Dendrograms, Minimum Spanning Trees and Feature Selection

Martine Labbé *[a,b], Mercedes Landete[c,d], Marina Leal[c,d]

[a]Computer Science Department, Université Libre de Bruxelles, Belgium.
[b]INOCS Team, INRIA, Lille, France.
[c]Centro de Investigación Operativa, Universidad Miguel Hernández,
03202 Elche, Alicante, Spain.
[d]Departamento de Estadística, Matemáticas e Informática, Universidad Miguel Hernández,
03202 Elche, Alicante, Spain.

July 20, 2022

Abstract

Feature selection is a fundamental process to avoid overfitting and to reduce the size of databases without significant loss of information that applies to hierarchical clustering. Dendrograms are graphical representations of hierarchical clustering algorithms that for single linkage clustering can be interpreted as minimum spanning trees in the complete network defined by the database. In this work, we introduce the problem that determines jointly a set of features and a dendrogram, according to the single linkage method. We propose different formulations that include the minimum spanning tree problem constraints as well as the feature selection constraints. Different bounds on the objective function are studied. For one of the models, several families of valid inequalities are proposed and the problem of separating them is studied. For another formulation, a decomposition algorithm is designed. In an extensive computational study, the effectiveness of the different models is discussed, the model with valid inequalities is compared with the decomposition algorithm. The computational results also illustrate that the integration of feature selection to the optimization model allows to keep a satisfactory percentage of information.

Keywords: Combinatorial optimization, Feature selection, Hierarchical clustering, Single linkage, Minimum spanning tree.

1 Introduction

Advances in technology have led to the existence of a huge number of databases containing an extensive amount of information. These large databases collect information on a multitude of features for a multitude of objects in different fields such as medicine, chemistry, astronomy,

*Corresponding Author, TO DETERMINE @ to determine.

or social sciences. A way to extract the information is to group the objects into clusters in order to analyze only some representatives of each cluster in detail. However, often in this multidimensional analysis, not all features are equally informative, some features are redundant and others simply do not provide information for classification. Many times more characteristics are used than necessary, even all of them, since a priori it is not known which ones are relevant. Hence, identifying the effective and relevant features may be necessary or useful. Feature selection is a common issue in optimization nowadays to approach, a part from overfitting, the tractability of big and stream data. When it comes to supervised clustering, feature selection is easier because multiple available tools that measure the relationship between clusters and features can be used. However, when the goal is unsupervised clustering, the choice is more complex. In this paper, feature selection in the area of unsupervised clustering known as hierarchical clustering is analyzed. Hierarchical clustering is used when the number of groups is unknown and when certain relationships between objects are sought. The choice of features in hierarchical clustering has to preserve the information not only for a given number of groups but for the entire tree. In the hierarchical clustering literature, there is an evident interest in feature selection. In Witten and Tibshirani [2010], the authors propose a framework for clustering which serves for hierarchical clustering, in which the features are selected making use of a lasso-type penalty function. In Questier et al. [2002], a feature selection approach for hierarchical clustering based on genetic algorithms is presented: a fitness function that tries to minimize the difference between the dissimilarity matrix of the original feature set and the one of the reduced feature sets is used. In Chavent et al. [2021] the authors combine hierarchical clustering of variables and feature selection using random forests.

The result of a hierarchical clustering is a tree that represents the connections of objects at different levels. The graphical representation of this tree is called a dendrogram. In a dendrogram, individuals are connected depending on their similarity; the more similar two individuals are, the sooner they are connected in the dendrogram (or the closer is their connection). Connections can be made according to different criteria or methods: connecting an individual or a group of individuals with nearest neighbor(s) (single linkage clustering), or with the furthest neighbor(s) (complete linkage clustering), or with the nearest(s) in average (simple average clustering), etcetera (see, for instance, Hansen and Jaumard [1997] and Nielsen [2016] and the references therein). Once obtained the dendrogram, clusters are determined by making cuts in the tree: individuals linked below the cut level remain in the same cluster. The comparisons of the results from various hierarchical clustering will compare the clustering for a sample of cutoff levels of the tree. Dendrograms are useful tools for representing an enormous amount of information in a visual way. In addition to serving to represent a hierarchical clustering, a dendrogram can be used to represent clustering of genes or relationships among various biological taxa, in the latter case it is also called a phylogenetic tree. Wherever a list of features or variables for a list of elements or samples is recorded, dendrograms allow visual classification. Dendrograms can appear in different shapes, it can be horizontal or vertical, linear or circular. Dendrograms are intermediate tools for more complex analysis as non hierarchical

clustering or heat-map creation. Different dendrogams for the same sample give complementary information. Dendrograms can be compared with expected clustering or pairwise compared by a tanglegram plot in which one faces the other and their labels are connected.

All the information required for the dendrogram of a set of points, when the single linkage clustering method is considered, is contained in the minimum spanning tree of the network (Gower and Ross [1969]). Thus, this particular dendrograms can be obtained by solving the Minimum Spanning Tree problem (MST) over the complete graph defined by the object distance matrix. Dendrograms are current graphics that are used in very diverse situations such as psychological data analysis Wang et al. [2020], medical data analysis Ghosal et al. [2020], economical data analysis O. Yim [2015] or sports Kahvecioğlu and Morton [2022], among others.

In this work we propose to obtain the dendrogram associated with the distance matrix determined by a set of objects by obtaining the MST for the graph defined by the same distance matrix. In this way, the problem of single linkage hierarchical clustering, which is usually approached from the statistical or machine learning point of view of lasso regression, can be formulated as an optimization problem. Assuming that the number of features to be kept in the analysis is limited and known, for example p, the problem we pose is to find the best dendrogram/MST with p features, where *best* indicates that the length of the spanning tree is as short as possible. The optimization problem consists of selecting the p features that lead to the smallest minimum spanning tree; thus, we are introducing feature selection in MST. From here on we will refer to this problem as the Feature Selection Minimum Spanning Tree problem (FSMST).

The research gap covered by the content of this paper is the mathematical optimization of the single linkage hierarchical clustering with feature selection. Other clustering problems have been analyzed from the mathematical optimization point of view, for instance in Benati et al. [2018] the problem of selecting features from the complete data set and cluster centers from a tentative set is modeled and solved. However, this is the first time for hierarchical clustering. And last but not least, it is the first time that feature selection is taken into consideration for the MST. The applicability of feature selection in MST is a secondary closed gap. The main contributions of this work can be itemized as follows:

i. A mixed-integer optimization model is proposed for solving single linkage clustering with feature selection. It is the first time in the literature that an optimization model is proposed for this problem.

ii. Tight lower bounds for the optimal value of the problem are introduced.

iii. Four different mixed-integer optimization (MIO) formulations are proposed.

iv. Valid inequalities are proposed to strengthened our models. Part of these valid inequalities are based on the lower bounds of the optimal value and part of them are based on the structure of minimum spanning trees.

v. A decomposition approach based on one of the models is designed.

vi. Extensive computational experiments are conducted. In the analysis of the results, we compare the models and evaluate the performance of feature selection in hierarchical clustering.

The remainder of this paper is organized as follows. In Section 2, the notation is introduced and the problem is defined. It is proved that the problem under study is NP-hard and also that different lower bounds for the optimal value of the model can be obtained depending on the selected features. In Section 3, a first mixed integer linear formulation is introduced. Section 4 is devoted to the study of valid inequalities for the formulation proposed in Section 3. Three alternative models are developed in Section 5. One of them is the main clue for a novel decomposition approach. Computational results are thoroughly reported in Section 6. Finally, we provide some discussion and conclusions in Section 7.

2 Notation, problem definition, complexity, lower bounds and general results

Let K be the set of m features observed for the n individuals of a sample. Let p be the number of features we aim to select from K. Let $G = (V, E)$ be the complete undirected graph whose n vertices represent the individuals. For every edge $e = (i, j) \in E$ and feature $k \in K$, let c_e^k be the distance of edge $e \in E$ according to feature $k \in K$, that is, the distance between individuals i and j for feature k. Let $\mathcal{T}$ be the set of all possible spanning trees. The length of spanning tree is equal to the sum of the distances of its edges.

We define the Feature Selection Minimum Spanning Tree problem (FSMST) as follows:

$$\min\{\sum_{k\in S}\sum_{e\in T} d_e^k : T \in \mathcal{T}, S \subseteq K, |S| = p\}$$

Problem FSMST consists thus in determining a spanning tree in graph G and selecting p features in such a way that the sum of the length of the tree for these p features is minimum.

The following example illustrates the problem.

Example 1. *In Figure 1, we present an instance of Problem FSMST with four nodes and three features. The three graphs show the costs of the connections for features 1, 2 and 3, respectively. If p is set to 2, the optimal solution consists in selecting features 1 and 2 and the tree containing edges (1,3), (1,4), (2,3), leading to an optimal value of 16.*

The following theorem states that the FSMST is NP-hard. The proof of the theorem proceeds by a reduction of the q-variable selection problem defined in Benati et al. [2018] and that they show to be NP-hard.

Theorem 1. *Problem FSMST is NP-hard.*

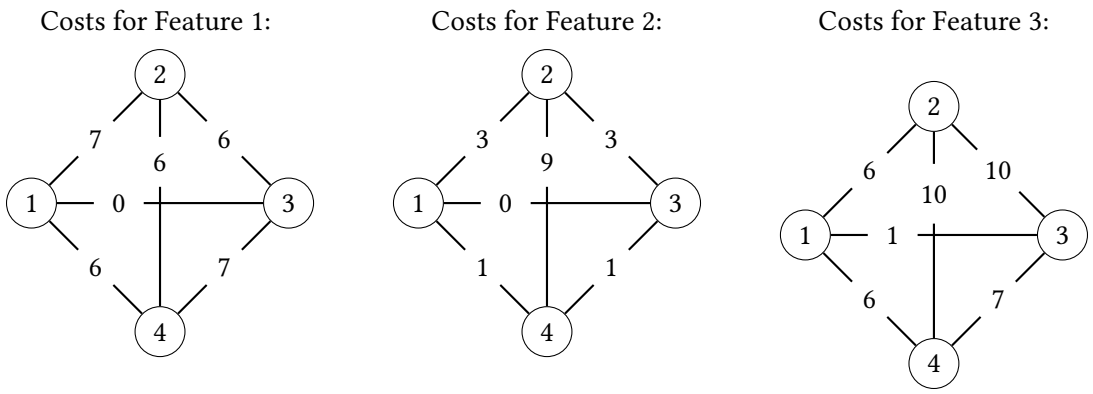

Figure 1: Small instance with $|V| = 4$ and $m = 3$.

