UNIVERSIDAD
TORCUATO DI TELLA

Típo de documento: Preprint

An integer programming approach for the hyper-rectangular clustering problem with axis-parallel clusters and outliers

Autoría ditelliana: Marenco, Javier

Fecha de publicación: 2023

Publicado como artículo en: Discrete Applied Mathematics, 341, 180–195 //ISSN-e: 1872-6771

¿Cómo citar este artículo?

Marenco, J. (2023).*An integer programming approach for the hyper-rectangular clustering problem with axis-parallel clusters and outliers. Preprint. Repositorio Digital Universidad Torcuato Di Tella. https://repositorio.utdt.edu/handle/20.500.13098/12149*

El presente documento se encuentra alojado en el Repositorio Digital de la Universidad Torcuato Di Tella bajo una licencia Atribución-NoComercial-CompartirIgual 4.0 Internacional CC BY-NC-SA 4.0

Dirección: https://repositorio.utdt.edu

Biblioteca Di Tella

An integer programming approach for the hyper-rectangular clustering problem with axis-parallel clusters and outliers

Javier Marenco

Business School, Universidad Torcuato Di Tella, Argentina
javier.marenco@utdt.edu

April 20, 2023

Abstract

We present a mixed integer programming formulation for the problem of clustering a set of points in $\mathbb{R}^d$ with axis-parallel clusters, while allowing to discard a pre-specified number of points, thus declared to be outliers. We identify a family of valid inequalities separable in polynomial time, we prove that some inequalities from this family induce facets of the associated polytope, and we show that the dynamic addition of cuts coming from this family is effective in practice.
KEYWORDS: clustering, integer programming, facets, cutting planes

1 Introduction

Given a nonempty set $\mathcal{X} = \{x^1, \dots, x^n\}$ of n points in $\mathbb{R}^d$ and an integer $p \geq 1$, the *hyper-rectangular clustering problem with axis-parallel clusters* consists in determining the p "smallest" axis-parallel hyper-rectangles $\mathbb{R}^d$ such that each point in $\mathcal{X}$ is included in at least one such hyper-rectangle. If we also specify a number $q \geq 0$ of possible *outliers*, then up to q points may be discarded and not be included in any hyper-rectangle.

Alternatively, we may consider a feasible solution to be a partition of a subset $\mathcal{X}' \subseteq \mathcal{X}$ of points with cardinality $|\mathcal{X}'| \geq n - q$ into subsets $C_1, \dots, C_p \subseteq \mathcal{X}'$, in such a way that $\mathcal{X}' = C_1 \cup \cdots \cup C_p$ and $C_i \cap C_j = \emptyset$ for $i, j \in \{1, \dots, p\}$, $i \neq j$. Such a partition $\mathcal{C} = \{C_1, \dots, C_p\}$ is called a *clustering* in this context. The *rectangular hulls* of $C_1, \dots, C_p$, namely the smallest inclusion-wise axis-parallel hyper-rectangles containing each cluster, are the rectangles to be determined as the solution to this problem.

Figure 1(a) depicts a sample instance in $\mathbb{R}^2$ (i.e., $d = 2$) with $n = 50$ points, whereas Figure 1(b) shows the optimal solution (with respect to the objective function to be presented later) for this instance when we are required to identify $p = 4$ clusters while being allowed to discard up to $q = 3$ points as outliers. The rectangles enclose the four clusters, and the three points not included in any rectangle are the points identified as outliers.

Hyper-rectangular clustering has been proposed as a model for *explainable clustering*, since it is straightforward to describe the obtained clusters by the bounds defining each hyper-rectangle. Indeed, if each coordinate corresponds to a relevant parameter in the application generating the given points, then clusters are specified by a lower and an upper bound on each parameter, and this is easier to communicate than a distance-based clustering. For example,

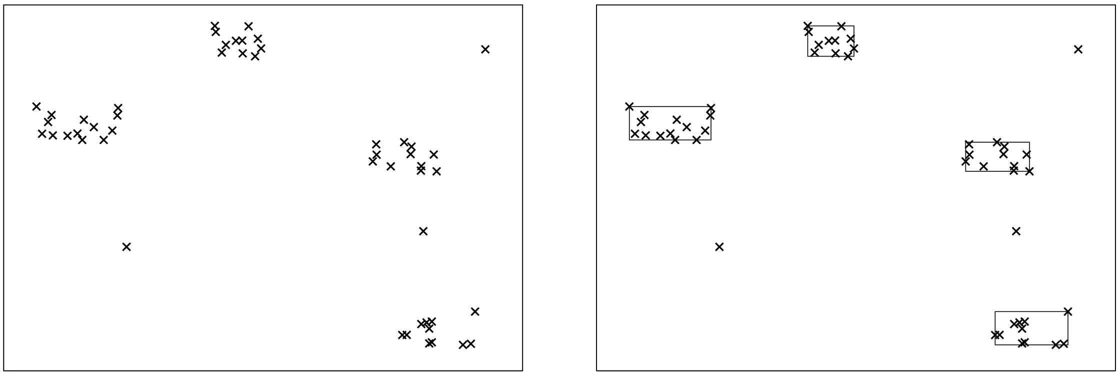

Figure 1: (a) Sample instance with dimension $d = 2$ and $n = 50$ points, and (b) optimal solution for this instance, with $p = 4$ clusters and $q = 3$ outliers.

in [3], axis-parallel hyper-rectangles are used to cluster points obtained by energy simulations, and such hyper-rectangles neatly describe the governing rules for each cluster in terms of the application parameters. In the particular application considered in [3], each hyper-rectangle describes a design strategy that, due to the rectangular shape of the clusters, can be easily communicated to the user.

In [7] an exhaustive analysis of rectangle-based clustering methods is performed, and heuristics are presented in order to find such clusterings in particular cases. In this work we tackle the case $q > 0$, namely the clustering may discard up to a pre-specified number q of points, which are thus declared to be outliers. To the best of the author's knowledge, the identification of outliers within axis-parallel hyper-rectangular clustering was first proposed in [10], by presenting a fast heuristic algorithm capable of discovering high-density regions (clusters) and low-density regions (outliers, negative clusters, holes, empty regions) at the same time. Outliers in the context of hyper-rectangular clustering have also been considered in [8] in the context of video processing.

It is out of the scope of this paper to discuss the merits of axis-parallel hyper-rectangular clustering models. Instead, we are interested in assessing how far standard mixed integer programming techniques can go at solving this kind of problems with optimality. The application of integer programming techniques to clustering and classification problems has been subject to great interest from the optimization community in recent years (see, e.g., [1, 2, 5, 12] and the recent survey [6]), and this work continues this line of research. To the best of the author's knowledge, the first integer programming approach for hyper-rectangular clustering was proposed in [11], where an integer programming formulation for the case $q = 0$ (i.e., all points must be clustered) is presented and is applied to image capturing. Integer programming concepts also appear in [9], where an integer-programming-based procedure is presented for identifying two hyper-rectangular clusters.

In this work we consider the total cluster span as the objective to be minimized. The *span* of a cluster $C \subseteq \mathcal{X}$ over the coordinate t is $\mathrm{span}_t(C) = \max\{x_t : x \in C\} - \min\{x_t : x \in C\}$ if $C \neq \emptyset$ and $\mathrm{span}_t(C) = 0$ otherwise, and the *total span* of C is $\mathrm{span}(C) = \sum_{t=1}^{d} \mathrm{span}_t(C)$. The *total span* of a clustering is the sum of the total spans of its constituent clusters. Such a linear objective function is not usual in clustering environments, but allows us to propose a clean integer programming approach, while at the same time providing a good clustering criterion. Indeed, although minimizing the total span can cause artifacts in the clustering, our experience shows that this situation is rare and that this objective function provides results that are visually consistent with the expectations for a reasonable clustering.

This work is organized as follows. Section 2 explores the computational complexity of this problem, showing that it is NP-hard in general. Section 3 presents a natural mixed integer programming formulation for the problem considered in this work. Section 4 explores a large family of valid inequalities, providing partial facetness results. Section 5 presents an exact separation procedure based on linear programming for the family of valid inequalities and reports computational experiments in order to evaluate this machinery. Finally, Section 6 closes the paper with concluding remarks and lines for future work. Throughout this work, for any nonnegative integer $m \in \mathbb{Z}_+$, we define $[m] := \{1, \ldots, m\}$ and $[m]_0 := \{0, \ldots, m\}$.

2 Computational complexity

In this section we explore the computational complexity of the hyper-rectangular clustering problem with axis-parallel clusters and outliers. Given a finite set $\mathcal{X} \subseteq \mathbb{R}^d$, integers $p \geq 1$ and $q \geq 0$, and a threshold $\kappa \in \mathbb{R}_+$, the decision version of the hyper-rectangular clustering problem with axis-parallel clusters and outliers consists in determining whether there exists a clustering with p clusters and up to q outliers with total span less than or equal to κ or not. Given a graph G and an integer $k \in \mathbb{Z}_+$, the decision version of the *maximum clique problem* consists in determining whether G contains a *clique* (i.e., a set of pairwise adjacent vertices) with cardinality greater than or equal to k or not. This problem is known to be NP-complete even when G is a regular graph [4].

Theorem 1 *The decision version of the hyper-rectangular clustering problem with axis-parallel clusters and outliers is NP-complete.*

Proof. It is not difficult to check that the decision version of the hyper-rectangular clustering problem with axis-parallel clusters and outliers belongs to NP, since we can nondeterministically generate a solution and check in polynomial time whether it is a valid solution with total span less than or equal to κ or not. In order to conclude the proof, we show that the decision version of the maximum clique problem for regular graphs can be reduced to the decision version of the hyper-rectangular clustering problem with axis-parallel clusters and outliers. To this end, consider an instance (G, k) (with $G = (V, E)$ an r-regular graph) of the decision version of the maximum clique problem, and assume w.l.o.g. $k \geq 3$. We construct an instance of the clustering problem as follows. We take $d := |E|$ and associate each coordinate axis with an edge of G. We define $\mathcal{X} := \{x^v\}_{v \in V}$ to include one point for every vertex v of G, defined as

$$x_e^v = \begin{cases} 1 & \text{if } v \text{ is incident to } e \\ 0 & \text{otherwise} \end{cases}$$

for every $e \in E$. We take $p := 1$, $q := |V| - k$, and $\kappa := rk - k(k-1)/2$. For any $A \subseteq V$, we define $C_A := \{x^v : v \in A\}$ to be the associated subset of points from $\mathcal{X}$. If $|A| \geq 3$ then at least one vertex in A is not incident to the edge e, for any $e \in E$. Hence, in this case we have that $\text{span}_e(C_A) = 1$ if some vertex in A is incident to e, and $\text{span}_e(C_A) = 0$ otherwise, for any $e \in E$. This implies that $\text{span}(C_A)$ equals the number of edges from E incident to at least one vertex from A, and the r-regularity of G implies $\text{span}(C_A) = r|A| - |E(A)|$, where $E(A) = \{uv \in E : u, v \in A\}$.

