A Mathematical Programming Approach to Hierarchical Clustering

Lavinia Amorosi[a,*], Justo Puerto[b], Carlos Valverde[b]

[a] *Department of Statistical Sciences, Sapienza University of Rome, Italy*
[b] *Department of Statistical Sciences and Operational Research, University of Seville, Spain*

Abstract

Hierarchical clustering is a statistical technique to study the occurring groups (clusters) within a dataset creating a hierarchy of clusters. This is represented by a rooted tree (dendrogram) whose leaves correspond to the data points, and each internal node represents the cluster containing its descendant leaves. Among methods to perform hierarchical clustering, the agglomerative ones are based on greedy procedures that return a sequence of nested partitions, where each level up joins two clusters of the lower partition relying on a local criterion. In this work, motivated by the lack of exact approaches that guarantee global optimality, we focus on a unified mathematical programming formalisation that embeds single and complete linkage procedures. Through preliminary experiments, we evaluate, according to different measures commonly used in this context, the dendrograms obtained from the exact resolution of the formulations and those produced by the greedy approach. Furthermore, by exploiting the mathematical formulation, we also present a scalable matheuristic algorithm capable of generating better quality dendrograms than those produced by the greedy approach, even for large-sized datasets.

Keywords: Data science; Hierarchical clustering; Mathematical programming;

1. Introduction

Cluster analysis is a statistical technique to study the occurring groups (clusters) within a dataset. Among the different types of cluster analysis, hierarchical clustering creates nested partitions of a dataset. This is represented by a rooted tree (dendrogram) whose leaves correspond to the data points, and each internal node represents the cluster containing its descendant leaves (Everitt et al., 2011). Differently from flat partition-based clustering, like k-means clustering, this method does not require the number of clusters to be specified at the beginning. When clusters may, themselves, be closely related to other clusters, and more distantly related to others, hierarchical clustering is more appropriate than the flat-based clustering. For example, in clustering of images, we might want not just to have a cluster of flowers, but roses and tulips within that. Or, for example, in clustering patient medical records, we might want respiratory illnesses as a super-cluster of pneumonia and influenza. There are several methods to perform hierarchical clustering that can be distinguished between agglomerative and divisive (Roux, 2018). In the first case, a bottom-up approach is adopted, that starts by considering each data point as its own cluster and merging them together into larger groups from the bottom up into a single giant cluster. In the latter case, starting with one cluster including all data points, recursive divisions in clusters are performed, based on a function of the similarities or distances in the data, until all clusters are singletons. As regards the agglomerative approaches, these are based on greedy procedures that return a sequence of nested partitions where each level up joins two clusters of the lower partition. The differences among procedures in this context are related to the way in which distances between clusters are computed at each level of the dendrogram. The most commonly adopted are: (i) the minimum distance among elements of the two clusters (single linkage), (ii) the maximum distance among elements of the two clusters (complete linkage), (iii) the average distance among elements of the two clusters (average linkage), (iv) the distance between centroids of the two clusters (centroid linkage) and (v)

*Corresponding author
Email addresses: lavinia.amorosi@uniroma1.it (Lavinia Amorosi), puerto@us.es (Justo Puerto), cvalverde@us.es (Carlos Valverde)

the within-cluster deviance (Ward's method). Moreover, the distance between single data points can be defined in different ways, as, for example, those induced by the l_p norm. The greedy procedures adopted in hierarchical clustering start by defining the initial distance matrix associated with the set of all individuals of the dataset under consideration. At each iteration, the clusters at minimum distance (the individuals at minimum distance at the first iteration) are joined and the distance matrix is consequently updated. The procedure stops when all the individuals are joined in a single cluster.

In this paper, we focus on a unified mathematical programming formalization that embeds single and complete linkage procedures adopted in hierarchical clustering. The goal is to evaluate, according to different measures commonly used in this context, the dendrograms obtained from the exact solution of the formulations and those produced by the greedy approach. Furthermore, we want to study the limits in terms of the maximum size of the datasets that can be solved to certified optimality in this way. Moreover, by exploiting the mathematical formulation, we also design a scalable matheuristic algorithm capable of generating better quality dendrograms than those produced by the greedy approach even for large size datasets.

2. State of the art

In recent years, a lot of effort has been devoted to improving data science techniques by taking advantage of optimisation advances. We can find a number of contributions in different fields such as regression analysis, classification, dimensionality reduction, correspondence analysis, canonical correlation, etc. (see, e.g., Amorosi et al. (2024), Benati and García (2014), Benati et al. (2017), Benati et al. (2018), Bertsimas and Shioda (2007), Blanco et al. (2018), Blanco et al. (2022), Blanquero et al. (2020), Carrizosa et al. (2021, 2023b,a), Gambella et al. (2021), Toriello and Vielma (2012), among many others).

Our focus in this paper is on improving a specific technique, namely hierarchical clustering, still marginally studied in literature from an optimisation perspective. Indeed, we can mention few previous approaches to finding hierarchical clusters via mathematical programming (see, e.g., Burgard et al. (2023), Chen and Wu (2005); Hansen and Jaumard (1997); Labbé et al. (2023); Nielsen (2016)).

Since the area of hierarchical clustering is very broad, we cannot be comprehensive. Therefore, in the following, we only review some of the most relevant contributions in the application of mathematical programming to hierarchical clustering.

We begin our review with the seminal paper by Gower and Ross (1969). These authors first proved that the result of hierarchical clustering, using nearest neighbour (single linkage), coincides with the minimum spanning tree. Later, Hansen and Jaumard (1997) gave the first mathematical programming formulations for understanding clustering analysis under the optimisation lens. Chen and Wu (2005) propose a clustering model based on integer programming 0-1 to maximise the associations between orders within each batch in a logistic application, where small orders are consolidated, trying to achieve high-volume order processing operations. Nielsen (2016) also provides a more recent overview of mathematical programming approaches to handle clustering analysis. Gilpin and Davidson (2017) formalise hierarchical clustering as an integer linear programming (ILP) problem with a natural objective function and the dendrogram properties enforced as linear constraints. We also include the contribution by Roy and Pokutta (2017), which improves the hierarchical clustering approximation algorithm of Dasgupta (2016) using an integer linear programming formulation. Another interesting contribution on this area is the paper by Cohen-Addad et al. (2019), which develops an axiomastic approach to define good objective functions for hierarchical clustering. This methodology allows one to analyse the performance of different algorithms, as well as provides better and faster algorithms for hierarchical clustering. Labbé et al. (2023) propose different formulations that include the constraints of the minimum spanning tree problem, as well as the constraints of feature selection to jointly determine a set of features and a dendrogram, according to the single-link method. Carrizosa et al. (2023b) tackle the problem of enhancing the interpretability of the results of cluster analysis by proposing two mathematical optimisation models, inspired by classic Location Analysis problems, that differ in the way individuals are allocated to prototypes. Finally, Carrizosa et al. (2023a) propose a methodology to find a clustered table with the highest χ^2 statistic from a given contingency table and a fixed granurality.

3. Problem Description

Let $X \in \mathbb{R}^{n\times p}$ be a dataset containing n observations o_i, $i \in N := \{1, \ldots, n\}$, each with p features $o_i \in \mathbb{R}^p$. Without loss of generality, we assume that the features measured in the dataset take real values. To

quantify the similarity between these observations, we consider the Euclidean distance as a default measure, although other metrics induced by l_p norms can also be used. Let $d_{ii'}$ denote the Euclidean distance between observations o_i and $o_{i'}$.

Agglomerative hierarchical clustering aims to construct a hierarchy of nested clusters, starting with each observation as an individual cluster and progressively merging them into larger clusters until all points belong to a single cluster. Let T be the set that represents the levels at which two clusters are merged during this process. Note that $|T| < n$, since the maximum number of joins that we can make to build a hierarchy of nested clusters is $n - 1$. In our models, we assume that we perform all the joins until all points belong to a single cluster. In this work, we propose two ways to evaluate the hierarchical process. The first consists of computing the total distance, named D, between the clusters that are merged at each level $t \in T$. The second, similar to the first one, consists of computing the total distance, named D_{total}, between all clusters that exist at each level $t \in T$. Thus, setting D and D_{total} as possible objectives to minimise, results in higher similarity clusters.

To illustrate the relationships between clusters based on the hierarchy built by the agglomerative process, the most commonly used diagram is the dendrogram. Starting from the bottom, with individual data points, each merge represents two clusters combining into one. The height at which clusters merge shows how similar or different they are, a lower merge represents a higher similarity, and the opposite. In other words, the goal of this work is to generate dendrograms with the smallest height possible to obtain more similar clusters at each level.

The decision to merge two clusters is guided by a linkage criterion, which determines the distance between two clusters based on the pairwise Euclidean distances between their observations. Two of the most commonly used linkage methods in agglomerative hierarchical clustering are single linkage and complete linkage. Taking into account the measures proposed before, these methods adopt a greedy strategy: at each level t, the two closest clusters are merged. If the single-linkage clustering method is considered, the greedy approach yields the optimal solution. This optimal solution is computed by solving the Minimum Spanning Tree for the weighted complete graph whose nodes are the observations and the weight of the edges, the metric distance between any pair of observations. However, for complete linkage, the greedy approach yields suboptimal solutions, as decisions are made locally at each level without considering their impact on subsequent merges. Figures 1 and 2 show the solutions obtained with complete linkage when we consider the greedy approach and the optimal one, respectively. The reader may note that the second join, represented in the third picture, is different between the two approaches. Although 1 and 5 are closer than 1 and 4, the latter join produces a new cluster whose consequent joins are better, as it is shown in terms of the total distance produced by the joins, namely, D.

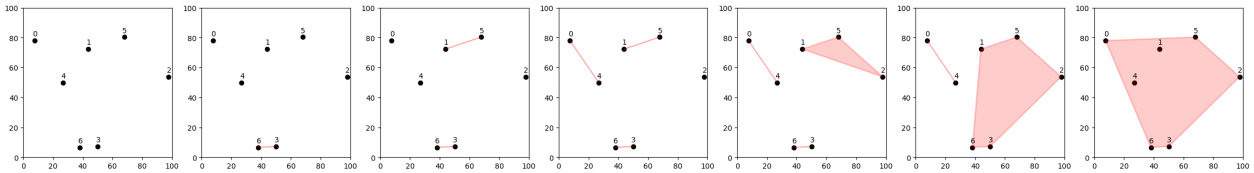

Figure 1: Sequence of joins associated with the **greedy** complete linkage approach with $D = \mathbf{301.33}$.

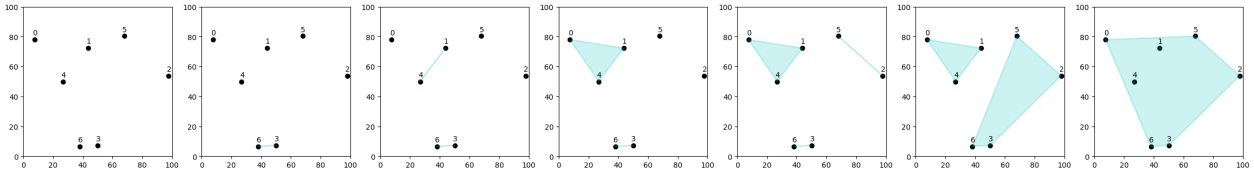

Figure 2: Sequence of joins associated with the **optimal** complete linkage approach with $D = \mathbf{289.64}$.

A widely recognized metric for evaluating clustering techniques is the Dunn Index (DI), which assesses the compactness and separation of clusters (Dunn, 1974). The goal of the Dunn Index is to identify clusters that are both compact, with low variance among members, and well-separated, with significant distance between different clusters.

