Computers & Operations Research 78 (2017) 230–242

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A comparative study of formulations and solution methods for the discrete ordered p-median problem

Martine Labbé [a], Diego Ponce [b,c], Justo Puerto [b,c,*]

[a] Computer Science Department, Université Libre de Bruxelles, Belgium
[b] IMUS, Instituto de Matemáticas, Universidad de Sevilla, Spain
[c] Departamento de Estadística e Investigación Operativa, Universidad de Sevilla, Spain

ARTICLE INFO

Article history:
Received 28 July 2015
Received in revised form
30 May 2016
Accepted 6 June 2016
Available online 8 June 2016

Keywords:
Discrete multifacility location
MIP formulations
Ordered median problem

ABSTRACT

This paper presents several new formulations for the Discrete Ordered Median Problem (DOMP) based on its similarity with some scheduling problems. Some of the new formulations present a considerably smaller number of constraints to define the problem with respect to some previously known formulations. Furthermore, the lower bounds provided by their linear relaxations improve the ones obtained with previous formulations in the literature even when strengthening is not applied. We also present a polyhedral study of the assignment polytope of our tightest formulation showing its proximity to the convex hull of the integer solutions of the problem. Several resolution approaches, among which we mention a branch and cut algorithm, are compared. Extensive computational results on two families of instances, namely randomly generated and from Beasley's OR-library, show the power of our methods for solving DOMP.

1. Introduction

Recognizing the need for more flexible logistic models, Nickel [13] proposed the Discrete Ordered Median Location Problem (DOMP) which could be used to model different locations problems, as the p-median or the p-center. It is a flexible formulation based on applying an ordered weighted averaging operator to the costs as they appear in the solution and taking them into account with a suitable n-vector λ.

Given a vector of weights, the ordered weighted average of n real numbers is obtained by first ranking those numbers by non-decreasing order and then computing the scalar product of the ranked allocation cost vector and the weight vector, see e.g. Nickel and Puerto [14].

Consider a set of clients and a set of candidate locations where some facility can be established. Further we are given the costs for allocating clients to facilities. DOMP consists in choosing p facility locations and assigning each client to a facility with smallest allocation cost in order to minimize a special objective function, the so-called ordered median function. Given a vector of weights, this function consists in an ordered weighted average of the allocation costs, namely it sorts these costs in non-decreasing sequence and then it performs the scalar product of this so obtained sorted cost vector and the given vector of weights. This objective function has been widely applied in the field of location analysis and distribution models [10,9,17,18]. In addition, it has the potential to yield new models for order statistics embedded within mathematical programming formulations; thus enlarging the applications of optimization tools to the resolution of statistical problems in data analysis.

DOMP is known to be $\mathcal{NP}$ – complete, see Nickel and Puerto [14].

The first formulation of DOMP, proposed by Nickel [13], consists in an integer nonlinear problem. Then, Boland et al. [3] propose several linearizations of Nickel's model.

Instances with up to 30 clients could be solved to optimality by Boland et al. [3]. Further, if clients and facility locations coincide and if the allocation cost of a client to itself is equal to zero (the so-called free self-service), then instances with up to 100 clients could be solved by Marín et al. [10,11]. We observe that in all previously considered formulations the gaps with respect to the linear programming relaxations of those models are rather large, as mentioned in all those papers.

In this paper, we propose new formulations for DOMP and we develop a theoretical comparison of the lower bounds obtained from their LP-relaxations and show that our new formulations are rather tight. Our theoretical results also attempt to shed some light on the polyhedral structure of the new formulation based on scheduling constraints. We conclude with extensive computational

* Corresponding author at: IMUS, Instituto de Matemáticas, Universidad de Sevilla, Spain.
E-mail addresses: mlabbe@ulb.ac.be (M. Labbé), dponce@us.es (D. Ponce), puerto@us.es (J. Puerto).

http://dx.doi.org/10.1016/j.cor.2016.06.004
0305-0548/

experiments to compare the respective efficiency of these formulations.

For the ease of presentation, we assume in some cases in this paper that the client and facility location sets coincide. However, it is important to remark that all the models and results presented extend to the general case in which client and facility location may differ since we do not impose that the cost of allocating a client to itself is equal to zero.

The remaining paper is organized as follows: in Section 2 we define the problem and some previous formulations. Next, we present new formulations for the DOMP. In addition, we analyze the relationship between the polytopes of the previously known formulations of this problem and we identify facets for the related assignment polytope in Section 3. Finally, some computational experiments are reported in Section 4.

2. The problem and some formulations

Let I be a set of n points which at the same time represent clients and potential facility locations.

The cost for serving client i's demand from facility j is denoted by C_{ij} and a facility can serve as many clients as needed, i.e. facilities are uncapacitated.

The Discrete Ordered Median Problem (DOMP) consists in

(i) determining a subset J of p facility locations, $J \subset I$, to open and
(ii) assigning clients to closest open facilities in order to minimize the ordered median objective function defined as follows.

Given the set J of p open facilities, let $c_i(J)$ represents the cost for allocating client i to some facility in J such that $c_i(J) = \min_{j \in J} C_{ij}$.

Now let us rank the costs $c_i(J)$, $i \in I$ by non-decreasing order of their values. These ordered costs are denoted by $c^k_{\le}(J)$ and verify

$c^1_{\le}(J) \le \cdots \le c^n_{\le}(J)$.

Then, given a vector $\lambda = (\lambda^k)^n_{k=1}$ satisfying $\lambda^k \ge 0$, $k = 1, \ldots, n$, the DOMP objective function, also called ordered median function, is defined as

$$\sum_{k=1}^{n} \lambda^k c^k_{\le}(J). \tag{1}$$

Note that this objective function provides a very general paradigm to encompass standard and new location models. For instance, if $\lambda^1 = \cdots = \lambda^n = 1$ we obtain the median objective, if $\lambda^1 = \lambda^2 = \cdots = \lambda^{n-1} = 0$, $\lambda^n = 1$ we obtain the center objective, if $\lambda^1 = \lambda^2 = \cdots = \lambda^{n-1} = \alpha$, $\lambda^n = 1$ we obtain a convex combination of median and center objectives (centdian), etcetera.

We define the p-facility Discrete Ordered Median Problem as determining the subset J, of p facilities to open in order to minimize the ordered median function:

$$\min_{J \subset I: |J| = p} \sum_{k=1}^{n} \lambda^k c^k_{\le}(J). \tag{DOMP}$$

2.1. Three-index formulation

The formulation that we present below, denoted by ($DOMP_1$), was introduced by Boland et al. [3]. It uses three-index variables $x_{ij}{}^k$ such that $x^k_{ij} = 1$, if client i is served by facility j and cost $c_i(J) = C_{ij}$ is the kth smallest in the ordered sequence $c_{\le}(J)$, $x^k_{ij} = 0$ otherwise. Further, it also uses location variables y_j such that $y_j = 1$ if $j \in J$ and $y_j = 0$ otherwise.

If $x^k_{ij} = 1$, we say that allocation of client i to facility j is in position k, or that couple ij is in position k:

$$(DOMP_1) \quad \min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \lambda^k C_{ij} x^k_{ij} \tag{2}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} \sum_{k=1}^{n} x^k_{ij} = 1 \quad i = 1, \ldots, n \tag{3}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x^k_{ij} = 1 \quad k = 1, \ldots, n \tag{4}$$

$$\sum_{k=1}^{n} x^k_{ij} \le y_j \quad i, j = 1, \ldots, n \tag{5}$$

$$\sum_{j=1}^{n} y_j = p \tag{6}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x^{k-1}_{ij} \le \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x^k_{ij} \quad k = 2, \ldots, n \tag{7}$$

$$x^k_{ij} \in \{0, 1\} \quad i, j, k = 1, \ldots, n \tag{8}$$

$$y_j \in \{0, 1\} \quad j = 1, \ldots, n. \tag{9}$$

By means of (3) we ensure that each location is served by exactly one facility. In the same way, in each position there must be exactly one allocation (4). We know that a client can be allocated to a facility only if this facility is open, i.e. $x^k_{ij} \le y_j$ for all i, j, k. Furthermore, each allocation of client to facility can be placed in at most one position. Hence, $x^k_{ij} \le y_j$ can be strengthened yielding constraint (5). The equality constraint (6) implies that there are exactly p open facilities. Inequality (7) imposes that the allocation cost in position $k-1$ cannot be greater than the one in position k. Finally, the variables are binary, see (8) and (9).

2.2. Two-index formulation

This formulation ($DOMP_2$) was described for the first time in Puerto [16] and Marín et al. [10] and later applied to a hub problem in Puerto et al. [17]. It considers a vector that contains all the different values in the cost matrix C, augmented with zero if it is not present in matrix C, as it is explained below.

Let C be a matrix and assume that it contains G different values such that $c_{ij} > 0$. Then, the $(G+1)-$dimensional vector $c_{(.)}$ is constructed as follows:

$c_{(0)} = 0 < c_{(1)} < c_{(2)} < \cdots < c_{(G-1)} < c_{(G)} = \max\{C_{ij}: i, j = 1, \ldots, n\}$.

To formulate the problem we need to define the following binary variables. Variable $x_{ij} = 1$ if client i is served by facility j and 0 otherwise, variable $y_j = 1$ if $j \in J$ and 0 otherwise and variable $u_{kh} = 1$ if the kth smallest allocation cost is greater than $c_{(h-1)}$ and 0 otherwise. Further, we set $u_{k0} = 1$ and $u_{k,G+1} = 0$, $k = 1, \ldots, n$.

