



HAL
open science

The double-assignment plant location problem with co-location

Alfredo Marín, Mercedes Pelegrín

► **To cite this version:**

Alfredo Marín, Mercedes Pelegrín. The double-assignment plant location problem with co-location. Computers and Operations Research, 2021, 126, pp.105059 -. 10.1016/j.cor.2020.105059 . hal-03493309

HAL Id: hal-03493309

<https://hal.science/hal-03493309>

Submitted on 17 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

The double-assignment plant location problem with co-location [☆]

Alfredo Marín^a, Mercedes Pelegrín^{b,*}

^a*Department of Statistics and Operational Research, University of Murcia, Spain*
^b*LIX, École Polytechnique, 91128 Palaiseau Cedex, France*

Abstract

In this paper, a new variant of the Simple Plant Location Problem is proposed. We consider additional conditions in the classic location-allocation problem for clients and facilities. Namely, some pairs have to be served by a common plant. The resulting problem can be addressed with existing models for the case of single assignment. However, to the best of our knowledge, the proposed setting when each client must be assigned to a couple of facilities is still unexplored. We examine the implications of adding such new constraints to standard formulations of the SPLP with double assignment. We compare the resulting formulations from a theoretical point of view. After that, we focus on the study of one of the models, which turned out to be a set packing problem. All the clique facets are identified and a separation algorithm is devised. Although the separation problem is proved to be NP-hard, our computational experience shows that the separation algorithm is effective and efficient, reducing computational times and duality gaps for all the instances tested.

Keywords: Discrete Location, Set Packing, Clique Facets, Separation Heuristic

2010 MSC: 90C10, 90B80, 90C90

[☆]Research supported by the *Ministerio de Economía y Competitividad*, project MTM2015-65915-R and *Ministerio de Educación, Cultura y Deporte*, PhD grant FPU15/05883.

*Corresponding author

Email addresses: amarin@um.es (Alfredo Marín),
pelegringarcia@lix.polytechnique.fr (Mercedes Pelegrín)

1. Introduction

Facility location has been one of the most fruitful areas within operations research (see [1, 2]). One of the reasons is that decision making certainly includes strategic infrastructure layout in almost every corporation, public or private. Industrial companies have to locate their facilities and warehouses so as to reach clients in their marketplaces; government agencies usually decide on the location of schools, hospitals, emergency services, etc. Other than its wide range of applications, the area also entails theoretical challenges, see for instance [3] or [4]. In this paper we study a variant of a well-known model in facility location, which, as far as we know, has not been investigated yet.

One of the seminal problems in discrete location is to decide on sites to install a set of identical facilities and on how clients are allocated to them. The problem, which has been known as Simple Plant Location Problem (SPLP) or Uncapacitated Facility Location Problem, has been extensively studied by operational researchers [3, 5, 6, 7, 8]. The simplicity of the model has been fundamental as a base for the development of locational analysis, and at the same time has given room for studying a range of variants. Extensions include those considering specialized facilities in different product types [9], customer preferences [10], risk of disruptions in the distribution system [11, 12] or models seeking for sustainable logistics [13]. For more references, see surveys [1] or [14], which focus on the SPLP. It is worth mentioning that [15] presented a Benders decomposition more recently, which found optimal solutions for previously unsolved instances of up to 3000 clients and location candidates and obtained speedups of several orders of magnitude with respect to other methods.

In this work, we propose a new modification of the SPLP, which considers new requirements in relation to the allocation of clients to facilities. The new variant can be considered to belong to the same family than that introduced in [20]. In that previous work, some pairs of clients were supposed to be *incompatible*, that is, they could not be assigned to the same facility. A set packing formulation for the resulting facility location problem was proposed and its poly-

hedral structure was studied, deriving different types of facets and separation algorithms to manage the inequalities within a branch and cut.

In our new scenario, some pairs of clients, which we will call *tied*, wish to be allocated to the same facility. This situation could be easily addressed by existing models for the SPLP, since allocation of each tied pair can be decided at once by adding their costs, just as if they were a single client. However, sometimes clients must be assigned to a couple of facilities, for instance when a backup service is needed, just as in [11]. When double assignment is considered, tied clients are pairs that like to be served by at least one common facility. Now, the new scenario gives rise to a new combinatorial problem that is a variant of the classic SPLP. Closely related topics are hub location, where origin and destination pairs have to be connected by using a couple of hubs (see for instance [16]) or warehouse location, where clients are served by a facility through a warehouse (see [17]). In facility location problems with capacities, clients are also frequently assigned to more than one facility, see for instance [18].

The above setting, which we will call double-assignment plant location with co-location, finds interesting applications in telecommunication networks design. A generic telecommunication network consists of a set of terminals (users), connected to concentrators (switches or multiplexers) and a backbone network which interconnects the concentrators. A primary problem in network design is to decide how many concentrators are needed and how the terminals should be assigned to the concentrators. These two decisions can be identified with that of facility location and allocation, a fact that was already observed in [19]. In the context of telecommunication networks, the setting we propose corresponds to a configuration in which some users must share a concentrator. This is a realistic assumption, since there could be users with special communication requirements that want to have a dedicated path to avoid the backbone network.

The contributions of the paper can be summarized as follows: (i) a new variant of the SPLP, which considers double assignment and clients ties is proposed; (ii) two formulations, inspired by classic two and three index facility location models, are presented, and their linear relaxations are theoretically compared

(something that to the best of our knowledge was not done before); (iii) all the clique facets of one of the formulations, which is a set packing, are disclosed; (iv) the corresponding separation problem is proved to be NP-hard and a heuristic algorithm is then proposed; (v) finally, a computational study to test the developed formulations and algorithm is conducted.

The paper is organized as follows. The next section introduces the double-assignment plant location problem with co-location, together with two integer programming formulations. In the following section, the linear relaxations of both are compared theoretically. Then, in Section 4 all the clique facets of one of the new models, which is a set packing problem, are described. The problem of separating them is proven to be NP-hard in Section 5, where a heuristic separation algorithm is also proposed. Finally, Section 6 reports the computational tests of the formulations and the heuristic, which includes a comparative analysis with respect to standard clique cuts incorporated by a commercial solver. Some conclusions close the paper.

2. Problem statement and formulations

Consider as initial setting that of the SPLP, where $I := \{1, \dots, n\}$ and $J := \{1, \dots, m\}$ are the sets of clients and candidate facilities respectively. Facility location and clients allocation decisions have been typically modeled with the following decision variables,

$y_j = 1$ iff no service is installed at candidate location j , $j \in J$,

$y'_j = 1$ iff a facility is installed at candidate location j , $j \in J$ and

$x_{ij} = 1$ iff client i is served by facility at j , $i \in I$, $j \in J$,

where y_j and y'_j satisfy $y_j = 1 - y'_j$ for all j . The classic formulation of the

SPLP is

$$\begin{aligned}
(\text{SPLP}) \quad \min \quad & \sum_{j \in J} f_j y'_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} = 1 & \forall i \in I & (1) \\
& x_{ij} \leq y'_j & \forall i \in I, \forall j \in J & (2) \\
& x_{ij}, y'_j \in \{0, 1\} & \forall i \in I, \forall j \in J, &
\end{aligned}$$

where f_j and c_{ij} stand for opening and allocation costs respectively. Constraints (1) guarantee that every client in I is assigned to one location candidate in J and (2) ensure that there is a facility at the given site. Many works consider
90 an alternative formulation of the SPLP, which replaces location variables y'_j by their complementary binary variables y_j . This simple idea allows to rewrite (SPLP) into the following set packing formulation, where M is a large enough constant (see [3] for more details),

$$\begin{aligned}
(\text{SPLP}_{\leq}) \quad \max \quad & \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} (M - c_{ij}) x_{ij} - Mn - \sum_{j \in J} f_j \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} \leq 1 & \forall i \in I & (3) \\
& x_{ij} + y_j \leq 1 & \forall i \in I, \forall j \in J \\
& x_{ij}, y_j \in \{0, 1\} & \forall i \in I, \forall j \in J.
\end{aligned}$$

When double assignment is considered, the problem is usually modeled with
95 two alternative approaches. One is based on the previous formulations, and consists of using the same variables and changing 1 on the right-hand side of (1) or (3) by 2. The second is to consider new allocation three-indexed variables to represent the pair of facilities that serve each client.

The double-assignment plant location problem with co-location, DPLP from
100 now on, consists of allocating each client to two facilities in such a way that tied clients share at least one. The following subsections present two alternative formulations of the DPLP, which are adapted from the aforementioned strategies to model double assignment. To model co-location relationships, we consider a graph $G_T = (I, E_T)$ that has one node per client and an edge $e = (i, i') \in E_T$

105 for every pair (i, i') of tied clients. Note that one client can be tied more than once and that co-location is not transitive but reciprocal.

Two-indexed formulation

Consider standard allocation variables and costs, x_{ij} and c_{ij} , together with location variables y_j and opening costs f_j . The DPLP₂ can be formulated by
 110 modifying (SPLP), as follows

$$\begin{aligned}
 \text{(DPLP}_2\text{)} \quad \min \quad & \sum_{j \in J} f_j(1 - y_j) + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{j \in J} x_{ij} = 2 \quad \forall i \in I \quad (4) \\
 & x_{ij} + y_j \leq 1 \quad \forall i \in I, \forall j \in J \quad (5) \\
 & x_{i'j} + x_{i'j'} \geq x_{ij} + x_{ij'} - 1 \quad \forall (i, i') \in E_T, \forall j, j' \in J : j < j' \quad (6) \\
 & x_{ij}, y_j \in \{0, 1\} \quad \forall i \in I, \forall j \in J.
 \end{aligned}$$

Constraints (4) stand for double assignment, while (5) guarantee that clients are only allocated to open facilities. Constraints (6) are the co-location constraints. They are only active when their right hand sides equal one, that is, when a client i is allocated to facilities j and j' ($x_{ij} = x_{ij'} = 1$). In this case, the constraints
 115 ensure that i' is allocated to at least one of these two facilities, j or j' , for every tied client, $(i, i') \in E_T$. In (DPLP₂), subscript 2 indicates the use of allocation variables with two indices, as opposed to the following alternative formulation.