The Dunn Index is defined as follows:

$$DI(k) = \frac{\min_{1\leq i<j\leq k} d(C_i, C_j)}{\max_{1\leq l\leq k} \Delta(C_l)},$$

where $d(C_i, C_j)$ represents the minimum distance between points in clusters C_i and C_j, $\Delta(C_l)$ denotes the maximum distance between any two points within cluster C_l, and k is the total number of clusters. Notably, the single linkage method prioritises minimising the numerator of the Dunn index by joining clusters with the closest pair of observations, whereas the complete linkage method focusses on minimising the denominator by controlling the cluster diameter.

Inspired by the Dunn index, we propose an extension of these linkage methods by introducing an α-weighted version that takes advantage of the strengths of both the single linkage and complete linkage approaches. This extension, named Optimal Dunn Linkage (ODL), seeks to overcome the limitations of greedy strategies by formulating the clustering process as mixed-integer linear programming problems that take into account the merges made at every hierarchical level. Furthermore, this approach incorporates the flexibility to build nested clusters. It allows one to decide on different measures to build the hierarchy or a different number of joins that are performed at each level. In this work, we assume that only two clusters are joined at each level.

Table 1 describes the parameters that serve as input to the models presented in Section 4.

Table 1: Nomenclature for the mathematical programming model

Problem Parameters

P. Name	Range	Description
n	$\mathbb{Z}_+$	Number of individuals in the dataset.
N	$\{1,\ldots,n\}$	Set of individuals in the dataset.
T	$\{1,\ldots,n-1\}$	Set of levels of the dendrogram.
$d_{ii'}$	$\mathbb{R}_+^{N\times N}$	Distance matrix containing the initial Euclidean distances between any pair of individuals in the dataset.
α	$[0,1]$	Weight factor in the Optimal Dunn Linkage formulation.

4. Mathematical Programming Formulations

In this section, we present two alternative MILP formulations for the ODL that will be compared computationally in Section 6. Firstly, we discuss in depth which are the objective functions that we are going to consider throughout the paper. Secondly, we describe a formulation whose nature is based on how the distance matrix is updated when we consider the single- or complete-linkage approach. It defines continuous variables that represent the distance between clusters for each level $t \in T$, which are updated sequentially using the distances calculated at the previous level $t-1$ and the join produced at the level $t-1$. Thirdly, we present an alternative formulation that computes the distance between clusters in an aggregated way, by controlling how clusters are built along the levels.

Evaluating the quality of a hierarchical clustering

The main goal of the models proposed in this section is to find better hierarchies of clusters by means of exact approaches. In Section 3, we briefly described what the two main criteria followed in this work are to evaluate a hierarchy. These criteria can be formalised by describing the corresponding objective functions. We define I_t as the set of existing clusters at level t, and $J_t \subseteq I_t$, the subset of clusters that are merged at level t. Let $d^t_{ii'}$ be the distance between existing clusters i and i' at level t.

The first objective function that we consider consists of minimising the overall distance of the pair of clusters that are joined at each level t:

$$D = \sum_{t\in T} \sum_{i,i'\in J_t} d^t_{ii'}.$$

This objective function allows us to focus on obtaining the best merges without taking into account the distances between the other clusters that are not involved in the joint.

Table 2: Summary of decision variables used in ODL-1

Binary and Integer Decision Variables

Name	Domain	Range	Description
x_i^t	$(i,t) \in N \times T$	$\{0,1\}$	1, if cluster i is involved in the joint at level t, 0, otherwise.
u_i^t	$(i,t) \in N \times T$	$\{0,1\}$	1, if cluster i, involved in the joint at level t, is the one with the lowest label, 0, otherwise.
$y_{ii'}^t$	$(i,i',t) \in N \times N \times T$	$\{0,1\}$	1, if clusters i and i' are merged at level t, 0, otherwise.
$\delta_{ii'i''}^{1t}$	$(i,i',i'',t) \in N \times N \times N \times T$	$\{0,1\}$	1, if the minimum distance between cluster i and i' is equal to the distance between i'' and i' at level t, 0, otherwise.
$\delta_{ii'i''}^{2t}$	$(i,i',i'',t) \in N \times N \times N \times T$	$\{0,1\}$	1, if the minimum distance between cluster i and i' is equal to the distance between i'' and i at level t, 0, otherwise.

Continuous Decision Variables

$d_{ii'}^t$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Distance between cluster i and cluster i' at level t.
$d_{\min ii'}^t$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Minimum distance between cluster i and cluster i' at level t.
$d_{\min ii'}^{0t}$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Distance between clusters i and i' when both/any of them are involved in the merge at level t.
$p_{\min ii'}^{lt}$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Auxiliary variable modelling the product of $d_{\min ii'}^t$ and x_i^t.
$p_{\min ii'}^{rt}$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Auxiliary variable modelling the product of $d_{\min ii'}^t$ and $x_{i'}^t$.
$p_{\min ii'}^{t}$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Auxiliary variable modelling the product of $d_{\min ii'}^t$ and $y_{ii'}^t$.
$d_{\min ii'}^{1t}$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Minimum distance between clusters i and i' when i is involved in the merge at level t.
$p_{\min ii'i''}^{1t}$	$(i,i',i'',t) \in N \times N \times N \times T$	$\mathbb{R}_+$	Auxiliary variable modelling the product of $d_{\min ii'}^t$ and $\delta_{ii'i''}^{1t}$.
$d_{\min ii'}^{2t}$	$(i,i',t) \in N \times N \times T$	$\mathbb{R}_+$	Minimum distance between clusters i and i' when i' is involved in the merge at level t.
$p_{\min ii'i''}^{2t}$	$(i,i',i'',t) \in N \times N \times N \times T$	$\mathbb{R}_+$	Auxiliary variable modelling the product of $d_{\min ii'}^t$ and $\delta_{ii'i''}^{2t}$.

On the other hand, the second objective function that we study, which minimises the overall distance of all the pairs of clusters existing at each level t, independently of whether they are merged or not, can be expressed as:

$$D_{total} = \sum_{t \in T} \sum_{i,i' \in I_t} d_{ii'}^t.$$

In this case, this objective function seeks to find the merge that produces clusters that are the closest after the joint.

These two objective functions are considered in the formulations presented in the following of this section. In Section 6 we provide a deep comparison and evaluation of the resulting hierarchies in terms of other commonly used measures in hierarchical clustering.

4.1. MILP based on updating the distance matrix

In this section, we present the formulation for the ODL based on the update of the distance matrix (ODL-1). Then, we set the constraints that model the process of merging clusters and the constraints that model how the distance matrices are updated. Finally, we combine all these constraints to define the exact formulation.

4.1.1. Modelling the hierarchy

In the following, we present the constraints that model the hierarchy of clusters. To build the hierarchy, we define the binary variable x_i^t that takes value 1 if the cluster i is involved in the joint at the level t. In addition, a second family of binary variables, u_i^t, is required to be introduced to ensure that the cluster with the lowest label involved at level t can not be merged in later joins. By means of these variables, the following constraints model the hierarchy of nested clusters:

$$\sum_{i \in N} x_i^t = 2, \qquad \forall t \in T, \qquad (\text{h}_1\text{-C}_1)$$

$$u_i^t \geq 1 - (1 - x_i^t) - (1 - x_{i'}^t), \qquad \forall i, i' \in N : i < i', \forall t \in T, \qquad (\text{h}_1\text{-C}_2)$$

$$u_i^t \leq x_i^t, \qquad \forall i \in N, \forall t \in T, \qquad (\text{h}_1\text{-C}_3)$$

$$1 - u_i^t \geq x_i^{t'}, \qquad \forall i \in N, \forall t, t' \in T : t' > t. \qquad (\text{h}_1\text{-C}_4)$$

Constraint (h_1-C_1) imposes that at each level the joint of two clusters must be performed. Constraints (h_1-C_2) represent the binary variables u_i^t. If cluster i is involved in the merge at level t and its label is the lowest in the set of merged clusters, then u_i^t takes the value 1. Through constraints (h_1-C_3) the value of the binary variable u_i^t is forced to be equal to 0 if cluster i is not involved in the joint performed at level t. Constraints (h_1-C_4) guarantee that if a cluster i is involved in the joint at level t and its label is the lowest one, it can no longer be involved in the joints performed in subsequent levels.

Depending on the approach adopted for updating distances between clusters after a new joint is performed, different hierarchical clustering can be obtained. In the following, we present mathematical programming constraints to represent the single and complete linkage.

4.1.2. Single linkage constraints

In single linkage, the updated distance between a cluster i not involved in the joint carried out at level t and the new cluster i' generated at level t is computed as the minimum distance between the elements of the cluster i and the elements belonging to the new generated cluster i'. Thus, it is possible to represent the distances between each pair of clusters at the following level $t+1$, represented by $d^{t+1}_{\min ii'}$. The representation considers both the distance between the clusters, represented by $d^t_{\min ii'}$, and the clusters involved in the merge produced at level t, determined by x_i^t:

$$\begin{aligned}
&d^0_{\min ii'} = d_{ii'}, && \forall i, i' \in N.\\
&d^{t+1}_{\min ii'} = d^t_{\min ii'} x_i^t x_{i'}^t + d^t_{\min ii'}(1 - x_i^t)(1 - x_{i'}^t)\\
&\qquad + \min_{i'' \neq i'} \{d^t_{\min ii'}, d^t_{\min i''i'} x_{i''}^t\} x_i^t (1 - x_{i'}^t)\\
&\qquad + \min_{i'' \neq i} \{d^t_{\min ii'}, d^t_{\min ii''} x_{i''}^t\} (1 - x_i^t) x_{i'}^t, && \forall i, i' \in N, \forall t \in T \setminus \{|T|\}. \qquad (\text{SL-G})
\end{aligned}$$

This expression states that the distance between clusters i and i' at level $t+1$ remains the same if both clusters are merged at level t or both clusters are not involved in the joint at level t. Otherwise, we need to update the distance between them by finding the other cluster i'' that is involved in the joint and taking the minimum of the distances between i'' and the other two clusters i and i'. In the following, we set the required constraints to obtain a linear form of (SL-G).

First, we introduce the auxiliary continuous variable $d^{0t}_{\min ii'}$, which represents the first two addends of (SL-G):

$$d^{0t}_{\min ii'} = d^t_{\min ii'}(1 - x_i^t)(1 - x_{i'}^t) + d^t_{\min ii'} x_i^t x_{i'}^t, \quad \forall i, i' \in N, \forall t \in T,$$

that is equivalent to the following constraint:

$$d^{0t}_{\min ii'} = d^t_{\min ii'} - d^t_{\min ii'} x_i^t - d^t_{\min ii'} x_{i'}^t + 2 d^t_{\min ii'} x_i^t x_{i'}^t, \quad \forall i, i' \in N, \forall t \in T, \qquad (d^0_{\min}\text{-C})$$

This expression leads to two products of a binary variable and a continuous variable, and another of three terms, composed of a continuous variable and two binary variables. The first two products are linearised by means of the McCormick envelopes (see McCormick (1976)). Let $p^{lt}_{\min ii'} = d^t_{\min ii'} x_i^t$ and $p^{rt}_{\min ii'} = d^t_{\min ii'} x_{i'}^t$ be the continuous variables that represent the slice products with respect to the first and second index of $d^t_{\min ii'}$, respectively. These variables are linearised as follows:

$$\begin{aligned}
&p^{lt}_{\min ii'} \leq M x_i^t, && \forall i, i' \in N, \forall t \in T, && (p^l_{\min}\text{-C}_1)\\
&p^{lt}_{\min ii'} \leq d^t_{\min ii'}, && \forall i, i' \in N, \forall t \in T, && (p^l_{\min}\text{-C}_2)\\
&p^{lt}_{\min ii'} \geq d^t_{\min ii'} - M(1 - x_i^t), && \forall i, i' \in N, \forall t \in T, && (p^l_{\min}\text{-C}_3)
\end{aligned}$$

$$\begin{aligned}
&p^{rt}_{\min ii'} \leq M x_{i'}^t, && \forall i, i' \in N, \forall t \in T, && (p^r_{\min}\text{-C}_1)\\
&p^{rt}_{\min ii'} \leq d^t_{\min ii'}, && \forall i, i' \in N, \forall t \in T, && (p^r_{\min}\text{-C}_2)\\
&p^{rt}_{\min ii'} \geq d^t_{\min ii'} - M(1 - x_{i'}^t), && \forall i, i' \in N, \forall t \in T, && (p^r_{\min}\text{-C}_3)
\end{aligned}$$

where M represents a big-M constant that is computed as the maximum distance between any pair of observations.

