A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences☆

Herminia I. Calvete[a,*], Carmen Galé[b], José A. Iranzo[c], José-Fernando Camacho-Vallejo[d], Martha-Selene Casas-Ramírez[d]

[a]*Departamento de Métodos Estadísticos, IUMA, Universidad de Zaragoza, Pedro Cerbuna 12, 50009 Zaragoza, Spain.*

[b]*Departamento de Métodos Estadísticos, IUMA, Universidad de Zaragoza, María de Luna 3, 50018 Zaragoza, Spain.*

[c]*Centro Universitario de la Defensa de Zaragoza, IUMA, Carretera de Huesca s/n, Zaragoza 50090, Spain.*

[d]*Facultad de Ciencias Físico-Matemáticas, Universidad Autónoma de Nuevo León, Avenida Universidad s/n, 66450 San Nicolás de los Garza, NL, México.*

Abstract

This paper addresses a generalized version of the facility location problem with customer preferences which includes an additional constraint on the number of customers which can be allocated to each facility. The model aims to minimize the total cost due to opening facilities and allocating customers while taking into account both customer preferences for the facilities and these cardinality constraints. First, two approaches to deal with this problem are proposed, which extend the single level and bilevel formulations of the problem in which customers are free to select their most preferred open facility. After analyzing the implications of assuming any of the two approaches, in this research, we adopt the approach based on the hierarchical character of the model which leads to the formulation of a bilevel optimization problem. Then, taking advantage of the characteristics of the lower level problem, a single level reformulation of the bilevel optimization model is developed based on duality theory which does not require the inclusion of additional binary variables. Finally, we develop a simple but effective matheuristic for solving the bilevel optimization problem whose general framework follows that of an evolutionary algorithm and exploits the bilevel structure of the model. The chromosome encoding pays attention to the upper level variables and controls the facilities which are open. Then, an optimization model is solved to allocate customers in accordance with their preferences and the availability of the open facilities. A computational experiment shows the effectiveness of the matheuristic in terms of the quality of the solutions yielded and the computing time.

Keywords: Facility location, Cardinality constraint, Capacity, Preferences, Bilevel optimization, Matheuristic, Evolutionary algorithm

1. Introduction

Facility location problems are amongst the most widely studied in the literature of Operations Research. They have been applied to locate production plants, warehouses, schools, fire stations, hospitals, etc., and thus play a central role in a great number of decision-making problems in both the public and the private sectors. In a nutshell, facility location problems are concerned with selecting the best placement for a number of facilities to serve a set of customers, in accordance with the optimality criteria established. The criteria proposed in the literature usually take into account costs and distances, as well as the service provided to customers. As stated by ReVelle et al. [27]: 'Even though the contexts in which these models are situated may differ, their main features are always the same: a space including a metric, customers whose locations in the given space are known, and facilities whose locations have to be determined according to some objective function.' Many variants of the problem have been proposed, which differ in the type of space considered (continuous, discrete, or with network structure), the objectives, the facility features (uncapacitated or capacitated), or the time horizon, amongst others. There is also a wide variety of solution methods, ranging from the exact solution of the optimization model to the developing of heuristic or metaheuristic procedures when the size of the problem or its complexity prevent the use of exact methods. Without being exhaustive, [12, 14, 21, 22, 25, 26, 27] and the references therein provide a comprehensive survey of the topic.

In this paper we focus on a generalized version of the simple plant location problem with order (SPLPO) first proposed by Hanjoul and Peeters [17]. The SPLPO extends the simple plant location problem (SPLP), a discrete facility location problem which consists of selecting some facilities to be opened from a set of candidates and determining the allocation of the customers to the open facilities, aiming to minimize the total cost due to opening facilities and allocating customers. Although some previous papers [28, 33] had considered

☆The research of H.I. Calvete, C. Galé and J.A. Iranzo has been funded by the Spanish Ministry of Economy, Industry and Competitiveness under grants ECO2013-47129-C4-4-R, ECO2016-76567-C4-3-R and by the Gobierno de Aragón under grant E58 (FSE) and E41-17R (FEDER 2014-2020 "Construyendo Europa desde Aragón"). The research of J.F. Camacho-Vallejo has been partially supported by the Mexican National Council for Science and Technology (CONACYT) through grant SEP-CONACYT CB-2014-01-240814, and grant PRODEP/511-6/17/7425 for research stays during his sabbatical year.

*Corresponding author

Email addresses: herminia@unizar.es (Herminia I. Calvete), cgale@unizar.es (Carmen Galé), joseani@unizar.es (José A. Iranzo), jose.camachovl@uanl.edu.mx (José-Fernando Camacho-Vallejo), martha.casasrm@uanl.edu.mx (Martha-Selene Casas-Ramírez)

the reaction of customers if they are free to choose an open facility by introducing additional constraints to assign customers to their closest facility, the idea of allowing the customers to select the facility they will patronize is credited to Hanjoul and Peeters [17]. These authors assumed that each customer had a preference ordering with respect to the facilities which depended on his/her personal characteristics, as well as the features of the sites and the trips to the sites. The authors proposed the SPLPO in which each customer chooses the open facility which is his/her most preferred one. This model assumes that the decision-maker in charge of selecting the facilities, the locator, knows the preference orderings of customers and takes them into account when selecting the facilities to be opened. In the mathematical model the preference orderings are written as a set of constraints which are added to the formulation of the SPLP. For solving the SPLPO, the authors developed a heuristic algorithm and solved small examples. Cánovas et al. [7] analyzed the SPLPO and strengthened the formulation by introducing valid inequalities and applying several preprocessing rules.

Hanjoul and Peeters [17] in their pioneering work also recognized the existence of an implicit hierarchical structure associated with the two interrelated subproblems involved in the SPLPO: the location problem, which refers to the selection of facilities, at the upper level of the hierarchy, and the allocation problem, which refers to the assignment of customers, at the lower level. Based on the fact that bilevel optimization models [11, 13] provide a framework to deal with decision processes involving a hierarchical structure, Hansen et al. [18], Vasilyev and Klimentova [31] and Vasilyev et al. [32] formulated the SPLPO as a bilevel model. The upper level decision maker, which selects the facilities, aims to minimize the total cost. In the lower level problem, customers are allocated aiming to minimize the sum of preferences (the smaller the value, the greater the preference). The authors assumed that the preference ordering is strict, that is, for any pair of facilities, each customer prefers one of them. Based on this assumption, the bilevel model can be rewritten as the single level formulation proposed in [7, 17]. Then, Hansen et al. [18] proposed a reformulation of this single level problem which dominated previous formulations from the point of view of their linear programming relaxation. Vasilyev et al. [32] used a new family of valid inequalities rather than increasing the number of variables. Based on the previous formulation, Vasilyev and Klimentova [31] developed a branch and cut method to find an optimal solution. Camacho-Vallejo et al. [6] proposed an evolutionary algorithm to solve the bilevel model.

The key point in the above mentioned strategies developed to deal with the SPLPO is the absence of a cardinality constraint associated with each facility which restricts the number of customers that can be allocated to it. This fact guarantees that, once it has been decided which facilities are to be opened, each customer can be allocated to his/her most preferred facility because there is enough room in each facility for them. This paper generalizes the SPLPO to include cardinality constraints, providing a more realistic approach, while aiming to maintain that each customer should be served in accordance with his/her preferences. This problem will be termed C-SPLPO. For this purpose, we assume the existence of an upper bound on the number of customers which can be allocated to each facility. Therefore, it is no longer true that customers can freely choose their most preferred open facility. As will be shown in Section 2, modeling this version of the SPLPO is not a trivial matter since the preference ordering interacts with the cardinality constraint. When there are several customers who want to be allocated to the same facility and this facility cannot serve all those customers, a conflict arises. In fact, it may be possible that no selection of facilities allows customers to be allocated to their most preferred open facility.

In this paper we analyze the implications of extending the two above mentioned approaches applied to deal with the SPLPO. As a result of this analysis, we propose to consider the implicit hierarchical structure of the C-SPLPO and model it using a bilevel optimization model. In the proposed model, the upper level decision maker selects the facilities to be opened while taking into account the reaction of customers. This model extends that proposed in [18] for the SPLPO. Typical examples of such cases are customers of public services like health care services, emergency services, social services, etc. In all of these cases, in general, the locator aims to minimize the total cost of locating facilities while bearing in mind customer preferences globally. For instance, let us consider the allocation of students to schools in a particular school district. For a given grade in elementary, middle or high school, every school can have one or more modules or classes, each being able to accommodate a certain number of students. The public administration knows the cost of opening a module in every school as well as the cost of transporting students from their homes to school. It also knows the preferences of each student regarding the school to which he/she would like to be allocated (obtained, for instance, from surveys). Then, the public administration plans the allocation aiming to minimize the total cost, but taking into account the students' preferences.

To the best of our knowledge, there are only two papers dealing with bilevel capacitated facility location problems. In Casas-Ramírez et al. [9] capacity is considered in the lower-level problem when allocating customers to facilities based on the customers preferences. The main difference with our problem, is that in [9] each customer has a demand and the capacity of a facility refers to the amount of demand it can deal with. These facts convert the lower level problem into the well-known generalized assignment problem, which is NP-hard. Hence, the special structure exploited in our research cannot be considered in the problem with generalized demands. Additionally, due to the complexity that exists in the lower-level in the problem presented in [9], semi-feasible bilevel solutions are proposed. In contrast, the evolutionary algorithm developed in the current research only deals with bilevel feasible solutions, thus guaranteeing the bilevel feasibility of the solution provided. Caramia and Mari [8] consider that the capacity of the facilities is a decision variable. That is, the leader decides the capacity of each located facility aiming to minimize costs. The lower-level problem consists of allocating customers to located facilities respecting their capacity, aiming to maximize the profit. In particular, their upper level problem has a constraint that purely depends on followers variables (coupling constraint). This is a crucial part of the decomposition approach proposed for solving the problem. The main difference with our proposed problem relies in the manner in which capacity is considered. In other words, they consider capacity as a decision while in our research capacity is a parameter. Moreover, in [8] no customer preferences are taken into consideration.

After selecting the hierarchical approach to model the C-SPLPO, a single level reformulation of the bilevel model is developed based on the fact that the coefficient matrix of the lower level problem is unimodular. Thus, the integrality condition on the lower level variables can be relaxed, the lower level problem can be reformulated as a linear program, and the complementary slackness conditions are necessary and sufficient for optimality. Unlike the classical reformulation of bilevel optimization problems using Karush-Kuhn-Tucker (KKT) conditions, the proposed reformulation avoids the inclusion of additional binary variables. The paper also develops a matheuristic whose general framework follows that of an evolutionary algorithm. The chromosome encoding concentrates on the upper level variables and controls the facilities which are open. Then, customers are allocated to the corresponding facilities by solving a biobjective transportation problem which takes into account the cardinality constraint of the open facilities and the optimistic approach assumed in the bilevel formulation. This procedure allows us to associate a bilevel feasible solution to every chromosome and thus evaluate its fitness as the upper level objective function of its associated bilevel feasible solution. The remainder of the paper is organized as follows. Section 2 discusses alternative formulations for the C-SPLPO and presents the mathematical formulation of the bilevel optimization model we propose. Section 3 goes on to transform the bilevel problem into a single level mixed integer optimization model using duality theory and the properties of the lower level problem. The matheuristic is developed in Section 4. Using a set of instances which are variants of established synthetic benchmark instances, section 5 analyzes the computational performance of the matheuristic and gives an in-depth insight into the differences of the above mentioned alternative formulations. Finally, Section 6 sets out future research directions and some concluding remarks.

2. Problem formulation

Consider a set of potential facilities $I = \{1, \dots, n\}$ and a set of customers $J = \{1, \dots, m\}$. Each facility $i \in I$ has associated a nonnegative fixed cost f_i which refers to opening/handling the facility, and a parameter q_i which indicates the maximum number of customers which can be allocated to it, called capacity. There is also a nonnegative cost c_{ij}, $i \in I$, $j \in J$, associated with allocating customer j to facility i. Moreover, we assume that each customer $j \in J$ has ranked the facilities from best to worst, i.e. has a set of predefined nonnegative preferences $g_{ij} \in \{1, \dots, n\}$, $i \in I$. We assume that the smaller the value, the greater the preference. The goal of the C-SPLPO is to select a subset of the potential facilities in order to minimize the total cost, bearing in mind their capacity and the reaction of customers in terms of their preferred facilities. Notice that if $q_i \geqslant m$, for all $i \in I$, the C-SPLPO reduces to the SPLPO.