Three-indexed formulation

Alternatively to standard allocation variables, one can use

120 $z_{ijk} = 1$ iff client i is served by facilities j and k ,

for all $i \in I, j, k \in J$ such that $j < k$. For notation simplicity, in the following, j, k, ℓ and t are indices in J if not stated otherwise. With the above variables and location y -variables, the problem can be formulated as a set packing problem in

the following way

$$\begin{aligned}
(\text{DPLP}_3) \quad & \min \quad \sum_{j \in J} f_j(1 - y_j) + \sum_{i \in I} \sum_{j \in J} \sum_{\substack{k \in J: \\ k > j}} (c_{ij} + c_{ik} - M) z_{ijk} + Mn \\
\text{s.t.} \quad & \sum_{j \in J} \sum_{k > j} z_{ijk} \leq 1 && \forall i \in I && (7) \\
& \sum_{k > j} z_{ijk} + \sum_{k < j} z_{ikj} + y_j \leq 1 && \forall i \in I, \forall j \in J && (8) \\
& z_{ijk} + \sum_{\ell \in J} \sum_{\substack{t > \ell: \\ \{j, k\} \cap \{\ell, t\} = \emptyset}} z_{i'\ell t} \leq 1 && \forall (i, i') \in E_T, j < k && (9) \\
& y_j \in \{0, 1\} && \forall j \in J, \\
& z_{ijk} \in \{0, 1\} && \forall i \in I, \forall j, k \in J : j < k
\end{aligned}$$

125 where M is a large enough constant. Constraints (7) and (8) would correspond with (1) and (2) of the SPLP. Co-location constraints (9) ensure that tied clients $(i, i') \in E_T$ are not allocated to non-overlapping pairs of facilities. Indeed, the constraints ensure that z_{ijk} and $z_{i'\ell t}$ cannot both take value one if $\{j, k\} \cap \{\ell, t\} = \emptyset$. Instead of (9), simply imposing $z_{ijk} + z_{i'\ell t}$ for all
130 $(i, i') \in E_T$ and different facility subscripts j, k, ℓ, t with $j < k$ and $\ell < t$ would have sufficed. Inequalities (9) are just one way of reinforce that simplest translation of co-location requirements into packing constraints. Finally, note that $\sum_{j \in J} \sum_{k > j} z_{ijk} = 1$ for an optimal solution, because otherwise $M \gg 0$ is added to the objective value.

135 In the following section, we explore the relation between x -variables and z -variables. Using that relation, we compare the constraints of formulations (DPLP₂) and (DPLP₃) and the optimal solutions of their linear relaxations, ultimately concluding which formulation gives better bounds.

3. Comparing the formulations

Variables x and z are clearly related to each other. Given a client i , z_{ijk} will be one if and only if x_{ij} and x_{ik} are. This is mathematically written as follows

$$\begin{aligned} z_{ijk} &= x_{ij}x_{ik} & \forall i \in I, j, k \in J, j < k \\ x_{ij} &= \sum_{k>j} z_{ijk} + \sum_{k<j} z_{ikj} & \forall i \in I, j \in J. \end{aligned}$$

140 We use the second formula, which is linear, to replace x in (DPLP₂) by the corresponding z -variables. The resulting formulation, which we name (DPLP'₃), will be written in terms of decision variables y and z . We will use (DPLP'₃) to compare the objective values of the linear relaxations of (DPLP₂) and (DPLP₃).

Starting with (4), we get

$$\sum_{j \in J} x_{ij} = \sum_{j \in J} \left(\sum_{k>j} z_{ijk} + \sum_{k<j} z_{ikj} \right) = \sum_{j \in J} \sum_{k>j} z_{ijk} + \sum_{j \in J} \sum_{k<j} z_{ikj}.$$

145 Constraints (4) can be written then as $\sum_{j \in J} \sum_{k>j} z_{ijk} + \sum_{j \in J} \sum_{k<j} z_{ikj} = 2$, for all $i \in I$. Second, observe that (5) is (8) by substitution of x_{ij} as a function of z . We last substitute in (6),

$$\begin{aligned} x_{i'j} + x_{i'j'} &\geq x_{ij} + x_{ij'} - 1 \quad \equiv \\ \sum_{k>j} z_{i'jk} + \sum_{k<j} z_{i'kj} + \sum_{k>j'} z_{i'j'k} + \sum_{k<j'} z_{i'kj'} &\geq \sum_{k>j} z_{ijk} + \sum_{k<j} z_{ikj} + \sum_{k>j'} z_{ij'k} + \sum_{k<j'} z_{ikj'} - 1 \equiv \\ \sum_{k>j} z_{ijk} + \sum_{k<j} z_{ikj} + \sum_{k>j'} z_{ij'k} + \sum_{k<j'} z_{ikj'} - \sum_{k>j} z_{i'jk} - \sum_{k<j} z_{i'kj} - \sum_{k>j'} z_{i'j'k} - \sum_{k<j'} z_{i'kj'} &\leq 1. \quad (10) \end{aligned}$$

In order to compare (10) with (9), we unfold the summation in the latter. For
150 every $(i, i') \in E_T$ and $j, k \in J$ such that $j < k$,

$$\sum_{\ell} \sum_{\substack{t>\ell: \\ \{j,k\} \cap \{\ell,t\} = \emptyset}} z_{i'\ell t} = 1 - \left(\sum_{j'>j} z_{i'jj'} + \sum_{j'<j} z_{i'j'j} + \sum_{j'>k} z_{i'kj'} + \sum_{\substack{j'<k: \\ j' \neq j}} z_{i'j'k} \right). \quad (11)$$

Let

$$Z_{i'jk} := \sum_{j'>j} z_{i'jj'} + \sum_{j'<j} z_{i'j'j} + \sum_{j'>k} z_{i'kj'} + \sum_{\substack{j'<k: \\ j' \neq j}} z_{i'j'k},$$

which will be one if i' is allocated to j , k or both and zero otherwise. Constraints (10) are then,

$$Z_{ijk} + z_{ijk} - Z_{i'jk} - z_{i'jk} \leq 1$$

while (9) can be rewritten as

$$z_{ijk} + 1 - Z_{i'jk} \leq 1. \quad (12)$$

We eventually obtain the following constraints and objective value for (DPLP₃):

$$\begin{aligned}
(\text{DPLP}'_3) \quad \min \quad & \sum_{j \in J} f_j(1 - y_j) + \sum_{i \in I} \sum_{j \in J} c_{ij} \left(\sum_{k > j} z_{ijk} + \sum_{k < j} z_{ikj} \right) \\
\text{s.t.} \quad & \sum_{j \in J} \sum_{k > j} z_{ijk} + \sum_{j \in J} \sum_{k < j} z_{ikj} = 2 & \forall i \in I & (13) \\
& \sum_{k > j} z_{ijk} + \sum_{k < j} z_{ikj} + y_j \leq 1 & \forall i \in I, \forall j \in J \\
& Z_{ijk} + z_{ijk} - Z_{i'jk} - z_{i'jk} \leq 1 & \forall (i, i') \in E_T, j < k & (14) \\
& z_{ijk}, y_j \in \{0, 1\} & \forall i \in I, \forall j, k \in J.
\end{aligned}$$

155 Having tight bounds is critical for a branch and cut solving approach. The following proposition shows that the LP bound obtained with (DPLP₃) is at least as good as that of (DPLP₂).

Proposition 1. *The optimal value of the linear relaxation of (DPLP₂) is less than or equal to that of (DPLP₃).*

Proof. Suppose that (\bar{z}, \bar{y}) is an optimal solution of the linear relaxation of (DPLP₃) and let \bar{f}_3 be its objective value,

$$\bar{f}_3 := \sum_{j \in J} f_j(1 - \bar{y}_j) + \sum_{i \in I} \sum_{j \in J} \sum_{\substack{k \in J: \\ k > j}} (c_{ij} + c_{ik} - M) \bar{z}_{ijk} + Mn.$$

We define

$$\bar{x}_{ij} = \sum_{k > j} \bar{z}_{ijk} + \sum_{k < j} \bar{z}_{ikj} \quad \forall i \in I, j \in J.$$

160 We will show that (\bar{x}, \bar{y}) is a feasible fractional solution of (DPLP₂) with objective value \bar{f}_3 . Since (DPLP₂) is a minimization problem, this will prove the proposition.

Given that (\bar{z}, \bar{y}) is optimal, we know that $\sum_j \sum_{k>j} \bar{z}_{ijk} = 1$ for all i . Then, (\bar{z}, \bar{y}) satisfies (13). Moreover, since (12) is stronger than (14), (\bar{z}, \bar{y}) is a feasible
 165 solution of the linear relaxation of (DPLP'₃). Since (DPLP'₃) is formulation (DPLP₂) when $x_{ij} = \sum_{k>j} z_{ijk} + \sum_{k<j} z_{ikj}$, we conclude that (\bar{x}, \bar{y}) is a feasible solution of the linear relaxation of (DPLP₂).

The objective value of (\bar{x}, \bar{y}) is

$$\begin{aligned} \bar{f}_2 &:= \sum_{j \in J} f_j(1 - \bar{y}_j) + \sum_{i \in I} \sum_{j \in J} c_{ij} \bar{x}_{ij} = \sum_{j \in J} f_j(1 - \bar{y}_j) + \sum_{i \in I} \sum_{j \in J} c_{ij} \left(\sum_{k>j} \bar{z}_{ijk} + \sum_{k<j} \bar{z}_{ikj} \right) \\ &= \sum_{j \in J} f_j(1 - \bar{y}_j) + \sum_{i \in I} \sum_{j \in J} \sum_{k>j} c_{ij} \bar{z}_{ijk} + \sum_{i \in I} \sum_{j \in J} \sum_{k<j} c_{ij} \bar{z}_{ikj}. \end{aligned}$$

On the other hand, the fact that $\sum_j \sum_{k>j} \bar{z}_{ijk} = 1$ for all $i \in I$ implies

$$\bar{f}_3 = \sum_{j \in J} f_j(1 - \bar{y}_j) + \sum_{i \in I} \sum_{j \in J} \sum_{\substack{k \in J: \\ k>j}} (c_{ij} + c_{ik}) \bar{z}_{ijk}.$$

After rearranging the summations, we get $\bar{f}_2 = \bar{f}_3$. □

Proposition 1 does not ensure that the LP bounds of (DPLP₂) and (DPLP₃)
 170 do not always coincide. The following example illustrates that the LP bound of (DPLP₃) can be strictly greater than that of (DPLP₂).