The product of three variables is linearised following the strategy in Speakman and Lee (2017). First, we define the binary variable $y^t_{ii'} = x^t_i x^t_{i'}$, that takes the value 1 if the clusters i and i' are merged at level t. To define $y^t_{ii'}$, we use the McCormick envelopes:

$$y^t_{ii'} \leq x^t_i, \qquad \forall i, i' \in N, \forall t \in T, \qquad (y\text{-C}_1)$$
$$y^t_{ii'} \leq x^t_{i'}, \qquad \forall i, i' \in N, \forall t \in T, \qquad (y\text{-C}_2)$$
$$y^t_{ii'} \geq x^t_i + x^t_{i'} - 1, \qquad \forall i, i' \in N, \forall t \in T. \qquad (y\text{-C}_3)$$

Then, we introduce the continuous variable $p^t_{\min\, ii'} = d^t_{\min\, ii'} y^t_{ii'}$, that is linearised in the same way:

$$p^t_{\min\, ii'} \leq M y^t_{ii'}, \qquad \forall i, i' \in N, \forall t \in T, \qquad (p_{\min}\text{-C}_1)$$
$$p^t_{\min\, ii'} \leq d^t_{\min\, ii'}, \qquad \forall i, i' \in N, \forall t \in T, \qquad (p_{\min}\text{-C}_2)$$
$$p^t_{\min\, ii'} \geq d^t_{\min\, ii'} - M(1 - y^t_{ii'}), \qquad \forall i, i' \in N, \forall t \in T. \qquad (p_{\min}\text{-C}_3)$$

Second, we introduce the auxiliary continuous variables $d^{1t}_{\min\, ii'}$ and $d^{2t}_{\min\, ii'}$, that represent the last two addends in (SL-G):

$$d^{1t}_{\min\, ii'} = \min_{i'' \neq i'} \{d^t_{\min\, ii'}, d^t_{\min\, i''i'} x^t_{i''}\} x^t_i (1 - x^t_{i'}), \qquad \forall i, i' \in N, \forall t \in T,$$
$$d^{2t}_{\min\, ii'} = \min_{i'' \neq i} \{d^t_{\min\, ii'}, d^t_{\min\, ii''} x^t_{i''}\} (1 - x^t_i) x^t_{i'}, \qquad \forall i, i' \in N, \forall t \in T.$$

They describe the minimum distance when one of the clusters is involved in the previous joint. These auxiliary variables require introducing two binary variables $\delta^{1t}_{ii'i''}$ and $\delta^{2t}_{ii'i''}$, respectively, to define the minimum. The constraints that express them are the following:

$$d^{1t}_{\min\, ii'} \leq d^t_{\min\, i''i'} + M(1 - x^t_{i''}) + M(1 - x^t_i(1 - x^t_{i'})), \qquad \forall i, i', i'' \in N : i'' \neq i', \forall t \in T, \qquad (d^1_{\min}\text{-C}_1)$$

$$d^{1t}_{\min\, ii'} \geq d^t_{\min\, i''i'} \delta^{1t}_{ii'i''} - M(1 - x^t_{i''}) - M(1 - x^t_i(1 - x^t_{i'})), \qquad \forall i, i', i'' \in N : i'' \neq i', \forall t \in T, \qquad (d^1_{\min}\text{-C}_2)$$

$$\delta^{1t}_{ii'i''} \leq x^t_{i''}, \qquad \forall i, i', i'' \in N : i'' \neq i', \forall t \in T, \qquad (d^1_{\min}\text{-C}_3)$$

$$\sum_{i'' \neq i'} \delta^{1t}_{ii'i''} = 1, \qquad \forall i, i' \in N, \forall t \in T. \qquad (d^1_{\min}\text{-C}_4)$$

$$d^{2t}_{\min\, ii'} \leq d^t_{\min\, ii''} + M(1 - x^t_{i''}) + M(1 - (1 - x^t_i) x^t_{i'}), \qquad \forall i, i', i'' \in N : i'' \neq i, \forall t \in T, \qquad (d^2_{\min}\text{-C}_1)$$

$$d^{2t}_{\min\, ii'} \geq d^t_{\min\, ii''} \delta^{2t}_{ii'i''} - M(1 - x^t_{i''}) - M(1 - (1 - x^t_i) - x^t_{i'}), \qquad \forall i, i', i'' \in N : i'' \neq i, \forall t \in T, \qquad (d^2_{\min}\text{-C}_2)$$

$$\delta^{2t}_{ii'i''} \leq x^t_{i''}, \qquad \forall i, i', i'' \in N : i'' \neq i, \forall t \in T. \qquad (d^2_{\min}\text{-C}_3)$$

$$\sum_{i'' \neq i} \delta^{2t}_{ii'i''} = 1, \qquad \forall i, i' \in N, \forall t \in T. \qquad (d^2_{\min}\text{-C}_4)$$

Constraints ($d^1_{\min}$-C$_1$), ($d^1_{\min}$-C$_2$) define the value of $d^{1t}_{\min\, ii'}$. If cluster i is involved in the merge and i' does not, to express the updated distance from i to i', we compute the minimum of the distance from cluster i to i' and the distance from cluster i and the other one involved in the join, i''. Constraints ($d^1_{\min}$-C$_3$) and

($d^1_{\min}$-C$_4$) ensure that the minimum distance is achieved in a single cluster that is involved in the merge at level t. In a similar way, $d^2_{\min}$ variables are represented by means of constraints ($d^2_{\min}$-C$_1$)-($d^2_{\min}$-C$_4$).

The representation of the minimum generates the products $p^{1t}_{\min\, ii'i''} = d^t_{\min\, ii'}\delta^{1t}_{ii'i''}$ and $p^{2t}_{\min\, ii'i''} = d^t_{\min\, ii'}\delta^{2t}_{ii'i''}$ that, again, are linearised as follows:

$$p^{1t}_{\min\, ii'i''} \leq M\delta^{1t}_{ii'i''}, \qquad \forall i,i',i'' \in N : i'' \neq i', \forall t \in T, \qquad (p^1_{\min}\text{-C}_1)$$
$$p^{1t}_{\min\, ii'i''} \leq d^t_{\min\, ii'}, \qquad \forall i,i',i'' \in N : i'' \neq i', \forall t \in T, \qquad (p^1_{\min}\text{-C}_2)$$
$$p^{1t}_{\min\, ii'i''} \geq d^t_{\min\, ii'} - M(1-\delta^{1t}_{ii'i''}), \qquad \forall i,i',i'' \in N : i'' \neq i', \forall t \in T. \qquad (p^1_{\min}\text{-C}_3)$$

$$p^{2t}_{\min\, ii'i''} \leq M\delta^{2t}_{ii'i''}, \qquad \forall i,i',i'' \in N : i'' \neq i, \forall t \in T, \qquad (p^2_{\min}\text{-C}_1)$$
$$p^{2t}_{\min\, ii'i''} \leq d^t_{\min\, ii'}, \qquad \forall i,i',i'' \in N : i'' \neq i, \forall t \in T, \qquad (p^2_{\min}\text{-C}_2)$$
$$p^{2t}_{\min\, ii'i''} \geq d^t_{\min\, ii'} - M(1-\delta^{2t}_{ii'i''}), \qquad \forall i,i',i'' \in N : i'' \neq i, \forall t \in T. \qquad (p^2_{\min}\text{-C}_3)$$

Once defined the addends in (SL-G), we set the variable $d_{\min}$ as:

$$d^{t+1}_{\min\, ii'} = d^{0t}_{\min\, ii'} + d^{1t}_{\min\, ii'} + d^{2t}_{\min\, ii'}, \quad \forall i,i' \in N, \forall t \in T \setminus \{|T|\}. \qquad (d_{\min}\text{-C})$$

4.1.3. Complete linkage constraints

In complete linkage, the updated distance between a cluster i, which is not involved in the merge performed at level t, and the newly formed cluster i at level t, is computed as the maximum distance between any element of cluster i and any element of the newly generated cluster i'. Thus, the distances $d^{t+1}_{ii'}$ between each pair of clusters at the next level $t+1$ can be expressed through the following set of constraints:

$$\begin{aligned} d^0_{\max\, ii'} &= d_{ii'}, && \forall i,i' \in N. \\ d^{t+1}_{\max\, ii'} &= d^t_{\max\, ii'}x^t_i x^t_{i'} + d^t_{\max\, ii'}(1-x^t_i)(1-x^t_{i'}) \\ &\quad + \max_{i'' \neq i'}\{d^t_{\max\, ii'}, d^t_{\max\, i''i'}x^t_{i''}\}x^t_i(1-x^t_{i'}) \\ &\quad + \max_{i'' \neq i}\{d^t_{\max\, ii'}, d^t_{\max\, ii''}x^t_{i''}\}(1-x^t_i)x^t_{i'}, && \forall i,i' \in N, \forall t \in T \setminus \{|T|\}. \end{aligned} \qquad (\text{CL-G})$$

In this case, different from the single linkage case, since both criteria consist of minimising, the maximum is attained without the necessity of defining a binary variable that selects which is the maximum one. The linearisation of (CL-G) is described as follows:

$$d^{t+1}_{\max\, ii'} \geq d^t_{\max\, ii'}, \qquad \forall i,i' \in N, \forall t \in T \setminus \{|T|\}, \qquad (d_{\max}\text{-C}_1)$$
$$d^{t+1}_{\max\, ii'} \geq d^t_{\max\, i''i'}x^t_{i''} - M(1-x^t_i(1-x^t_{i'})), \qquad \forall i,i',i'' \in N : i'' \neq i, \forall t \in T \setminus \{|T|\}, \qquad (d_{\max}\text{-C}_2)$$
$$d^{t+1}_{\max\, ii'} \geq d^t_{\max\, i''i'}x^t_{i''} - M(1-(1-x^t_i)x^t_{i'}), \qquad \forall i,i',i'' \in N : i'' \neq i', \forall t \in T \setminus \{|T|\}. \qquad (d_{\max}\text{-C}_3)$$

For each pair $i,i' \in N$ the distance $d^{t+1}_{\max\, ii'}$ at level $t+1$ must be greater than or equal to the one associated to the previous level t. This is imposed via constraints ($d_{\max}$-C$_1$). In particular, if both i and i' are not involved in the joint performed at level t, the distance between them does not change. Via constraints ($d_{\max}$-C$_2$), if cluster i is involved in the joint at level t and cluster i' is not, the new distance between cluster i and cluster i' is given by the maximum distance between $d^t_{\max\, ii'}$ and the distance between cluster i'', involved in the joint at level t, and cluster i'. Note that, because of constraint ($d_{\max}$-C$_3$), in addition to variable $x^t_{i'}$, only one variable $x^t_{i''}$ is equal to 1. If, on the contrary, cluster i is not involved in the joint at level t and cluster i' is involved, the new distance between cluster i and cluster i' is given by the maximum distance between $d^t_{\max\, ii'}$ and the distance between cluster i and cluster i'' involved in the merge at level t.