Proof. An instance of the recognition version of the q-variable selection problem defined in Benati et al. [2018] is given by a set of objects I, a set of centers J to which the objects must be assigned and a set of features K. Further, cost $c^k_{i,j}$ represents the cost associated to feature k for assigning object i to center j. Finally, the q-variable selection problem asks whether there exists a selection of q features and an assignment of the objects to centers such that the sum of its costs for all selected features is less than or equal to a given value B. Given an instance of the q-variable selection problem, one can construct an instance of FSMST as follows. The set of nodes (individuals) is $I \cup J \cup \{s\}$, where s is a dummy node. For each feature k the edges cost d^k_{ij} are defined as:

- $d^k_{sj} = 0$, for all $j \in J$,
- $d^k_{ij} = c^k_{ij}$, for all $i \in I$ and $j \in J$.

All other edges costs are equal to an arbitrary large value M for all features. These edge costs are chosen large enough to ensure that none of these other edges is ever included in a minimum spanning tree whatever the subset of features considered.

In the same vein, the cost of edges linking the centers j to s are equal to zero for all features, all edges $\{s, j\}$, for all j belonging to any MST, for any feature selection. In consequence there exists a solution to the q-variable selection problem with value lower than or equal to a given threshold B iff FSMST has a solution (ST) with value lower than or equal to B. □

Next, the following five remarks establish lower bounds on the objective value of FSMST. In short, two general bounds LB_1 and LB_2 and two families of lower bounds for the cases in which a specific feature k is selected, LB^k_1 and LB^k_2.

Let us denote by $\mathcal{S}^p(a)$ the set of the p smallest entries of an m-dimensional vector a and by $\mathcal{VS}^p(a)$ the value of that sum: $\mathcal{VS}^p(a) = \min_y\{\sum_{k=1}^m a^k y^k : \sum_{k=1}^m y^k = p, 0 \leq y^k \leq 1, i = 1, \ldots, m\}$. Moreover, let us denote by $\mathcal{MST}(b)$ and $\mathcal{VMST}(b)$ a minimum spanning tree and its value in G with edge cost vector $b = (b_e)_{e \in E}$. Finally, c_e represents the cost vector of an edge with one coordinate for each feature $k = 1, \ldots, m$ and c^k represents the cost vector of a feature with one coordinate for each edge $e \in E$.

Remark 1. *The value* $LB_1 = \mathcal{VS}^p\big((\mathcal{VMST}(c^1), \ldots, \mathcal{VMST}(c^m)\big)$ *is a lower bound on the optimal value of Problem FSMST.*

Remark 2. *Let us sort by non decreasing order the lengths of the MSTs corresponding to each feature considered individually. From now on,* $L_1 \leq L_2 \leq \ldots \leq L_m$*. If feature* k *is chosen and* $\mathcal{VMST}(c^k) > L_p$*, then* LB_1 *can be improved and the lower bound is* $LB_1^k = LB_1 - L_p + \mathcal{VMST}(c^k)$*.*

Remark 3. *The length of a MST with costs* $C_e = \mathcal{VS}^p(c_e)$ *is an lower bound on the optimal value of FSMST,* LB_2*. Explicitly,* $LB_2 = \mathcal{VMST}(C)$*, where* $C = (C_e)_{e \in E}$*.*

Remark 4. *For each edge* $e \in E$*, let us sort all the cost* c_e^k *by non decreasing order:* $C_e^1 \leq C_e^2, \ldots, C_e^n$*. If feature* k *is chosen, the bound* C_e *on the contribution to the total cost of an edge* e *can be improved:*

$$C_e^* = \begin{cases} C_e - C_e^p + c_e^k, & \text{if } c_e^k > C_e^p \\ C_e, & \text{otherwise} \end{cases}$$

and the length of the MST with costs C_e^* *is a new lower bound* LB_2^k *on the optimal value of FSMST.*

Remark 5. *If feature* k *is selected, the lower bound for the problem is* $LB^k = \max\{LB_1^k, LB_2^k\}$*.*

The last result in this section states that given two spanning trees of a graph, it is always possible to compute the length of one from the other. The lemma also states the exact formula to apply.

Theorem 2. *Let* $G = (V, E)$ *be a graph with edge costs* d_e *for all* $e \in E$ *and, for any subgraph* S *of* G*, denote by* $d(S)$ *the sum of the costs of the edges belonging to* S*. Consider any two spanning trees* T *and* T' *of* G*.*

i. There exists a one-one mapping $\sigma : T' \setminus T \to T \setminus T'$ *such that for each edge* $e \in T' \setminus T, \sigma(e) \in T \setminus T'$ *and belongs to the unique path in* T *linking the end vertices of edge* e*,*

ii. $d(T') = d(T) + \sum_{e \in T' \setminus T}(d_e - d_{\sigma(e)})$*.*

Proof. First, given that all spanning trees contain $|V|-1$ edges, $|T' \setminus T| = |T \setminus T'|$. Next, let $e \in T' \setminus T$. Given that T is a spanning tree, removing e from T determines a partition of the vertices into two subsets and e is the unique edge of T belonging to the so obtained edge cut. Further, there exists a unique path, denoted by $P_T(e)$ in T that connects the end vertices of edge e. At least one edge, say $\sigma(e)$, of this path belongs to the cut (otherwise T is not connected) and thus cannot belong to T'. We have,

$$d(T) = d(T \setminus \{e\} \cup \{\sigma(e)\}) + (d_e - d_{\sigma(e)}).$$

In addition, $T \setminus \{e\} \cup \{\sigma(e)\}$ constitutes a tree with one more edge in common with T'. We can repeat this swapping of edges until all edges of $T' \setminus T$ have been considered, yielding the desired equation. □

In the following, let $P_T(e)$ be the path in tree T that connects the end vertices of edge e.

3 First model

In order to formulate FSMST as a MIO problem, we consider the following two sets of binary decision variables: For each feature $k \in K$, variable y^k takes value one only when feature k is selected. For each edge $e = (i, j) \in E$, variable x_e is one if edge e is in the tree. Finally, let $T(x)$ be the tree determined by a vector x.

$$\min_{y,d,x} \quad \sum_{e \in E} \sum_{k=1}^{m} c_e^k y^k x_e \tag{1}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} y^k = p, \tag{2}$$

$$y^k \in \{0, 1\}, \quad k = 1, \ldots, m, \tag{3}$$

$$T(x) \in \mathcal{T}, \tag{4}$$

$$x_e \in \{0, 1\}, e \in E. \tag{5}$$

The goal is to minimize the length of a MST in G endowed with edge weights defined as the sum of distances corresponding to p features. Constraints (2) and (3) ensure that we select p features. For each $e \in E$, we assume that its contribution to the objective function is the sum of the distances corresponding to the selected features. Constraint (4) states that x defines a spanning tree in G. This constraint (4) can be replaced by at least five different sets of constraints, Labbé et al. [2019].

In order to linearize the objective which contains the products of variables, a new family of variables z_e^k $(= y^k x_e)$ is introduced and the following new set of constraints is added:

$$z_e^k \geq y^k + x_e - 1, \quad e \in E,\ k = 1, \ldots, m, \tag{6}$$

$$z_e^k \leq y^k, \quad e \in E,\ k = 1, \ldots, m, \tag{7}$$

$$z_e^k \leq x_e, \quad e \in E,\ k = 1, \ldots, m, \tag{8}$$

$$z_e^k \geq 0, \quad e \in E,\ k = 1, \ldots, m. \tag{9}$$

Constraints (6)-(8) describe the Boolean quadratic polytope, see Letchford and Sørensen [2014] and Padberg [1989]. In this case, the following equalities can be added to the description of the Boolean polytope or alternatively replace some of them.

Proposition 1. *The set of feasible points of the system* (2)-(9) *is unchanged if* (6) *and* (7) *are replaced by*

$$\sum_{k=1}^{m} z_e^k = p x_e, \quad e \in E, \tag{10}$$

$$\sum_{e \in E} z_e^k = (|V|-1) y^k, \quad k = 1, \ldots, m. \tag{11}$$

Proof. Given a feasible solution (x, y, z), let E_0 be the subset of E with $x_e = 1$ and let K_0 be the subset of p features with $y^k = 1$. From (7) and (8) it holds that $z_e^k = 0$ for all $e \notin E_0$ and for all

$k \notin K_0$. If $e \in E_0$, then equality (10) becomes

$$\sum_{k \in K_0} z_e^k = p$$

and provided that $z_e^k \leq 1$ for all $k \in K_0$ it only holds if $z_e^k = 1$ for all $k \in K_0$. Analogously, for a given $k \in K_0$ equality (11) becomes

$$\sum_{e \in E_0} z_e^k = |V| - 1$$

and it only holds if $z_e^k = 1$ for all $e \in E_0$. □

If we consider for $T(X)$ the formulation proposed in Martin [1991], the complete mixed-integer linear optimization (MILO) model for the FSMST reads:

$$\begin{aligned}
(\mathrm{M}_1) \quad & \min_{x,z} \sum_{e \in E} \sum_{k=1}^{m} c_e^k z_e^k \\
& \text{s.t.} \quad (2), (3), (5) - (9) \\
& \sum_{e \in E} x_e = n - 1, && (12) \\
& w_{rij} + w_{rji} = x_e, \quad e = (i,j) \in E,\ r = 1, ..., n,\ i, j \neq r && (13) \\
& \sum_{(i',j) \in E:\ \substack{(i'=r \wedge j=i) \vee \\ (i'=i \wedge j=r)}} x_{i'j} + \sum_{j=1,\ldots,n: j \neq r} w_{rij} \leq 1, \quad i, r = 1, \ldots, n,\ i \neq r, && (14) \\
& w_{rij} \geq 0, \quad i, j, r = 1, \ldots, n, && (15)
\end{aligned}$$

Given its polynomial number of variables and constraints, it can be solved with off-the-shelf integer solvers.