The problem to solve is

$$(DOMP_2) \quad \min \sum_{k=1}^{n} \sum_{h=1}^{G} \lambda^k (c_{(h)} - c_{(h-1)}) u_{kh} \tag{10}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} y_j = p \tag{11}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, \dots, n \tag{12}$$

$$x_{ij} \le y_j \quad i, j = 1, \dots, n \tag{13}$$

$$u_{kh} \ge u_{k,h+1} \quad k = 1, \dots, n, \quad h = 1, \dots, G-1 \tag{14}$$

$$u_{k+1,h} \ge u_{kh} \quad k = 1, \dots, n-1, \quad h = 1, \dots, G \tag{15}$$

$$\sum_{i=1}^{n} \sum_{\substack{j=1: \\ c_{ij} > c_{(h-1)}}}^{n} x_{ij} = \sum_{k=1}^{n} u_{kh} \quad h = 1, \dots, G \tag{16}$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n \tag{17}$$

$$u_{kh} \in \{0, 1\} \quad k = 1, \dots, n, \quad h = 1, \dots, G \tag{18}$$

$$y_j \in \{0, 1\} \quad j = 1, \dots, n. \tag{19}$$

The objective function (10) is equivalent to (1). Assume that the kth smallest allocation cost is equal to $c_{(h_k)}$ for some h_k; then by the definition of the variable u_{kh_k}

$$\sum_{h=1}^{G} (c_{(h)} - c_{(h-1)}) u_{kh} = \sum_{h=1}^{h_k} (c_{(h)} - c_{(h-1)}) = c_{(h_k)} - c_{(0)} = c_{(h_k)}$$

provided that $u_{kh_k} = 1$ and $u_{k,h_k+1} = 0$. Finally,

$$\sum_{k=1}^{n} \sum_{h=1}^{G} \lambda^k (c_{(h)} - c_{(h-1)}) u_{kh} = \sum_{k=1}^{n} \lambda^k \sum_{h=1}^{G} (c_{(h)} - c_{(h-1)}) u_{kh} = \sum_{k=1}^{n} \lambda^k c_{(h_k)}$$
$$= \sum_{i=1}^{n} \lambda_i c_i(J).$$

The equality constraint (11) ensures that there are exactly p facilities to be located. Constraints (12) and (13) state that each client is served by one open facility. Equality (16) ensures a good definition of the variable u_{kh} and it relates the sorting (u_{kh}) and design variables (x_{ij}). We need to impose some sorting constraints on the u_{kh} variables (15). Constraints (14) are redundant but they are included because, according to Marín et al. [10], it significantly strengthen the formulation. Furthermore, all variables are binary (17)–(19).

Note that others two index formulations has been proposed in Marín et al. [10,11]. However, they are only valid if $c_{ii} = 0 \ \forall\, i = 1, \dots, n$ (*free self-service*).

2.3. A new formulation for DOMP

There are several formulations for DOMP but they all have large integrality gap, as observed previously in the literature see Boland et al. [3], Marín et al. [10,11] and Puerto [16]. The main motivation for addressing a new formulation for the DOMP relies on the attempt to reduce this gap.

First, we introduce the following notation:

$$ij \prec \hat{\imath}\hat{\jmath} \equiv \begin{cases} c_{ij} < c_{\hat{\imath}\hat{\jmath}} \\ \text{or} \\ c_{ij} = c_{\hat{\imath}\hat{\jmath}} \ \text{ and } \ i < \hat{\imath} \\ \text{or} \\ c_{ij} = c_{\hat{\imath}\hat{\jmath}}, \ i = \hat{\imath} \ \text{ and } \ j < \hat{\jmath} \end{cases} \qquad ij \succ \hat{\imath}\hat{\jmath} \equiv \begin{cases} c_{ij} > c_{\hat{\imath}\hat{\jmath}} \\ \text{or} \\ c_{ij} = c_{\hat{\imath}\hat{\jmath}} \ \text{ and } \ i > \hat{\imath} \\ \text{or} \\ c_{ij} = c_{\hat{\imath}\hat{\jmath}}, \ i = \hat{\imath} \ \text{ and } \ j > \hat{\jmath} \end{cases}$$
$$ij \preceq \hat{\imath}\hat{\jmath} \equiv \begin{cases} ij \prec \hat{\imath}\hat{\jmath} \\ \text{or} \\ ij = \hat{\imath}\hat{\jmath} \end{cases} \qquad ij \succeq \hat{\imath}\hat{\jmath} \equiv \begin{cases} ij \succ \hat{\imath}\hat{\jmath} \\ \text{or} \\ ij = \hat{\imath}\hat{\jmath}. \end{cases} \tag{20}$$

The above relationship induces a strict total order among the couples ij. Its use avoids to consider multiple, symmetric feasible solutions coming from the structure of the matrix C.

2.3.1. New three index formulation

Our new formulation uses the same variables and constraints as in the three-index formulation, $DOMP_1$, except that (7) are replaced by (21) that are called *order constraints*.

The resulting formulation is denoted $DOMP_3$:

$$(DOMP_3) \quad \min \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \lambda^k c_{ij} x_{ij}^k$$

s. t. (3), (4), (5), (6), (8), (9)

$$\sum_{\hat{\imath}=1}^{n} \sum_{\substack{\hat{\jmath}=1: \\ \hat{\imath}\hat{\jmath} \succeq ij}}^{n} x_{\hat{\imath}\hat{\jmath}}^{k-1} + \sum_{\hat{\imath}=1}^{n} \sum_{\substack{\hat{\jmath}=1: \\ \hat{\imath}\hat{\jmath} \preceq ij}}^{n} x_{\hat{\imath}\hat{\jmath}}^{k} \quad i, j = 1, \dots, n, \ k = 2, \dots, n. \tag{21}$$

The rationale behind constraints (21) is illustrated by the following example.

Example 1. Consider the following matrix.

$$C = \begin{pmatrix} 0 & 2 & 7 & 4 \\ 1 & 0 & 5 & 5 \\ 3 & 6 & 0 & 2 \\ 9 & 4 & 1 & 0 \end{pmatrix}$$

The order of couples ij by means of the above preference order is

$11 \prec 22 \prec 33 \prec 44 \prec 21 \prec 43 \prec 12 \prec 34 \prec 31 \prec 14 \prec 42 \prec 23 \prec 24 \prec 32 \prec 13 \prec 41.$

The columns of Fig. 1 represent the n^2 possible assignments of clients to facilities whereas its rows represent the n positions in DOMP objective function. Each point (bullet or circle) thus represents a variable $x_{ij}{}^k$ and the bullets correspond to variables which cannot take value 1 simultaneously because, in any feasible solution, the cost of couples assigned to consecutive positions should be non-decreasing.

So, Fig. 1 corresponds to the following inequality:

$$\sum_{i=1}^{n} \sum_{\substack{j=1: \\ \hat{\imath}\hat{\jmath} \succeq 43}}^{n} x_{\hat{\imath}\hat{\jmath}}^{2} + \sum_{i=1}^{n} \sum_{\substack{j=1: \\ \hat{\imath}\hat{\jmath} \preceq 43}}^{n} x_{\hat{\imath}\hat{\jmath}}^{3} \le 1,$$

or equivalently

$$x_{11}^3 + x_{22}^3 + x_{33}^3 + x_{44}^3 + x_{21}^3 + x_{43}^3 + x_{43}^2 + x_{12}^2 + x_{34}^2 + x_{31}^2 + x_{14}^2 + x_{42}^2$$
$$+ x_{23}^2 + x_{24}^2 + x_{32}^2 + x_{13}^2 + x_{41}^2 \le 1.$$

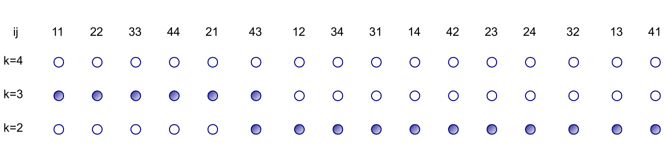

Fig. 1. An order constraint for the case, $n = 4$.

Note that the number of order constraints is $O(n^3)$. Further they can be seen as cliques of a conflict graph induced by the incompatibility among $x_{ij}{}^k$ variables on three index formulations [12,7].

In a similar way, we can choose several allocations which are sorted and identify a new type of constraints. Let s be a positive integer with $s \le n-1$ and let $(i_1 j_1), (i_2 j_2), \dots, (i_s j_s)$ be s couples of

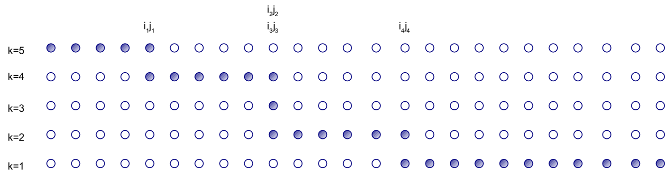

Fig. 2. A staircase constraint for the case, $n = 5$.

clients and facilities such that $i_rj_r \le i_{r+1}j_{r+1}$ for all $r = 1, \dots, s-1$. Then the following family of inequalities, called *staircase inequalities* is valid for $k = s, \dots, n$:

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_1j_1}}^{n} x_{ij}^{k} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_rj_r\\ ij\le i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} + \sum_{i=1}^{n}\sum_{\substack{j=1:\\ ij\ge i_sj_s}}^{n} x_{ij}^{k-s} \le 1. \tag{22}$$

Fig. 2 provides an example of staircase inequality when $n=5$.

Notice that there exists an exponential number of additional staircase constraints. But the question is whether these new constraints actually strengthen our formulation. The answer is provided by the following result that states that all of them are implied by those with a single step, i.e. the order constraints.

Proposition 1. *Staircase inequalities (22) with $i_rj_r \le i_{r+1}j_{r+1}$ can be obtained as an affine combination of (21) and (4).*

Proof. Let s be a positive integer such that $s \le n-1$, then by (21) we obtain that also the following s inequalities hold:

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_rj_r}}^{n} x_{ij}^{k-(r-1)} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_rj_r}}^{n} x_{ij}^{k-r} \le 1 \quad r = 1, \dots, s. \tag{23}$$

Now, since $i_rj_r \le i_{r+1}j_{r+1}$ using (4), we obtain the following $s-1$ equations:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^{k-r} = 1 \quad r = 1, \dots, s-1. \tag{24}$$

Adding all inequalities (23) we obtain a new inequality:

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_1j_1}}^{n} x_{ij}^{k} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_1j_1}}^{n} x_{ij}^{k-1} + \cdots + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_sj_s}}^{n} x_{ij}^{k-(s-1)} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_sj_s}}^{n} x_{ij}^{k-s} \le s.$$

After rearranging, we get

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_1j_1}}^{n} x_{ij}^{k} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_rj_r}}^{n} x_{ij}^{k-r} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_sj_s}}^{n} x_{ij}^{k-s}$$
$$\le s.$$

Next, we conveniently split some terms of the above inequality to get:

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_1j_1}}^{n} x_{ij}^{k} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_rj_r\\ ij\le i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij> i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r}$$
$$+ \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_rj_r}}^{n} x_{ij}^{k-r} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij> i_rj_r\\ ij\le i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_sj_s}}^{n} x_{ij}^{k-s}$$
$$\le 1 + (s-1). \tag{25}$$

On the other hand, using equality (24) we can write

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_rj_r}}^{n} x_{ij}^{k-r} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij> i_rj_r\\ ij\le i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij> i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} = 1 \quad r$$
$$= 1, \dots, s-1.$$

Adding the above equations for all $r = 1, \dots, s-1$, we obtain

$$\sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_rj_r}}^{n} x_{ij}^{k-r} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij> i_rj_r\\ ij\le i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij> i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r}$$
$$= s-1.$$

Finally, using the above equation in (25) results in

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\le i_1j_1}}^{n} x_{ij}^{k} + \sum_{r=1}^{s-1}\sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_rj_r\\ ij\le i_{r+1}j_{r+1}}}^{n} x_{ij}^{k-r} + \sum_{i=1}^{n} \sum_{\substack{j=1:\\ ij\ge i_sj_s}}^{n} x_{ij}^{k-s} \le 1,$$

which is a staircase inequality. □

2.3.2. A two index formulation with scheduling constraints

Formulation $DOMP_3$ is rather efficient and tight whenever there are few ties in the structure of the allocation costs. This is for instance the case of problems with assignment costs based on flat costs (for instance randomly generated). However, if the number of ties in the allocation costs is large the number of binary variables and constraints is relatively large, as compared with similar numbers in formulation $DOMP_2$. In order to exploit this advantage without losing the usage of scheduling constraints, that relate the formulation with the stable set problem, we will develop in the following another formulation.