In order to formulate the C-SPLPO, the first issue we should notice is that there is not a single way of considering the reaction of customers. We can include the individual customer preferences as constraints, thus extending the classical formulation of the SPLPO by Hanjoul and Peeters [17]. Or the preferences can be considered globally, aiming to minimize a function of them. In this paper, we propose to consider the utilitarian approach in which the goal is to minimize the sum of the utilities of the customers, where the utility or satisfaction level is measured through the customers ranking of the facility. This formulation is appropriate for public services in which instead of seeking to satisfy the preferences of each individual customer, the utilitarian approach is taken to evaluate the satisfaction of customers as a whole. This formulation would extend the bilevel formulation of Hansen et al. [18]. Unlike the SPLPO case, the two approaches are not equivalent when there is an upper bound on the number of customers which can be allocated to every facility. Therefore, to assume one formulation or the other can provide very different results. Next, we consider both formulations and analyze their impact on the feasible region and the optimal solution.

We define the variables:

$$y_i = \begin{cases} 1, & \text{if facility } i \text{ is selected to be open} \\ 0, & \text{otherwise} \end{cases} \qquad i \in I$$

$$x_{ij} = \begin{cases} 1, & \text{if customer } j \text{ is allocated to the facility } i \\ 0, & \text{otherwise} \end{cases} \qquad i \in I,\ j \in J$$

When there is no need to explicitly identify the indices of the variables, we will denote by y and x the variables $\{y_i\}_{i\in I}$ and $\{x_{ij}\}_{i\in I, j\in J}$, respectively.

2.1. C-SPLPO-1: Individual customer preferences as constraints

Model C-SPLPO-1 incorporates individual customer preferences as constraints. It can be formulated as:

$$\min_{x,y} \quad \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij} x_{ij} \tag{1a}$$

subject to

$$\sum_{i\in I} x_{ij} = 1, \quad j \in J \tag{1b}$$

$$\sum_{\{k: g_{ij} \geqslant g_{kj}\}} x_{kj} \geqslant y_i, \quad i \in I,\ j \in J \tag{1c}$$

$$\sum_{j\in J} x_{ij} \leqslant q_i y_i, \quad i \in I \tag{1d}$$

$$y_i \in \{0,1\}, \quad i \in I \tag{1e}$$

$$x_{ij} \in \{0,1\}, \quad i \in I,\ j \in J \tag{1f}$$

The objective function (1a) minimizes the total cost due to opening facilities and allocating customers. Constraints (1b) guarantee that each customer is allocated to exactly one facility. Constraints (1c) ensure that if the i-th facility is opened, then the customer j must be allocated to a facility that is at least as good as i according to his/her preference ordering. Cardinality constraints (1d) enforce that customers can only be allocated to open facilities and, besides, they guarantee that as many customers can be allocated to each facility as its capacity allows. Finally, constraints (1e) and (1f) impose that all variables are binary.

In this model, every customer must be allocated to his/her most preferred open facility, i.e. if a particular facility is open, due to constraints (1c) every customer for which this facility is the most preferred among all the open facilities should be allocated to it. However, this may be impossible because of cardinality constraints (1d). Therefore, if none of the ways in which facilities could be selected to be opened allows this allocation, the C-SPLPO-1 would be infeasible. Thus, to be able to provide an optimal solution, this formulation requires the existence of selections of the open facilities in such a way that every customer can be allocated to his/her most preferred open facility. This will be reconsidered again in section 2.3 with the help of an illustrative numerical example.

2.2. C-SPLPO-2: A bilevel approach

This formulation considers the underlying hierarchical structure of the problem. The upper level decision maker decides on the open facilities, while the lower level decision maker allocates customers to the open facilities. The goal of the former is to minimize total cost; the goal of the latter is to minimize the total preference. Therefore, the C-SPLPO-2 can be formulated as the following binary bilevel optimization problem:

$$\min_{y} \quad \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \tag{2a}$$

subject to

$$\sum_{i\in I} q_i y_i \geqslant m \tag{2b}$$

$$y_i \in \{0,1\}, \quad i \in I \tag{2c}$$

where, for every y fixed, x solves the problem:

$$\min_{x} \quad \sum_{i\in I}\sum_{j\in J} g_{ij}x_{ij} \tag{2d}$$

subject to

$$\sum_{i\in I} x_{ij} = 1, \quad j \in J \tag{2e}$$

$$\sum_{j\in J} x_{ij} \leqslant q_i y_i, \quad i \in I \tag{2f}$$

$$x_{ij} \in \{0,1\}, \quad i \in I,\ j \in J \tag{2g}$$

The objective function (2a) minimizes the total cost due to opening facilities and allocating customers. Constraint (2b) guarantees that open facilities provide enough room to allocate all the customers and constraints (2c) ensure that variables y are binary. Constraints (2b) and (2c) provide the constraint region of variables y, which will be denoted by S_y. The lower level problem is represented by (2d)-(2g). Note that the binary variables of the upper level are parameters of the lower level problem. The objective function (2d) minimizes the global preference of customers. Constraints (2e) ensure that each customer is allocated to exactly one facility. Cardinality constraints (2f) enforce that customers can only be allocated to open facilities and, besides, they guarantee that as many customers can be allocated to each facility as its capacity allows. Finally, constraints (2g) guarantee that variables x are binary. Constraint (2b) is redundant. Indeed, to be a bilevel feasible solution the point (y, x) needs x to be an optimal solution of the lower level problem and constraints (2e) of this problem require all customers to be allocated. Hence, the global capacity provided by the open facilities needs to be at least m. However, we keep this constraint because in the algorithm it allows us to reduce the set in which the upper level variables need to be searched. All the values y which do not satisfy constraint (2b) are implicitly discarded because the corresponding lower level problem is not feasible. Notice that, unlike C-SPLPO-1, in C-SPLPO-2 individual preferences are not imposed as constraints. Instead, given the values of the variables y, the lower level problem provides the allocation of customers which minimizes the total preference.

For $y \in S_y$, a feasible solution of the bilevel problem (2) is obtained by solving the lower level problem (2d)-(2g). One main concern in bilevel optimization is the existence of multiple optima for the lower level problem. This fact can result in an ill-posed bilevel optimization model. To overcome this difficulty, several approaches have been proposed in the bilevel optimization literature, the most common being the optimistic approach in which the upper level decision maker is enabled to select the lower level optimal solution that suits him/her best [11, 13].

Papers dealing with the bilevel approach of the SPLPO assume that for each customer $j \in J$, preferences are distinct, i.e. $g_{ij} \neq g_{i'j}$, for all $i, i' \in I$. This assumption guarantees that the bilevel model is well-posed since there exists a unique optimal allocation of customers for any arbitrary selection of the variables y [18]. However, that assumption no longer ensures the uniqueness of the optimal solution of the lower level problem when the cardinality constraint is added. In this paper we assume the optimistic approach to the bilevel formulation of the C-SPLPO-2, which is equivalent to assuming that the objective function (2a) is minimized over y and x. Under this assumption, for a given $y \in S_y$ we need to choose the optimal solution of the lower level problem (2d)-(2g) with the best value of the objective function (2a). This can be done by solving the following modified lower level problem in which we lexicographic optimize two objective functions. The first one refers to the preferences (as in problem (2d)-(2g)) and the second one refers to the allocating cost:

$$
\begin{aligned}
\operatorname{lex}\min_{x} \quad & \left(\sum_{i\in I}\sum_{j\in J} g_{ij}x_{ij}, \quad \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \right) \\
\text{subject to} \quad & \\
& \sum_{i\in I} x_{ij} = 1, \quad j \in J \\
& \sum_{j\in J} x_{ij} \leqslant q_i y_i, \quad i \in I \\
& x_{ij} \in \{0,1\}, \quad i \in I,\ j \in J
\end{aligned}
$$

Lexicographic optimization assumes that the objectives are ranked in order of importance

Table 1: Data of the illustrative examples

	f_i	$\hat{f}_i$	q_i	$\hat{q}_i$	c_{ij} R_1	R_2	R_3	R_4	R_5	R_6
F_1	5	5	2	2	5	1	6	10	9	10
F_2	7	7	3	3	3	4	4	6	8	8
F_3	7	7	3	3	5	3	3	9	6	5
F_4	5	25	2	6	9	1	5	8	5	2

	g_{ij}						$\hat{g}_{ij}$					
	R_1	R_2	R_3	R_4	R_5	R_6	R_1	R_2	R_3	R_4	R_5	R_6
F_1	4	1	2	1	1	3	1	2	2	1	3	2
F_2	1	2	4	3	2	1	3	1	1	3	1	1
F_3	3	4	1	4	4	2	2	3	3	2	2	3
F_4	2	3	3	2	3	4	4	4	4	4	4	4

and the objective functions are minimized one at a time in order of priority. Hence, the main criterion $\sum_{i\in I}\sum_{j\in J} g_{ij}x_{ij}$ is minimized first. Then, the second criterion $\sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij}$ is minimized subject to achieving the optimum with respect to the first criterion. If there are multiple optimal solutions to the lower level problem (first criterion), choosing among them the optimal solution of the second criterion guarantees that the best solution for the leader (optimistic approach) is selected. Notice that the term $\sum_{i\in I} f_i y_i$ has been suppressed because it is constant while solving the lower level problem.

Next we show main differences in modeling the C-SPLPO using formulations (1) and (2).

2.3. Highlighting the differences between C-SPLPO-1 and C-SPLPO-2

From our point of view, the main difficulty with formulation (1) is that it can be 'too restrictive' in the sense that this model reduces, even dramatically, the number of ways in which the selection of facility locations can be done, thus worsening the optimal objective function value. We explain this with the help of the following illustrative examples. Let us assume that there are four facilities, F_1, …, F_4, and six customers, R_1, …, R_6. First, we consider the costs f_i and c_{ij}, capacities q_i, and preferences g_{ij} provided in Table 1.

Figure 1 summarizes the results. It shows the elements of set S_y, i.e. the feasible ways of selecting facilities to allocate six customers, and the related customer allocations according to model C-SPLPO-1 (left-hand-side) and model C-SPLPO-2 (right-hand-side). Notice that model C-SPLPO-1 is not feasible. None of the selections allows every customer to be allocated to a facility in accordance with his/her preference ordering. For instance, in Figure 1a, facilities F_2 and F_3 are chosen to be open. This selection is not feasible because the capacity of F_2 is 3, and five customers should be allocated to it. The same happens with Figure 1c, in which facilities F_1, F_2 and F_3 are open. In this case, F_1 is preferred by three customers but it can serve at most two customers. The remaining selections are not feasible for analogous reasons.

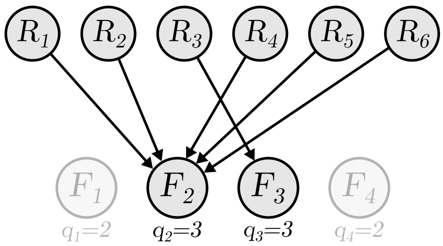
In contrast, the feasible region of the model C-SPLPO-2 consists of the six feasible solutions shown in the right-hand-side of Figure 1. In the optimal solution, facilities F_2 and F_3 are open, the total cost being 46. Customers R_1, R_2 and R_5 are allocated to facility F_2, whereas customers R_3, R_4 and R_6 are allocated to facility F_3. Notice also that customers R_1, R_2, R_3 and R_5 are allocated to their most preferred open facility.

Now, let us modify the previous example. Instead of fixed costs f_i, capacities q_i and preferences g_{ij}, we consider the values $\hat{f}_i$, $\hat{q}_i$ and $\hat{g}_{ij}$ provided in Table 1. In this case, the model C-SPLPO-1 has a single feasible solution, and so it is the optimal solution. Only facility F_4 is open, the total cost being 55. Obviously, all customers are allocated to F_4, which is the only open facility and thus their most preferred. However, it is worth pointing out that, according to preferences $\hat{g}_{ij}$, F_4 is the least preferred facility by all customers $\hat{g}_{4j} = 4, j = 1, \ldots, 6$. The feasible region of the model C-SPLPO-2 consists now of ten feasible solutions, and the optimal solution opens F_2 and F_3 at a cost of 50. Customers R_2, R_3 and R_6 are allocated to facility F_2, whereas customers R_1, R_4 and R_5 are allocated to facility F_3. All customers except customer R_5 are allocated to their most preferred open facility. Moreover, customers R_2, R_3 and R_6 are allocated to their most preferred facility ($\hat{g}_{22} = \hat{g}_{23} = \hat{g}_{26} = 1$), and customers R_1, R_4 and R_5 are allocated to their second most preferred facility ($\hat{g}_{31} = \hat{g}_{34} = \hat{g}_{35} = 2$). Notice also that the optimal objective function value of the C-SPLPO-2 is smaller than that of the C-SPLPO-1.