The following example illustrates the interest in considering constraints (10) and (11).

Example 1. *(cont.) If $p = 1$ in the instance considered in Example 1, the optimal value for the linear relaxation of M_1 is zero and corresponds with the solution $y^1 = y^2 = 0.5$, $x_{(1,3)} = 1$, $x_{(1,4)} = x_{(2,3)} = x_{(2,4)} = x_{(3,4)} = 0.5$. However, if constraints (10)-(11) replace constraints (6)-(7), the optimal value for the linear relaxation is 16 (the integer optimal value) for $y^1 = y^2 = 1$, and $x_{(1,3)} = x_{(1,4)} = x_{(2,3)} = 1$.*

4 Strengthening Formulation M_1

Let v^* denote the optimal value of Formulation M_1. In this section several families of inequalities defining a lower bound on v^* are proposed. Four of these families state that v^* is larger then or equal to certain linear combination of the bounds presented in Section 2. Other seven families of valid inequalities use, as coefficients, different values of $\mathcal{S}^p$, $\mathcal{VS}^p$, $\mathcal{MST}$ or $\mathcal{VMST}$.

4.1 Valid inequalities based on lower bounds

Proposition 2. *Let LB be any lower bound for the problem (LB can be LB_1 or LB_2 or any other). Let $S \subseteq S^{>} = \{k \in K : LB^k > LB\}$. Without loss of generality, let assume that $LB < LB^{s_1} \leq \ldots \leq LB^{s_{|S|}}$, Let $LB^{s_0} = LB$. Then, the following inequality is valid.*

$$v^* \geq LB + \sum_{j=1}^{|S|} (LB^{s_j} - LB^{s_{j-1}}) y^{s_j}. \tag{16}$$

Proof. Given a solution (x, y, z, w) of M_1, let $S^* = \{s \in S : y^s = 1\}$. If $S^* = \emptyset$, then inequality (16) trivially holds. Otherwise, let $S^* = \{s_1^*, \ldots, s_{|S^*|}^*\}$ and $LB_{s_0^*} = LB$. The right hand side of inequality (16) is

$$LB + \sum_{j=1}^{|S|} (LB^{s_j} - LB^{s_{j-1}}) y^{s_j} \leq LB + \sum_{j=1}^{|S^*|} (LB^{s_j^*} - LB^{s_{j-1}^*})$$

since for each $s_t^* \in S^*$ if $s_t^* = s_q$, then $LB_{s_{t-1}^*} \leq LB_{s_{q-1}}$. Moreover, $LB + \sum_{t=1}^{|S^*|} (LB_{s_t^*} - LB_{s_{t-1}^*}) = LB_{s_{|S^*|}^*}$ and by definition $LB_{s_{|S^*|}^*} \leq v^*$. Hence, inequality (16) is again satisfied. □

These inequalities can be separated by solving a longest path problem in an acyclic network, which can be done in polynomial time, see Karger et al. [1997], Uehara and Uno [2004] and Ioannidou et al. [2011].

Proposition 3. *Let $L_1 \leq L_2 \leq \ldots \leq L_m$ be the lengths of the MSTs obtained when only one feature distance is considered. The following inequalities are valid:*

$$v^* \geq LB_1 + \sum_{k=1}^{n} (\mathcal{VMST}(c^k) - L_p)^+ y^k, \tag{17}$$

$$v^* \geq LB_2 + \sum_{k=1}^{n} (LB_2^k - LB_2)^+ y^k. \tag{18}$$

Proof. The validity follows from Remarks 2 and 4 and the fact that, in a feasible solution, the number of selected features is equal to p. □

Proposition 4. *Let $\bar{T}$ be a MST for the costs $C_e, e \in E$. Its length is given by LB2. The following inequality is valid:*

$$v^* \geq LB_2 + \sum_{e \in E \setminus \bar{T}} \min_{f \in P_T(e)} (C_e - C_f)^+ x_e. \tag{19}$$

Proof. Let T be a solution tree whose value for FSMST is equal to v^*. We show that it satisfies the above inequality. First, we know that $s^* \geq C(T) = \sum_{e \in T} C_e$.

Next, consider an edge $e \in \bar{T} \setminus T$ and the partition of the vertices obtained when deleting this edge e from $\bar{T}$. There exists an edge, say f of $T \setminus \bar{T}$ whose end vertices belong to the different sets of the partition. Consider the new tree $T' = T \setminus \{f\} \cup \{e\}$ with cost $C(T')$. We have

$$C(T') = C(T) - C_f + C_e \leq C(T) - C_f + C_{e(f)},$$

since e belongs to the unique path linking the end vertices of f in $\bar{T}$ and $e(f)$ is the edge of $\bar{T} \setminus T$ on that path with the highest cost. The tree T' has one more edge in common with $\bar{T}$ than T. We can apply this exchange of edges until we obtain the MST $\bar{T}$, yielding

$$C(T) \geq LB_2 + \sum_{f \in T \setminus \bar{T}} (C_f - C_{f(e)}),$$

that is the value of the right hand side of the valid inequality for the tree T. □

Proposition 4 cannot be generalized to any lower bound, as for instance LB_1, because in general, there is not a MST whose cost is the lower bound.

Remark 6. *Let UB be an upper bound of the problem. If* $\mathcal{VMST}(c^k) > UB$, *then* $y^k = 0$.

4.2 Valid inequalities based on $\mathcal{S}^p, \mathcal{VS}^p, \mathcal{MST}$ or $\mathcal{VMST}$.

Proposition 5. *The following inequalities are valid:*

$$v^* \geq \sum_{k=1}^{n} \mathcal{VMST}(c^k) y^k, \tag{20}$$

$$v^* \geq \sum_{e \in E} \mathcal{VS}^p(c_e) x_e. \tag{21}$$

Further, inequality (20) *is stronger than* (17).

Proof. For any feasible solution to M_1, its value $v = \sum_{e \in E} \sum_{k=1}^{n} c_e^k x_e y^k$. The inequalities come for the facts that $\sum_{e \in E} c_e^k x_e \geq \mathcal{VMST}(c^k)$ and $\sum_{k=1}^{n} c_e^k y^k \geq \mathcal{VS}(c_e)$, respectively.

Next, let S be the set of p features with smallest $\mathcal{VMST}(c^k)$, i.e. $S = \mathcal{S}^p(\mathcal{VMST}(c^k))$. The right hand side of inequality (17) reads:

$$\begin{aligned}
&\sum_{k \in S} \mathcal{VMST}(c^k) + \sum_{k \in K \setminus S} (\mathcal{VMST}(c^k) - \mathcal{VMST}(c^p)) y^k \\
&= \sum_{k \in S} \mathcal{VMST}(c^k) + \sum_{k \in K \setminus S} \mathcal{VMST}(c^k) y^k - \mathcal{VMST}(c^p) \sum_{k \in K \setminus S} y^k \\
&= \sum_{k \in S} \mathcal{VMST}(c^k) + \sum_{k \in K \setminus S} \mathcal{VMST}(c^k) y^k - \mathcal{VMST}(c^p)(p - \sum_{k \in S} y^k) \\
&= \sum_{k \in S} (\mathcal{VMST}(c^k) - \mathcal{VMST}(c^p)(1 - y^k)) + \sum_{k \in K \setminus S} \mathcal{VMST}(c^k) y^k \\
&\leq \sum_{k=1}^{n} \mathcal{VMST}(c^k) y^k,
\end{aligned}$$

since $\mathcal{VMST}(c^k) \leq \mathcal{VMST}(c^p)$ for all $k \in S$. □

Proposition 6. *The following inequalities are valid.*

For every $\bar{k} \in K$*:*

$$v^* \geq p \sum_{e \in E} c_e^{\bar{k}} x_e + \sum_{k \in K} \mathcal{VMST}(c^k - c^{\bar{k}}) y^k. \tag{22}$$

For every edge $\bar{e} \in E$*:*

$$v^* \geq (n-1) \sum_{k \in K} c_{\bar{e}}^k y^k + \sum_{e \in E} \mathcal{VS}^p(c_e - c_{\bar{e}}) x_e. \tag{23}$$

Proof. For a given feasible solution to M_1, let $F = \{k \in K : y^k = 1\}$ and $T = \{e \in E : x_e = 1\}$. Its value for M_1 is equal to:

$$\begin{aligned} v &= \sum_{e\in T}\sum_{k\in F} c_e^k \\ &= p\sum_{e\in E} c_e^{\bar{k}} x_e + \sum_{k\in K}\sum_{e\in E}(c_e^k - c_e^{\bar{k}})x_e y^k \\ &\geq p\sum_{e\in E} c_e^{\bar{k}} x_e + \sum_{k\in K}\mathcal{VMST}(c^k - c^{\bar{k}})y^k. \end{aligned}$$

Similarly,

$$\begin{aligned} v &= (n-1)\sum_{k\in K} c_{\bar{e}}^k y^k + + \sum_{k\in K}\sum_{e\in E}(c_e^k - c_{\bar{e}^k})x_e y^k \\ &\geq (n-1)\sum_{k\in K} c_{\bar{e}}^k y^k + \sum_{e\in E}\mathcal{VS}^p(c_e - c_{\bar{e}})x_e. \end{aligned}$$

□

Proposition 7. *Let $\{K_\ell, \ell \in L\}$ be a partition of the feature set K into $|L|$ nonempty subsets and let $\{k_\ell\}, \ell \in L$ be a subset of $|L|$ distinct features. Then, the following inequality is valid.*

$$v^* \geq p\sum_{e\in E}\min_{\ell\in L}\{c_e^{k_\ell}\}x_e + \sum_{\ell\in L}\sum_{k\in K_\ell}\mathcal{VMST}(c^k - c^{k_\ell})y^k. \tag{24}$$

Proof. For a given feasible solution to M_1, its value v satisfies:

$$\begin{aligned} v &= \sum_{\ell\in L}\sum_{e\in E} c_e^{k_\ell} x_e \sum_{k\in K_\ell} y^k + \sum_{\ell\in L}\sum_{k\in K_\ell}\sum_{e\in E}(c_e^k - c_e^{k_\ell})x_e y^k, \\ &\geq \sum_{\ell\in L}\sum_{e\in E}\min_{\ell\in L}\{c_e^{k_\ell}\}x_e \sum_{k\in K_\ell} y^k + \sum_{\ell\in L}\sum_{k\in K_\ell}\sum_{e\in E}(c_e^k - c_e^{k_\ell})x_e y^k, \\ &\geq p\sum_{e\in E}\min_{\ell\in L}\{c_e^{k_\ell}\}x_e + \sum_{\ell\in L}\sum_{k\in K_\ell}\mathcal{VMST}(c^k - c^{k_\ell})y^k. \end{aligned}$$

□

Note that the coefficient of variable x_e can be (slightly) improved if there are less than p features in the subset K_ℓ for which $c_e^{k_\ell}$ is minimum.