Example 1. Consider an instance with $I = J = \{1, \dots, 6\}$, opening costs $f_j = 20$ for all $j \in J$ and allocation costs given by the following matrix

$$c = \begin{pmatrix} 0 & 30 & 76 & 25 & 18 & 48 \\ 30 & 0 & 45 & 50 & 14 & 38 \\ 76 & 45 & 0 & 37 & 28 & 53 \\ 25 & 50 & 37 & 0 & 60 & 15 \\ 18 & 14 & 28 & 60 & 0 & 27 \\ 48 & 38 & 53 & 15 & 27 & 0 \end{pmatrix}.$$

Figure 1 shows G_T for this example. The optimal solution of this instance is 272, with three facilities opened.

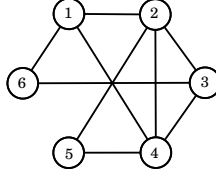


Figure 1: Graph G_T of Example 1

The optimal solution of the linear relaxation of $(DPLP_2)$ is

$$\begin{aligned}
x_{11} &= 0.5 & x_{14} &= 1 & x_{15} &= 0.5 \\
x_{21} &= 0.5 & x_{22} &= 0.5 & x_{25} &= 0.5 & x_{26} &= 0.5 \\
x_{33} &= 1 & x_{34} &= 1 \\
x_{44} &= 1 & x_{46} &= 1 \\
x_{52} &= 0.5 & x_{55} &= 0.5 & x_{56} &= 1 \\
x_{64} &= 1 & x_{66} &= 1 \\
y_1 &= 0.5 & y_2 &= 0.5 & y_3 &= 1 & y_4 &= 1 & y_5 &= 0.5 & y_6 &= 1,
\end{aligned}$$

with optimal value 266.

The linear relaxation $(DPLP_3)$ has optimal solution

$$\begin{aligned}
z_{115} &= 0.6 & z_{145} &= 0.4 \\
z_{215} &= 0.6 & z_{225} &= 0.4 \\
z_{334} &= 0.4 & z_{335} &= 0.6 \\
z_{414} &= 0.2 & z_{436} &= 0.2 & z_{445} &= 0.4 & z_{446} &= 0.2 \\
z_{515} &= 0.4 & z_{525} &= 0.4 & z_{556} &= 0.2 \\
z_{645} &= 0.4 & z_{646} &= 0.4 & z_{656} &= 0.2 \\
y_1 &= 0.6 & y_2 &= 0.4 & y_3 &= 1 & y_4 &= 0.8 & y_5 &= 1 & y_6 &= 0.6,
\end{aligned}$$

175 with optimal value 268.4.

4. Clique facets

In this section, we identify some facets of the integer polytope of $(DPLP_3)$,

$$\mathcal{B}_{dplp} := \text{conv}\{(z, y) \in \{0, 1\}^{n \cdot \frac{m(m-1)}{2}} \times \{0, 1\}^m : (7) - (9)\}.$$

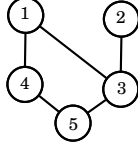


Figure 2: A graph G_T of five tied clients, Example 2

In order to study the facial arrangement of \mathcal{B}_{dplp} , we will leverage the set packing structure of (DPLP₃). Any set packing formulation, i.e., any linear program with a 0/1 constraint matrix and a vector of ones as right-hand side, can be uniquely identified with a graph. It is usually called conflict graph or intersection graph, and has one node per variable and edges between nodes of variables that appear in the same constraint. A feasible solution of the set packing corresponds with a subset of non-adjacent nodes (packing, stable set, independent set) of the conflict graph. It is standard to identify facets of a set packing with some structures in its conflict graph, see for instance [8, 20]. In this section, we will identify the cliques of G_{dplp} , the conflict graph of (DPLP₃), which are known to define facets [21]. Note that, throughout the paper, cliques will be maximal complete subgraphs.

Before starting with the polyhedral analysis of (DPLP₃), the following example illustrates the structure of G_{dplp} for a small instance of the problem. Given the density of the conflict graph, each family of constraints in (DPLP₃), namely (7), (8) and (9), is illustrated separately for a restricted number of nodes.

Example 2. *We consider an instance with five clients and four potential facility locations. The graph describing the tied pairs, G_T , is shown on Figure 2. Figures 3-5 illustrate constraints of (DPLP₃) on G_{dplp} . Circular nodes correspond with z -variables, and are tagged with proper subscripts ijk . Square nodes stand for y -variables. For a clear illustration, only a sample of the edges are shown. Circular nodes are arranged in groups forming a matrix, each group corresponding to a*

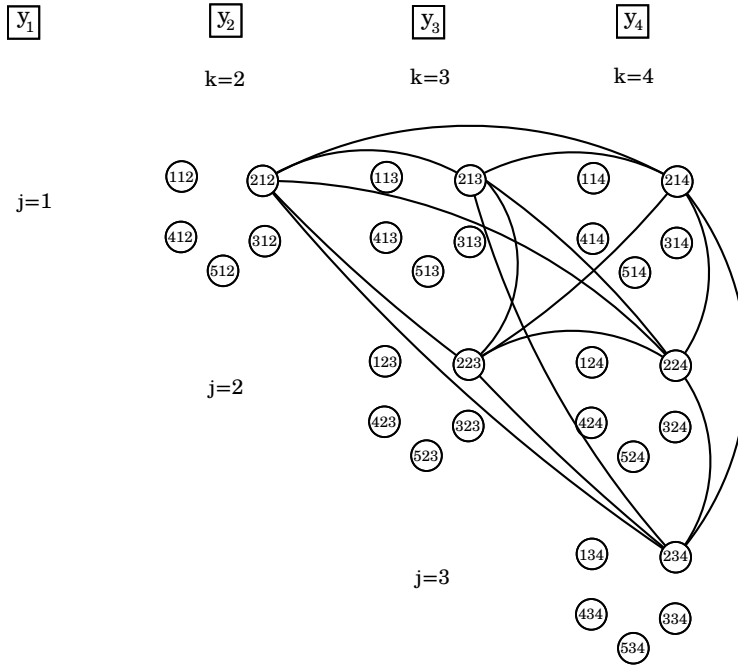


Figure 3: Edges in G_{dplp} corresponding with (7) and $i = 2$

pair of facilities j, k with $j < k$ and having five nodes, one for each of the five
 200 clients.

Figure 3 depicts how edges link nodes that refer to the same client and corresponds to constraints (7) of $(DPLP_3)$. Note that they define cliques. In fact, Figure 3 shows a clique of six nodes associated to client 2.

Figure 4 illustrates that nodes corresponding with the same client and facility
 205 are adjacent to the node of that facility. Such subsets of nodes also define cliques in G_{dplp} . The edges shown in this figure are due to constraints (8) in the model.

Finally, Figure 5 shows that the groups of nodes displayed in each entry (j, k) are also interconnected by means of the co-location constraints (9).

A first observation is that (9) can be generalized. Take the example depicted in Figure 5 and clients $i = 3$ and $i' = 5$, $(i, i') \in E_T$. If we add two new plants with indices 5 and 6, we will have co-location constraints

$$z_{312} + z_{534} + z_{535} + z_{536} + z_{545} + z_{546} + z_{556} \leq 1,$$

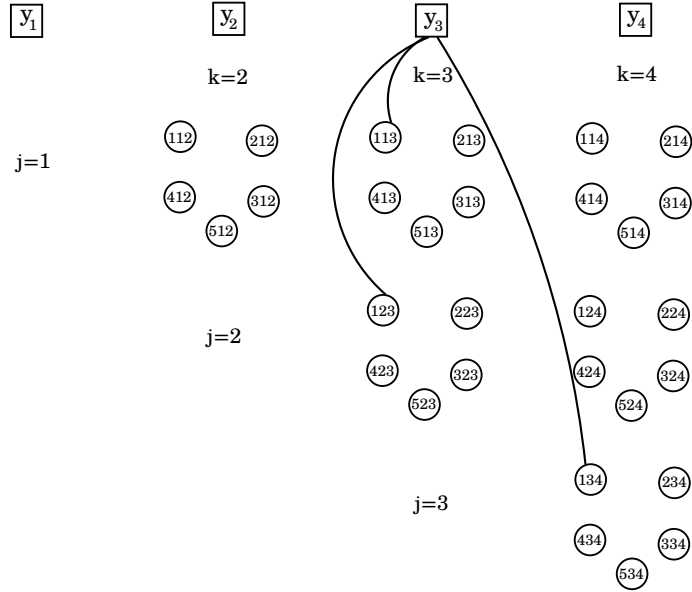


Figure 4: Edges in G_{dplp} corresponding with (8) when $i = 1$ and $j = 3$

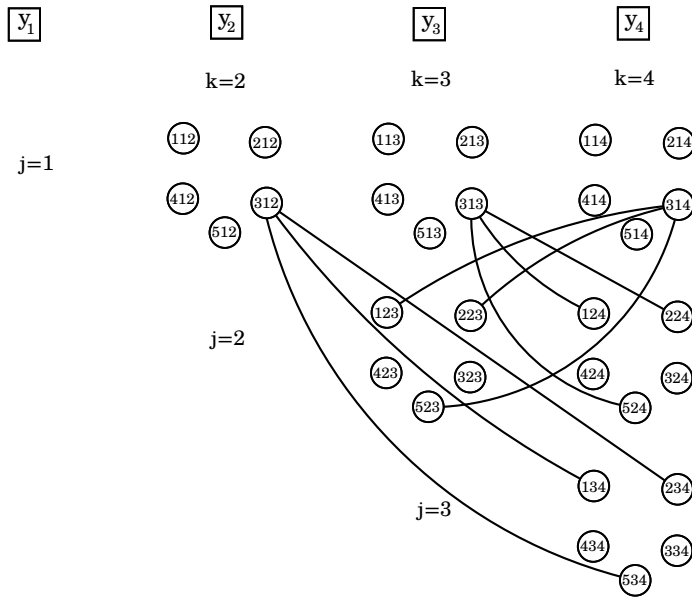


Figure 5: Edges in G_{dplp} corresponding with (9), graph in Figure 2 and $i = 3$, $i' = 1, 2, 5$, $j = 1$, $k = 2, 3, 4$