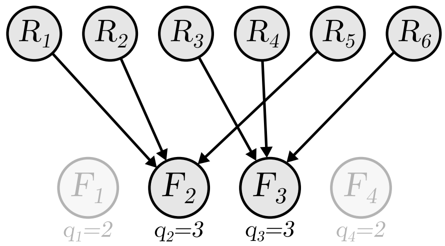
Given the above considerations, we can conclude that C-SPLPO-1 would be appropriate for modeling the C-SPLPO if it is compulsory to guarantee the individual preferences of every customer. Otherwise, if customer preferences are looked upon as a goal to be aimed at, model C-SPLPO-2 seems to be more suitable for the C-SPLPO. Hence, from this point on, when referring to this problem, we will refer to the bilevel formulation. Next, we focus on how to solve model C-SPLPO-2.

3. Reformulating the C-SPLPO-2 as a single level problem

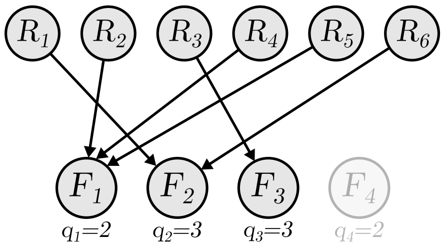
The C-SPLPO-2 is a bilevel integer optimization problem with binary variables at both levels. General bilevel integer models are very difficult to deal with [16, 23, 30]. However, by adequately managing the properties of the lower level problem (2d)-(2g), the C-SPLPO-2



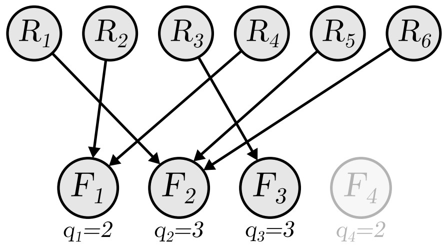
(a) Not feasible



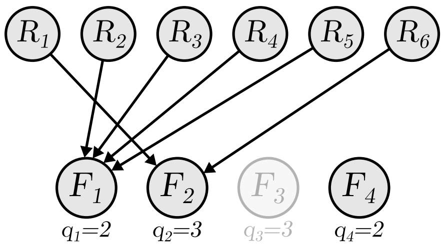
(b) Objective function = 46



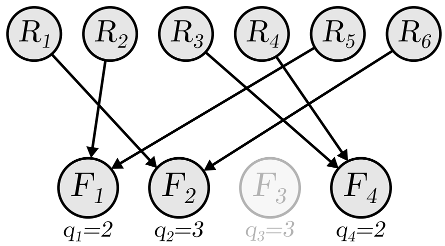
(c) Not feasible



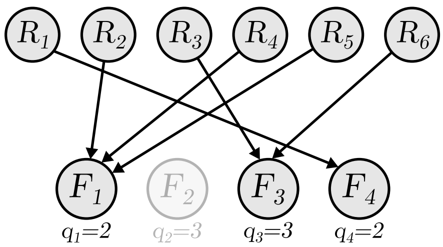
(d) Objective function = 52



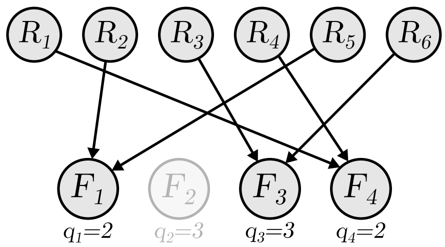
(e) Not feasible



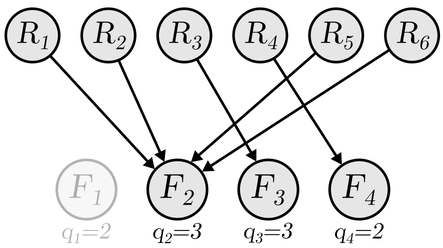
(f) Objective function = 51



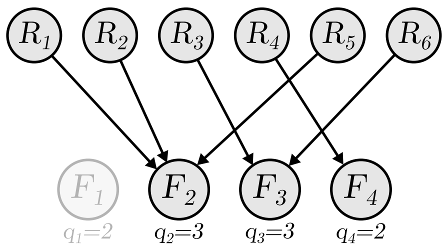
(g) Not feasible



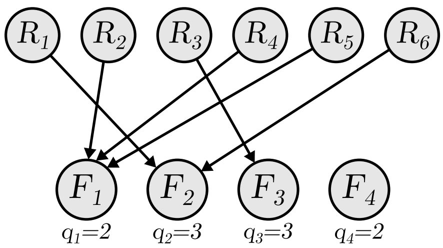
(h) Objective function = 52



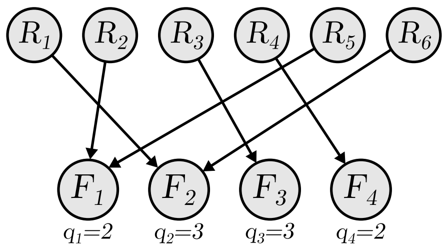
(i) Not feasible



(j) Objective function = 50



(k) Not feasible



(l) Objective function = 56

Figure 1: The six feasible ways of selecting facilities to allocate six customers and related customer allocations according to model C-SPLPO-1 (left-hand-side) and model C-SPLPO-2 (right-hand-side)

can be reformulated as a single level optimization problem.

One of the most frequent approaches to solve bilevel optimization problems consists in reformulating them as single level problems by replacing the lower level problem by its necessary and sufficient optimality conditions (when they exist). Then, standard optimization techniques can be applied for solving them. That reformulation usually involves additional variables as well as nonlinear terms that can be linearized at the cost of including binary variables. This approach can be applied to handle the C-SPLPO-2 after realizing that the coefficient matrix of the lower level problem is unimodular once the upper level variables are stated. Hence, it can be written as a linear optimization problem for which necessary and sufficient optimality conditions exists. Moreover, the proposed reformulation avoids the use of nonlinear terms, and consequently does not need to include additional binary variables.

For a given value of the upper level variables $\tilde{y} \in S_y$, let $I(\tilde{y}) = \{i \in I : \tilde{y}_i = 1\}$. Notice that, for each $j \in J$, $\tilde{x}_{ij} = 0$ for all $i \notin I(\tilde{y})$ due to constraint (2f). Moreover, the lower level problem can be treated as a transportation problem in which there are $|I(\tilde{y})|$ origin points (where $|I(\tilde{y})|$ stands for the cardinality of $I(\tilde{y})$), which are the open facilities, each with a supply q_i, $i \in I(\tilde{y})$, and the destination points are the customers, each one with a unit demand. Therefore, the binary constraint on the variable x_{ij} can be substituted by a nonnegativity constraint [3]. As a result, the lower level optimization problem can be stated as:

$$\min_{x} \quad \sum_{i \in I(\tilde{y})} \sum_{j \in J} g_{ij} x_{ij} \tag{3a}$$

subject to

$$\sum_{i \in I(\tilde{y})} x_{ij} = 1, \quad j \in J \tag{3b}$$

$$\sum_{j \in J} -x_{ij} \geqslant -q_i, \quad i \in I(\tilde{y}) \tag{3c}$$

$$x_{ij} \geqslant 0, \quad i \in I(\tilde{y}),\ j \in J \tag{3d}$$

The dual of problem (3) is:

$$\begin{aligned} \max_{u,v} \quad & \sum_{j \in J} u_j - \sum_{i \in I(\tilde{y})} q_i v_i \\ \text{subject to} \quad & \\ & u_j - v_i \leqslant g_{ij}, \quad i \in I(\tilde{y}),\ j \in J \\ & v_i \geqslant 0, \quad i \in I(\tilde{y}) \end{aligned} \tag{4}$$

where $\{u_j\}_{j \in J}$ are the dual variables associated with constraints (3b), and $\{v_i\}_{i \in I(\tilde{y})}$ are the dual variables associated with constraints (3c).

Since problem (3) has an optimal solution, so does the dual problem, and both optimal objective function values coincide. Therefore, by applying duality theory, $\{\tilde{x}_{ij}, \tilde{u}_j, \tilde{v}_i\}_{i\in I(\tilde{y}), j\in J}$ are optimal solutions, respectively, of problem (3) and its dual (4) if and only if:

$$\sum_{i\in I(\tilde{y})} \sum_{j\in J} g_{ij}\tilde{x}_{ij} = \sum_{j\in J} \tilde{u}_j - \sum_{i\in I(\tilde{y})} q_i\tilde{v}_i$$

$$\sum_{i\in I(\tilde{y})} \tilde{x}_{ij} = 1, \quad j \in J$$

$$\sum_{j\in J} \tilde{x}_{ij} \leqslant q_i, \quad i \in I(\tilde{y})$$

$$\tilde{u}_j - \tilde{v}_i \leqslant g_{ij}, \quad i \in I(\tilde{y}),\ j \in J$$

$$\tilde{x}_{ij} \geqslant 0, \quad i \in I(\tilde{y}),\ j \in J$$

$$\tilde{v}_i \geqslant 0, \quad i \in I(\tilde{y})$$

Therefore, the C-SPLPO-2 can be stated as the following single level mixed integer linear optimization problem:

$$\min_{y,x,u,v} \quad \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \tag{5a}$$

subject to

$$\sum_{i\in I} q_i y_i \geqslant m \tag{5b}$$

$$\sum_{i\in I} x_{ij} = 1, \quad j \in J \tag{5c}$$

$$\sum_{j\in J} x_{ij} \leqslant q_i y_i, \quad i \in I \tag{5d}$$

$$u_j - v_i \leqslant g_{ij} + M(1 - y_i), \quad i \in I,\ j \in J \tag{5e}$$

$$v_i \leqslant M y_i, \quad i \in I \tag{5f}$$

$$\sum_{i\in I}\sum_{j\in J} g_{ij}x_{ij} = \sum_{j\in J} u_j - \sum_{i\in I} q_i v_i \tag{5g}$$

$$y_i \in \{0, 1\}, \quad i \in I \tag{5h}$$

$$x_{ij} \geqslant 0, \quad i \in I,\ j \in J \tag{5i}$$

$$v_i \geqslant 0, \quad i \in I \tag{5j}$$

where M is a constant big enough to guarantee that constraints (5e) are restrictive only when $y_i = 1$, i.e. the facility is open, and constraints (5f) restrict the value of v_i only if the facility is closed. The importance of selecting an appropriate value of M has been recognized in [20]. In this paper, taking into account the properties of the transportation problem, the value of M can be bounded. Indeed, in the transportation problem, multipliers u_j, v_i are computed as sums and differences of g_{ij} corresponding to basic variables, so they are bounded by $n(n+m)$. In order to avoid possible round-off computational error problems associated with a large value of M, as well as tightening the constraints (5e) and (5f), in the next Theorem we derive upper bounds on the value of the dual variables.

Theorem 1. *For a given value of the upper level variables $\tilde{y} \in S_y$, let x^* be an optimal solution of the lower level problem (3) and (u^*, v^*) be an optimal solution of the dual problem associated with the lower level (4). Then, there exists an optimal solution of the dual problem $(\hat{u}, \hat{v})$ so that there exists $i_0 \in I(\tilde{y})$ such that $\hat{v}_{i_0} = 0$. Moreover, $\hat{v}_i \leqslant n$, for all $i \in I(\tilde{y})$, and $\hat{u}_j \leqslant n$, for all $j \in J$.*

Proof. If there exists $i_0 \in I(\tilde{y})$ such that $v^*_{i_0} = 0$, we take $(\hat{u}, \hat{v}) = (u^*, v^*)$ and the result is at hand.