The new formulation, called $DOMP_{3C}$, is an extension of $DOMP_3$ using the rationale of $DOMP_2$. Moreover, it provides a compact form for representing ties in the allocation costs.

Consider a new set of binary variables, v_{kh} such that $v_{kh} = 1$ if the kth smallest allocation cost is $c_{(h)}$ and 0 otherwise. Next, $DOMP_{3C}$ is a new valid formulation for DOMP:

$$(DOMP_{3C}) \quad \min \quad \sum_{k=1}^{n}\sum_{h=0}^{G} \lambda^k c_{(h)} v_{kh} \tag{26}$$

s. t. (11), (12), (13)

$$\sum_{h=0}^{G} v_{kh} = 1 \quad k = 1, \dots, n \tag{27}$$

$$\sum_{i=1}^{n} \sum_{\substack{j=1:\\ C_{ij}=c_{(h)}}}^{n} x_{ij} = \sum_{k=1}^{n} v_{kh} \quad h = 0, \dots, G \tag{28}$$

$$\sum_{h'<h} v_{k+1,h'} + \sum_{h'\ge h} v_{kh'} \le 1 \quad \begin{array}{l} k = 1, \dots, n-1 \\ h = 1, \dots, G \end{array} \tag{29}$$

$$x_{ij}, y_j, v_{kh} \in \{0,1\} \quad \begin{array}{l} i,j,k = 1, \dots, n \\ h = 0, \dots, G. \end{array} \tag{30}$$

Clearly, the objective function (26) accounts for the ordered weighted sum of the allocation costs. Constraints (28) state that the number of allocations that are attained at the value $c_{(h)}$ regardless of the level k that they occupy (v_{kh} variables) must be equal to the number of allocations of clients i to facility j with ij such that C_{ij} is equal to $c_{(h)}$ (see Fig. 3). Finally, constraints (29) are scheduling constraints based on costs values rather than in couples ij of client-facility (see Fig. 4).

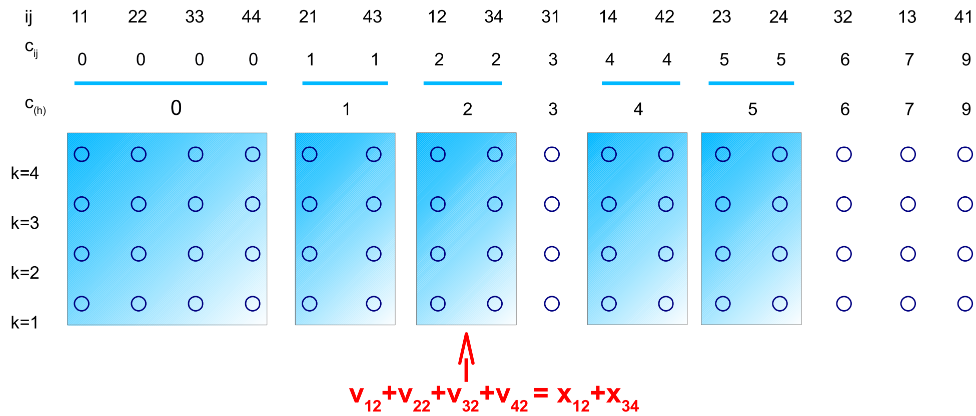

Fig. 3. The rationale of constraints (28).

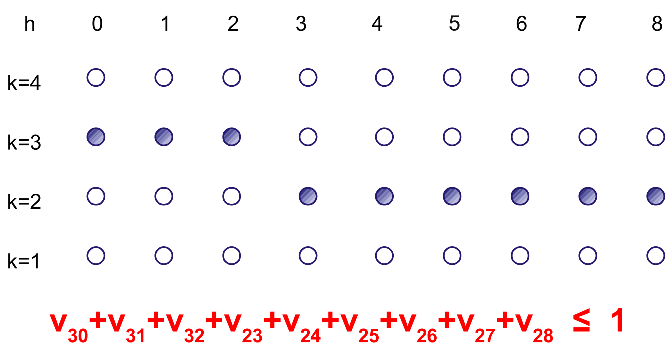

Fig. 4. The rationale of constraints (29).

2.4. An aggregated formulation

Here, we introduce another formulation ($DOMP_4$) based on the aggregation of order constraints from $DOMP_3$ corresponding to the same position. It therefore requires a smaller number of constraints:

$$(DOMP_4) \quad \min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \lambda^k c_{ij} x_{ij}^k$$

s. t. (3), (4), (5), (6), (8), (9)

$$\sum_{i}^{n}\sum_{j}^{n}\left(\sum_{i'=1}^{n}\sum_{\substack{j'=1:\\ i'j'\le ij}}^{n} x_{i'j'}^{k} + \sum_{i'=1}^{n}\sum_{\substack{j'=1:\\ i'j'\ge ij}}^{n} x_{i'j'}^{k-1}\right) \le n^2 \quad k=2,\dots,n. \tag{31}$$

The new constraints (31), that we call *weak order constraints*, ensure that if a couple ij occupies the kth position then in $(k-1)$th position there must be a more preferred allocation. It is due to the coefficients of each variable in the inequality. In each constraint there are two different positions, k and $k-1$, so that, by (4), two variables must take value one and all the others will be equal to zero. If we do not take into account the variables taking the value zero and we assume that the variables with value one for positions k and $k-1$ are in position s and t, respectively, we have the following:

$$(n^2-(s-1))x_{i_s j_s}^{k} + t x_{i_t j_t}^{k-1} \le n^2,$$

which is valid if and only if $t < s$.

Aggregating the scheduling constraints in $DOMP_{3C}$ for the different values of the costs, namely in h=1,…,G, results in a new valid model. This model is the aggregated version of $DOMP_{3C}$ that we denote as $DOMP_{4C}$:

$$(DOMP_{4C}) \quad \min \quad \sum_{k=1}^{n}\sum_{h=0}^{G} \lambda^k c_{(h)} v_{kh}$$

s. t. (11), (12), (13), (27), (28), (30)

$$\sum_{h=1}^{G}\left(\sum_{h'<h} v_{kh'} + \sum_{h'\ge h} v_{k-1h'}\right) \le G \quad k=2,\dots,n. \tag{32}$$

Using the rationale of (7), since there is only one binary variable v_{kh} in each position k by (27), the following constraints are valid inequalities for both formulations $DOMP_{3C}$ and $DOMP_{4C}$:

$$\sum_{h=1}^{G} c_{(h)} v_{k-1h} - \sum_{h=1}^{G} c_{(h)} v_{kh} \le 0 \quad k=2,\dots,n. \tag{33}$$

In fact, these constraints can also define another valid formulation for DOMP replacing (32) by (33) in the above formulation. In our experiments, we will not use this last possibility and instead, we shall use (33) as valid inequalities to strengthen $DOMP_{3C}$ and $DOMP_{4C}$.

3. Theoretical results

In this section, we provide a theoretical comparison of the four formulations presented in Section 2 and some polyhedral results regarding our formulation $DOMP_3$. Our goal is to state the formal relationships between the lower bounds provided by the linear relaxations of the considered formulations. In addition, we also give some families of tight valid inequalities which are proven to be facets of the polytope defined by assignment constraints (see Section 3.2.)

3.1. Comparison of formulations

We denote by $z_l(\cdot)$ the value of the objective function of $DOMP_l$ evaluated at the point $(\cdot)$, by P_l the polytope defining the feasible set of the linear relaxation of formulation $DOMP_l$, and by P_l^I the convex hull of the integer solutions within that polytope.

Consider the following mapping:

$$f: [0,1]^{n^3}\times[0,1]^{n} \longrightarrow [0,1]^{n^2}\times[0,1]^{n}\times[0,1]^{nG}$$
$$(x_{ij}^k, y_j) \longmapsto (x_{ij}, y_j, u_{kh})$$

defined by the following two equations:

$$x_{ij} = \sum_{k=1}^{n} x_{ij}^k \quad i,j=1,\dots,n \tag{34}$$

and

$$u_{kh}=\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k}\quad k=1,\dots,n,\ \ h=1,\dots,G. \tag{35}$$

For $h=1,\dots,G$, let us define $k(h)=\min\{k\colon u_{kh}-u_{k,h+1}>0\}$ being $u_{k,G+1}=u_{kG}$. We assume that if $u_{kh}-u_{k,h+1}=0$ for all k, $k(h)=+\infty$. Next, for each $h=1,\dots,G$ let $x_{(ij)_h}$ be the non-null variable $x_{ij}>0$ such that the couple (ij) is the most preferred, in the pairwise strict order introduced in (20), among those satisfying $C_{ij}=c_{(h)}$.

