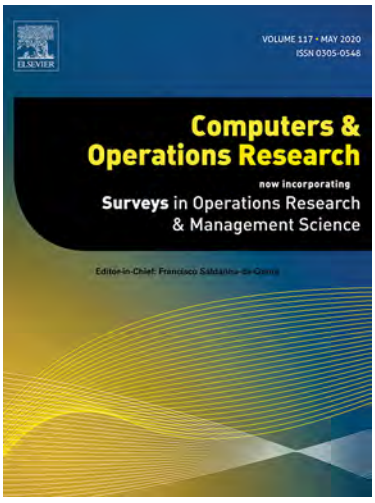
Journal Pre-proofs



On the multisource hyperplanes location problem to fitting set of points

Víctor Blanco, Alberto Japón, Diego Ponce, Justo Puerto

PII: S0305-0548(20)30241-0
DOI: https://doi.org/10.1016/j.cor.2020.105124
Reference: CAOR 105124

To appear in: *Computers and Operations Research*

Received Date: 4 February 2020
Accepted Date: 7 October 2020

Please cite this article as: V. Blanco, A. Japón, D. Ponce, J. Puerto, On the multisource hyperplanes location problem to fitting set of points, *Computers and Operations Research* (2020), doi: https://doi.org/10.1016/j.cor.2020.105124

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier Ltd.

Journal Pre-proofs

On the multisource hyperplanes location problem to fitting set of points

Víctor Blanco[a], Alberto Japón[b], Diego Ponce[b], Justo Puerto[b]

[a] *IEMath-GR, Universidad de Granada*
[b] *IMUS, Universidad de Sevilla*

Abstract

In this paper we study the problem of locating a given number of hyperplanes minimizing an objective function of the closest distances from a set of points. We propose a general framework for the problem in which norm-based distances between points and hyperplanes are aggregated by means of ordered median functions. A compact Mixed Integer Linear (or Non Linear) programming formulation is presented for the problem and also an extended set partitioning formulation with a huge number of variables is derived. We develop a column generation procedure embedded within a branch-and-price algorithm for solving the problem by adequately performing its preprocessing, pricing and branching. We also analyze geometrically the optimal solutions of the problem, deriving properties which are exploited to generate initial solutions for the proposed algorithms. Finally, the results of an extensive computational experience are reported. The issue of scalability is also addressed showing theoretical upper bounds on the errors assumed by replacing the original datasets by aggregated versions.

Keywords: Hyperplanes Location, Mixed Integer Non Linear Programming, Column Generation.
2010 MSC: 52C35, 90B85, 90C11, 90C30.

1. Introduction

Location Analysis deals with the determination of the *optimal* positions of facilities to satisfy the demand of a set of customers. The problems analyzed in the field are diverse but can be usually classified as: Discrete Location problems (DLP) and Continuous Location problems (CLP). In the first family, a set of potential facilities is previously given and the goal is to select, among them, the optimal ones under one or more criteria. The main tools for solving these problems come from Discrete Optimization, or more precisely, from Integer Linear Programming. In the second family of problems, the facilities have to be located in a continuous space and then, convex analysis and global optimization tools are needed to solve the problems. The most popular problem in the latter family is the Weber problem (Weber, 1909) in which a single point-facility has to be positioned on the plane so as to minimize the overall sum of the (Euclidean) distances to a set of (planar) demand points. The applications of both types of location problems are vast. DLP are more common in the location of *physical* facilities (as ATMs, supermarkets, stations, etc), while CLP are more useful

Email addresses: vblanco@ugr.es (Víctor Blanco), ajapon1@us.es (Alberto Japón), dponce@us.es (Diego Ponce), puerto@us.es (Justo Puerto)

Journal Pre-proofs

when locating facilities in telecommunication networks (as wifi routers, servers, etc) or even to provide the sets of potential facilities for a DLP.

In this paper we study a problem that falls into the family of CLP. More specifically, we focus on the determination of *optimal* hyperplanes fitting a given finite set of demand points. The location of a single hyperplane is a classical problem that has been addressed in different fields. On the one hand, this problem clearly extends the classical Weber problem, but where instead of locating zero-dimensional facilities one looks for locating higher dimensional structures. On the other hand, in Statistics and Data Analysis, the determination of a hyperplane minimizing the sum of squares of vertical residuals is key for estimating a multivariate linear regression model using the Least Sum of Squares (LSS) method, credited to Gauss (1809). One can also find recent useful applications, both in Location Science and Data Analysis, for the problem of finding *optimal* hyperplanes fitting a set of points. For instance, Espejo and Rodríguez-Chía (2011) deals with the location of a rapid transit line on the plane to be used as an alternative transportation mean. Analogously, the widely used Support Vector Machine (SVM) methodology due to Cortes and Vapnik (1995), is also based on constructing a hyperplane minimizing certain loss functions of the distances to a given set of points.

Scanning the literature one can find that most of the attention has been devoted to finding hyperplanes with any of the following assumptions (see e.g., (Martini and Schöbel, 1998; Schöbel, 1999, 2003, 2015; Martini and Schöbel, 2001; Plastria and Carrizosa, 2001; Brimberg et al., 2002, 2003; Blanco et al., 2018; Bradley and Mangasarian, 2000)): (a) the problem is embedded on the plane; (b) a single hyperplane has to be located; (c) the vertical distance between each point and the hyperplane is considered; or (d) the residuals are aggregated by the sum or the maximum operators. Our goal here is to study a generalization of this problem in which, we construct simultaneously a given number, p, of hyperplanes in any finite dimensional space, $\mathbb{R}^d$, by minimizing a rather general globalizing function, an ordered median function, of the residuals from the points to the fitting bodies. Ordered median functions aggregate the set of distances from the demand points to their closest hyperplanes (residuals) by means of a sorting, weighting averaging operation: distances are sorted and then their weighted sum is performed. The sum and maximum functions can be easily represented as ordered median functions with adequate choices of the weights inducing the median and center objective functions. Also, the k-centrum (sum of the k-th largest distances) or the cent-dian (convex combination of the sum and the max criteria) can be cast within this family of functions. In addition, different point-to-hyperplane norm-based distances are considered as a measure of the residuals of the fitting. Thus, this paper naturally extends the analysis performed in Blanco et al. (2018) where the location of a single ordered median hyperplane was studied.

As in the classical Weber problem (Weber, 1909), the extension from the location of one to several facilities (the so-called multisource problem) is not trivial (Blanco et al., 2016). Actually, while the classical single-facility point location problem with standard distances (ℓ_τ, polyhedral, etc) can be formulated as a Second Order Cone programming problem (Blanco et al., 2014) (being then polynomially solvable), its multisource version becomes a non-convex NP-hard problem (Blanco et al., 2016).

In the case of locating hyperplanes, the situation is even harder, since the location of a single hyperplane is, in general, an NP-hard problem (see Blanco et al. (2018)) whose exact solution can be obtained, using Mixed Integer Linear Programming, for vertical and polyhedral norm based residuals, while for general ℓ_τ-based residuals one has to resort to global optimization tools.

The problem considered in this paper is not fully new although, in our opinion, it has not been fully analyzed and there is room for further improvement. In particular, similar problems have

Journal Pre-proofs

been analyzed from the Data Analysis field, and different names have been adopted. In the so-called *Clusterwise Linear Regression* (CLR) problem, a set of observations is provided and the goal is to cluster them by means of the sum of the squared residuals of several multivariate regression models (Späth, 1982; Hennig, 1999; Carbonneau et al., 2014; Park et al., 2017; Gitman et al., 2018). In (Bertsimas and Shioda, 2007), classification and regression are simultaneously performed, and also clustering by classical linear regression approaches. Finally, in (Bradley and Mangasarian, 2000), the clusters are constructed based on the closest distances to *optimal* hyperplanes in a given d-dimensional space. In the so-called *Piecewise Linear Regression* problem, a dependent variable is partitioned into p intervals and it adjusts linear bodies to each of them (see (McGee and Carleton, 1970)). However, only local search heuristic algorithms have been proposed for these problems, alternating clustering and regression techniques sequentially. Carbonneau et al. (2014) present a column generation algorithm for the (planar) clusterwise regression problem with sum of squared residuals which combined with some heuristic strategies outperforms previous results in the literature. Moreover, Park et al. (2017) generalized the clusterwise regression problem by allowing each entity to have more than one observation and propose an exact mathematical programming-based approach relying on column generation, and several heuristics.

The main contributions of this paper are:

1. To provide a general framework for the simultaneous location of several hyperplanes to fit a data set using mathematical programming tools. We formulate the problem by using general norm-based error measures of the distance from points to hyperplanes and ordered median functions to aggregate the residuals. This approach generalizes both the standard multisource regression (Carbonneau et al., 2014; Park et al., 2017) and also the more recent proposal for the $p = 1$ case (Blanco et al., 2018).
2. To develop two exact solution methods:
   (a) One based on a compact formulation, that for vertical residuals (resulting in a Mixed Integer Second Order Cone Optimization problem) and for polyhedral norm-based residuals (resulting in a Mixed Integer linear Programming Problem) can be solved by using some of the available off-the-shell solvers.
   (b) A novel branch-and-price algorithm, based on a set partitioning formulation for the problem, combining several features as preprocessing, exact and heuristic pricing, and Ryan-and-Foster branching.
3. To prove some geometrical characterizations of ordered median optimal hyperplanes that are incorporated in the preprocessing phase of our column generation approach.
4. To compare the proposed approaches on a extensive battery of computational experiments on both real and synthetic instances.
5. To derive upper bounds on the error assumed by aggregation procedures on original datasets that allow to control the scalability of the proposed approaches.

The rest of the paper is organized as follows. In Section 2 we introduce the problem and fix the notation for the rest of the sections. This section also contains two illustrative examples taken from the literature. Section 3 is devoted to a first compact formulation for the problem. This formulation has a polynomial number of variables and constraints but its performance is not always good since it has a large integrality gap. For that reason, in Section 4 we develop an alternative set partitioning formulation that is solved (exactly, for vertical and polyhedral-norm based residuals) within a branch-and-price (B&P) algorithm using column generation at each node of the branching tree. This section describes all the elements of this B&P: initialization, pricing (exact and heuristic)

Journal Pre-proofs

and branching. Section 5 reports our computational results based on two different datasets: the classical 50 points dataset by Eilon et al. (1971) and another synthetic dataset randomly generated. Section 6 is devoted to explore scalability issues and finally Section 7 draws some conclusions and future extensions.

2. Multisource Location of Hyperplanes

In this section we describe the problem under study and fix the notation for the rest of the paper.

We are given a set of n observations/demand points (denoted as points from now on) in $\mathbb{R}^d$, $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and $p \in \mathbb{Z}_+$ $(p > 0)$. Our goal is to find p hyperplanes in $\mathbb{R}^d$ that minimize an objective function of the closest distances from points to hyperplanes. We denote the index sets of demand points and hyperplanes by $I = \{1, \ldots, n\}$ and $J = \{1, \ldots, p\}$, respectively. Given $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}$, we denote by

$$\mathcal{H}(\boldsymbol{\beta}, \alpha) = \{y \in \mathbb{R}^d : \boldsymbol{\beta}^t y + \alpha = 0\},$$

the hyperplane in $\mathbb{R}^d$ with coefficients $\boldsymbol{\beta}$ and intercept α (here v^t stands for the transpose of the vector $v \in \mathbb{R}^d$).

Several elements are involved when finding the *best* p hyperplanes to fit a set of demand points. In what follows we describe them:

- *Residuals:* The point-to-hyperplane measure of closeness. Given a demand point $a = (a_1, \ldots, a_d) \in \mathbb{R}^d$ and a hyperplane $\mathcal{H}(\boldsymbol{\beta}, \alpha)$, how far/close is the point from the hyperplane? The classical fitting methods use the so-called *vertical-distance* measure, which given a reference coordinate, say the d-th, computes the deviation $a_d + \frac{\alpha}{\beta_d} + \sum_{\ell=1}^{d-1} \frac{\beta_\ell}{\beta_d} a_\ell$, whenever $\beta_d \neq 0$. However, it has been already proposed that the use of more general distance measures based on norms may be advisable. In particular, some authors (see e.g., Blanco et al. (2018, 2020b)) have shown the usefulness of norm-based distances, such as polyhedral, or ℓ_τ-distances $(\tau \geq 1)$. Among them, we mention, for their importance, the Manhattan (ℓ_1-norm), the Tchebyshev (ℓ_∞-norm) or the Euclidean (ℓ_2-norm) distances.

  Thus, for a point $a \in \mathbb{R}^d$ and a hyperplane $\mathcal{H}(\boldsymbol{\beta}, \alpha)$, we consider the residual from a to $\mathcal{H}(\boldsymbol{\beta}, \alpha)$ as:

  $$\varepsilon_a(\boldsymbol{\beta}, \alpha) = \mathrm{D}\Big(a, \mathcal{H}(\boldsymbol{\beta}, \alpha)\Big) := \min\{\mathrm{D}(a, y) : y \in \mathcal{H}(\boldsymbol{\beta}, \alpha)\},$$

  where D is a norm-based distance or the vertical distance in $\mathbb{R}^d$ (see Mangasarian (1999); Blanco et al. (2018) for further details on this projection).