Otherwise, $v^*_i > 0$, for all $i \in I(\tilde{y})$. Then, by applying the complementary slackness conditions,

$$\sum_{j \in J} x^*_{ij} = q_i, \qquad i \in I(\tilde{y})$$

Let $k = \min_{i \in I(\tilde{y})}\{v^*_i\} > 0$, and $i_0 = \arg\min_{i \in I(\tilde{y})}\{v^*_i\}$. We define

$$\hat{u}_j = u^*_j - k, \;\; j \in J \qquad \hat{v}_i = v^*_i - k, \;\; i \in I(\tilde{y})$$

Notice that $(\hat{u}, \hat{v})$ is a feasible solution of problem (4):

$$\hat{u}_j - \hat{v}_i = u^*_j - v^*_i \leqslant g_{ij}, \qquad i \in I(\tilde{y}), \;\; j \in J$$

$$\hat{v}_i \geqslant 0, \qquad i \in I(\tilde{y})$$

Moreover,

$$\sum_{j \in J} \hat{u}_j - \sum_{i \in I(\tilde{y})} q_i \hat{v}_i = \sum_{j \in J} u^*_j - mk - \sum_{i \in I(\tilde{y})} q_i v^*_i + k \sum_{i \in I(\tilde{y})} q_i = \sum_{j \in J} u^*_j - \sum_{i \in I(\tilde{y})} q_i v^*_i + k \left(\sum_{i \in I(\tilde{y})} q_i - m \right)$$

On the other hand,

$$\sum_{i \in I(\tilde{y})} q_i = \sum_{i \in I(\tilde{y})} \sum_{j \in J} x^*_{ij} = \sum_{j \in J} \sum_{i \in I(\tilde{y})} x^*_{ij} = \sum_{j \in J} 1 = m$$

Thus, $\sum_{i\in I(\tilde{y})} q_i - m = 0$ and $(\hat{u},\hat{v})$ is an optimal solution of the dual problem (4). For this optimal solution, $\hat{v}_{i_0} = 0$, and the first part of the Theorem follows. Furthermore, for the facility i_0, $\hat{u}_j - \hat{v}_{i_0} \leqslant g_{i_0 j}$, $j \in J$. Hence,

$$\hat{u}_j \leqslant g_{i_0 j} \leqslant n, \; j \in J.$$

Finally, let $i \in I(\tilde{y})$. If $x^*_{ij} = 0$, for all $j \in J$, i.e. no customer is allocated to this facility, then $\sum_{j\in J} x^*_{ij} = 0 < q_i$, and thus $\hat{v}_i = 0$. Otherwise, let $j_0 \in J$ such that $x^*_{ij_0} = 1$. By the complementary slackness conditions, $\hat{u}_{j_0} - \hat{v}_i = g_{ij_0}$. Therefore, $\hat{v}_i = \hat{u}_{j_0} - g_{ij_0} \leqslant n$. □

Corollary 2. *For every facility $i \in I$ and customer $j \in J$, the number of facilities n is a valid constant M for constraints (5e) and (5f).*

As mentioned above the reformulation (5) does not involve additional binary variables. Note that the classical reformulation of a bilevel problem into a single level optimization model using KKT conditions would need to include a binary variable and two additional constraints for linearizing each product constraint. On the other hand, model (5) is a mixed integer optimization problem and thus can be solved by using standard optimization techniques. This approach is in general useful for solving small and medium-size problems, but it may not be competitive for large problems. Therefore, in the next section we will develop CLOA (Capacitated Location Ordering Algorithm), a matheuristic which proves to be quite efficient according to the extensive computational experiment carried out.

4. CLOA: A matheuristic to solve the C-SPLPO-2

CLOA is a matheuristic which combines the framework of an evolutionary algorithm with the lexicographic optimization of a transportation problem aiming to provide good feasible solutions for C-SPLPO-2. Evolutionary algorithms [1, 10, 24] have been increasingly applied to solve different kinds of optimization problems, especially combinatorial optimization problems, because they are able to provide good solutions to complex problems in reasonable computational time. A key point when designing evolutionary algorithms is to identify good convenient manners of encoding solutions as chromosomes. As in biological evolution, an evolutionary algorithm consists of a population of chromosomes which evolves to create offspring. Each chromosome is given a fitness value which measures its quality. Some parents and offspring are selected in accordance with a preset criterion based on the fitness function to survive to the next population. The aim is to guide the population to include better chromosomes through the generations. The algorithm proceeds through successive iterations until the stopping condition is met. The solution associated with the best chromosome is provided as the solution of the problem considered.

CLOA uses the general structure of an evolutionary algorithm, but also takes into account the characteristics of the bilevel model to use compact chromosomes and optimization to provide bilevel feasible solutions. Next we explain the main characteristics of the matheuristic developed.

4.1. Chromosome encoding, feasible solution construction and fitness evaluation

Let (y, x) be a bilevel feasible solution. The purpose is to encode its information in a compact chromosome. Note that x is completely determined after knowing the value of y, since x is the optimal solution of the corresponding lower level problem which provides the best value of the upper level objective function. Therefore, we encode each chromosome C as a binary n-dimensional vector which provides the value of variables y. That is, the components of the chromosome indicate if the corresponding facility is open or not:

$$C_i = \begin{cases} 1, & \text{if facility } i \text{ is open} \\ 0, & \text{otherwise} \end{cases}, \quad i \in I$$

Let C be a chromosome. First, its feasibility in terms of being able to allocate all the customers is checked. If $\sum_{i\in I} q_i C_i < m$ the chromosome is repaired. For this purpose, as many times as needed to achieve $\sum_{i\in I} q_i C_i \geqslant m$, a gene $C_j = 0$ is randomly selected and switched.

After repairing the chromosome C (if needed), its associated bilevel feasible solution (y, x) is computed. From a chromosome C, the value of variables y is directly obtained, $y_i = C_i$, $i \in I$. Moreover, $x_{ij} = 0$ for all $j \in J$ and $i \notin I(y)$. The value of the remaining variables x is obtained by solving the lexicographic optimization problem:

$$\begin{aligned} \operatorname*{lex\,min}_{x} \quad & \left(\sum_{i\in I(y)} \sum_{j\in J} g_{ij} x_{ij}, \quad \sum_{i\in I(y)} \sum_{j\in J} c_{ij} x_{ij} \right) \\ \text{subject to} \quad & \\ & \sum_{i\in I(y)} x_{ij} = 1, \quad j \in J \\ & \sum_{j\in J} x_{ij} \leqslant q_i, \quad i \in I(y) \\ & x_{ij} \geqslant 0, \quad i \in I(y),\ j \in J \end{aligned} \tag{6}$$

This problem can be solved using standard techniques, i.e. first solving the problem with the lower level objective function $\sum_{i\in I(y)} \sum_{j\in J} g_{ij} x_{ij}$. After solving this problem, the second objective function $\sum_{i\in I(y)}\sum_{j\in J} c_{ij}x_{ij}$ is considered, and a constraint is included in which $\sum_{i\in I(y)}\sum_{j\in J} g_{ij}x_{ij}$ equals the optimal value of the first problem. An optimal solution of this second problem is an optimal solution of the lexicographic problem (6) [15]. However, if this approach is used, the problem solved in the second place lose the transportation structure. To maintain this structure and take advantage of the efficiency of the transportation algorithm, we propose to use a similar approach to that developed in [4, 5]. Based on this, when solving problem (6) both objective functions are simultaneously considered. Thus, a bidimensional vector of reduced costs is associated with each variable. The first component is the reduced cost with respect to the first objective function $\sum_{i\in I(y)}\sum_{j\in J} g_{ij}x_{ij}$, computed in the usual way for transportation problems. The second component is the reduced cost with respect to the second objective function $\sum_{i\in I(y)}\sum_{j\in J} c_{ij}x_{ij}$, computed in the same way. Then, reduced costs are checked in accordance with their lexicographic character. If they are all lexicographically nonnegative, i.e. the first nonzero component is nonnegative, the lexicographic optimization problem has reached its optimal solution; otherwise, the variable having the lexicographically smallest reduced cost vector is selected to enter the basis and an iteration of the usual transportation algorithm is applied.

After computing (y, x), a final check is made about the facilities which actually need to be open. If $C_i = 1$, equivalently $y_i = 1$, but $x_{ij} = 0$ for all $j \in J$, then this facility has no customers allocated. Therefore, it can be closed. This update is made by switching $C_i = 0$, and $y_i = 0$.

Then, the fitness of the updated (if needed) chromosome C is defined as the value of the upper level objective function of the bilevel feasible solution (y, x) associated with the chromosome, i.e. the objective function value of the C-SPLPO-2:

$$fitness(C) = \sum_{i\in I} f_i y_i \; + \; \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \tag{7}$$

4.2. The initial population

The algorithm handles populations, denoted *Pop*, of size p_{size}. All the chromosomes in *Pop* are distinct. The initial population is formed with p_{size} randomly generated chromosomes. Initially $Pop = \emptyset$, and successive non repeated chromosomes are added to *Pop* as they are created. For the purpose of favoring diversification, first a random number $p \in [0, 1]$ is generated for each chromosome. Then, for $i \in I$ a random number $rn_i \in [0, 1]$ is selected. If $rn_i \leqslant p$, then $C_i = 1$, i.e. the facility is open. Otherwise, $C_i = 0$. After this process, the chromosome is checked for feasibility and repaired (if needed) as explained in section 4.1. If the resulting chromosome is already in *Pop*, it is rejected, and the process of creating a new

Input

The current population of chromosomes, Pop

A new chromosome $C = (C_1, \ldots, C_n)$

Procedure

While $\sum_{i \in I} q_i C_i < m$,

  Randomly select $i \in I$ such that $C_i = 0$

  Set $C_i = 1$.

If $C = (C_1, \ldots, C_n) \in Pop$ reject C. Stop.

Let $y_i = C_i$, $i \in I$.

Let $x_{ij} = 0$, $j \in J$, $i \notin I(y)$.

Solve the lexicographic problem (6).

Let (y, x) the bilevel feasible solution.

While $C_i = 1$ and $x_{ij} = 0$, for all $j \in J$,

  Set $C_i = 0$ and $y_i = 0$

If $C = (C_1, \ldots, C_n) \in Pop$ reject C. Stop.

Compute $fitness(C) = \sum_{i \in I} f_i y_i \ + \ \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$

Figure 2: Procedure for repairing a chromosome and computing its fitness

chromosome starts again.

After accepting the chromosome, its associated bilevel feasible solution (y, x) is computed and the final check is made as explained in section 4.1. Finally, the chromosome is checked again to see whether it is in Pop and it is rejected if this is the case, starting again the process of creating a new chromosome.

The whole procedure of repairing a chromosome, if needed, rejecting it, computing its associated bilevel feasible solution, and computing its fitness is displayed in Figure 2. This routine is applied not only to the chromosomes in the initial population but to every chromosome which is generated when the algorithm proceeds.

4.3. Crossover, mutation and survivor selection

Using the crossover and mutation operations the algorithm constructs offspring, i.e. new chromosomes which are potential members of the next population. From the parent population, each chromosome C^r, $r = 1, \ldots, p_{size}$ is checked to undergo crossover with probability p_c. If C^r is selected, the single point crossover operation is applied. That is, a chromosome from the current population (other than chromosome C^r) and a crossover point are randomly selected. Two offspring are created by combining the parents at the crossover point, i.e. all genes beyond the crossover point in either parent are swapped between both parents. In the following example, the crossover point is after the 5th gene:

Parent 1:	$(1, 0, 1, 1, 1, \mid 1, 0, 0)$	Offspring 1:	$(1, 0, 1, 1, 1, 0, 0, 1)$
Parent 2:	$(0, 0, 1, 0, 1, \mid 0, 0, 1)$	Offspring 2:	$(0, 0, 1, 0, 1, 1, 0, 0)$

Then, each offspring is selected for the mutation operation with probability p_m. If a chromosome is selected to undergo mutation, an integer number in $\{1, \ldots, n\}$ is randomly generated and the corresponding gene is switched, i.e. the facility is changed to close if it was open, and is changed to open if it was closed.

After crossover and mutation, the procedure of repairing a chromosome and computing its fitness shown in Figure 2 is applied. From the complete set of distinct chromosomes formed by the current population and the non repeated offspring, the best p_{size} chromosomes with respect to the fitness function are kept for the next population Pop (elitist survivor selection). The algorithm iterates until a termination condition is met. In the implementation of the algorithm, computing time has been chosen as the stopping criterion. Upon termination, the bilevel feasible solution associated with the chromosome which has the least fitness value is provided as the solution of the C-SPLPO-2.

5. Computational study

This section is devoted to presenting and discussing the computational experiments carried out. The numerical experiments have been performed on a PC Intel Core i7-6700 with 3.4 gigahertz, 32.0 gigabyte of RAM and Windows 10 64-bit as the operating system. The algorithm CLOA has been coded in Dev-C++ 5.11 under C++ language.