Since $p = 1$ and $q = |V| - k$, any feasible clustering is composed by exactly one cluster with k or more points. For any $A \subseteq V$ and $v \in V$, if we take $A' = A \cup \{v\}$ then the r-regularity of G implies that $\text{span}(C_{A'}) = r|A'| - |E(A')| \geq r|A| - |E(A)| = \text{span}(C_A)$. This implies that there exists an optimal clustering with the only cluster having exactly k points, since larger clusters cannot have a strictly smaller span. Let $A \subseteq V$ with $|A| = k$ such that the solution given by the single cluster C_A is optimal. We have $\text{span}(C_A) = rk - |E(A)| \geq rk - k(k-1)/2$, since $|E(A)| \leq k(k-1)/2$ for any subset A with k vertices. This implies that $\text{span}(C_A) = rk - k(k-1)/2$ if A is a k-clique in G, and $\text{span}(C_A) > rk - k(k-1)/2$ otherwise. Therefore, the instance $(\mathcal{X}, p, q)$ admits a clustering with total span less than or equal to κ if and only if G has a clique with cardinality greater than or equal to k. Since this reduction is polynomial, the result follows. □

Theorem 1 states that the general version of the hyper-rectangular clustering problem with axis-parallel clusters and outliers is NP-complete. However, a polynomial algorithm can be provided when the dimension is fixed to $d = 1$. This case corresponds to covering a set of points in the real line with segments of minimum total length, while discarding at most q points. To this end, fix $d = 1$ and consider a finite set $\mathcal{X} = \{x^1, \ldots, x^n\} \subseteq \mathbb{R}$ such that $x^i \leq x^{i+1}$ for $i = 1, \ldots, n-1$. Construct a digraph $G = (V, E)$ as follows. We take $V := \{v^i_{jk} : i \in [n]_0, j \in [p]_0, k \in [q]_0\}$, and we will interpret the vertex v^i_{jk} as indicating that all the points in $\{x^1, \ldots, x^i\}$ have been considered, j clusters have been formed, and k points from $\{x^1, \ldots, x^i\}$ have been discarded as outliers. We also define $E := E_1 \cup E_2$, where

$$\begin{aligned} E_1 &:= \{(v^i_{jk}, v^{i'}_{j+1,k}) : i, i' \in [n]_0, i < i', j \in [p-1]_0, k \in [q]_0\}, \\ E_2 &:= \{(v^i_{jk}, v^{i+1}_{j,k+1}) : i \in [n-1]_0, j \in [p]_0, k \in [q-1]_0\}. \end{aligned}$$

Traversing the arc $(v^i_{jk}, v^{i'}_{j+1,k}) \in E_1$ represents the inclusion of the cluster $\{x^{i+1}, \ldots, x^{i'}\}$ in the solution, whereas traversing the arc $(v^i_{jk}, v^{i+1}_{j,k+1})$ represents that the point x^{i+1} is being declared to be an outlier in the solution. Thus, any path from v^0_{00} to any vertex in $T := \{v^n_{jk}\}_{j \in [p], k \in [q]_0}$ corresponds to a feasible clustering of $\mathcal{X}$. Each edge $(v^i_{jk}, v^{i'}_{j+1,k}) \in E_1$ is assigned the weight $x^{i'} - x^{i+1}$, and each edge in E_2 is assigned null weight. In this setting, a shortest path from v^0_{00} to any vertex in T corresponds to an optimal solution of the 1-dimensional hyper-rectangular clustering problem with axis-parallel clusters and outliers.

These results leave open the question of whether there exists a fixed dimension d such that the d-dimensional hyper-rectangular clustering problem with axis-parallel clusters and outliers is NP-hard. As a first step, it would be interesting to explore whether a similar construction as the one provided above can be given for $d = 2$. However, for the general case with arbitrary dimension, we must resort to non-polynomial algorithms in order to find optimal solutions (unless P = NP) and, due to this fact, in the remainder of this work we consider integer programming techniques for the general version of this problem.

3 Integer programming formulation

We consider the following mixed integer program representing the clustering problem that we are interested in solving. For $i \in [n] = \{1, \ldots, n\}$ and $c \in [p] = \{1, \ldots, p\}$, we consider the binary variable z_{ic} representing whether x^i is assigned to the cluster c or not. Also, for

$c \in [p]$ and $t \in [d] = \{1, \dots, d\}$, the real variables $l_{tc}, r_{tc} \in \mathbb{R}$ represent a lower and an upper bound, respectively, for the points in the cluster c in the coordinate t. For $t \in [d]$, define $X_t := \{x_t : x \in \mathcal{X}\}$, $\min_t := \min(X_t)$, and $\max_t := \max(X_t)$. In this setting, we can formulate the problem as follows.

$$\begin{aligned}
\min \quad & \sum_{c=1}^{p} \sum_{t=1}^{d} r_{tc} - l_{tc} && && (1)\\
\text{s.t.} \sum_{c=1}^{p} z_{ic} \quad & \leq \quad 1 && \forall i \in [n], && (2)\\
l_{tc} + (\max_t - x_t^i) z_{ic} \quad & \leq \quad \max_t && \forall i \in [n], c \in [p], t \in [d], && (3)\\
r_{tc} + (\min_t - x_t^i) z_{ic} \quad & \geq \quad \min_t && \forall i \in [n], c \in [p], t \in [d], && (4)\\
l_{tc} \quad & \leq \quad r_{tc} && \forall c \in [p], t \in [d], && (5)\\
\sum_{c=1}^{p} \sum_{i=1}^{n} z_{ic} \quad & \geq \quad n - q, && && (6)\\
\min_t \leq l_{tc}, r_{tc} \quad & \leq \quad \max_t && \forall c \in [p], t \in [d], && (7)\\
z_{ic} \quad & \in \quad \{0, 1\} && \forall i \in [n], c \in [p]. && (8)
\end{aligned}$$

The objective function asks to minimize the sum of the total cluster spans. Constraints (2) ask every point to be assigned to at most one cluster, and a point is considered to be an outlier if it is assigned to no cluster. Constraints (3)-(4) bind the variables, in such a way that $l_{tc} \leq x_t^i \leq r_{tc}$ if the point i is assigned to the cluster c. Constraints (5) avoid bound crossings in empty clusters, whereas constraints (6) specify that at most q outliers can be selected. Finally, constraints (7) impose bounds for the l- and the r-variables, and constraints (8) specify that the z-variables are binary.

The formulation (2)-(8) has obvious symmetry issues (i.e., every clustering admits more than one representation within the model, by renaming the cluster indices), which could be problematic when attempting the solution of this model with general integer programming solvers. We can add any of the following families of symmetry-breaking constraints to this basic formulation in order to mitigate this situation:

$$\begin{aligned}
\sum_{i=1}^{n} z_{i,c-1} \quad & \leq \quad \sum_{i=1}^{n} z_{ic} && \forall c \in [p], c > 1, && (9)\\
\sum_{i=1}^{n} i \; z_{i,c-1} \quad & \leq \quad \sum_{i=1}^{n} i \; z_{ic} && \forall c \in [p], c > 1, && (10)\\
z_{ic} + \sum_{c'=c+1}^{p} z_{jc'} \quad & \leq \quad 1 + \sum_{k \in [n] : x_1^k < x_1^i} z_{kc} && \forall i, j \in [n], x_1^j < x_1^i, \forall c \in [p]. && (11)
\end{aligned}$$

The symmetry-breaking constraints (9) ask for the clusters to be ordered in size, whereas (10) enforce the same condition for the sum of the points indices of all clusters. Finally, the symmetry-breaking constraints (11) assert that if x^i is the point in the cluster c with the smallest value in the coordinate x_1 (i.e., $z_{ic} = 1$ and $z_{kc} = 0$ for every $k \in [n]$ with $x_1^k < x_1^i$), then x^j cannot be assigned to a cluster with index greater than c if $x_1^j < x_1^i$. This implies that the clusters are ordered with respect to the smallest coordinate x_1 of their points.

We get similar symmetry-breaking constraints by replacing the coordinate x_1 by some other coordinate in (11). Unfortunately, as we shall report in Section 5, the addition of each of the families (9)-(11) of symmetry-breaking constraints does not appear to be effective in practice, so we stick to the formulation (1)-(8) in this work.

Definition 1 *We define $\mathcal{P}(\mathcal{X}, p, q)$ to be the convex hull of all vectors $(z, l, r) \in \mathbb{R}^{np+pd+pd}$ satisfying (2)-(8).*

Throughout this paper we assume that for every coordinate $t \in [d]$ there exist points $x, x' \in \mathcal{X}$ with $x_t \neq x'_t$, i.e., that not all points have the same value in this coordinate. If this condition does not hold, then the coordinate t does not provide any information and can be eliminated (by projecting all points onto the space defined by the remaining coordinates), and this operation does not change the optimal solutions. We state this assumption for future reference.

Assumption 1 *For every $t \in [d]$ there exist points $x, x' \in \mathcal{X}$ with $x_t \neq x'_t$.*

If $\mathcal{C} = \{C_1, \ldots, C_p\}$ is a clustering of $\mathcal{X}$ (i.e., $C_i \subseteq \mathcal{X}$ for $i \in [p]$, $|C_1 \cup \cdots \cup C_p| \geq n - q$, and $C_i \cap C_j = \emptyset$ for $i, j \in [p]$, $i \neq j$), then we define $\chi^{\mathcal{C}} = (z, l, r)$ to be the solution constructed as follows. For $i \in [n]$ and $c \in [p]$, we take $z_{ic} = 1$ if $x^i \in C_c$ and $z_{ic} = 0$ otherwise. For $t \in [d]$ and $c \in [p]$ such that $C_c \neq \emptyset$, we take $l_{tc} = \min\{x_t : x \in C_c\}$ and $r_{tc} = \max\{x_t : x \in C_c\}$. Finally, for $t \in [d]$ and $c \in [p]$ such that $C_c = \emptyset$, we take $l_{tc} = \min_t$ and $r_{tc} = \max_t$. In other words, $\chi^{\mathcal{C}}$ encodes the clustering $\mathcal{C}$ in the formulation variables, with $\{l_{tc}, r_{tc}\}_{t\in[d]}$ representing the smallest axis-parallel rectangle containing C_c, for every $c \in [p]$ with $C_c \neq \emptyset$, and $\{l_{tc}, r_{tc}\}_{t\in[d]}$ representing the largest possible rectangle (satisfying the variable bounds) for every $c \in [p]$ with $C_c = \emptyset$.

For $c \in [p]$, we call $\mathcal{C}(c)$ to the clustering obtained by assigning all points to the cluster c, i.e., $\mathcal{C} = \{C_1, \ldots, C_p\}$ with $C_c = \mathcal{X}$ and $C_{c'} = \emptyset$ for $c' \in [p]$, $c' \neq c$.

Theorem 2 *The polytope $\mathcal{P}(\mathcal{X}, p, q)$ is full-dimensional if and only if (a) Assumption 1 holds, (b) $q \geq 1$, and (c) either $p \geq 2$ or there exist at most q points in $\mathcal{X}$ with $x_t = \min_t$ and there exist at most q points in $\mathcal{X}$ with $x_t = \max_t$, for every $t \in [d]$.*

Proof. Assume the hypotheses (a)-(c) hold, and let $(\zeta, \lambda, \rho) \in \mathbb{R}^{np+pd+pd}$ and $\pi_0 \in \mathbb{R}$ such that $\zeta z + \lambda l + \rho r = \pi_0$ for every $(z, l, r) \in \mathcal{P}(\mathcal{X}, p, q)$. We shall show that $(\zeta, \lambda, \rho) = \mathbf{0}$, thus proving that $\mathcal{P}(\mathcal{X}, p, q)$ is full-dimensional.
Claim 1: $\boldsymbol{\zeta_{ic} = 0}$ for every $\boldsymbol{i \in [n]}$ and every $\boldsymbol{c \in [p]}$. Consider the clustering $\mathcal{C}(c)$ assigning all points to the cluster c and construct the clustering $\mathcal{C}'$ from $\mathcal{C}(c)$ by leaving the point x^i as an outlier, which is allowed by the hypothesis (b). The solutions $\chi^{\mathcal{C}}$ and $\chi^{\mathcal{C}'}$ satisfy $(\zeta, \lambda, \rho)\chi^{\mathcal{C}} = \pi_0 = (\zeta, \lambda, \rho)\chi^{\mathcal{C}'}$ and only differ in the z_{ic}-variable, hence $\zeta_{ic} = 0$. $\Diamond$
Claim 2: $\boldsymbol{\lambda_{tc} = \rho_{tc} = 0}$ for every $\boldsymbol{t \in [d]}$ and every $\boldsymbol{c \in [p]}$. Consider the hypothesis (c), and assume first $p > 1$. The clustering $\mathcal{C}(c')$, for $c' \in [p]$, $c' \neq c$, assigns all points to the cluster c', so the solution $(\bar{z}, \bar{l}, \bar{r}) = \chi^{\mathcal{C}(c')}$ has $\bar{l}_{tc} = \min_t$, since C_c is empty in $\mathcal{C}(c')$. Assumption 1 implies that $\min_t < \max_t$, so construct the solution $\chi' = (\bar{z}, \hat{t}, \bar{r})$ from $\chi^{\mathcal{C}(c')}$ by setting $\hat{l}_{tc} = \min_t + \varepsilon$, with $0 < \varepsilon < \max_t - \min_t$ and leaving the remaining variables unchanged. The existence of $\chi^{\mathcal{C}(c')}$ and χ' shows that $\lambda_{tc} = 0$, and a symmetric construction shows that $\rho_{tc} = 0$.