- *Allocation Rule*: Given a set of hyperplanes and a point, once the residuals to each of the hyperplanes are calculated, one has to allocate the point to a single hyperplane. Different alternatives can be considered, as the allocation to the closest or the furthest hyperplane. In our framework we assume, as usual in Location Analysis, that each point is allocated to the hyperplane with the smallest residual, i.e., for a point $a \in \mathbb{R}^d$ and an arrangement of hyperplanes $\mathbb{H} = \Big\{\mathcal{H}(\boldsymbol{\beta}_j, \alpha_j) : j \in J\Big\}$, the final residual point-to-hyperplanes is computed as:

  $$\varepsilon_a\Big(\mathbb{H}\Big) = \min_{j \in J} \varepsilon_a(\boldsymbol{\beta}_j, \alpha_j),$$

  and the hyperplane, $\mathcal{H}(\boldsymbol{\beta}_j, \alpha_j)$, reaching such a minimum is the one where a is allocated to (in case of ties among more than one hyperplane, a random assignment is performed).

Journal Pre-proofs

- *Aggregation of Residuals*: Given a set of points and an arrangement of hyperplanes, once the residuals are computed with respect to the arrangement, and in order to find the p hyperplanes that best fit the n data points, a global measure of goodness must be chosen for aggregating the residuals. The classical aggregation functions are the sum or maximum of squared residuals. Most of these criteria can be cast within the framework of the family of ordered median aggregation criteria. More explicitly, given $x_1, \ldots, x_n \in \mathbb{R}^d$, an arrangement of hyperplanes $\mathbb{H} = \left\{\mathcal{H}(\boldsymbol{\beta}_j, \alpha_j) : j \in J\right\}$, and $\lambda \in \mathbb{R}^n_+$ (with $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$) the λ-ordered median function is defined as:

$$\mathsf{OM}_\lambda(\varepsilon_1, \ldots, \varepsilon_n) = \sum_{i=1}^{n} \lambda_i \, e_{(i)}, \tag{OMF}$$

  where $e_{(1)}, \ldots, e_{(n)}$ are defined such that $e_{(i)} \in \{\varepsilon_{x_1}(\mathbb{H}), \ldots, \varepsilon_{x_n}(\mathbb{H})\}$ for all $i \in I$ and $e_{(1)} \geq \cdots \geq e_{(n)}$. Observe that particular cases of ordered median problem are the sum ($\lambda_i = 1$, $i = 1, \ldots, n$), the maximum ($\lambda_1 = 1$, $\lambda_i = 0$, $i \neq 1$), the k-centrum ($\lambda_i = 1$, $i = 1, \ldots, k$, $\lambda_j = 0$, $j > k$) or the ρ-centdian, a convex combination of sum and max criterion ($\lambda_1 = 1, \lambda_i = \rho, i = 2, \ldots, n$), for $0 < \rho < 1$.

Summarizing all the above considerations, the Multisource Ordered Median Fitting Hyperplanes Problem (MOMFHP) can be stated as the problem of finding $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p \in \mathbb{R}^d$ and $\alpha_1, \ldots, \alpha_p \in \mathbb{R}$ solving the following optimization problem:

$$\begin{aligned} \min & \sum_{i \in I} \lambda_i \, e_{(i)} \\ \text{s.t. } & e_i \geq \min_{j \in J} \varepsilon_{x_i}(\boldsymbol{\beta}_j, \alpha_j), \forall i \in I, \\ & \boldsymbol{\beta}_j \in \mathbb{R}^d, \alpha_j \in \mathbb{R}, \forall j \in J, \\ & e_i \geq 0, \forall i \in I. \end{aligned} \tag{MOMFHP$_0$}$$

where e_i represents the residual for the i-th point in the data set, for all $i \in I$.

(MOMFHP) appears when different trends or clouds have to be differentiated on the demand points, and then, different hyperplanes want to be used to fitting to the points, such that the global error assumed, when the points are allocated to their closest hyperplanes, is as small as possible. In Figure 1 we illustrate a set of demand points in the plane which could be clustered into three groups according to different linear trends which are drawn in gray color. In the following example we illustrate the problem under analysis in two classical instances.

Example 2.1. *In the seminal paper by McGee and Carleton (1970), the authors illustrate the Clusterwise Linear Regression method with two instances. The first instance, (Quandt, 1958), consists of* 20 *points on the plane,* $\{x_1, \ldots, x_{20}\}$ *generated as follows:*

$$\begin{aligned} x_{i2} &= 2.5 + 0.7x_{i1} + \epsilon_i, \text{for } i = 1, \ldots, 12, \text{ and} \\ x_{i2} &= 5 + 0.5x_{i1} + \epsilon_i, \text{for } i = 13, \ldots, 20, \end{aligned}$$

where ϵ *is randomly generated as a univariate normal distribution with mean* 0 *and standard deviation* 1*.*

We run our model with this dataset choosing as residuals the ℓ_1*-norm projection of the data onto the hyperplanes, and four different ordered median criteria: Weber, Center,* $\lceil \frac{n}{2} \rceil$*-Centrum*

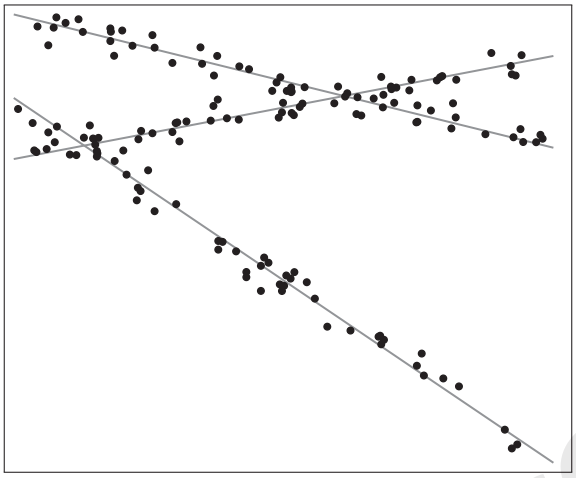


Figure 1: Illustration of a feasible solution of our problem for a set of demand points.

($\lambda = (\overbrace{1,\ldots,1}^{\lceil \frac{n}{2} \rceil},0,\ldots,0)$) and 0.9-centdian ($\lambda = (1, 0.9, \ldots, 0.9)$). The results are shown in Figure 2.

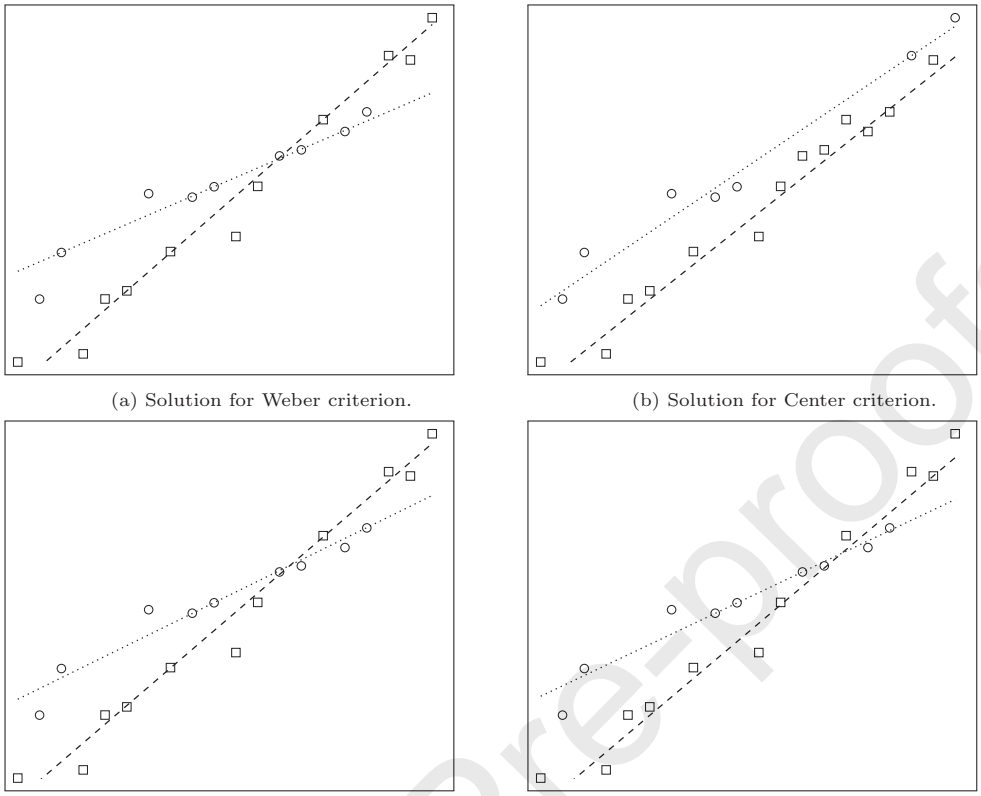
McGee and Carleton (1970) also analyzed a real instance, the `Boston` *dataset. It was motivated by the fact that regional stock exchanges were hurt by the abolition of give-ups in 1968. The model tries to analyze the dollar volume of sales on the Boston Stock Exchange with respect to dollar volumes for the New York and American Stock Exchanges, based on a dataset with* 35 *monthly observations from January 1967 to November 1969. One can observe, in the results shown in (Figure 3), that our models are able to adequately cast the trends of these observations.*

3. A Compact Formulation for (MOMFHP$_0$)

In this section we provide a mathematical programming formulation for (MOMFHP$_0$). The main components which involve decisions in this problem, and that have to be adequately included in a suitable formulation, are the representation of general norm-based residuals and the aggregation of residuals using an ordered median function. We describe here how to incorporate all these elements into a mathematical programming formulation which in many cases is suitable to be solved with any of the available MILP/MISOCO solvers.

Theorem 3.1. *Let $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, $p \in \mathbb{Z}_+$ ($p > 0$) and $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. Then,* (MOMFHP$_0$)

Journal Pre-proofs



(a) Solution for Weber criterion.

(b) Solution for Center criterion.

(c) Solution for k-Centrum criterion.

(d) Solution for Centdian criterion.

Figure 2: Lines obtained for Quandt dataset for ℓ_1-norm residuals and different criteria.

can be equivalently reformulated as follows:

$$\min \sum_{k\in I} u_k + \sum_{i\in I} v_i \tag{MOMFHP}$$

$$s.t.\ u_k + v_i \geq \lambda_k e_i, \forall i,k \in I, \tag{1}$$

$$e_i \geq \varepsilon_{x_i}(\boldsymbol{\beta}_j, \alpha_j) - M_{ij}(1 - z_{ij}), \forall i \in I, j \in J, \tag{2}$$

$$\sum_{j=1}^{p} z_{ij} = 1, \forall i \in I, \tag{3}$$

$$z_{ij} \in \{0,1\}, \forall i \in I, j \in J, \tag{4}$$

$$e_i \in \mathbb{R}_+, \forall i \in I, \tag{5}$$

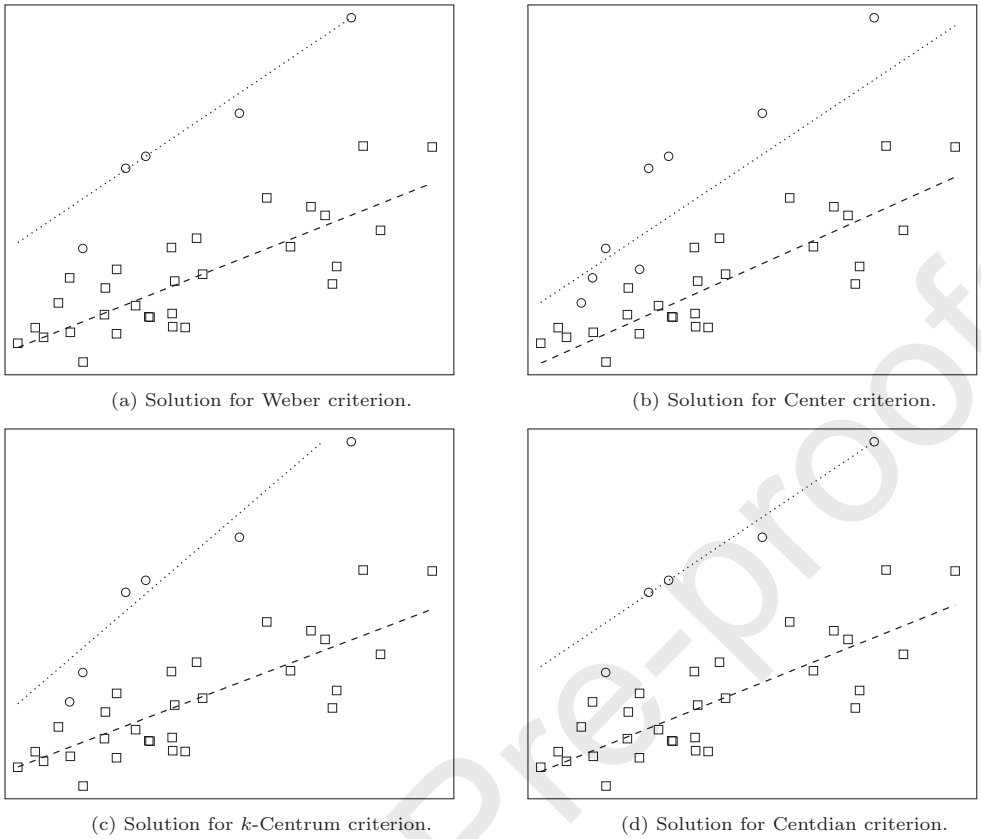
$$\boldsymbol{\beta}_j \in \mathbb{R}^d, \alpha_j \in \mathbb{R}, \forall j \in J, \tag{6}$$

$$u_k, v_i \in \mathbb{R}, \forall i,k \in I. \tag{7}$$

where M_{ij} are upper bounds on the residual values $\varepsilon_{x_i}(\boldsymbol{\beta}_j, \alpha_j)$, for all $i \in I, j \in J$.