Observe that this couple can be formally defined as the minimal in the order induced by the relation $\prec$among those with $C_{ij}=c_{(h)}$, namely:

$$(ij)_h=\min_{(\prec)}\{ij\colon x_{ij}>0 \text{ and } C_{ij}=c_{(h)}\}.$$

Based on the above couples, for any feasible solution $(x_{ij},y_j,u_{kh})\in P_2$ we construct, sequentially, a feasible solution of P_1. Indeed, for each $h=1,\dots,G$ and $C_{ij}=c_{(h)}$, we construct sequentially in the strict order given by (20) and from $k=1$ until $k=n$:

$$x_{ij}^{k}=\begin{cases}0 & \text{if } k<k(h) \text{ or } x_{ij}=0,\\ \min\left\{x_{ij}-\sum_{\ell<k}x_{ij}^{\ell},\ u_{kh}-u_{k,h+1}\right\} & \text{if } k\geq k(h) \text{ and } ij=(ij)_h,\\ \min\left\{x_{ij}-\sum_{\ell<k}x_{ij}^{\ell},\ u_{kh}-u_{k,h+1}-\sum_{(i'j')\prec ij}x_{i'j'}^{k}\right\} & \text{if } k\geq k(h) \text{ and } ij\succ(ij)_h.\end{cases} \tag{36}$$

Now, using the above definition we introduce the mapping g:

$$\begin{aligned} g\colon [0,1]^{n^2}\times[0,1]^{n}\times[0,1]^{nG} &\longrightarrow [0,1]^{n^3}\times[0,1]^{n}\\ (x_{ij},y_j,u_{kh}) &\longmapsto (x_{ij}^{k},y_j)\end{aligned}$$

where ${x_{ij}}^{k}$ is given by (36).

These two mappings f and g relate the space of feasible solutions to $DOMP_1$, $DOMP_3$ and $DOMP_4$ with the space of feasible solutions to $DOMP_2$ and conversely.

Observation 1.

- *For any points $(x_{ij}^{k},y_j)\in P_l$ and $f(x_{ij}^{k},y_j)$, $z_l(x_{ij}^{k},y_j)=z_2(f(x_{ij}^{k},y_j))$ for $l=1,3,4$.*
- *For any points $(x_{ij},y_j,u_{kh})\in P_2$ and $g(x_{ij},y_j,u_{kh})$, $z_2(x_{ij},y_j,u_{kh})=z_l(g(x_{ij},y_j,u_{kh}))$ for $l=1,3,4$.*

We begin by analyzing the strength of the lower bounds provided by the continuous relaxation of formulations $DOMP_l$ for $l=1,2,3,4$.

Theorem 1. *Let $p=(x_{ij},y_j,u_{kh})$. If $p\in P_2$ then $g(p)\in P_1$.*

Proof. Let $(x_{ij},y_j,u_{kh})\in P_2$. By construction of (36), we have that for any k, $\sum_{i=1}^{n}\sum_{j=1:\, C_{ij}=c_{(h)}}^{n} x_{ij}^{k}=u_{kh}-u_{k,h+1}$. Moreover, by (16), we get $\sum_{i=1}^{n}\sum_{j=1:\, C_{ij}=c_{(h)}}^{n} x_{ij}=\sum_{k=1}^{n}(u_{kh}-u_{k,h+1})$, and adding over k proves (35).

Thus, it follows that $\sum_{i=1}^{n}\sum_{j=1:\, C_{ij}=c_{(h)}}^{n} x_{ij}=\sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{j=1:\, C_{ij}=c_{(h)}}^{n} x_{ij}^{k}$ and then by the construction in (36) we obtain that $x_{ij}=\sum_{k=1}^{n}x_{ij}^{k}$. This argument proves that the point (x_{ij}^{k},y_j), provided by (36), also satisfies (34).

Next, since $(x_{ij},y_j,u_{kh})\in P_2$, it satisfies (12), (13) and (11) and using that $x_{ij}=\sum_{k=1}^{n}x_{ij}^{k}$ it follows that (x_{ij}^{k},y_j) fulfills (3), (5) and (6), respectively.

Further, note that

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}=c_{(h)}}}^{n} x_{ij}^{k}=\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k}-\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}>c_{(h)}}}^{n} x_{ij}^{k}=u_{kh}-u_{k,h+1}.$$

Hence,

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij}^{k}&=\sum_{h=0}^{G}\left(\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}=c_{(h)}}}^{n}x_{ij}^{k}\right)=\sum_{h=0}^{G}(u_{kh}-u_{k,h+1})=u_{k0}-u_{k,G+1}\\ &=1-0=1,\end{aligned}$$

i.e. the point (x_{ij}^{k},y_j) satisfies (4).

Now we show that (15) and (35) imply (7). Replacing the variables in (15) using (35) we obtain: $\sum_{i=1}^{n}\sum_{j=1:\, C_{ij}\geq c_{(h)}}^{n} x_{ij}^{k}\geq\sum_{i=1}^{n}\sum_{j=1:\, C_{ij}\geq c_{(h)}}^{n} x_{ij}^{k-1}$, which implies that

$$(c_{(h)}-c_{(h-1)})\left(\sum_{\substack{i=1\\ :C_{ij}\geq}}^{n}\sum_{\substack{j=1\\ c_{(h)}}}^{n}x_{ij}^{k}-\sum_{\substack{i=1\\ :C_{ij}\geq}}^{n}\sum_{\substack{j=1\\ c_{(h)}}}^{n}x_{ij}^{k-1}\right)\geq 0\quad\forall\, h=1,\dots,G.$$

Next, adding the above inequalities for all h yields

$$\begin{aligned}&\sum_{h=1}^{G-1}\left[c_{(h)}\left(\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(h)}}}^{n}x_{ij}^{k}-\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(h)}}}^{n}x_{ij}^{k-1}\right)\right.\\ &\qquad\left.-c_{(h)}\left(\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(h+1)}}}^{n}x_{ij}^{k}-\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(h+1)}}}^{n}x_{ij}^{k-1}\right)\right]\\ &\qquad+c_{(G)}\left(\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(G)}}}^{n}x_{ij}^{k}-\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(G)}}}^{n}x_{ij}^{k-1}\right)\\ &\qquad-c_{(0)}\left(\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(1)}}}^{n}x_{ij}^{k}-\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}\geq c_{(1)}}}^{n}x_{ij}^{k-1}\right)\geq 0.\end{aligned}$$

Given that $c_{(0)}=0$, we get

$$\sum_{h=1}^{G}\left[c_{(h)}\left(\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}=c_{(h)}}}^{n}x_{ij}^{k}-\sum_{i=1}^{n}\sum_{\substack{j=1:\\ C_{ij}=c_{(h)}}}^{n}x_{ij}^{k-1}\right)\right]\geq 0$$

which is equivalent to

$$\sum_{i=1}^{n}\sum_{j=1}^{n}C_{ij}x_{ij}^{k}-\sum_{i=1}^{n}\sum_{j=1}^{n}C_{ij}x_{ij}^{k-1}\geq 0.$$

Finally, since $(x_{ij},y_j,u_{kh})\in P_2$ it satisfies $0\leq x_{ij}\leq 1$, $0\leq u_{kh}\leq 1$. Hence, by (34) and (36) $0\leq x_{ij}^{k}\leq 1$. Furthermore $0\leq y_j\leq 1$ is satisfied. □

Next, we prove the relationship between the feasible regions of formulations $DOMP_3$ and $DOMP_2$.

Theorem 2. $f(P_3)\subset P_2$.

Proof. Let $(x_{ij}^{k},y_j)\in P_3$, by (3) and (34), (12) is satisfied and by (5) and (34), (13) is satisfied.

We observe that (x_{ij}^{k},y_j) satisfies

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k} \geq \sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h+1)}}}^{n} x_{ij}^{k}.$$

Then using (35) we obtain $u_{kh} \geq u_{k,h+1}$ which proves (14).

In order to verify (15), by (4) we know that

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}<c_{(h)}}}^{n} x_{ij}^{k+1} + \sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k+1} = 1$$

and by (21)

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}<c_{(h)}}}^{n} x_{ij}^{k+1} + \sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k} \leq 1.$$

So,

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k+1} \geq \sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k}$$

and, using (35), constraint (15) is satisfied.

Eqs. (6) and (11) are the same.

It is clear that

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}>c_{(h-1)}}}^{n}\sum_{k=1}^{n} x_{ij}^{k} = \sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n}\sum_{k=1}^{n} x_{ij}^{k}.$$

By (34), the LHS is

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}>c_{(h-1)}}}^{n}\sum_{k=1}^{n} x_{ij}^{k} = \sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}>c_{(h-1)}}}^{n} x_{ij},$$

and by (35) the RHS is

$$\sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n}\sum_{k=1}^{n} x_{ij}^{k} = \sum_{k=1}^{n} u_{kh}.$$

By replacing both, (16) is satisfied.

In addition, all the variables are greater than or equal to zero and lower than or equal to one according with (34), (35), (3) and (4), since

$$x_{ij} = \sum_{k=1}^{n} x_{ij}^{k} \leq \sum_{j=1}^{n}\sum_{k=1}^{n} x_{ij}^{k} = 1,$$

$$u_{kh} = \sum_{i=1}^{n}\sum_{\substack{j=1:\\ c_{ij}\geq c_{(h)}}}^{n} x_{ij}^{k} \leq \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^{k} = 1. \quad \square$$

As shown by Example 2 (Cont'd), the inclusion of $f(P_3)$ into P_2 is strict, i.e. there exists a point in P_2 which cannot be obtained as the image of a point from P_3 by mapping f.

Example 2 (*continues=exa:cont*). We consider matrix C of Example 1. Further, $n=4$ and $p=2$. We choose the optimal solution of the linear relaxation of $DOMP_2$ with $\lambda = (1, 1, 1, 3)$ and observe that it is a fractional vertex of P_2 (see Tables 1 and 2).

Now, we show that there is no point in P_3 corresponding to that fractional solution in P_2. From $x_{ij} = \sum_{k=1}^{n} x_{ij}^{k}$ we deduce that $x_{ij}^{k} = 0$ for all i, j, k such that $x_{ij} = 0$. Next, from the equation $u_{kh} = \sum^{n}_{\substack{i,j=1\\ c_{ij}\geq c(h)}} x_{ij}^{k}$ we can conclude that point $(x_{ij}^{k}, y_j) \in P_3$ should satisfy:

$$x_{21}^{1} + x_{43}^{1} = \tfrac{1}{2}, \quad x_{21}^{2} + x_{43}^{2} = \tfrac{1}{2}, \quad x_{21}^{3} + x_{43}^{3} = \tfrac{1}{2}, \quad x_{21}^{4} + x_{43}^{4} = \tfrac{1}{2}. \qquad (37)$$

Further, constraints (4) for $k=2,3$ state that: $x_{11}^{2} + x_{33}^{2} + x_{21}^{2} + x_{43}^{2} = 1$ and $x_{11}^{3} + x_{33}^{3} + x_{21}^{3} + x_{43}^{3} = 1$, respectively. Thus, combining these two equations with those above it results that $x_{11}^{2} + x_{33}^{2} = \frac{1}{2}$ and $x_{11}^{3} + x_{33}^{3} = \frac{1}{2}$.

