.06	7	3375.9	134.83	7
1	1	1	0	3239.7	6.96	7	3094.1	123.98	7
1	1	1	1	3229.1	6.95	7	3094.1	124.43	7
Sample (300)									
0	0	0	0	29,917.0	3600.00	0	57,146.0	3600.00	0
1	0	0	0	395,799.2	3600.00	0	40 245.3	3600.00	0
0	1	0	0	82,681.6	2013.81	6	60 333.7	2995.39	4
0	1	1	0	13,246.8	393.49	6	4834.3	284.08	7
1	1	1	0	4624.5	12.25	7	4387.3	258.03	7
1	1	1	1	4632.5	12.45	7	4387.3	262.19	7
Sample (400)									
0	0	0	0	19,447.9	3600.00	0	43,183.8	3600.00	0
1	0	0	0	289,341.7	3600.00	0	32,858.2	3600.00	0
0	1	0	0	61,044.7	3600.00	0	50,033.9	3600.00	0
0	1	1	0	18,772.3	525.20	7	5914.5	461.67	7
1	1	1	0	5366.1	16.29	7	5430.1	415.82	7
1	1	1	1	5356.3	16.24	7	5430.1	408.90	7
Sample (500)									
0	0	0	0	13,686.3	3600.00	0	37,474.9	3600.00	0
1	0	0	0	217,871.4	3600.00	0	25,936.2	3600.00	0
0	1	0	0	37,497.1	3600.00	0	35,275.6	3600.00	0
0	1	1	0	19,621.3	1307.09	5	6170.0	627.29	7
1	1	1	0	5953.0	20.67	7	6998.6	672.39	7
1	1	1	1	5956.7	20.94	7	6998.6	681.69	7

Table 12 continued

Setting				Quadratic model			Epigraph model		
bary	conv	cone	dist	#nodes	time	#solved	#nodes	time	#solved
Sample (600)									
0	0	0	0	9475.0	3600.00	0	29,099.4	3600.00	0
1	0	0	0	181,258.9	3600.00	0	21,186.1	3600.00	0
0	1	0	0	28,615.5	3600.00	0	28,174.6	3600.00	0
0	1	1	0	21,447.4	1941.54	4	7426.4	913.55	7
1	1	1	0	5837.4	61.37	6	8077.5	948.55	7
1	1	1	1	5863.2	61.20	6	8077.5	974.16	7
Sample (700)									
0	0	0	0	7734.4	3600.00	0	26 465.3	3600.00	0
1	0	0	0	152,656.8	3600.00	0	19,163.8	3600.00	0
0	1	0	0	23,220.9	3600.00	0	21,774.4	3600.00	0
0	1	1	0	30,603.9	1780.14	6	7814.5	1131.49	7
1	1	1	0	6372.9	138.53	5	8800.0	1257.95	7
1	1	1	1	6325.5	140.55	5	8800.0	1274.35	7
Sample (800)									
0	0	0	0	6139.6	3600.00	0	22,311.9	3600.00	0
1	0	0	0	130,506.2	3600.00	0	17,108.1	3600.00	0
0	1	0	0	18,217.5	3600.00	0	17,203.4	3600.00	0
0	1	1	0	26,364.9	2506.24	4	9310.2	1518.02	7
1	1	1	0	7861.5	80.38	6	9628.0	1570.32	7
1	1	1	1	7840.7	79.63	6	9628.0	1546.33	7
Sample (900)									
0	0	0	0	5808.0	3600.00	0	20,471.8	3600.00	0
1	0	0	0	118,504.8	3600.00	0	15,074.9	3600.00	0
0	1	0	0	16,275.8	3600.00	0	13,287.0	3600.00	0
0	1	1	0	25,767.8	3174.30	3	9624.2	1885.05	7
1	1	1	0	5741.3	301.63	4	9732.6	1898.40	7
1	1	1	1	7102.9	177.60	5	9732.6	1862.99	7
Sample (1000)									
0	0	0	0	4839.0	3600.00	0	17,887.6	3600.00	0
1	0	0	0	104,220.5	3600.00	0	14,923.6	3600.00	0
0	1	0	0	12,403.4	3600.00	0	11,910.6	3600.00	0
0	1	1	0	23,446.3	3372.88	2	10,227.0	2127.64	6
1	1	1	0	8852.9	109.65	6	10,752.2	2148.66	6
1	1	1	1	7873.6	182.62	5	10,840.9	2089.13	6

Table 13 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using or not the OA cuts for the epigraph model

Setting	Epigraph model		
	#nodes	time	#solved
Sample (100)			
w/o OA cuts	2171.7	42.65	7
w/ OA cuts	2459.9	50.21	7
Sample (200)			
w/o OA cuts	3094.1	124.43	7
w/ OA cuts	3327.1	177.07	7
Sample (300)			
w/o OA cuts	4387.3	262.19	7
w/ OA cuts	5051.0	385.62	7
Sample (400)			
w/o OA cuts	5430.1	408.90	7
w/ OA cuts	6171.4	624.12	7
Sample (500)			
w/o OA cuts	6998.6	681.69	7
w/ OA cuts	7249.4	1004.41	7
Sample (600)			
w/o OA cuts	8077.5	974.16	7
w/ OA cuts	7792.4	1292.30	7
Sample (700)			
w/o OA cuts	8800.0	1274.35	7
w/ OA cuts	9289.0	1754.04	7
Sample (800)			
w/o OA cuts	9628.0	1546.33	7
w/ OA cuts	10,139.6	2224.56	7
Sample (900)			
w/o OA cuts	9732.6	1862.99	7
w/ OA cuts	10,387.2	2300.25	6
Sample (1000)			
w/o OA cuts	10,840.9	2089.13	6
w/ OA cuts	9838.3	2622.94	4

From this experiment, we also conclude that, in general, the quadratic model performs better regarding running times, whereas the epigraph model performs better regarding the dual bounds, which in turn leads to more instances being solved.

9 Conclusion

Solving the MSSC problem to global optimality is a very challenging task that already has received considerable attention in the literature. Nevertheless, the problem is far from being

Table 14 Comparison of mean number of nodes, mean running time per instance (in seconds), and number of solved instances using different branching rules

Setting	Quadratic model			Epigraph model		
	#nodes	time	#solved	#nodes	time	#solved
Sample (100)						
Standard	1715.9	1.91	7	2459.9	50.21	7
Distance	2220.6	2.50	7	1990.6	46.67	7
Entropy	2133.9	2.38	7	2043.8	41.30	7
Sample (200)						
Standard	3229.1	6.78	7	3327.1	177.07	7
Distance	3715.7	8.33	7	3626.5	218.33	7
Entropy	3652.8	7.94	7	3572.1	164.08	7
Sample (300)						
Standard	4632.5	12.25	7	5051.0	385.62	7
Distance	5134.7	14.29	7	5174.6	493.21	7
Entropy	4445.0	35.91	6	4959.5	354.50	7
Sample (400)						
Standard	5356.3	16.11	7	6171.4	624.12	7
Distance	5601.1	18.33	7	5934.1	792.31	7
Entropy	5169.0	44.84	6	6164.3	571.01	7
Sample (500)						
Standard	5956.7	21.09	7	7249.4	1004.41	7
Distance	5086.9	58.55	6	7653.5	1182.79	7
Entropy	6205.8	59.10	6	7124.9	902.84	7
Sample (600)						
Standard	5863.2	61.93	6	7792.4	1292.30	7
Distance	7754.0	34.30	7	9088.0	1763.90	7
Entropy	6690.4	27.86	7	7922.3	1083.07	7
Sample (700)						
Standard	6325.5	139.71	5	9289.0	1754.04	7
Distance	7288.0	82.41	6	10,056.9	2147.21	7
Entropy	7625.7	34.99	7	9250.8	1540.31	7
Sample (800)						
Standard	7840.7	78.82	6	10,139.6	2224.56	7
Distance	6844.6	163.07	5	10,594.9	2518.07	5
Entropy	7535.0	151.02	5	10,435.5	1969.20	6
Sample (900)						
Standard	7102.9	174.98	5	10 387.2	2300.25	6
Distance	6872.1	166.80	5	8988.8	2390.23	5
Entropy	6307.4	158.35	5	11,012.2	2308.25	5