4.1.4. A first formulation for the Optimal Dunn Linkage based on updating the distance matrix

In this subsection, we put all the constraints together to propose the first formulation for the ODL.

$$\textit{minimize} \quad D = \sum_{t \in T}\sum_{i,i' \in N} d^t_{ii'}y^t_{ii'} \qquad (\text{ODL-1})$$

$$
\begin{aligned}
\text{subject to} \quad & d^t_{ii'} \geq (1-\alpha) d^t_{\min\, ii'} + \alpha d^t_{\max\, ii'}, \quad \forall i, i' \in N, \forall t \in T, \\
& (\text{h}_1\text{-C}_1) - (\text{h}_1\text{-C}_4), \\
& (d_{\min}\text{-C}), (d^0_{\min}\text{-C}), (d^1_{\min}\text{-C}_1) - (d^1_{\min}\text{-C}_4), (d^2_{\min}\text{-C}_1) - (d^2_{\min}\text{-C}_4), \\
& (p^l_{\min}\text{-C}_1) - (p^l_{\min}\text{-C}_3), (p^r_{\min}\text{-C}_1) - (p^r_{\min}\text{-C}_3), (p_{\min}\text{-C}_1) - (p_{\min}\text{-C}_3), \\
& (p^1_{\min}\text{-C}_1) - (p^1_{\min}\text{-C}_3), (p^2_{\min}\text{-C}_1) - (p^2_{\min}\text{-C}_3), \\
& (d_{\max}\text{-C}_1) - (d_{\max}\text{-C}_3), \\
& (y\text{-C}_1) - (y\text{-C}_3)
\end{aligned}
$$

The objective function accounts for the total distance between the clusters that are merged at each level $t \in T$. As exposed in Subsection 4, this objective function can be replaced by D_{total}, that considers the overall distances between any pair of clusters existing at each level t. The first constraint represents the α-weighted sum of the minimum and maximum distance between clusters i and i' at level t. The second family of constraints models the hierarchy of nested clusters. The family of constraints appearing in the third, fourth and fifth line represents the single linkage. The family of constraints in the sixth line accounts for the complete linkage. Finally, the last set of constraints linearises the product of x variables by means of the y variables.

The ODL-1 formulation gives us a first way to extend and combine the single and complete linkage approaches in a unique model. This formulation is a natural way of proceeding, since the concept of updating matrix distances is the first draft used to explain the foundations of hierarchical clustering. However, the main drawback of this formulation is the excessive use of the McCormick envelopes to linearise all the products of continuous and binary variables defined throughout the model. These linearisations yield big-M constraints that make the linear relaxation weaker. It motivates the modelling of an alternative formulation that tries to simplify the representation of the ODL in terms of the number of variables and constraints.

4.2. MILP based on clique partitioning

The combinatorial structure of the constraints of the second approach followed in this work is borrowed from a previous clustering model, namely the clique partitioning (see Bertsimas and King (2016); Bertsimas and Shioda (2007); Grötschel and Wakabayashi (1990)). However, in our model, we impose a hierarchy of nested clusters.

Table 3: Summary of decision variables used in the alternative mathematical programming model

Binary and Integer Decision Variables			
Name	**Domain**	**Range**	**Description**
$x^t_{ii'}$	$(i,i',t) \in N \times N \times T$	$\{0,1\}$	1, if observations i and i' are in the same cluster at level t, 0, otherwise.
z^t_{ic}	$(i,c,t) \in N \times N \times T$	$\{0,1\}$	1, if observation i belongs to cluster c at level t, 0, otherwise.
$y^t_{cc'}$	$(c,c',t) \in N \times N \times T$	$\{0,1\}$	1, if clusters c and c' are joint at level t, 0, otherwise.
u^t_c	$(c,t) \in N \times T$	$\{0,1\}$	1, if cluster c involved in the joint at level t is the one with lowest label, 0, otherwise.
$\delta^t_{ii'cc'}$	$(i,i',c,c',t) \in N \times N \times N \times N \times T$	$\{0,1\}$	1, if the minimum distance between cluster c and c' at level t is equal to the distance between i and i', where $i \in c$ and $i' \in c'$, 0, otherwise.
$\mu^t_{ii'cc'}$	$(i,i',c,c',t) \in N \times N \times N \times N \times T$	$\{0,1\}$	1, if i and i' belong to c and c', respectively, at level t, 0, otherwise.
$\nu^t_{ii'cc'}$	$(i,i',c,c',t) \in N \times N \times N \times N \times T$	$\{0,1\}$	1, if i and i' belong to clusters c and c' at level t and the pair (i,i') gives the minimum distance between clusters c and c', 0, otherwise.
Continuous Decision Variables			
$d^t_{cc'}$	$(c,c',t) \in N \times N \times T$	$\mathbb{R}_+$	Distance between cluster c and cluster $c^{'}$ at level t.
$d^t_{\min\, cc'}$	$(c,c',t) \in N \times N \times T$	$\mathbb{R}_+$	Minimum distance between cluster c and cluster c' at level t.
$d^t_{\max\, cc'}$	$(c,c',t) \in N \times N \times T$	$\mathbb{R}_+$	Maximum distance between cluster c and cluster c' at level t.

4.2.1. Modelling the partition

To give a formulation for the Optimal Dunn Linkage based on clique partitioning (ODL-2), we first define a partition for each level $t \in T$ in the dendrogram. Let $x^t_{ii'}$ be a binary variable that is active if observations i and i' belong to the same cluster at level t. Let z^t_{ic} also be a binary variable that indicates if observation i

belongs to cluster c at level t. Then, the following constraints represent a partition at each level $t \in T$ of the dendrogram:

$$x^t_{ii'} \leq z^t_{ic} + \sum_{\substack{c' \in N \\ c' \geq i' \\ c' \neq c}} z^t_{i'c'}, \qquad \forall i, i', c \in N : i < i', i \leq c, \forall t \in T, \qquad \text{(p-C}_1\text{)}$$

$$x^t_{ii'} \geq z^t_{ic} + z^t_{i'c} - 1, \qquad \forall i, i', c \in N : i < i' \leq c, \forall t \in T, \qquad \text{(p-C}_2\text{)}$$

$$z^t_{ic} \leq z^t_{cc}, \qquad \forall i, c \in N : i \leq c, \forall t \in T, \qquad \text{(p-C}_3\text{)}$$

$$\sum_{\substack{c \in N \\ c \geq i}} z^t_{ic} = 1, \qquad \forall i \in N, \forall t \in T, \qquad \text{(p-C}_4\text{)}$$

$$\sum_{c \in N} z^t_{cc} = |T| - t + 1, \qquad \forall t \in T. \qquad \text{(p-C}_5\text{)}$$

Constraints (p-C$_1$) ensure that if observations i and i' are in the same cluster at level t, then there is at least one representative of a cluster where both observations can be assigned. Constraints (p-C$_2$) guarantee that $x^t_{ii'}$ takes the value 1 if they are assigned to the same cluster c at level t. Constraints (p-C$_3$) state that each observation is assigned to a cluster represented by a observation i if i is assigned to this cluster. Constraints (p-C$_4$) guarantee that each observation belongs to just one cluster and this is represented by the observation with the highest index. Finally, constraints (p-C$_5$) guarantee that the number of clusters existing at level t is $|T| - t + 1$.

In the following, we model the hierarchy of clusters. Let $y^t_{cc'}$ be a binary variable that takes value equal to 1 if clusters c and c' are merged at level t. Again, a binary variable u^t_c is required to be introduced to ensure that the cluster with the highest label involved at level t cannot be merged in later levels. By means of these variables, the following constraints model the hierarchy of nested clusters:

$$\sum_{\substack{c, c' \in N \\ c < c'}} y^t_{cc'} = 1, \qquad \forall t \in T, \qquad \text{(h}_2\text{-C}_1\text{)}$$

$$z^{t+1}_{ic'} \geq y^t_{cc'} + z^t_{ic} - 1, \qquad \forall i, c, c' : i \leq c, i \leq c', c < c', \forall t \in T \setminus \{|T|\}, \qquad \text{(h}_2\text{-C}_2\text{)}$$

$$z^{t+1}_{ic'} \geq y^t_{cc'} + z^t_{ic'} - 1, \qquad \forall i, c, c' : i \leq c, i \leq c', c < c', \forall t \in \ T \setminus \{|T|\}, \qquad \text{(h}_2\text{-C}_3\text{)}$$

$$u^t_c \geq y^t_{cc'}, \qquad \forall c, c' \in N : c < c', \forall t \in T, \qquad \text{(h}_2\text{-C}_4\text{)}$$

$$1 - u^t_c \geq y^{t'}_{cc'}, \qquad \forall c, c' \in N : c < c', \forall t, t' \in T : t' > t, \qquad \text{(h}_2\text{-C}_5\text{)}$$

$$x^{t+1}_{ii'} \geq x^t_{ii'}, \qquad \forall i, i' \in N : i < i', \forall t \in T \setminus \{|T|\}. \qquad \text{(h}_2\text{-C}_6\text{)}$$

Constraints (h$_2$-C$_1$) ensure that two clusters are merged at each level $t \in T$. Constraints (h$_2$-C$_2$) and (h$_2$-C$_3$) state that if c and c' are merged at level t and the observation i belongs to any of these clusters, then i must belong to the new merged cluster represented by the highest index, c', at level $t+1$. Constraints (h$_2$-C$_4$) ensure that u^t_c is equal to 1 if cluster c is involved in a joint at level t. Constraints (h$_2$-C$_5$) impose that if a cluster c is involved in the joint at level t and its label is the lowest one, it can no longer be involved in the joints performed in subsequent levels. Finally, constraints (h$_2$-C$_6$) state that if observations i and i' belong to the same cluster at level t, then they belong to the same cluster at subsequent levels.

The benefits of representing a nested partition of clusters at each level of the hierarchy can be observed in the following sections by allowing us to define the single and complete linkage constraints in a more clever way.

4.2.2. Single linkage constraints

Let $d^t_{\min cc'}$ be a continuous variable that represents the minimum distance between cluster c and c' at level t. The representation of the partition allows us to compute this value by means of the original distance between the observations belonging to each cluster. The following constraints determine the minimum distance between any pair of clusters:

$$d^t_{\min cc'} \leq d_{ii'} + M(1 - z^t_{ic} z^t_{i'c'}), \qquad \forall i, i', c, c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T, \qquad \text{(SL}_2\text{-G)}$$

$$d^t_{\min\, cc'} \geq d_{ii'} z^t_{ic} z^t_{i'c'} \delta^t_{iicc'}, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T,$$
$$\sum_{\substack{i,i' \in N \\ i<i'}} \delta^t_{ii'cc'} = y^t_{cc'}, \qquad \forall c,c' \in N : c < c', \forall t \in T.$$

The first group of constraints states that the minimum distance between clusters c and c' at level t must be computed by the observations belonging to c and c', respectively. To represent the minimum, it is required to introduce the binary variable $\delta^t_{ii'cc'}$, which takes the value 1 if the minimum distance between c and c' is attained in $d_{ii'}$, where i belongs to cluster c and i' belongs to cluster c'. In addition, a big-M is introduced as an upper bound of the distance, that we consider to be the maximum distance between any pair of observations. The second family of constraints imposes that the minimum between clusters c and c' must be achieved in a single pair of observations (i,i') such that i belongs to c and i' belongs to c'. The reader may note that, in this case, this family of constraints can be linearised by following the same strategy as the one in ($d^0_{\min}$-C). Finally, the third group of constraints ensure that this minimum must be achieved only if clusters c and c' are merged at level t.