On the other hand, the order constraints (21) for $i = 2, j = 1, k = 2$ and $i = 2, j = 1, k = 3$ require that $x_{ij}{}^{k}$ fulfills $x_{11}^{2} + x_{33}^{2} + x_{21}^{2} + x_{21}^{1} + x_{43}^{1} \leq 1$ and $x_{11}^{3} + x_{33}^{3} + x_{21}^{3} + x_{43}^{3} + x_{43}^{2} \leq 1$, respectively. Combining these inequalities with the results above yields that $x_{21}^{2} = x_{43}^{2} = 0$ which contradicts the equation $x_{21}^{2} + x_{43}^{2} = \frac{1}{2}$ included in (37).

Our following result states the relationship between the feasible regions of the formulations $DOMP_3$ and $DOMP_1$.

Theorem 3. $P_3 \subset P_1$.

Proof. Using Theorems 1 and 2, it follows that

$P_3 \subset g(P_2) \subset P_1$. $\square$

Our next goal is to relate the polytope P_{3C} of the linear relaxation of $DOMP_{3C}$ with the previously considered polytopes.

Consider the following linear transformation:

$$\begin{aligned} L\colon [0,1]^{nG} &\longrightarrow [0,1]^{nG},\\ u_{kh} &\longmapsto v_{kh} \end{aligned}$$

defined by the following equations:

$v_{kG} = u_{kG}$

and

Table 2
x-values and **y**-values for a fractional feasible solution of Example 1 using formulation $DOMP_2$.

x_{ij}	$j=1$	$j=2$	$j=3$	$j=4$
$i=1$	1	0	0	0
$i=2$	1	0	0	0
$i=3$	0	0	1	0
$i=4$	0	0	1	0
y_j	1	0	1	0

Table 1
u-values for a fractional feasible solution of Example 1 using formulation $DOMP_2$.

u_{kh}	$c_{(0)} = 0$	$c_{(1)} = 1$	$c_{(2)} = 2$	$c_{(3)} = 3$	$c_{(4)} = 4$	$c_{(5)} = 5$	$c_{(6)} = 6$	$c_{(7)} = 7$	$c_{(8)} = 9$
$k=1$	1	$\frac{1}{2}$	0	0	0	0	0	0	0
$k=2$	1	$\frac{1}{2}$	0	0	0	0	0	0	0
$k=3$	1	$\frac{1}{2}$	0	0	0	0	0	0	0
$k=4$	1	$\frac{1}{2}$	0	0	0	0	0	0	0

$$v_{kh} = u_{kh} - u_{k,h+1} \quad k = 1, \dots, n, \;\; h = 0, \dots, G-1. \tag{38}$$

Consider as well the inverse L^{-1}

$$u_{kh} = \sum_{h'=h}^{G} v_{kh'} \quad k = 1, \dots, n, \;\; h = 0, \dots, G. \tag{39}$$

Let us denote by z_l^{LP} the optimal value of the LP-relaxation of $DOMP_l$ for $l = 1, 2, 3, 4, 3C, 4C$.

Theorem 4. *The linear relaxation of formulation $DOMP_{3C}$ is equal to the linear relaxation of formulation $DOMP_2$ modulo the linear transformation L, i.e. $L(P_2) = P_{3C}$ and $z_2^{LP} = z_{3C}^{LP}$.*

Proof. First, we check that both objective functions are equivalent,

$$\begin{aligned}
\sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k(c_{(h)} - c_{(h-1)})u_{kh} &= \sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k(c_{(h)} - c_{(h-1)})\left(\sum_{h'=h}^{G} v_{kh'}\right) \\
&= \sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k c_{(h)}\left(\sum_{h'=h}^{G} v_{kh'}\right) \\
&\quad - \sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k c_{(h-1)}\left(\sum_{h'=h}^{G} v_{kh'}\right) \\
&= \sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k c_{(h)} v_{kh} + \sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k c_{(h)}\left(\sum_{h'=h+1}^{G} v_{kh'}\right) \\
&\quad - \sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k c_{(h-1)}\left(\sum_{h'=h}^{G} v_{kh'}\right) \\
&= \sum_{k=1}^{n}\sum_{h=1}^{G}\lambda^k c_{(h)} v_{kh} = \sum_{k=1}^{n}\sum_{h=0}^{G}\lambda^k c_{(h)} v_{kh}.
\end{aligned}$$

Now, we show that the image $L^{-1}(P_{3C}) = P_2$. Then remarking that the linear transformation has full rank, the proof will be completed.

First, constraints (11), (12) and (13) are common to $DOMP3C$ and $DOMP2$.

Then, by definition, $u_{k0} = 1, \; k = 1, \dots, n$. So

$$\sum_{h'=0}^{G} v_{kh'} = 1.$$

From (15),

$$\sum_{h'=h}^{G} v_{k+1,h'} \ge \sum_{h'=h}^{G} v_{kh'},$$

which is equivalent to

$$1 - \sum_{h'=0}^{h-1} v_{k+1,h'} \ge \sum_{h'=h}^{G} v_{kh'}.$$

And this is constraint (29). Writing (16) for h and $h+1$, we get

$$\sum_{i=1}^{n}\sum_{\substack{j=1: \\ c_{ij} > c_{(h-1)}}}^{n} x_{ij} = \sum_{k=1}^{n} u_{kh} \quad \text{and} \quad \sum_{i=1}^{n}\sum_{\substack{j=1: \\ c_{ij} > c_{(h)}}}^{n} x_{ij} = \sum_{k=1}^{n} u_{k,h+1},$$

and subtracting the second from the first we obtain

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{\substack{j=1: \\ c_{ij} = c_{(h)}}}^{n} x_{ij} &= \sum_{k=1}^{n} u_{kh} - \sum_{k=1}^{n} u_{k,h+1} = \sum_{k=1}^{n}\sum_{h'=h}^{G} v_{kh'} - \sum_{k=1}^{n}\sum_{h'=h+1}^{G} v_{kh'} \\
&= \sum_{k=1}^{n} v_{kh},
\end{aligned}$$

which is equivalent to (28).

Finally, from (15) and (39) and the fact that $0 \le u_{kh} \le 1$, we obtain that $v_{kh} \in [0, 1]$. ◻

From the previous theorem one can easily obtain the relationship between the polytopes of feasible solutions of formulations $DOMP_{3C}$ and $DOMP_{4C}$.

Corollary 1. $P_{3C} \subset P_{4C}$.

We are now in position to present the overall relationship among the LP-relaxation values of all the considered formulations.

Corollary 2. $z_3^{LP} \ge z_{3C}^{LP} = z_2^{LP} \ge z_1^{LP}$, $z_3^{LP} \ge z_4^{LP}$ *and* $z_{3C}^{LP} \ge z_{4C}^{LP}$.

Proof. This is an immediate consequence of Observation 1 and Theorems 1–4. ◻

Observe that we have not proved a theoretical relationship between P_2 and P_4, but, as we will see in Section 4, empirically, we have observed that the solutions of the LP-relaxation of $DOMP_4$ provide in most cases a better bound than that of $DOMP_2$. However cases exist where it is the contrary. Theoretically, the formulation $DOMP_3$ gives a better or equal bound than $DOMP_2$ whose bound, as one can see in Section 4, concurs experimentally with the objective value of the linear relaxation of $DOMP_1$. Therefore, we know that $DOMP_3$ is the tightest formulation among the four presented in this paper.

Using similar arguments as those developed in the proofs of Theorems 1 and 2 we can formally state the validity of our formulations $DOMP_3$ and $DOMP_4$.

Theorem 5.

1. *If* $(x_{ij}, y_j, u_{kh}) \in P_2^I$ *then* $g(x_{ij}, y_j, u_{kh}) \in P_3^I$. *Conversely, for any* $(x_{ij}^k, y_j) \in P_3^I$, $f(x_{ij}^k, y_j) \in P_2^I$.
2. *If* $(x_{ij}, y_j, u_{kh}) \in P_2^I$ *then* $g(x_{ij}, y_j, u_{kh}) \in P_1^I$. *Conversely, for any* $(x_{ij}^k, y_j) \in P_1^I$, $f(x_{ij}^k, y_j) \in P_2^I$.
3. $P_3^I = P_4^I$.

Proof. We prove Theorem 5 in three steps.

1. $\mathbf{g(P_2^I) = P_3^I}$
   (a) $\mathbf{g(P_2^I) \subset P_3^I}$
   Let $(x_{ij}, y_j, u_{kh}) \in P_2^I$ and consider its projection, (x_{ij}^k, y_j) defined by (34), (35) and (36) onto P_3. This point satisfies (3), (5), (6) and (21).
   In order to see that the point (x_{ij}^k, y_j) satisfies (4), by (35) we have $\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k = u_{k0} \in \{0, 1\}$ for all $k = 1, \dots, n$ and by (12) it follows that $\sum_{i=1}^n \sum_{j=1}^n x_{ij} = n$. Then using (34) we obtain $\sum_{k=1}^n \left(\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k\right) = n$. Next, the only possibility for the above sum to be equal to n is that $\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k = 1$ for all $k = 1, \dots, n$, and (4) is satisfied.
   It remains to prove that (x_{ij}^k, y_j) is integer. It is clear that y_j is integer because it constitutes the same vector as in P_2. Finally, the ${x_{ij}}^k$ variables are also binary due to the fact that they are the product of binary variables, see (36).
   (b) $\mathbf{P_3^I \subset g(P_2^I)}$
   To prove the converse, we observe by that every integer solution in P_3^I provides a projected integer solution and we have seen in Theorem 2 that this point satisfies the inequalities defining P_2.
2. $\mathbf{P_1^I = g(P_2^I)}$
   (a) $\mathbf{g(P_2^I) \subset P_1^I}$
   Every integer solution in P_2 is projected into an integer solution and, from Theorem 1 this point satisfies the inequalities defining P_1.
   (b) $\mathbf{P_1^I \subset g(P_2^I)}$
   Conversely, let $(x_{ij}^k, y_j) \in P_1^I$ and its projection (x_{ij}, y_j, u_{kh}) by

means of (34) and (35).