Assume now that $p = 1$ and that the set $A = \{x \in \mathcal{X} : x_t = \min_t\}$ has cardinality at most q, according to the hypothesis (c). Consider the clustering $\mathcal{C} = \{\mathcal{X}\backslash A\}$, namely the only available cluster contains all points with the exception of those in A. Let $(\bar{z}, \bar{l}, \bar{r}) = \chi^{\mathcal{C}}$. Assumption 1 implies that $\bar{l}_{t1} > \min_t$, since all points with $x_t = \min_t$ are outliers in this clustering. Construct a new solution $(\bar{z}, \hat{l}, \bar{r})$ from $(\bar{z}, \bar{l}, \bar{r})$ by setting $\hat{l}_{t1} = \bar{l}_{t1} - \varepsilon$, for some $0 < \varepsilon < \min\{x_t : x \in \mathcal{X}\backslash A\} - \min_t$ and leaving the remaining variables unchanged. The existence of these two solutions shows that $\lambda_{t1} = 0$. The symmetric construction, now assuming that there exist at most q points in $\mathcal{X}$ with $x_t = \max_t$, shows that $\rho_{t1} = 0$. $\Diamond$

Since $(\zeta, \lambda, \rho) = \mathbf{0}$, we conclude that $\mathcal{P}(\mathcal{X}, p, q)$ is full-dimensional.

In order to prove the converse implication, we show that if either of the hypotheses (a)-(c) does not hold, then there exists an equality satisfied by all feasible solutions in $\mathcal{P}(\mathcal{X}, p, q)$, which hence is not full-dimensional. If the hypothesis (a) does not hold, then $\min_t = \max_t$ for some $t \in [d]$, and this implies $\min_t = l_{tc} = r_{tc} = \max_t$ for every $c \in [n]$. If the hypothesis (b) does not hold, then $q = 0$ and this implies $\sum_{c=1}^{p} z_{ic} = 1$ for every $i \in [n]$, since no outliers are allowed. Finally, if the hypothesis (c) does not hold, then $p = 1$ and there exist $t \in [d]$ and $A \subseteq \mathcal{X}$ with $|A| = q + 1$ such that either (c') $x_t = \min_t$ for all $x \in A$ or (c") $x_t = \max_t$ for all $x \in A$. Since $p = 1$ and at most q outliers can be identified, then at least one point from A belongs to the unique cluster in any feasible solution, hence either $l_{tc} = \min_t$ in every feasible solution if (c') holds, or $l_{tc} = \max_t$ in every feasible solution if (c") holds. $\square$

Standard solution procedures for general integer linear programs usually involve the resolution of the linear relaxation of the given formulation, i.e., the linear program obtained by relaxing the integrality constraints. The optimal value of the linear relaxation provides the first dual bound for a branch and bound procedure, hence its value is of great interest in this context. Unfortunately, the linear relaxation of the formulation (1)-(8) is weak, as the following proposition shows.

Proposition 1 *If $p \geq 2$, then the optimal solution of the linear program (1)-(7) plus the variable bounds $0 \leq z_{ic} \leq 1$ for $i \in [n]$ and $c \in [p]$ has null objective value.*

Proof. We claim that the solution $(\bar{z}, \bar{l}, \bar{r})$ given by $\bar{z}_{ic} = 1/p$ for every $i \in [n]$ and $c \in [p]$, and $\bar{l}_{tc} = \bar{r}_{tc} = (\min_t + \max_t)/2$ for every $t \in [d]$ and $c \in [p]$ is feasible for the linear program (1)-(7). This solution has null objective value, thus settling the proposition. Constraints (2) and (5) are trivially satisfied. For every $i \in [n]$, $c \in [p]$, and $t \in [d]$, the following calculation shows that constraint (3) is satisfied:

$$\begin{aligned}
\bar{l}_{tc} + (\max_t - x_t^i)\bar{z}_{ic} &= \frac{\min_t + \max_t}{2} + \frac{\max_t - x_t^i}{p} \\
&\leq \frac{\min_t + \max_t}{2} + \frac{\max_t - \min_t}{p} \\
&\leq \frac{\min_t + \max_t}{2} + \frac{\max_t - \min_t}{2} = \max_t.
\end{aligned}$$

The first inequality stems from the fact that $x_t^i \geq \min_t$, whereas the second inequality is implied by $p \geq 2$. A similar calculation shows that constraint (4) is satisfied:

$$\begin{aligned}
\bar{r}_{tc} + (\min_t - x_t^i)\bar{z}_{ic} &= \frac{\min_t + \max_t}{2} + \frac{\min_t - x_t^i}{p} \\
&\geq \frac{\min_t + \max_t}{2} + \frac{\min_t - \max_t}{p} \\
&\geq \frac{\min_t + \max_t}{2} + \frac{\min_t - \max_t}{2} = \min_t.
\end{aligned}$$

Again, the first inequality is a consequence of $x_t^i \leq \max_t$, whereas the second inequality is implied by $p \geq 2$ and $\min_t - \max_t \leq 0$. The left-hand-side of constraint (6) equals n, so this constraint is also satisfied. Finally, $(\bar{z}, \bar{l}, \bar{r})$ satisfies the variable bounds, and we conclude that this solution is indeed feasible. □

A direct consequence of Proposition 1 is that the initial dual bound within a branch and bound procedure will be the worst possible when $p \geq 2$. This suggests that the application of cutting planes may be useful within such a procedure, and the next section explores potential cuts within this setting.

4 A familiy of valid inequalities

Throughout this section we fix a cluster $c \in [p]$ and a coordinate $t \in [d]$, and we shall consider valid inequalities involving the associated variables, i.e., $\{z_{ic}\}_{i\in[n]} \cup \{l_{tc}, r_{tc}\}$. The following theorem identifies conditions ensuring validity for a general valid inequality on these variables.

Theorem 3 *Fix $c \in [p]$ and $t \in [d]$, and let $\alpha, \beta \geq 0$ and $\gamma \in \mathbb{R}^n$ with $\gamma \geq \mathbf{0}$. Assume $p \geq 2$ or $q = n$. The inequality*

$$\alpha\, r_{tc} - \beta\, l_{tc} \;\geq\; \sum_{i=1}^{n} \gamma_i z_{ic} - \delta \tag{12}$$

is valid for $\mathcal{P}(\mathcal{X}, p, q)$ if and only if (a) $\alpha x_2 - \beta x_1 \geq \sum_{i\in[n]:x_1\leq x_t^i\leq x_2} \gamma_i - \delta$ for every $x_1, x_2 \in X_t$ with $x_1 \leq x_2$, (b) $\delta \geq (\beta - \alpha)\max_t$, and (c) $\delta \geq (\beta - \alpha)\min_t$.

Proof. Assume first that the hypotheses (a)-(c) hold, and consider any feasible solution $(\bar{z}, \bar{l}, \bar{r}) \in \mathcal{P}(\mathcal{X}, p, q) \cap \mathbb{Z}^{np} \times \mathbb{R}^{pd+pd}$. Define $I = \{i \in [n] : \bar{z}_{ic} = 1\}$ to be the set of point indices assigned to the cluster c by this solution, and assume first $I \neq \emptyset$. Call $x_1 = \min\{x_t^i : i \in I\}$ and $x_2 = \max\{x_t^i : i \in I\}$. The following calculation shows that $(\bar{z}, \bar{l}, \bar{r})$ satisfies (12):

$$\begin{aligned}
\sum_{i=1}^{n} \gamma_i \bar{z}_{ic} = \sum_{i\in I} \gamma_i &= \sum_{i\in I:x_1\leq x_t^i\leq x_2} \gamma_i \\
&\leq \sum_{i\in[n]:x_1\leq x_t^i\leq x_2} \gamma_i \;\leq\; \alpha\, x_2 - \beta\, x_1 + \delta \;\leq\; \alpha\, \bar{r}_{tc} - \beta\, \bar{l}_{tc} + \delta.
\end{aligned}$$

The first inequality follows from the fact that $\gamma \geq \mathbf{0}$, the second inequality is a restatement of the hypothesis (a), and the third inequality is implied by $\bar{l}_{tc} \leq x_1 \leq x_2 \leq \bar{r}_{tc}$ and $\alpha, \beta \geq 0$. Assume that $I = \emptyset$. The model constraint (5) implies that $\alpha\bar{r}_{tc} - \beta\bar{l}_{tc} \geq (\alpha-\beta)\bar{l}_{tc}$. If $\alpha - \beta \geq 0$ then $(\alpha - \beta)\bar{l}_{tc} \geq (\alpha - \beta)\min_t \geq -\delta$, the last inequality implied by the hypothesis (c), so (12) is satisfied. If $\alpha - \beta < 0$ then $(\alpha - \beta)\bar{l}_{tc} \geq (\alpha - \beta)\max_t \geq -\delta$, the last inequality implied by the hypothesis (b), and again (12) is satisfied.

For the converse implication, assume (12) is a valid inequality. Let $x_1, x_2 \in X_t$ with $x_1 \leq x_2$, and construct a feasible solution $(\bar{x}, \bar{l}, \bar{r})$ such that exactly the points in $A = \{i \in [n] : x_1 \leq x^i_t \leq x_2\}$ are assigned to the cluster c. This is possible since either $p \geq 2$ (and thus the remaining points can be assigned to some other cluster) or $q = n$ (and thus any set of points can be discarded from all clusters). Define also $\bar{l}_{tc} := x_1$ and $\bar{r}_{tc} := x_2$, so we have

$$\alpha x_2 - \beta x_1 \;=\; \alpha \bar{r}_{tc} - \beta \bar{l}_{tc} \;\geq\; \sum_{i=1}^{n} \gamma_i \bar{z}_{ic} - \delta \;=\; \sum_{i \in [n] : x_1 \leq x^i_t \leq x_2} \gamma_i - \delta. \tag{13}$$

The inequality in (13) stems from the fact that $(\bar{x}, \bar{l}, \bar{r})$ satisfies (12), and this calculation shows that the hypothesis (a) is satisfied. Consider now any solution $(\hat{x}, \hat{l}, \hat{r})$ such that $\hat{z}_{ic} = 0$ for every $i \in [n]$ (again, such a solution exists since $p \geq 2$ or $q = n$) and with $\hat{l}_{tc} = \hat{r}_{tc} = \max_t$. Since $(\hat{x}, \hat{l}, \hat{r})$ satisfies (12), we have

$$(\alpha - \beta)\max_t \;=\; \alpha \hat{r}_{tc} - \beta \hat{l}_{tc} \;\geq\; \sum_{i=1}^{n} \gamma_i \hat{z}_{ic} - \delta \;=\; -\delta, \tag{14}$$

hence the hypothesis (b) is satisfied. A similar construction with a solution having $l_{tc} = r_{tc} = \min_t$ shows that $(\alpha - \beta)\min_t \geq -\delta$, and so the hypothesis (c) is satisfied. □

Consider, e.g., the point set $\mathcal{X} = \{x^i = (i, i)\}_{i=1}^4$. If $p \geq 2$, then the inequality $r_{tc} - l_{tc} \geq \sum_{i=1}^4 z_{ic} - 1$ is valid for any $t \in [d]$ and any $c \in [p]$, since $r_{tc} - l_{tc} \geq \max\{x^i_t : z_{ic} = 1\} - \min\{x^i_t : z_{ic} = 1\}$, and this last expression is greater than or equal to the number of points in c minus 1. This inequality belongs to the family identified in Theorem 3, and can be easily shown to be valid using this theorem.