To linearise ($d_{\min}$-C$_2$), we first introduce the binary variable $\mu^t_{ii'cc'}$ that takes the value 1 if i and i' belong to clusters c and c' at level t, respectively. Then, $\mu^t_{ii'cc'}$ can be described using the McCormick's envelopes:

$$\mu^t_{ii'cc'} \leq z^t_{ic}, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T, \qquad (\mu\text{-C}_1)$$
$$\mu^t_{ii'cc'} \leq z^t_{i'c'}, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T, \qquad (\mu\text{-C}_2)$$
$$\mu^t_{ii'cc'} \geq z^t_{ic} + z^t_{i'c'} - 1, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T. \qquad (\mu\text{-C}_3)$$

Secondly, we define the binary variable $\nu^t_{ii'cc'}$ that is active if i and i' belong to clusters c and c' at level t and the pair (i,i') gives the minimum distance between clusters c and c'. Again, this variable linearises the product of $\mu^t_{ii'cc'}$ and $\delta^t_{ii'cc'}$ as follows:

$$\nu^t_{ii'cc'} \leq \mu^t_{ii'cc'}, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T, \qquad (\nu\text{-C}_1)$$
$$\nu^t_{ii'cc'} \leq \delta^t_{ii'cc'}, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T, \qquad (\nu\text{-C}_2)$$
$$\nu^t_{ii'cc'} \geq \mu^t_{ii'cc'} + \delta^t_{ii'cc'} - 1, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T. \qquad (\nu\text{-C}_3)$$

Once defined the products in (SL$_2$-G), we set the variable $d_{\min}$ by means of the following linear constraints:

$$d^t_{\min\, cc'} \leq d_{ii'} + M(1 - \mu^t_{ii'cc'}), \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T, \qquad (d_{\min}\text{-C}_1)$$
$$d^t_{\min\, cc'} \geq d_{ii'} \nu^t_{ii'cc'}, \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T, \qquad (d_{\min}\text{-C}_2)$$
$$\sum_{\substack{i,i' \in N \\ i<i'}} \delta^t_{ii'cc'} = y^t_{cc'}, \qquad \forall c,c' \in N : c < c', \forall t \in T. \qquad (d_{\min}\text{-C}_3)$$

4.2.3. Complete linkage constraints

Let $d^t_{\max\, cc'}$ be a continuous variable that represents the maximum distance between cluster c and c' at level t. In this case, the following constraint is the only one required to model the maximum distance between clusters:

$$d^t_{\max\, cc'} \geq d_{ii'}(z^t_{ic} + z^t_{i'c'} - 1), \qquad \forall i,i',c,c' \in N : c < c', i \leq c, i' \leq c', \forall t \in T. \qquad (d_{\max}\text{-C})$$

4.2.4. A second formulation for the Optimal Dunn Linkage based on clique partitioning

In this subsection, we put all the constraints together to propose the second formulation for the ODL.

$$\begin{aligned} \textit{minimize} \quad & D = \sum_{\substack{c,c' \in N \\ c<c'}} \sum_{t \in T} d^t_{cc'} y^t_{cc'} && \text{(ODL-2)} \\ \text{subject to} \quad & d^t_{cc'} \geq (1-\alpha) d^t_{\min\, cc'} + \alpha d^t_{\max\, cc'}, \quad \forall c,c' \in N : c < c', \forall t \in T, \\ & (\text{p-C}_1) - (\text{p-C}_5), \end{aligned}$$

$(h_2\text{-}C_1) - (h_2\text{-}C_6)$,
$(d_{\min}\text{-}C_1) - (d_{\min}\text{-}C_3)$,
$(\mu\text{-}C_1) - (\mu\text{-}C_3), (\nu\text{-}C_1) - (\nu\text{-}C_3)$,
$(d_{\max}\text{-}C)$

The objective function accounts for the total distance between the clusters that are merged at each level $t \in T$. As exposed in previous sections, this objective function can be replaced by D_{total}, that considers the overall distances between any pair of clusters existing at each level t. The first constraint represents the α-weighted sum of the minimum and maximum distance between clusters c and c' at level t. The second family of constraints models the partitioning produced at each level. The third family represents the hierarchy of nested clusters. The family of constraints appearing in the fourth and fifth lines represents the single linkage. The family of constraints in the sixth line accounts for the complete linkage.

5. Matheuristic Algorithm

In this section, we describe a matheuristic algorithm to deal with larger-size datasets. It exploits the mathematical formulations presented in Section 4 restricted to the maximum approachable size $\bar{n} = 8$, and it is based on iterative improvements of the solution (the dendrogram) provided by the greedy approach to hierarchical clustering. A sketch of the main steps of this procedure follows:

(0) Given a dataset of n individuals, compute the initial solution (dendogram) $(\bar{x}, \bar{D})$ by applying the greedy algorithm;

(1) Identify a cluster of size no lower than a given threshold $\bar{l}$ and not greater than $\bar{n}$ individuals in the solution $(\bar{x}, \bar{D})$, where $\bar{n}$ is the maximum size that can be faced by the exact solution of the formulation;

(2) Solve the complete linkage formulation on the subset of the $\bar{n}$ individuals identified at (1). Let (x^*, D^*) be the optimal solution.

(3) Build the subdendogram associated with the optimal solution (x^*, D^*) generated at (2) and save the associated order of joins;

(4) Update distances between the latest cluster generated at (3) and the remaining individuals in the original dataset;

(5) Apply the greedy algorithm to the updated dataset (with $n - \bar{n} - 1$ individuals);

(6) Repeat (1)-(5) until all individuals of the original dataset are included in a single cluster (root of the dendogram).

6. Preliminary Computational Experiments

In this section, we present preliminary results obtained by implementing the matheuristic algorithm presented in Section 5 and solving by Gurobi the mathematical formulations presented in Section 4. We tested the proposed approach on both large-sized artificial datasets and real datasets adopted also in Vichi et al. (2022). Regarding artificial datasets, we consider the hypercube $[0, 5]^{10}$ related to 10 features and select a number of vertices of the hypercube that range in the interval $[5, 10]$ to identify the centre of a multivariate normal distribution to generate an equivalent number of clusters. We generate a number of data points for each cluster in the set $\{10, 15, 20\}$ and, in order to model different levels of separation between clusters, we consider three different covariance matrices: $\Sigma = \sigma^2 I_{10}$, where $\sigma^2 \in \{1, 4, 9\}$ and I_{10} denotes the identity matrix of dimension 10. We obtain, respectively, well-separated, partially-separated, and complete-overlapped clusters. For each combination of number of clusters, number of data points per cluster, and level of separation between clusters, we generate 5 datasets. Regarding real datasets, we report their main characteristics (number of observations n, number of attributes p, and number of clusters K) in Table 4.

On both artificial and real datasets we run two different versions of the matheuristic algorithm based on the two general formulations presented in Section 4. For each version, we consider the weight values α in

Abbreviation	n	p	K
Coffee	43	12	2
Ruspini	75	2	4
Seeds	210	7	3
Thyroid	215	5	3
Wine	178	13	3

Table 4: List of real datasets

the set $\{0, 0.25, 0.5, 0.75, 1\}$. Moreover, for each formulation, we consider two different objective functions: (i) the minimisation of the overall distance of pairs of clusters that are joined at each level (D) and (ii) the minimisation of the overall distance of all pairs of clusters existing at each level, independently of whether they are merged or not (D_{total}). We set a time limit of 1 hour for each run of the matheuristic algorithm and a time limit of 1 minute for each call of Gurobi within the procedure.

We code the matheuristic and exact formulation of the model in Python 3.8.10. The mathematical programming formulation is implemented in Gurobi 10.0.2 (linux64). All tests are run on an AMD® Epyc 7452 32-Core Processor, using up to 6 threads.

Recalling that a dendrogram is a tree-like structure that represents the hierarchical relationships between clusters of data points, evaluating the goodness of a dendrogram means checking whether the order in which objects are grouped (as shown in the dendrogram) accurately reflects the distances or similarities between the original data points. To this end, we consider different well-known measures of quality that are: (i) the Cophenetic Correlation Coefficient and (ii) the Goodman-Kruskal's gamma. The Cophenetic Correlation Coefficient is a statistical measure used to evaluate the quality of a hierarchical clustering algorithm (Sokal and Rohlf (1962)) and is defined via the following formula:

$$I_C = \frac{\sum_{t\in T}\sum_{i,i'\in N:i<i'}(d_{ii'}-\bar{d})(d^t_{ii'}y^t_{ii'}-\overline{d^t})}{\sqrt{\sum_{i,i'\in N:i<i'}(d_{ii'}-\bar{d})^2\sum_{t\in T}\sum_{i,i'\in N:i<i'}(d^t_{ii'}y^t_{ii'}-\overline{d^t})^2}},$$

where $\bar{d}$ is the average initial distance between pairs of individuals in the dataset, $d^t_{ii'}y^t_{ii'}$ ($i, i' \in N$ and $t \in T$) is the cophenetic distance between individuals i and i' (height of the level t at which those two observations are first joined) and $\overline{d^t}$ is the average cophenetic distance associated with the dendrogram. It quantifies the correlation between the original distances of data points in the dataset and the cophenetic distances (heights of the dendrogram at which each pair of clusters is first joined in the same cluster). Thus, it is a measure of how faithfully the tree represents the dissimilarities among observations. It ranges between -1 and 1, with 1 indicating perfect preservation of the original distances and values closer to 0 indicating poor preservation. The Goodman-Kruskal's gamma is another measure of association that can be used to evaluate how well a dendrogram preserves the relationships between objects in the original data. Specifically, it is computed as follows:

$$I_{GK} = \frac{s^+ - s^-}{s^+ + s^-},$$

where s^+ represents the number of concordant pairs of clusters and s^- represents the number of discordant pairs of clusters. We define two pairs of clusters concordant if the relation between their original distances is preserved by the cophenetic distances and discordant otherwise.

As for the Cophenetic Correlation Coefficient, its value ranges between -1 and 1. A value equal to 1 indicates that the dendrogram perfectly preserves the original order of distances. A value equal to 0 means that the dendrogram does not provide a meaningful representation of the original relationships. A value equal to -1 shows that the dendrogram completely misrepresents the original relationships.

Table 5 reports the main results related to the artificial datasets obtained by comparing the matheuristic algorithm (ODL-1) and (ODL-2), based on the two general formulations presented in Section 4, and the greedy algorithm (greedy). For each level of separation between clusters (variance), for each value of the weight α and for each solution strategy, we report the average and standard deviation of, respectively, the computational time, the Cophenetic Correlation Coefficient and the Goodman-Kruskal's gamma. We always distinguish between the two different objective functions (D and D_{total}). Note that we report aggregated

data, with respect to the number of clusters and number of data points, because we observed that they do not have a significant impact on the behaviour of the different solution strategies. This can be seen, for example, in Figure 3. It shows the boxplots of the running time for each solution strategy and number of clusters distinguishing between the two objective functions. From them we can observe that the relations between the solution strategies do not change when the number of clusters. The greedy is, as expected, the fastest one, followed by the matheuristic based on the second general formulation and the matheuristic based on the first general formulation.