Proposition 8. *Let F be a subset of p features. The following inequality is valid:*

$$v^* \geq \sum_{k\in F}\sum_{e\in E} c_e^k x_e + \sum_{k\notin F}\min_{k'\in F}\mathcal{VMST}(c^k - c^{k'})y^k. \tag{25}$$

Proof. For a given solution x, y to M_1, let F' represent the set of selected features, i.e. $F' = \{k : y^k = 1\}$. Since $|F'| = |F|$, we may define a one-one mapping, say $\epsilon(k)$ that assigns an element of $F \setminus F'$ to each element of $F' \setminus F$ so that

$$\sum_{k\in F'}\sum_{e\in E} c_e^k x_e = \sum_{k\in F}\sum_{e\in E} c_e^k x_e + \sum_{k\in F'\setminus F}\sum_{e\in E}(c_e^k - c_e^{\epsilon(k)}))x_e.$$

Consequently,

$$v^* \geq \sum_{k\in F}\sum_{e\in E} c_e^k x_e + \sum_{k\notin F}\sum_{e\in E}(c_e^k - c_e^{\epsilon(k)})x_e y^k.$$

The valid inequality is obtained by noting that

$$\sum_{e\in E}(c_e^k - c_e^{\epsilon(k)})x_e \geq \min_{k'\in F\setminus F'} \mathcal{VMST}(c^k - c^{k'}).$$

□

Proposition 9. *Let T be a spanning tree. The following inequality is valid:*

$$v^* \geq \sum_{e\in T}\sum_{k=1}^{m} c_e^k y^k + \sum_{e\in E\setminus T} \min_{f\in P_T(e)} \mathcal{S}^p(c_e - c_f)x_e, \tag{26}$$

where $c_e = (c_e^k)_{k=1}^m$ and $c_e - c_f$ is thus the difference of the cost vectors of edges e and f.

Proof. Let x and y define a feasible solution to M_1. Using Lemma 2, we have:

$$\sum_{e\in E}\sum_{k=1}^{m} c_e^k y^k x_e \geq \sum_{e\in T}\sum_{k=1}^{m} c_e^k y^k + \sum_{e\in E\setminus T}\sum_{k=1}^{m}(c_e^k - c_{\sigma(e)}^k)y^k x_e$$

The appropriate lower bound is then obtained by noticing that

$$\sum_{k=1}^{m}(c_e^k - c_{\sigma(e)}^k)y^k \geq \mathcal{S}^p(c_e - c_{\sigma(e)}),$$

then, replacing $\sigma(e)$ by the edge f of $P_T(e)$ that provides the smallest value for $\mathcal{S}^p(c_e - c_f)$. □

The following example illustrates the last two families of valid inequalities.

Example 1. *(cont): For $F = \{1, 2\}$, cut (25) would read:*

$$\begin{aligned} v^* \geq\ & 7x_{(1,2)} + 0x_{(1,3)} + 6x_{(1,4)} + 6x_{(2,3)} + 6x_{(2,4)} + 7x_{(3,4)} + \\ & 3x_{(1,2)} + 0x_{(1,3)} + 1x_{(1,4)} + 3x_{(2,3)} + 9x_{(2,4)} + 1x_{(3,4)} + \\ & \min\{\mathcal{VMST}(c^3 - c^1), (c^3 - c^2)\}y^3. \end{aligned}$$

Since $\mathcal{VMST}(c^3 - c^1) = -1$ and $\mathcal{VMST}(c^3 - c^2) = 5$, the cut reads

$$v^* \geq 10x_{(1,2)} + 7x_{(1,4)} + 9x_{(2,3)} + 15x_{(2,4)} + 8x_{(3,4)} - y^3$$

For this example, the family of valid inequalities (25) has three cuts: one for $F = \{1, 2\}$, one for $F = \{1, 3\}$ and another for $F = \{2, 3\}$.

For the spanning tree (1,3), (1,4), (2,3), cut (26) would read:

$$\begin{aligned} v^* \geq\ & (6 + 0 + 6)y^1 + (3 + 0 + 1)y^2 + (10 + 1 + 6)y^3 + \\ & \min\{S^p(c_{(1,2)} - c_{(2,3)}), S^p(c_{(1,2)} - c_{(1,3)})\}x_{(1,2)} + \\ & \min\{S^p(c_{(2,4)} - c_{(1,3)}), S^p(c_{(2,4)} - c_{(1,4)}), S^p(c_{(2,4)} - c_{(2,3)})\}x_{(2,4)} + \\ & \min\{S^p(c_{(3,4)} - c_{(1,3)}), S^p(c_{(3,4)} - c_{(1,4)})\}x_{(3,4)}, \end{aligned}$$

leading to

$$v^* \geq 12y^1 + 4y^2 + 17y^3 - 1x_{(1,2)} + 0x_{(2,4)} + 1x_{(3,4)}.$$

For this example, the family of valid inequalities (26) *has 16 cuts, one for each spanning tree in the graph.*

4.3 Other valid inequalities

Proposition 10. *Let e be an arc in the graph. Let $M_e = \{k \in K : e \in \mathcal{MST}(c^k)\}$. If $|M_e| \geq p$, then the inequalities*

$$x_e + p - 1 \geq \sum_{k \in M} y^k \qquad \forall M \subseteq M_e : |M| = p \tag{27}$$

are valid inequalities for M_1.

Proof. If for the p selected features there is a MST using e, then e is in the optimal solution of the problem. □

5 Alternative models and decomposition approach

In this section, three new models for Problem FSMST are proposed. The new models presented in this section have the advantage that they use fewer variables. The model presented in Section 3 uses x-variables and w-variables to define the tree, y-variables to deal with the features and z-variables to linearize the products of x and y-variables. The first two models that we propose in this section do not need the z-variables since the feasible region is described without products of variables. The last model in this section only uses y-variables which makes it much smaller. In the second part of this section we propose a decomposition algorithm for the formulation with fewer variables.

The following proposition states that a valid formulation for FSMST is obtained by minimizing v^* over the cardinality constraint for the features, the spanning tree constraints and either inequalities (25) for all subset of p features or (26) for all spanning trees. This result confers a special relevance to valid inequalities from Propositions 8 and 9 .

Proposition 11. *Let $\mathcal{F}$ denote the set of all subsets of p features and $\mathcal{T}$ denote the set of all spanning trees T. Both, models*

$$\begin{aligned}
(M_2) \quad \min_{x,y,v} \quad & v \\
s.t. \quad & (2), (3), (12) - (15) \\
& v \geq \sum_{k \in F} \sum_{e \in E} c_e^k x_e + \sum_{k \notin F} \min_{k' \in F} \mathcal{VMST}(c^k - c^{k'}) y^k, \quad \forall F \in \mathcal{F}
\end{aligned}$$

and

$$
\begin{aligned}
(M_3) \quad \min_{x,y,v} \quad & v \\
s.t. \quad & (2), (3), (12) - (15) \\
& v \geq \sum_{e \in T} \sum_{k=1}^{m} c_e^k y^k + \sum_{e \in E \setminus T} \min_{f \in P_T(e)} \mathcal{S}^p (c_e - c_f) x_e, \quad T \in \mathcal{T}
\end{aligned}
$$

constitute valid formulations for FSMST.

Proof. Since we have an inequality of type (26) for each spanning tree and an inequality (25) for each subset of p features, we have in particular those for the optimal spanning tree and for the optimal subset of features, respectively. These inequalities provide the optimal value for FSMST. □

The main advantage of the models in the above proposition is that they do not make use of variables z_e^k. Thus, neither do they need constraints (5)-(9). However, an exponential number of constraints (26) or (25) are required.

Observing FSMST, one realises that it is composed by two simpler problems: the Feature Selection Problem and the Minimum Spanning Tree Problem. Complete linear description of the convex hull of binary solutions for these two separate problems are well known, and even more, separately they can be solved by using very fast algorithms. However, when we try to solve them together, as problem FSMST, we need to include binary constraints on the variables and, as expected and as will be shown later in the numerical experiments, the problem becomes much harder to solve. Thus, given this particular structure of FSMST, one may think that it is an ideal problem for decomposition. The following theorem proposes a formulation involving only the y-variables but an exponential number of constraints whose separation problem amounts to finding a minimum spanning tree with respect to some particular costs.

Theorem 3. *The following model*

$$
\begin{aligned}
(M_4) \quad \min_{y,v} \quad & v \\
s.t. \quad & v \geq \mathcal{VMST}(\sum_{k \in F} c^k) + \sum_{k \notin F} \min_{k' \in F} \mathcal{VMST}(c^k - c^{k'}) y^k, \quad F \in \mathcal{F} \qquad (28) \\
& \sum_{k=1}^{K} y^k = p, \\
& y^k \in \{0,1\}, \quad k = 1, \ldots, m,
\end{aligned}
$$

is a valid formulation of FSMST.

Proof. First, given that for all feasible feature set $F \in \mathcal{F}$ and for all feasible vector x, i.e. corresponding to spanning trees, $\sum_{k \in F} \sum_{e \in E} c_e^k x_e \geq \mathcal{VMST}(\sum_{k \in F} c^k)$. Hence M_4 constitutes a relaxation of M_2 and $v^*(M_4) \leq v^*(M_2)$.