Table 14 continued

Setting	Quadratic model			Epigraph model		
	#nodes	time	#solved	#nodes	time	#solved
Sample (1000)						
Standard	7873.6	180.79	5	9838.3	2622.94	4
Distance	9924.7	62.60	7	9021.6	2705.16	4
Entropy	8833.3	200.34	5	11,400.1	2663.63	5

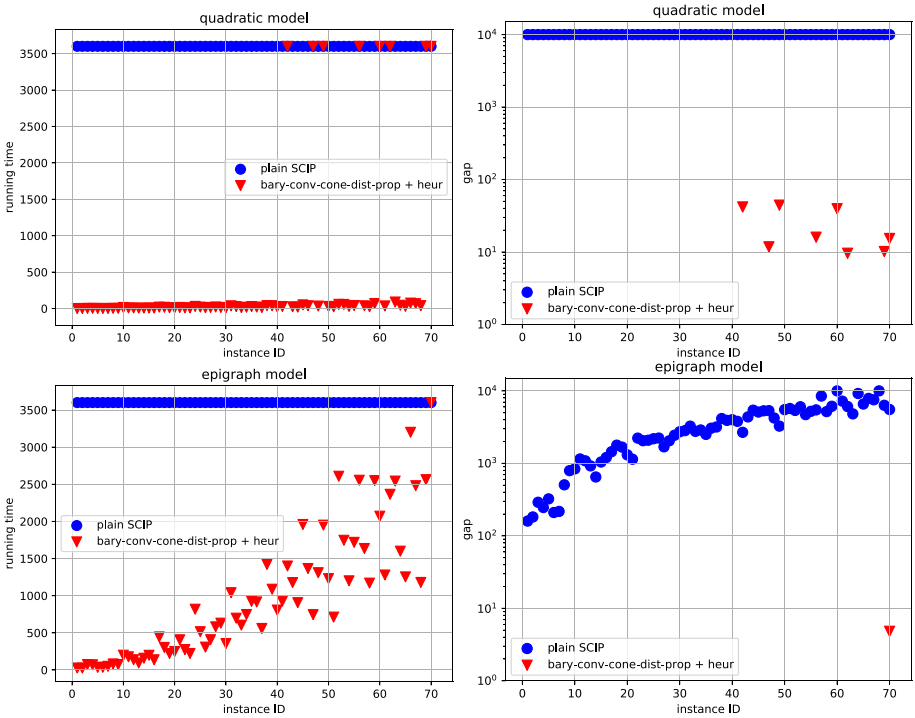

Fig. 12 Running times and gaps comparison between plain SCIP and SCIP enabled with the best setting for the sampled instances and for 2 clusters

"practically solved". In this paper, we propose different techniques (including propagation, cutting planes, branching rules, or primal heuristics) that can be incorporated in a branch-and-bound framework for solving the problem. Our extensive numerical study shows that these novel techniques significantly help to improve the solution process. On the one hand, we can now solve instances that have not been solvable before. On the other hand, the optimality gaps for those instances that remain unsolvable are significantly reduced.

Not surprisingly, there are still some ideas left for future research. Let us sketch two of them. First, we show that our techniques can be used to globally solve instances of moderate size. Thus, our methods could also be used in solution approaches for the MSSC problem that rely on reducing the dimension or the size of the originally given problem; see, e.g., Hua et al. [35]. Second, there further exist variants of the MSSC problem with additional side constraints as discussed in, e.g., Liberti and Manca [38]. Such side constraints allow

for solution techniques that are feasibility-based, whereas all our techniques are optimality-based. Hence, a combination of both could yield an overall branch-and-bound framework that is even more effective for side-constrained MSSC problems.

Acknowledgements The second author thanks the DFG for their support within RTG 2126 "Algorithmic Optimization". Moreover, the last author thanks the DFG for their support within projects A05 and B08 in CRC TRR 154.

Funding Open Access funding enabled and organized by Projekt DEAL.

Appendix A. List of SCIP heuristics disabled in our experiments

We disabled the following SCIP heuristics: alns, mpec, subnlp, rens, locks, objpscostdiving, distributiondiving, clique, nlpdiving, rins, linesearchdiving, conflictdiving, crossover, fracdiving, guideddiving, pscostdiving, randrounding, veclendiving, adaptivediving.

Appendix B. Numerical results per instance

Here we present the detailed numerical results obtained for each instance. The running time (in seconds) is encoded as "time". The gap (in percentage) between the primal and dual bounds is encoded as "gap".

See Tables 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, and 33.

Table 15 Results for 2 clusters using plain SCIP

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	909,913	3600.00	∞	18,956	3600.00	170.24
German22	207,026	900.31	0.00	1343	17.25	0.00
German59	203,438	3600.00	∞	112,482	3600.00	32.98
body-measurements	38,934	3600.00	∞	8862	3600.00	3338.16
cities-coord-202	84,088	3600.00	∞	30,276	3600.00	284.64
cities-coord-666	16,459	3600.00	∞	19,784	3600.00	2886.05
concrete-compressive	12,496	3600.00	∞	8505	3600.00	79149.66
glass-identification	240,834	3600.00	∞	12,176	3600.00	1920.57
image-segmentation	3026	3600.00	∞	2870	3600.00	8405.89
padberg-rinaldi-hole-dri	2654	3600.00	∞	114,594	3600.00	∞
reinelt-hole-drilling	2265	3600.00	∞	235,206	3600.00	∞
ruspini	874,673	3600.00	∞	137,981	3600.00	132.28
telugu-indian-vowel	846	3600.00	∞	13,681	3600.00	5082.69
a1	3024	3600.00	∞	88,111	3600.00	∞
a2	5338	3600.00	∞	67,159	3600.00	∞
a3	6852	3600.00	∞	41,941	3600.00	∞
dim	3175	3600.00	∞	738	3600.00	∞
g2-2-30	13,927	3600.00	∞	10,273	3600.00	22444.38
g2-2-50	11,804	3600.00	∞	10,837	3600.00	13433.08
g2-2-70	2376	3600.00	∞	8235	3600.00	19192.83
s1	5032	3600.00	∞	58,633	3600.00	∞
s2	5050	3600.00	∞	47,121	3600.00	∞
s3	5035	3600.00	∞	55,728	3600.00	∞
s4	5029	3600.00	∞	53,218	3600.00	∞
unbalance	6544	3600.00	∞	37,468	3600.00	∞