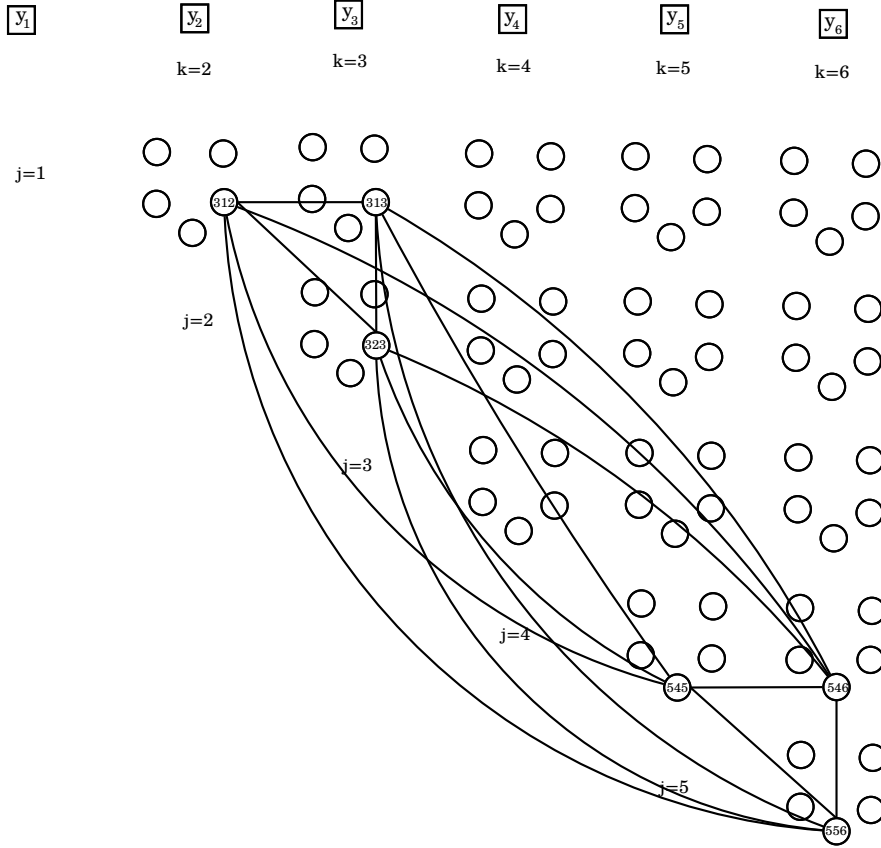


Figure 6: Clique in $G_{dplp}(15)$ with $(3, 5) \in E_T$, $J_1 = \{1, 2, 3\}$, $J_2 = \{4, 5, 6\}$

$$z_{313} + z_{524} + z_{525} + z_{526} + z_{545} + z_{546} + z_{556} \leq 1$$

for $(j, k) = (1, 2)$ and $(j, k) = (1, 3)$, respectively. However,

$$z_{312} + z_{313} + z_{323} + z_{545} + z_{546} + z_{556} \leq 1,$$

which is illustrated by Figure 6, is also valid and not written in the formulation.

210 This corresponds to forbidding the pairs of facilities that serve tied clients 3 and 5 to be in two non-overlapping subsets of plants $J_1, J_2 \subseteq J$, $J_1 \cap J_2 = \emptyset$. In the example above, $J_1 = \{1, 2, 3\}$ and $J_2 = \{4, 5, 6\}$, while (9) stand for the particular case of $J_1 = \{j, \ell\}$ and $J_2 = J \setminus \{j, \ell\}$.

Proposition 2. Let $(i_1, i_2) \in E_T$ and $J_1, J_2 \subset J$ both containing at least two

215 different facilities and such that $J_1 \cap J_2 = \emptyset$. The following inequalities are valid for (DPLP₃)

$$\sum_{j_1 \in J_1} \sum_{\substack{k_1 \in J_1: \\ k_1 > j_1}} z_{i_1 j_1 k_1} + \sum_{j_2 \in J_2} \sum_{\substack{k_2 \in J_2: \\ k_2 > j_2}} z_{i_2 j_2 k_2} \leq 1. \quad (15)$$

Moreover, (15) are facets of \mathcal{B}_{dplp} if and only if $J_1 \cup J_2 = J$. In particular, (9) are facets.

Proof. The fact that (15) are valid is clear from the problem definition. Indeed, 220 using (9) and the hypothesis $J_1 \cap J_2 = \emptyset$, we know that

$$z_{i_1 j_1 k_1} + \sum_{j_2 \in J_2} \sum_{\substack{k_2 \in J_2: \\ k_2 > j_2}} z_{i_2 j_2 k_2} \leq 1$$

holds for every $j_1, k_1 \in J_1$, $k_1 > j_1$. On the other hand, due to (7), a double summation on the two subscripts running in J_1 can be included in the previous inequality, yielding (15).

For the second statement of the proposition, we will prove that

$$C := \left(\bigcup_{j_1 \in J_1} \bigcup_{\substack{k_1 \in J_1: \\ j_1 > k_1}} \{z_{i_1 j_1 k_1}\} \right) \cup \left(\bigcup_{j_2 \in J_2} \bigcup_{\substack{k_2 \in J_2: \\ j_2 > k_2}} \{z_{i_2 j_2 k_2}\} \right)$$

define a clique in G_{dplp} if and only if $J_1 \cup J_2 = J$.

225 Suppose first that $G_{dplp}[C]$ is a clique and there is $j' \in J \setminus \{J_1 \cup J_2\}$. In this case

$$\sum_{\substack{k_1 \in J_1: \\ j' > k_1}} z_{i_1 j' k_1} + \sum_{\substack{k_1 \in J_1: \\ j' < k_1}} z_{i_1 k_1 j'} + \sum_{j_1 \in J_1} \sum_{\substack{k_1 \in J_1: \\ j_1 > k_1}} z_{i_1 j_1 k_1} + \sum_{j_2 \in J_2} \sum_{\substack{k_2 \in J_2: \\ j_2 > k_2}} z_{i_2 j_2 k_2} \leq 1 \quad (16)$$

would be valid and stronger than (15), which is a contradiction.

Conversely, suppose that $J_1 \cup J_2 = J$. From the problem definition, we know that $G_{dplp}[C]$ is a complete subgraph. Then, we have to prove that $G_{dplp}[C]$ 230 is a maximal complete subgraph. If there is $v \notin C$ such that $G_{dplp}[C \cup \{v\}]$ is complete, then v does not correspond with a y -variable. This is clear, since a node y_j is adjacent only to z_{ijk} and z_{ikj} for all i and k . Since $J_1 \cap J_2 = \emptyset$, there cannot be a y -node adjacent to every node in C . Suppose then that v is

a z -node, i.e., $G_{dplp}[C \cup \{z_{i'j'k'}\}]$ is a complete subgraph, for some i', j', k' . We
 235 distinguish between two different cases.

1. If $i' = i_1$ (resp. $i' = i_2$), one of the facilities j' or k' is not in J_1 (resp. J_2), otherwise the node was already in C . Without loss of generality, let j' be that facility. Since $J_1 \cup J_2 = J$, then $j' \in J_2$ (resp. J_1). But this is a contradiction because $z_{i_1j'k'}$ and $z_{i_2j'k_2} \in C$ are not adjacent (resp. $z_{i_2j'k'}$ and $z_{i_1j'k_1} \in C$), for any $k_2 \in J_2, k_2 > j'$ (resp. $k_1 \in J_1, k_1 > j'$).
 240
2. If $i' \neq i_1$ and $i' \neq i_2$, edges between $z_{i'j'k'}$ and nodes in C will all correspond with co-location constraints. In particular, $\{i', i_1, i_2\}$ define a complete subgraph in G_T . Again, due to the fact that $J_1 \cup J_2 = J$, there cannot be such co-location constraints between $z_{i'j'k'}$ and $z_{i_1j_1k_1}$ if j' or k' are in J_1 , or between $z_{i'j'k'}$ and $z_{i_2j_2k_2}$ if j' or k' are in J_2 .
 245

□

Following a similar idea, more valid inequalities can be derived for every triangle $\{i, i', i''\} \subseteq I$ in G_T and partition $\{J_1, J_2, J_3\}$ of J with J_i consisting of at least two facilities each. In general, this can be done for every subset of
 250 pairwise tied clients and every partition having the same size. The following proposition states that all the clique facets of \mathcal{B}_{dplp} can be obtained in this way or are of the form (7) or (8).

Theorem 1. *The only clique facets of \mathcal{B}_{dplp} are (7), (8) and*

$$\sum_{i \in C_T} \sum_{j \in J_i} \sum_{\substack{k \in J_i: \\ k > j}} z_{ijk} \leq 1, \quad (17)$$

for all $C_T \subseteq I$ such that $G_T[C_T]$ is a complete subgraph, and for all $J_i \subseteq J$,
 255 $i \in C_T$, such that $|J_i| \geq 2$, $\cup_{i \in C_T} J_i = J$ and $J_i \cap J_{i'} = \emptyset$ for every pair $i \neq i'$.

Proof. Suppose that $\pi z + \rho y \leq 1$ is a clique facet of \mathcal{B}_{dplp} . We will show that it has to be one of the inequalities of the statement.

We analyze first the case in which $\rho_{j'} = 1$ for some $j' \in J$, which readily implies $\rho_j = 0$ for all $j \neq j'$. Variables with positive π -coefficients have to be

260 in the neighborhood of $y_{j'}$ in G_{dplp} , $N(y_{j'})$. But $N(y_{j'}) = \{z_{ij'k} : i \in I, k > j'\} \cup \{z_{ikj'} : i \in I, k < j'\}$. In particular, z -variables in $N(y_{j'})$ all have one of its plant indices equal to j' . As a consequence, these variables are never adjacent due to a co-location constraint. Due to the problem constraints, they will be adjacent if and only if they have the same client index. In other words, there is a clique containing $y_{j'}$ for each $i \in I$, namely $C_{ij} = \{y_{j'}\} \cup \{z_{ij'k}\}_{j' < k} \cup \{z_{ikj'}\}_{k < j'}$,
 265 which corresponds to (8).