Next, consider an optimal solution (v^*, y^*) to M_4 and let F^* be the set of features k such that $y^{*k} = 1$. We have, $\mathcal{VMST}(\sum_{k\in F*} c^k) \leq v^*(M_4) \leq v^*(M_2) \leq \mathcal{VMST}(\sum_{k\in F*} c^k)$, where the first inequality is obtained by using inequality (28) for $F = F^*$ and the third one comes from the fact that F^* and the corresponding minimum spanning tree constitute a feasible solution to M_2 with value $\mathcal{VMST}(\sum_{k\in F*} c^k)$.

□

The idea of the Decomposition Algorithm is to solve at each iteration a restricted integer master problem that involves inequality (28) for only a subfamily, say $\bar{\mathcal{F}}$, of feature sets. If its optimal solution (v^*, y^*) satisfies inequality (28) for the set F^* of features defined by y^* then, following the same reasoning as in the proof of Theorem 3, the algorithm terminates and the optimal solution has been found. Otherwise, inequality (28) for set F^* is added to the restricted master, upper and lower bounds are updated and the algorithm proceeds to the next iteration. Given that a cut corresponding to a different feature subset is added at each iteration, the algorithm terminates in a finite number of iterations. The description of our solution method is presented in Algorithm 1.

The Master Problem in Algorithm 1 can be reinforced by including all the previously developed valid inequalities and remarks that involve only y variables ((16), (17), (18), (20) and Remark 6). We will call this reinforcement of the algorithm *Algorithm 1 reinforced*. Further, $\min_{k'\in F} \mathcal{VMST}(c^k - c^{k'})$ can be efficiently computed by using Kruskal and sorting algorithms.

6 Computational results

This section is devoted to present the numerical results obtained in a computational experiment conducted to compare the performance of the different formulations, valid inequalities and algorithm (Sect. 6.1), and to analyze the behaviour of the feature selection in hierarchical clustering minimizing the total size of the tree (Sect. 6.2).

In order to conduct the numerical study, we generated instances of different sizes. We first fixed the number of individuals (n), features (K) and features to select (p). The different combinations chosen for these parameters along the study are shown in the first three columns of each table reporting results. In particular, $n \in \{20, 40, 50, 200, 400\}$, $K \in \{9, 10, 11, 12, 15\}$ and $p \in \{4, 5, 6, 7\}$. Once these parameters fixed, to create different random instances, we generated random costs by using normal and uniform distributions (one for each feature). Values were later normalized. For each combination of parameters n, K and p, five different instances were generated.

The computational experiment was carried out on a personal computer with Intel® Core i7-1065G7, 1.30GHz and 1.50GHz with 16 GB RAM. The optimization problems were solved exactly by using Gurobi Version: 9.1.2.

Result: $UB = LB = v^*(P)$

$UB \leftarrow \infty$;

$LB \leftarrow -\infty$;

$\tau \leftarrow 0$;

$\Omega = \{\emptyset\}$;

Choose $\bar{y} = (\bar{y}^1, \ldots, \bar{y}^K)$ a configuration of p features;

while $UB > LB$ **do**

Do $\tau = \tau + 1$;

Do $\Omega = \Omega \cup \{\tau\}$;

Find a MST, x^τ, for the complete graph with arc costs $c_e = \sum_{k=1}^{m} c_e^k \bar{y}^k$;

If $v = \sum_{e \in E} \sum_{k=1}^{K} c_e^k \bar{y}^k x_e^\tau < UB$, then do $UB = v$;

Find

$$\bar{c}_k^\tau = \begin{cases} \min_{k':\bar{y}_{k'}^{\tau-1}=0} VMST(c^k - c^{k'}) & \text{if } \bar{y}_k = 0, \\ 0 & \text{if } \bar{y}_k = 1 \end{cases}$$

Solve:

$$\begin{aligned} \min_{y,\gamma} \quad & \gamma && \text{(Master Problem)} \\ \text{s.t.} \quad & \sum_{k=1}^{K} y^k = p, \\ & y^k \in \{0,1\}, \quad k = 1, \ldots, K, \\ & \gamma \geq \sum_{e \in E} \sum_{k=1}^{m} c_e^k x_e^\nu + \sum_{k=1}^{m} \bar{c}_k^\nu y^k, \quad \nu \in \Omega \end{aligned}$$

Do $\bar{y} = y^*$ and $LB = \gamma^*$;

end

Algorithm 1: Decomposition Algorithm with (28)

6.1 Formulations, valid inequalities and algorithm comparison

When stating the first model, that is, M_1, we proved in Proposition 1 that different combinations of constraints lead to the same feasible region. We first compare, in terms of gap and resolution time, the performance of the following three constraints combinations:

- M_1.
- M_1^1, which is M_1 plus (10) and (11).
- M_1^2, which is M_1 removing (6) and (7), and adding (10) and (11).

Table 1 reports the results of this comparison. To calculate the gap, we compute the difference between the optimal value of the mixed integer model and the optimal value of the linear relaxation, and divide it by the optimal value of the mixed integer model.

We can observe in Table 1 that there is a clearly winner combination with respect to the gap, M_1^1. The gap is always smaller, that is, the optimal value of the linear relaxation is always closer to optimal value of Problem FSMST when M_1 contains constraints (10) and (11). The optimal value of the linear relaxation of M_1 is always zero giving a gap equal to 1, hence, this is always the worst possible combination. In M_1^2, the gap is always higher than M_1^1, but much smaller than 1; for example, for the first instance type of the table, with $n = 20, k = 12$ and $p = 5$, the gap for M_1^1 is 0.14 meanwhile for M_1^2 is 0.23. If we consider time, there is again an obvious loser combination, M_1; however, depending on the instance type, M_1^1 or M_1^2 perform better. For example, for the last instance with $n = 40, K = 10, p = 5$, the average time required to solve the problem was 397.62 seconds for M_1^1 and 645.36 for M_1^2; meanwhile for the third instance of the same combination of parameters, M_1^1 solved the problem to optimality in 385.51 seconds and M_1^2 in 150.9 seconds. Therefore, M_1^1 and M_1^2 seem both reasonable combinations to model FSMST.

We next compare the performance of the different proposed valid inequalities in M_1^1, in Table 2, and in M_1^2, in Table 3.

In Tables 2 and 3, we show the resulting gap and resolution time when each valid inequality is added to formulation M_1^1 and M_1^2, respectively. The reason for not including in the Tables 2 and 3 some non-dominated valid inequalities is that they were useless or very time consuming. We also considered adding combinations of valid inequalities to the models, however, few significant changes were observed. We highlight in blue color the best time and gap for each type of instances. We also emphasize in green color the values (time or gap) that despite not being the best ones for the corresponding instance type, improve the value (time or gap) obtained for the model without including such additional valid inequality.

We can observe in Table 2 that valid inequality (18) added to M_1^1 reports the smallest gaps for all the instances, for example, for the last instance of size $n = 20, K = 12, p = 5$, the gap is reduced from 0.11 to 0.05. By contrast, there is not a valid inequality that leads to the smaller times for all the instances. In 5 out of the 15 considered instances, M_1^1 without any additional valid inequality is faster. In other 5 instances, M_1^1 + (19) is the fastest; for example, in the forth instance of size $n = 50, K = 9, p = 4$, the execution time is reduced from 1199.51 seconds to

			M_1^1		M_1		M_1^2	
n	**K**	p	**Gap**	**Time**	**Gap**	**Time**	**Gap**	**Time**
20	12	5	0.14	11.96	1.0	18.19	0.23	14.37
20	12	5	0.13	14.08	1.0	21.84	0.22	9.58
20	12	5	0.12	9.58	1.0	15.34	0.19	7.7
20	12	5	0.13	11.16	1.0	16.62	0.25	9.72
20	12	5	0.11	13.91	1.0	17.7	0.21	11.42
40	10	5	0.11	338.49	1.0	1682.8	0.18	179.82
40	10	5	0.09	385.51	1.0	1488.79	0.17	150.9
40	10	5	0.1	508.19	1.0	1740.72	0.17	142.92
40	10	5	0.12	490.47	1.0	1587.25	0.21	670.45
40	10	5	0.11	397.62	1.0	1666.11	0.18	645.36
50	9	4	0.08	645.48	1.0	TL	0.17	1250.39
50	9	4	0.09	669.77	1.0	TL	0.18	1604.33
50	9	4	0.07	366.13	1.0	TL	0.17	1148.86
50	9	4	0.08	1199.51	1.0	TL	0.19	1121.32
50	9	4	0.08	777.22	1.0	TL	0.15	1278.56

Table 1: Comparison of formulation M_1 containing different constraints combinations according to Proposition 1.

838.04. The 5 remaining instances are solved faster by adding valid inequality (20), (21) or (27). Even though adding valid inequalities (22) or (23) do not report the best time or gap in any case, there exist an instance for which the execution time is improved with respect to M_1^1. In Table 2, M_1^1 +(16) is the only valid inequality not reporting any improvement; however, this is not the case in Table (3).

When we consider as the base model M_1^2, Table 3, we can see that once more, incorporating (18) to M_1^2 results in the smallest gaps. Again, in the last instance of size $n = 20, K = 12, p = 5$, the gap is reduced in this case from 0.21 to 0.06. A similar behaviour that in Table 2 can be observed with respect to times: in 6 out of the 15 instances the best times are reported for M_1^2, in other 5 instances the best times are achieved for M_1^2 +(19), and the remaining by adding either (18) or (21). Unlike Table 2, in Table 3, we can observe that adding (16) to M_1^2 reduces almost all the gaps, and valid inequalities (22) and (23) do not imply any improvement, in fact, for the latter, few instances (5 out of 15) are solved within the time limit.

If we compare the best gaps in Table 2 versus the best gaps in Table 3, we can assert that the smallest gaps are obtained for M_1^1 + (18). If we examine times in in Table 2 versus times in Table 3, we can see that for the first type of instances, $n = 20, K = 12, p = 5$, times are very similar. However, in medium size instances, $n = 40, K = 10, p = 5$, 3 out of the 5 instances are solved much faster by using M_1^2 + (19), the remaining two by using M_1^1 + (19). The smallest execution times to solve the last group of instances, $n = 50, K = 9, p = 4$, are all in Table 2.