Table 16 Results for 2 clusters using enabled heuristics

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	818,860	3600.00	∞	15,462	3600.00	219.06
German22	112,049	327.16	0.00	1141	14.39	0.00
German59	266,506	3600.00	∞	142,167	3600.00	32.20
body-measurements	100,130	3600.00	∞	7959	3600.00	2441.41
cities-coord-202	74,901	3600.00	∞	37,155	3600.00	234.43
cities-coord-666	13,015	3600.00	∞	16,950	3600.00	1980.38
concrete-compressive	26,517	3600.00	∞	4933	3600.00	17012.20
glass-identification	253,985	3600.00	∞	10,690	3600.00	905.82
image-segmentation	2666	3600.00	∞	1601	3600.00	28016.73
padberg-rinaldi-hole-dri	4703	3600.00	∞	101,082	3600.00	∞
reinelt-hole-drilling	2081	3600.00	∞	189,387	3600.00	∞
ruspini	736,505	3600.00	∞	142,065	3600.00	85.06
telugu-indian-vowel	852	3600.00	∞	10,886	3600.00	2774.03
a1	5913	3600.00	∞	76,717	3600.00	∞
a2	5156	3600.00	∞	24,795	3600.00	∞
a3	7083	3600.00	∞	38,235	3600.00	∞
dim	3991	3600.00	∞	1257	3600.00	∞
g2-2-30	13,412	3600.00	∞	9103	3600.00	5636.36
g2-2-50	10,475	3600.00	∞	9832	3600.00	8397.81
g2-2-70	11,617	3600.00	∞	11,084	3600.00	8115.49
s1	4840	3600.00	∞	29,981	3600.00	∞
s2	9902	3600.00	∞	47,364	3600.00	∞
s3	9786	3600.00	∞	44,516	3600.00	∞
s4	4917	3600.00	∞	45,528	3600.00	∞
unbalance	6075	3600.00	∞	28,532	3600.00	∞

Table 17 Results for 2 clusters using enabled heuristics and barycenter propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	1,291,451	3600.00	390.75	36,671	3600.00	133.79
German22	735	1.00	0.00	603	3.14	0.00
German59	298,474	3600.00	60.67	181,024	3302.02	0.00
body-measurements	175,149	3600.00	4020.08	10,707	3600.00	2023.87
cities-coord-202	209,526	3600.00	182.11	53,429	3600.00	135.06
cities-coord-666	6326	3600.00	2030.10	29,584	3600.00	1624.17
concrete-compressive	78,715	3600.00	5124.69	5448	3600.00	18755.18
glass-identification	378,677	3600.00	327.75	13,634	3600.00	449.85
image-segmentation	13,938	3600.00	41771.35	251	3600.00	∞
padberg-rinaldi-hole-dri	215,038	3600.00	5111.74	115,222	3600.00	∞
reinelt-hole-drilling	548,760	3600.00	1821.46	243,529	3600.00	∞
ruspini	371,336	3600.00	172.24	202,409	3600.00	62.62
telugu-indian-vowel	482,986	3600.00	1188.41	11,452	3600.00	2802.01
a1	154,455	3600.00	9892.16	76,186	3600.00	∞
a2	79,802	3600.00	14609.94	28,009	3600.00	∞
a3	50,747	3600.00	28536.14	8957	3600.00	∞
dim	20,279	3600.00	12806.54	938	3600.00	∞
g2-2-30	68,620	3600.00	10686.30	9677	3600.00	8018.12
g2-2-50	69,035	3600.00	8460.72	11,622	3600.00	6694.39
g2-2-70	75,071	3600.00	6784.35	10,224	3600.00	6502.58
s1	78,784	3600.00	29879.42	30,042	3600.00	∞
s2	82,041	3600.00	18242.68	34,579	3600.00	∞
s3	87,351	3600.00	17978.65	32,344	3600.00	∞
s4	87,945	3600.00	17583.74	35,208	3600.00	∞
unbalance	70,452	3600.00	5680.62	22,020	3600.00	∞

Table 18 Results for 2 clusters using enabled heuristics and convexity propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	1,262,448	3600.00	∞	38,295	3600.00	130.60
German22	621	1.47	0.00	523	2.55	0.00
German59	2887	10.46	0.00	1979	17.74	0.00
body-measurements	232,593	3600.00	∞	12,806	3600.00	1701.41
cities-coord-202	150,793	293.82	0.00	37,339	1333.65	0.00
cities-coord-666	179,651	3600.00	12.50	17,055	3600.00	93.41
concrete-compressive	771	3600.00	∞	6482	3600.00	31862.98
glass-identification	210,831	3600.00	∞	12,077	3600.00	885.51
image-segmentation	1946	3600.00	∞	2394	3600.00	28016.73
padberg-rinaldi-hole-dri	111	3600.00	∞	33,732	3600.00	∞
reinelt-hole-drilling	231	3600.00	∞	131,094	3600.00	∞
ruspini	9953	20.64	0.00	1957	36.14	0.00
telugu-indian-vowel	171	3600.00	∞	8986	3600.00	354.14
a1	183	3600.00	∞	40,666	3600.00	∞
a2	377	3600.00	∞	20,874	3600.00	∞
a3	64	3600.00	∞	12,275	3600.00	∞
dim	3278	3600.00	∞	1502	3600.00	∞
g2-2-30	107,172	3600.00	∞	5668	3600.00	29.91
g2-2-50	196,144	3600.00	∞	3547	3600.00	34.49
g2-2-70	226,646	3600.00	∞	2775	3600.00	92.31
s1	84	3600.00	∞	22,269	3600.00	∞
s2	182	3600.00	∞	18,286	3600.00	∞
s3	89	3600.00	∞	23,013	3600.00	∞
s4	98	3600.00	∞	22,590	3600.00	∞
unbalance	98	3600.00	∞	26,895	3600.00	∞

Table 19 Results for 2 clusters using enabled heuristics and convexity+cone propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	1,254,550	3600.00	∞	38,867	3600.00	114.71
German22	539	1.40	0.00	269	1.28	0.00
German59	2139	7.71	0.00	627	7.45	0.00
body-measurements	247,712	3600.00	∞	10,235	3600.00	1460.22
cities-coord-202	49,373	142.42	0.00	4759	155.58	0.00
cities-coord-666	63,323	773.03	0.00	3644	525.23	0.00
concrete-compressive	962	3600.00	∞	6464	3600.00	31862.98
glass-identification	216,426	3600.00	∞	12,824	3600.00	877.58
image-segmentation	2175	3600.00	∞	2193	3600.00	28016.73
padberg-rinaldi-hole-dri	53	3600.00	∞	44,395	3600.00	∞
reinelt-hole-drilling	26	3600.00	∞	161,496	3600.00	∞
ruspini	6875	15.21	0.00	549	10.29	0.00
telugu-indian-vowel	361	3600.00	∞	8627	3600.00	171.65
a1	340	3600.00	∞	40,987	3600.00	∞
a2	22	3600.00	∞	10,912	3600.00	∞
a3	219	3600.00	∞	5419	3600.00	∞
dim	3318	3600.00	∞	1655	3600.00	∞
g2-2-30	172,536	3600.00	∞	6249	3460.79	0.00
g2-2-50	205,102	3600.00	∞	4509	3600.00	11.25
g2-2-70	190,472	3600.00	∞	3714	3600.00	44.85
s1	206	3600.00	∞	15,937	3600.00	∞
s2	58	3600.00	∞	14,702	3600.00	∞
s3	70	3600.00	∞	14,203	3600.00	∞
s4	34	3600.00	∞	16,303	3600.00	∞
unbalance	24	3600.00	∞	20,735	3600.00	∞

Table 20 Results for 2 clusters using enabled heuristics and barycenter+convexity+cone propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	1,333,676	3600.00	180.59	47,822	3600.00	84.02
German22	155	0.21	0.00	179	0.96	0.00
German59	509	0.98	0.00	631	7.57	0.00
body-measurements	316,874	3600.00	3781.81	12,521	3600.00	1281.91
cities-coord-202	3385	5.74	0.00	3954	139.40	0.00
cities-coord-666	2800	14.04	0.00	4116	562.66	0.00
concrete-compressive	91,207	3600.00	5087.22	5158	3600.00	18171.42
glass-identification	542,435	3600.00	326.10	13,759	3600.00	442.45
image-segmentation	17,008	3600.00	39499.81	257	3600.00	∞
padberg-rinaldi-hole-dri	1469	3600.00	23.52	80,303	3600.00	∞
reinelt-hole-drilling	1788	3600.00	19.02	155,237	3600.00	∞
ruspini	487	0.95	0.00	505	9.96	0.00
telugu-indian-vowel	28,388	3600.00	60.12	7274	3600.00	143.63
a1	1718	3600.00	25.16	35,274	3600.00	∞
a2	936	3600.00	30.58	13,296	3600.00	∞
a3	886	3600.00	84.06	6954	3600.00	∞
dim	21,686	3600.00	12465.58	884	3600.00	∞
g2-2-30	6383	114.73	0.00	5959	3600.00	7.10
g2-2-50	14,995	217.03	0.00	5967	3600.00	13.32
g2-2-70	36,329	370.00	0.00	2531	3600.00	19.68
s1	2367	3600.00	244.12	13,093	3600.00	∞
s2	676	3600.00	173.69	15,993	3600.00	∞
s3	3971	3600.00	133.13	11,323	3600.00	∞
s4	4708	3600.00	102.62	23,422	3600.00	∞
unbalance	181	3600.00	∞	15,874	3600.00	∞