Suppose now that $\rho_j = 0$ for all $j \in J$. Let

$$C := \{z_{ijk} : \pi_{ijk} = 1, i \in I, j, k \in J, j < k\}$$

be the nodes of G_{dplp} corresponding with the clique facet $\pi z \leq 1$. Let

$$C_T := \{i \in I : z_{ijk} \in C \text{ for some } j, k\}$$

be the subset of clients that appear in the indices of the variables of C . We analyze two cases.

1. $C_T = \{i'\}$. In this case, $C \subseteq \{z_{i'jk} : j, k \in J, j < k\}$. Due to constraints
 270 (7), it has to be $C = \{z_{i'jk} : j, k \in J, j < k\}$. In fact, clique facet $\pi z \leq 1$ corresponds precisely to (7).
2. $|C_T| > 1$. In this case, edges of $G_{dplp}[C]$ between nodes associated to different clients have to be due to (9), which are defined for every $(i, i') \in E_T$. As a consequence, $G_T[C_T]$ is also a complete subgraph. For every
 275 $i \in C_T$, let $J_i^1 := \{j \in J : z_{ijk} \in C \text{ for some } k\}$ and $J_i^2 := \{k \in J : z_{ijk} \in C \text{ for some } j\}$. We also define $J_i := J_i^1 \cup J_i^2$, the set of facilities indices that appear with client i in the variables of C . First, $J_i \cap J_{i'} = \emptyset$ for all $i, i' \in C_T$ since $\pi z \leq 1$ will not be valid otherwise. On the other hand, for all $i \in C_T$, $\{(j, k) \in J \times J : z_{ijk} \in C\} \subseteq \{(j, k) \in J_i \times J_i : j < k\}$. If,
 280 for some $i' \in C_T$, there was (j', k') contained in the second subset but not in the first, $\pi z \leq 1$ could be improved by adding $z_{i'j'k'}$ to its left-hand side, which contradicts the fact that it is a facet. Hence, $\{(j, k) \in J \times J : z_{ijk} \in C\} = \{(j, k) \in J_i \times J_i : j < k\}$. It follows that $\pi z \leq 1$ has the form

of (17). It remains to prove that $\cup_{i \in C_T} J_i = J$. But now it is easy to see
 285 that $\pi z \leq 1$ could be improved if there was $j' \in J \setminus (\cup_{i \in C_T} J_i)$, just by
 adding j' to one of the subsets J_i .

□

Corollary 1. *All the constraints of (DPLP₃) are facets.*

Remark 1. *In Theorem 1, $G_T[C_T]$ is a complete graph but C_T is not necessarily
 290 a clique. Indeed, (15) are a particular case of (17) when $C_T = \{i_1, i_2\}$ for
 $(i_1, i_2) \in E_T$, which is not necessarily a maximal complete subgraph. This is
 interesting because, to the best of our knowledge, there is no precedence in set
 packing of a clique facet ultimately induced by a not maximal complete subgraph.*

5. Clique separation

295 The only clique facets of \mathcal{B}_{dplp} that are not included in (DPLP₃) are that of
 family (17) with $|C_T| > 2$ or $|C_T| = 2$ and $|J_i| \geq 2$, $i \in C_T$. This section focuses
 on the latter, which belong to family (15). Given a tied pair $(i_1, i_2) \in E_T$, (15)
 define exponentially many inequalities, even when $J_1 \cup J_2 = J$. A separation
 algorithm is then needed to manage these inequalities within a branch and cut
 300 scheme. We will first prove the theoretical computational complexity of this
 problem and then provide a heuristic algorithm to approach it. These results
 can be extended to separation of the more general family (17). On the one hand,
 the theoretical complexity does not improve when the size of the partition (which
 coincides with $|C_T|$) increases. On the other hand, we will see that the proposed
 305 heuristic can be easily adapted. However, exhaustive separation of (17) is not
 expected to be rewarding. We will prefer to address the case $|C_T| = 2$ when
 moving separation from theory to practice (see the computational study of next
 section).

Given a fractional optimal solution \bar{z} , and given an edge $(i_1, i_2) \in E_T$, the
problem of separation of (15) is to find the partition of J in J_1 and J_2 that

maximizes

$$\sum_{j_1 \in J_1} \sum_{\substack{k_1 \in J_1: \\ j_1 > k_1}} \bar{z}_{i_1 j_1 k_1} + \sum_{j_2 \in J_2} \sum_{\substack{k_2 \in J_2: \\ j_2 > k_2}} \bar{z}_{i_2 j_2 k_2}. \quad (18)$$

This problem can be identified with a max-cut problem in a graph. Given a graph $G = (V, E)$ with weights on its edges, the max-cut is to determine a subset $U \subseteq V$ such that the sum of the weights in the cut—the set of edges with one endnode in U and the other in $V \setminus U$ —is maximal. We identify the separation with the following especial case of max-cut, which we name *double-cut*.

Definition 1 (DOUBLE-CUT). Let $G = (V, E)$ be a graph and let W^1 and W^2 be two matrices of weights for its edges, $W^1 = (w_{jk}^1)_{i,k \in V}$, $W^2 = (w_{jk}^2)_{i,k \in V}$, $w_{jk}^1, w_{jk}^2 \in \mathbb{R}$. Given $W \in \mathbb{R}$, *DOUBLE-CUT* is to decide whether there is $V_1 \subseteq V$ such that $\sum_{j \in V_1} \sum_{\substack{k \in V_1: \\ k > j}} w_{jk}^1 + \sum_{j \notin V_1} \sum_{\substack{k \notin V_1: \\ k > j}} w_{jk}^2 \geq W$.

We will show that the following problem can be reduced to the decision problem of Definition 1 in polynomial time.

Definition 2 (PARTITION). Let c_1, \dots, c_n be integer, $c_i \in \mathbb{Z}$. *PARTITION* is to decide whether there is $S \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S} c_i = \sum_{i \notin S} c_i$.

The partition problem is one of the six basic NP-complete problems in [22]. Then, giving a polynomial time reduction of *PARTITION* to *DOUBLE-CUT* implies that the latter is also NP-complete. The following proposition states and proves such a result.

Proposition 3. *DOUBLE-CUT* is NP-complete.

Proof. Suppose that $c_1, \dots, c_n \in \mathbb{Z}$ is an instance of *PARTITION*. We will build an instance of *DOUBLE-CUT* that has positive answer if and only if there is $S \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S} c_i = \sum_{i \notin S} c_i$.

Consider an instance of *DOUBLE-CUT* with $G = K_n$ the complete graph

of n nodes and the following weights

$$w_{jk}^1 = w_{jk}^2 = -c_j \cdot c_k \quad \forall j, k \in V,$$

$$W = \frac{1}{2} \sum_{j \in V} c_j^2 - \frac{1}{4} \left(\sum_{j \in V} c_j \right)^2.$$

DOUBLE-CUT has a positive answer if and only if there is $V_1 \subseteq V$ such that

$$\sum_{j \in V_1} \sum_{\substack{k \in V_1: \\ k > j}} -c_j \cdot c_k + \sum_{j \notin V_1} \sum_{\substack{k \notin V_1: \\ k > j}} -c_j \cdot c_k \geq \frac{1}{2} \sum_{j \in V} c_j^2 - \frac{1}{4} \left(\sum_{j \in V} c_j \right)^2. \quad (19)$$

We know that

$$\sum_{j \in V_1} \sum_{\substack{k \in V_1: \\ k > j}} -c_j \cdot c_k = -\frac{1}{2} \left[\left(\sum_{j \in V_1} c_j \right) \cdot \left(\sum_{k \in V_1} c_k \right) - \sum_{j \in V_1} c_j^2 \right].$$

The left-hand side of (19) is then

$$\begin{aligned} & -\frac{1}{2} \left[\left(\sum_{j \in V_1} c_j \right) \cdot \left(\sum_{k \in V_1} c_k \right) + \left(\sum_{j \notin V_1} c_j \right) \cdot \left(\sum_{k \notin V_1} c_k \right) - \sum_{j \in V} c_j^2 \right] \\ & = \frac{1}{2} \left[\sum_{j \in V} c_j^2 - \left(\sum_{j \in V_1} c_j \right)^2 - \left(\sum_{j \notin V_1} c_j \right)^2 \right]. \end{aligned}$$

In general, function $X^2 + Y^2$ when $X + Y$ is constant attains its minimum at $X = Y$. If we call

$$X = \sum_{j \in V_1} c_j \quad \text{and} \quad Y = \sum_{j \notin V_1} c_j,$$

the left-hand side of (19) attains its maximum when $X^2 + Y^2$ attains its minimum, i.e., when $\sum_{j \in V_1} c_j = \sum_{j \notin V_1} c_j = (\sum_{j \in V} c_j)/2$. In such a case, the left-hand side of (19) would be

$$\frac{1}{2} \left[\sum_{j \in V} c_j^2 - 2 \cdot \left(\frac{\sum_{j \in V} c_j}{2} \right)^2 \right] = \frac{1}{2} \sum_{j \in V} c_j^2 - \frac{1}{4} \left(\sum_{j \in V} c_j \right)^2 = W.$$

330 Then, DOUBLE-CUT can only have positive answer when (19) is satisfied as an equality, which happens if and only if there is $V_1 \subseteq V$ such that $\sum_{j \in V_1} c_j = \sum_{j \notin V_1} c_j$. \square

Corollary 2. *The separation of (15) when $J_1 \cup J_2 = J$ is NP-hard.*

Given the computational complexity of separation, we propose an algorithm
 335 that, given an optimal fractional solution \bar{z} and a pair of tied clients $(i_1, i_2) \in$
 E_T , tries to maximize (18) heuristically. An initial partition $J_1, J_2 \subseteq J$ is
 considered at first. Facilities are moved from J_1 to J_2 and viceversa whenever
 this produces an increment of (18) and the number of elements is not below three
 for any of the subsets. If (18) is eventually greater than one, the corresponding
 340 inequality is added as a cut. Algorithm 1 describes the pseudo code of the
 separation heuristic, which is applied for every pair in E_T in each callback. Note
 that the idea of the algorithm can be also applied to i_1, i_2, i_3 pairwise adjacent,
 i.e., inducing a triangle in G_T . Partition will be then made of three subsets,
 $J_1, J_2, J_3 \subseteq J$, and a larger combinatorial number of possible movements among
 345 their components have to be checked.