It is interesting to note that the hypothesis $p \geq 2$ or $q = n$ is indeed necessary for Theorem 3. To this end, consider again the point set $\mathcal{X} = \{x^i = (i, i)\}_{i=1}^4$. Assume $p = 1$ and $q = 1$, and consider the inequality $r_{tc} - l_{tc} \geq z_{1c} + z_{4c} + 1$ for any $t \in [d]$ and $c = 1$. This inequality is valid since at least three points from $\mathcal{X}$ must be assigned to the only available cluster in any feasible solution, so $r_{tc} - l_{tc} \geq 2$. Furthermore, $z_{1c} = z_{4c} = 1$ implies $r_{tc} - l_{tc} = 3$, so the inequality is satisfied by all feasible solutions. However, neither of the hypotheses (a)-(c) in Theorem 3 is satisfied for this inequality, which is nevertheless valid for this instance. If $p = 1$ and $q \leq n - 1$, then the hypotheses (a)-(c) ensure the validity of (12) but, as the previous counterexample shows, the converse implication does not hold in general.

Theorem 3 provides a general characterization of valid inequalities with $\gamma \geq 0$ for a fixed cluster $c \in [c]$ and a fixed coordinate axis $t \in [d]$, showing that it suffices to consider the intervals between every pair of points along the coordinate axis t instead of resorting to all possible subsets of points. Nevertheless, the number of parameters governing each valid inequality is large and may not provide an intuition of the meaning provided by the inequality. Such a number of parameters can be reduced when $\alpha > \beta$ and $x \geq \mathbf{0}$ for every $x \in \mathcal{X}$, by taking $\lambda := \frac{\alpha}{\alpha - \beta}$, $\gamma'_i := \frac{\gamma_i}{\alpha - \beta}$ for every $i \in [n]$, and $\delta' := \frac{\delta}{\alpha - \beta}$. In this setting we can assume that $\delta = (\beta - \alpha)\min_t$, hence (12) can be written as

$$\lambda r_{tc} + (1 - \lambda) l_{tc} \;\geq\; \min_t + \sum_{i=1}^{n} \gamma'_i z_{ic}. \tag{15}$$

Theorem 3 implies that the inequality (15) is valid if and only if

$$\lambda x_2 + (1-\lambda)x_1 \;\geq\; \min_t + \sum_{i\in[n]:x_1\leq x_t^i\leq x_2} \gamma_i'$$

for every $x_1, x_2 \in X_t$, $x_1 \leq x_2$.

The family of valid inequalities identified by Theorem 3 includes facet-defining inequalities, as the following result shows.

Theorem 4 *Let $c \in [p]$, $t \in [d]$, and $s \in [n]$, and assume $x_t^1 \leq x_t^2 \leq \cdots \leq x_t^n$. Also suppose that (a) Assumption 1 holds, (b) $q \geq 1$, and (c) $p \geq 2$. Fix $\alpha, \beta \geq 0$ and let $\delta \geq \max\{(\beta-\alpha)\min_t, (\beta-\alpha)\max_t\}$. If*

- *(i) $\alpha x_t^i + \delta \geq \beta x_t^{i+1}$ for $i = 1, \ldots, n-1$,*
- *(ii) there exists $i_1 \in \{1, \ldots, n-1\}$ such that (ii') $\alpha x_t^{i_1} + \delta = \beta x_t^{i_1+1}$ and (ii") if $i_1 < s$ then $\min_{t'} < x_{t'}^{i_1} < \max_{t'}$ for every $t' \in [d]$, $t' \neq t$, whereas if $i_1 \geq s$ then $\min_{t'} < x_{t'}^{i_1+1} < \max_{t'}$ for every $t' \in [d]$, $t' \neq t$, and*
- *(iii) there exists $i_2 \in \{i_1+2, \ldots, n-1\}$ such that (iii') $\alpha x_t^{i_2} + \delta = \beta x_t^{i_2+1}$ and (iii") $x_t^{i_1+1} < x_t^{i_2}$,*

then the inequality (12) defines a facet of $\mathcal{P}(\mathcal{X}, p, q)$, with $\gamma_s := (\alpha-\beta)x_t^s + \delta$, $\gamma_i := \beta(x_t^{i+1} - x_t^i)$ for $i = 1, \ldots, s-1$, and $\gamma_i := \alpha(x_t^i - x_t^{i-1})$ for $i = s+1, \ldots, n$.

Proof. We first show that (12) is a valid inequality with this definition of γ and δ, by resorting to Theorem 3. We have that $\alpha, \beta \geq 0$ by the hypotheses, and we must also check that $\gamma \geq \mathbf{0}$ in order to apply Theorem 3. The definition of γ_i for $i \neq s$ directly implies $\gamma_i \geq 0$ since $\alpha, \beta \geq 0$ and $x_t^j \leq x_t^{j+1}$ for $j = 1, \ldots, n-1$, so we are left to verify $\gamma_s \geq 0$. To this end, consider the following cases.

- If $\alpha \geq \beta$ then $\delta \geq (\beta-\alpha)\min_t$ implies $\gamma_s = (\alpha-\beta)x_t^s + \delta \geq (\alpha-\beta)(x_t^s - \min_t) \geq 0$, since $\alpha - \beta \geq 0$ and $x_t^s - \min_t \geq 0$.
- If $\alpha < \beta$ then $\delta \geq (\beta-\alpha)\max_t$ implies $\gamma_s = (\alpha-\beta)x_t^s + \delta \geq (\alpha-\beta)(x_t^s - \max_t) \geq 0$, since $\alpha - \beta < 0$ and $x_t^s - \max_t \leq 0$.

We now check that the hypothesis (a) of Theorem 3 is satisfied. To this end, consider any two $x_1, x_2 \in X_t$ with $x_1 \leq x_2$, and let $j, k \in [n]$, $j \leq k$, such that $\{i \in [n] : x_1 \leq x_t^i \leq x_2\} = [j, k] := \{j, \ldots, k\}$ (thus implying $x_t^j = x_1$ and $x_t^k = x_2$).

- If $k < s$ then $\sum_{i=j}^k \gamma_i = \beta(x_t^{k+1} - x_t^j)$. The following calculation shows that the hypothesis (a) is satisfied:

  $$\alpha x_2 - \beta x_1 \;=\; \alpha x_t^k - \beta x_t^j \;\geq\; \beta(x_t^{k+1} - x_t^j) - \delta \;=\; \sum_{i=j}^k \gamma_i - \delta.$$

  In this expression, the inequality is equivalent to $\alpha x_t^k + \delta \geq \beta x_t^{k+1}$, which is implied by the hypothesis (i).

- If $j > s$ then $\sum_{i=j}^{k} \gamma_i = \alpha(x_t^k - x_t^{j-1})$. A similar calculation shows that in this case the hypothesis (a) is also satisfied:

$$\alpha x_2 - \beta x_1 \;=\; \alpha x_t^k - \beta x_t^j \;\geq\; \alpha(x_t^k - x_t^{j-1}) - \delta \;=\; \sum_{i=j}^{k} \gamma_i - \delta.$$

  Again, the inequality in this expression simplifies to $\alpha x_t^{j-1} + \delta \geq \beta x_t^j$, which is implied by the hypothesis (i).

- Finally, if $j \leq s \leq k$, then $\sum_{i=j}^{k} \gamma_i = \alpha x_t^k - \beta x_t^j + \delta = \alpha x_2 - \beta x_1 + \delta$, so the hypothesis (a) is trivially satisfied with equality.

Since $\delta \geq \max\{(\beta-\alpha)\min_t, (\beta-\alpha)\max_t\}$, we have that $\delta \geq (\beta-\alpha)\max_t$ and $\delta \geq (\beta-\alpha)\min_t$, so the hypotheses (b) and (c) of Theorem 3 are satisfied. Therefore, Theorem 3 ensures that (12) is a valid inequality for this particular definition of γ and δ.

Now for facetness. By Theorem 2, the hypotheses (a)-(c) ensure that $\mathcal{P}(\mathcal{X}, p, q)$ is full-dimensional. Let F be the face of $\mathcal{P}(\mathcal{X}, p, q)$ induced by (12) and let $(\zeta, \lambda, \rho) \in \mathbb{R}^{np+pd+pd}$ and $\pi_0 \in \mathbb{R}$ such that $\zeta z + \lambda l + \rho r = \pi_0$ for every $(z, l, r) \in F$. We shall show that (ζ, λ, ρ) is a multiple of the coefficient vector of (12) which, combined with the full dimensionality of $\mathcal{P}(\mathcal{X}, p, q)$, will allow us to conclude that F is indeed a facet of this polytope.

The hypothesis (ii) ensures the existence of $i_1 \in \{1, \dots, n-1\}$ such that $\alpha x_t^{i_1} + \delta = \beta x_t^{i_1+1}$. For any $c' \in [p]$, $c' \neq c$, let $\chi_{c'}^1$ be the solution specified as follows.

- If $i_1 < s$, define $\bar{\mathcal{C}} = \{\bar{C}_1, \dots, \bar{C}_p\}$ to be the clustering given by $\bar{C}_c = \{x^{i_1}\}$, $\bar{C}_{c'} = \chi \backslash \bar{C}_c$, and $\bar{C}_{c''} = \emptyset$ for every $c'' \in [p] \backslash \{c, c'\}$, and take $\chi_{c'}^1 = (\bar{z}, \bar{l}, \bar{r}) := \chi^{\bar{\mathcal{C}}}$. The following calculation shows that $\chi_{c'}^1$ satisfies (12) with equality:

$$\begin{aligned}\sum_{j=1}^{n} \gamma_j \bar{z}_{jc} - \delta \;=\; \gamma_{i_1} - \delta \;&=\; \beta(x_t^{i_1+1} - x_t^{i_1}) - \delta \;=\; (\alpha x_t^{i_1} + \delta) - \beta x_t^{i_1} - \delta \\ &=\; \alpha x_t^{i_1} - \beta x_t^{i_1} \;=\; \alpha \bar{r}_{tc} - \beta \bar{l}_{tc}.\end{aligned}$$

- If $i_1 \geq s$, define $\bar{\mathcal{C}} = \{\bar{C}_1, \dots, \bar{C}_p\}$ to be the clustering given by $\bar{C}_c = \{x^{i_1+1}\}$, $\bar{C}_{c'} = \chi \backslash \bar{C}_c$, and $\bar{C}_{c''} = \emptyset$ for every $c'' \in [p] \backslash \{c, c'\}$, and take $\chi_{c'}^1 = (\bar{z}, \bar{l}, \bar{r}) := \chi^{\bar{\mathcal{C}}}$. Again, the following calculation shows that $\chi_{c'}^1$ satisfies (12) with equality:

$$\begin{aligned}\sum_{j=1}^{n} \gamma_j \bar{z}_{jc} - \delta \;=\; \gamma_{i_1+1} - \delta \;&=\; \alpha(x_t^{i_1+1} - x_t^{i_1}) - \delta \;=\; \alpha x_t^{i_1+1} - (\beta x_t^{i_1+1} - \delta) - \delta \\ &=\; \alpha x_t^{i_1+1} - \beta x_t^{i_1+1} \;=\; \alpha \bar{r}_{tc} - \beta \bar{l}_{tc}.\end{aligned}$$

We define $\chi_{c'}^2$ similarly, by considering a clustering in which c is composed only by x^{i_2} if $i_2 < s$ and only by x^{i_2+1} otherwise.