Since the C-SPLPO-2 has not been previously studied, no benchmark instances are available. Therefore, we decided to adapt two groups of instances which have been used as benchmark instances for the capacitated facility location problem. The first group of instances is described by Holmberg et al. [19], and can be downloaded from `http://www.di.unipi.it/optimize/`. It comprises four sets of randomly generated test instances (sets S_1, S_2, S_3 and S_5), and one more set based on vehicle routing problems used by Solomon [29] (set S_4), each with different sizes and properties. The second group of instances is composed of a subset of 20 instances of those described by Avella and Boccia [2], which are publicly available at `http://www.ing.unisannio.it/boccia/CFLP.htm`. They are organized into five subsets in accordance with their size. The first instance in each category according to the classification in [2] has been selected. Table 2 summarizes the instance sizes. Altogether

Table 2: The sizes of the test instances.

Set	Subset	Instances	# of facilities (n)	# of customers (m)
Holmberg et al.	S_1	P_1 - P_{12}	10	50
	S_2	P_{13} - P_{24}	20	50
	S_3	P_{25} - P_{40}	30	150
	S_4	P_{41} - P_{55}	10 - 30	70 - 100
	S_5	P_{56} - P_{71}	30	200
Avella and Boccia	S_6	i300-1, -6, -11, -16	300	300
	S_7	i3001500-1, -6, -11, -16	300	1500
	S_8	i500-1, -6, -11, -16	500	500
	S_9	i700-1, -6, -11, -16	700	700
	S_{10}	i1000-1, -6, -11, -16	1000	1000

91 instances have been tested, ranging from small-size (10 facilities and 50 customers) to very large-size (1000 facilities and 1000 customers). For all the instances, we have maintained the location of potential facilities and customers, as well as the cost c_{ij}. Moreover, we assigned a capacity q_i to the facility i as $q_i = \lceil Q_i/\overline{d} \rceil$ where $\lceil . \rceil$ denotes the ceiling function, Q_i is the original capacity, and $\overline{d}$ is the average of the original customer demands. To assign the preferences, we have applied the procedure proposed by Cánovas et al. [7], which generates random preferences but maintains some rationality with respect to the allocation costs. These authors propose to generate fake costs $\tilde{c}_{ij}$, for each pair (i, j), using a triangular distribution defined on the interval $[m_j, M_j]$, where $m_j = \min\{c_{ij} : i \in I\}$ and $M_j = \max\{c_{ij} : i \in I\}$, with c_{ij} as the peak of the distribution. After ordering the fake costs for each customer j, the facility i_1 with the lowest value $\tilde{c}_{i_1 j}$ is the most preferred facility of the customer j and so on until the least preferred facility is reached which corresponds to the facility with the highest fake cost.

In the following subsections, we present the results of the computational experiment. First, we have analyzed the impact of the population size and the crossover and mutation probabilities on the quality of the solution provided by CLOA. Based on this study, we have selected the value of those parameters. Then, we have compared the results provided by CLOA with this selection and the optimal solution (when available) provided by CPLEX. Finally, using the same instances, we have solved models C-SPLPO-1 and C-SPLPO-2, together with the single level variant of the problem without customer preferences (the so-called relaxed problem in bilevel optimization) to show that they are structurally different.

5.1. Selecting the configuration of CLOA

As mentioned above, the purpose of the first part of the computational study is to assess the influence of the value of the population size and the crossover and mutation probabilities

Table 3: Algorithm configurations.

Configuration	p_{size}	p_c	p_m
cfg_1	50	0.5	0.5
cfg_2	50	0.5	0.9
cfg_3	50	0.9	0.5
cfg_4	50	0.9	0.9
cfg_5	100	0.5	0.5
cfg_6	100	0.5	0.9
cfg_7	100	0.9	0.5
cfg_8	100	0.9	0.9

based on the results of a 2^3 factorial design. The factors and levels considered are: population size ($p_{size} = 50$, $p_{size} = 100$), crossover probability ($p_c = 0.5$, $p_c = 0.9$), and mutation probability ($p_m = 0.5$, $p_m = 0.9$). Table 3 displays the eight configurations of the algorithm. Each of the test instances has been solved five times under each algorithm configuration, 40 times in total. The termination condition of the algorithm has been established in terms of computing time. Sets S_1, S_2 and S_4 are given 1 second of computing time; set S_3, 5 seconds; set S_5, 10 seconds; and sets S_6 to S_{10}, 300 seconds. The statistical analysis has been carried out using Minitab®, release 17.

In order to select the best configuration, for each instance we compute CLOA_{best}, the best value of the objective function of problem C-SPLPO-2 obtained in the 40 runs of the instance. Then, for a particular instance and run, we consider a success to be when its objective function value equals CLOA_{best}. All the runs of all the instances in sets S_1, S_2, S_3 and S_4 provide the best value CLOA_{best}. Therefore, to select the best configuration we have applied an analysis of variance separately to set S_5, and sets S_6 to S_{10} together. The results for set S_5 indicate that p_{size} (26.12% of variability explained) and p_m (34.78% of variability explained) are significant factors. For the sets S_6 to S_{10}, the same factors are significant. The parameter p_{size} explains 47.61% of the variability, whereas p_m explains 34.08% of the variability. No interaction is significant in either case. In both cases, the effect is negative in the sense that the greater the value, the lower the mean of success, that is the percentage of success for the configuration. For each of these sets of instances, we have also computed the percentage of gap defined as the value of the objective function of problem (2) in the instance run minus CLOA_{best}, divided by CLOA_{best}, and multiplied by 100 to get a percentage. As an illustration, Figure 3 shows the minimum, the average and the maximum of the percentage of gap for every configuration in sets S_5 to S_{10}. Set S_5 contains 16 instances, whereas sets S_6 to S_{10} contain 4 instances each. We see that, as expected, when the size of the problem increases, the percentage of gap and its variability

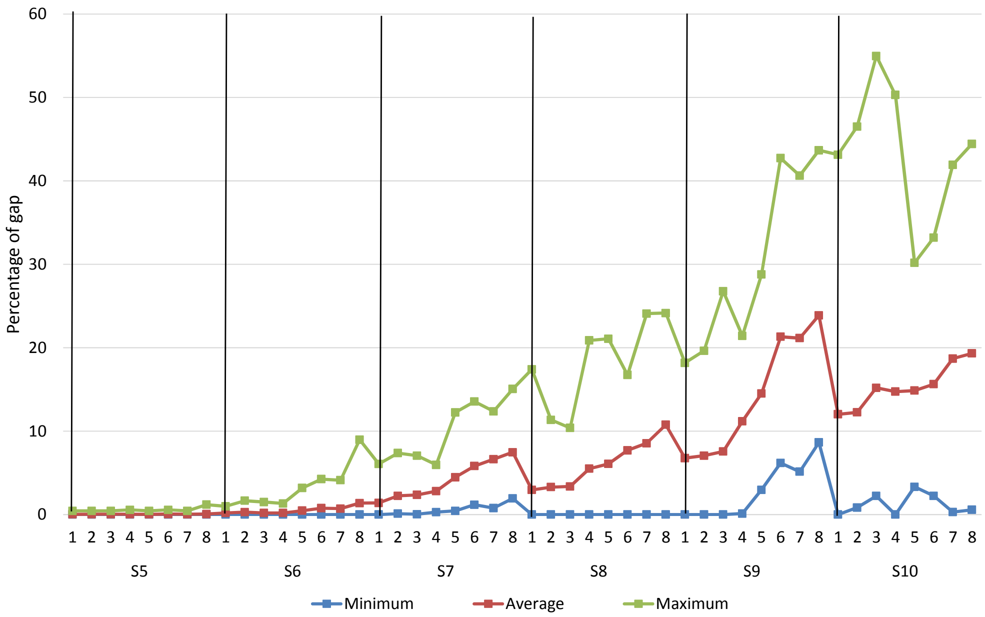


Figure 3: Percentage of gap for every configuration of CLOA in sets S_5 to S_{10}

also increase. Moreover, for most problems a configuration having $p_{size} = 50$ is better. As a result of the statistical analysis, we have selected for CLOA the configuration cfg$_1$ in which $p_{size} = 50$, $p_c = 0.5$ and $p_m = 0.5$. From now on, when we refer to CLOA we assume these values for the parameters.

5.2. Measuring the quality of CLOA

Next, our purpose is to measure the quality of CLOA by comparing the results provided by the selected configuration of CLOA with the optimal solution (or the best known feasible solution) of problem (5) provided by IBM ILOG CPLEX 12.9.0. For this purpose we have considered two variants of the CPLEX settings. In the first one, called CPLEX-1, the default settings are implemented. In the second one, called CPLEX-2, the search strategy is changed from the best-bound to the depth-first strategy. The CLOA stopping criterion has been set to 1 second of computing time for sets S_1, S_2 and S_4, 5 seconds for set S_3, 10 seconds for set S_5, and 1800 seconds for sets S_6 to S_{10}. Each of the test instances has been solved five times. The CPLEX stopping criterion was set at 7200 seconds. When the run is interrupted before providing the optimal solution, the best solution at this time is saved and the letters TL are written when the computing time is displayed. According to Corollary 2, the value of M has been set to n.

We separately analyze results of the small and large instances. Table 4 displays the results for the instances based on Holmberg et al. instances (sets S_1 to S_5). The first and second columns show the instance set and instance name, respectively. The third to seventh columns display the information provided by CPLEX-1: Obj$_1$ is the objective function value of the optimal solution (best feasible solution if the run is interrupted); Gap_1 is the relative gap; $\#\text{nodes}_1$ is the number of branching nodes; T_{best1} is the time at which CPLEX finds the reported solution; and T_{tot1} is the total required CPU time. The eighth to twelfth columns display the same information but corresponding to CPLEX-2. When Obj_2 is equal to Obj_1 a symbol '=' is written in the column of Obj_2. The thirteenth to sixteenth columns in Table 4 display the results provided by CLOA: Obj_{min} and Obj_{max} are the minimum and the maximum objective function values obtained in the five runs; $\overline{\text{T}}_{best}$ is the average of the CPU time at which CLOA finds the reported solution in the five runs; and T_{tot} is the total CPU time assigned to each run. If Obj_{min} coincides with the best value provided by CPLEX, i.e. $\text{Obj}_{min} = \min\{\text{Obj}_1, \text{Obj}_2\}$, a symbol '=' is written in the column of Obj_{min}. When Obj_{min} and Obj_{max} coincide, a symbol '=' is also written in the column of Obj_{max}. Moreover, in each row, the smallest value of the objective function is written in bold. All times are in seconds.

Both variants of CPLEX are only able to solve to optimality 51 out of the 55 instances of sets S_1 to S_4 and none of set S_5. CPLEX-1 and CPLEX-2 provide the same objective function value in 64 instances out of the 71 instances. For the remaining seven instances, CPLEX-1 provides a better value than CPLEX-2 in four instances (all of them in set S_5). For all the 51 instances for which CPLEX provides the optimal solution, CLOA obtains it as well (instances P_1 to P_{28}, P_{33} to P_{55}). Moreover, CLOA yields the same objective function value in these instances in all the five runs (Obj_{min} and Obj_{max} are equal). For the remaining 20 instances, CLOA provides at least the same objective function value than the best value of CPLEX in all the five runs, except for instance P58 where in just one out of the five runs the objective function value is 1.001 times greater than the best value provided by CPLEX. Moreover, for instance P59 in two out of the five runs and for instance P68 in four out of the five runs, the objective function value provided by CLOA strictly improves the best value of CPLEX. It is worth remarking that CLOA provides a value better than or equal to the best value provided by CPLEX in 354 out of the 355 runs of the experiment.