6. Computational study

The aim of our computational study is twofold. First, we are interested
 in providing an empirical comparison between the LP bounds of (DPLP₂) and
 (DPLP₃). Second, we intend to validate the separation heuristic applied on
 350 (DPLP₃) and test its relative performance with respect to bare formulations
 (DPLP₂) and (DPLP₃).

6.1. Experimental setup

The machine used was an Intel Xeon CPU at 3Ghz×8, with 16GB of RAM.
 Our testbed was based on four instances of the OR-Library, namely pmed1,
 355 pmed2, pmed3 and pmed4, see [23]. In the implementation of (DPLP₃), instead
 of (7), we consider the equality constraints $\sum_{j \in J} \sum_{k > j} z_{ijk} = 1$ for all $i \in I$,
 which avoids the use of a big M in the objective.

As already mentioned in previous section, the heuristic described on Algo-
 rithm 1, which address the case $|C_T| = 2$, can be adapted to separate other
 360 inequalities of the family (17). A preliminary computational study was con-
 ducted to check the performance of the heuristic when adapted to $|C_T| \in \{2, 3\}$.

Algorithm 1 Separation heuristic

- Input** \bar{z} : fractional solution of (DPLP₃).
 $(i_1, i_2) \in E_T$: pair of tied clients.
 ϵ : tolerance for cut violation.
- Variables** J_1 : first subset of facility indices.
 J_2 : second subset of facility indices.
 $Z_{i_1j}(J_1)$: the contribution of j to the left-hand side of the cut when $j \in J_1$.
 $Z_{i_2j}(J_2)$: the contribution of j to the left-hand side of the cut when $j \in J_2$.
- Output** *cut*: a cut in the family (15) violated by \bar{z} , if exists.
- Step 1** $J_1 = \{1, \dots, \lfloor (m+1)/2 \rfloor\}$, $J_2 = \{\lfloor (m+1)/2 \rfloor + 1, \dots, m\}$.
- Repeat**
- Step 2** *improved* = *false*
- Step 3** If $|J_1| \geq 3$, for all $j \in J_1$ do:
- 3.1** $Z_{i_1j}(J_1) = \sum_{\substack{k_1 \in J_1: \\ k_1 < j}} \bar{z}_{i_1 k_1 j} + \sum_{\substack{k_1 \in J_1: \\ k_1 > j}} \bar{z}_{i_1 j k_1}$.
- 3.2** $Z_{i_2j}(J_2) = \sum_{\substack{k_2 \in J_2: \\ k_2 < j}} \bar{z}_{i_2 k_2 j} + \sum_{\substack{k_2 \in J_2: \\ k_2 > j}} \bar{z}_{i_2 j k_2}$.
- 3.3** If $Z_{i_1j}(J_1) < Z_{i_2j}(J_2)$ then
 $J_1 = J_1 \setminus \{j\}$; $J_2 = J_2 \cup \{j\}$;
 improved = *true*; break;
- Step 4** If $|J_2| \geq 3$, for all $j \in J_2$ do:
- 4.1** $Z_{i_1j}(J_1) = \sum_{\substack{k_1 \in J_1: \\ k_1 < j}} \bar{z}_{i_1 k_1 j} + \sum_{\substack{k_1 \in J_1: \\ k_1 > j}} \bar{z}_{i_1 j k_1}$.
- 4.2** $Z_{i_2j}(J_2) = \sum_{\substack{k_2 \in J_2: \\ k_2 < j}} \bar{z}_{i_2 k_2 j} + \sum_{\substack{k_2 \in J_2: \\ k_2 > j}} \bar{z}_{i_2 j k_2}$.
- 4.3** If $Z_{i_1j}(J_1) > Z_{i_2j}(J_2)$ then
 $J_1 = J_1 \cup \{j\}$; $J_2 = J_2 \setminus \{j\}$;
 improved = *true*; break;
- Until** not *improved*
- Step 5** If $\sum_{j_1 \in J_1} \sum_{\substack{k_1 \in J_1: \\ j_1 > k_1}} \bar{z}_{i_1 j_1 k_1} + \sum_{j_2 \in J_2} \sum_{\substack{k_2 \in J_2: \\ j_2 > k_2}} \bar{z}_{i_2 j_2 k_2} > 1 + \epsilon$ then
 cut = $\sum_{j_1 \in J_1} \sum_{\substack{k_1 \in J_1: \\ j_1 > k_1}} z_{i_1 j_1 k_1} + \sum_{j_2 \in J_2} \sum_{\substack{k_2 \in J_2: \\ j_2 > k_2}} z_{i_2 j_2 k_2} \leq 1$
 else *cut* = \emptyset
- Step 6** Return *cut*.
-

Graph G_T was generated having many triangles, so that structures $|C_T| = 3$ were found. For every $i \in \{1, \dots, \lfloor n/2 \rfloor\}$, E_T had edges $(i, n - i + 1)$, $(i, 2i)$, $(2i, n - i + 1)$, $(n - i, n - i + 1)$ and $(i, n - i)$. The results are shown on Table 3.

365 Dual bounds are the same in almost all the cases, except for those depicted in bold. Running times are not always smaller for one of the alternatives. However, we observe that the improvement of the adapted heuristic is not significant in comparison to those cases when it represents the worst alternative. For instance, when $n = 30$ in file `pmed3`, the adapted heuristic spends more than

370 100 seconds more than Algorithm 1. Conversely, the adapted heuristic saves no more than 30 seconds with respect to Algorithm 1 in the best cases. In view of these results, we decided to conduct experiments only with the heuristic of Algorithm 1.

For our computational study, we have taken $n = m = 10, 15, 20, 25, 30, 35,$

375 40. Since Beasley's instances have 100 clients, we have to trim Beasley's matrices (c_{ij}) to obtain the desired size. A range of constant vectors are considered for the opening costs, for all $j \in J$, $f_j = 50, 75, 100, 125, 150, 175, 200$. These costs will imply opening from 12.5% to 68% of candidate facilities in the optimal solution. The testbed is then made of $4(\text{Beasley's}) \times 7(\text{size}) \times 7(\text{opening costs}) = 196$ instances in total. Graph G_T is generated depending on the size of the instance, i.e., on n . For every $i \in \{1, \dots, \lfloor n/2 \rfloor\}$ edges $(i, n - i + 1)$ and $(i, 2i)$ are added to E_T , which gives graphs with density $\frac{2}{n} \times 100(\%)$.

380

Each instance of the testbed is solved under four different configurations in FICO XPRESS MOSEL 64-bit v4.8.4. Default cutting planes, heuristics and

385 presolving strategies are disabled for all the experiments. The configurations are

- C_1 : formulation (DPLP₂),
 - C_2 : formulation (DPLP₃),
 - C_3 : formulation (DPLP₃) and callbacks to Algorithm 1 at the root node
- 390 of the branching tree,

File	n	f_j	OPT	LP		Time (s.)		NodesBB	
				$ C_T = 2$	$ C_T \in \{2, 3\}$	$ C_T = 2$	$ C_T \in \{2, 3\}$	$ C_T = 2$	$ C_T \in \{2, 3\}$
pmed1	10	50	921	921.0	921.0	0.1	0.3	1	1
pmed1	10	100	1161	1161.0	1161.0	0.1	0.1	1	1
pmed1	20	50	2272	2272.0	2272.0	6.3	5.4	1	1
pmed1	20	100	2688	2688.0	2688.0	5.7	4.7	1	1
pmed1	30	50	3410	3410.0	3410.0	139.9	122.0	1	1
pmed1	30	100	3943	3943.0	3943.0	183.7	149.2	1	1
pmed2	10	50	1340	1340.0	1340.0	0.2	0.1	1	1
pmed2	10	100	1590	1590.0	1590.0	0.2	0.1	1	1
pmed2	20	50	2717	2717.0	2717.0	6.3	4.5	1	1
pmed2	20	100	3175	3175.0	3175.0	7.6	5.1	1	1
pmed2	30	50	3969	3969.0	3969.0	110.6	125.1	1	1
pmed2	30	100	4499	4499.0	4499.0	131.4	126.1	1	1
pmed3	10	50	976	976.0	976.0	0.1	0.2	1	2
pmed3	10	100	1209	1209.0	1209.0	0.1	0.1	1	1
pmed3	20	50	2595	2588.0	2588.0	16.5	20.8	5	21
pmed3	20	100	2983	2975.2	2979.3	15.4	14.4	13	3
pmed3	30	50	4108	4108.0	4108.0	295.3	379.8	1	1
pmed3	30	100	4687	4672.4	4673.0	332.4	483.4	21	11
pmed4	10	50	1442	1442.0	1442.0	0.1	0.1	1	1
pmed4	10	100	1701	1701.0	1701.0	0.1	0.1	1	1
pmed4	20	50	3165	3163.0	3165.0	11.3	11.2	1	1
pmed4	20	100	3627	3627.0	3627.0	10.9	11.7	1	1
pmed4	30	50	4491	4491.0	4491.0	183.5	168.4	1	1
pmed4	30	100	5019	5019.0	5019.0	140.4	154.8	1	1

Table 3: Preliminary test to check the performance when adapting the heuristic

- C_4 : formulation (DPLP₃) and callbacks to Algorithm 1 throughout the branching tree.

Time limit is set to half an hour for all the experiments. The last two configurations solved all the instances within the time limit. Formulations (DPLP₂) and (DPLP₃) solved 122 and 110 instances, respectively.