Table 21 Results for 2 clusters using enabled heuristics, barycenter+convexity+cone+distance propagators, and OA cuts (only applicable for epigraph model)

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	1,594,034	3600.00	163.35	89,287	3600.00	53.12
German22	155	0.18	0.00	247	1.12	0.00
German59	509	0.83	0.00	571	5.63	0.00
body-measurements	374,487	3600.00	3670.49	14,984	3600.00	1609.83
cities-coord-202	3383	5.40	0.00	4146	111.59	0.00
cities-coord-666	2800	13.41	0.00	5023	648.54	0.00
concrete-compressive	99,409	3600.00	5038.68	5290	3600.00	14466.75
glass-identification	653,118	3600.00	326.46	20,036	3600.00	286.45
image-segmentation	18,400	3600.00	38483.96	1200	3600.00	10007.79
padberg-rinaldi-hole-dri	1469	3600.00	23.52	598	3600.00	882.60
reinelt-hole-drilling	1788	3600.00	19.02	625	3600.00	250.56
ruspini	487	0.92	0.00	573	7.89	0.00
telugu-indian-vowel	28,388	3600.00	60.12	6074	3600.00	132.42
a1	1718	3600.00	25.16	7080	3600.00	754130.50
a2	936	3600.00	30.58	828	3600.00	3488806.65
a3	886	3600.00	84.06	49	3600.00	26935734.14
dim	24,323	3600.00	12218.34	573	3600.00	∞
g2-2-30	6383	87.79	0.00	6989	2924.55	0.00
g2-2-50	14,995	172.40	0.00	7800	3600.00	11.49
g2-2-70	36,329	320.03	0.00	5514	3600.00	19.66
s1	2367	3600.00	244.12	183	3600.00	1124074.89
s2	676	3600.00	173.69	211	3600.00	2062192.77
s3	2815	3600.00	220.62	213	3600.00	12046.53
s4	2657	3600.00	182.74	1512	3600.00	11820.60
unbalance	181	3600.00	∞	461	3600.00	269814.70

Table 22 Results for 2 clusters using enabled heuristics, propagators, cuts, and entropy branching rule

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	1,462,878	3600.00	226.87	68,847	3600.00	169.43
German22	357	0.43	0.00	155	0.69	0.00
German59	765	1.32	0.00	649	4.26	0.00
body-measurements	410,938	3600.00	3632.71	18,072	3600.00	1396.68
cities-coord-202	3541	5.47	0.00	3249	68.49	0.00
cities-coord-666	2604	11.45	0.00	5308	690.09	0.00
concrete-compressive	104,887	3600.00	5502.52	10,227	3600.00	10001.54
glass-identification	650,169	3600.00	567.57	26,316	3600.00	304.25
image-segmentation	22,334	3600.00	51123.06	885	3600.00	9677.16
padberg-rinaldi-hole-dri	1080	3600.00	32.60	748	3600.00	56479.73
reinelt-hole-drilling	1857	3600.00	16.07	1243	3600.00	686.79
ruspini	1039	1.81	0.00	555	5.95	0.00
telugu-indian-vowel	160,309	580.64	0.00	13,930	3600.00	695.29
a1	2863	3600.00	18.17	3496	3600.00	177852.67
a2	936	3600.00	30.58	1677	3600.00	26840744.46
a3	886	3600.00	84.06	1604	3600.00	1784421.45
dim	11,684	3600.00	12621.01	1488	3600.00	∞
g2-2-30	6123	79.84	0.00	6399	2314.59	0.00
g2-2-50	8169	3600.00	6.84	9044	3600.00	15.52
g2-2-70	39,249	455.03	0.00	6310	3600.00	32.64
s1	2367	3600.00	244.12	1690	3600.00	91869.90
s2	676	3600.00	173.69	433	3600.00	163413.35
s3	2815	3600.00	220.62	173	3600.00	71533.78
s4	2657	3600.00	182.74	2898	3600.00	600335.95
unbalance	181	3600.00	∞	459	3600.00	179555.61

Table 23 Results for 2 clusters using enabled heuristics, propagators, cuts, and distance branching rule

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	1,445,510	3600.00	273.88	84,174	3600.00	94.85
German22	237	0.31	0.00	119	0.61	0.00
German59	585	1.16	0.00	511	4.13	0.00
body-measurements	374,844	3600.00	2935.07	23,410	3600.00	1291.14
cities-coord-202	3387	5.31	0.00	2483	98.77	0.00
cities-coord-666	3374	51.94	0.00	3790	933.78	0.00
concrete-compressive	96,489	3600.00	5507.69	5754	3600.00	9840.40
glass-identification	642,905	3600.00	510.17	22,912	3600.00	278.33
image-segmentation	19,217	3600.00	20930.03	1810	3600.00	5250.19
padberg-rinaldi-hole-dri	1569	3600.00	23.35	562	3600.00	2523.55
reinelt-hole-drilling	1739	3600.00	20.79	1020	3600.00	284.00
ruspini	775	1.43	0.00	447	4.82	0.00
telugu-indian-vowel	235,233	900.46	0.00	10,812	3600.00	1517.12
a1	1812	3600.00	26.21	700	3600.00	1048553.26
a2	1072	3600.00	27.26	979	3600.00	1638209.19
a3	886	3600.00	84.06	218	3600.00	4281949.92
dim	25,692	3600.00	12414.26	1036	3600.00	∞
g2-2-30	5273	78.59	0.00	5879	2728.94	0.00
g2-2-50	16,711	213.54	0.00	7746	3600.00	25.14
g2-2-70	60,799	1077.49	0.00	7419	3600.00	25.05
s1	2367	3600.00	244.12	145	3600.00	240688.00
s2	676	3600.00	173.69	430	3600.00	138269.06
s3	2815	3600.00	220.62	299	3600.00	5086.40
s4	2657	3600.00	182.74	1454	3600.00	12635.99
unbalance	181	3600.00	∞	857	3600.00	13305.57