Proof. First, observe that given a set of residuals $e_1, \ldots, e_n \geq 0$, the evaluation of the objective

Journal Pre-proofs



(a) Solution for Weber criterion.

(b) Solution for Center criterion.

(c) Solution for k-Centrum criterion.

(d) Solution for Centdian criterion.

Figure 3: Lines obtained for Boston dataset for ℓ_1-norm residuals and different criteria.

function in (MOMFHP$_0$) requires sorting and averaging them (the residuals) with the λ-weights. In Blanco et al. (2014), the authors proved that the computation of $\sum_{k=1}^{n} \lambda_k e_{(k)}$ can be done by means of the optimal value of the following Linear Programming Problem (see Blanco et al. (2014)):

$$\sum_{k\in I} \lambda_k e_{(k)} = \left\{ \begin{array}{ll} \min & \sum_{k\in I} u_k + \sum_{i\in I} v_i \\ \text{s.t.} & u_k + v_i \geq \lambda_k e_i \quad \forall k, i \in I, \\ & u, v \in \mathbb{R}^n. \end{array} \right.$$

Thus, the objective function in (MOMFHP$_0$) can be replaced by the above objective function and the constraints incorporated to the rest of constraints in the model.

In order to identify the point-to-hyperplane allocation we consider the following set of binary variables:

$$z_{ij} = \left\{ \begin{array}{ll} 1 & \text{if the } i\text{-th observation is assigned to } \mathcal{H}(\boldsymbol{\beta}_j, \alpha_j), \\ 0 & \text{otherwise,} \end{array} \right.$$

for all $i \in I$ and $j \in J$.

Journal Pre-proofs

Note that with our allocation rule, an observation can be always assigned to a hyperplane that reaches the minimum residual among all the possible assignments to the p hyperplanes.

Finally, using the variables previously described, the objective function computes the ordered median function of the residuals. Constraints (2) assure the correct definition of the residuals e_i and the allocation to their correct hyperplane. Indeed, if $z_{ij} = 1$ this constraint forces e_i to take the value of $\varepsilon_{x_i}(\boldsymbol{\beta}_j, \alpha_j)$. Constraints (3) assure that only one of these variables will be equal to 1, which in turns forces by the minimization character of the objective function to be the one with the correct assignment. Finally, (4)–(7) are the domains of the variables. □

Remark 3.2. *Observe that the different choices of ordered median functions are embedded into constraint* (1)*. In some particular cases, this formulation can be simplified avoiding useless variables and constraints.*

- p**-Median Problem** *(*$\lambda = (1, \ldots, 1)$*) In this case, since the ordering does not affect the aggregation operator, the u and v-variables can be avoided, and the problem simplifies to:*

$$\begin{aligned} \min \quad & \sum_{i \in I} e_i \\ \text{s.t.} \quad & (2) - (6). \end{aligned}$$

- p**-Center Problem** *(*$\lambda = (1, 0, \ldots, 0)$*): For the Center problem, one can represent the objective function,* $\max_{i \in I} e_i$*, by using an auxiliary variable,* t*, in the usual manner:*

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & (2) - (6), \\ & t \geq e_i, \forall i \in I, \end{aligned}$$

- p-k**-Center Problem** *(*$\lambda = (\overbrace{1, \ldots, 1}^{k}, 0, \ldots, 0)$*): For the k-Centrum problem, in Ogryczak and Tamir (2003) the authors derive a formulation similar to the one for the center problem:*

$$\begin{aligned} \min \quad & k\,t + \sum_{i \in I} r_i \\ \text{s.t.} \quad & (2) - (6), \\ & r_i \geq e_i - t, \forall i \in I, \\ & t \geq 0, \\ & r_i \geq 0, \forall i \in I. \end{aligned}$$

Note also that the explicit expression of $\varepsilon_{x_i}(\boldsymbol{\beta}_j, \alpha_j)$ and then, the difficulty of the optimization problem above, depends (apart from the binary variables that appears in the problem) on the choice of the distance measure D which defines the residuals of the fitting. In what follows we describe general shapes for the distances inducing the residuals and how they can be incorporated to (MOMFHP).

Journal Pre-proofs

3.1. Vertical Distance Residuals

Although not rigorously a distance measure, the so-called *vertical distance* is a very common measure for computing the residuals in Data Analysis. The vertical distance is computed as the absolute deviation, with respect to one of the coordinates, of the hyperplane. Without loss of generality, we consider that the deviation is computed with respect to the d-th coordinate, and then, one can assume that $\beta_{jd} = -1$ for $j \in J$. Given $a \in \mathbb{R}^d$ and a hyperplane $\mathcal{H}(\alpha, \boldsymbol{\beta})$ the vertical distance residual is calculated as:

$$\varepsilon_x(\boldsymbol{\beta}, \alpha) = \left| a_d - \alpha - \sum_{\ell=1}^{d-1} \boldsymbol{\beta}_\ell a_\ell \right|.$$

This measure can be incorporated to (MOMFHP), replacing (2) by the following set of linear constraints:

$$e_i \geq x_{id} - \alpha_j - \sum_{\ell=1}^{d-1} \boldsymbol{\beta}_{j\ell} x_{i\ell} - M_{ij}(1 - z_{ij}), \forall i \in I, j \in J,$$
$$e_i \geq -x_{id} + \alpha_j + \sum_{\ell=1}^{d-1} \boldsymbol{\beta}_{j\ell} x_{i\ell} - M_{ij}(1 - z_{ij}), \forall i \in I, j \in J.$$

Thus, becoming (MOMFHP) a Mixed Integer Linear Programming problem.

Remark 3.3 (Support Vector Regression)**.** *One particular case in which vertical residuals are used in Machine Learning tools is in Support Vector Regression (SVR). Vapnik (2013) proposed this methodology for obtaining regression models based on Support Vector Machines as introduced in Cortes and Vapnik (1995). The method is based on fitting a hyperplane to the set of points* $\{x_1, \ldots, x_n\}$ *with a modified vertical distance, such that only the residuals greater than a given threshold* $\Delta \geq 0$ *are accounted, apart from maximizing the separation between the observations at each of the sides of the hyperplanes. SVR can be modeled as follows:*

$$\begin{aligned} \min & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C \sum_{i \in N} e_i \\ \text{s.t. } & e_i \geq \left| x_{id} - \sum_{\ell=1}^{d-1} \boldsymbol{\beta}_k x_{i\ell} - \alpha \right| - \Delta, \forall i \in I, \\ & \boldsymbol{\beta} \in \mathbb{R}^{d-1}, \alpha \in \mathbb{R}, \\ & e_i \geq 0, \forall i \in I, \end{aligned}$$

where C *is a given parameter.*

Observe that the measure used in this approach is nothing but a truncated version of the vertical distance:

$$\varepsilon_a(\boldsymbol{\beta}, \alpha) = \begin{cases} |a_d - \alpha - \sum_{\ell=1}^{d-1} \boldsymbol{\beta}_\ell a_\ell| & \text{if } |a_d - \alpha - \sum_{\ell=1}^{d-1} \boldsymbol{\beta}_\ell a_\ell| > \Delta, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, this shape of the residuals can also be embedded in our multisource framework, just by adding to the objective functions the terms measuring the norms of the coefficients of the hyperplanes, i.e.,

replacing the objective function in (MOMFHP) *by*

$$\frac{1}{2}\sum_{j\in J}\|\boldsymbol{\beta}_j\|_2^2+\sum_{i\in I}\lambda_i\, e_{(i)}.$$

In case $p=1$ *and* $\lambda=(1,\ldots,1)$*, we obtain classical SVR taking also into account the parameter* Δ*, but more flexible counterparts can be generated with our framework.*

3.2. Norm-based Residuals

For general norm-based distances, a given observation $y^t=(y_1,\ldots,y_d)$ and a set of p hyperplanes defined by $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_p\in\mathbb{R}^d$ and $\alpha_1,\ldots,\alpha_p\in\mathbb{R}$ inducing the arragement $\mathbb{H}=\left\{\mathcal{H}(\boldsymbol{\beta}_j,\alpha_j):j\in J\right\}$, based on (Mangasarian, 1999, Theorem 2.1), the projection, $\hat{y}$, of y consistent with the residual ε induced by a norm $\|\cdot\|$ is

$$\hat{y}=y_{-0}-\min_{j\in J}\frac{\alpha_j+\boldsymbol{\beta}_j^t y}{\|(\boldsymbol{\beta}_{j1},\ldots,\boldsymbol{\beta}_{jd})\|^*}\kappa(\boldsymbol{\beta}_j),$$

where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$ and $\kappa(\boldsymbol{\beta})=\arg\max_{\|z\|=1}(\boldsymbol{\beta}_{j1},\ldots,\boldsymbol{\beta}_{jd})^t z$. Moreover, the residuals can be written as:

$$\varepsilon_y(\mathbb{H})=\min_{j\in J}\frac{|\alpha_j+\boldsymbol{\beta}_j^t y|}{\|(\boldsymbol{\beta}_{j1},\ldots,\boldsymbol{\beta}_{jd})\|^*}. \tag{8}$$

Remark 3.4 (ℓ_1-norm case)**.** *In the case of the* ℓ_1*-norm residuals, the expression above, for the hyperplane* $\mathcal{H}(\boldsymbol{\beta}_j,\alpha_j)$*, reduces to:*

$$\varepsilon_y(\boldsymbol{\beta}_j,\alpha_j)=\frac{|\alpha_j+\boldsymbol{\beta}_j^t y|}{\max_{\ell=1,\ldots,d}|\boldsymbol{\beta}_{j\ell}|}, \tag{9}$$

Journal Pre-proofs

Journal Pre-proofs

and constraints (2) *can be replaced in* (MOMFHP) *by:*

$$e_i \geq \alpha_j + \sum_{\ell=1}^{d} \boldsymbol{\beta}_{j\ell} x_{i\ell} - M_{ij}(1 - z_{ij}), \forall i \in I, \forall j \in J \tag{10}$$

$$e_i \geq -\alpha_j - \sum_{\ell=1}^{d} \boldsymbol{\beta}_{j\ell} x_{i\ell} - M_{ij}(1 - z_{ij}), \forall i \in I, \forall j \in J \tag{11}$$

$$\boldsymbol{\beta}_{j\ell} = \eta^+_{j\ell} - \eta^-_{j\ell}, \forall j \in J, \ell = 1, \ldots, d, \tag{12}$$

$$\eta^+_{j\ell} \leq U_{j\ell}\, \xi_{j\ell}, \forall j \in J, \ell = 1, \ldots, d, \tag{13}$$

$$\eta^-_{j\ell} \leq U_{j\ell}\, (1 - \xi_{j\ell}), \forall j \in J, \ell = 1, \ldots, d, \tag{14}$$

$$\theta_{j\ell} = \eta^+_{j\ell} + \eta^-_{j\ell}, \forall j \in J, \ell = 1, \ldots, d, \tag{15}$$

$$\theta_{j\ell} \leq 1, \forall j \in J, \ell = 1, \ldots, d, \tag{16}$$

$$\theta_{j\ell} \geq \mu_{j\ell}, \forall j \in J, \ell = 1, \ldots, d, \tag{17}$$

$$\sum_{\ell=1}^{d} \mu_{j\ell} = 1, \forall j \in J, \tag{18}$$

$$\eta^+_{j\ell}, \eta^-_{j\ell}, \theta_{j\ell} \in \mathbb{R}^d_+, \forall j \in J, \ell = 1, \ldots, d, \tag{19}$$

$$\mu_{j\ell}, \xi_{j\ell} \in \{0, 1\}, \forall j \in J, \ell = 1, \ldots, d, \tag{20}$$

where M_{ij} *and* $U_{j\ell}$ *are big enough constants.*

We have introduced in the above formulation some new variables to model the ℓ_∞*-distance in the denominator of the residual* (9)*. In particular, for each* $j \in J$*, the* d*-dimensional variable* $\boldsymbol{\theta}_j$ *models the vector* $(|\boldsymbol{\beta}_{j1}|, \ldots, |\boldsymbol{\beta}_{jd}|)$ *for which the maximum has to be taken;* $\eta^+_{j\ell}$ *represents* $\max\{\boldsymbol{\beta}_{j\ell}, 0\}$ *and* $\eta^-_{j\ell}$ *the amount* $\max\{-\boldsymbol{\beta}_{j\ell}, 0\}$*, for all* $\ell = 1, \ldots, d$*. Clearly, one has that* $\boldsymbol{\beta}_j = \boldsymbol{\eta}^+_j - \boldsymbol{\eta}^-_j$ *and* $\boldsymbol{\theta}_j = \boldsymbol{\eta}^+_j + \boldsymbol{\eta}^-_j$ *as imposed in constraints* (12)-(15)*, where the auxiliary variables* ξ *enforce that for each coordinate, either the positive or the negative part assumes value zero (avoiding other types of decompositions). Constraints* (16)*,* (17) *and* (18) *assure that* $\max_{j \in J} |\boldsymbol{\beta}_j| = 1$ *via the auxiliary binary variables* $\mu_{j\ell} \in \{0, 1\}$ *that take value* 1 *in exactly one position (the one where the maximum is achieved).*