Instances			M_1^1		M_1^1 +(16)			M_1^1 +(18)		M_1^1 +(19)		M_1^1 +(27)			M_1^1 +(20)		M_1^1 +(21)		M_1^1 +(22)		M_1^1 +(23)	
n	K	p	Gap	Time	Gap	Cuts	Time	Gap	Time	Gap	Time	Gap	Cuts	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
20	12	5	0.14	11.96	0.14	2	17.55	0.08	17.34	0.14	15.34	0.14	2	14.04	0.14	14.46	0.14	16.6	0.14	14.89	0.14	19.97
20	12	5	0.13	14.08	0.13	2	15.52	0.07	13.6	0.13	15.23	0.13	1	11.37	0.13	10.07	0.13	13.04	0.13	12.37	0.13	13.58
20	12	5	0.12	9.58	0.12	2	14.17	0.06	8.73	0.12	11.17	0.12	0	9.49	0.12	8.89	0.12	11.55	0.12	10.77	0.12	13.54
20	12	5	0.13	11.16	0.13	2	16.76	0.07	13.14	0.13	13.35	0.13	2	12.38	0.13	11.66	0.13	12.0	0.13	13.38	0.13	14.53
20	12	5	0.11	13.91	0.11	2	16.35	0.05	14.89	0.11	14.57	0.11	0	12.81	0.11	14.03	0.11	13.9	0.11	15.48	0.11	19.1
40	10	5	0.11	338.49	0.11	2	363.41	0.06	731.16	0.11	389.01	0.11	0	317.09	0.11	410.21	0.11	385.46	0.11	563.84	0.11	637.59
40	10	5	0.09	385.51	0.09	2	505.04	0.05	554.78	0.09	440.72	0.09	0	418.45	0.09	411.39	0.09	423.32	0.09	723.3	0.09	734.31
40	10	5	0.1	508.19	0.1	2	761.91	0.07	688.2	0.1	420.83	0.1	0	517.98	0.1	497.57	0.1	502.98	0.1	894.55	0.1	1009.84
40	10	5	0.12	490.47	0.12	2	883.37	0.07	728.09	0.12	376.99	0.12	0	497.09	0.12	575.21	0.12	582.45	0.12	722.21	0.12	747.4
40	10	5	0.11	397.62	0.11	2	783.59	0.08	866.05	0.11	291.37	0.11	0	406.8	0.11	433.8	0.11	494.09	0.11	557.8	0.11	655.54
50	9	4	0.08	645.48	0.08	2	932.44	0.05	1242.59	0.08	974.53	0.08	0	686.52	0.08	800.03	0.08	765.2	0.08	1370.72	0.08	875.86
50	9	4	0.09	669.77	0.09	2	766.13	0.07	1201.78	0.09	522.73	0.09	0	688.53	0.09	697.96	0.09	991.86	0.09	1517.92	0.09	974.53
50	9	4	0.07	366.13	0.07	2	583.48	0.05	696.03	0.07	577.57	0.07	1	571.93	0.07	607.21	0.07	618.85	0.07	708.62	0.07	668.41
50	9	4	0.08	1199.51	0.08	2	1151.62	0.05	982.05	0.08	838.04	0.08	0	1214.72	0.08	1093.7	0.08	957.11	0.08	1298.12	0.08	1578.78
50	9	4	0.08	777.22	0.08	2	824.48	0.06	1501.15	0.08	736.03	0.08	0	796.79	0.08	697.28	0.08	579.15	0.08	822.21	0.08	860.91

Table 2: Numerical comparison of Formulation M_1^1 strengthened with different valid inequalities

We now analyze the performance of the decomposition algorithm. We show in Table 3, for each instance, the smallest (best) resolution times of Tables 1 and 2, the resolution times of Algorithm 1, and the total times of Algorithm 1 reinforced (with all developed valid inequalities and remarks that involve only y variables). In particular, Algorithm 1 reinforced is Algorithm 1 + (16) + (18) + (20) + Remark 6. Furthermore, we include in this table the total number of cuts of type (28) required in M_4, and how many of these cuts, Algorithm 1 and Algorithm 1 reinforced required to reach the optimal solution, that is, to converge. We start by comparing resolution times. We can observe in Table 4 that instances with $n = 20, K = 12, p = 5$, are solved very fast by some of the formulations, since the number of variables in this case are moderate; however, the two versions of the algorithm require some more time to get the optimal since the number of cuts that need to be computed and included is high, 792 for Algorithm 1 and between 66 and 249 for Algorithm 1 reinforced, depending on the instance. Nevertheless, for the rest of the instances, solving the problems by using any of the two versions of the algorithm is always faster than solving it using the formulations M_1^1 or M_1^2. The differences between times are very high, for instance, for the first instance of size $n = 50, K = 9, p = 4$, the fastest this is solved using the formulations is in 645.48 seconds, meanwhile using Algorithm 1 it is solved in 67.71 seconds and using Algorithm 1 reinforced in 7.2 seconds. If we compare Algorithm 1 and Algorithm 1 reinforced, we can see that the convergence of Algorithm 1 is always attained when all the cuts are inserted, which is equivalent to finding the solution by enumeration; however, the number of cuts that need to be inserted in Algorithm 1 reinforced to converge is significantly smaller: in instances of size $n = 40, K = 10, p = 5$, it goes from 55 to 62, out of 252, and in instances of size $n = 50, K = 9, p = 4$, it goes from 15 to 25, out of 126; which is translated in a huge reduction of time. Again, we highlight in blue the smallest resolution times.

Instances			M_1^2		M_1^2 +(16)			M_1^2 +(18)		M_1^2 +(19)		M_1^2 +(27)			M_1^2 +(20)		M_1^2 +(21)		M_1^2 +(22)		M_1^2 +(23)	
n	K	p	Gap	Time	Gap	Cuts	Time	Gap	Time	Gap	Time	Gap	Cuts	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
20	12	5	0.23	13.87	0.21	3	87.3	0.09	23.93	0.23	15.61	0.23	2	14.33	0.23	12.62	0.23	13.52	0.23	18.8	0.23	102.61
20	12	5	0.22	10.09	0.21	3	74.96	0.09	11.45	0.22	12.77	0.22	1	13.05	0.22	13.75	0.22	10.14	0.22	14.18	0.22	79.87
20	12	5	0.19	8.1	0.18	3	67.74	0.09	13.65	0.19	8.79	0.19	0	8.93	0.19	10.27	0.19	8.73	0.19	14.25	0.19	88.51
20	12	5	0.25	10.03	0.23	3	98.38	0.09	9.98	0.25	10.87	0.25	2	12.81	0.25	11.69	0.25	11.11	0.25	14.27	0.25	92.63
20	12	5	0.21	11.71	0.19	3	328.67	0.06	15.66	0.21	12.04	0.21	0	13.92	0.21	13.42	0.21	12.99	0.21	15.47	0.21	78.74
40	10	5	0.18	174.1	0.16	3	TL	0.07	647.79	0.18	147.02	0.18	0	160.03	0.18	227.86	0.18	519.98	0.18	187.87	0.18	TL
40	10	5	0.17	156.13	0.16	3	TL	0.07	177.18	0.17	104.48	0.17	0	133.65	0.17	154.48	0.17	96.98	0.17	175.4	0.17	TL
40	10	5	0.17	140.63	0.16	4	TL	0.08	530.11	0.17	124.81	0.17	0	127.48	0.17	584.11	0.17	478.13	0.17	639.31	0.17	TL
40	10	5	0.21	591.7	0.19	3	TL	0.08	682.25	0.21	572.51	0.21	0	606.35	0.21	739.15	0.21	549.96	0.21	854.04	0.21	TL
40	10	5	0.18	527.72	0.17	3	696.02	0.08	764.8	0.18	463.98	0.18	0	538.07	0.18	598.29	0.18	552.1	0.18	790.93	0.18	TL
50	9	4	0.17	1258.83	0.16	3	TL	0.08	1710.35	0.17	1276.07	0.17	0	1445.02	0.17	1694.41	0.17	1432.76	0.17	1399.83	0.17	TL
50	9	4	0.18	TL	0.18	3	TL	0.07	TL	0.18	TL	0.18	0	TL	0.18	TL	0.18	1387.15	0.18	1730.3	0.18	TL
50	9	4	0.17	1303.81	0.16	3	TL	0.07	1469.83	0.17	1484.08	0.17	1	1420.25	0.17	1743.8	0.17	1330.57	0.17	TL	0.17	TL
50	9	4	0.19	1293.96	0.18	3	TL	0.07	1459.61	0.19	1416.6	0.19	0	1270.52	0.19	1638.27	0.19	1329.16	0.19	TL	0.19	TL
50	9	4	0.15	1451.73	0.14	3	TL	0.08	1210.57	0.15	1527.09	0.15	0	1458.55	0.15	1555.22	0.15	1130.98	0.15	TL	0.15	TL

Table 3: Numerical comparison of Formulation M_1^2 strengthened with different valid inequalities

Instances			**Best Time**	**Total Cuts**	**Algorithm 1**		**Alg. 1 reinforced**	
n	K	p	Time		Cuts	Time	Cuts	Time
20	12	5	11.96	792	792	401.29	249	103.87
20	12	5	9.58	792	792	402.95	216	84.73
20	12	5	7.7	792	792	426.12	99	31.04
20	12	5	9.72	792	792	378.25	89	28.04
20	12	5	11.42	792	792	551.41	66	19.33
40	10	5	179.82	252	252	102.67	62	29.38
40	10	5	150.9	252	252	102.85	55	25.49
40	10	5	142.92	252	252	108.1	56	25.93
40	10	5	376.99	252	252	159.0	56	26.09
40	10	5	291.37	252	252	180.34	57	26.55
50	9	4	645.48	126	126	67.71	21	7.2
50	9	4	522.73	126	126	91.92	15	5.74
50	9	4	366.13	126	126	91.61	15	5.12
50	9	4	838.04	126	126	91.38	15	7.51
50	9	4	579.15	126	126	65.13	25	15.16

Table 4: Numerical comparison of best time from Formulation M_1^1 and M_1^2, Algorithm 1 and Algorithm 1 reinforced.