Table 24 Results for 3 clusters using plain SCIP

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	328,941	3600.00	∞	15,619	3600.00	2920.40
German22	571,240	3600.00	∞	10,291	72.95	0.00
German59	123,665	3600.00	∞	88,236	3600.00	545.93
body-measurements	12,390	3600.00	∞	2842	3600.00	25911.96
cities-coord-202	46,192	3600.00	∞	20,439	3600.00	2308.37
cities-coord-666	16,802	3600.00	∞	10,309	3600.00	30744.67
concrete-compressive	1998	3600.00	∞	2794	3600.00	∞
glass-identification	58,342	3600.00	∞	7863	3600.00	2890.58
image-segmentation	1030	3600.00	∞	215	3600.00	∞
padberg-rinaldi-hole-dri	2578	3600.00	∞	6	3600.00	∞
reinelt-hole-drilling	1046	3600.00	∞	220,258	3600.00	∞
ruspini	169,938	3600.00	∞	154,343	3600.00	1144.37
telugu-indian-vowel	864	3600.00	∞	3971	3600.00	127881.44
a1	10,052	3600.00	∞	162	3600.00	∞
a2	10,825	3600.00	∞	1165	3600.00	∞
a3	0	3600.00	∞	7602	3600.00	∞
dim	1295	3600.00	∞	259	3600.00	∞
g2-2-30	1972	3600.00	∞	4387	3600.00	284987.85
g2-2-50	2059	3600.00	∞	4542	3600.00	180381.38
g2-2-70	1204	3600.00	∞	3030	3600.00	∞
s1	5072	3600.00	∞	24,037	3600.00	∞
s2	5038	3600.00	∞	15,072	3600.00	∞
s3	5055	3600.00	∞	2529	3600.00	∞
s4	5056	3600.00	∞	56,226	3600.00	∞
unbalance	6530	3600.00	∞	4841	3600.00	∞

Table 25 Results for 3 clusters using enabled heuristics

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	256,654	3600.00	∞	12,293	3600.00	1564.99
German22	1,054,474	3600.00	∞	11,121	86.71	0.00
German59	270,221	3600.00	∞	115,879	3600.00	307.58
body-measurements	15,441	3600.00	∞	2697	3600.00	∞
cities-coord-202	48,173	3600.00	∞	29,327	3600.00	597.37
cities-coord-666	18,724	3600.00	∞	12,652	3600.00	3497.69
concrete-compressive	4809	3600.00	∞	2059	3600.00	626164.64
glass-identification	82,131	3600.00	∞	8477	3600.00	9802.40
image-segmentation	1097	3600.00	∞	334	3600.00	∞
padberg-rinaldi-hole-dri	7066	3600.00	∞	2883	3600.00	∞
reinelt-hole-drilling	2091	3600.00	∞	279,756	3600.00	∞
ruspini	307,187	3600.00	∞	174,971	3600.00	262.23
telugu-indian-vowel	858	3600.00	∞	5074	3600.00	31376.38
a1	2952	3600.00	∞	708	3600.00	∞
a2	5176	3600.00	∞	942	3600.00	∞
a3	0	3600.00	∞	74	3600.00	∞
dim	1630	3600.00	∞	342	3600.00	∞
g2-2-30	1508	3600.00	∞	3883	3600.00	149048.30
g2-2-50	1490	3600.00	∞	4571	3600.00	429183.10
g2-2-70	1246	3600.00	∞	3982	3600.00	48797.08
s1	4858	3600.00	∞	23,482	3600.00	∞
s2	9740	3600.00	∞	15,041	3600.00	∞
s3	4889	3600.00	∞	17,745	3600.00	∞
s4	4950	3600.00	∞	47,178	3600.00	∞
unbalance	6224	3600.00	∞	12,385	3600.00	∞

Table 26 Results for 3 clusters using enabled heuristics and barycenter propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	280,605	3600.00	1734.55	28,221	3600.00	880.33
German22	15,992	23.58	0.00	6493	21.91	0.00
German59	314,057	3600.00	241.47	268,192	3600.00	251.06
body-measurements	43,177	3600.00	11222.63	8046	3600.00	∞
cities-coord-202	103,462	3600.00	943.44	60,570	3600.00	423.04
cities-coord-666	11,930	3600.00	5752.10	19,756	3600.00	4775.80
concrete-compressive	6277	3600.00	∞	2448	3600.00	∞
glass-identification	118,483	3600.00	1069.56	14,447	3600.00	1443.69
image-segmentation	3574	3600.00	∞	322	3600.00	∞
padberg-rinaldi-hole-dri	78,593	3600.00	∞	450	3600.00	∞
reinelt-hole-drilling	255,437	3600.00	833137.03	443,947	3600.00	∞
ruspini	292,533	3600.00	715.82	211,537	3600.00	237.44
telugu-indian-vowel	230,083	3600.00	143344.25	6129	3600.00	30813.12
a1	59,081	3600.00	∞	142	3600.00	∞
a2	11,561	3600.00	∞	594	3600.00	∞
a3	0	3600.00	∞	74	3600.00	∞
dim	6135	3600.00	∞	323	3600.00	∞
g2-2-30	11,274	3600.00	399488.15	4714	3600.00	788662.43
g2-2-50	13,451	3600.00	206676.83	5048	3600.00	449014.04
g2-2-70	5850	3600.00	∞	4677	3600.00	∞
s1	16,290	3600.00	∞	23,627	3600.00	∞
s2	15,558	3600.00	∞	76,626	3600.00	∞
s3	19,054	3600.00	∞	1999	3600.00	∞
s4	20,110	3600.00	∞	35,733	3600.00	∞
unbalance	1	3600.00	∞	15,560	3600.00	∞

Table 27 Results for 3 clusters using enabled heuristics and convexity propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	442,269	3600.00	∞	36,077	3600.00	497.90
German22	122,884	273.42	0.00	4506	13.19	0.00
German59	694,633	3600.00	∞	371,493	3600.00	58.32
body-measurements	60,146	3600.00	∞	7474	3600.00	14488.38
cities-coord-202	636,337	3600.00	∞	102,010	3600.00	737.75
cities-coord-666	26,103	3600.00	∞	45,587	3600.00	6491.97
concrete-compressive	1	3600.00	∞	1	3600.00	∞
glass-identification	51,612	3600.00	∞	11,479	3600.00	9409.18
image-segmentation	977	3600.00	∞	388	3600.00	∞
padberg-rinaldi-hole-dri	3483	3600.00	∞	2883	3600.00	∞
reinelt-hole-drilling	1372	3600.00	∞	415,922	3600.00	∞
ruspini	753,684	3600.00	∞	323,701	3600.00	248.52
telugu-indian-vowel	410,343	3600.00	∞	8846	3600.00	20873.74
a1	368	3600.00	∞	708	3600.00	∞
a2	2688	3600.00	∞	942	3600.00	∞
a3	0	3600.00	∞	74	3600.00	∞
dim	1343	3600.00	∞	429	3600.00	∞
g2-2-30	8084	3600.00	∞	14,399	3600.00	∞
g2-2-50	12,276	3600.00	∞	5698	3600.00	104726.70
g2-2-70	3745	3600.00	∞	3829	3600.00	96142.33
s1	3447	3600.00	∞	53,673	3600.00	∞
s2	582	3600.00	∞	24,445	3600.00	∞
s3	322	3600.00	∞	11,619	3600.00	∞
s4	448	3600.00	∞	49,473	3600.00	∞
unbalance	4324	3600.00	∞	16,934	3600.00	∞

Table 28 Results for 3 clusters using enabled heuristics and convexity+cone propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	229,614	3600.00	∞	34,610	3600.00	531.20
German22	234,561	770.50	0.00	6162	21.35	0.00
German59	501,525	3600.00	∞	145,215	1285.61	0.00
body-measurements	55,863	3600.00	∞	8049	3600.00	12590.36
cities-coord-202	233,440	3600.00	∞	91,917	3600.00	275.14
cities-coord-666	26,776	3600.00	∞	18,046	3600.00	2763.85
concrete-compressive	1	3600.00	∞	1	3600.00	∞
glass-identification	72,842	3600.00	∞	11,375	3600.00	9515.46
image-segmentation	882	3600.00	∞	384	3600.00	∞
padberg-rinaldi-hole-dri	8466	3600.00	∞	2883	3600.00	∞
reinelt-hole-drilling	1022	3600.00	∞	350,191	3600.00	∞
ruspini	710,135	3600.00	∞	123,603	1317.54	0.00
telugu-indian-vowel	90,336	3600.00	∞	7121	3600.00	37240.30
a1	0	3600.00	∞	708	3600.00	∞
a2	27,022	3600.00	∞	942	3600.00	∞
a3	0	3600.00	∞	74	3600.00	∞
dim	1579	3600.00	∞	395	3600.00	∞
g2-2-30	23,034	3600.00	∞	4736	3600.00	440006.61
g2-2-50	21,069	3600.00	∞	4186	3600.00	7295513.20
g2-2-70	30,670	3600.00	∞	3783	3600.00	∞
s1	10,659	3600.00	∞	49,155	3600.00	∞
s2	1772	3600.00	∞	22,270	3600.00	∞
s3	7235	3600.00	∞	2985	3600.00	∞
s4	4709	3600.00	∞	49,473	3600.00	∞
unbalance	3769	3600.00	∞	16,966	3600.00	∞