Thus, the formulation assures that $\max_{l=1,\ldots d} |\boldsymbol{\beta}_{j\ell}| = 1$*, and then, the expression of the residual for the point* x_i *becomes* $\varepsilon_{x_i}(\boldsymbol{\beta}_j, \alpha_j) = |\alpha_j + \boldsymbol{\beta}^t_j x_i|$ *for all* $i \in I$ *and* $j \in J$*.*

In this case, also (MOMFHP) *becomes a Mixed Integer Linear Programming problem.*

4. Set Partitioning formulation

In this section we alternatively reformulate (MOMFHP$_0$) as a set partitioning problem (SPP) (see e.g., Balas and Padberg (1976)). Our SPP is based on the idea that once the p clusters of demand points are known, (MOMFHP$_0$) reduces to finding the optimal hyperplanes for each of those clusters in which all the residuals are aggregated by means of an ordered median function. In particular, let S be a cluster of observations $S \subseteq I$. To this cluster S, we associate a hyperplane $\mathcal{H}$ which induces the residuals of the observations in S. Let $R = (S, \mathcal{H})$ be the pair composed by cluster S and hyperplane $\mathcal{H}$, we denote by e^i_R the marginal contribution of observation i in the cluster, and let $e_R = (e^i_R)_{(i \in S)}$ the vector of residuals induced by individuals in S with respect to

Journal Pre-proofs

the hyperplane $\mathcal{H}$. Next, let $\bar{e}_R$ be the cost of cluster S, i.e., the overall aggregation of the residuals of the data in S ($\bar{e}_R = \sum_{i \in S} e_R^i$). Finally, for each pair $R = (S, \mathcal{H})$ we define the variable

$$y_R = \begin{cases} 1 & \text{if cluster } S \text{ is selected and its associate hyperplane is } \mathcal{H}, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{R}$ be the set containing all possible pairs $R = (S, \mathcal{H})$.

The set partitioning formulation for (MOMFHP$_0$) is:

$$\min \sum_{i \in I} \sum_{R=(S,\mathcal{H}) \in \mathcal{R}: i \in S} \lambda_i e_R^{(i)} y_R \tag{21}$$

$$\text{s.t.} \sum_{R=(S,\mathcal{H}) \in \mathcal{R}} y_R = p, \tag{22}$$

$$\sum_{R=(S,\mathcal{H}) \in \mathcal{R}: i \in S} y_R = 1, \forall\, i \in I, \tag{23}$$

$$y_R \in \{0,1\},\ \forall R \in \mathcal{R}. \tag{24}$$

where $e_R^{(i)}$ is the i-th element in the sorted sequence of (active) residuals. In the above formulation the objective function computes the ordered median aggregation of the residuals (each demand point i allocated to its cluster S). Constraint (22) assures that p clusters have to be computed and constraints (23) that each observation belongs to a single cluster.

In the same manner that we formulate the ordered median objective function in the compact formulation we can equivalently reformulate the problem above as follows:

$$\min \sum_{k \in I} u_k + \sum_{i \in I} v_i \tag{25}$$

$$\text{s.t.} \quad u_k + v_i \geq \lambda_k \sum_{R \in \mathcal{R}: i \in S} e_R^i y_R, \ \ \forall i, k \in I, \tag{26}$$

$$\sum_{R \in \mathcal{R}} y_R = p, \tag{27}$$

$$\sum_{R \in \mathcal{R}: i \in S} y_R = 1, \ \ \forall\, i \in I, \tag{28}$$

$$y_R \in \{0,1\}, \ \ \forall R \in \mathcal{R}, \tag{29}$$

$$u_k, v_i \in \mathbb{R}, \ \ \forall i, k \in I.$$

This problem will be referred to as the *Master Problem*.

The problem above, although adequately solves the problem of finding the p hyperplanes once the optimal clusters are computed, has a huge number of variables (and coefficients to incorporate to constraints (26)). Thus, we propose a column generation (CG) approach for solving, efficiently, the problem above by adding new variables to the model as needed and not considering all of them at the same time. A pseudocode indicating the general procedure is shown in Algorithm 1.

Initially, a (small) subset of the y-variables is considered (those indexed by the sets in $\mathcal{R}_0$) and a relaxed version of the problem is solved with only these variables. It implies to compute the

amounts e_R^i for all $R \in \mathcal{R}_0$ and $i \in S$. Next, it has to be checked whether the optimality condition is satisfied. If it is not the case, a new set of variables is found and added to the relaxed problem and the procedure is repeated.

Algorithm 1: General Scheme for the CG approach.

Data: $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, $p \in \mathbb{Z}_+$ $(p > 0)$, $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$.

1. **Preprocessing:** Compute a set of initial clusters with their associated hyperplanes for the problem $\mathcal{R}_0 = \{R_1, \ldots, R_K\}$ with $R_k = (S_k, \mathcal{H}_k)$, $S_k \subseteq I$ for all $k = 1, \ldots, K$.

2. **Relaxed Master:** Solve the relaxed master problem:

$$
\begin{aligned}
\min & \sum_{k \in I} u_k + \sum_{i \in I} v_i \\
\text{s.t. } & u_k + v_i \geq \lambda_k \sum_{R \in \mathcal{R}_0 : i \in S} e_R^i y_R, \forall i, k \in I, \\
& \sum_{R \in \mathcal{R}_0} y_R = p, \\
& \sum_{R \in \mathcal{R}_0 : i \in S} y_R = 1, \forall\, i \in I, \\
& 0 \leq y_R \leq 1, \ \forall R \in \mathcal{R}_0, \\
& u_k, v_i \in \mathbb{R}, \ \ \forall i, k \in I.
\end{aligned}
\tag{RMP}
$$

3. **New Columns** : Check if new columns have to be added to (RMP).

   if *Optimality is satisfied* **then**
   $\mathcal{C}^* = \{R \in \mathcal{R} : y_R^* = 1\}$.
   else
   Update $\mathcal{R}_0$ with the new columns and go to **2.**
   end

Result: $\{\mathcal{H}(\boldsymbol{\beta}_S, \alpha_S) : R = (S, \mathcal{H}(\boldsymbol{\beta}_S, \alpha_S)) \in \mathcal{C}^*\}$.

The crucial steps in the implementation of the CG approach are the following:

1. *Preprocessing:* Generation of initial feasible solutions induced by a set of initial clusters and their associated hyperplanes (and their costs). This step may be improved by the theoretical properties verified by the corresponding optimal hyperplanes. First, we have heuristically generated different types of initial variables (see Section 4.1). Second, we also have implemented different initial solutions based on properties of the optimal solution of median and center hyperplanes (see Section 4.2).
2. *Pricing:* As already mentioned, in set partitioning problems, instead of solving initially the problem with the whole set of variables, the variables have to be incorporated *on-the-fly* by solving adequate pricing subproblems derived from previously computed solutions until the optimality of the solution is guaranteed.

Journal Pre-proofs

3. *Branching:* The rule that creates new nodes of the branch and bound tree when a fractional solution is found at a node of the search tree. In this problem, we have adapted the Ryan-and-Foster branching complemented by a secondary ad-hoc branching in some special situations.

In what follows we describe how each of the items above is performed in our proposal.

4.1. Preprocessing

In the preprocessing phase, we generate different types of initial solutions, which implies the initialization of the CG algorithm with a given set of variables.

We consider different types of initial solutions derived from the construction of hyperplanes fitting the sets of points. First, to initialize the pool of columns, $\mathcal{R}_0$, we randomly generate hyperplanes passing through d original points. Among the various strategies compared, we have eventually implemented one that chooses all possible pairs of original points and performs completions with $d-2$ points randomly chosen among the remaining. These d points determine a variable and a unique hyperplane passing through all of them. This strategy augment, at most, $\frac{n(n-1)}{2}$ new variables into the pool each time it is repeated. In addition, we also augment to the pool all variables associated to singletons. Finally, apart from the above initial columns, we also charge an initial heuristic solution (in the y-variables) so as to have a good upper bound in our branch-and-price algorithm and assuring that the problem is feasible at the root node of the branch-and-price tree. Our algorithm chooses at random p mutually disjoint subsets of d points and finds the hyperplanes determined by those p sets of d points. Next, the remaining points are assigned to the closest hyperplanes among those p. Then, we choose at random one of these p-clusters and perform a 1-interchange heuristic among its points generating a new hyperplane that replaces, one at a time, the one currently considered in the configuration until the first iteration where no improvement is possible. The neighborhood for the points that determine each hyperplane is formed by the points in its cluster that do not belong to the hyperplane. In case the hyperplane spans all the points in its cluster the new point is chosen randomly among the points not spanned by the current hyperplanes of the remaining clusters. The incumbent set of hyperplanes and their corresponding allocations define a set of columns that gives rise to an initial solution that is loaded into the solver.

4.2. Median and center optimal hyperplanes

We have used the following properties to build the initial solutions of our CG approach since they determine optimal hyperplanes for specific objective functions, see e.g., Schöbel (2003).

Lemma 4.1. *The following properties are verified:*

1. Weak incidence property*: There exists an optimal median hyperplane passing through d affinely independent points.*
2. Pseudo-halving property*: Every optimal median hyperplane, $\mathcal{H}(\beta^*, \alpha^*)$ verifies*
$$\#\Big\{i \in I : x_i \in \mathcal{H}^-(\beta^*, \alpha^*)\Big\} \leq \tfrac{n}{2} \text{ and } \#\Big\{i \in I : x_i \in \mathcal{H}^+(\beta^*, \alpha^*)\Big\} \leq \tfrac{n}{2}.$$
3. Weak blockedness property*: There exists an optimal center hyperplane that is at maximum distance from $d+1$ of the points.*
4. Parallel facets property*: There exists an optimal center hyperplane that is parallel to a facet of the convex hull of the given points.*

For the more general ordered median objective function, we have proved the following result that characterizes the ordered median hyperplanes. In what follows, we derive a novel result for these hyperplanes that will be useful in the preprocessing phase of our CG approach.

Let us introduce the following notation:

- Let $\mathcal{B}$ be the subdivision of the space of coefficients of the hyperplanes (β, α), $\mathbb{R}^{d+1}$, induced by the following arrangement of hyperplanes:

$$B_{ij}^{ab} = \Big\{(\beta, \alpha) \in \mathbb{R}^{d+1} : a(\beta x_i + \alpha) = b(\beta x_j + \alpha)\Big\}, \forall i, j \in I, a, b \in \{-1, 1\}.$$

- Let $\mathcal{S}$ be the subdivision of the space of coefficients of the hyperplanes (β, α), $\mathbb{R}^{d+1}$, induced by the following arrangement of hyperplanes

$$S_i = \Big\{(\beta, \alpha) \in \mathbb{R}^{d+1} : \beta x_i + \alpha = 0\Big\}, \forall i \in I.$$

Lemma 4.2. *If $\mathcal{H}(\beta, \alpha)$ is an optimal ordered median hyperplane then (β, α) is an extreme point of a cell in the subdivision of $\mathbb{R}^{d+1}$ induced by the intersection $\mathcal{B} \cap \mathcal{S}$.*

Proof. For a given hyperplane $\mathcal{H}(\beta, \alpha)$, let us consider the objective function of the problem, namely $\sum_{i \in I} \lambda_i e_{(i)}$, where $e_i = D(\mathcal{H}(\beta, \alpha), x_i)$.