Claim 1: $\zeta_{ic'} = 0$ for every $i \in [n]$ and every $c' \in [p]$, $c' \neq c$. If $i < s$, then consider the solution $\chi := \chi_{c'}^1$ if $i \neq i_1$ and $\chi := \chi_{c'}^2$ otherwise. If $i = s$ then take $\chi := \chi_{c'}^1$. Finally, if $i > s$, then consider the solution $\chi := \chi_{c'}^1$ if $i \neq i_1 + 1$ and $\chi := \chi_{c'}^2$ otherwise. Call $(\bar{z}, \bar{l}, \bar{r}) := \chi$, and note that $\bar{z}_{ic'} = 1$. Construct $\hat{\chi} = (\hat{z}, \bar{l}, \bar{r})$ from χ by setting $\hat{z}_{ic'} = 0$ and leaving the remaining variables unchanged (this construction is possible since the hypothesis

(b) allows at least one outlier in any solution). Since $z_{ic'}$ has null coefficient in (12), then $\hat{\chi} \in F$. Since χ and $\hat{\chi}$ only differ in the $z_{ic'}$-variable, then $\zeta\bar{z} + \lambda\bar{l} + \rho\bar{r} = \pi_0 = \zeta\hat{z} + \lambda\bar{l} + \rho\bar{r}$ implies $\zeta_{ic'} = 0$. $\diamondsuit$

Claim 2: $\boldsymbol{\lambda_{t'c} = \rho_{t'c} = 0}$ for every $\boldsymbol{t' \in [d]}$, $\boldsymbol{t' \neq t}$. Let $c' \in [p]$, $c' \neq c$, and consider the solution $\chi = (\bar{z}, \bar{l}, \bar{r}) := \chi^1_{c'}$. Construct the solution $\hat{\chi} = (\bar{z}, \bar{l}, \hat{r})$ from χ by setting $\hat{r}_{t'c} = \bar{r}_{t'c} + \varepsilon$, for some $0 < \varepsilon < \max_{t'} - \bar{r}_{t'c}$ and leaving the remaining variables unchanged (this construction is possible since $\bar{r}_{t'c} = x^{i_1}_{t'} < \max_{t'}$ if $i_1 < s$ and $\bar{r}_{t'c} = x^{i_1+1}_{t'} < \max_{t'}$ if $i_1 \geq s$, by the hypothesis (ii")). We have $\hat{\chi} \in F$ since $r_{t'c}$ has null coefficient in (12), and the existence of χ and $\hat{\chi}$ in F only differing in the $r_{t'c}$-variable implies $\rho_{t'c} = 0$. Consider now the solution $\tilde{\chi} = (\bar{z}, \hat{l}, \hat{r})$ constructed from $\hat{\chi}$ by setting $\hat{l}_{t'c} = \bar{l}_{t'c} - \varepsilon$, for some $0 < \varepsilon < \bar{l}_{t'c} - \min_{t'c}$, and leaving the remaining variables unchanged (which again is possible since $\bar{l}_{t'c} = x^{i_1}_{t'} > \min_{t'}$ if $i_1 < s$ and $\bar{l}_{t'c} = x^{i_1+1}_{t'} > \min_{t'}$ if $i_1 \geq s$, by the hypothesis (ii")). Again, $\tilde{\chi} \in F$ and the existence of $\hat{\chi}$ and $\tilde{\chi}$ in F only differing in the $l_{t'c}$-variable implies $\lambda_{t'c} = 0$. $\diamondsuit$

Claim 3: $\boldsymbol{\lambda_{t'c'} = \rho_{t'c'} = 0}$ for every $\boldsymbol{t' \in [d]}$ and every $\boldsymbol{c' \in [p]}$, $\boldsymbol{c' \neq c}$. Consider the clustering $\mathcal{C}(c)$ assigning all points to the cluster c. The solution $\chi = (\bar{z}, \bar{l}, \bar{r}) := \chi^{\mathcal{C}(c)}$ has $\sum_{i=1}^n \gamma_i \bar{z}_{ic} = \alpha \max_t - \beta \min_t + \delta$, $\bar{l}_{tc} = \min_t$, and $\bar{r}_{tc} = \max_t$, so χ satisfies (12) with equality and thus $\chi \in F$. Consider the solution $\hat{\chi} = (\bar{z}, \bar{l}, \hat{r})$ constructed from $(\bar{z}, \bar{l}, \bar{r})$ by setting $\hat{r}_{t'c'} = \bar{r}_{t'c'} - \varepsilon = \max_{t'} - \varepsilon$, for $0 < \varepsilon < \max_{t'} - \min_{t'}$, and leaving the remaining variables unchanged (this construction is possible since Assumption 1 ensures that $\min_{t'} < \max_{t'}$). We have $\hat{\chi} \in F$ since $r_{t'c'}$ has null coefficient in (12), and the existence of χ and $\hat{\chi}$ in F only differing in the $r_{t'c'}$-variable implies $\rho_{t'c'} = 0$. Consider now the solution $\tilde{\chi} = (\bar{z}, \hat{l}, \hat{r})$ constructed from $\hat{\chi}$ by setting $\hat{l}_{t'c'} = \hat{r}_{t'c'}$ and leaving the remaining variables unchanged. Again, $\tilde{\chi} \in F$ and the existence of $\hat{\chi}$ and $\tilde{\chi}$ in F only differing in the $l_{t'c'}$-variable implies $\lambda_{t'c'} = 0$. $\diamondsuit$

In order to conclude the proof, we show that (ζ, λ, ρ) is a multiple of the coefficient vector of (12). Fix $c' \in [p]$, $c' \neq c$. For $i = 0, \ldots, n - s$, define the clustering $\mathcal{C}^i = \{C^i_1, \ldots, C^i_p\}$ to be $C^i_c = \{x^s, \ldots, x^{s+i}\}$, $C^i_{c'} = \chi \backslash C^i_c$, and $C^i_{c''} = \emptyset$ for $c'' \in [p] \backslash \{c, c'\}$, and call $\bar{\chi}^i = (\bar{z}^i, \bar{l}^i, \bar{r}^i) := \chi^{\mathcal{C}^i}$. The following calculation shows that $\bar{\chi}^i$ satisfies (12) with equality:

$$\begin{aligned}
\sum_{j=1}^{n} \gamma_j \bar{z}_{jc} \;=\; \sum_{j=s}^{s+i} \gamma_j \;&=\; (\alpha - \beta) x^s_t + \delta + \sum_{j=s+1}^{s+i} \alpha (x^j_t - x^{j-1}_t) \\
&=\; \alpha x^{s+i}_t - \beta x^s_t + \delta \;=\; \alpha \bar{r}_{tc} - \beta \bar{l}_{tc} + \delta.
\end{aligned}$$

Similarly, for $i = 1, \ldots, s-1$, define the clustering $\mathcal{D}^i = \{D^i_1, \ldots, D^i_p\}$ to be $D^i_c = \{x^{s-i}, \ldots, x^s\}$, $D^i_{c'} = \chi \backslash D^i_c$, and $D^i_{c''} = \emptyset$ for $c'' \in [p] \backslash \{c, c'\}$, and call $\hat{\chi}^i = (\hat{z}^i, \hat{l}^i, \hat{r}^i) := \chi^{\mathcal{D}^i}$. Again, we have that $\hat{\chi}^i$ satisfies (12) with equality:

$$\begin{aligned}
\sum_{j=1}^{n} \gamma_j \hat{z}_{jc} \;=\; \sum_{j=s-i}^{s} \gamma_j \;&=\; (\alpha - \beta) x^s_t + \delta + \sum_{j=s-i}^{s-1} \beta (x^{j+1}_t - x^j_t) \\
&=\; \alpha x^s_t - \beta x^{s-i}_t + \delta \;=\; \alpha \hat{r}_{tc} - \beta \hat{l}_{tc} + \delta.
\end{aligned}$$

It is easy to verify that $\{\bar{\chi}^i\}_{i=0}^{n-s} \cup \{\hat{\chi}^i\}_{i=1}^{s-1}$ are linearly independent, since each of these points has a z-variable with value 1 that has value 0 in the previous points. Indeed, if we consider the sequence $A = (\bar{\chi}^0, \ldots, \bar{\chi}^{n-s}, \hat{\chi}^1, \ldots, \hat{\chi}^{s-1})$, then

- for $i = 0, \ldots, n - s$, the solution $\bar{\chi}^i$ has $z_{s+i,c} = 1$ and this variable is set to 0 in the previous solutions in the sequence A, and

- for $i = 1, \ldots, s-1$, the solution $\hat{\chi}^i$ has $z_{s-i,c} = 1$ and this variable is set to 0 in the previous solutions in the sequence A.

This implies that the solutions in A are linearly independent, so they are affinely independent.

The solution $\chi^1_{c'}$ is affinely independent with respect to the solutions in A, since $z_{sc} = 1$ for every solution in A but the variable z_{sc} is set to 0 in $\chi^1_{c'}$. So we conclude that $A' := A \cup \{\chi^1_{c'}\}$ is a sequence of affinely independent solutions. We complete the construction by claiming that $\chi^2_{c'}$ is affinely independent with respect to the solutions in A'.

- If $i_1 < s$, then all the solutions in A' satisfy the equation

$$l_{tc} \;=\; x^s_t - \sum_{i=1}^{s-1} (x^{i+1}_t - x^i_t) z_{ic} - (x^s_t - x^{i_1+1}_t)\,(1 - z_{sc}), \tag{16}$$

  but this equation is not satisfied by $\chi^2_{c'}$, which has $z_{i_2 c} = 1$ and $z_{ic} = 0$ for $i \in [n] \backslash \{i_2\}$. Indeed, take $(\tilde{z}, \tilde{l}, \tilde{r}) := \chi^2_{c'}$ and consider the following cases. If $i_2 < s$ then $\tilde{l}_{tc} = x^{i_2}_t$ and the right-hand-side of (16) equals $x^{i_2}_t + x^{i_1+1}_t - x^{i_2+1}_t$, which differs from $\tilde{l}_{tc}$ by the hypothesis (iii”). If $i_2 \geq s$ then $\tilde{l}_{tc} = x^{i_2+1}_t$ and the right-hand-side of (16) equals $x^{i_1+1}_t$, which again differs from $\tilde{l}_{tc}$ by the hypothesis (iii”).

- If $i_1 \geq s$, then all the solutions in A' satisfy the equation

$$r_{tc} \;=\; x^s_t + \sum_{i=s+1}^{n} (x^i_t - x^{i-1}_t) z_{ic} - (x^s_t - x^{i_1}_t)\,(1 - z_{sc}), \tag{17}$$

  but again this equation is not satisfied by $\chi^2_{c'}$. Indeed, take $(\tilde{z}, \tilde{l}, \tilde{r}) := \chi^2_{c'}$ and consider the following cases. If $i_2 < s$ then $\tilde{r}_{tc} = x^{i_2}_t$ and the right-hand-side of (17) equals $x^{i_1}_t$, which differs from $\tilde{r}_{tc}$ by the hypothesis (iii”). If $i_2 \geq s$ then $\tilde{r}_{tc} = x^{i_2+1}_t$ and the right-hand-side of (17) equals $x^{i_2}_t + x^{i_1}_t - x^{i_2+1}_t$, which differs from $\tilde{r}_{tc}$ by the hypothesis (iii”).