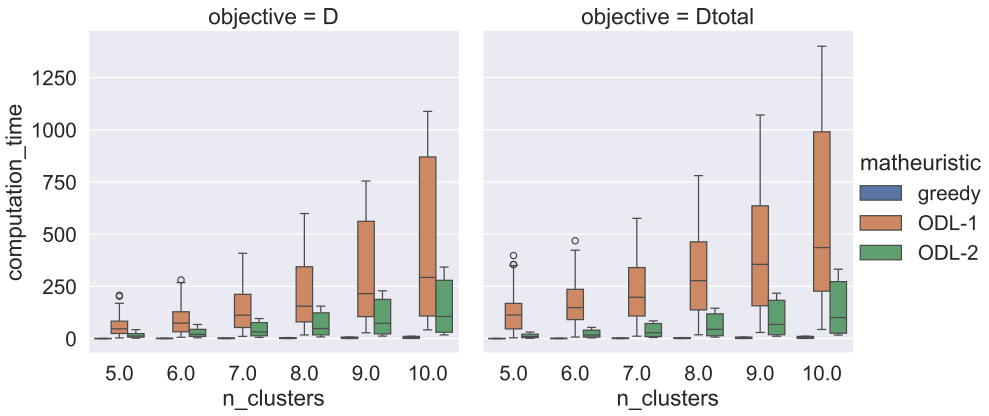

Figure 3: Boxplots of the computational time for artificial datasets

From both Table 5 and Figure 3 we can observe that the adoption of the second general formulation significantly improves the matheuristic running time for both objective functions. For the formulation where the overall distance is minimised (D_{total}) the decrease in running time is even more significant with a maximum saving in average running time, with respect to the matheuristic based on the first formulation, equal to 83.6%. As regards the two measures of quality adopted to evaluate the dendrograms associated with the solutions, we can see that, although the differences are not huge, the average values of the Cophenetic Correlation Coefficient associated with the matheuristic algorithm is always better than the one associated with the greedy algorithm, with only few exceptions. In particular, we can observe that the adoption of the matheuristic approach based on the second general formulation (ODL-2) and the first objective function (D) produces the best results in terms of the average values of this quality measure. This can also be better seen from Figure 4 showing the boxplots of the Cophenetic Correlation Coefficient for the different values of the weight α and for the two objective functions (D and D_{total}) distinguishing between the different solution approaches: the greedy algorithm and the matheuristic algorithm based on the two general formulations proposed (ODL-1) and (ODL-2). The same observations can be made for the Goodman-Kruskal's gamma in Table 5 and Figure 5. This permits us to conclude that, relying on the results obtained from the artificial instances, the proposed matheuristic algorithm based on the second formulation represents a good compromise between running time and quality of the solution obtained. In addition, the reader may observe in Figures 4 and 5 that, in general, the values of I_C and I_{GK} are higher for intermediate values of α. All these results justify the definition of the model detailed in Section 4.

Similarly, Table 6 and Table 7 report the main results related to real datasets. For each solution strategy, the weight value α and the objective function (D or D_{total}), it shows the running time (Time) And the values of the two measures of quality adopted: the Cophenetic Correlation Coefficient (I_C) and the Goodman-Kruskal's gamma (I_{GK}). We can observe that on real datasets the improvement in terms of running time derived from the adoption of the matheuristic algorithm based on the second general formulation is even more significant. In particular for the biggest size datasets Seeds and Thyroid, the decrease in running time is by an order of magnitude. This is particularly evident by looking at Figure 6 which reports the boxplots of the computational time for real datasets, always distinguishing between solutions strategies and objective functions.

As regards the two measures of quality, different from the results obtained for the artificial datasets, they show a more fluctuating behaviour both with respect to the objective function and the matheuristic version

Table 5: Comparison between matheuristic and greedy algorithm on artificial datasets

			Time				I_C				I_{GK}			
		objective	mean		std		mean		std		mean		std	
Variance	alpha	algorithm	D	D_{total}	D	D_{total}	D	D_{total}	D	D_{total}	D	D_{total}	D	D_{total}
1	0	greedy	2.358	2.358	2.854	2.853	0.828	0.828	0.054	0.054	0.583	0.583	0.163	0.163
		ODL-1	218.152	297.538	239.459	264.674	0.757	0.572	0.117	0.189	0.515	0.418	0.169	0.179
		ODL-2	53.146	51.337	63.511	62.957	0.759	0.74	0.103	0.129	0.527	0.517	0.161	0.177
	0.25	greedy	2.351	2.366	2.837	2.871	0.832	0.832	0.036	0.036	0.576	0.576	0.118	0.118
		ODL-1	236.028	352.202	248.058	300.899	0.848	0.661	0.039	0.147	0.624	0.481	0.134	0.167
		ODL-2	63.6	59.385	73.279	71.671	0.848	0.732	0.036	0.105	0.624	0.498	0.116	0.142
	0.5	greedy	2.35	2.349	2.842	2.844	0.832	0.832	0.038	0.038	0.556	0.556	0.131	0.131
		ODL-1	212.585	339.801	237.66	284.366	0.855	0.766	0.033	0.072	0.659	0.55	0.092	0.137
		ODL-2	67.7	62.092	77.804	75.134	0.851	0.79	0.036	0.053	0.644	0.546	0.102	0.121
	0.75	greedy	2.346	2.347	2.827	2.826	0.828	0.828	0.041	0.041	0.55	0.55	0.125	0.125
		ODL-1	203.23	336.705	234.485	293.652	0.854	0.837	0.031	0.036	0.663	0.634	0.086	0.091
		ODL-2	71.379	64.81	81.752	79	0.856	0.823	0.032	0.036	0.664	0.59	0.091	0.094
	1	greedy	2.347	2.35	2.829	2.829	0.822	0.822	0.042	0.042	0.547	0.547	0.129	0.129
		ODL-1	150.733	152.46	218.168	218.006	0.845	0.845	0.037	0.037	0.644	0.625	0.104	0.11
		ODL-2	65.353	61.205	80.897	79.649	0.85	0.812	0.035	0.039	0.66	0.574	0.097	0.108
4	0	greedy	2.348	2.353	2.852	2.862	0.437	0.437	0.077	0.077	0.292	0.292	0.067	0.067
		ODL-1	213.611	304.899	226.331	262.656	0.415	0.362	0.076	0.093	0.28	0.252	0.067	0.074
		ODL-2	49.967	48.238	59.633	59.093	0.416	0.413	0.077	0.077	0.281	0.277	0.068	0.066
	0.25	greedy	2.331	2.343	2.82	2.827	0.532	0.532	0.069	0.069	0.372	0.372	0.087	0.087
		ODL-1	238.669	364.421	245.666	294.686	0.572	0.53	0.062	0.074	0.421	0.384	0.077	0.088
		ODL-2	63.83	59.751	72.37	71.071	0.574	0.547	0.061	0.064	0.418	0.381	0.08	0.087
	0.5	greedy	2.33	2.323	2.816	2.814	0.522	0.522	0.067	0.067	0.365	0.365	0.087	0.087
		ODL-1	214.337	347.944	236.601	293.376	0.583	0.557	0.053	0.062	0.423	0.399	0.078	0.078
		ODL-2	69.009	63.517	79.75	77.275	0.585	0.568	0.049	0.049	0.431	0.4	0.076	0.075
	0.75	greedy	2.323	2.333	2.801	2.834	0.506	0.506	0.08	0.08	0.345	0.345	0.092	0.092
		ODL-1	201.893	330.636	228.041	274.546	0.588	0.581	0.048	0.053	0.43	0.418	0.074	0.08
		ODL-2	71.75	65.485	81.226	79.13	0.592	0.579	0.052	0.051	0.443	0.41	0.08	0.079
	1	greedy	2.316	2.33	2.788	2.821	0.475	0.475	0.076	0.076	0.314	0.314	0.086	0.086
		ODL-1	149.242	151.294	215.98	215.608	0.581	0.581	0.05	0.048	0.44	0.43	0.069	0.066
		ODL-2	65.983	61.764	81.463	79.958	0.588	0.57	0.05	0.047	0.445	0.4	0.07	0.069
9	0	greedy	2.336	2.323	2.823	2.801	0.466	0.466	0.055	0.055	0.311	0.311	0.043	0.043
		ODL-1	195.435	279.144	222.702	256.433	0.44	0.406	0.061	0.073	0.296	0.278	0.045	0.048
		ODL-2	47.646	46.184	57.135	56.832	0.442	0.436	0.064	0.062	0.297	0.294	0.046	0.043
	0.25	greedy	2.303	2.292	2.778	2.761	0.442	0.442	0.064	0.064	0.292	0.292	0.057	0.057
		ODL-1	235.023	359.962	244.15	291.249	0.475	0.434	0.06	0.073	0.337	0.304	0.061	0.073
		ODL-2	64.228	60.162	72.576	71.439	0.476	0.457	0.052	0.052	0.343	0.323	0.051	0.055
	0.5	greedy	2.293	2.29	2.774	2.758	0.394	0.394	0.069	0.069	0.258	0.258	0.062	0.062
		ODL-1	218.297	352.786	239.444	290.695	0.455	0.44	0.059	0.06	0.311	0.305	0.067	0.066
		ODL-2	69.853	63.906	77.754	76.184	0.463	0.457	0.055	0.058	0.326	0.313	0.061	0.064
	0.75	greedy	2.285	2.289	2.754	2.762	0.362	0.362	0.069	0.069	0.229	0.229	0.067	0.067
		ODL-1	202.465	340.802	229.944	284.71	0.445	0.449	0.059	0.063	0.311	0.311	0.067	0.075
		ODL-2	72.581	66.012	82.099	79.918	0.449	0.452	0.051	0.052	0.313	0.306	0.059	0.065
	1	greedy	2.285	2.283	2.766	2.751	0.327	0.327	0.071	0.071	0.206	0.206	0.063	0.063
		ODL-1	149.016	151.174	213.939	213.943	0.416	0.421	0.06	0.06	0.283	0.287	0.069	0.072
		ODL-2	66.173	62.024	81.49	80.042	0.42	0.415	0.053	0.053	0.289	0.269	0.06	0.064

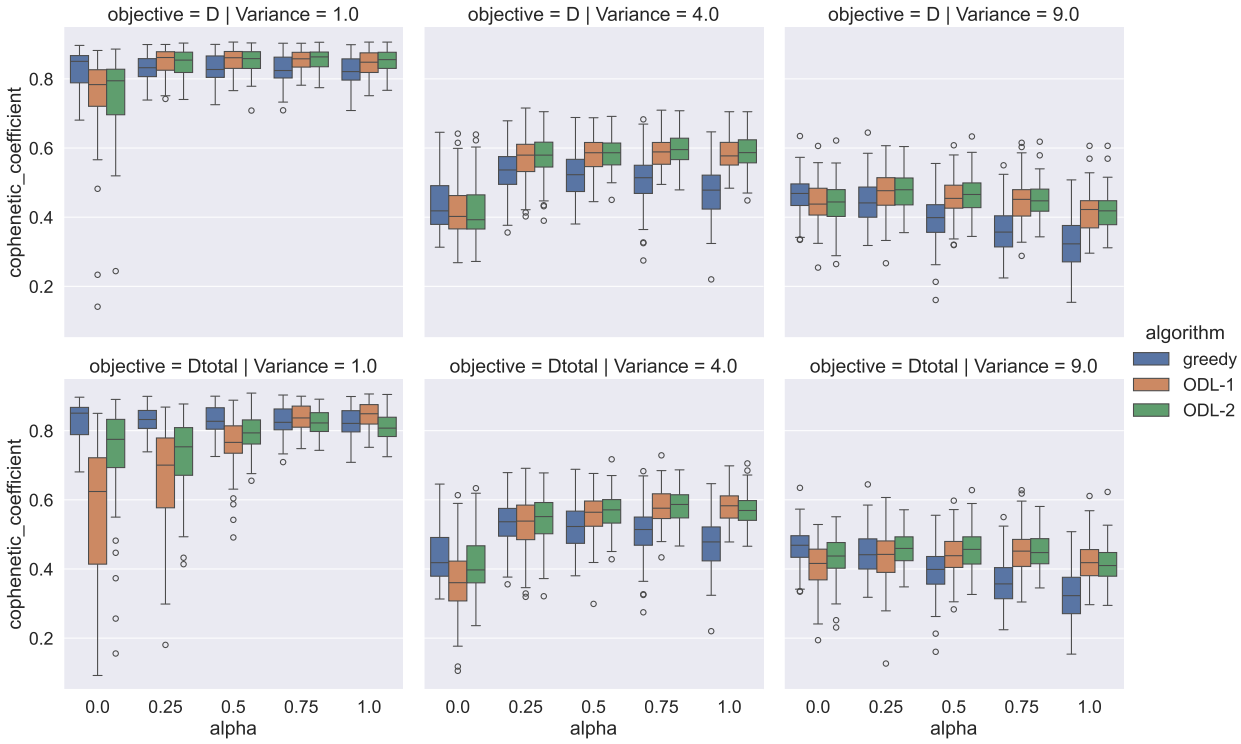

Figure 4: Boxplots of the cophenetic correlation coefficient for artificial datasets