Tables 4 and 5 show the computational results. The LP gap, running time and the number of nodes explored during the branching are displayed in different columns for the different configurations. The LP gap is relative to the optimal value, and is calculated as a percentage. When Algorithm 1 is used, the LP gap is calculated using the LP bound after adding the cuts at the root node. Due to the large number of instances, we have to display average results. Table 4 shows results on average throughout different costs f_j considered, for every Beasley's file and size. On the other hand, computational results are averaged by input size n on Table 5, where a row is displayed for every Beasley's file and opening cost constant vector. The following section discusses the results.

6.2. Comparative analysis

From Table 4, we can see that (DPLP₂) and (DPLP₃) fail to solve the bigger instances, namely those with $n = 35, 40$. Moreover, (DPLP₃) still has difficulties to cope when $n = 30$. Using the separation heuristic with (DPLP₃) can dramatically improve the running time to the point of solving all the instances within 538 seconds. Such an improvement with respect to (DPLP₂) and (DPLP₃) begins to appear with $n = 20$. Columns concerning the nodes of the branch and bound tree also reflect a better performance when the separation is incorporated to the solving procedure. Nevertheless, as opposed to running times, (DPLP₃) explores less nodes than (DPLP₂) to obtain the optimal solution. This makes very much sense since (DPLP₃) LP bounds are better. The number of variables to manage is probably one of the factors that influence higher running times of (DPLP₃). The fact that many of the instances are solved at the root when cuts are added is consistent with the extremely tight LP gaps that we get in those cases. On the other hand, as expected from the theoretical findings, the

File	n	LP GAP			Time (s.)				NodesBB			
		C_1	C_2	C_3	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
pmed1	10	4.3	2.2	0.1	0.0	0.3	0.2	0.0	29	19	1	1
	15	14.5	8.8	0.0	0.4	3.5	1.0	0.4	3664	572	1	1
	20	13.5	9.0	0.1	21.3	41.6	4.0	2.4	83068	5408	1	1
	25	12.8	8.5	0.0	267.2	456.9	15.2	9.4	459087	22114	1	1
	30	15.7	9.8	0.1	722.5	1050.3	105.5	50.7	528145	11028	20	7
	35	18.4	11.5	0.0	1088.2	1210.7	293.5	149.6	385108	4265	1	1
pmed2	40	21.2	14.0	0.1	1200.2	1213.3	885.1	488.4	303871	1877	4	4
	10	13.0	6.8	0.0	0.1	0.5	0.1	0.1	218	115	1	1
	15	14.2	8.8	0.0	0.3	3.7	0.5	0.5	1566	955	1	1
	20	16.0	8.7	0.0	3.5	35.9	2.5	2.5	15747	5916	3	2
	25	17.6	8.1	0.0	37.3	165.0	9.2	9.3	61588	5258	2	1
	30	19.9	11.9	0.2	879.9	1203.5	47.4	48.7	600064	14171	18	9
pmed3	35	21.0	12.7	0.1	1200.2	1209.4	113.2	130.2	365431	5046	5	4
	40	22.2	13.6	0.1	1200.1	1214.8	483.3	509.7	209979	1660	12	6
	10	6.4	2.0	0.0	0.0	0.3	0.0	0.1	18	16	1	1
	15	11.5	5.8	0.1	0.3	4.3	0.4	0.4	820	598	1	1
	20	22.2	14.6	0.0	250.1	357.3	3.0	2.9	747206	67113	1	1
	25	20.2	15.0	0.0	514.6	649.6	11.0	11.0	942326	26338	2	1
pmed4	30	21.0	14.6	0.0	916.3	1204.2	48.9	48.0	814889	13579	2	1
	35	25.4	17.1	0.2	1200.2	1209.7	216.8	219.7	472411	4249	26	1
	40	24.0	15.7	0.0	1200.3	1214.3	534.6	538.4	238041	1494	2	2
	10	14.6	6.5	0.0	0.3	0.7	0.1	0.1	180	253	1	1
	15	11.7	6.4	0.0	1.0	3.5	0.4	0.4	1300	560	1	1
	20	13.1	7.0	0.0	8.0	26.3	2.4	2.4	16690	2244	2	1
pmed4	25	18.5	10.6	0.2	469.7	675.4	11.6	11.3	713877	29653	8	4
	30	18.8	9.8	0.0	621.9	1116.3	39.2	40.6	514557	7849	2	1
	35	20.8	12.6	0.0	1200.2	1210.1	133.3	138.5	380402	4950	2	2
	40	22.2	14.3	0.1	1200.4	1210.0	460.5	466.6	229642	3005	17	3

Table 4: Average computational results as a function of source file and n

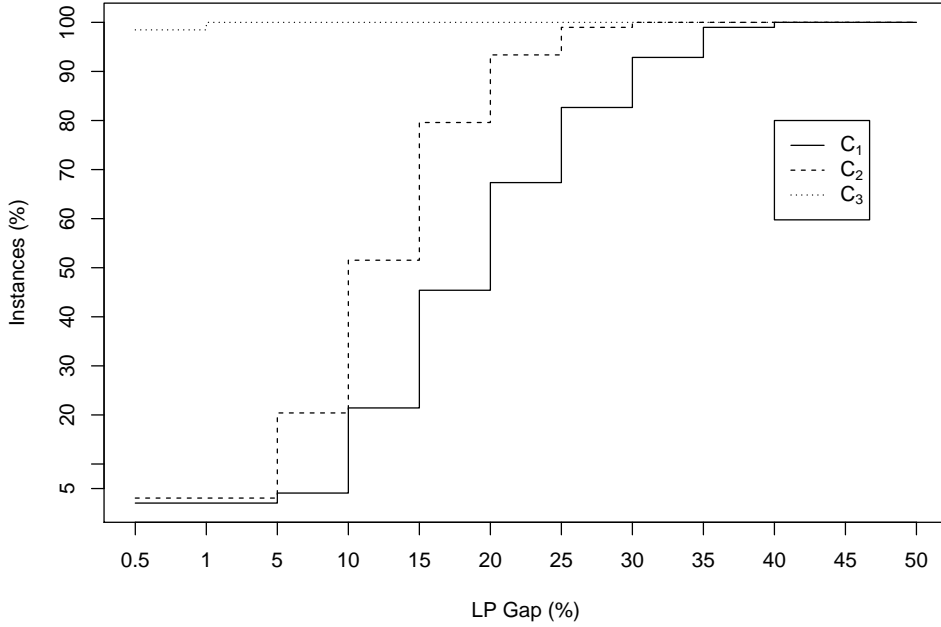


Figure 7: Percentage of instances within different LP gaps

LP gap is always smaller for (DPLP₃), being the difference with (DPLP₂) more noticeable with the increase of the problem size.

Table 5 allows to compare the formulations when opening costs change. We observe that the instances with tightest gaps are those with highest opening costs, for which fewer facilities are open in an optimal solution. Conversely, running times do not seem to directly depend on f_j in overall terms, nor does the number of nodes explored in the branching.

We further provide several charts that better illustrates the comparison of the four configurations. These depict, for every configuration, an step function to represent the number of instances solved in total after some time or some nodes of the branching tree. LP gaps throughout the testbed are also represented with step functions.

File	f_j	LP GAP			Time (s.)				NodesBB			
		C_1	C_2	C_3	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
pmed1	50	25.8	16.6	0.0	703.7	667.7	164.0	90.3	750966	15994	2	2
	75	20.0	12.6	0.0	594.9	702.2	182.0	98.4	366724	12417	1	1
	100	15.8	10.1	0.1	525.9	616.7	199.7	104.8	217873	6765	12	5
	125	12.6	8.1	0.0	517.8	551.5	197.9	105.7	214929	3921	9	3
	150	10.1	6.4	0.0	368.9	534.1	186.3	102.2	80211	2830	1	1
	175	8.6	5.3	0.0	351.8	478.7	187.2	102.9	103523	2052	2	3
	200	7.5	4.6	0.1	236.9	425.6	187.4	96.6	28744	1304	3	2
pmed2	50	28.7	16.9	0.1	527.4	554.7	79.0	86.5	377056	8417	4	2
	75	23.7	13.7	0.0	530.3	578.9	85.9	94.1	285738	8135	4	3
	100	19.6	11.3	0.0	521.0	553.1	107.4	112.4	191922	4750	16	4
	125	16.3	9.1	0.0	516.0	542.7	91.6	100.7	165141	3559	1	1
	150	13.9	7.6	0.0	422.5	536.5	92.0	100.9	96590	3260	2	2
	175	11.8	6.5	0.1	412.3	535.3	97.8	105.9	74852	2485	5	4
	200	9.9	5.5	0.1	391.7	531.5	102.6	100.6	63294	2513	12	8
pmed3	50	29.9	19.2	0.1	834.4	825.1	101.7	103.7	1161895	46920	2	2
	75	24.7	15.8	0.0	753.3	816.7	125.6	128.2	794468	33435	9	3
	100	20.4	13.1	0.0	681.9	731.8	124.2	118.7	560321	13682	3	1
	125	17.1	11.1	0.0	540.4	603.0	129.7	124.8	297781	7011	1	1
	150	14.8	9.7	0.0	521.7	560.8	123.1	123.0	216003	4973	3	2
	175	12.8	8.5	0.1	387.8	551.6	117.0	120.3	111752	4010	14	11
	200	11.1	7.5	0.0	362.3	550.6	93.4	101.8	73489	3356	3	2
pmed4	50	28.7	16.9	0.0	690.6	670.9	79.7	82.8	642622	16419	2	2
	75	23.3	13.2	0.0	687.4	686.7	85.1	88.5	548154	12506	2	2
	100	19.1	10.6	0.1	593.0	694.4	91.4	98.4	343380	8601	1	1
	125	15.6	8.5	0.0	432.8	586.2	88.4	92.6	116445	3913	1	1
	150	13.0	7.0	0.1	379.9	593.2	92.9	92.1	72110	3416	5	2
	175	10.8	5.8	0.0	363.3	514.9	114.9	106.7	87922	2167	16	2
	200	9.1	5.0	0.1	354.6	496.2	95.0	98.7	46012	1492	5	4

Table 5: Average computational results as a function of source file and costs f_j