This shows that $A' \cup \{\chi^2_{c'}\}$ is a sequence of $n+2$ affinely independent solutions. Construct now the system of linear equations $(\zeta, \lambda, \rho)(\chi - \chi^2_{c'}) = 0$ for $\chi \in A'$, where (ζ, λ, ρ) are the unknowns. Claims 1-3 imply that $np + 2pd - (n+2)$ unknowns are null, hence if we eliminate these unknowns we are left with $n+1$ equations for $n+2$ unknowns. Since the solutions in $A' \cup \{\chi^2_{c'}\}$ are affinely independent, then the vectors $\{\chi - \chi^2_{c'}\}_{\chi \in A'}$ are linearly independent, so the set of solutions to this system of linear equations has dimension 1. Since any multiple of the coefficient vector of (12) is a solution to this system (as all points in F satisfy (12) with equality), we conclude that the only solutions to this system are the multiples of the coefficient vector of (12). Therefore, (ζ, λ, ρ) is a multiple of the coefficient vector of (12) which, therefore, defines a facet of $\mathcal{P}(\mathcal{X}, p, q)$. $\square$

The particular definition of γ in Theorem 4 implies that a solution in which the cluster c only contains the point x^s trivially satisfies (12) with equality. Furthermore, if the points $s+1, \ldots, n$ are sequentially added to the cluster c then the inequality keeps being satisfied with equality, and the same happens in the points $s-1, \ldots, 1$ are sequentially added to the cluster c. For example, consider the instance $\chi = \{(i,i)\}_{i=1}^5$ with $p = 2$ and $q = 1$. The inequality $r_{tc} - l_{tc} \geq \sum_{i \in [n]} z_{ic} - 1$ corresponds to taking $\alpha = \beta = \delta = 1$, $s = 3$, and defining γ according to the statement of Theorem 4. In this setting, Theorem 4 directly implies that this inequality induces a facet of $\mathcal{P}(\mathcal{X}, p, q)$.

5 Computational experiments

We have implemented a branch and cut procedure for the hyper-rectangular clustering problem with axis-parallel clusters and outliers, and this section reports this implementation and the obtained computational results.

We use the valid inequalities identified in Theorem 3 as cuts in the implementation. Theorem 3 shows that it suffices to check $O(n^2)$ conditions in order to guarantee validity of the inequality (12). This allows for a polynomial separation procedure for these inequalities, via linear programming. Given a fractional solution (z^*, l^*, r^*), we consider the following formulation, which includes the coefficients α, β, $\{\gamma_i\}_{i\in[n]}$, and δ of the inequality (12) as the model variables.

$$
\begin{aligned}
\max \qquad & \beta\, l^*_{tc} - \alpha\, r^*_{tc} + \sum_{i=1}^{n} z^*_{ic}\,\gamma_i - \delta && (18)\\
\sum_{i\in[n]:x_1\le x_t^i\le x_2} \gamma_i \ \le\ & x_2\alpha - x_1\beta + \delta \qquad \forall x_1, x_2 \in X_t, x_1 \le x_2 && (19)\\
\beta \max\nolimits_t - \alpha \max\nolimits_t \ \le\ & \delta && (20)\\
\beta \min\nolimits_t - \alpha \min\nolimits_t \ \le\ & \delta && (21)\\
\alpha + \beta \ =\ & n+1 && (22)\\
\alpha, \beta \ \ge\ & 0 && (23)\\
\gamma_i \ \ge\ & 0 \qquad \forall i \in [n] && (24)
\end{aligned}
$$

The objective function asks to maximize the cut depth. Constraints (19)-(21) enforce the validity conditions (a)-(c) specified by Theorem 3, whereas constraint (22) normalizes the coefficients in order to prevent this linear program from being unbounded. If the optimal value of this linear program is positive, then an optimal solution provides the coefficients of a violated inequality (12) and viceversa. Since the fractional solution (z^*, l^*, r^*) only participates in the objective function, we can set up one such linear program for each axis $t \in [d]$, and warm-start the resolution by updating the coefficients in the objective function every time a new fractional solution must be separated.

It is possible to provide a linear number of (stronger) conditions ensuring the validity of the inequality (12), which could be helpful within the separation stage. The following direct corollary to Theorem 3 states these conditions.

Corollary 1 *Fix $c \in [p]$ and $t \in [d]$, and let $\alpha, \beta \ge 0$ and $\gamma \in \mathbb{R}^n$ with $\gamma \ge \mathbf{0}$. Assume $p \ge 2$ or $q = n$, and also suppose $x_t^i \le x_t^{i+1}$ for $i = 1, \ldots, n-1$. If (a') $\gamma_i \le \beta(x_t^{i+1} - x_t^i)$ for $i = 1, \ldots, n-1$ and (b') $(\alpha - \beta)x_t^i \ge \gamma_i - \delta$ for $i = 1, \ldots, n$, then the inequality (12) is valid for $\mathcal{P}(\mathcal{X}, p, q)$.*

Proof. It suffices to show that the hypotheses (a') and (b') imply the conditions (a)-(c) in Theorem 3. Consider first any $x_1, x_2 \in X_t$, $x_1 \le x_2$, and let $i, j \in [n]$ such that $x_t^i = x_1$ and $x_t^j = x_2$. By summing the following conditions coming from the hypotheses (a') and (b') we get that the condition (a) in Theorem 3 is satisfied for x_1 and x_2:

$$\begin{array}{lrcl}
\text{(a')} & \beta(x_t^{i+1} - x_t^i) & \geq & \gamma_i \\
\text{(a')} & \beta(x_t^{i+2} - x_t^{i+1}) & \geq & \gamma_{i+1} \\
\text{(a')} & \beta(x_t^{i+3} - x_t^{i+2}) & \geq & \gamma_{i+2} \\
 & \vdots & \vdots & \vdots \\
\text{(a')} & \beta(x_t^{j} - x_t^{j-1}) & \geq & \gamma_{j-1} \\
\text{(b')} & (\alpha-\beta)x_t^j & \geq & \gamma_j - \delta
\end{array}$$

For condition (b) in Theorem 3, the hypothesis (b') implies $(\alpha - \beta)\max_t = (\alpha - \beta)x_t^n \geq \gamma_n - \delta \geq -\delta$, since $\gamma \geq \mathbf{0}$. A similar calculation with x^1 shows that the condition (c) in Theorem 3 is also satisfied. □

Corollary 1 provides sufficient conditions for (12) to be valid, but these conditions are not necessary. For example, consider the instance $\mathcal{X} = \{(i,i)\}_{i=1}^3$ and fix a cluster index $c \in [p]$. For $q = 3$, the inequality $3r_{1c} - l_{1c} \geq 1.1\ z_{2c} + 2$ is valid for this instance, but it does not satisfy the hypothesis (a') for $i = 2$.

The conditions ensuring validity given by Corollary 1 allow to simplify the search of a violated cut. Namely, by assuming that $\gamma_i = \min\{\delta + (\alpha-\beta)x_t^i, \beta(x_t^{i+1} - x_t^i)\}$ for $i \in [n]$, any feasible solution with positive objective function to the following linear program identifies such a violated cut.

$$\max \quad \beta\, l_{tc}^* - \alpha\, r_{tc}^* + \sum_{i=1}^{n} z_{ic}^*\, \gamma_i - \delta \tag{25}$$

$$\beta(x_t^{i+1} - x_t^i) \geq \gamma_i \qquad i = 1,\ldots,n-1 \tag{26}$$

$$(\alpha-\beta)x_t^i \geq \gamma_i - \delta \qquad i = 1,\ldots,n \tag{27}$$

$$\alpha + \beta = n+1 \tag{28}$$

$$\alpha, \beta \geq 0 \tag{29}$$

$$\gamma_i \geq 0 \quad \forall i \in [n] \tag{30}$$

Due to the previous observations, solving this linear program does not provide an exact algorithm for separating this family of valid inequalities, i.e., an optimal value with nonpositive objective function does not guarantee that there are no violated cuts from this familiy. Nevertheless, it is interesting to consider this formulation since it is composed by a linear number of constraints, and it might be worthwile to solve this smaller linear program instead of the quadratically-sized formulation (18)-(24).

We have implemented the branch and cut procedure within the framework provided by Cplex 12.4. The controller code is implemented in Java, interfacing with Cplex through the Concert API. The linear programs for separating the inequalities (12) are also implemented with the Cplex linear programming solver, and is attached to the main integer programming solver of Cplex with a user-cut callback.

In order to conduct experiments with controlled instances having predictable optima, we use synthetic instances randomly generated with the procedure specified in Algorithm 1. This procedure takes as input the dimension d, the number n of points to be generated, the number p of clusters to generate, the number q of outliers to generate, and a parameter $s \in [0,1]$

specifying the *dispersion* for the generated clusters. In this pseudocode, we assume the existence of a procedure choose(A) that selects a point from the (finite or infinite) set $A \subseteq \mathbb{R}^d$ with uniform distribution. In lines 1–5 a set of p (= number of clusters) points is generated, which will act as the originating points for the clusters. In lines 6–11 the p clusters are generated, by constructing $n-q$ points around the originating points. The dispersion of these clusters is controlled by the parameter s. Finally, in lines 12–15, q additional points are added within the range $[-1,1]^d$, which will act as the outliers to be identified. If the parameter s is small enough, then the generated instance will be very likely to contain p clearly-identifyable clusters and up to q points clearly separated from these clusters. This allows us to control the characteristic of the instances in our experiments. The source code and the instance generator can be found at `https://github.com/jmarenco/clusterswithoutliers`.

Algorithm 1 Generation of synthetic instances.

```
1:  C ← ∅
2:  for i = 1 to p do                        ▷ Generates p cluster "centers"
3:      x ← choose([−1, 1]^d)
4:      C ← C ∪ {x}
5:  end for
6:  𝒳 ← ∅
7:  for i = 1 to n − q do                    ▷ Generates n − q points around the cluster centers
8:      x ← choose(C)
9:      y ← choose([−s/2, s/2]^d)
10:     𝒳 ← 𝒳 ∪ {x + y}
11: end for
12: for i = 1 to q do                        ▷ Generates q outliers
13:     x ← choose([−1, 1]^d)
14:     𝒳 ← 𝒳 ∪ {x}
15: end for
16: return 𝒳
```

The primary objective of the experimentation reported in this section is to evaluate the contribution of the family of valid inequalities (12) within a cutting plane environment in order to solve the clustering problem with optimality. A first observation towards this objetive is that the dynamic addition of cuts coming from the separation procedure specified in Section 4 achieves a dramatic improvement in the total number of nodes in the enumeration tree in the instances solved with optimality. As a representative example, Figure 2(a) shows the number of nodes for the out-of-the-box algorithm implemented by Cplex and the branch and cut procedure specified previously for $d = 2$, $p = 5$, and $q = 3$, and $n = 10, \dots, 60$. In these experiments we set a time limit of 600 seconds. As mentioned before, since the optimal value of the linear relaxation is 0 for $p \geq 2$, then the improvement of the dual bound becomes a key ingredient of a successful procedure. Cuts coming from the inequalities (12) enable such an improvement, and this explains the reduction in the number of overall nodes for the instances solved with optimality within the time limit.