Table 29 Results for 3 clusters using enabled heuristics and barycenter+convexity+cone propagator

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	348,209	3600.00	1890.73	29,323	3600.00	767.07
German22	7569	10.77	0.00	4576	18.24	0.00
German59	534,709	1949.13	0.00	143,355	1226.50	0.00
body-measurements	74,701	3600.00	15297.12	8419	3600.00	16458.64
cities-coord-202	218,931	3600.00	727.52	92,471	3600.00	246.47
cities-coord-666	26,602	3600.00	58814.95	18,797	3600.00	3456.63
concrete-compressive	1	3600.00	∞	1	3600.00	∞
glass-identification	145,443	3600.00	1034.68	14,558	3600.00	1439.54
image-segmentation	4374	3600.00	∞	317	3600.00	∞
padberg-rinaldi-hole-dri	150,154	3600.00	∞	450	3600.00	∞
reinelt-hole-drilling	379,375	3600.00	∞	363,311	3600.00	∞
ruspini	819,053	3600.00	871.99	145,815	1793.26	0.00
telugu-indian-vowel	281,326	3600.00	∞	9796	3600.00	637541.95
a1	111,630	3600.00	∞	142	3600.00	∞
a2	30,873	3600.00	∞	594	3600.00	∞
a3	0	3600.00	∞	74	3600.00	∞
dim	7302	3600.00	∞	294	3600.00	∞
g2-2-30	35,125	3600.00	99442.00	3825	3600.00	∞
g2-2-50	34,063	3600.00	69060.41	3766	3600.00	960564.45
g2-2-70	12,876	3600.00	∞	5841	3600.00	293061.36
s1	5624	3600.00	∞	21,594	3600.00	∞
s2	5496	3600.00	∞	65,969	3600.00	∞
s3	1599	3600.00	∞	1365	3600.00	∞
s4	65,193	3600.00	∞	35,733	3600.00	∞
unbalance	5514	3600.00	∞	13,148	3600.00	∞

Table 30 Results for 3 clusters using enabled heuristics, barycenter+convexity+cone propagators, and OA cuts

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	348,209	3600.00	1890.73	37080	3600.00	711.88
German22	7569	10.77	0.00	3247	12.81	0.00
German59	534,709	1949.13	0.00	90712	594.55	0.00
body-measurements	74,701	3600.00	15297.12	11605	3600.00	41450.30
cities-coord-202	218,931	3600.00	727.52	90,433	3600.00	187.54
cities-coord-666	26,602	3600.00	58814.95	21,894	3600.00	2596.30
concrete-compressive	1	3600.00	∞	1	3600.00	∞
glass-identification	145,443	3600.00	1034.68	14,860	3600.00	947.23
image-segmentation	4374	3600.00	∞	304	3600.00	∞
padberg-rinaldi-hole-dri	150,154	3600.00	∞	1027	3600.00	∞
reinelt-hole-drilling	379,375	3600.00	∞	9789	3600.00	174671.81
ruspini	819,053	3600.00	871.99	216,584	2304.35	0.00
telugu-indian-vowel	281,326	3600.00	∞	9217	3600.00	97407.83
a1	111,630	3600.00	∞	558	3600.00	∞
a2	30,873	3600.00	∞	747	3600.00	∞
a3	0	3600.00	∞	533	3600.00	∞
dim	7302	3600.00	∞	544	3600.00	∞
g2-2-30	35,125	3600.00	99442.00	4829	3600.00	1247530.48
g2-2-50	34,063	3600.00	69060.41	5487	3600.00	2292990.33
g2-2-70	12,876	3600.00	∞	3349	3600.00	195477.72
s1	5624	3600.00	∞	1730	3600.00	∞
s2	5496	3600.00	∞	837	3600.00	∞
s3	1599	3600.00	∞	920	3600.00	∞
s4	65,193	3600.00	∞	1004	3600.00	∞
unbalance	5514	3600.00	∞	971	3600.00	∞

Table 31 Results for 3 clusters using enabled heuristics, barycenter+convexity+cone+distance propagators, and OA cuts

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	446,923	3600.00	1849.96	59,165	3600.00	659.99
German22	7569	8.77	0.00	3247	10.52	0.00
German59	534,709	1597.28	0.00	91,402	503.05	0.00
body-measurements	83,144	3600.00	14592.81	15,451	3600.00	37113.59
cities-coord-202	239,312	3600.00	699.49	130,068	3600.00	147.72
cities-coord-666	27,921	3600.00	57591.83	29,912	3600.00	1974.12
concrete-compressive	1	3600.00	∞	1	3600.00	∞
glass-identification	168,283	3600.00	1013.11	19,760	3600.00	817.90
image-segmentation	4830	3600.00	∞	393	3600.00	∞
padberg-rinaldi-hole-dri	174,185	3600.00	∞	3265	3600.00	∞
reinelt-hole-drilling	419,551	3600.00	∞	14,618	3600.00	52928.16
ruspini	906,635	3600.00	796.01	218,941	1775.63	0.00
telugu-indian-vowel	325,351	3600.00	∞	10,984	3600.00	97407.83
a1	143,309	3600.00	∞	699	3600.00	∞
a2	41,088	3600.00	∞	803	3600.00	∞
a3	0	3600.00	∞	1335	3600.00	∞
dim	7947	3600.00	∞	682	3600.00	∞
g2-2-30	35,530	3600.00	98431.46	5812	3600.00	1203432.46
g2-2-50	37,600	3600.00	63371.45	6492	3600.00	683571.57
g2-2-70	13,865	3600.00	∞	4075	3600.00	126844.00
s1	4586	3600.00	∞	1798	3600.00	∞
s2	11,726	3600.00	∞	941	3600.00	∞
s3	1599	3600.00	∞	997	3600.00	∞
s4	52,430	3600.00	∞	1093	3600.00	∞
unbalance	10,689	3600.00	∞	1340	3600.00	15265661.41

Table 32 Results for 3 clusters using enabled heuristics, propagators, cuts, and entropy branching rule

Instance	quadratic model			epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	343,703	3600.00	1477.66	71,751	3600.00	22683.70
German22	45,150	52.35	0.00	5333	12.16	0.00
German59	936,893	3600.00	56.04	336,433	1635.47	0.00
body-measurements	60,997	3600.00	14613.99	14,271	3600.00	16576.14
cities-coord-202	296,551	3600.00	301.34	155,242	3600.00	421.92
cities-coord-666	26,269	3600.00	7757.89	45,297	3600.00	2234.12
concrete-compressive	1	3600.00	∞	1	3600.00	∞
glass-identification	169,038	3600.00	2547.47	19,694	3600.00	4728.44
image-segmentation	4722	3600.00	∞	1068	3600.00	∞
padberg-rinaldi-hole-dri	171,210	3600.00	∞	11,198	3600.00	∞
reinelt-hole-drilling	418,898	3600.00	∞	45,383	3600.00	∞
ruspini	786,957	3600.00	384.48	393,872	3600.00	225.40
telugu-indian-vowel	317,636	3600.00	∞	8674	3600.00	194927.78
a1	138,799	3600.00	∞	1212	3600.00	∞
a2	41,024	3600.00	∞	594	3600.00	∞
a3	0	3600.00	∞	276	3600.00	∞
dim	6776	3600.00	∞	1002	3600.00	∞
g2-2-30	30,811	3600.00	72575.07	8337	3600.00	42157.54
g2-2-50	41,034	3600.00	61675.17	6234	3600.00	62328.59
g2-2-70	9287	3600.00	∞	5348	3600.00	174453.15
s1	4586	3600.00	∞	252	3600.00	∞
s2	11,726	3600.00	∞	2999	3600.00	∞
s3	1599	3600.00	∞	1978	3600.00	∞
s4	52,430	3600.00	∞	3053	3600.00	∞
unbalance	10,689	3600.00	∞	409	3600.00	∞