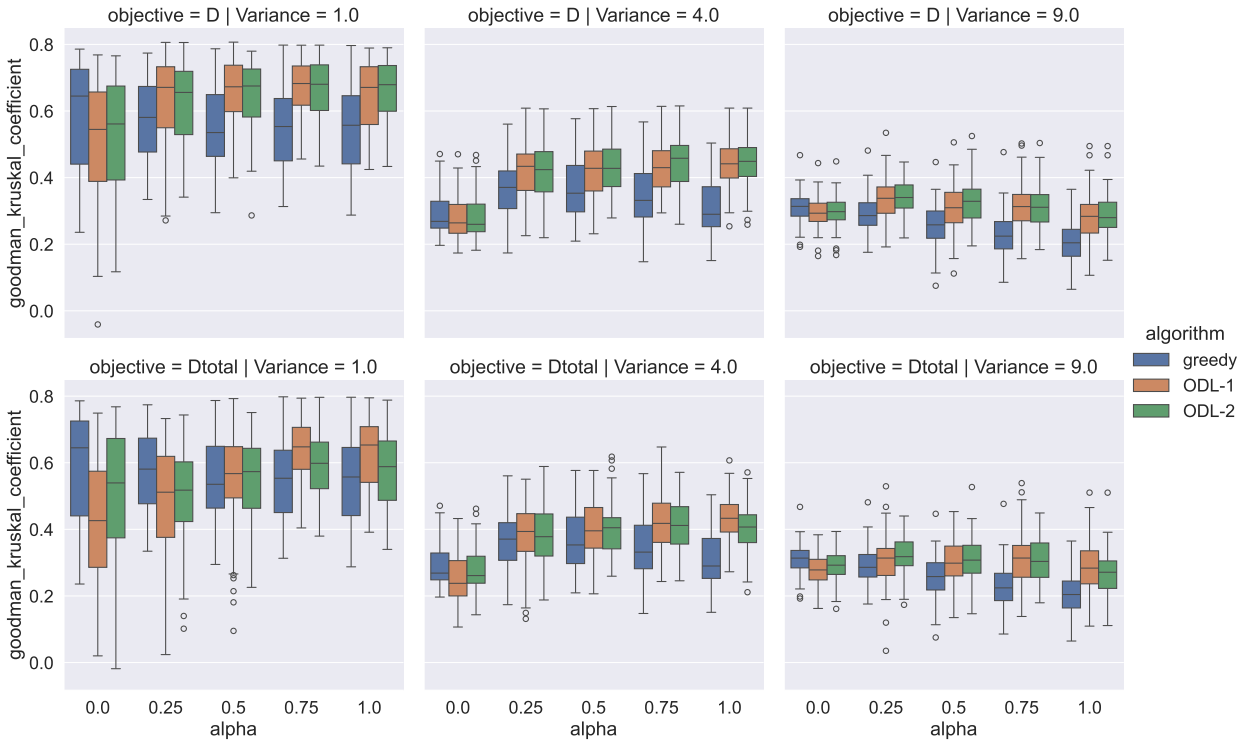

Figure 5: Boxplots of the Goodman-Kruskal correlation coefficient for artificial datasets

adopted. However, focusing on the cases where the weight $\alpha = 1$, that is on the complete linkage approach,

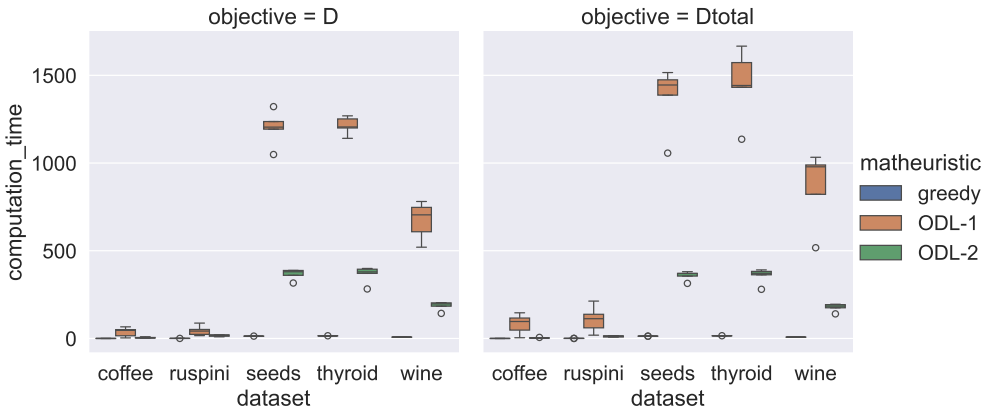

Figure 6: Boxplots of the computational time for real datasets

we can observe that the value of at least one of the two measures associated with the proposed approach is always better than the one associated with the greedy algorithm. In particular, we can see a significant difference in both values of the Cophenetic Correlation Coefficient and Goodman-Kruskal's gamma for the Seed dataset and the Wine dataset. This can be better visualised from the boxplots reported in Figure 7 and Figure 8. From these preliminary results on real datasets, we can conclude that the proposed approach represents a good alternative to the greedy algorithm to build dendrograms based on complete linkage.

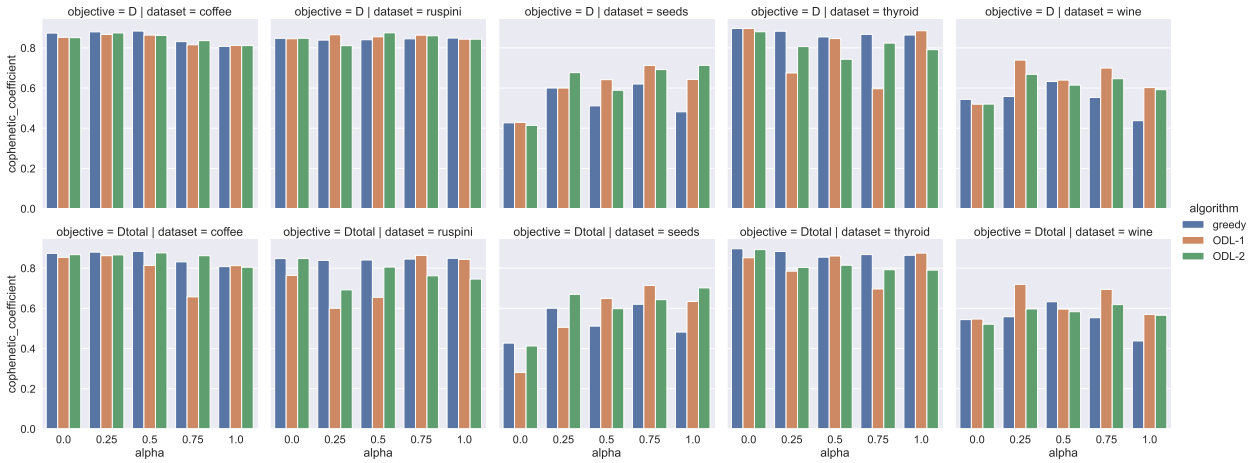

Figure 7: Boxplots of the cophenetic correlation coefficient for real datasets

7. Conclusions

In this study, we introduce a novel linkage method that combines the single and complete linkage methods through a weighted linear combination of them. We propose two distinct mathematical programming formulations to address the hierarchical clustering problem. The first formulation updates the distance matrix between clusters iteratively, while the second one relies on a clique partitioning approach. These formulations enable the exact solution of small-to-medium size datasets and serve as the foundation for matheuristic solution procedures aimed at handling larger datasets.

The proposed solution strategies are evaluated using benchmark datasets from the literature. The computational experiments demonstrate that the proposed approaches yield high-quality hierarchical clusterings, as measured by the cophenetic and Goodman-Kruskal coefficients. Remarkably, the results confirm that it is feasible to solve the Optimal Dunn Linkage model for small datasets using the Gurobi solver. For larger

Table 6: Comparison between matheuristic and greedy algorithm on real datasets

dataset	alpha	objective algorithm	Time D	Time D_{total}	I_C D	I_C D_{total}	I_{GK} D	I_{GK} D_{total}
Coffee	0	greedy	0.058	0.058	0.874	0.874	0.686	0.686
		ODL-1	66.294	96.675	0.852	0.854	0.652	0.659
		ODL-2	2.706	2.191	0.851	0.867	0.647	0.656
	0.25	greedy	0.058	0.057	0.88	0.88	0.767	0.767
		ODL-1	15.404	47.711	0.866	0.862	0.714	0.709
		ODL-2	9.917	6.039	0.874	0.866	0.742	0.653
	0.5	greedy	0.057	0.058	0.884	0.884	0.772	0.772
		ODL-1	46.11	146.291	0.863	0.813	0.68	0.68
		ODL-2	5.592	3.554	0.862	0.876	0.737	0.749
	0.75	greedy	0.057	0.057	0.831	0.831	0.742	0.742
		ODL-1	50.401	116.429	0.815	0.657	0.655	0.633
		ODL-2	2.567	2.057	0.836	0.862	0.736	0.696
	1	greedy	0.058	0.057	0.807	0.807	0.708	0.708
		ODL-1	3.133	4.111	0.812	0.812	0.659	0.657
		ODL-2	1.842	1.225	0.811	0.804	0.627	0.625
Ruspini	0	greedy	0.381	0.367	0.848	0.848	0.736	0.736
		ODL-1	23.218	60.621	0.845	0.764	0.734	0.546
		ODL-2	9.519	8.345	0.848	0.848	0.731	0.733
	0.25	greedy	0.351	0.363	0.839	0.839	0.614	0.614
		ODL-1	87.613	213.026	0.865	0.6	0.748	0.337
		ODL-2	20.569	16.087	0.811	0.692	0.513	0.426
	0.5	greedy	0.354	0.357	0.84	0.84	0.611	0.611
		ODL-1	42.072	137.844	0.855	0.655	0.63	0.364
		ODL-2	20.24	14.952	0.875	0.805	0.752	0.713
	0.75	greedy	0.362	0.363	0.845	0.845	0.613	0.613
		ODL-1	51.367	112.739	0.863	0.863	0.717	0.715
		ODL-2	19.301	13.799	0.86	0.762	0.716	0.531
	1	greedy	0.354	0.363	0.849	0.849	0.612	0.612
		ODL-1	16.056	18.798	0.843	0.844	0.712	0.713
		ODL-2	12.862	7.705	0.843	0.745	0.711	0.5
Seeds	0	greedy	13.466	13.027	0.427	0.427	0.414	0.414
		ODL-1	1321.545	1474.331	0.429	0.28	0.409	0.299
		ODL-2	316.078	314.049	0.414	0.412	0.402	0.396
	0.25	greedy	12.906	13.065	0.6	0.6	0.635	0.635
		ODL-1	1236.58	1444.856	0.6	0.505	0.531	0.451
		ODL-2	359.925	355.65	0.677	0.669	0.601	0.594
	0.5	greedy	13.074	13.089	0.511	0.511	0.362	0.362
		ODL-1	1204.725	1387.036	0.642	0.649	0.612	0.604
		ODL-2	380.322	370.476	0.589	0.598	0.536	0.535
	0.75	greedy	12.823	12.794	0.62	0.62	0.496	0.496
		ODL-1	1193.169	1515.758	0.713	0.714	0.67	0.671
		ODL-2	388.083	371.005	0.692	0.643	0.678	0.587
	1	greedy	12.839	13.333	0.482	0.482	0.343	0.343
		ODL-1	1048.527	1056.598	0.643	0.634	0.574	0.557
		ODL-2	387.568	380.59	0.713	0.702	0.678	0.641