This reduction has the additional computational cost of the linear-programming-based separation procedure, hence the overall running time does not decrease proportionally to the node reduction. Figure 2(b) shows the overall running time to optimality for the same instances, showing nevertheless a reduction but of a more modest nature than for the total

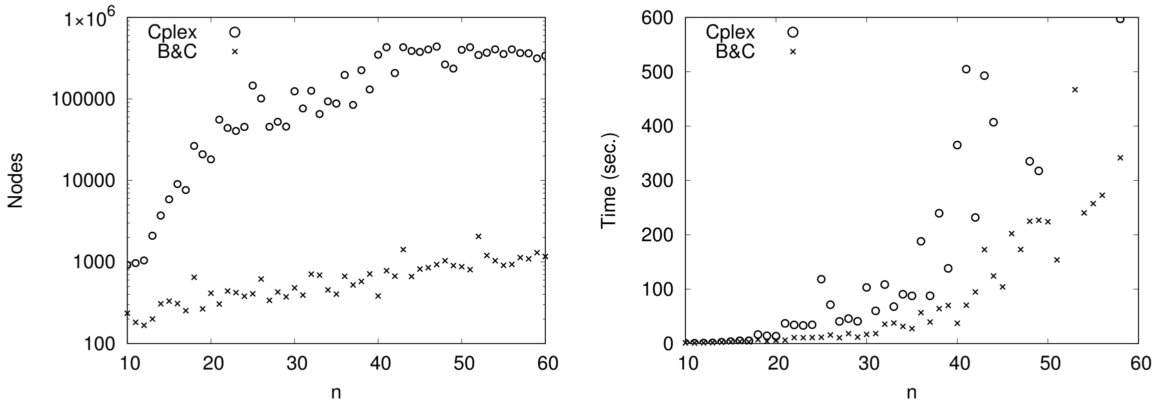

Figure 2: (a) Nodes in the enumeration tree (vertical axis in logarithmic scale) and (b) overall running time to optimality for Cplex and the branch and cut procedure over instances with $d = 2$, $p = 5$, and $q = 3$, and $n = 10, \ldots, 60$.

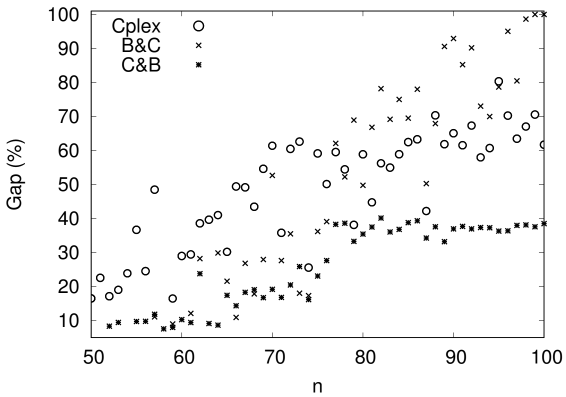

Figure 3: Optimality gaps after 600 seconds for Cplex, the branch and cut procedure, and a cut and branch procedure over instances with $d = 2$, $p = 5$, and $q = 3$, and $n = 50, \ldots, 100$.

number of nodes. The instances not shown could not be solved with optimality within the time limit. The execution of the linear-programming-based separation procedure imposes a non-negligible overhead to the overall branch and cut (B&C) procedure, which can be prohibitive for large instances, as Figure 3 shows. This figure represents the optimality gaps for instances not solved within the time limit, showing that at about $n = 80$ points Cplex achieves better optimality gaps than the branch and cut procedure. In this situation, a cut and branch (C&B) scheme (i.e., applying cuts only at the root node) is a better choice, consistently providing gaps below 40% for instances up to $n = 100$.

As mentioned in Section 4, we can set up one linear program (18)-(24) for each coordinate and warm-start the resolution by updating the coefficients in the objective function every time a new fractional solution must be separated. When this strategy is implemented, for an 18-point instance the total running time is reduced from 18.9 to 5.18 seconds. In our implementation, we set up one Cplex instance for each such linear program and we initialize

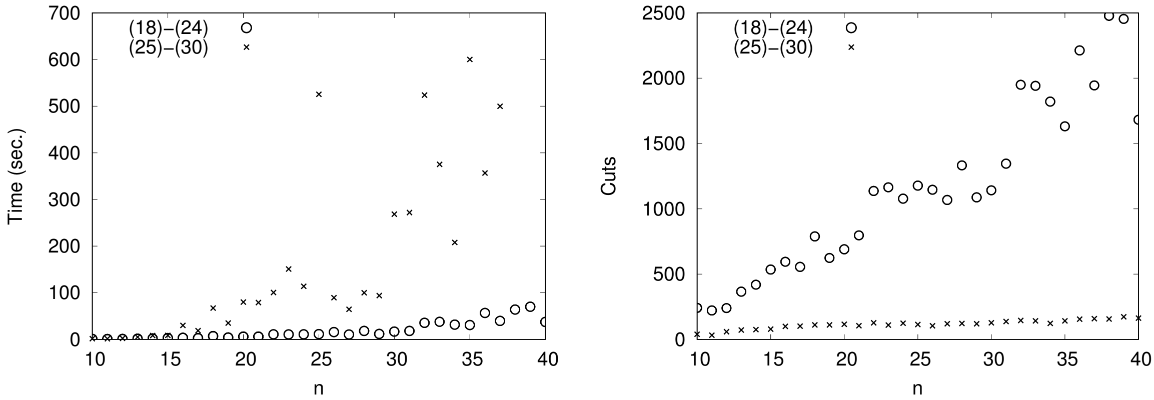

Figure 4: (a) Overall running time to optimality and (b) generated cuts when separating with the $O(n^2)$-sized linear programming model (18)-(24) and the $O(n)$-sized linear programming model (25)-(30) over instances with $d = 2$, $p = 5$, and $q = 3$, and $n = 10, \ldots, 40$.

the model with constraints (19)-(24). These Cplex instances are kept open throughout the overall branch and cut procedure. When the separation procedure is called, the objective function of each model is updated with the fractional solution to be separated, and the resolution is resumed until optimality is reached. Since these models are linear programs, this procedure is quite efficient. We could try to run each Cplex instance in a separate thread in order to make further gains but, in our implementation with a four-core CPU, this strategy was not effective. In the 18-point instance mentioned before, this threaded implementation needed 7.01 seconds to achieve optimality. This might be due to the additional overhead needed to control the threads or due to a collision between this mechanism and the threaded implementation of Cplex. Furthermore, this behavior did not seem to be affected by explicitly asking Cplex to use one thread or to use the Primal Simplex algorithm. This behavior was consistently observed in other instances and, due to this fact, we inactivated the use of threads in the separation procedure.

The previous experiments were performed with the linear program (18)-(24) for exactly separating the inequalities (12). Figure 4 reports a comparison between this procedure and the heuristic separation with the smaller linear program (25)-(30) for instances of different sizes. The exact separation finds a much larger number of cuts and helps to achieve a total running time to optimality that is smaller than the time to optimality obtained with the heuristic separation given by the formulation (25)-(30). The total number of nodes in the enumeration tree is also larger in this case. Due to these facts, in the remainder of this section we stick to the exact separation procedure given by the linear program (18)-(24).

The experiments reported in Figure 2 and Figure 3 suggest that it might be worthwhile to calibrate the cutting aggressiveness within the branch and cut procedure. To this end, we have performed an extensive number of runs with several instances with $d = 2$, $p = 5$, $q = 3$, and $n = 35$, generated with different pseudorandom seeds. Figure 5(a) shows the number of nodes in the enumeration tree as a function of the number of cut rounds per node in the enumeration tree (i.e., the maximum number of times that the separation procedure is sequentially executed at each node). The right-hand-side of this plot corresponds to applying 1 to 20 cut rounds per node, and the rightmost data points correspond to applying an unlimited

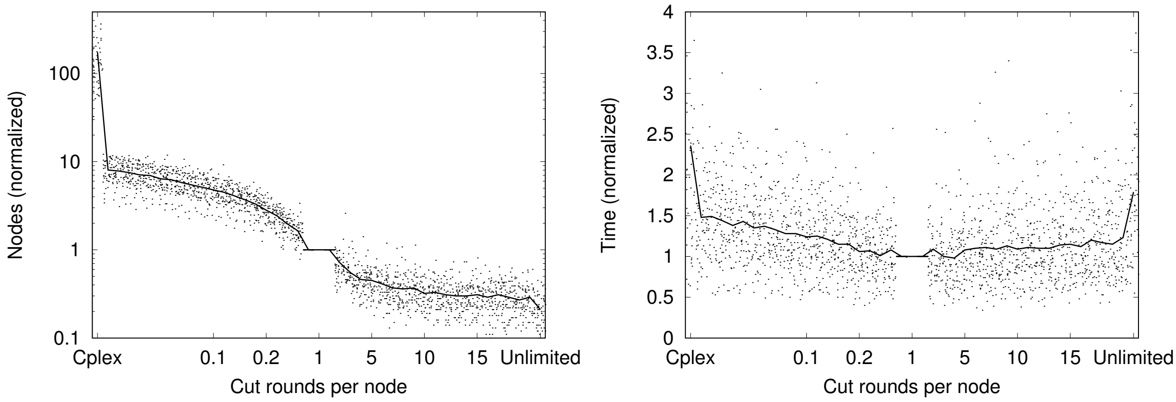

Figure 5: (a) Nodes in the enumeration tree (vertical axis in logarithmic scale) and (b) overall running time to optimality as a function of the cut rounds per node over instances with $d = 2$, $p = 5$, $q = 3$, and $n = 35$, normalized by the measurements for one cut round per node.

number of cut rounds, until no more cuts are found at each node. On the other hand, the left-hand-side of this plot corresponds to applying one round of cuts every certain number of nodes (a parameter sometimes known as *skip factor*). As an example, the value 0.2 in the horizontal axis corresponds to applying one round of cuts every 5 nodes. The leftmost data points correspond to applying no cut rounds, i.e., show the behavior of Cplex with out-of-the-box parameters. In order to properly compare measurements coming from different instances, all results are normalized by the measurements corresponding to one cut round per node, so these data points appear as 1 in this plot. Each point corresponds to an instance, and is sligthly perturbed in the horizontal axis in order to enhance the presentation. The solid line shows the average over all instances.

As Figure 5(a) shows, an aggresive cut strategy allows to greatly improve the number of nodes in the enumeration tree. However, this improvement in the number of nodes involves an additional computational cost and, in order to evaluate this tradeoff, Figure 5(b) shows the total time to achieve optimality in the same setting. In this case, it can be seen that the best-performing strategy consists in setting around one cut round per node in the enumeration tree. When exactly one cut round per node is implemented, the number of nodes in the enumeration tree is reduced 174.33 times with respect to the out-of-the-box algorithm implemented by Cplex, and the overall running time is reduced 2.41 times with respect to Cplex.