Table 33 Results for 3 clusters using enabled heuristics, propagators, cuts, and distance branching rule

Instance	Quadratic model			Epigraph model		
	#nodes	time	gap	#nodes	time	gap
Fisher150iris	429,083	3600.00	2347.78	92,344	3600.00	1390.57
German22	11,515	15.42	0.00	2855	6.51	0.00
German59	295,515	1301.84	0.00	148,496	711.93	0.00
body-measurements	80,408	3600.00	29828.45	12,074	3600.00	∞
cities-coord-202	254,608	3600.00	879.26	163,946	3600.00	470.39
cities-coord-666	39,469	3600.00	16406.07	74,088	3600.00	∞
concrete-compressive	1	3600.00	∞	1	3600.00	∞
glass-identification	167,142	3600.00	1986.31	17,442	3600.00	1406.33
image-segmentation	7073	3600.00	∞	3111	3600.00	∞
padberg-rinaldi-hole-dri	175,960	3600.00	∞	10,875	3600.00	106261.12
reinelt-hole-drilling	411,784	3600.00	∞	5711	3600.00	∞
ruspini	919,006	3600.00	578.31	221,022	1548.04	0.00
telugu-indian-vowel	340,173	3600.00	∞	15,303	3600.00	∞
a1	141,729	3600.00	∞	7476	3600.00	∞
a2	41,553	3600.00	∞	526	3600.00	∞
a3	0	3600.00	∞	894	3600.00	∞
dim	9647	3600.00	∞	1015	3600.00	∞
g2-2-30	40,223	3600.00	231297.98	7569	3600.00	∞
g2-2-50	37,147	3600.00	33690.28	5833	3600.00	∞
g2-2-70	33,182	3600.00	∞	5532	3600.00	65019.56
s1	4586	3600.00	∞	4366	3600.00	∞
s2	11,726	3600.00	∞	3905	3600.00	∞
s3	1599	3600.00	∞	7583	3600.00	∞
s4	52,430	3600.00	∞	10,129	3600.00	∞
unbalance	10,398	3600.00	∞	1212	3600.00	∞

References

1. Achterberg, T., Koch, T., Martin, A.: Branching rules revisited. Oper. Res. Lett. **33**(1), 42–54 (2005). https://doi.org/10.1016/j.orl.2004.04.002
2. Aloise, D., Deshpande, A., Hansen, P., Popat, P.: NP-hardness of Euclidean sum-of-squares clustering. Mach. Learn. **75**, 245–248 (2009). https://doi.org/10.1007/s10994-009-5103-0
3. Aloise, D., Hansen, P.: A branch-and-cut SDP-based algorithm for minimum sum-of-squares clustering. Pesquisa Operacional **29**, 503–516 (2009). https://doi.org/10.1590/S0101-74382009000300002
4. Aloise, D., Hansen, P.: Evaluating a branch-and-bound RLT-based algorithm for minimum sum-of-squares clustering. J. Global Optim. **49**, 449–465 (2011). https://doi.org/10.1007/s10898-010-9571-3
5. Aloise, D., Hansen, P., Liberti, L.: An improved column generation algorithm for minimum sum-of-squares clustering. Math. Program. **131**, 195–220 (2012). https://doi.org/10.1007/s10107-010-0349-7
6. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The Quickhull algorithm for convex hulls. ACM Trans. Math. Softw. **22**(4), 469–483 (1996). https://doi.org/10.1145/235815.235821
7. Brusco, M.J.: A Repetitive Branch-and-Bound Procedure for Minimum Within-Cluster Sums of Squares Partitioning. Psychometrika **71**(2), 347–363 (2006). https://doi.org/10.1007/s11336-004-1218-1
8. Burgard, J.P., Costa, C.M., Schmidt, M.: Decomposition methods for Robustified k-means clustering problems: if less conservative does not mean less bad. Ann. Oper. Res. (2022). https://doi.org/10.1007/s10479-022-04818-w

9. Chen, C., Luo, J., Parker, K.: Image segmentation via adaptive Kmean clustering and knowledge-based morphological operations with biomedical applications. IEEE Trans. Image Process. **7**(12), 1673–1683 (1998). https://doi.org/10.1109/83.730379
10. Cuesta-Albertos, J.A., Fraiman, R.: Impartial trimmed k-means for functional data. Comput. Stat. Data Anal. **51**(10), 4864–4877 (2007). https://doi.org/10.1016/j.csda.2006.07.011
11. Dasgupta, S.: The hardness of k-means clustering. Tech. rep. Technical Report CS2008-0916. University of California, Department of Computer Science and Engineering. (2007). http://cseweb.ucsd.edu/~dasgupta/papers/kmeans.pdf
12. Datta, S., Datta, S.: Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics **19**(4), 459–466 (2003). https://doi.org/10.1093/bioinformatics/btg025
13. De Rosa, A., Khajavirad, A.: The ratio-cut polytope and K-means clustering. SIAM J. Optim. **32**(1), 173–203 (2022). https://doi.org/10.1137/20M1348601
14. Deza, M.M., Laurent, M.: Geometry of Cuts and Metrics. Springer, Berlin (1997). https://doi.org/10.1007/978-3-642-04295-9
15. Diehr, G.: Evaluation of a branch and bound algorithm for clustering. SIAM J. Sci. Stat. Comput. **6**(2), 268–284 (1985). https://doi.org/10.1137/0906020
16. Dua, D., Graff, C.: UCI Machine Learning Repository. (2017). http://archive.ics.uci.edu/ml
17. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**(3), 307–339 (1986). https://doi.org/10.1007/BF02592064
18. du Merle, O., Hansen, P., Jaumard, B., Mladenovic, N.: An interior point algorithm for minimum sum-of-squares clustering. SIAM J. Sci. Comput. **21**(4), 1485–1505 (1999). https://doi.org/10.1137/S1064827597328327
19. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugen. **7**(2), 179–188 (1936). https://doi.org/10.1111/j.1469-1809.1936.tb02137.x
20. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. Math. Program. **66**(1), 327–349 (1994). https://doi.org/10.1007/BF01581153
21. Floudas, C., Aggarwal, A., Ciric, A.: Global optimum search for nonconvex NLP and MINLP problems. Comput. Chem. Eng. **13**(10), 1117–1132 (1989). https://doi.org/10.1016/0098-1354(89)87016-4
22. Fränti, P., Sieranoja, S.: k-means properties on six clustering benchmark datasets. Appl. Intell. **48**(12), 4743–4759 (2018). https://doi.org/10.1007/s10489-018-1238-7
23. Fränti, P., Sieranoja, S.: How much can k-means be improved by using better initialization and repeats? Pattern Recogn. **93**, 95–112 (2019). https://doi.org/10.1016/j.patcog.2019.04.014
24. Fukuda, K.: cdd/cdd+ Reference Manual. In: Institute for Operations Research, ETH-Zentrum, pp. 91–111 (1997)
25. Fukunaga, K., Narendra, P., Koontz, W.: A branch and bound clustering algorithm. IEEE Trans. Comput. **24**(09), 908–915 (1975). https://doi.org/10.1109/T-C.1975.224336
26. Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W.-K., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., Hendel, G., Hojny, C., Koch, T., Le Bodic, P., Maher, S.J., Matter, F., Miltenberger, M., Mühmer, E., Müller, B., Pfetsch, M.E., Schlösser, F., Serrano, F., Shinano, Y., Tawfik, C., Vigerske, S., Wegscheider, F., Weninger, D., Witzig, J.: The SCIP Optimization Suite 7.0. eng. Tech. rep. 20-10. Takustr. **7**, 14195 Berlin: ZIB (2020)
27. Gilpin, A., Sandholm, T.: Information-theoretic approaches to branching in search. Discrete Optim. **8**(2), 147–159 (2011). https://doi.org/10.1016/j.disopt.2010.07.001
28. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theor. Comput. Sci. **38**, 293–306 (1985). https://doi.org/10.1016/0304-3975(85)90224-5
29. Grötschel, M.H.: Solution of large-scale symmetric travelling salesman problems. Math. Program. **51**, 141–202 (1991). https://doi.org/10.1007/BF01586932
30. Guns, T., Dao, T.-B.-H., Vrain, C., Duong, K.-C.: Repetitive branch-andbound using constraint programming for constrained minimum sum-of-squares clustering. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence (ECAI'16). IOS Press, NLD, pp. 462–470 (2016). https://doi.org/10.3233/978-1-61499-672-9-462
31. Han, S.: Spatial stratification and socio-spatial inequalities: the case of Seoul and Busan in South Korea. Human. Soc. Sci. Commun. **9**(1), 23 (2022). https://doi.org/10.1057/s41599-022-01035-5
32. He, H., Chen, J., Jin, H., Chen, S.-H.: Trading strategies based on K-means clustering and regression models. In: Chen, S.-H., Wang, P.P., Kuo, T.-W. (eds.), Computational Intelligence in Economics and Finance: Volume II, pp. 123–134. Springer, Berlin (2007). https://doi.org/10.1007/978-3-540-72821-4_7
33. Heinz, G., Peterson, L.J., Johnson, R.W., Kerk, C.J.: Exploring relationships in body dimensions. J. Stat. Educ. (2003). https://doi.org/10.1080/10691898.2003.11910711