Finally, we show in Table 5, the performance of Algorithm 1 reinforced when solving bigger instances that cannot be solved by using any of the formulations (within the time limit of 1800 seconds). We can see that instances with 200 individuals, 11 different features and $p = 6$, are solved in approximately 350 seconds, and only around 80 of the 462 different cuts must be included to reach an optimal solution. When we increase the number of features to $K = 15$ and $p = 7$, the number of different cuts increases to 6435, nevertheless, the algorithm finds an optimal solution by including approximately 45 of them, which lasts around 250 seconds. The bigger instances we consider, 400 individuals, 15 features and $p = 7$, are harder to solve, all of them require more than 1800 seconds, even though the number of cuts that have to be included is not so high, between 67 and 71, but the mixed integer Master Problem that has to be solved 71 times becomes harder after each iteration. Note that the bigger instances we are able to solve with the formulations are of size $n = 50, K = 9, p = 4$, and with Algorithm 1 reinforced we increased this size to $n = 400, K = 15, p = 7$.

We show in Figure 2 the convergence of one of these instances of size $n = 400, K = 15, p = 7$, that is, the lower and upper bound in each iteration. We can see that the upper bound reaches very fast the optimal value, but it takes until 69 iterations to increase the lower bound until such value. We repeatedly observed this behaviour in some of the instances: more iterations are needed to increase the lower bound until the optimal value than to decrease the upper bound.

Instances			**Total cuts**	**Alg. 1 reinforced**	
n	K	p		Cuts	Time
200	11	6	462	79	328.97
200	11	6	462	82	350.17
200	11	6	462	80	338.07
200	11	6	462	82	353.23
200	11	6	462	82	348.6
200	15	7	6435	44	270.48
200	15	7	6435	45	257.66
200	15	7	6435	45	288.42
200	15	7	6435	41	251.27
200	15	7	6435	49	274.23
400	15	7	6435	67	1869.19
400	15	7	6435	74	2261.1
400	15	7	6435	69	2078.81
400	15	7	6435	69	2379.48
400	15	7	6435	71	3876.7

Table 5: Numerical results of Algorithm 1 reinforced

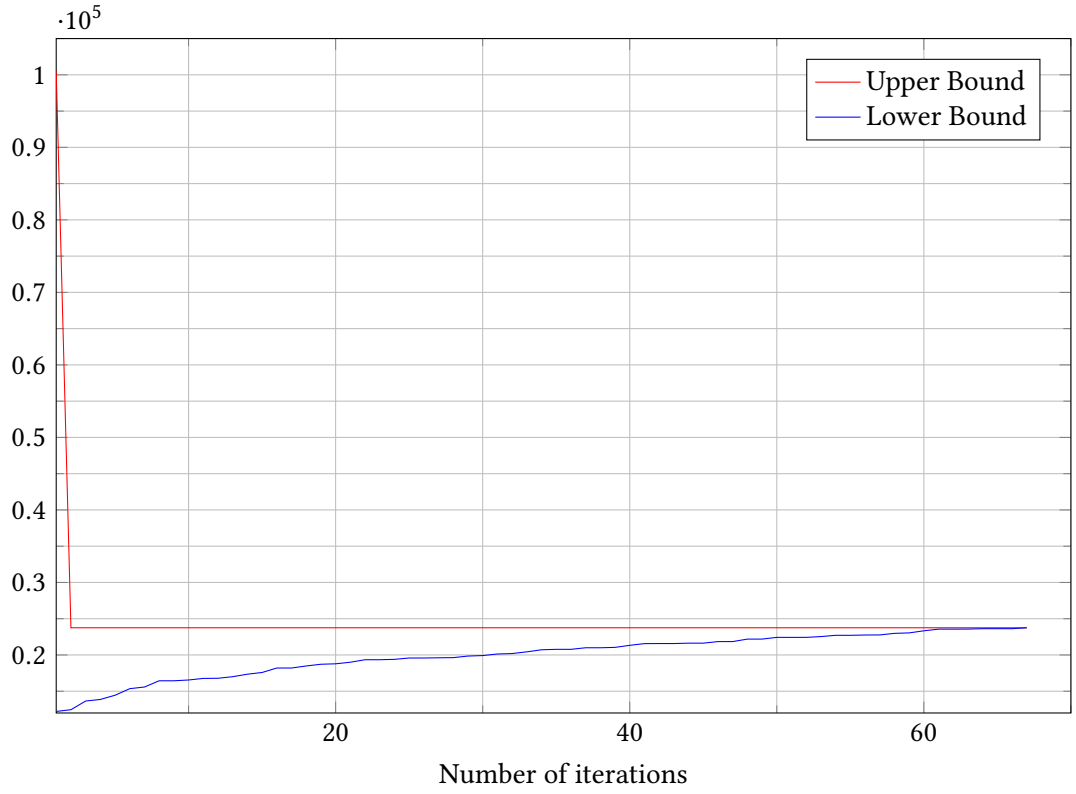

Figure 2: Upper and lower bounds obtained in each iteration of Algorithm 1 reinforced for an instance with 400 nodes, 15 features and $p = 7$

6.2 Feature selection in MST insights

In this section, we analyze the quality of the obtained solution. We do so by checking the similarity between clusters defined when all the features are considered and clusters when only the *best* p features, according to our model, are taken into account. Furthermore, we check if the clustering produced when choosing p random features is similar to the clustering obtained when the best p features are chosen; that is, we compare if selecting p random features reports similar results to selecting p features according to our model, in such case, applying FSMST would be useless.

To conduct this comparison, following Questier et al. [2002], we consider the Wallace measure, Wallace [1983], to quantitatively compare two clustering results of the same dataset. This is a standard method for comparing two hierarchical clusterings. Given two clusterings G and H, the Wallace measure provides the probability that a randomly chosen pair of objects is within the same class in one clustering G, given that it is in the same class in clustering H. If we denote by n_{ij} the number of objects classified in group g_j of partition G and in h_i of partition H, and by $n_{i.} = \sum_{j=1,\ldots,m} n_{ij}$, the Wallace measure can be defined as:

$$S_W = \frac{\sum_{i=1}^{l} \sum_{j=1}^{m} \binom{n_{ij}}{2}}{\sum_{i=1}^{l} \binom{n_{i.}}{2}}.$$

We show, in Figures 3 and 4, a graphic with the Wallace measure values, y-axis, when we compare the clustering for different numbers of groups, x-axis, when all features are selected versus selecting only p. That is, we show how similar the clusters created using the FSMST model are to the clusters created accounting for all the features. We show such results for an instance with $n = 50$ and $K = 9$ and $p = 2, 4, 6, 8$ in Figure 3, and in an instance with $n = 200$ and $K = 15$ and $p = 3, 7, 12$ in Figure 4. We can observe that for smaller number of groups, the clusters generated for the FSMST for different p- values are quite similar to those generated by using all the features. For example, in Figure 4, for most of the values of p, until around 45 groups, the probability that two individuals that are in the same group when all the features are considered remain in the same group when only p are selected, according to our model, is higher than 0.5, which implies that the groups are rather similar. Even for $p = 3$, the case in which fewer features are accounted to create the groups, until around 40 groups, this probability is above 0.5. A similar behavour can be seen in Figure 3. When the number of features selected, p, increases, we can observe that the clusters determined by our model tend be more similar to those created for all the features.The lines for the different values of p are crossed because the grouping of a population in a given number of clusters does not have to be followed by the same variables as the grouping for another number of clusters. It is even possible that the best way to group into t clusters uses totally different variables than the classification of the same population into q clusters. For example, the Wallace values for $p = 12$ are always above the others in Figure 4, but this is not the case for $p = 8$ in Figure 3.

In Figures 5 and 6, we show, for the same instances than before, the comparison of clusters created when p random features are selected with the clusters generated when using the FSMST model. We can observe that the clusters are, in general, very different, in fact, we can observe

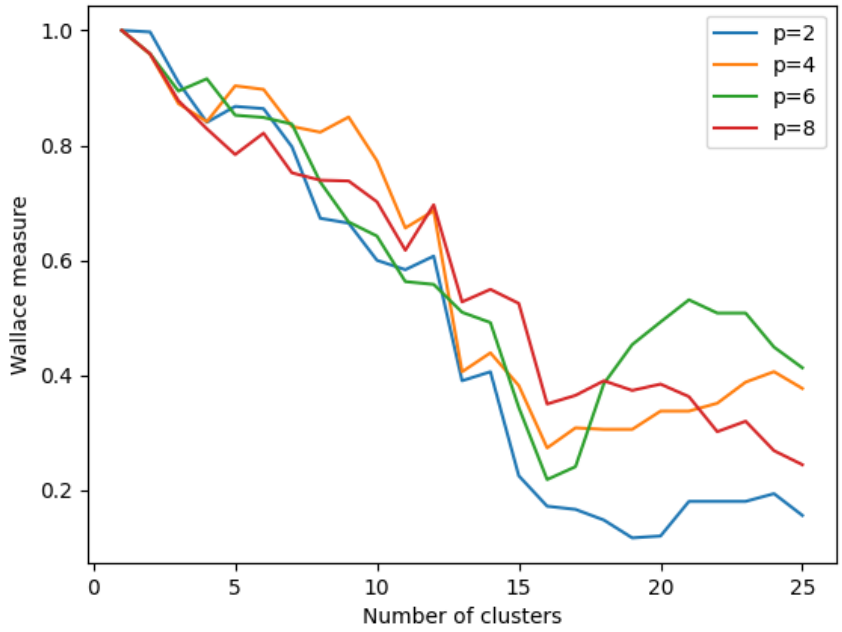

Figure 3: Comparison of clustering with reduced features set, using FSMST, vs all features in terms of Wallace measure (y-axis), for different number of groups (x-axis) and different p values, for an instance with $n = 50$ and $K = 9$.