It is easy to see that (3), (5) and (6) are equivalent to (12), (13) and (11), respectively.

Since all the variables are positive the point (x_{ij}^k, y_j) satisfies

$$\sum_{C_{ij}\geq c_{(h)}}^{n} x_{ij}^k \geq \sum_{C_{ij}\geq c_{(h+1)}}^{n} x_{ij}^k,$$

and (14) holds.

Furthermore, (16) holds by the change of variable defined in (34) and (35).

In inequalities (7), we can order the cost without loss of generality. Next by (4), there will be a unique variable with value one in each position. Therefore we obtain the following constraint for each different cost:

$$\sum_{C_{ij}\geq c_{(h)}}^{n} x_{ij}^{k+1} \geq \sum_{C_{ij}\geq c_{(h)}}^{n} x_{ij}^k \quad h = 1, \ldots, G, \ \ k = 1, \ldots, n-1$$

and point (x_{ij}, y_j, u_{kh}) satisfies (15). Furthermore, this point is integer.

3. $\mathbf{P_3^I = P_4^I}$

(a) $\mathbf{P_4^I \subset P_3^I}$

Let $(x_{ij}^k, y_j) \in P_4^I$. In particular, the weak order constraint is satisfied. By (4) we have

$$\sum_{\substack{i'=1\\ :i'j'}}^{n} \sum_{\substack{j'=1\\ \preceq ij}}^{n} x_{i'j'}^k \leq 1$$

and

$$\sum_{\substack{i'=1\\ :i'j'}}^{n} \sum_{\substack{j'=1\\ \succcurlyeq ij}}^{n} x_{i'j'}^{k-1} \leq 1,$$

where both inequalities are a sum of binary variables. So there are two possibilities:

$$\sum_{i'=1}^{n} \sum_{\substack{j'=1:\\ i'j'\preceq ij}}^{n} x_{i'j'}^k + \sum_{i'=1}^{n} \sum_{\substack{j'=1:\\ i'j'\succcurlyeq ij}}^{n} x_{i'j'}^{k-1} \leq 1, \quad \text{or} \tag{40}$$

$$\sum_{i'=1}^{n} \sum_{\substack{j'=1:\\ i'j'\preceq ij}}^{n} x_{i'j'}^k + \sum_{i'=1}^{n} \sum_{\substack{j'=1:\\ i'j'\succcurlyeq ij}}^{n} x_{i'j'}^{k-1} = 2. \tag{41}$$

In case (41), there will be two different costs $\hat{\imath}\hat{\jmath} \prec \tilde{i}\tilde{j}$ (note that $\hat{\imath}\hat{\jmath} = \tilde{i}\tilde{j}$ is not possible by (6)) such that $x_{\hat{\imath}\hat{\jmath}}^k = 1$ and $x_{\tilde{i}\tilde{j}}^{k-1} = 1$. This fact contradicts the ordering relationship. Thus, the only possibility is that case (40) holds and, in consequence, constraint (21) is satisfied.

(b) $\mathbf{P_3^I \subset P_4^I}$

Let (x_{ij}^k, y_j) be an integer point satisfying P_3^I. It is clear that this integer point also belongs to P_4^I. □

Corollary 3. $P_1^I = P_3^I = P_4^I = g(P_2^I)$.

3.2. On the polytope defined by the assignment constraints

The goal of this section is to provide some results about the facial structure of the polytope corresponding to the assignment constraints of formulation $DOMP_3$. We restrict ourselves to the analysis of formulation $DOMP_3$ since, according to Corollary 2, it gives the tightest formulation to DOMP among those studied in this paper. Similar studies of (simpler) polytopes related to location problems have been carried out by e.g. Arbib et al. [1], Guignard [8], Cornuéjols and Thizy [4], de Farias Jr. [5] and Vasilyev et al. [19].

For a given set J of p open facilities, we define the assignment polytope $P_3(J)$ of $DOMP_3$ as follows:

$$\sum_{j\in J} \sum_{k=1}^{n} x_{ij}^k \leq 1 \quad i = 1, \ldots, n \tag{42}$$

$$\sum_{i=1}^{n} \sum_{j\in J} x_{ij}^k \leq 1 \quad k = 1, \ldots, n \tag{43}$$

$$\sum_{k=1}^{n} x_{ij}^k \leq 1 \quad i = 1, \ldots, n, j \in J \tag{44}$$

$$\sum_{i'j'\succcurlyeq ij} x_{i'j'}^{k-1} + \sum_{i'j'\preceq ij} x_{i'j'}^k \leq 1 \quad i = 1, \ldots, n, j \in J, k = 2, \ldots, n \tag{45}$$

$$x_{ij}^k \geq 0 \quad i, k = 1, \ldots, n, j \in J. \tag{46}$$

We show which of the constraints that describe this polytope are facet inducing. Clearly, this contributes to the good quality of the LP-relaxation of $DOMP_3$, since this assignment polytope represents the underlying structure of the problem once the set of open facilities is determined.

We summarize the polyhedral properties of the convex hull of $P_3(J) \cap \{0, 1\}^{n^2\times p}$, namely $P_3^I(J)$, in the following result.

Proposition 2.

1. $dim(P_3(J)) = n^2 p$.
2. *Constraints (42) and (46) induce facets of* $P_3^I(J)$.

Proof. Constraints (42)–(46) define a particular packing polytope which has been studied by Padberg [15] among others. The results are consequences of this observation and the fact that variables appearing in (42) define maximal cliques in the conflict graph associated to the problem, see Padberg [15]. □

Observe that constraints (44) do not induce facets of $P_3(J)$ since they are dominated by constraints (42).

Let us denote by $(ij)^s$, $s = 1, \ldots, n^2$ the couple client i facility j where C_{ij} is the sth lowest cost over the cost matrix C. For instance, $(ij)^1$ and $(ij)^{n^2}$ are, respectively, the most and the least preferred couples in the sorted list of costs of the cost matrix C.

Definition 1. We call a pair ij generic if for some feasible set J such that $j \in J$, it satisfies

1. *Let* $\hat{\imath}\hat{\jmath}$ *be the couple client* $\hat{\imath}$ *facility* $\hat{\jmath} \in J$ *such that no couple* $i'j'$, $j' \in J$ *satisfying* $\hat{\imath}\hat{\jmath} \prec i'j' \prec ij$ *does exist, i.e.* $\hat{\imath}\hat{\jmath}$ *is the couple immediately before ij. Then,* $\hat{\imath} \neq i$.
2. *Let* $\tilde{\imath}\tilde{\jmath}$ *be the couple client* $\tilde{\imath}$ *facility* $\tilde{\jmath} \in J$ *such that no couple* $i'j'$, $j' \in J$ *satisfying* $ij \prec i'j' \prec \tilde{\imath}\tilde{\jmath}$ *does exist, i.e.* $\hat{\imath}\hat{\jmath}$ *is the couple immediately after ij. Then,* $\tilde{\imath} \neq i$.

Intuitively, a pair ij is generic with respect to a feasible solution set J if the remaining feasible allocation costs are well distributed around it. That is, there are costs of different clients surrounding the ij cost (C_{ij}) in the sorted list of costs.

Proposition 3. *If ij is generic for the feasible set J then*

$$\sum_{i'j'\succcurlyeq ij} x_{i'j'}^{k-1} + \sum_{i'j'\preceq ij} x_{i'j'}^k + \sum_{l=k+1}^{n} x_{(ij)^1}^l + \sum_{l=1}^{k-2} x_{(ij)^{n^2}}^l \leq 1 \quad k = 2, \ldots, n \tag{47}$$

is facet defining for the assignment polytope $P_3^I(J)$.

Proof. According to Definition 1, let us denote by $\hat{\imath}\hat{\jmath}$ and $\tilde{\imath}\tilde{\jmath}$ the

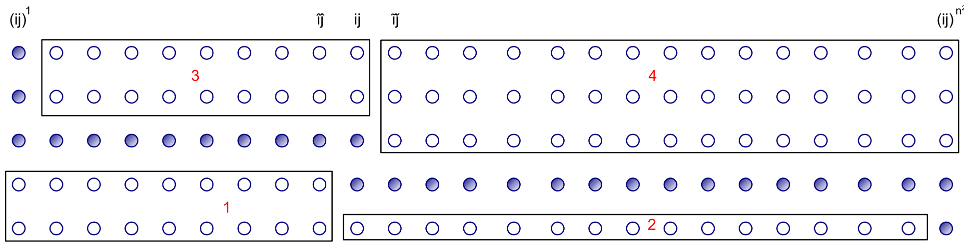

Fig. 5. Constraints (47)'s scheme.

couples immediately before and after the couple ij in the sorted list of costs (see Fig. 5).

In the following we prove that this family of constraints is facet defining inequalities showing that they are maximal cliques. Specifically, we show that no additional variable can be added to those inequalities which is, at the same time, incompatible with all of those that already appear in it.

Next we prove the above claim. We show that for each variable that does not belong to the considered clique, there is one in the clique compatible with it.

1. For all $i'j' \prec ij$, $l \le k-1$ if $i' \neq i$, variables $x^l_{i'j'}$ and ${x_{ij}}^k$ are compatible. Otherwise, variables $x^l_{i'j'}$ and $x^k_{\hat{i}\hat{j}}$ are compatible (Case 1 in Fig. 5).
2. For all $i'j' \succeq ij$, $l < k-1$ and $i'j' \neq (ij)^{n^2}$ if $i' \neq i$ variables $x^l_{i'j'}$ and ${x_{ij}}^k$ are compatible. Otherwise, variables $x^l_{i'j'}$ and $x^k_{\hat{i}\hat{j}}$ are compatible (Case 2 in Fig. 5).
3. For all $i'j' \preceq ij$, $l = k+1, \ldots, n$ and $i'j' \neq (ij)^1$ if $i' \neq i$, variables $x^l_{i'j'}$ and x^{k-1}_{ij} are compatible. Otherwise, variables $x^l_{i'j'}$ and $x^{k-1}_{\tilde{i}\tilde{j}}$ are compatible (Case 3 in Fig. 5).
4. For all $i'j' \succ ij$, $l = k, \ldots, n$ if $i' \neq i$ variables $x^l_{i'j'}$ and x^{k-1}_{ij} are compatible. Otherwise, variables $x^l_{i'j'}$ and $x^{k-1}_{\tilde{i}\tilde{j}}$ are compatible (Case 4 in Fig. 5). □

Remark that in our computational experiments, we did not reinforce constraints (45) because our preprocessing procedure (see Section 4) sets to zero all the variables appearing in these reinforcements associated with the costs of couples $(ij)^1$ and $(ij)^{n^2}$. Thus, in most cases in our formulation $DOMP_3$, after preprocessing those constraints are already facet inducing.