Table 5 is similar to Table 4 and presents the results for the instances based on Avella and Boccia instances (sets S_6 to S_{10}). The first and second columns show the instance set and instance name, respectively. For the variants of CPLEX we have not included the column T_{tot} since CPLEX was not able to solve any instance of these sets within the computing time prescribed (7200 seconds). For CLOA, we have included two more columns which include the relative Gap defined as:

$$\text{Min}_{Gap} = \frac{\text{Obj}_{min} - \min\{\text{Obj}_1, \text{Obj}_2\}}{\min\{\text{Obj}_1, \text{Obj}_2\}}\,, \qquad \text{Max}_{Gap} = \frac{\text{Obj}_{max} - \min\{\text{Obj}_1, \text{Obj}_2\}}{\min\{\text{Obj}_1, \text{Obj}_2\}}$$

Concerning CPLEX performance, in instances i300-11, i500-6, i500-16, i700-16, i1000-11 and i1000-16 both variants of CPLEX provide the same value of the objective function (6 out of the 20 instances). For the remaining 14 instances, CPLEX-1 is better than CPLEX-2 in ten instances and worse in the other four. As could be expected, there is more variability in the results provided by CLOA in these sets than in sets S_1 to S_5. Only in 5 out of the 20 instances does CLOA provide the same value in the five runs. For the remaining instances Obj_{max} is lower than 1.078 times Obj_{min}, except for instance i1000-6, for which it is 1.279 times the minimum value. If we look at the best solution provided by CLOA, we see that Obj_{min} is better than the best solution provided by CPLEX in 11 out of the 20 instances (all the instances of set S_7, three out of the four instances of sets S_8 and S_9 and one instance of set S_{10}), it is equal in the four instances of set S_6, and it is worse in the remaining five instances. Moreover, for 10 out of the 11 instances for which the best solution provided by CLOA is strictly better than the best solution provided by CPLEX (Min_{Gap} is negative) the worst solution provided by CLOA, i.e. Obj_{max}, is also better (Max_{Gap} is negative). Hence, in these 10 instances CLOA is better than CPLEX in all the five runs. Moreover, in the remaining instance i700-6, CLOA is better than CPLEX-2 in all the five runs and is better than CPLEX-1 in three out of the five runs.

Finally, we analyze the total computing times. Except for the smaller instances P_1 to P_{12} (set S_1) and for instances P_{41}, P_{44}, P_{47}, P_{50}, P_{52}, and P_{54} (set S_4), whose CPLEX CPU times are less than 1 second compared to CLOA CPU times of 1 second, the CPU times of CLOA are noticeably shorter for the remaining 73 out of 91 instances. If we analyze by sets, in set S_2 CLOA uses 1 second, whereas the average time for CPLEX-1 is 56.41 seconds and for CPLEX-2 it is 61.83 seconds. For instances in set S_3, CLOA uses 5 seconds, whereas variants CPLEX-1 and CPLEX-2 were interrupted after 7200 seconds in 4 out of the 16 instances and the average time of the remaining 12 instances is 2096.31 seconds and 1173.63 seconds, respectively. In set S_4, CLOA is given 1 second, whereas the average time for CPLEX-1 is 46.60 seconds and for CPLEX-2 it is 53.84 seconds. For the remaining sets, CPLEX was interrupted after 7200 seconds whereas CLOA is given 10 seconds for instances in set S_5, and 1800 seconds for sets S_6 to S_{10}. Regarding the times at which CPLEX and CLOA find the reported solution, $\overline{\text{T}}_{best} < \min\{\text{T}_{best1}, \text{T}_{best2}\}$ for all the instances except instance i3001500-6. Moreover, while $\min\{\text{T}_{best1}, \text{T}_{best2}\}$ ranges from 0.11 to 7044.45, $\overline{\text{T}}_{best}$ ranges from 0.01 to 1811.76.

Table 4: Computational results for sets S_1 to S_5. Termination condition is 7200 seconds for CPLEX. Times in seconds

Set	Inst	CPLEX-1					CPLEX-2					CLOA			
		Obj_1	Gap_1	$\sharp nodes_1$	T_{best1}	T_{tot1}	Obj_2	Gap_2	$\sharp nodes_2$	T_{best2}	T_{tot2}	Obj_{min}	Obj_{max}	$\overline{T}_{best}$	T_{tot}
S_1	P_1	**18592**	0.000	0	0.20	0.21	=	0.000	0	0.17	0.18	=	=	0.02	1
	P_2	**17658**	0.000	0	0.15	0.15	=	0.000	0	0.17	0.18	=	=	0.02	1
	P_3	**19058**	0.000	0	0.17	0.18	=	0.000	426	0.19	0.35	=	=	0.02	1
	P_4	**20442**	0.000	0	0.23	0.23	=	0.000	451	0.22	0.48	=	=	0.02	1
	P_5	**18552**	0.000	0	0.13	0.13	=	0.000	0	0.11	0.12	=	=	0.01	1
	P_6	**17806**	0.000	0	0.14	0.15	=	0.000	0	0.14	0.15	=	=	0.01	1
	P_7	**19206**	0.000	0	0.11	0.11	=	0.000	0	0.11	0.12	=	=	0.01	1
	P_8	**20606**	0.000	0	0.11	0.11	=	0.000	0	0.11	0.11	=	=	0.01	1
	P_9	**17651**	0.000	0	0.23	0.23	=	0.000	0	0.16	0.26	=	=	0.02	1
	P_{10}	**17146**	0.000	0	0.14	0.20	=	0.000	690	0.19	0.40	=	=	0.02	1
	P_{11}	**18146**	0.000	680	0.17	0.54	=	0.000	749	0.16	0.36	=	=	0.02	1
	P_{12}	**19146**	0.000	678	0.16	0.37	=	0.000	688	0.17	0.37	=	=	0.02	1
S_2	P_{13}	**17745**	0.000	197415	12.66	71.17	=	0.000	269079	23.78	83.26	=	=	0.11	1
	P_{14}	**16720**	0.000	225374	65.72	90.06	=	0.000	357709	17.98	93.48	=	=	0.14	1
	P_{15}	**18120**	0.000	178772	22.63	75.53	=	0.000	280890	18.19	83.19	=	=	0.14	1
	P_{16}	**19427**	0.000	159386	2.28	73.19	=	0.000	224740	14.00	69.41	=	=	0.10	1
	P_{17}	**17613**	0.000	146354	9.28	47.12	=	0.000	207243	15.75	53.68	=	=	0.15	1
	P_{18}	**16718**	0.000	175675	10.64	55.67	=	0.000	281857	28.14	60.76	=	=	0.13	1
	P_{19}	**18118**	0.000	150527	22.98	48.38	=	0.000	218738	31.84	56.07	=	=	0.15	1
	P_{20}	**19518**	0.000	143839	43.81	55.28	=	0.000	197603	22.92	54.80	=	=	0.14	1
	P_{21}	**17253**	0.000	119591	15.67	37.14	=	0.000	184091	29.14	45.88	=	=	0.19	1
	P_{22}	**16407**	0.000	153053	28.97	48.97	=	0.000	251974	2.86	54.95	=	=	0.22	1
	P_{23}	**17607**	0.000	106123	10.83	41.27	=	0.000	166802	37.11	45.38	=	=	0.15	1
	P_{24}	**18807**	0.000	93649	1.61	33.14	=	0.000	144218	0.94	41.16	=	=	0.19	1
S_3	P_{25}	**40164**	0.000	1274873	35.11	1086.17	=	0.000	750081	501.58	1220.14	=	=	0.53	5
	P_{26}	**39266**	0.000	1323237	854.78	4068.63	=	0.000	896675	131.80	1254.66	=	=	0.60	5
	P_{27}	**40866**	0.000	1144976	87.50	1044.05	=	0.000	884447	162.63	1300.21	=	=	0.64	5
	P_{28}	**42466**	0.000	1104449	1404.19	3880.24	=	0.000	774584	307.13	1249.26	=	=	0.54	5
	P_{29}	**44715**	0.325	3594968	928.16	TL	=	0.672	3580431	299.92	TL	=	=	0.76	5
	P_{30}	**43480**	0.445	1626961	6670.91	TL	=	0.688	3503469	2412.56	TL	=	=	1.05	5
	P_{31}	**45480**	0.492	861208	7107.77	TL	=	0.661	3681348	1037.34	TL	=	=	1.03	5
	P_{32}	47596	0.460	947452	6554.44	TL	**47480**	0.634	3158375	3113.63	TL	=	=	1.04	5
	P_{33}	**40428**	0.000	1700967	136.91	4907.95	=	0.000	1057463	267.02	1880.30	=	=	0.55	5
	P_{34}	**39517**	0.000	1640881	1214.72	1702.77	=	0.000	1161190	246.86	2148.97	=	=	0.49	5
	P_{35}	**41117**	0.000	1621622	4263.80	6217.31	=	0.000	1233866	280.97	2303.86	=	=	0.48	5
	P_{36}	**42717**	0.000	1588836	1195.27	1728.26	=	0.000	1151204	307.78	1996.69	=	=	0.47	5
	P_{37}	**33134**	0.000	151919	17.98	129.53	=	0.000	106603	46.83	180.43	=	=	0.33	5
	P_{38}	**32486**	0.000	159248	62.42	130.57	=	0.000	122609	87.59	207.11	=	=	0.35	5
	P_{39}	**33486**	0.000	142946	59.98	133.32	=	0.000	89212	43.55	174.16	=	=	0.34	5
	P_{40}	**34486**	0.000	135483	65.89	126.94	=	0.000	75243	11.69	167.78	=	=	0.32	5
S_4	P_{41}	**11574**	0.000	0	0.27	0.28	=	0.000	0	0.28	0.28	=	=	0.05	1
	P_{42}	**9708**	0.000	52949	12.38	15.03	=	0.000	39438	1.38	16.58	=	=	0.12	1

Continued on next page

Table 4 – continued from previous page

Set	Inst	CPLEX-1					CPLEX-2					CLOA			
		Obj_1	Gap_1	$\sharp nodes_1$	T_{best1}	T_{tot1}	Obj_2	Gap_2	$\sharp nodes_2$	T_{best2}	T_{tot2}	Obj_{min}	Obj_{max}	$\overline{T}_{best}$	T_{tot}
	P_{43}	**8637**	0.000	114100	21.11	80.73	=	0.000	387359	30.20	128.16	=	=	0.15	1
	P_{44}	**16426**	0.000	0	0.27	0.26	=	0.000	339	0.27	0.46	=	=	0.04	1
	P_{45}	**12514**	0.000	32068	2.16	20.31	=	0.000	78939	17.01	34.62	=	=	0.14	1
	P_{46}	**10741**	0.000	1070008	45.30	267.62	=	0.000	586595	56.84	214.76	=	=	0.42	1
	P_{47}	**13534**	0.000	0	0.27	0.28	=	0.000	0	0.27	0.28	=	=	0.03	1
	P_{48}	**11070**	0.000	21643	12.52	18.41	=	0.000	40757	20.63	37.52	=	=	0.13	1
	P_{49}	**9175**	0.000	663600	19.41	151.50	=	0.000	314893	43.84	138.60	=	=	0.14	1
	P_{50}	**16749**	0.000	0	0.27	0.28	=	0.000	468	0.27	0.55	=	=	0.05	1
	P_{51}	**15510**	0.000	192444	4.03	48.52	=	0.000	186121	32.55	76.57	=	=	0.20	1
	P_{52}	**21872**	0.000	0	0.31	0.30	=	0.000	0	0.14	0.25	=	=	0.04	1
	P_{53}	**20358**	0.000	199432	6.23	53.16	=	0.000	173908	22.06	79.79	=	=	0.22	1
	P_{54}	**19114**	0.000	0	0.13	0.28	=	0.000	0	0.13	0.27	=	=	0.04	1
	P_{55}	**17405**	0.000	137991	3.63	42.10	=	0.000	112199	6.72	78.94	=	=	0.24	1
S_5	P_{56}	**68082**	0.613	1084783	3650.72	TL	68205	0.672	2964041	278.88	TL	=	=	7.01	10
	P_{57}	**72582**	0.577	751024	6603.20	TL	72705	0.633	3100838	6813.63	TL	=	=	6.52	10
	P_{58}	**83082**	0.506	678488	459.33	TL	83205	0.554	3015296	6482.08	TL	=	83205	6.04	10
	P_{59}	74853	0.538	1481097	6574.09	TL	=	0.608	2991438	6806.88	TL	**74524**	74853	5.88	10
	P_{60}	**62434**	0.584	1331926	6507.02	TL	63793	0.660	2211503	6982.28	TL	=	=	1.75	10
	P_{61}	66696	0.561	1325976	6496.22	TL	**65434**	0.611	2383798	329.27	TL	=	=	1.74	10
	P_{62}	**72434**	0.494	1215719	1139.55	TL	=	0.547	1487204	6705.86	TL	=	=	2.07	10
	P_{63}	**66192**	0.526	1313219	6515.75	TL	=	0.596	1716062	417.61	TL	=	=	1.42	10
	P_{64}	**61953**	0.558	912276	6247.59	TL	=	0.647	6020843	195.11	TL	=	=	1.79	10
	P_{65}	**64353**	0.491	1537185	6745.45	TL	=	0.606	4297881	2525.58	TL	=	=	1.73	10
	P_{66}	**69953**	0.443	792721	1726.97	TL	=	0.547	2676110	721.61	TL	=	=	1.83	10
	P_{67}	**64962**	0.456	2031206	6662.70	TL	=	0.592	3310714	542.52	TL	=	=	1.12	10
	P_{68}	63870	0.568	1425019	6492.56	TL	=	0.659	3859757	6518.69	TL	**63798**	63870	3.02	10
	P_{69}	67109	0.541	1284200	6691.63	TL	**66870**	0.618	2866572	6642.13	TL	=	=	2.43	10
	P_{70}	**73870**	0.479	1313442	6601.77	TL	=	0.551	2055025	297.86	TL	=	=	3.09	10
	P_{71}	**67801**	0.522	983156	7044.45	TL	=	0.594	2057502	7074.45	TL	=	=	1.98	10