Adding a large number of cuts may have a negative impact on the overall performance due to the resulting enlargement of the linear programming models to be solved at each node in the enumeration tree. Although the previous results suggest that executing roughly one cut round per node is a good idea, it might be the case that the improvement is given only by the best cuts and not by the sheer number of added cuts. In order to evaluate this situation, we report experiments in which we discard a cut (12) separating the fractional solution (z^*, l^*, r^*) if $\alpha r^*_{tc} - \beta l^*_{tc} - \sum_{i=1}^{n} \gamma_i z^*_{ic} + \delta \geq -\tau$, for some parameter $\tau \geq 0$. The parameter τ specifies the minimum *cut depth* in order to add a cut. Figure 6 shows (a) the number of added cuts and (b) the total nodes in the enumeration tree as a function of τ for 20 similar instances generated with different seeds, normalized by the number of cuts for $\tau = 0$ for each instance, so the measurements are comparable. As expected, the number of cuts decreases and the

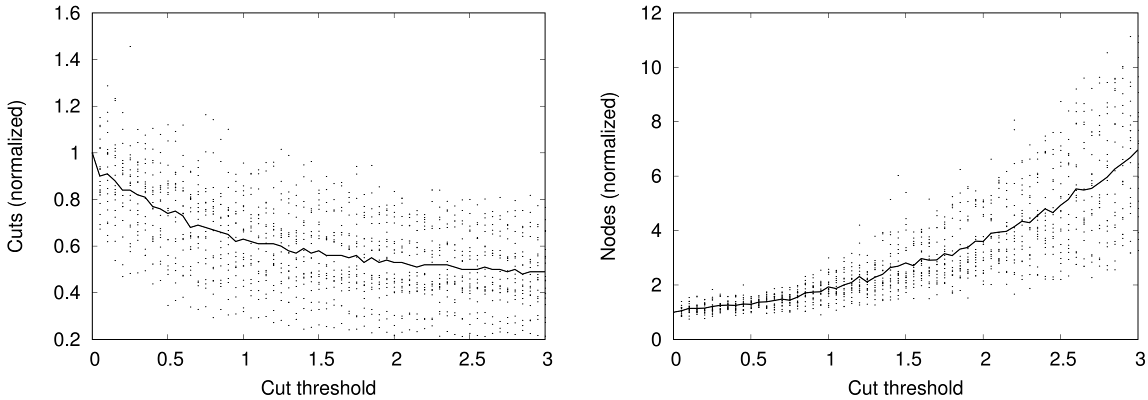

Figure 6: (a) Number of added cuts and (b) nodes in the enumeration tree for the branch and cut procedure (with one cut round per node) as a function of the cuth depth threshold τ over instances with $d = 2$, $p = 5$, $q = 3$, and $n = 30$, generated with 20 different seeds.

number of nodes increases as the threshold τ becomes more restrictive. Figure 7 reports the total running time to optimality (again, normalized by the measurement for $\tau = 0$) and, in this case, the overall trend suggests that taking $\tau \in [0.5, 0.8]$ is a reasonable choice for this parameter. A similar behavior was observed for instances generated with different parameters. This suggests that limiting the added cuts may not be a good strategy. Since the separation time is not negligible, discarding a cut that is not deep enough does not seem to be a good idea once the separation procedure has been performed.

As mentioned in Section 3, the addition of the symmetry-breaking constraints (9)-(11) to the formulation does not improve running times to optimality. Table 1 reports representative experiments with instances from 10 to 60 points, with $d = 2$, $p = 4$, and $q = 3$. For the original formulation and for the addition of each family of symmetry-breaking constraints, the column "Time/Gap" shows the total time to optimality in seconds and, if the time limit of 600 seconds is attained, the optimality gap is reported as a percenteage. On the other hand, the column "Nodes" reports the number of nodes in the enumeration tree. Quite surprisingly, this table suggests that the addition of symmetry-breaking constraints has a negative effect on the performance of the overall procedure. Similar results were obtained with instances generated with different parameters and settings for cut rounds per node.

As a final experiment, Figure 8 compares the overall running time to optimality and the number of nodes in the enumeration tree when the objective function (1) is replaced by the quadratic objective function asking to minimize the total area of the hyper-rectangles defining the clusters, namely $\sum_{c\in[p]}\prod_{t\in[d]}(r_{tc}-l_{tc})$. Cplex can handle such a formulation with both integer and continuous variables, and a quadratic objective function. As Figure 8(a) shows, running times to optimality are much higher in this case than with the linear objective function (1). Furthermore, although the dynamical addition of cuts (see Figure 8(b)) allows to decrease the total number of nodes in the enumeration tree, the overall running time is not greatly improved and is even increased in some cases.

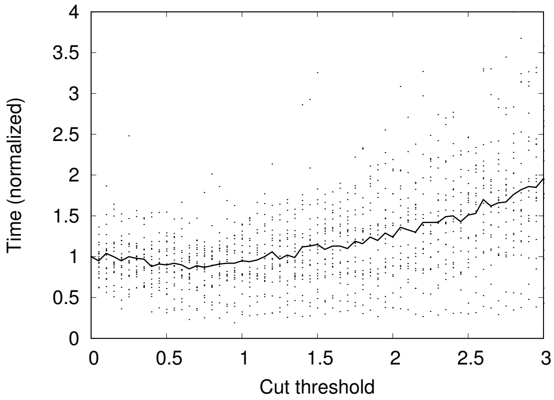

Figure 7: Time to optimality for the branch and cut procedure (with one cut round per node) as a function of the cut depth threshold τ over instances with $d = 2$, $p = 5$, $q = 3$, and $n = 30$, generated with 20 different seeds.

n	None		(9)		(10)		(11)	
	Time/Gap	Nodes	Time/Gap	Nodes	Time/Gap	Nodes	Time/Gap	Nodes
10	1.01	235	2.57	904	3.21	975	1.96	543
15	2.98	331	12.99	1755	15.43	2273	11.21	1037
20	6.14	413	46.36	2710	57.43	2679	56.47	2218
25	11.51	407	114.69	2979	148.07	4156	204.55	4016
30	16.67	482	110.69	2177	238.58	5786	291.99	3403
35	27.35	403	5.73%	2666	251.86	2560	5.68%	2539
40	46.64	382	6.10%	2071	13.72%	1951	7.44%	2057
45	104.59	819	17.98%	1267	29.92%	1553	32.41%	921
50	224.66	874	28.97%	1044	54.62%	1248	53.92%	820
55	257.34	909	42.71%	1355	74.55%	958	59.88%	684
60	1.05%	1168	62.97%	987	74.44%	864	98.54%	520
Avg.	1.05%	583.91	27.41%	1810.45	49.45%	2273.00	42.98%	1705.27

Table 1: Computational results with one cut round per node and with the addition of symmetry-breaking constraints, for $d = 2$, $p = 4$, and $q = 3$.

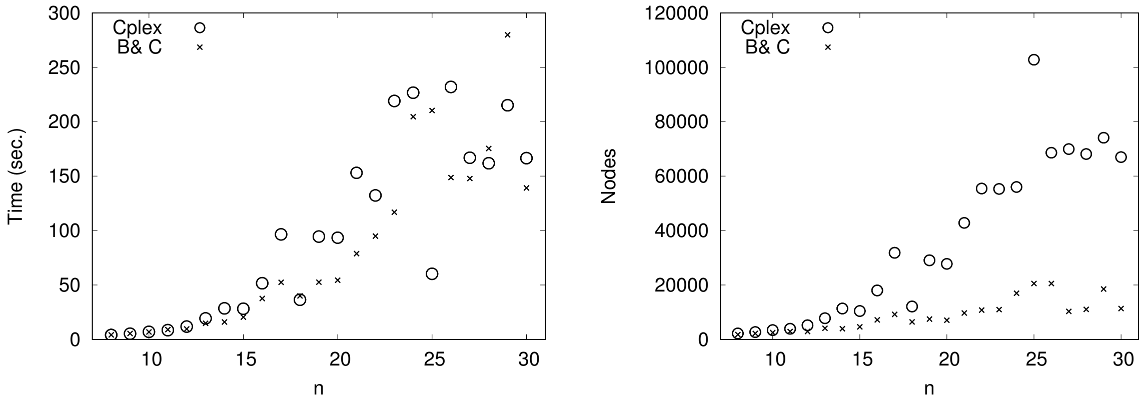

Figure 8: (a) Overall running time to optimality and (b) nodes in the enumeration tree for the quadratic objective function asking to minimize the total area of the hyper-rectangles defining the clusters, for Cplex with default parameters and with the dynamical addition of cuts, respectively, for $d = 2$, $p = 2$, and $q = 3$, and $n = 5, \dots, 30$.

6 Concluding remarks

In this work we have started a polyhedral study of the hyper-rectangular clustering problem with axis-parallel clusters and outliers, showing that the dynamical addition of cuts to a natural integer programming formulation of this problem may be effective in practice. The results in this work push the instance size solvable with optimality within a reasonable time a little further away, with respect to the use of a state-of-the-art integer programming solver. In this sense, it would be interesting to continue this study in order to explore how far can integer-programming-based techniques can reach at solving this problem.

As future work, it is important to consider better separation strategies, since the separation overhead degrades the overall performance for large instances. It would also be interesting to consider valid inequalities involving more than one cluster. Preliminary explorations of the associated polytopes show that they admit many facet-inducing inequalities with such properties, so it could be worthwhile to perform this study. If a potential family of such inequalities can be separated exactly with techniques similar to the ones presented in this work, this could also provide interesting practical benefits.

An undesirable property of the formulation (2)-(8) is the presence of symmetry among the clusters. Unfortunately, the addition of straightforward symmetry-breaking constraints does not seem to improve running times in our experiments, having in fact the opposite effect. In this setting, it may be worthwhile to explore column-generation-based procedures over extended formulations, although it is not clear how to branch within such a procedure and thus achieving optimality may turn out to be a nontrivial issue. The exploration of effective symmetry-breaking techniques for this formulation is of interest in this context, and is left as future work.

The size of instances solvable with optimality with the current techniques is rather small, and the above-mentioned potential lines for future research could make progress in this sense. However, optimality for large instances may possibly be out of reach. If this is the case, then it could also be relevant to explore integer-programming-based heuristics, maybe relying on the solution of the linear relaxation of a compact formulation or on a small subset of columns within an extended formulation. Given the interest in clustering methods, this line of research could be potentially very relevant for the data science community.

Acknowledgement. The author would like to express his gratitude towards the anonymous reviewers, whose comments and remarks greatly helped to improve this manuscript.

References

[1] Bertsimas, D., Dunn, J.: Optimal classification trees. Machine Learning 106 1039–1082 (2017).

[2] Bertsimas, D., Shioda, R.: Classification and regression via integer optimization. Operations Research 55-2 252–271 (2007).

[3] Bhatia, A., Garg, V., Haves, P., Pudi, V.: Explainable clustering using hyper-rectangles for building energy simulation data. IOP Conference Series: Earth and Environmental Science 238 012–068 (2019).

[4] Brandes, U., Holm, E., Karrenbauer, A.: Cliques in regular graphs and the core-periphery problem in social networks. COCOA 2016: Combinatorial Optimization and Applications 175–186 (2016).

[5] Carrizosa, E., Romero Morales, D.: Supervised classification and mathematical optimization. Computers and Operations Research 40-1 150–165 (2013).

[6] Gambella, C., Ghaddar, B., Naoum-Sawaya, J.: Optimization problems for machine learning: A survey. European Journal of Operational Research 290-3 807–828 (2021).

[7] Gao, B.: Hyper-rectangle-based discriminative data generalization and applications in data mining. PhD Thesis, Simon Fraser University, 2007.

[8] Lee, S., Chung, C.: Hyper-rectangle based segmentation and clustering of large video data sets. Information Sciences 141 (1-2) 139–168 (2002).

[9] Mago, V., Bhatia, N., Park, S.: Classification with axis-aligned rectangular boundaries. In: Mago, V., Bhatia, N. (eds.) "Cross-Disciplinary Applications of Artificial Intelligence and Pattern Recognition", Information Science Reference (2012).

[10] Ordóñez, C., Omiecinski, E., Navathe, S., Ezquerra, N.: A clustering algorithm to discover low and high density hyper-rectangles in subspaces of multidimensional data. Georgia Institute of Technology Technical Report GIT-CC-99-20 (1999).

[11] Park, S., Kim, J.: Unsupervised clustering with axis-aligned rectangular regions. Stanford University Technical Report (2009).

[12] Verwer, S., Zhang, Y.: Learning optimal classification trees using a binary linear program formulation. Proceedings of the AAAI Conference on Artificial Intelligence 33-01 1625–1632 (2019).