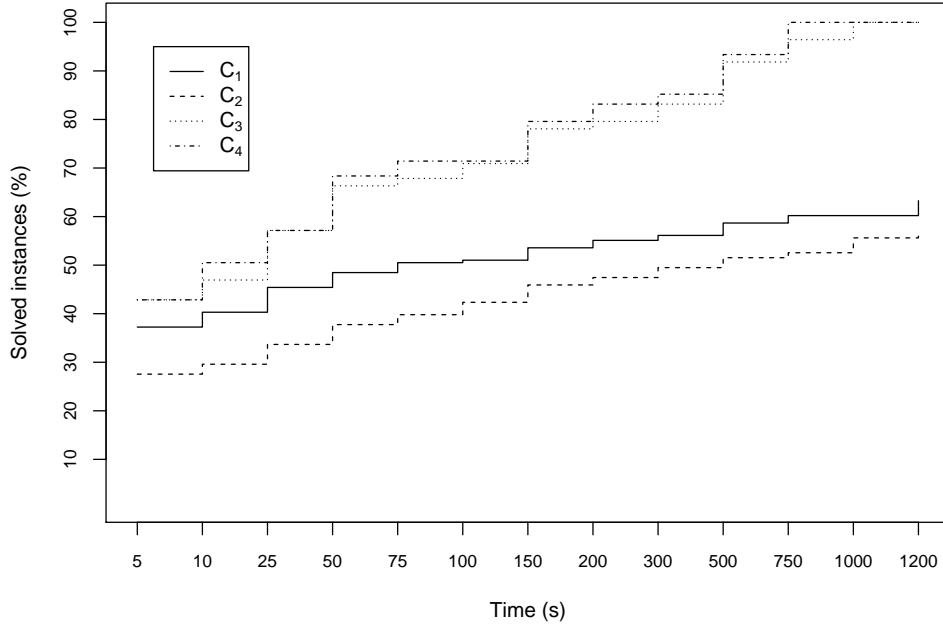


Figure 8: Percentage of solved instances as a function of time

To begin with, a comparison of the LP gaps is given in Figure 7. The ordinate axis shows the percentage of instances that have an LP gap smaller than each LP gap mark. The figure shows relative gaps within the optimum—
 435 which is known for all the instances—and the linear relaxations of (DPLP₂) and (DPLP₃) when no cut is added and after adding the cuts to the root. On the one hand, experiments support Proposition 1, i.e., the LP gap is always tighter for (DPLP₃). On the other hand, applying the separation heuristic at
 440 the root surprisingly closes the gap in almost all the cases. This suggests that clique facets (15) are very close to cover the side of the polyhedron \mathcal{B}_{dplp} that faces towards the direction of optimization.

Figure 8 gives a comparison of the running times of the four configurations tested. The abscissa axis shows the time in seconds. The ordinate axis shows the

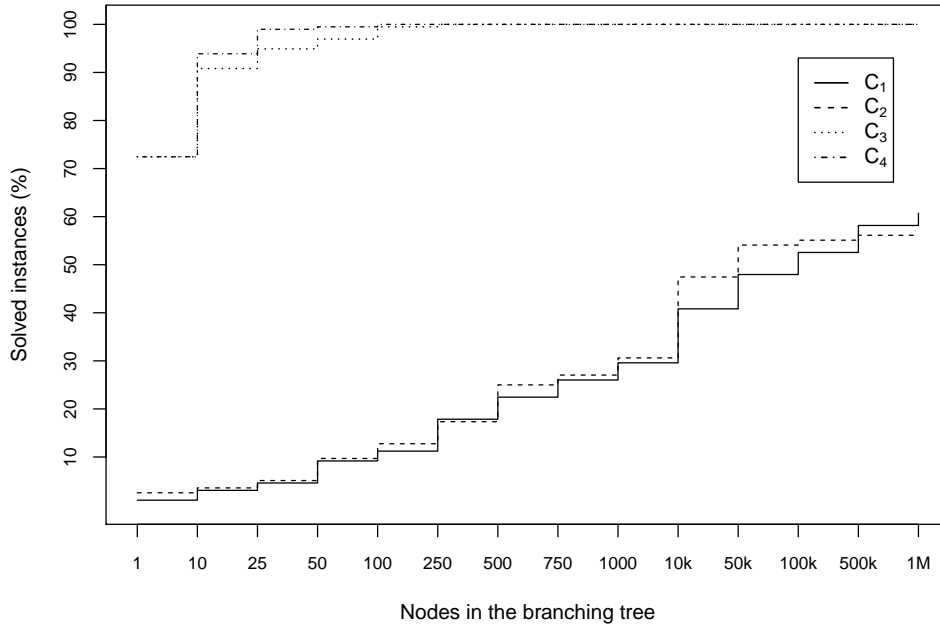


Figure 9: Percentage of solved instances as a function of the nodes of the branching tree

445 percentage of instances that are solved before each time tick mark. According to our experiments, (DPLP₂) clearly outperforms (DPLP₃). However, when the separation heuristic is incorporated to (DPLP₃), things turn out differently. The resulting configurations solve about 70% of the instances in 75 seconds, while (DPLP₂) can only solve 62% of the instances within the time limit.

450 Finally, Figure 9 compares the four configurations in terms of the nodes of the branching tree. The abscissa axis shows the number of nodes in the tree, and the ordinate axis depicts the percentage of instances solved up to those number of nodes. Formulations (DPLP₂) and (DPLP₃) present a very similar performance in overall terms. The use of the separation heuristic sharply reduces
 455 the size of the tree, which consists of only one node for more than 70% of the instances and no more than 50 nodes throughout the testbed.

7. Conclusions

In this work, a new variant of one of the fundamental problems in discrete location is proposed and studied. Specifically, we propose a modification of the Simple Plant Location Problem where double assignment is considered and some pairs of clients have to share at least one common facility. A set packing three-indexed formulation is proposed and its linear relaxation is proved to provide better bounds than that of a formulation with classic variables. The facial structure of the proposed set packing polytope is investigated and all its clique facets are uncovered. A separation heuristic for the devised inequalities is designed and tested in a computational study. Our experiments reflect the utility of clique inequalities and support the theoretical comparison of the formulations considered.

Our work can be extended for the more general case of multiple assignment (each client allocated to three, four... facilities). We could write a formulation analogous to (DPLP₃) but with four, five... indices for the z -variables. The ideas to obtain the clique facets would be still valid for the resulting formulation. However, the number of variables would grow exponentially with the number of assignments considered.

References

- [1] J. Krarup, P. Pruzan, The simple plant location problem: Survey and synthesis, *European Journal of Operational Research* 12 (1983) 36–81.
- [2] G. Laporte, S. Nickel, F. S. da Gama (Eds.), *Location Science*, Springer International Publishing, 2015.
- [3] D. Cho, E. Johnson, W. Padberg, M. Rao, On the uncapacitated plant location problem I: valid inequalities and facets, *Mathematics of Operations Research* 8 (4) (1983) 579–589.
- [4] M. Labbé, H. Yaman, Projecting the flow variables for hub location problems, *Networks: An International Journal* 44 (2) (2004) 84–93.

- 485 [5] G. Cornuéjols, M. Fisher, G. Nemhauser, On the uncapacitated location problem, *Annals of Discrete Mathematics* 1 (1977) 163–177.
- [6] M. Guignard, Fractional vertices, cuts and facets of the simple plant location problem, *Mathematical Programming* 12 (1980) 150–162.
- [7] G. Cornuéjols, J. Thizy, Some facets of the simple plant location polytope, 490 *Mathematical Programming* 23 (1982) 50–74.
- [8] L. Cánovas, M. Landete, A. Marín, On the facets of the simple plant location packing polytope, *Discrete Applied Mathematics* 124 (2002) 27–53.
- [9] A. Warszawski, Multi-dimensional location problems, *Operational Research Quarterly* 24 (2) (1973) 165–179.
- 495 [10] L. Cánovas, S. García, M. Labbé, A. Marín, A strengthened formulation for the simple plant location problem with order, *Operations Research Letters* 35 (2) (2007) 141–150.
- [11] C. Swamy, D. B. Shmoys, Fault-tolerant facility location, *ACM Transactions on Algorithms* 4 (4) (2008) 51.
- 500 [12] T. Cui, Y. Ouyang, Z. Shen, Reliable facility location design under the risk of disruptions, *Operations Research* 58 (4) (2010) 998–1011.
- [13] T. Xifeng, Z. Ji, X. Peng, A multi-objective optimization model for sustainable logistics facility location, *Transportation Research Part D: Transport and Environment* 22 (2013) 45–48.
- 505 [14] E. Fernández, M. Landete, Fixed-charge facility location problems, in: G. Laporte, S. Nickel, F. S. da Gama (Eds.), *Location Science*, Springer International Publishing, 2015, pp. 47–77.
- [15] M. Fischetti, I. Ljubić, M. Sinnl, Redesigning benders decomposition for large-scale facility location, *Management Science* 63 (7) (2016) 2146–2162.

- 510 [16] I. Contreras, Hub location problems, in: G. Laporte, S. Nickel, F. S. da Gama (Eds.), *Location Science*, Springer International Publishing, 2015, pp. 311–344.
- [17] L. Kaufman, M. Eede, P. Hansen, A plant and warehouse location problem, *Journal of the Operational Research Society* 28 (3) (1977) 547–554.
- 515 [18] L. Wu, X. Zhang, J. Zhang, Capacitated facility location problem with general setup cost, *Computers & Operations Research* 33 (5) (2006) 1226–1241.
- [19] E. Gourdin, M. Labbé, H. Yaman, Telecommunication and location, in: Z. Drezner, H. Hamacher (Eds.), *Facility Location: Applications and Theory*, Vol. 45, Springer, 2002, Ch. 9, pp. 275–305.
- 520 [20] A. Marín, M. Pelegrín, Adding incompatibilities to the simple plant location problem: Formulation, facets and computational experience, *Computers and Operations Research* 104 (2019) 174–190.
- [21] M. Padberg, On the facial structure of set packing polyhedra, *Mathematical Programming* 5 (1973) 199–215.
- 525 [22] M. Garey, D. Johnson, *Computers and intractability : a guide to the theory of NP-completeness*, W.H. Freeman & Co, New York, 1979.
- [23] J. Beasley, Or-library, Available at <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html> (1990).