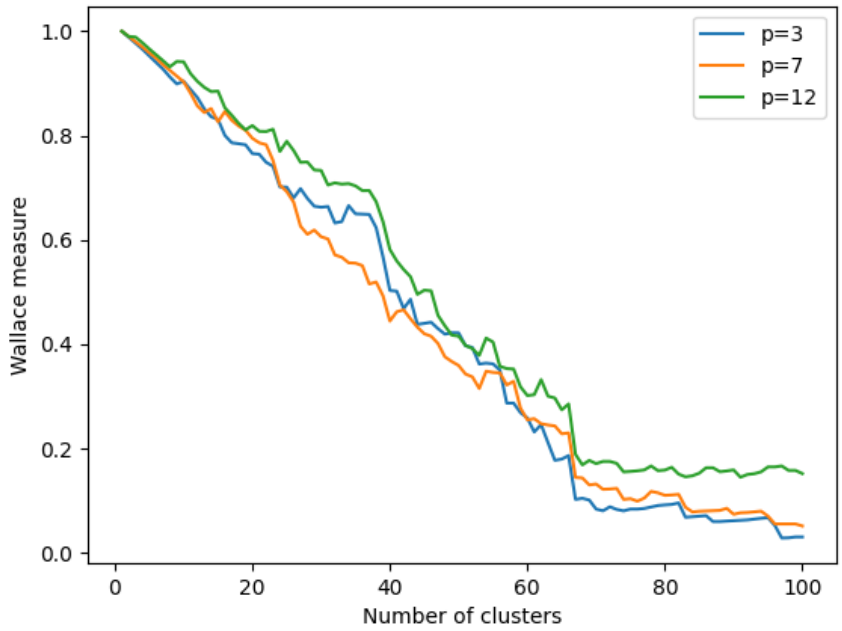

Figure 4: Comparison of clustering with reduced features set, using FSMST, vs all features in terms of Wallace measure (y-axis), for different number of groups (x-axis) and different p values, for an instance with $n = 200$ and $K = 15$.

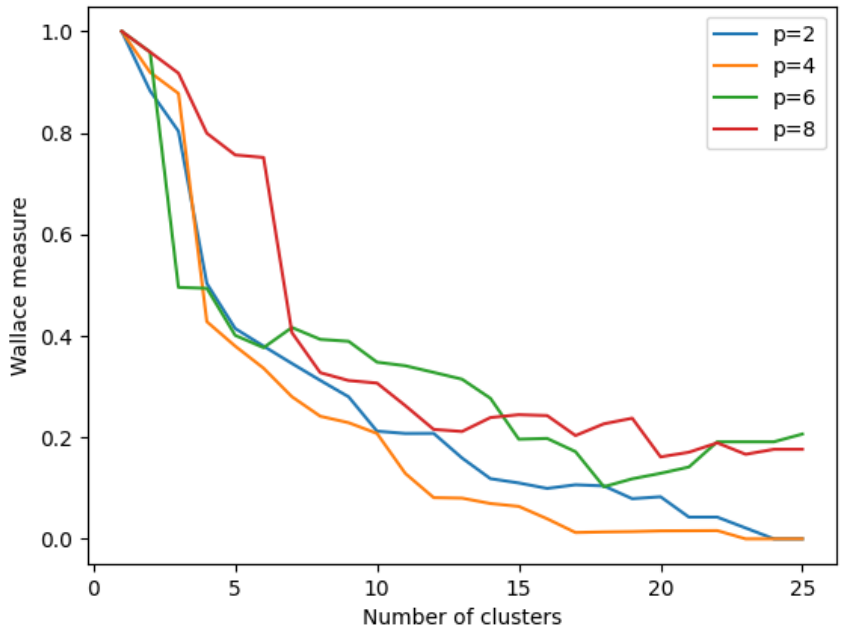

Figure 5: Comparison of clustering with p random features set vs set determined by FSMST, in terms of Wallace measure (y-axis), for different number of groups (x-axis) and different p values, for an instance with $n = 50$ and $K = 9$.

how the Wallace measure decreases to less than 0.5 very quickly, for most of the p values. This measure is below 0.5 for all the $p - values$ for more than 8 groups for the instance with $n = 50$ and for more than 18 groups for the instance with $n = 200$. These results highlight the utility of reducing the selection of features using the proposed approach.

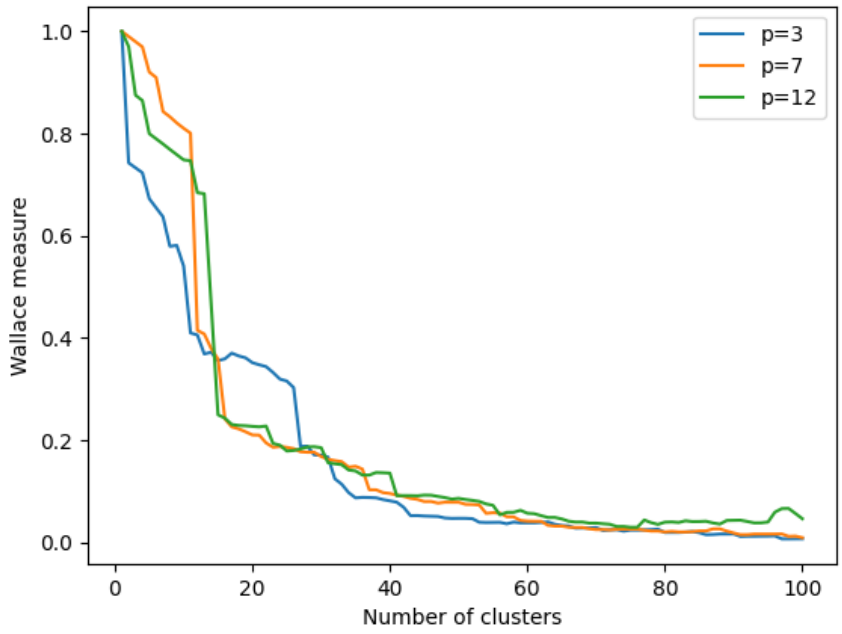

Figure 6: Comparison of clustering with p random features set vs set determined by FSMST, in terms of Wallace measure (y-axis), for different number of groups (x-axis) and different p values, for an instance with $n = 200$ and $K = 15$.

7 Conclusions

In this work, we proposed a framework for selecting features of a data set with the goal of obtaining a dendrogram using the single linkage method, that is, a minimum spanning tree, with the minimum possible total costs. Such a dendrogram/MST is also determined within the proposed framework.

We developed different formulations. For the first one, we proposed several valid inequalities which result in a reduction of the resolution times and also of the integrality gap. For the rest, we designed a decomposition algorithm reinforced with valid inequalities, which exploits the structure of the involved problems: feature selection and MST. The latest formulation provided the fastest resolution times for most of the studied instances.

We conducted a numerical study to test the proposed formulations, valid inequalities and decomposition algorithm, and to get insights about the designed framework. This study revealed that the incorporation of feature selection to the MST model allows to maintain a higher percentage of original information and relationships; this percentage decreases when the features are selected randomly.

The study of more general objective functions and the incorporation of features selection in other hierarchical clustering methods will be the topic of forthcoming works.

Acknowledgements

This work was partially supported by the Spanish Ministry of Science and Innovation through projects PGC2018-099428-B-100, RED2018-102363-T and PID2020-114594GB-C21, also by PROMETEO/2021/063 funded by the governments of Spain and the Valencian Autonomous Region, and by P18-FR-1422 and US-1256951 funded by Junta de Andalucía and FEDER.

References

S. Benati, S. García, and J. Puerto. Mixed integer linear programming and heuristic methods for feature selection in clustering. Journal of the Operational Research Society, 69(9):1379–1395, 2018.

M. Chavent, R. Genuer, and J. Saracco. Combining clustering of variables and feature selection using random forests. Communications in Statistics - Simulation and Computation, 50(2): 426–445, 2021.

S. Ghosal, R. Bhattacharyya, and M. Majumder. Impact of complete lockdown on total infection and death rates: A hierarchical cluster analysis. Diabetes Metabolic Syndrome: Clinical Research Reviews, 14(4):707–711, 2020.

J. C. Gower and G. J. S. Ross. Minimum spanning trees and single linkage cluster analysis. Journal of the Royal Statistical Society. Series C (Applied Statistics), 18(1):54–64, 1969.

P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. Mathematical programming, 79(1):191–215, 1997.

K. Ioannidou, G. Bertzios, and S. Nikolopoulos. The longest path problem has a polynomial solution on interval graphs. Algorithmica, 61:320–341, 2011.

G. Kahvecioğlu and D. P. Morton. Optimal hierarchical clustering on a graph. Networks, 79(2): 143–163, 2022.

D. Karger, R. Motwani, and G. Ramkumar. On approximating the longest path in a graph. Algorithmica, 18:82–98, 1997.

M. Labbé, M. Pozo, and J. Puerto. Computational comparisons of different formulations for the stackelberg minimum spanning tree game. International Transactions in Operational Research, 00:1–22, 2019.

A. N. Letchford and M. M. Sørensen. A new separation algorithm for the boolean quadric and cut polytopes. Discrete Optimization, 14:61–71, 2014. ISSN 1572-5286.

R. K. Martin. Using separation algorithms to generate mixed integer model reformulations. Operations Research Letters, 10(3):119–128, 1991.

F. Nielsen. Hierarchical clustering. In Introduction to HPC with MPI for Data Science, pages 195–211. Springer, 2016.

K. R. O. Yim. Hierarchical cluster analysis: comparison of three linkage measures and application to psychological data. The quantitative methods for psychology, 11(1):8–21, 2015.

M. Padberg. The boolean quadric polytope: Some characteristics, facets and relatives. Mathematical Programming, 45:139–172, 1989.

F. Questier, B. Walczak, D. Massart, C. Boucon, and S. de Jong. Feature selection for hierarchical clustering. Analytica Chimica Acta, 466(2):311–324, 2002. ISSN 0003-2670. doi: https://doi.org/10.1016/S0003-2670(02)00591-3. URL https://www.sciencedirect.com/science/article/pii/S0003267002005913.

R. Uehara and Y. Uno. Efficient algorithms for the longest path problem. In: Fleischer, R., Trippen, G. (eds) Algorithms and Computation. ISAAC 2004. Lecture Notes in Computer Science, vol 3341. Springer, Berlin, Heidelberg, 2004.

D. L. Wallace. A method for comparing two hierarchical clusterings: comment. Journal of the American Statistical Association, 78(383):569–576, 1983.

S. Wang, H. Liu, H. Pu, and H. Yang. Spatial disparity and hierarchical cluster analysis of final energy consumption in china. Energy, 197:117195, 2020. ISSN 0360-5442.

D. Witten and R. Tibshirani. A framework for feature selection in clustering. Journal of the American Statistical Association, 105(490):713–726, 2010.