Observe that within each cell of the subdivision $\mathcal{B}$ the sorting of the residuals does not change since this subdivision is the one induced by the equations $|\beta x_i + \alpha| = |\beta x_j + \alpha|$. In addition, in each cell of the subdivision $\mathcal{S}$ the sign of $\beta x_i + \alpha$ is either positive or negative (but does not change) for each $i \in I$. Therefore, if $C \in \mathcal{B} \cap \mathcal{S}$ is a cell in the subdivision induced by $\mathcal{B} \cap \mathcal{S}$, there is permutation σ that fixes the sorting of the residuals and also a constant vector $(sign(\beta x_1 + \alpha), \ldots, sign(\beta x_n + \alpha)) \in \{-1, 1\}^n$ such that

$$\sum_{i \in I} \lambda_i e_{(i)} = \sum_{i \in I} \lambda_i \frac{sign(\beta x_i + \alpha)(\beta x_i + \alpha)}{\|\beta\|^*} = \frac{\sum_{i \in I} \lambda_i sign(\beta x_i + \alpha)(\beta x_i + \alpha)}{\|\beta\|^*}.$$

The above function is the ratio of a non-negative linear function and a convex function, then it is quasiconcave provided that $(\beta, \alpha) \in C$. Therefore, it attains its minima at the extreme points of this region. Hence, if $\mathcal{H}(\beta, \alpha)$ is an optimal ordered median hyperplane (β, α) must be an extreme point of some of those cells. □

The above result allows us to interpret optimal ordered median hyperplanes also in terms of a geometrical description as those that meet d conditions between the following cases: i) passing through points x_i, $i \in I$, and ii) being at the same distance of two points x_i, x_j, $i, j \in I$. Optimal ordered median hyperplanes must also satisfy, for some $k = 1, \ldots, d$, the following property: it contains k points x_i, $i \in I$ and it is at the same distance from $d - k$ pairs x_i, x_j $i, j \in I$.

In our computational results we have computed the initial solutions and the initial pool of variables for the objective functions of type median, k-centrum and centdian, using the weak incidence property, whereas for the center objective function we use the weak blockedness property.

Journal Pre-proofs

4.3. Pricing problem

Certifying optimality in a CG approach avoiding the inclusion of all the columns into the relaxed master problem, (RMP), requires testing whether a new tentative column must be added to the problem. In case no new candidates are added to the master problem, the optimality is guaranteed, otherwise, one should add the new columns and repeat the process (Step 3 in Algorithm 1). Searching for new columns to be added to the model will be performed by looking at the dual formulation of the set partitioning formulation.

Let γ be the dual variable for constraint (27), ϕ_i the dual variables for constraints (28) and δ_{ik} the dual variables for constraints (26). Then, the dual of the Master Problem is the following:

$$\begin{aligned}
&\max -p\gamma + \sum_{i\in I}^{n} \phi_i \\
&\text{s.t. } \sum_{k\in I} \delta_{ik} = 1, \forall i \in I, \\
&\quad \sum_{i\in I} \delta_{ik} = 1, \forall k \in I, \\
&\quad - \sum_{\substack{i\in S:\\ R=(S,\mathcal{H})}} \sum_{k\in I} \lambda_k e_R^i \delta_{ik} - \gamma + \sum_{\substack{i\in S:\\ R=(S,\mathcal{H})}} \phi_i \leq 0, \forall R \subseteq \mathcal{R}_0, \\
&\quad \delta_{ik}, \gamma, \phi_i \geq 0.
\end{aligned}$$

Hence, for any $R \subset \mathcal{R}_0$, since y_R does not appear in the objective function, the reduced cost for variable y_R is:

$$\bar{e}_R = \gamma - \sum_{\substack{i\in S:\\ R=(S,\mathcal{H})}} \phi_i + \sum_{\substack{i\in S:\\ R=(S,\mathcal{H})}} \sum_{k\in I} \lambda_k e_R^i \delta_{ik}.$$

Then, given an optimal dual solution $(\gamma^*, \phi^*, \delta^*)$, and considering the binary variables

$$w_i = \begin{cases} 1 & \text{if the } i\text{-th point is chosen for the cluster } S \text{ defining the pair } R \text{ indexing the} \\ & \text{new column,} \\ 0 & \text{otherwise,} \end{cases}$$

the pricing problem is to choose the subset S with minimum reduced cost, i.e., to solve:

$$\begin{aligned}
&\min -\sum_{i\in I} \phi_i^* w_i + \gamma^* + \sum_{i\in I} c_i^* r_i \\
&\text{s.t. } z_i \geq \varepsilon_{x_i}(\boldsymbol{\beta}, \alpha)), \ \ \forall i \in I, \\
&\quad r_i \geq z_i - M(1 - w_i), \ \ \forall i \in I, \\
&\quad w_i \in \{0, 1\}, \ \ \forall i \in I, \\
&\quad z_i, r_i \geq 0, \ \ \forall i \in I, \\
&\quad \boldsymbol{\beta} \in \mathbb{R}^d, \alpha \in \mathbb{R}.
\end{aligned}$$

where $c_i^* = \sum_{k=1}^{n} \lambda_k \delta_{ik}^*$, $\forall i \in I$.

If the optimal value of this problem is negative, the new column $y_{\hat{R}}$ is added to the pool, where $\hat{R} = (\hat{S}, \hat{\mathcal{H}}(\boldsymbol{\beta}, \alpha))$ and $\hat{S} = \{i : w_i = 1\}$, since its reduced cost in the (RMP) is negative, and thus, it improves the objective function of the master problem. Otherwise, optimality is certified and we are finished.

Journal Pre-proofs

4.3.1. Heuristic pricing

The exact pricing routine described above is an NP-hard problem and thus in general, it takes time finding new columns to be added to the pool or to certify optimality of the reduced master problem. This last task cannot be avoided, provided that we design an exact solution algorithm. Nevertheless, in many occasions finding promising new variables can be done at very low computational time resorting to heuristic schemes.

In our problem, we propose to test hyperplanes chosen from a discrete set of potential candidates. To do so, we set a d-dimensional grid on the normalized space of α and $\boldsymbol{\beta}$ coefficients. Each point represents a hyperplane to be tested. Once the candidate $(\alpha, \boldsymbol{\beta})$ is chosen we determine which set of points S is going to be added to the cluster encoded by the new variable y_R. This is done with a simple greedy rule: choose those points with negative reduced cost with respect to $\mathcal{H}(\beta, \alpha)$.

If after this process we find a hyperplane that produces a negative reduced cost, we add this new column to the pool. Otherwise, we proceed with the exact pricer. This scheme speeds up the search for new columns without loosing the exactness of the whole algorithm.

4.4. Branching

The set partitioning formulation of the MOMFHP$_0$ is often not solved at the root node, in contrast with what is stated in Park et al. (2017). Thus, some branching strategy must be implemented to cope with the branch and bound search. Ryan-Foster (R-F) is one of the most popular techniques for branching in set partitioning problems (see Ryan and Foster (1981)). If a fractional solution is reached at a node, R-F creates two new branches as follows: Given two elements $i_1, i_2 \in I$, they may never go together on a set in the whole branch, or they may always go together, i.e., if one of them belongs to a set S, the other one must also be included in S.

To implement this branching, we can take advantage of the w_i variables defined on the previous section for the pricing subproblem, to easily adapt this way of branching in our problem, by means of the following constraints:

A) $w_{i_1} + w_{i_2} = 1$ ensuring that elements i_1 and i_2 are not assigned to the same hyperplane.

B) $w_{i_1} = w_{i_2}$ ensuring that elements i_1 and i_2 are assigned to the same hyperplane.

Moreover, in our formulation there is a new case in which, despite the fact of having fractional solutions on a node, we will not create new branches following the R-F rule. This fact is motivated because in our problem may appear different columns (different y-variables) but being associated to the same set S, although possibly with different hyperplanes.

Let $S \subseteq I$ be a subset of points and let $y_{R^1}, ..., y_{R^q}$ be fractional variables for the same set S although with different hyperplanes $\mathcal{H}(\beta^i, \alpha^i)$, $i = 1 \dots q$, with $q > 1$, namely, $R^i = (S, \mathcal{H}(\beta^i, \alpha^i))$, such that $\sum_{i=1}^{q} y_{R^i} = 1$. If there are no more fractional variables, or the rest of the fractional variables of the node satisfy the same conditions for some other subsets of points, we cannot apply R-F rule and either the node need not be branched (see Theorem 4.3 and Remark 4.4) or a different branching strategy must be implemented in these cases.

Without loss of generality, we will describe the new branching for the case in which two fractional variables, y_{R^1} and y_{R^2}, with hyperplanes $\mathcal{H}(\beta^1, \alpha^1)$ and $\mathcal{H}(\beta^2, \alpha^2)$, are obtained in a node for the same subset S. In this situation, the new branching rule that we propose creates three new branches as follows:

Journal Pre-proofs

1. A branch where $y_{R^1} = 1$ meaning that this variable will be in the solution in this branch. This is easily implemented in the pricing routine since it amounts to avoid considering the elements in S in any further column in that branch because they are already in the set S which is part of the solution. Therefore, it suffices to fix the variables $w_i = 0,\ \forall i \in S$ in all the subproblems in the branch.

2. Analogously, it creates another branch where $y_{R^2} = 1$.

3. The third branch sets $y_{R^1} = y_{R^2} = 0$. This branch represents the case in which none of the original fractional solutions are part of the integer solution. Once again, this can be enforced by adding the following constraints to the pricing subproblems of the branch:

$$\left(\left(|S| - \sum_{i\in S} w_i\right) + \left||S| - \sum_{i\in I} w_i\right| + \sum_{\ell=1}^{d} |\beta_\ell^j - \beta_\ell^*| + |\alpha^j - \alpha^*|\right) \cdot M \geq 1, \quad j = 1, 2,$$

for a big enough constant M, where β^* and α^* define the new hyperplane $\mathcal{H}(\beta^*, \alpha^*)$. These constraints will make the problem infeasible if and only if all the individuals in S, and only the individuals of S, belong to the new solution, and moreover, the new solution provides a hyperplane $\mathcal{H}(\beta^*, \alpha^*)$ that is equal to $\mathcal{H}(\beta^1, \alpha^1)$ or $\mathcal{H}(\beta^2, \alpha^2)$.

The alternative branching may be necessary in case of using general norm based residuals. Nevertheless, as we show below, the situation is simpler using vertical distance residuals.

Theorem 4.3. *Ryan and Foster branching is enough in the set partitioning formulation of* (MOMFHP$_0$) *for the vertical distance residuals: If for a subset of points $S \subseteq I$, there exist fractional solutions $0 < y_{R^i} < 1$ with $R^i = (S, \mathcal{H}(\beta^i, \alpha^i))$ for i=1,2 at a node of the branch and bound tree then there exists an explicit solution that combines these variables to obtain a single one satisfying $y_{R^*} = 1$ with $R^* = (S, \mathcal{H}(\beta^*, \alpha^*))$.*

Proof. Let us consider a subset of points $S \subseteq I$. At a fractional node, we can have two possible scenarios: 1) $\#\{y_R \neq 0\} = 1$, hence, it would exist a single hyperplane (a facility) $\mathcal{H}(\beta, \alpha)$ that would serve the points of S, and hence, R-F branching is enough, and 2) $\#\{R : \text{such that } R = (S, \cdot) \text{ and } y_R \neq 0\} > 1$. This latter case needs a further analysis since it may seem as if more than one facility would need to be involved to optimally serve the points in S.

Without loss of generality we can assume $\#\{R = (S, \cdot) : y_R \neq 0\} = 2$ (a case with cardinality greater than 2 can be treated sequentially by smaller problems with two solutions). In this situation there are two variables, y_{R^1} and y_{R^2}, with values σ and $1 - \sigma$, $\sigma \in (0, 1)$, so that $y_{R^1} + y_{R^2} = 1$. These variables are represented by two hyperplanes $\mathcal{H}(\beta^1, \alpha^1)$ and $\mathcal{H}(\beta^2, \alpha^2)$, where the cost of a point $i \in S$ with coordinates $x \in \mathbb{R}^d$, e_i, is given by $e_i = \sigma D(x, \mathcal{H}(\beta^1, \alpha^1)) + (1-\sigma)D(x, \mathcal{H}(\beta^2, \alpha^2))$. We prove that the hyperplane $\mathcal{H}(\beta^*, \alpha^*)$ defined as

$$\mathcal{H}(\beta^*, \alpha^*) = \left\{z \in \mathbb{R}^d \ : \ \sigma(\alpha^1 + \beta^1 z) + (1 - \sigma)(\alpha^2 + \beta^2 z) = 0\right\},$$

satisfies that $D(x, \mathcal{H}(\beta^*, \alpha^*)) \leq \sigma D(x, \mathcal{H}(\beta^1, \alpha^1)) + (1 - \sigma)D(x, \mathcal{H}(\beta^2, \alpha^2))$ for vertical distance residuals, and this would mean that there exists a unique hyperplane that optimally serves all the points in S. Therefore, considering y_{R^*}, no further branching is required.