Table 5: Computational results for sets S_6 to S_{10}. Termination condition is 7200 seconds for CPLEX and 1800 seconds for CLOA. Times in seconds

Set	Inst	CPLEX-1				CPLEX-2				CLOA				
		Obj_1	Gap_1	$\sharp nodes_1$	T_{best1}	Obj_2	Gap_2	$\sharp nodes_2$	T_{best2}	Obj_{min}	Min_{Gap}	Obj_{max}	Max_{Gap}	$\overline{T}_{best}$
S_6	i300-1	11799.10	0.063	25452	7017.91	**11789.92**	0.070	50377	3319.66	=	0.000	11797.36	0.001	687.37
	i300-6	**6452.21**	0.124	23492	6533.13	6486.80	0.138	45265	2558.66	=	0.000	=	0.000	367.71
	i300-11	**3946.48**	0.132	25966	39.41	=	0.157	53386	39.08	=	0.000	=	0.000	32.72
	i300-16	**3222.57**	0.218	17050	5148.72	3250.72	0.249	52175	1509.77	=	0.000	=	0.000	28.19
S_7	i3001500-1	91377.53	0.719	5	3617.55	94033.17	0.727	4	3867.45	**80772.31**	−0.116	82339.48	−0.099	1749.80
	i3001500-6	88067.57	0.772	0	6321.72	90895.69	0.780	0	400.91	**69769.42**	−0.208	70433.13	−0.200	1607.24
	i3001500-11	75304.07	0.764	0	6584.45	80832.17	0.786	0	1959.63	**64740.03**	−0.140	65289.57	−0.133	1358.00
	i3001500-16	79627.58	0.779	0	1799.59	80434.28	0.782	0	2090.98	**62895.47**	−0.210	63004.71	−0.209	970.71
S_8	i500-1	19458.32	0.078	1679	TL	**19392.37**	0.075	1918	5210.33	19568.00	0.009	19803.15	0.021	1741.83
	i500-6	9261.79	0.169	1963	6348.53	=	0.171	1995	6813.50	**9222.24**	−0.004	9238.75	−0.002	1290.47
	i500-11	6693.55	0.221	1915	TL	6740.27	0.230	1603	7003.42	**6663.40**	−0.005	=	−0.005	454.70
	i500-16	5243.15	0.271	880	6565.17	=	0.271	2142	5912.06	**5139.52**	−0.020	=	−0.020	162.47
S_9	i700-1	**29437.46**	0.091	166	5015.59	29492.16	0.093	297	4011.17	38934.41	0.323	41978.44	0.426	1803.17
	i700-6	13415.45	0.194	69	3513.27	13503.82	0.199	80	1791.56	**13281.61**	−0.010	13481.70	0.005	1737.27
	i700-11	9921.47	0.274	32	4030.76	9912.15	0.274	25	3953.84	**9613.85**	−0.030	9650.29	−0.026	1611.92
	i700-16	7514.83	0.321	49	1351.45	=	0.321	65	1363.20	**7368.49**	−0.019	7385.80	−0.017	1090.71
S_{10}	i1000-1	**41883.62**	0.114	0	6818.28	47289.94	0.215	0	4149.86	82839.93	0.978	85779.56	1.048	1811.76
	i1000-6	21379.67	0.233	0	6343.61	**20324.35**	0.193	0	6549.70	28052.77	0.380	35901.92	0.766	1790.47
	i1000-11	**14221.12**	0.278	0	6950.56	=	0.278	0	6618.20	15176.85	0.067	15830.43	0.113	1780.83
	i1000-16	12464.31	0.359	0	6456.81	=	0.359	0	6580.52	**11908.39**	−0.045	12219.67	−0.020	1742.24

5.3. Comparing models C-SPLPO-1 and C-SPLPO-2

Next, aiming to emphasize the differences between the models C-SPLPO-1 and C-SPLPO-2, we include in this section a thorough analysis of the structure of solutions obtained for both models. We also include in this analysis the relaxed problem in which the preferences of customers are not taken into account. This model is formulated as:

$$
\begin{aligned}
\min_{x,y} \quad & \textstyle\sum_{i\in I} f_i y_i \ + \ \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \\
\text{subject to} \quad & \\
& \textstyle\sum_{i\in I} q_i y_i \geqslant m \\
& \textstyle\sum_{i\in I} x_{ij} = 1, \quad j \in J \\
& \textstyle\sum_{j\in J} x_{ij} \leqslant q_i y_i, \quad i \in I \\
& x_{ij} \in \{0,1\}, \quad i \in I,\ j \in J \\
& y_i \in \{0,1\}, \quad i \in I
\end{aligned}
$$

In order to use the same software for the three models, we have selected the solution provided by CPLEX. Moreover, we have chosen CPLEX-1 since, according to the CPLEX documentation, it is, in general, quicker. For the model C-SPLPO-1, there are five instances of sets S_9 and S_{10} for which CPLEX has not provided any information after 7200 seconds of computing time. For these instances, we have also tried the variant CPLEX-2, but the same results are obtained. Hence, only the information provided by the remaining 86 instances is used to assess the differences between the models.

For all the instances, the optimal solution (or the best known feasible solution) of the three models is different. For these solutions, Table 6 displays the value of the total cost, the sum of preferences, and the number of unsatisfied customers (i.e. customers who are allocated to an open facility which is not their most preferred one among the open facilities), as well as the total CPU time in seconds required by CPLEX-1. C-SPLPO-1 is not feasible for instances P_{44} and P_{54} and, when it is feasible, it provides the largest cost, thus confirming that the C-SPLPO-1 is, in general, a very restrictive model. Obviously, the relaxed problem provides a lower bound on the objective function value for both models C-SPLPO-1 and C-SPLPO-2. Concerning the sum of preferences, as expected, the worst value is provided by the relaxed model (which does not take the preferences into account). The number of unsatisfied customers is zero for the C-SPLPO-1 since this is a constraint of the model, but it is not very large for the C-SPLPO-2 with respect to the number of customers. Finally, although CPU times are longer, in general, for the C-SPLPO-2, the matheuristic which is the subject of this paper requires much shorter computing times. As an illustration, the optimal solution corresponding to instance P_1 has been included in Table 7.

Table 6: Comparative results. Times in seconds

Set	Inst	Total cost			Sum of preferences			# of unsatisfied customers			T_{tot}		
		Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2
S_1	P_1	8806	19779	18592	279	84	77	43	0	5	0.05	0.15	0.21
	P_2	7856	19047	17658	279	84	77	43	0	5	0.05	0.03	0.15
	P_3	9286	20247	19058	285	84	77	42	0	5	0.04	0.02	0.18
	P_4	10686	21447	20442	285	84	85	42	0	6	0.06	0.04	0.23
	P_5	8669	21434	18552	261	67	80	43	0	5	0.04	0.03	0.12
	P_6	7663	20573	17806	261	67	80	43	0	5	0.05	0.03	0.15
	P_7	9232	22173	19206	265	67	80	42	0	5	0.08	0.03	0.11
	P_8	10632	23773	20606	265	67	80	42	0	5	0.05	0.03	0.11
	P_9	8420	18858	17651	243	84	104	40	0	3	0.04	0.17	0.23
	P_{10}	7611	18204	17146	250	84	98	42	0	5	0.04	0.14	0.20
	P_{11}	8890	19404	18146	254	84	98	39	0	5	0.02	0.14	0.54
	P_{12}	10090	20604	19146	254	84	98	39	0	5	0.06	0.17	0.37
S_2	P_{13}	8051	19983	17745	488	137	150	43	0	12	0.07	0.50	71.17
	P_{14}	7092	18493	16720	523	137	151	45	0	7	0.07	0.45	90.06
	P_{15}	8737	20093	18120	488	137	151	43	0	7	0.10	0.48	75.53
	P_{16}	10275	21522	19427	485	130	166	41	0	12	0.13	0.44	73.19
	P_{17}	8049	19467	17613	482	109	138	43	0	8	0.10	1.08	47.12
	P_{18}	7092	18425	16718	523	109	138	45	0	8	0.09	1.45	55.67
	P_{19}	8735	20025	18118	482	109	138	43	0	8	0.10	1.14	48.38
	P_{20}	10318	21511	19518	527	143	138	43	0	8	0.24	1.37	55.28
	P_{21}	8049	18672	17253	482	135	151	43	0	2	0.06	0.90	37.14
	P_{22}	7092	17635	16407	523	135	151	45	0	2	0.07	0.90	48.97
	P_{23}	8735	19035	17607	482	135	151	43	0	2	0.12	0.99	41.27
	P_{24}	10210	20380	18807	486	131	151	41	0	2	0.26	1.38	33.14
S_3	P_{25}	11504	51812	40164	2123	496	576	124	0	28	0.18	14.06	1086.17
	P_{26}	10725	50345	39266	2148	496	576	125	0	28	0.25	11.84	4068.63
	P_{27}	12129	52145	40866	2100	496	576	124	0	28	0.20	14.44	1044.05
	P_{28}	13529	53945	42466	2100	496	576	124	0	28	0.20	12.89	3880.24
	P_{29}	12179	50392	44715	2269	383	437	133	0	9	0.17	6.22	TL
	P_{30}	11088	48951	43480	2366	383	458	136	0	6	0.32	5.68	TL
	P_{31}	13088	51151	45480	2366	383	458	136	0	6	0.19	5.92	TL
	P_{32}	15088	53351	47596	2366	383	437	136	0	9	0.20	5.87	TL
	P_{33}	11594	43449	40428	2174	472	521	127	0	1	0.18	6.14	4907.95
	P_{34}	10597	42287	39517	2175	472	549	128	0	4	0.14	5.42	1702.77
	P_{35}	12197	44087	41117	2175	472	549	128	0	4	0.27	6.22	6217.31
	P_{36}	13797	45887	42717	2175	472	549	128	0	4	0.21	6.37	1728.26
	P_{37}	11248	36489	33134	2224	694	814	120	0	8	0.17	5.91	129.53
	P_{38}	10543	35866	32486	2176	694	814	120	0	8	0.15	4.97	130.57
	P_{39}	11816	37066	33486	2277	694	814	124	0	8	0.16	5.75	133.32
	P_{40}	13016	38266	34486	2277	694	814	124	0	8	0.26	5.74	126.94
S_4	P_{41}	6614	13718	11574	513	167	252	79	0	18	0.06	0.04	0.28
	P_{42}	5615	10384	9708	789	479	423	60	0	33	0.17	0.83	15.03
	P_{43}	5214	8958	8637	1062	469	407	52	0	10	0.32	8.21	80.73
	P_{44}	7028	Not feasible	16426	472		203	82		24	0.05		0.26