4. Computational study

In order to test the performance of our new formulations for DOMP, we have performed intensive computational tests comparing results with respect to previous available formulations of DOMP (see Boland et al. [3], Domínguez-Marín [6], Nickel [13], Nickel and Puerto [14]) and Marín et al. [11]).

4.1. Description of the test instances

We use two different types of instances. First, we consider random instances in which the elements of the cost matrix are integer numbers randomly generated between 10,000 and 100,000. The second set of instances consists in p-median instances from OR_Lib, Beasley [2].

Regarding the random instances; we vary the number of clients n in $\{10, 20, 30, 40, 50\}$ and for each n, we consider three possible values for the number of facilities to be open: $p = \lfloor \frac{n}{4} \rfloor, \lfloor \frac{n}{3} \rfloor, \lfloor \frac{n}{2} \rfloor$.

As for the p-median instances from Beasley's library, we have selected graphs corresponding to $p-med1, \ldots, p-med20$ with up to 400 nodes from the original data. Each set of nodes is divided in two disjoint subsets containing, respectively, the set of clients and the set of facilities. (The reader may observe that in these instances we force these two sets to be disjoint). Next, each cost C_{ij} is computed as the shortest path between client i and facility j in the resulting complete graph induced by the above described set of nodes.

Table 3
Types of λ – vectors used in experiments.

Notation	λ-vector	Name
T1	$(1, 1, \ldots, 1, 1)$	p-median
T2	$(0, 0, \ldots, 0, 1)$	p-center
T3	$(0, 0, \ldots, 0, 0, \underbrace{1, 1, \ldots, 1, 1}_{k})$	k-centrum
T4	$(\underbrace{0, 0, \ldots, 0, 0}_{k_1}, 1, 1, \ldots, 1, 1, \underbrace{0, 0, \ldots, 0, 0}_{k_2})$	(k_1+k_2)– trimmed mean
T5	$(0, 1, 0, 1, 0, 1, 0, 1, \ldots)$	–
T6	$(\ldots, 0, 0, 1, 0, 0, 1)$	–
T7	λ random	Random
T8	$(\alpha, \alpha, \ldots, \alpha, \alpha, 1)$	Centdian

Eight different types of λ-vectors are tested. Their description is provided in Table 3. We consider, among others, p-median, p-center, p–k-centrum, p-trimmed mean, random and p-α-centdian problems. In the k-centrum case, $k = \lfloor \frac{n}{2} \rfloor$. In the $(k_1 + k_2)$-trimmed mean, $k1 = k2 = \lfloor \frac{n}{10} \rfloor$. When λ is taken as a random vector we generate 5 instances which contain values randomly drawn between 1 and 100. Finally, we use $\alpha = 0.5$ in the centdian case.

For each type of data and each possible values of parameters n, p and vector λ, the results presented consist in average values over five instances. This results in an overall number of 900 tested instances.

4.2. Preprocessing

We present in this section two preprocessings that are based on a similar rationale to those already developed in [10,17]; although adapted to the new formulations on this paper. The first one is based on feasibility and the second on optimality.

Claim 1 (*Feasibility based preprocessing*).

1. *Let* $l(h) = |\{i: \exists\, j \in \{1, \ldots, n\}$ *satisfying* $C_{ij} \le c_{(h)}\}|$ *and* $u(h) = |\{i: \exists\, j \in \{1, \ldots, n\}$ *satisfying* $C_{ij} \ge c_{(h)}\}|$.
   Then,
   (a) $v_{kh} = 0$ *and* $u_{kh} = 1$ $k = l(h)+1, \ldots, n$, $h = 1 \ldots, G.$; *and*
   (b) $v_{kh} = 0$ *and* $u_{kh} = 0$ $k = 1, \ldots, n-u(h)$, $h = 1 \ldots, G.$
2. *Let* $l(ij) = |\{i': \min_{\hat{j}} i'\hat{j} \prec ij, i' \neq i\}|$ *and* $u(ij) = |\{i': \max_{\hat{j}} i'\hat{j} \succ ij, i' \neq i\}|$.
   Then,
   (a) $x^k_{ij} = 0$ *for all* $i, j = 1, \ldots, n$, $k = l(ij)+2, \ldots, n$; *and*
   (b) $x^k_{ij} = 0$ *for all* $i, j = 1, \ldots, n$, $k = 1, \ldots, n-u(ij)-1$.

This claim formalizes the fact that a given cost C_{ij} can appear in position k of a feasible solution only if there are at least $k-1$

Table 4
Number of variables fixed by preprocessing.

n (Random instances)	10 (%)	20 (%)	30 (%)	40 (%)	50 (%)	**Average** (%)
Claim 1.2	6.40	9.7	6.2	4.67	3.93	**8.07**
Claim 2	0.00	5.00	30.00	9.99	9.99	**7.00**
Total	29.68	29.61	30.43	32.28	31.97	**31.40**
n (Beasley instances)		50	100	150	200	**Average**
Claim 1.1		27.03	3.86	33.42	3.0	**30.83**
Claim 2		8.91	6.85	5.87	5.6	**6.7**
Total		27.38	30.33	29.67	27.12	**28.62**

Table 5
Average integrality gaps.

Formulation	**GAP (%)**
$DOMP_1$	7.34
$DOMP_2$	7.34
$DOMP_3$	2.29
$DOMP_4$	6.27
$DOMP_{4\cap 1}$	6.06

allocations costs lower than or equal to and $n-k$ greater than or equal to C_{ij}. We observe that this preprocessing can remove feasible solutions of the problem although it does not affect its solution since they cannot be optimal.

Claim 2. *Let ij be a couple client facility. If* $|\{j' : C_{ij} > C_{ij'}\}| > n - p$, *then* $x_{ij} = 0$ *and* $x_{ij}^k = 0$ *for* $k{=}1,\ldots,n$.

This second claim removes some feasible solutions of the problems which cannot be optimal because they are dominated for other feasible solutions with smaller objective value.

Table 4 shows the percentage of variables v_{kh}, u_{kh}, x_{ij} and ${x_{ij}}^k$ fixed by our preprocessing based on Claims 1 and 2. Notice that the percentage of two or three index variables fixed to zero in random instances is almost equal because assignment costs in random instances are almost all different. Furthermore, we observe that the percentage of fixed variables slightly decreases with n. Observe that Beasley instances have a larger number of ties in the distances (allocation costs). For this reason, these instances are solved using our compact formulations $DOMP_{3C}$ and $DOMP_{4C}$, whereas those with random data are solved with the formulations that do not take advantage of ties, namely $DOMP_3$ and $DOMP_4$. Hence, Beasley instances are preprocessed with Claim 1.1 (for v variables) and Random instances with Claim 1.2 (for ${x_{ij}}^k$ variables).

4.3. Computational results

All our experiments have been carried out on a PC with two Intel Xeon processors with 3.46 GHz and 48 GB of RAM. The models were written in Mosel and solved using Xpress IVE 7.3. To have a clean comparison of our solution approaches, all automatic cuts from Xpress have been disabled. We now report a summary of our computational experiments. Detailed information can be found in the material included in Appendix A of this paper. In particular, we report results for the different types of lambda vectors from Table 3.

4.3.1. Random allocation costs data sets

Table 5 provides a comparison of the LP-relaxations of models $DOMP_1$, $DOMP_2$, $DOMP_3$, $DOMP_4$ and $DOMP_{4\cap 1}$, averaging for $n{=}20$ and $n{=}40$ and all possible values of p and λ. The last model $DOMP_{4\cap 1}$ consists in $DOMP_4$ to which constraints (7) of $DOMP_1$ have been appended. Specifically, we report the integrality gap defined as $GAP = \frac{z^* - z_l^{LP}}{z^*}$, where z^* and ${z_l}^{LP}$ represent the optimal value of $DOMP_l$ and its LP-relaxation, respectively. We observe that the values of the integrality gap vary between 2.29% and 7.34%, obtained in formulations $DOMP_3$ the best and $DOMP_2$ or $DOMP_1$ the worst. In all cases, the LP-gaps are good but specially in formulation $DOMP_3$ which seems to be rather tight. We remark also the small difference that is obtained adding constraints (7) to the formulation $DOMP_4$ in terms of integrality gap. Moreover, we point out that we have obtained the same LP-gap with formulations $DOMP_1$ and $DOMP_2$ for all the tested instances. (It is still an open question whether this is also theoretically true.) Thus, from Table 5 one could conclude that the best formulation is $DOMP_3$. In spite of that, the large number of inequalities ($O(n^3)$) used in the model makes it rather slow whenever the number of clients n is of moderate size ($n > 50$).

Table 6
CPU-time and number of nodes of the different formulations for n = 10, 20, 30.

Solution approach	**Time**	**#nodes**
$DOMP_3(B\&B)$	463.24(8)	46.20
$DOMP_{4\cap 1}(B\&B)$	18.64	1389.51
$DOMP_{4\cap 1}(B\&C-3)$	52.25	24.94
$DOMP_4(B\&C-3)$	39.39	29.37

In order to define a solution approach which presents the best performance, we have conducted a preliminary computational test with instance sizes $n = 10, 20, 30$. Our first strategy consists in solving $DOMP_3$ with a pure branch-and-bound. Our second strategy, $DOMP_{4\cap 1}(B\&B)$, solves $DOMP_{4\cap 1}$ with a pure branch-and-bound. The third approach, $DOMP_{4\cap 1}(B\&C-3)$, starts by solving the LP-relaxation of $DOMP_4$ and then adds inequalities (7) at the root node and *order constraints* (21) as long as they are violated by the current solution of the LP. The reader may observe that both families of inequalities are cliques in the conflict graph induced by the three index variables of our formulation. Therefore, *order constraints* could in principle be added by standard clique cuts generation techniques implemented in Xpress. Nevertheless, our own implementation is more efficient since the separation of the entire family of valid inequalities (21) can be performed in $O(n^3)$ by sequentially updating the l.h.s. value of the *order constraints* when switching from a couple ij to the adjacent one in the same position k. The fourth and last strategy $DOMP_4(B\&C-3)$ is a branch and cut algorithm based upon $DOMP_4$ adding only valid inequalities from (21).