If we consider the normalized hyperplanes $\mathcal{H}(\beta^1, \alpha^1)$, and $\mathcal{H}(\beta^2, \alpha^2)$, such that $\beta_d^1 = \beta_d^2 = -1$, then $\beta_d^* = -1$, the vertical distance from x to $\mathcal{H}(\beta^*, \alpha^*)$ is

$$D_v(x, \mathcal{H}(\beta^*, \alpha^*)) = \left| x_d - \alpha^* - \sum_{\ell=1}^{d-1} \beta_\ell^* x_\ell \right|.$$

Hence,

$$\begin{aligned} D_v(x, \mathcal{H}(\beta^*, \alpha^*)) &= \left| x_d - \alpha^* - \sum_{\ell=1}^{d-1} \beta_\ell^* x_\ell \right| \\ &= \left| \sigma x_d + (1-\sigma) x_d - (\sigma\alpha^1 + (1-\sigma)\alpha^2) - \sum_{\ell=1}^{d-1} (\sigma\beta_\ell^1 + (1-\sigma)\beta_\ell^2) x_\ell \right| \\ &\leq \sigma \left| x_d - \alpha^1 - \sum_{\ell=1}^{d-1} \beta_\ell^1 x_\ell \right| + (1-\sigma) \left| x_d - \alpha^2 - \sum_{\ell=1}^{d-1} \beta_\ell^2 x_\ell \right| \\ &= \sigma D_v(x, \mathcal{H}(\beta^1, \alpha^1)) + (1-\sigma) D_v(x, \mathcal{H}(\beta^2, \alpha^2)). \end{aligned}$$

□

Remark 4.4. *We prove that under mild conditions, R-F branching is also enough for the ℓ_1-norm based residuals.*

Without loss of generality, assume that there is a solution with two fractional variables y_S^1 and y_S^2, with values σ and $(1-\sigma)$, and corresponding hyperplanes $\mathcal{H}(\beta^1, \alpha^1)$ and $\mathcal{H}(\beta^2, \alpha^2)$. Let us define the set $SP = \left\{ j : |\beta_j^1| = |\beta_j^2| = 1, j = 1, \ldots, d \right\}$. Hence, if $SP \neq \emptyset$, RF-branching is enough for the ℓ_1-norm residuals.

Indeed, let $\hat{\jmath}$ be an index such that $|\beta_{\hat{\jmath}}^1| = |\beta_{\hat{\jmath}}^2| = 1$.

It is clear that for any $\hat{\jmath} \in SP$, $\beta^*(\hat{\jmath}) = \sigma\beta^1 + (1-\sigma)\beta_{\hat{\jmath}}^1 \cdot \beta_{\hat{\jmath}}^2 \beta^2 \in \mathbb{R}^d$ satisfies $\|\beta_{\hat{\jmath}}^*\|_\infty = 1$. Consider for any $\hat{\jmath} \in SP$ the hyperplane

$$\mathcal{H}(\beta^*(\hat{\jmath}), \alpha^*) = \left\{ z \in \mathbb{R}^d \ : \ \sigma(\alpha^1 + \beta^1 z) + (1-\sigma)\beta_{\hat{\jmath}}^1 \cdot \beta_{\hat{\jmath}}^2 (\alpha^2 + \beta^2 z) = 0 \right\},$$

then for any individual $i \in S$ with coordinates $x_i \in \mathbb{R}^d$, taking into account that $||\beta^1||_\infty = ||\beta^2||_\infty = 1$, we obtain that

$$\begin{aligned} D_{\ell_1}(x_i, \mathcal{H}(\beta^*(\hat{\jmath}), \alpha^*)) &= \frac{|\alpha^* + \sum_{\ell=1}^d \beta_\ell^* x_{i\ell}|}{||\beta^*||_\infty} \\ &\leq \sigma \frac{|\alpha^1 + \sum_{\ell=1}^d \beta_\ell^1 x_{i\ell}|}{||\beta^1||_\infty} + (1-\sigma) \frac{|\alpha^2 + \sum_{\ell=1}^d \beta_\ell^2 x_{i\ell}|}{||\beta^2||_\infty} \\ &= \sigma D_{\ell_1}(x_i, \mathcal{H}(\beta^1, \alpha^1)) + (1-\sigma) D_{\ell_1}(x_i, \mathcal{H}(\beta^2, \alpha^2)). \end{aligned}$$

and hence, $y_{R^*} = 1$ for $R^* = (S, \mathcal{H}(\beta^*(\hat{\jmath}), \alpha^*))$ is an optimal solution for the problem.

5. Computational Results

A series of computational experiments has been performed in order to test the two proposed methodologies. We consider two different sets of instances, one based on Eilon et al. (1971) dataset and another on synthetic data. For all of them we solve (MOMFHP$_0$) for four different objective

Journal Pre-proofs

Journal Pre-proofs

functions: Weber (W), Center (C), $\lceil \frac{n}{2} \rceil$-Centrum (K) ($\lambda = (\overbrace{1, \ldots, 1}^{\lceil \frac{n}{2} \rceil}, 0, \ldots, 0)$) and 0.9-Centdian (D) ($\lambda = (1, 0.9, \ldots, 0.9)$) and with the two proposed approaches: the compact approach based on formulation (MOMFHP) and with the branch-and-price methodology. We test the performance of the algorithms on two different types of residuals: ℓ_1-norm based residuals and absolute value vertical distance residuals.

The models were coded in C and solved with SCIP v.6.0.1 (Gleixner et al., 2018) using as optimization solver CPLEX 12.8 in a Mac OS El Capitan with a Core i7 CPU clocked at 2.8 GHz and 16GB of RAM memory. A time limit of 5 hours was fixed for all the instances. It is well-known in the field of location analysis that continuous multifacility ordered median problems are very difficult to solve and already problems of moderate sizes ($n = 50$ demand points) can not often be solved to optimality (see e.g., Blanco et al. (2016)). The same or even a harder behavior should be expected here since these problems introduce a new degree of difficulty in the representation of general distance based residuals.

5.1. Eilon et al. (1971) dataset

First, we tested our approach on instances based on the classical planar 50-points dataset provided by Eilon et al. (1971). We randomly generate five instances from such a dataset with sizes $n \in \{20, 30, 40, 45\}$ and the entire complete original instance with $n = 50$. We run the models for $p \in \{2, 5\}$ hyperplanes. The average results obtained for these instances are shown in tables 1 and 2. There, for each combination of n (size of the instance), p (number of hyperplanes to be located) and `type` (ordered median objective function to be minimized), we provide both for the compact formulation MOMFHP (Compact) and for the branch-and-price (B&P) approach: the CPU time in seconds needed to solve the problem (`Time`) and within parentheses the number of unsolved instances (`(#Unsolved)`), the MIP Gap in percentage (`GAP(%)`) remaining after the time limit, the number of nodes (`Nodes`) explored in the branch and bound tree and the RAM memory (`Memory (MB)`) in Megabytes required during the execution process. Within each column (`Time`, `GAP`, `Nodes` and `Memory`), we highlight in bold the best result between the two formulations, namely Compact or B&P. Table 1 gives the results for the models with vertical distance residuals while Table 2 provides the results for the ℓ_1-norm residuals.

As expected, the difficulty of the problem increases with n and p. Problems with smaller n are easier and $p = 2$ is also easier than $p = 5$. We also observe in Table 1 that the B&P approach is more efficient than formulation MOMFHP for $p = 5$ but not for $p = 2$.

For that type of problem, $p = 2$, with vertical distance residuals the compact formulation is able to solve most of the instances and for those not solved the gap at termination is smaller than the corresponding for the B&P. Nevertheless, the behavior for $p = 5$ is the opposite and B&P solves more instances and reports smaller gap that the compact formulation. As it can be expected the number of nodes to be explored in order to solve the problems is several orders of magnitude larger for the compact formulation than for the B&P algorithm. This fact shows that the former formulation is weaker (also confirmed by the LP bound) than the latter thus requiring many more nodes to be explored to solve the problems, implying a better scalability of the B&P approach. In addition, MOMFHP requires very large RAM memory resources since already for $n = 50$ points, it demands, in some cases, more than 11 GB whereas B&P solves the problems using at most 4 GB of RAM memory.

Turning to Table 2 we observe, as expected, that using ℓ_1-norm residuals make problems harder to solve mainly due to the representation of the projections point-to-hyperplane stated in Remark

Journal Pre-proofs

3.4. In this case, the overall gaps increase from 29.36% and 11.24% in Table 1, for MOMFHP and B&P, respectively, to 62.69% and 20.53%. This behavior is more severe for MOMFHP because already for $n = 20$ and $p = 5$ that formulation is not able to certify optimality for any of the problems regardless of the type within the time limit. On the contrary, B&P is affected less and its behavior is similar to what one observes for vertical distance in Table 1. The rest of comments regarding number of nodes and memory requirements are similar to those given previously for vertical distances.

5.2. Synthetic Instances

We have also randomly generated another set of instances to evaluate the performance of the two solution approaches depending on the space dimension (d). We have generated five instances of random points in the unit hypercube for each meaningful combination of $n \in \{20, 30, 40, 45, 50\}$, $p \in \{2, 5, 10\}$ and $d \in \{2, 3, 8\}$ (note that for these datasets, we have included additionally $p = 10$ to analyze how increasing the number of hyperplanes affects the complexity for larger space dimension ($d = 8$)). At this point, it is important to point out that several combinations of the above factors result in trivial problems, for instance for $n = 20$ and $p = 10$ there is always a solution passing through all the points and thus with zero objective value. All those cases that give rise to trivial solutions are not reported. Table 3 reports the results for the models with vertical distance residuals while Table 4 provides the results for the ℓ_1-norm residuals. We report the same information as the one provided in the previous section but this time the results do not distinguish the type of objective function but the dimension of the space. (Needless to say that all the results disaggregated also by type are available upon request.)

For this dataset the results reinforce our previous observations in that for problems with vertical distances (see Table 3), MOMFHP is much weaker than B&P for $p = 5, 10$ and in any dimension. In this case, however as seen in Table 3 there are cases where for $p = 2$ MOMFHP (see column *Compact*) is more efficient. Turning to problems with ℓ_1-norm residuals the performance is more homogeneous and B&P is more efficient than MOMFHP for all n and d and for $p = 5, 10$. For $p = 2$, Compact solves more instances than B&P for $n = 20, 40$ and 45, less instances for $n = 20$ and the same number for $n = 50$. Nevertheless, for those instances that are not solved Compact reports larger gaps than B&P in all cases. Once again, one observes that problems with ℓ_1-norm residuals are more difficult than with vertical residuals. The overall gaps increase from 51.49% and 29.93% in Table 3, for MOMFHP and B&P, respectively, to 83.78% and 37.41% in Table 4.

6. Scalability: Bounding the error in aggregation procedures

This section is devoted to analyze the issue of scalability of our approach. We are aware that the methodology based on a branch and price algorithm may be computationally costly (we refer the reader to the Section 5 for further details). For that reason, we derive an approach that allows one to handle large data sets with appropriate error bounds.

Our approach is based on aggregating data to reduce the dimensionality of the original problem so that our branch and price approach can properly handle the problem. The important issue is that we can provide error bounds on these approximations that monotonically decrease with the quality of the aggregation. Obviously, aggregation strategies are not new since they have been already applied in some other areas although mostly from a heuristic point of view (see e.g., Current and Schilling (1987, 1990))

			Time (#Unsolved)				GAP(%)		Nodes		Memory (MB)	
n	type	p	Compact		B&P		Compact	B&P	Compact	B&P	Compact	B&P
20	W	2	**2.08**		68.15		0.00	0.00	2878	**2**	**3**	23
		5	8686.50	(2)	**23.48**		18.67	**0.00**	23623465	**16**	2226	**13**
	K	2	**2.10**		69.18		0.00	0.00	2878	**2**	**3**	23
		5	8703.61	(2)	**23.31**		18.70	**0.00**	23560539	**16**	2221	**13**
	D	2	**6.70**		86.14		0.00	0.00	2904	**2**	**3**	24
		5	9803.60	(2)	**43.96**		18.67	**0.00**	24131841	**26**	2161	**15**
	C	2	**0.11**		3171.27		0.00	0.00	**35**	5070	**1**	1347
		5	**103.70**		2798.43		0.00	0.00	204693	**12259**	**36**	300
30	W	2	**52.13**		1439.90		0.00	0.00	60401	**11**	**9**	109
		5	—	(5)	**654.68**		87.13	**0.00**	22547601	**109**	7194	**47**
	K	2	**52.77**		1440.09		0.00	0.00	60401	**11**	**9**	109
		5	—	(5)	**653.84**		87.11	**0.00**	22459376	**109**	7161	**47**
	D	2	**57.57**		2410.21		0.00	0.00	62889	**7**	**9**	107
		5	—	(5)	**242.66**		83.94	**0.00**	22252049	**25**	7240	**41**
	C	2	**0.17**		—	(5)	**0.00**	25.73	**40**	2833	**3**	4033
		5	**349.26**		1275.77	(3)	**0.00**	28.76	503141	**7537**	**89**	1318
40	W	2	**1870.93**		—	(5)	**0.00**	11.71	1453146	**1**	**91**	251
		5	—	(5)	**10694.92**	**(3)**	99.96	**1.88**	15197462	**280**	9801	**141**
	K	2	**1923.60**		—	(5)	**0.00**	11.73	1453146	**1**	**91**	248
		5	—	(5)	**10574.62**	**(3)**	99.96	**1.77**	15210786	**281**	9792	**142**
	D	2	**1765.95**		—	(5)	**0.00**	10.56	1290038	**1**	**87**	244
		5	—	(5)	**3346.72**		99.83	**0.00**	14810951	**864**	9700	**183**
	C	2	**0.26**		17046.73	(4)	**0.00**	22.50	**81**	408	**4**	1264
		5	**982.10**		8626.58	(2)	**0.00**	18.17	1196810	**1358**	**205**	1559
45	W	2	**10238.75**		—	(5)	**0.00**	27.97	6492828	**1**	**219**	351
		5	—	(5)	—	(5)	99.86	**29.67**	12121664	**32**	10427	**139**
	K	2	10438.84		**3858.26**	**(2)**	**0.00**	5.05	6532368	**9**	**208**	401
		5	—	(5)	—	(5)	100.00	**47.44**	12720196	**25**	11047	**142**
	D	2	**10192.16**		—	(5)	**0.00**	45.37	6066819	**1**	**217**	336
		5	—	(5)	**7525.10**		99.46	**0.00**	12193404	**2383**	9076	**336**
	C	2	**0.35**		706.28	(4)	**0.00**	7.58	**133**	2286	**5**	932
		5	**1268.05**		15430.62	(4)	**0.00**	36.94	1570195	**1490**	**244**	2245
50	W	2	—	(1)	—	(1)	22.46	**3.48**	9401360	**1**	1593	**582**
		5	—	(1)	—	(1)	100.00	**52.33**	10760962	**1**	11353	**127**
	K	2	—	(1)	—	(1)	22.57	**3.48**	9275884	**1**	1584	**583**
		5	—	(1)	—	(1)	100.00	**52.33**	10743398	**1**	11335	**126**
	D	2	—	(1)	—	(1)	21.06	**2.84**	8902849	**1**	1238	**515**
		5	—	(1)	—	(1)	100.00	**46.21**	9732814	**1**	9864	**134**
	C	2	**0.29**		—	(1)	**0.00**	19.79	**37**	372	**6**	923
		5	**1778.36**		—	(1)	**0.00**	43.86	2234747	**470**	**281**	1432
Total Average:			**2521.67**	**(57)**	2481.66	(73)	29.36	**11.24**	7737963	**1120**	2888	**517**