Table 7: Comparison between matheuristic and greedy algorithm on artificial datasets (continue)

dataset	alpha	objective algorithm	Time D	Time D_{total}	I_C D	I_C D_{total}	I_{GK} D	I_{GK} D_{total}
Thyroid	0	greedy	15.389	15.624	0.896	0.896	0.712	0.712
		ODL-1	1199.864	1431.597	0.896	0.851	0.701	0.696
		ODL-2	282.21	280.179	0.88	0.892	0.695	0.703
	0.25	greedy	15.102	15.293	0.883	0.883	0.74	0.74
		ODL-1	1251.463	1442.071	0.675	0.785	0.663	0.724
		ODL-2	370.978	362.947	0.806	0.804	0.727	0.688
	0.5	greedy	14.967	15.205	0.854	0.854	0.637	0.637
		ODL-1	1269.044	1572.623	0.846	0.86	0.743	0.797
		ODL-2	379.513	371.344	0.743	0.814	0.691	0.681
	0.75	greedy	14.973	15.34	0.867	0.867	0.702	0.702
		ODL-1	1205.633	1666.316	0.597	0.696	0.624	0.643
		ODL-2	394.817	382.329	0.824	0.792	0.717	0.703
	1	greedy	14.913	15.27	0.864	0.864	0.643	0.643
		ODL-1	1140.848	1135.71	0.885	0.875	0.742	0.747
		ODL-2	399.121	390.906	0.791	0.79	0.693	0.717
Wine	0	greedy	7.431	7.579	0.544	0.544	0.442	0.442
		ODL-1	780.801	978.498	0.519	0.546	0.415	0.434
		ODL-2	142.996	140.217	0.52	0.521	0.417	0.42
	0.25	greedy	7.438	7.368	0.558	0.558	0.456	0.456
		ODL-1	747.199	1032.87	0.739	0.719	0.641	0.621
		ODL-2	184.037	174.83	0.668	0.597	0.582	0.55
	0.5	greedy	7.208	7.626	0.633	0.633	0.465	0.465
		ODL-1	704.659	989.004	0.639	0.596	0.449	0.391
		ODL-2	197.141	184.595	0.614	0.583	0.537	0.424
	0.75	greedy	7.646	7.352	0.553	0.553	0.396	0.396
		ODL-1	608.345	821.564	0.699	0.694	0.587	0.586
		ODL-2	204.234	192.849	0.647	0.618	0.531	0.494
	1	greedy	7.612	7.418	0.437	0.437	0.287	0.287
		ODL-1	520.319	517.202	0.602	0.569	0.437	0.319
		ODL-2	202.889	195.048	0.592	0.565	0.423	0.413

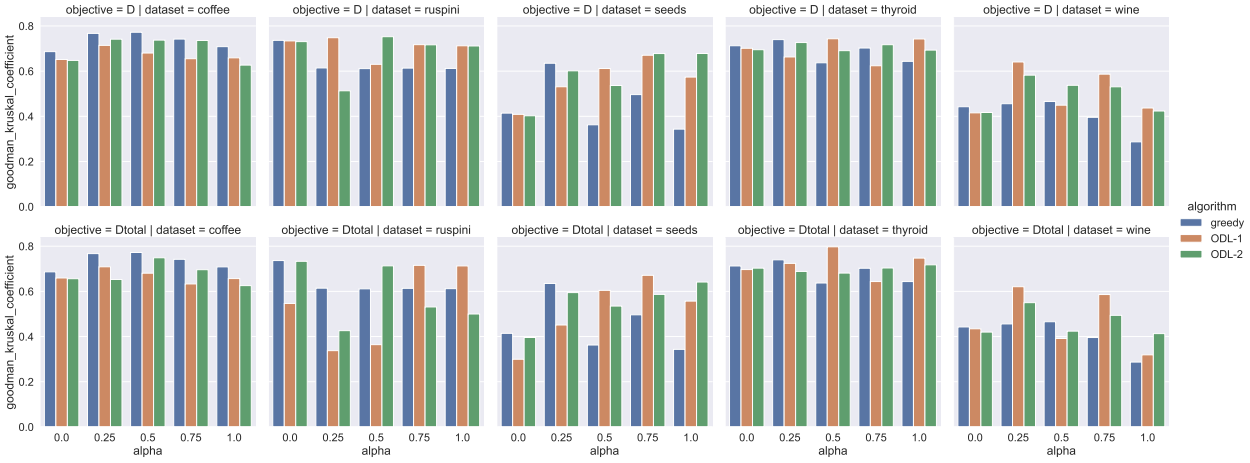

Figure 8: Boxplots of the Goodman-Kruskal correlation coefficient for real datasets

datasets, alternative strategies are required, but leveraging the mathematical programming formulations allows for the development of algorithms that achieve high-quality solutions, often surpassing the performance of traditional greedy methods. In particular, the matheuristic approach based on the clique partitioning formulation proves highly effective for medium-sized datasets, achieving the best values for both cophenetic and Goodman-Kruskal coefficients.

The formulations and algorithms presented in this work provide a foundational step toward more sophisticated models of hierarchical clustering. Future research should focus on developing faster and more accurate algorithms to tackle larger instances. Potential extensions of this work include the exploration of merging

more than two clusters at each hierarchical level, enhancing the flexibility of dendrogram representations; the development of new metrics to evaluate dendrogram quality; and the integration of multi-objective optimization techniques to account for multiple evaluation criteria. These extensions represent promising directions for future research and will be the focus of subsequent studies.

References

Amorosi, L., Padellini, T., Puerto, J., and Valverde, C. (2024). A Mathematical Programming Approach to Sparse Canonical Correlation Analysis. *Expert Systems with Applications*, *237*, 121293. doi:10.1016/j.eswa.2023.121293.

Benati, S., and García, S. (2014). A mixed integer linear model for clustering with variable selection. *Computers & Operations Research*, *43*, 280–285. doi:10.1016/j.cor.2013.10.005.

Benati, S., García, S., and Puerto, J. (2018). Mixed integer linear programming and heuristic methods for feature selection in clustering. *Journal of the Operational Research Society*, *69*, 1379–1395. doi:10.1080/01605682.2017.1398206.

Benati, S., Puerto, J., and Rodríguez-Chía, A. M. (2017). Clustering data that are graph connected. *European Journal of Operational Research*, *261*, 43–53. doi:10.1016/j.ejor.2017.02.009.

Bertsimas, D., and King, A. (2016). OR Forum—An Algorithmic Approach to Linear Regression. *Operations Research*, *64*, 2–16. doi:10.1287/opre.2015.1436.

Bertsimas, D., and Shioda, R. (2007). Classification and Regression via Integer Optimization. *Operations Research*, *55*, 252–271. doi:10.1287/opre.1060.0360.

Blanco, V., Japón, A., and Puerto, J. (2022). A mathematical programming approach to SVM-based classification with label noise. *Computers & Industrial Engineering*, *172*, 108611. doi:10.1016/j.cie.2022.108611.

Blanco, V., Puerto, J., and Salmerón, R. (2018). Locating hyperplanes to fitting set of points: A general framework. *Computers & Operations Research*, *95*, 172–193. doi:10.1016/j.cor.2018.03.009.

Blanquero, R., Carrizosa, E., Molero-Río, C., and Romero Morales, D. (2020). Sparsity in optimal randomized classification trees. *European Journal of Operational Research*, *284*, 255–272. doi:10.1016/j.ejor.2019.12.002.

Burgard, J. P., Moreira Costa, C., Hojny, C., Kleinert, T., and Schmidt, M. (2023). Mixed-integer programming techniques for the minimum sum-of-squares clustering problem. *Journal of Global Optimization*, *87*, 133–189. doi:10.1007/s10898-022-01267-4.

Carrizosa, E., Guerrero, V., and Romero Morales, D. (2023a). On mathematical optimization for clustering categories in contingency tables. *Advances in Data Analysis and Classification*, *17*, 407–429. doi:10.1007/s11634-022-00508-4.

Carrizosa, E., Kurishchenko, K., Marín, A., and Romero Morales, D. (2023b). On clustering and interpreting with rules by means of mathematical optimization. *Computers & Operations Research*, *154*, 106180. doi:10.1016/j.cor.2023.106180.

Carrizosa, E., Molero-Río, C., and Romero Morales, D. (2021). Mathematical optimization in classification and regression trees. *TOP*, *29*, 5–33. doi:10.1007/s11750-021-00594-1.

Chen, M.-C., and Wu, H.-P. (2005). An association-based clustering approach to order batching considering customer demand patterns. *Omega*, *33*, 333–343. doi:10.1016/j.omega.2004.05.003.

Cohen-Addad, V., Kanade, V., Mallmann-trenn, F., and Mathieu, C. (2019). Hierarchical Clustering: Objective Functions and Algorithms. *J. ACM*, *66*, 26:1–26:42. doi:10.1145/3321386.

Dasgupta, S. (2016). A cost function for similarity-based hierarchical clustering. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing* STOC '16 (pp. 118–127). New York, NY, USA: Association for Computing Machinery. doi:10.1145/2897518.2897527.

Dunn, J. C. (1974). Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*, *4*, 95–104. doi:10.1080/01969727408546059.

Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Chichester: Wiley.

Gambella, C., Ghaddar, B., and Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, *290*, 807–828. doi:10.1016/j.ejor.2020.08.045.

Gilpin, S., and Davidson, I. (2017). A flexible ILP formulation for hierarchical clustering. *Artificial Intelligence*, *244*, 95–109. doi:10.1016/j.artint.2015.05.009.

Gower, J. C., and Ross, G. J. S. (1969). Minimum Spanning Trees and Single Linkage Cluster Analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *18*, 54–64. doi:10.2307/2346439.

Grötschel, M., and Wakabayashi, Y. (1990). Facets of the clique partitioning polytope. *Mathematical Programming*, *47*, 367–387. doi:10.1007/BF01580870.

Hansen, P., and Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming*, *79*, 191–215. doi:10.1007/BF02614317.

Labbé, M., Landete, M., and Leal, M. (2023). Dendrograms, minimum spanning trees and feature selection. *European Journal of Operational Research*, *308*, 555–567. doi:10.1016/j.ejor.2022.11.031.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, *10*, 147–175. doi:10.1007/BF01580665.

Nielsen, F. (2016). Hierarchical Clustering. In F. Nielsen (Ed.), *Introduction to HPC with MPI for Data Science* (pp. 195–211). Cham: Springer International Publishing. doi:10.1007/978-3-319-21903-5_8.

Roux, M. (2018). A Comparative Study of Divisive and Agglomerative Hierarchical Clustering Algorithms. *Journal of Classification*, *35*, 345–366. doi:10.1007/s00357-018-9259-9.

Roy, A., and Pokutta, S. (2017). Hierarchical Clustering via Spreading Metrics. *Journal of Machine Learning Research*, *18*, 1–35.

Sokal, R. R., and Rohlf, F. J. (1962). The Comparison of Dendrograms by Objective Methods. *Taxon*, *11*, 33–40. doi:10.2307/1217208.

Speakman, E., and Lee, J. (2017). Quantifying Double McCormick. *Mathematics of Operations Research*, *42*, 1230–1253. doi:10.1287/moor.2017.0846.

Toriello, A., and Vielma, J. P. (2012). Fitting piecewise linear continuous functions. *European Journal of Operational Research*, *219*, 86–95. doi:10.1016/j.ejor.2011.12.030.

Vichi, M., Cavicchia, C., and Groenen, P. J. F. (2022). Hierarchical Means Clustering. *Journal of Classification*, *39*, 553–577. doi:10.1007/s00357-022-09419-7.