Our results are reported in Table 6. There we have included the CPU-times, in seconds, and the number of nodes in the B&B tree for solving instances to optimality within 2 h of CPU-time. The numbers between parentheses indicate the number of unsolved instances within the time limit. We observe that on average $DOMP_{4\cap 1}(B\&B)$ is the strategy that solves problems faster even though it has to visit the largest number of nodes in the B&B tree. This is explained by the fact that it is the most compact formulation (with the smallest number of inequalities) and therefore, it can be easily solved at each node. On the other hand, $DOMP_3(B\&B)$ is the heaviest one (in terms of LP representation) giving rise to worse CPU-times although it visits few nodes in the searching phase. In between, we found the two branch-and-cut procedures that we have tested $DOMP_{4\cap 1}(B\&C-3)$ and $DOMP_4(B\&C-3)$. In the implementation of this two *B&C* approaches we have tested the separation of maximal clique inequalities over the conflict graph as an alternative to our own separation procedure. Nevertheless, our separation algorithm applied on inequalities (7) and (21) gives better results. From the above two tables, we can conclude that the best strategies to be tested in the intensive computational tests are

Table 7
Summary of results with random matrices.

n	p	Time (# unsolved)	
		$DOMP_2$(B&B)	$DOMP_{4\cap 1}$(B&B)
10	2	1.12	0.42
10	3	0.54	0.25
10	5	0.18	0.09
20	5	22.21	5.80
20	6	9.05	4.09
20	10	3.33	1.54
30	7	161.32	121.04
30	10	66.85	22.70
30	15	33.15	11.87
40	10	449.11	625.68
40	13	232.79	174.62
40	20	116.99	49.13
50	12	1870.80(2)	3184.54(6)
50	16	988.08	804.36
50	25	771.51(2)	157.44
Average		**315.14**	**344.24**

Table 8
CPU-Time of the different formulations for different values of p.

p	$DOMP_2$(B&B)	$DOMP_{4\cap 1}$(B&B)
$\lfloor \frac{n}{4} \rfloor$	500.91(2)	787.49(6)
$\lfloor \frac{n}{3} \rfloor$	259.46	201.20
$\lfloor \frac{n}{2} \rfloor$	185.03(2)	44.01

$DOMP_{4\cap 1}(B\&B)$ and, at times, $DOMP_4(B\&C-3)$.

To finish, Table 7 allows us to compare our best strategy, i.e. $DOMP_{4\cap 1}(B\&B)$, with a branch-and-bound approach based on $DOMP_2$, as well as to determine the size of instances that can be solved within a reasonable time limit of 2 h. This table is organized in four columns. The first two columns show the size n and p of the instances. The last two columns show the average CPU-time in seconds necessary for solving those instances applying formulation $DOMP_2(B\&B)$ and $DOMP_{4\cap 1}(B\&B)$. The numbers between parentheses indicate the number of unsolved instances within the time limit of 2 h.

From Table 7 we remark that $DOMP_{4\cap 1}(B\&B)$ performs similarly as $DOMP_2(B\&B)$ (it improves the behavior of $DOMP_2(B\&B)$ only for some instance sizes). It is better for data instances with $n<40$ in all combinations of p. In addition, for larger n, i.e. $n=40,\ 50$, it is also better except if p is relatively small as compared with n. Furthermore, we also observe that for $n=50$ both models fail to solve some instances for the smallest tested value of $p=12$. Finally, Table 8 allows us to conclude that three index with aggregated scheduling constraints performs the best whenever the number of facilities to be located is not too small compared to the number of possible locations.

4.3.2. Beasley's data set

The large number of ties within the cost matrices of this data set suggests that on this second part of the study $DOMP_{4C}$ is the appropriate formulation to solve the problems. For each cost matrix we solve each instance with $DOMP_2(B\&B)$ and $DOMP_{4C}(B\&C-3C)$, i.e. formulation $DOMP_{4C}$ within a branch and cut scheme separating inequalities (29). For each value of n we solve those problems for the number of open facilities p, suggested in the original data from Beasley's library, and for all the considered vectors of λ shown in Table 3.

Table 9
Summary of results using Beasley's data set.

Problem	n	p	Time (#unsolved)	
			$DOMP_2$(B&B)	$DOMP_{4C}$(B&C − 3C)
pmed1	50	5	94.65	94.66
pmed2	50	10	79.24	40.58
pmed3	50	10	103.84	37.58
pmed4	50	20	34.94	8.57
pmed5	50	33	20.44	3.90
pmed6	100	5	805.02	4757.27(5)
pmed7	100	10	617.60	2708.85(2)
pmed8	100	20	848.95(1)	543.26
pmed9	100	40	130.28	45.87
pmed10	100	67	45.84	22.01
pmed11	150	5	1353.85(1)	3702.40(2)
pmed12	150	10	1667.06(1)	4235.25(4)
pmed13	150	30	2030.82(1)	3808.45(3)
pmed14	150	60	509.60	302.93
pmed15	150	100	88.48	52.17
pmed16	200	5	2760.99(1)	5940.66(6)
pmed17	200	10	2940.22(1)	5670.05(6)
pmed18	200	40	1830.27(1)	4129.70(3)
pmed19	200	80	358.11	264.35
pmed20	200	133	256.21	150.76

Table 9 shows the average results for these instances. Detailed information for each λ can be found in the Appendix. This table is organized in five columns. The first three columns show the name of the instance problem and its size n and p. The last two columns show the CPU-time for solving those instances applying strategies $DOMP_2(B\&B)$ and $DOMP_{4C}(B\&C-3C)$. The numbers between parentheses indicate the number of instances that could not be solved to optimality within the time limit of 2 h. We can see that in 11 out of 20 instances $DOMP_{4C}(B\&C-3C)$ is faster than $DOMP_2(B\&B)$. This behavior confirms that both formulations have a rather similar performance. In spite of that, we observe that the use of our new formulation outperforms $DOMP_2$ provided that n is of moderate size $n<50$ or whenever the size of p relative to n is not too small, namely $p/n\geq 0.2$. This behavior allows us to conclude that $DOMP_{4C}(B\&C-3C)$ is advisable to be used, at least, in those cases.

5. Concluding remarks

This paper presents new formulations for the Discrete Ordered Median Problem based on order constraints (21) that are valid for the general non free self-service case. Furthermore, we prove theoretical relationships, in terms of their LP-gap, for different formulations of DOMP. According to the theoretical and computational results obtained in this paper, the main quality of the new formulations is that they provide substantial improvement of the integrality gap with respect to previously known ones.

We have observed that the LP-gap of $DOMP_1$ and $DOMP_2$ is always equal. It is currently an open question whether this property holds in general. This question will be a subject of our future research.

This paper has also opened another interesting line of research that consists in finding extensions of some of the existing formulations to exploit special structures of the lambda coefficients, as for instance the one in Marín et al. [11]. Extensions based on the results in [11] seem to require additional variables to handle the non free self-service case. A similar rationale can be also applied to the new formulations in this paper. Further theoretical and computational comparisons of the above mentioned new approach will be the subject of a follow up paper.

Acknowledgment

The research of the first author was partially supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office and Spanish MTM2012-36163-C06-04. The research of the second and third authors was partially supported by the project FQM-5849 (Junta de Andalucía\FEDER) and MTM2010-19576-C02-01 (MICINN, Spain).

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cor.2016.06.004.

References

[1] Arbib C, Labbé M, Servilio M. Scheduling two chains of unit jobs on one machine: a polyhedral study. Networks 2011;58(2):103–13.
[2] Beasley JE. OR-Library; 2012. people.brunel.ac.uk/mastjjb/jeb/info.html.
[3] Boland N, Domínguez-Marín P, Nickel S, Puerto J. Exact procedures for solving the discrete ordered median problem. Comput Oper Res 2006;33(11):3270–300 ISSN 0305-0548.
[4] Cornuéjols G, Thizy JM. Some facets of the simple plant location polytope. Math Program 1982(23):50–74.
[5] de Farias Jr IR. A family of facets for the uncapacitated p-median polytope. Oper Res Lett 2001;28(4):161–7.
[6] Domínguez-Marín P. The discrete ordered median problem: models and solution methods. New York: Springer; 2003.
[7] Fernández E, Puerto J, Rodríguez-Chía AM. On discrete optimization with ordering. Ann Oper Res 2013;207(1):83–96.
[8] Guignard M. Fractional vertices cuts and facets of the simple plant location problem. Math Program Study 1980(12):152–60.
[9] Kalcsics J, Nickel S, Puerto J, Rodríguez-Chía AM. Distribution systems design with role dependent objectives. Eur J Oper Res 2010;202:491–501.
[10] Marín A, Nickel S, Puerto J, Velten S. A flexible model and efficient solution strategies for discrete location problems. Discrete Appl Math 2009;157(5):1128–45 ISSN 0166-218X.
[11] Marín A, Nickel S, Velten S. An extended covering model for flexible discrete and equity location problems. Math Methods Oper Res 2010;71(1):125–63.
[12] Nemhauser GL, Trotter LE. Vertex packings: structural properties and algorithms. Math Program 1975;8:232–48.
[13] Nickel S. Discrete ordered weber problems. In: Operations research proceedings 2000; Berlin Heidelberg: Springer-Verlag; 2001. p. 71–6.
[14] Nickel S, Puerto J. Location theory: a unified approach. Berlin Heidelberg: Springer-Verlag; 2005.
[15] Padberg MW. On the facial structure of set packing polyhedra. Math Program 1973;5:199–215.
[16] Puerto J. A new formulation of the capacitated discrete ordered median problem with $\{0, 1\}-$ assignment. In: Operations research proceedings 2007, vol. 1; 2008. Springer. p. 165–70. ISBN 978-3-540-77902-5.
[17] Puerto J, Ramos AB, Rodríguez-Chía AM. Single-allocation ordered median hub location problems. Comput Oper Res 2011;38(2):559–70 ISSN 0305-0548.
[18] Puerto J, Ramos AB, Rodríguez-Chía AM. A specialized branch & bound & cut for single-allocation ordered median hub location problems. Discrete Appl Math 2013;161(16–17):2624–46.
[19] Vasilyev I, Klimentova X, Boccia M. Polyhedral study of simple plant location problem with order. Oper Res Lett 2013(2):153–8.