Table 1: Results for Eilon et al. (1971) instances for vertical distance.

Journal Pre-proofs

			Time (#Unsolved)				GAP(%)		Nodes		Memory (MB)	
n	type	p	Compact		B&P		Compact	B&P	Compact	B&P	Compact	B&P
20	W	2	166.73		**136.75**		0.00	0.00	163750	**21**	**17**	30
		5	—	(5)	**111.13**		100.00	**0.00**	29829455	**32**	10437	**14**
	K	2	167.49		**136.65**		0.00	0.00	163750	**21**	**17**	30
		5	—	(5)	**109.85**		100.00	**0.00**	30416596	**32**	10655	**14**
	D	2	624.85		**103.66**		0.00	0.00	690721	**26**	40	**30**
		5	—	(5)	**56.87**		100.00	**0.00**	31528234	**15**	10624	**13**
	C	2	**0.98**		4877.58	(2)	**0.00**	5.13	**1398**	3597	**3**	2141
		5	—	(5)	**11288.38**	**(3)**	100.00	**12.23**	40775405	**18269**	7513	**2118**
30	W	2	9743.92	(2)	**4509.86**		28.75	**0.00**	13086135	**26**	1187	**123**
		5	—	(5)	**947.24**		98.89	**0.00**	20504433	**35**	10616	**39**
	K	2	9724.20	(2)	**4507.75**		28.74	**0.00**	13098248	**26**	1188	**123**
		5	—	(5)	**927.14**		98.88	**0.00**	21254042	**35**	11006	**39**
	D	2	2897.96	(3)	**4595.89**		26.93	**0.00**	13057250	**27**	1724	**127**
		5	—	(5)	**1885.54**		100.00	**0.00**	20197549	**140**	10972	**49**
	C	2	**2.92**		3154.04	(3)	**0.00**	19.61	1192	**507**	**4**	2154
		5	**2055.73**	**(4)**	—	(5)	80.00	**46.40**	25951648	**3028**	8158	**3031**
40	W	2	—	(5)	—	(5)	42.82	**6.40**	13047861	**1**	2218	**201**
		5	—	(5)	—	(5)	100.00	**39.89**	14778541	**49**	10698	**98**
	K	2	—	(5)	—	(5)	42.95	**6.40**	12998370	**1**	2214	**201**
		5	—	(5)	—	(5)	100.00	**39.20**	15280145	**59**	11070	**101**
	D	2	—	(5)	—	(5)	65.74	**7.03**	10642421	**1**	1809	**213**
		5	—	(5)	—	(5)	100.00	**35.47**	14043320	**111**	9495	**145**
	C	2	**2.64**		13923.59	(4)	**0.00**	39.56	3792	**124**	**5**	1593
		5	—	(5)	—	(5)	100.00	**60.84**	23716675	**299**	11125	**1002**
45	W	2	—	(5)	—	(5)	42.39	**4.85**	10512338	**1**	2795	**287**
		5	—	(5)	—	(5)	100.00	**49.90**	11757058	**2**	10826	**97**
	K	2	—	(5)	—	(5)	49.79	**25.56**	10903011	**1**	2767	**299**
		5	—	(5)	—	(5)	100.00	**55.13**	12704430	**1**	11911	**98**
	D	2	—	(5)	—	(5)	62.64	**24.59**	8683785	**1**	2263	**296**
		5	—	(5)	—	(5)	100.00	**48.34**	11591052	**3**	9553	**95**
	C	2	**2.13**		—	(5)	**0.00**	41.79	2251	**37**	**6**	1036
		5	—	(5)	—	(5)	100.00	**61.79**	22769758	**133**	11064	**728**
50	W	2	—	(1)	—	(1)	96.53	**8.22**	7215656	**1**	3368	**371**
		5	—	(1)	—	(1)	100.00	**53.68**	11063050	**1**	9171	**109**
	K	2	—	(1)	—	(1)	96.51	**8.22**	7288856	**1**	3402	**371**
		5	—	(1)	—	(1)	100.00	**53.68**	11115826	**1**	9212	**109**
	D	2	—	(1)	—	(1)	96.34	**8.21**	6792290	**1**	2722	**338**
		5	—	(1)	—	(1)	100.00	**55.80**	11147925	**1**	9840	**118**
	C	2	**4.21**		—	(1)	**0.00**	47.22	5241	**15**	**8**	680
		5	—	(1)	—	(1)	100.00	**63.66**	19632691	**52**	9468	**488**
Total Average:			1580.36	(123)	**2172.46**	**(100)**	62.69	**20.53**	13958539	**794**	5756	**508**

Table 2: Results for Eilon et al. (1971) instances for ℓ_1-distance.

Journal Pre-proofs

			Time (#Unsolved)				GAP(%)		Nodes		Memory (MB)	
n	d	p	Compact		B&P		Compact	B&P	Compact	B&P	Compact	B&P
20	2	2	**11.00**		56.48		0.00	0.00	2710	**139**	**2**	49
		5	3765.45	(13)	**413.44**	**(4)**	46.32	**1.25**	13909807	**40836**	2209	**430**
	3	2	**4.39**		233.22		0.00	0.00	2922	**417**	**3**	137
		5	—	(20)	**21.86**	**(5)**	100.00	**19.56**	19586370	**8330**	629	**29**
	8	2	**30.88**		1506.93		0.00	0.00	29777	**1530**	**3**	782
30	2	2	**55.66**		2083.69	(1)	**0.00**	1.34	44038	**1039**	**7**	1218
		5	1735.01	(15)	**729.11**	**(5)**	74.14	**11.01**	11575613	**7989**	4058	**1274**
		10	—	(20)	**30.71**	**(5)**	100.00	**11.05**	15536270	**10046**	1572	**298**
	3	2	**60.69**		6473.90	(4)	**0.00**	5.77	48533	**1741**	**8**	2737
		5	—	(20)	**7749.20**	**(9)**	100.00	**18.32**	15098323	**5033**	3187	**384**
	8	2	**414.28**		—	(20)	**0.00**	67.19	270055	**779**	**23**	1516
40	2	2	**1490.88**		6080.82	(19)	**0.00**	14.85	903463	**805**	**56**	2186
		5	7047.41	(16)	**6557.14**	**(11)**	75.78	**18.52**	10566235	**3642**	5303	**1187**
		10	—	(20)	**1331.86**	**(5)**	100.00	**19.66**	12196952	**1770**	2179	**159**
	3	2	**1164.17**		—	(20)	**0.00**	18.04	726140	**466**	**40**	1579
		5	—	(20)	—	(20)	100.00	**57.67**	12378840	**705**	5061	**412**
		10	—	(20)	**884.80**	**(15)**	100.00	**43.81**	12359358	**584**	1121	**79**
	8	2	**8455.38**		—	(20)	**0.00**	71.44	4005359	**59**	**140**	417
45	2	2	**8065.22**	**(6)**	—	(20)	**4.56**	25.70	5700797	**587**	**316**	2353
		5	11523.46	(16)	**10873.49**	**(15)**	75.04	**33.86**	9858979	**2061**	6104	**945**
		10	—	(20)	**2683.05**	**(5)**	100.00	**20.37**	11236807	**889**	2124	**111**
	3	2	**8390.99**		—	(20)	**0.00**	25.64	4494533	**235**	**127**	1205
		5	—	(20)	—	(20)	100.00	**63.19**	11396830	**353**	5914	**344**
		10	—	(20)	**2403.31**	**(9)**	100.00	**28.47**	10911247	**272**	1471	**71**
	8	2	**283.87**	**(15)**	—	(20)	**33.32**	67.26	5409179	**11**	962	**324**
		5	—	(20)	—	(20)	100.00	100.00	11094838	**1**	329	**64**
50	2	2	**0.36**	**(3)**	—	(4)	18.70	**6.95**	6135466	**393**	**1132**	2361
		5	**12373.18**	**(3)**	—	(4)	75.02	**51.32**	8984044	**801**	7563	**729**
		10	—	(4)	**6654.32**	**(1)**	100.00	**20.01**	10892069	**363**	2483	**83**
	3	2	**2.05**	**(3)**	—	(4)	**20.60**	30.71	6182966	**112**	**989**	1209
		5	—	(4)	—	(4)	100.00	**65.74**	10380563	**157**	5969	**268**
		10	—	(4)	—	(4)	100.00	**68.93**	10139295	**214**	1537	**75**
	8	2	**272.73**	**(3)**	—	(4)	**45.65**	67.16	4649063	**2**	1243	**407**
		5	—	(4)	—	(4)	100.00	100.00	10407264	**1**	1472	**83**
Total Average:			**2625.00**	**(309)**	2251.36	(321)	51.49	**29.93**	7735135	**3287**	1718	**773**

Table 3: Results for synthetic instances for vertical distance.

Journal Pre-proofs

			Time (#Unsolved)				GAP(%)		Nodes		Memory (MB)	
n	d	p	Compact		B&P		Compact	B&P	Compact	B&P	Compact	B&P
20	2	2	**2799.81**		540.45	(1)	**0.00**	1.50	4193628	**545**	**216**	424
		5	17545.62	(19)	**1160.12**	**(3)**	100.00	**4.15**	35717983	**5312**	9878	**922**
	3	2	6691.57	(7)	**4226.37**		8.41	**0.00**	17136729	**617**	696	**693**
		5	—	(20)	**147.31**	**(5)**	100.00	**16.99**	39208660	**3325**	5832	**52**
	8	2	—	(20)	**10356.65**	**(12)**	100.00	**36.68**	32145208	**324**	488	**476**
30	2	2	2491.37	(11)	**8898.27**	**(3)**	43.06	**4.35**	11908005	**293**	2283	**1230**
		5	—	(20)	**5575.20**	**(7)**	100.00	**14.72**	24028581	**1361**	11038	**1765**
		10	17960.21	(19)	**327.34**	**(5)**	100.00	**14.56**	24603724	**1162**	8567	**169**
	3	2	14.09	(15)	**9603.14**	**(13)**	50.87	**18.57**	13408414	**178**	2644	**894**
		5	—	(20)	**6628.48**	**(11)**	100.00	**22.45**	23654639	**631**	9211	**785**
	8	2	—	(20)	—	(20)	100.00	**85.77**	18682135	**14**	3855	**171**
40	2	2	**2.09**	**(15)**	13973.80	(19)	63.24	**14.27**	8495764	**76**	2259	**1074**
		5	—	(20)	—	(20)	100.00	**42.62**	17696588	**193**	10524	**904**
		10	11767.31	(19)	**7695.73**	**(6)**	100.00	**22.17**	17749077	**212**	8306	**45**
	3	2	**38.58**	**(15)**	—	(20)	64.33	**31.75**	7710536	**25**	2929	**601**
		5	—	(20)	—	(20)	100.00	**69.83**	19758253	**76**	9542	**378**
		10	—	(20)	—	(20)	100.00	**63.85**	16865456	**111**	5522	**44**
	8	2	—	(20)	—	(20)	100.00	**73.86**	12839616	**2**	4272	**192**
45	2	2	**2.61**	**(15)**	16547.12	(19)	61.73	**16.31**	7295436	**26**	2189	**1101**
		5	—	(20)	**9227.47**	**(17)**	100.00	**45.50**	13916126	**714**	9640	**729**
		10	—	(20)	**9606.52**	**(9)**	100.00	**24.11**	14582103	**234**	8573	**61**
	3	2	**28.92**	**(15)**	—	(20)	74.40	**25.74**	5456543	**10**	2359	**602**
		5	—	(20)	—	(20)	100.00	**68.74**	15523670	**31**	9216	**294**
		10	—	(20)	**5573.00**	**(18)**	100.00	**60.65**	14067756	**54**	6130	**59**
	8	2	—	(20)	—	(20)	100.00	**69.62**	10392241	**1**	4242	**243**
		5	—	(20)	—	(20)	100.00	100.00	13168789	**1**	1284	**57**
50	2	2	**1.59**	**(3)**	7203.33	(3)	71.34	**10.99**	6941692	**11**	3250	**950**
		5	—	(4)	**8172.47**	**(3)**	100.00	**46.58**	15611084	**492**	10326	**704**
		10	—	(4)	**7236.13**	**(3)**	100.00	**29.85**	14012908	**273**	6934	**72**
	3	2	**48.23**	**(3)**	—	(4)	60.70	**13.84**	5691897	**5**	1877	**1068**
		5	—	(4)	—	(4)	100.00	**85.37**	14771026	**9**	8255	**199**
		10	—	(4)	—	(4)	100.00	**67.47**	12123118	**26**	5274	**69**
	8	2	—	(4)	—	(4)	100.00	**64.78**	9864048	**1**	3915	**345**
		5	—	(4)	—	(4)	100.00	100.00	10436698	**1**	3278	**69**
Total Average:			2961.44	(480)	**4924.89**	**(377)**	83.78	**37.41**	16590341	**569**	5435	**531**

Table 4: Results for synthetic instances for ℓ_1 distance.