34. Horst, R., Tuy, H.: Global Optimization. Springer, Berlin (1996). https://doi.org/10.1007/978-3-662-03199-5
35. Hua, K., Shi, M., Cao, Y.: A Scalable deterministic global optimization algorithm for clustering problems. In: International Conference on Machine Learning. PMLR, pp. 4391–4401 (2021). https://proceedings.mlr.press/v139/hua21a.html
36. Kaibel, V., Peinhardt, M., Pfetsch, M.E.: Orbitopal fixing. Discret. Optim. **8**(4), 595–610 (2011). https://doi.org/10.1016/j.disopt.2011.07.001
37. Kaibel, V., Pfetsch, M.E.: Packing and partitioning orbitopes. Math. Program. **114**(1), 1–36 (2008). https://doi.org/10.1007/s10107-006-0081-5
38. Liberti, L., Manca, B.: Side-constrained minimum sum-of-squares clustering: mathematical programming and random projections. J. Global Optim. (2021). https://doi.org/10.1007/s10898-021-01047-6
39. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982). https://doi.org/10.1109/TIT.1982.1056489
40. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pp. 281–297. University of California Press, Berkeley (1967). https://projecteuclid.org/euclid.bsmsp/1200512992
41. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar k-means problem is NP-hard. In: Theoretical Computer Science 442. Special Issue on the Workshop on Algorithms and Computation (WALCOM 2009), pp. 13–21 (2012). https://doi.org/10.1016/j.tcs.2010.05.034
42. Padberg, M., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev. **33**(1), 60–100 (1991). https://doi.org/10.1137/1033004
43. Pal, S.K., Majumder, D.D.: Fuzzy sets and decision making approaches in vowel and speaker recognition. IEEE Trans. Syst. Man Cybern. **7**(8), 625–629 (1977). https://doi.org/10.1109/TSMC.1977.4309789
44. Peng, J., Wei, Y.: Approximating k-means-type clustering via semidefinite programming. SIAM J. Optim. **18**(1), 186–205 (2007). https://doi.org/10.1137/050641983
45. Peng, J., Xia, Y.: A cutting algorithm for the minimum sum-of-squared error clustering. In: Proceedings of the 2005 SIAM International Conference on Data Mining, pp. 150–160 (2005). https://doi.org/10.1137/1.9781611972757.14
46. Peng, J., Xia, Y.: A new theoretical framework for k-means-type clustering. In: Foundations and advances in data mining. Springer, Berlin, pp. 79–96 (2005). https://doi.org/10.1007/11362197_4
47. Piccialli, V., Sudoso, A.M., Wiegele, A.: SOS-SDP: an exact solver for minimum sum-of-squares clustering. INFORMS J. Comput. **34**(4), 2144–2162 (2022). https://doi.org/10.1287/ijoc.2022.1166
48. Plastria, F.: Formulating logical implications in combinatorial optimisation. Eur. J. Oper. Res. **140**(2), 338–353 (2002). https://doi.org/10.1016/S0377-2217(02)00073-5
49. Prasad, M.N., Hanasusanto, G.A.: Improved conic reformulations for *k*-means clustering. SIAM J. Optim. **28**(4), 3105–3126 (2018). https://doi.org/10.1137/17M1135724
50. Quesada, I., Grossmann, I.E.: An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Comput. Chem. Eng. **16**(10–11), 937–947 (1992). https://doi.org/10.1016/0098-1354(92)80028-8
51. Reinelt, G.: TSPLIB-A traveling salesman problem library. ORSA J. Comput. **3**(4), 376–384 (1991). https://doi.org/10.1287/ijoc.3.4.376
52. Ruspini, E.H.: Numerical methods for fuzzy clustering. Inf. Sci. **2**(3), 319–350 (1970). https://doi.org/10.1016/S0020-0255(70)80056-1
53. Sangalli, L.M., Secchi, P., Vantini, S., Vitelli, V.: k-mean alignment for curve clustering. Comput. Stat. Data Anal. **54**(5), 1219–1233 (2010). https://doi.org/10.1016/j.csda.2009.12.008
54. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
55. Sherali, H.D., Desai, J.: A global optimization RLT-based approach for solving the hard clustering problem. J. Global Optim. **32**, 281–306 (2005). https://doi.org/10.1007/s10898-004-2706-7
56. Sobol', I.: On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput. Math. Math. Phys. **7**(4), 86–112 (1967). https://doi.org/10.1016/0041-5553(67)90144-9
57. Späth, H.: Cluster Analysis Algorithms for Data Reduction and Classification of Objects. Horwood, Bristol (1980)
58. Steinley, D.: K-means clustering: a half-century synthesis. Br. J. Math. Stat. Psychol. **59**(1), 1–34 (2006). https://doi.org/10.1348/000711005X48266
59. Tan, M.P., Broach, J.R., Floudas, C.A.: A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning. J. Global Optim. **39**, 323–346 (2007). https://doi.org/10.1007/s10898-007-9140-6
60. Tîrnăucă, C., Gómez-Pérez, D., Balcázar, J.L., Montaña, J.L.: Global optimality in k-means clustering. Inf. Sci. **439–440**, 79–94 (2018). https://doi.org/10.1016/j.ins.2018.02.001

61. Zheng, A., Jiang, B., Li, Y., Zhang, X., Ding, C.: Elastic K-means using posterior probability. PLOS ONE **12**(12), e0188252 (2017). https://doi.org/10.1371/journal.pone.0188252