Continued on next page

Table 6 – continued from previous page

Set	Inst	Total cost			Sum of preferences			# of unsatisfied customers			T_{tot}		
		Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2
	P_{45}	6229	13291	12514	843	310	316	63	0	7	0.13	1.27	20.31
	P_{46}	5582	11243	10741	1086	592	490	62	0	25	0.29	6.36	267.62
	P_{47}	5903	15420	13534	515	174	176	77	0	26	0.08	0.03	0.28
	P_{48}	5708	11951	11070	759	315	339	66	0	12	0.17	0.77	18.41
	P_{49}	5279	10230	9175	1037	431	478	63	0	15	0.13	10.18	151.50
	P_{50}	8654	17985	16749	580	178	222	95	0	6	0.05	0.07	0.28
	P_{51}	7333	15706	15510	1087	338	303	79	0	7	0.16	1.90	48.52
	P_{52}	9148	25668	21872	534	133	211	88	0	24	0.07	0.03	0.30
	P_{53}	8529	22132	20358	971	245	306	83	0	15	0.10	1.62	53.16
	P_{54}	8726	Not feasible	19114	567		190	88		26	0.05		0.28
	P_{55}	7658	19499	17405	1073	387	369	84	0	17	0.13	0.98	42.10
S_5	P_{56}	20761	75155	68082	2966	310	414	189	0	29	0.26	8.76	TL
	P_{57}	25553	80255	72582	3064	310	414	188	0	29	1.12	10.73	TL
	P_{58}	36053	92155	83082	3064	310	414	188	0	29	0.90	13.76	TL
	P_{59}	26642	83136	74853	3033	312	401	189	0	24	0.46	12.79	TL
	P_{60}	20543	68236	62434	3018	476	646	190	0	25	0.29	161.57	TL
	P_{61}	24458	71836	66696	3074	476	585	188	0	23	0.35	182.31	TL
	P_{62}	32512	80236	72434	3008	476	646	188	0	25	2.49	184.42	TL
	P_{63}	25106	72638	66192	3022	476	646	186	0	25	0.44	95.73	TL
	P_{64}	20530	64542	61953	3014	596	709	190	0	13	0.29	249.38	TL
	P_{65}	24445	67300	64353	3070	694	709	188	0	13	0.43	199.13	TL
	P_{66}	31313	72900	69953	2982	694	709	178	0	13	4.11	278.33	TL
	P_{67}	24891	67716	64962	3093	694	711	181	0	12	0.42	328.09	TL
	P_{68}	20530	71032	63870	3014	453	610	190	0	37	0.29	170.55	TL
	P_{69}	24511	74370	67109	3102	517	678	189	0	44	0.49	227.54	TL
	P_{70}	32171	82070	73870	3149	517	610	185	0	37	1.47	195.90	TL
	P_{71}	25465	77207	67801	3075	393	586	189	0	37	0.50	224.66	TL
S_6	i300-1	10559.80	18541	11799.10	46026	1886	2304	293	0	80	4.13	TL	TL
	i300-6	5361.02	8163.36	6452.21	44635	3195	4339	285	0	49	5.70	TL	TL
	i300-11	3142.82	5146.54	3946.48	46315	5483	6196	280	0	30	3.77	TL	TL
	i300-16	2272.12	4181.38	3222.57	47570	6315	8205	278	0	41	5.2	TL	TL
S_7	i3001500-1	23172.70	233656	91377.50	226679	1602	9475	1472	0	339	387.08	TL	TL
	i3001500-6	18974.30	224767	88067.60	225761	1702	18605	1465	0	333	128.66	TL	TL
	i3001500-11	17793.50	220396	75304.10	230382	1782	26557	1450	0	513	232.13	TL	TL
	i3001500-16	17374.50	242466	79627.60	225872	1510	37481	1462	0	402	195.98	TL	TL
S_8	i500-1	17413.40	33572.90	19458.30	127779	2953	4243	491	0	134	7.64	TL	TL
	i500-6	7346.76	16494.60	9261.79	124307	4370	7646	483	0	90	12.29	TL	TL
	i500-11	4850.38	10170.40	6693.55	125888	6918	10397	481	0	74	14.16	TL	TL
	i500-16	3581.54	7760.18	5243.15	125726	12147	15191	467	0	58	17.77	TL	TL
S_9	i700-1	25991.20	155840	29437.50	241733	1269	5971	694	0	202	18.03	TL	TL
	i700-6	10308	132438	13415.40	247920	1249	10391	693	0	131	22.36	TL	TL
	i700-11	6910.59	27775.30	9921.47	241570	5184	15117	673	0	142	44.91	TL	TL
	i700-16	4743.60		7514.83	245992		23653	674		67	63.81		TL
S_{10}	i1000-1	36418.50		41883.60	486914		8201	995		234	75.63		TL

Continued on next page

Table 6 – continued from previous page

Set	Inst	Total cost			Sum of preferences			# of unsatisfied customers			T_{tot}		
		Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2	Relaxed	C-SPLPO-1	C-SPLPO-2
	i1000-6	15806.60		21379.70	489503		13910	980		148	197.82		TL
	i1000-11	9684.96		14221.10	497608		22439	979		161	85.23		TL
	i1000-16	7640.39		12464.30	507818		29592	971		190	376.20		TL

Table 7: Optimal solutions of instance P_1: Customers allocated to each facility.

Facility	Relaxed	C-SPLPO-1	C-SPLPO-2
1	18, 29, 33, 39, 42, 46, 48, 50	6, 12, 13, 17, 18, 25, 31, 49, 50	6, 13, 17, 18, 25, 31, 44, 49, 50
2	3, 10, 27, 30, 43	Closed	3, 4, 27, 41, 43
3	2, 7, 14, 25, 31, 47	8, 26, 27, 29, 33, 39, 43, 45, 46	8, 26, 29, 39, 42, 45, 46
4	5, 13, 16, 21, 34, 38, 41	Closed	7, 11, 12, 22, 28, 32, 33, 34
5	8, 9, 17, 20, 22, 24, 36, 37, 40, 45, 49	1, 2, 3, 4, 5, 9, 14, 15, 19, 28, 36	1, 2, 5, 9, 14, 15, 16, 19, 36, 37
6	Closed	Closed	Closed
7	4, 23, 26, 28, 32, 44	20, 23, 30, 35, 41	20, 23, 24, 30, 35, 40, 47
8	Closed	Closed	Closed
9	1, 6, 12, 15	10, 21, 24, 38	10, 21, 38, 48
10	11, 19, 35	7, 11, 16, 22, 32, 34, 37, 40, 42, 44, 47, 48	Closed

6. Conclusions and further research directions

Most of the literature on location problems in which customers are allowed to select the open facility they will patronize assumes that each facility can accommodate as many customers as required. As a consequence, each customer can freely select a facility in accordance with his/her preferences. A more realistic model arises when the existence of a cardinality constraint on the number of customers who can be allocated to each facility is taken into account.

In this paper, the implications of extending original models with preferences to handle cardinality constraints are analyzed. The extension of the single level formulation seems to be only appropriate if it is compulsory to guarantee individual preferences. Otherwise, the bilevel optimization extension provides a wider approach to the problem. In the upper level of the hierarchy, the decision maker controls which facilities are open aiming to minimize the total cost. In the second level of the hierarchy, the customer allocation is solved aiming to minimize the global customer preference. Properties of the lower level problem allow us to reformulate the bilevel model as a single level model without including additional binary variables. This reformulated model can be solved by using standard mathematical techniques.

We have also developed a matheuristic for solving the bilevel model. This algorithm evolves as an evolutionary algorithm does, but also involves solving a lexicographic optimization model to compute feasible solutions of the bilevel model. The algorithm has proved to be very fast and efficient in the computational experimentation carried out. Moreover, it is worth pointing out that the algorithm can also be applied to solve problem (2) with different and even more complicated upper level objective functions. Only a single change in the algorithm would be needed that would affect the computation of the fitness function.

Future lines of research could consider other ways of dealing with preferences when assessing the reaction of customers. Minimizing the sum of preferences can yield an unfair solution as it compensates between the preferences of different customers. Therefore, other criteria such as minimizing the maximum of the preferences could be appraised. It is worth pointing out that, in this case, the lower level problem is NP-hard. Although CLOA can be adapted to deal with this problem, by substituting the objective function in problem (6) by $\text{lex}\min_{x} \left(\max_{i \in I(y), j \in J} g_{ij} x_{ij}, \quad \sum_{i \in I(y)} \sum_{j \in J} c_{ij} x_{ij} \right)$, due to the complexity of this problem only small or medium size problems could be expected to be handled.

7. References

[1] M. Affenzeller, S. Wagner, S. Winkler, and A. Beham. *Genetic algorithms and genetic programming: modern concepts and practical applications*. Chapman & Hall/CRC, 2009.

[2] P. Avella and M. Boccia. A cutting plane algorithm for the capacitated facility location problem. *Computational Optimization and Applications*, 43(1):39–65, 2009.

[3] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Network Flows*. Wiley-Interscience, Hoboken, NJ, third edition, 2005.

[4] H.I. Calvete and P.M. Mateo. Lexicographic optimisation in generalised network flow problems. *Journal of the Operational Research Society*, 49(5):519–529, 1998.

[5] H.I. Calvete, C. Galé, and J.A. Iranzo. Planning of a decentralized distribution network using bilevel optimization. *Omega*, 49:30–41, 2014.

[6] J.F. Camacho-Vallejo, A.E. Cordero-Franco, and R.G. González-Ramírez. Solving the bilevel facility location problem under preferences by a Stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering*, (ID 430243):1–15, 2014.

[7] L. Cánovas, S. García, M. Labbé, and A. Marín. A strengthened formulation for the simple plant location problem with order. *Operations Research Letters*, 35:141–150, 2007.

[8] M. Caramia and R. Mari. A decomposition approach to solve a bilevel capacitated facility location problem with equity constraints. *Optimization Letters*, 10(5):997–1019, 2016.

[9] M.S. Casas-Ramírez, J.F. Camacho-Vallejo, and I.A. Martínez-Salazar. Approximating solutions to a bilevel capacitated facility location problem with customer's patronization toward a list of preferences. *Applied Mathematics and Computation*, 319:369–386, 2018.

[10] R. Chion, T. Weise, and Z. Michalewicz, editors. *Variants of Evolutionary Algorithms for Real-World Applications*. Springer, Berlin, 2012.

[11] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel programming. *Annals of Operations Research*, 153:235–256, 2007.

[12] M.S. Daskin. *Network and Discrete Location. Models, Algorithms, and Applications*. Wiley, Hoboken, NJ, 2nd edition, 2013.

[13] S. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization*, 52:333–359, 2003.

[14] Z. Drezner and H.W. Hamacher, editors. *Facility Location. Applications and Theory*. Springer, Berlin, 2002.

[15] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, Heildeberg, 2nd edition, 2005.

[16] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.

[17] P. Hanjoul and D. Peeters. A facility location problem with clients' preference orderings. *Regional Science and Urban Economics*, 17:451–473, 1987.

[18] P. Hansen, Y. Kochetov, and N. Mladenović. Lower bounds for the uncapacitated facility location problem with user preferences. *Les Cahiers du GERAD*, G-2004-24, 2004.

[19] K. Holmberg, M. Ronnqvist, and D. Yuan. An exact algorithm for the capacitated facility location problem with single sourcing. *European Journal of Operational Research*, 113:544–559, 1999.

[20] T. Kleinert, M. Labbé, F. Plein, and M. Schmidt. There's No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization. Working paper, June 2019. URL `https://hal.inria.fr/hal-02106642`.

[21] A. Klose and A. Drexl. Facility location models for distribution system design. *European Journal of Operational Research*, 162(1):4–29, 2005.

[22] G. Laporte, S. Nickel, and F. Saldanha da Gama, editors. *Location Science*. Springer, 2015.

[23] L. Lozano and J. Cole Smith. A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research*, 65(3):768–786, 2017.

[24] Z. Michalewick. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, 3rd edition, 1996.

[25] S. Nickel and J. Puerto. *Location Theory. A Unified Approach*. Springer, Berlin, 2005.

[26] C.S. ReVelle and H.A. Eiselt. Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19, 2005.

[27] C.S. ReVelle, H.A. Eiselt, and M.S. Daskin. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184(3):817–848, 2008.

[28] P. Rojeski and C.S. ReVelle. Central facilities location under an investment constraint. *Geographical Analysis*, 2(4):343–360, 1970.

[29] M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35:254–265, 1987.

[30] S. Tahernejad, T.K. Ralphs, and S.T. DeNegre. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. Technical report, COR@L Technical Report 16T-015-R3, 2017.

[31] I.L. Vasilyev and K.B. Klimentova. The branch and cut method for the facility location problem with client's preferences. *Journal of Applied and Industrial Mathematics*, 4(3):441–454, 2010.

[32] I.L. Vasilyev, K.B. Klimentova, and Y.A. Kochetov. New lower bounds for the facility location problem with client's preferences. *Computational Mathematics and Mathematical Physics*, 49(6):1010–1020, 2009.

[33] J.L. Wagner and L.M. Falkson. The optimal nodal location of public facilities with price-sensitive demand. *Geographical Analysis*, 7(1):69–83, 1975.