Journal Pre-proofs

Journal Pre-proofs

Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ be a set of demands points. Aggregating X into a new set of demand points X' consists of replacing X by a smaller (multi)set $X' = \{x'_1, \ldots, x'_n\}$ and to assign each point x_i in X to a point x'_i in X' (since the cardinality of the different elements of X' is smaller than the cardinality of X, several x_i may be assigned to the same x'_i and thus actually, some of the elements in X' coincide). A possible choice can be substituting the set of original demand points by the centroids obtained by any of the available clustering techniques. In any case, when solving (MOMFHP$_0$) for X' instead of using X one incurs in aggregation errors.

Let $\mathbb{H}$ be the optimal arrangement of p hyperplanes for the problem and $\mathbf{e} = (e_1, \ldots, e_n)$ with $e_i = \varepsilon_{x_i}\Big(\mathbb{H}\Big)$, for $i \in I$, the residuals with respect to $\mathbb{H}$. Analogously, let $\mathbb{H}'$ be the optimal arrangement for the demand points in X' and $\mathbf{e}'$ the vector of residuals.

Theorem 6.1. *Let* $T = \max_{i=1,\ldots,n} \mathrm{D}(x_i, x'_i)$. *Then, the following relation holds:*

$$|\mathrm{OM}_\lambda(\mathbf{e}') - \mathrm{OM}_\lambda(\mathbf{e})| \leq 2\mathrm{OM}_\lambda(T, \ldots, T). \tag{30}$$

Proof. First of all, observe that, based on the triangular inequality, for any $\mathbb{H}$

$$\varepsilon_{x_i}(\mathbb{H}) \leq \varepsilon_{x'_i}(\mathbb{H}) + \mathrm{D}(x_i, x'_i), \forall i = 1, \ldots, n.$$

Let us also consider the vector $\mathbf{t} = (\mathrm{D}(x_1, x'_1), \ldots, \mathrm{D}(x_n, x'_n))$ of distances from the original points in X to their corresponding points in X' and denote by $\tilde{\mathbf{e}} = (\varepsilon_{x'_1}(\mathbb{H}), \ldots, \varepsilon_{x'_n}(\mathbb{H}))$. Since the function OM is non-decreasing monotone and sublinear, it follows that:

$$\mathsf{OM}_\lambda(\mathbf{e}) \leq \mathsf{OM}_\lambda(\tilde{\mathbf{e}} + \mathbf{t}) \leq \mathsf{OM}_\lambda(\tilde{\mathbf{e}}) + \mathsf{OM}_\lambda(\mathbf{t}).$$

Hence, since $T \geq \mathrm{D}(x_i, x'_i)$ for all $i = 1, \ldots, n$, we get that:

$$|\mathsf{OM}_\lambda(\mathbf{e}) - \mathsf{OM}_\lambda(\tilde{\mathbf{e}})| \leq \mathsf{OM}_\lambda(T, \ldots, T).$$

From the above inequality we can apply (Geoffrion, 1977, Theorem 5) to conclude that

$$|\mathsf{OM}_\lambda(\mathbf{e}') - \mathsf{OM}_\lambda(\mathbf{e})| \leq 2\mathsf{OM}_\lambda(T, \ldots, T).$$

□

The difference considered in the above theorem is the excess due to the implementation of an approximate solution based on the reduced model with data set X' rather than the correct optimal solution for the original data in the larger set X. This result allows us to scale our CG algorithm to problems of any size using aggregation techniques and providing estimates on the deviation from the optimal value.

We illustrate the application of the above result including the percent error obtained aggregating to 20 points some of our random problems with 50 points by the 20-mean clustering technique. As one can see in Table 5 the percent errors are small. Observe that in some cases they are even negative, for problems that were not solved to optimality, and where the hyperplanes obtained by aggregating points, once evaluated on the actual 50 points, produce a smaller error than the upper bound found by the algorithm on the original dataset.

		error (%)	
p	type	$d = 2$	$d = 3$
2	W	3.48	2.78
	K	-17.84	-16.52
	C	4.40	5.90
	D	3.59	2.87
5	W	-0.16	1.21
	K	-9.17	-2.99
	C	6.60	20.86
	D	0.16	-11.16

Table 5: % aggregation errors for 50 points problems and vertical distance.

7. Conclusions

This paper considers the problem of locating a given number of hyperplanes in order to minimize an objective function of the distances from a set of points. Each point is assigned to its closest hyperplane, thus inducing as many clusters as the number of fitting hyperplanes. The distance from each point to its corresponding fitting hyperplane can be seen as a residual and these residuals are aggregated using ordered median functions that are ordered weighted averages representing different types of utilities. Two exact approaches are presented to solve the problem. The first one is based on a compact mixed integer formulation whereas the second one is an extended set partitioning formulation that is handled by a branch-and-price approach. To enhance the performance of this last method we have developed a generator of initial feasible solutions based on geometrical properties of the optimal solutions of the hyperplane location problem that we have also derived in this paper, and that are used to initialize the column generation routine of this branch-and-price. We have also presented a heuristic pricing strategy that is used in combination with the exact one to speed up some pricing iterations. We report the comparison of both method to solve the problem in two different datasets on an extensive battery of computational experiments. The issue of scalability of the exact methods is also analyzed obtaining theoretical upper bounds of the error induced by some aggregated versions of the original dataset.

A possible extension to be developed in a follow up paper is the development of alternative heuristic algorithms capable to solve the problem for large instances. In view of the applications of the proposed methodology in machine learning, other types of tools could be also explored under the multisource ordered median paradigm, as for instance Support Vector Machines, where a first attempt have been already proposed by Blanco et al. (2020a).

Acknowledgements

This research has been partially supported by Spanish Ministry of Education and Science/FEDER grant number MTM2016-74983-C02-(01-02), and projects FEDER-US-1256951, CEI-3-FQM331 and *NetmeetData*: Ayudas Fundación BBVA a equipos de investigación científica 2019.

Journal Pre-proofs

Journal Pre-proofs

References

Balas, E. and Padberg, M. W. (1976). Set partitioning: A survey. *SIAM review*, 18(4):710–760.

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

Blanco, V., Japón, A., and Puerto, J. (2020a). Optimal arrangements of hyperplanes for svm-based multiclass classification. *Advances in Data Analysis and Classification*, 14:175–199.

Blanco, V., Puerto, J., and El-Haj Ben-Ali, S. (2014). Revisiting several problems and algorithms in continuous location with ℓ_τ-norms. *Computational Optimization and Applications*, 58(3):563–595.

Blanco, V., Puerto, J., and El-Haj Ben-Ali, S. (2016). Continuous multifacility ordered median location problems. *European Journal of Operational Research*, 250(1):56–64.

Blanco, V., Puerto, J., and Rodríguez-Chía, A. M. (2020b). On ℓ_p-support vector machines and multidimensional kernels. *Journal of Machine Learning Research*.

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

Bradley, P. S. and Mangasarian, O. L. (2000). K-plane clustering. *Journal of Global Optimization*, 16(1):23–32.

Brimberg, J., Juel, H., and Schöbel, A. (2002). Linear facility location in three dimensions—models and solution methods. *Operations Research*, 50(6):1050–1057.

Brimberg, J., Juel, H., and Schöbel, A. (2003). Properties of three-dimensional median line location models. *Annals of Operations Research*, 122(1-4):71–85.

Carbonneau, R. A., Caporossi, G., and Hansen, P. (2014). Globally optimal clusterwise regression by column generation enhanced with heuristics, sequencing and ending subset optimization. *J. Classif.*, 31(2):219–241.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Current, J. R. and Schilling, D. A. (1987). Elimination of source a and b errors in p-median location problems. *Geographical Analysis*, 19(2):95–110.

Current, J. R. and Schilling, D. A. (1990). Analysis of errors due to demand data aggregation in the set covering and maximal covering location problems. *Geographical Analysis*, 22(2):116–126.

Eilon, S., Watson-Gandy, C. D. T., and Christofides, N. (1971). *Distribution management : mathematical modelling and practical analysis*. London : Griffin.

Espejo, I. and Rodríguez-Chía, A. M. (2011). Simultaneous location of a service facility and a rapid transit line. *Computers & operations research*, 38(2):525–538.

Gauss, C. F. (1809). *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*, volume 7. Perthes et Besser.

Journal Pre-proofs

Geoffrion, A. (1977). Objective function approximations in mathematical programming. *Mathematical Programming*, 13:23–39.

Gitman, I., Chen, J., Lei, E., and Dubrawski, A. (2018). Novel prediction techniques based on clusterwise linear regression. *arXiv preprint arXiv:1804.10742*.

Gleixner, A., Bastubbe, M., Eifler, L., Gally, T., Gamrath, G., Gottwald, R. L., Hendel, G., Hojny, C., Koch, T., Lübbecke, M. E., Maher, S. J., Miltenberger, M., Müller, B., Pfetsch, M. E., Puchert, C., Rehfeldt, D., Schlösser, F., Schubert, C., Serrano, F., Shinano, Y., Viernickel, J. M., Walter, M., Wegscheider, F., Witt, J. T., and Witzig, J. (2018). The SCIP Optimization Suite 6.0. Technical report, Optimization Online.

Hennig, C. (1999). Models and methods for clusterwise linear regression. In *Classification in the Information Age*, pages 179–187. Springer.

Mangasarian, O. L. (1999). Arbitrary-norm separating plane. *Operations Research Letters*, 24(1-2):15–23.

Martini, H. and Schöbel, A. (1998). Median hyperplanes in normed spaces—a survey. *Discrete Applied Mathematics*, 89(1-3):181–195.

Martini, H. and Schöbel, A. (2001). Median and center hyperplanes in minkowski spaces—a unified approach. *Discrete Mathematics*, 241(1-3):407–426.

McGee, V. E. and Carleton, W. T. (1970). Piecewise regression. *Journal of the American Statistical Association*, 65(331):1109–1124.

Ogryczak, W. and Tamir, A. (2003). Minimizing the sum of the k largest functions in linear time. *Information Processing Letters*, 85(3):117–122.

Park, Y. W., Jiang, Y., Klabjan, D., and Williams, L. (2017). Algorithms for generalized clusterwise linear regression. *INFORMS Journal on Computing*, 29(2):301–317.

Plastria, F. and Carrizosa, E. (2001). Gauge distances and median hyperplanes. *Journal of Optimization Theory and Applications*, 110(1):173–182.

Quandt, R. E. (1958). The estimation of the parameters of a linear regression system obeying two separate regimes. *Journal of the american statistical association*, 53(284):873–880.

Ryan, D. M. and Foster, A. (1981). An integer programming approach to scheduling. In Wren, A., editor, *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, pages 269–280. North-Holland, Amsterdan.

Schöbel, A. (1999). *Locating lines and hyperplanes: theory and algorithms*, volume 25. Springer Science & Business Media.

Schöbel, A. (2003). Anchored hyperplane location problems. *Discrete and Computational Geometry*, 29(2):229–238.

Schöbel, A. (2015). Location of dimensional facilities in a continuous space. In *Location Science*, pages 135–175. Springer.

Journal Pre-proofs

Späth, H. (1982). A fast algorithm for clusterwise linear regression. *Computing*, 29(2):175–181.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Weber, A. (1909). *Ueber den standort der industrien*, volume 1. Verlag J.C.B.Mohr, Tübingen.

Journal Pre-proofs