




ε -Constraint method for bi-objective competitive facility location problem with uncertain demand scenario

Vladimir Beresnev^{1,2,3} · Andrey Melnikov^{1,2} 

Received: 16 September 2018 / Accepted: 26 June 2019 / Published online: 6 July 2019

© The Association of European Operational Research Societies and Springer-Verlag GmbH Berlin Heidelberg 2019

Abstract

We consider a model of two parties' competition organized as a Stackelberg game. The parties open their facilities intending to maximize profit from serving the customers that behave following a binary rule. The set of customers is unknown to the party which opens its facilities first and is called the Leader. Instead, a finite list of possible scenarios specifying this set is provided to the Leader. One of the scenarios is to be realized in the future before the second party, called the Follower, would make their own decision. The scenarios are supplied with known probabilities of realization, and the Leader aims to maximize both the probability to get a profit not less than a specific value, called a guaranteed profit, and the value of a guaranteed profit itself. We formulate the Leader's problem as a bi-objective bi-level mathematical program. To approximate the set of efficient solutions of this problem, we develop an ε -constraint method where a branch-and-bound algorithm solves a sequence of bi-level problems with a single objective. Based on the properties of feasible solutions of a bi-level program and mathematical programming techniques, we developed three upper bound procedures for the branch-and-bound method mentioned. In numerical experiments, we compare these procedures with each other. Besides that, we discuss relations of the model under investigation and the stochastic competitive location model with uncertain profit values.

Keywords Location · Bi-level programming · Multi-criteria optimization · Stackelberg game · Uncertainty

✉ Andrey Melnikov
melnikov@math.nsc.ru

Vladimir Beresnev
beresnev@math.nsc.ru

¹ Sobolev Institute of Mathematics, Novosibirsk, Russia

² Department of Mechanics and Mathematics, Novosibirsk State University, Novosibirsk, Russia

³ Department of Informational Technologies, Novosibirsk State University, Novosibirsk, Russia

Mathematics Subject Classification 90C11 · 90B50 · 90B80

1 Introduction

Consider a situation when a firm enters the market by opening facilities providing some service to customers. In general, a decision about facilities' location must anticipate the competitors' reaction which is rational and prescribed by a solution of some optimization problem. Competitive nature of the process cannot be ignored since facilities' income depends on the further actions of competitors. We assume that there is a single competitor on the market or all the competitors can be aggregated into a single one. Then, we can consider the party locating facilities as the Leader in a Stackelberg game (Stackelberg 1952). The second player of this game, called the Follower, represents the competitor who makes its decision when knowing the Leader's one.

Parameters of demand for products or services, provided by the competing parties, are affected by many factors and are hardly predictable. Besides the uncertainty inherent to predicted data, the demand is sensitive to external factors influencing its structure such as economic and political circumstances. We assume the Leader to consider a finite number of possible demand scenarios. Each scenario fully characterizes the set of customers with their attributes and has a known probability of realization. One of the scenarios will be implemented in the future, but, at the moment of making a decision, the available information is insufficient to determine which one it would be.

In a Stackelberg game framework, the Follower decides after the Leader. It provides the Follower with knowledge about the location of the Leader's facilities. Moreover, additional pieces of evidence about the demand scenario reveal themselves, so the Follower can distinguish the scenario realized before making their own decision. Thus, unlike the Leader, the Follower is in a situation of full information when all the uncertainties have vanished. In this aspect, the BCompFLP can be considered as a so-called two-stage model, where decisions referred to as here-and-now decisions are made before revealing the values of uncertain parameters. When these values are known, other decisions called wait-and-see are computed. In this terminology, the Leader's location decisions are here-and-now decisions, whereas the Follower's location and customer's distribution are wait-and-see decisions since these values are computed when the scenario is implemented.

Similarly to maximization facility location problem (Krarup and Pruzan 1983), we assume the parties to maximize their profit calculated as an income from service the customers minus fixed costs of open facilities. In a situation of uncertain demand scenario, the Leader's income is a random variable with a discrete distribution. To compare different Leader's solutions with each other, numeric characteristics of the income distribution such as expected value, value-at-risk (VaR_α), conditional value-at-risk ($CVaR_\alpha$), and others could be used (Rockafellar and Uryasev 2002). The expectation is naturally the most widely used metric. In the paper (Yanikoğlu and Kuhn 2018), the Leader's objective function includes a mathematical expectation of the revenue from serving the customers with uncertain demand volumes. The authors consider the lower-level problem to be a linear programming problem with continuous variables. In our case, the wait-and-see variables are involved into the Follower's

location problem having a form of mixed-integer programming problem (MIP) being significantly harder. We assume that the Leader maximizes a guaranteed profit instead of the average one. At the same time, the Leader might be interested in knowing not only the safest and conservative solution but more risky and profitable ones as well. To satisfy this need, we supplement the Leader's problem with the second objective function, representing a probability of obtaining a guaranteed profit. Following this way, we come to a bi-objective competitive facility location problem (BCompFLP), where both the Leader's profit and the probability of getting it are maximized on the upper level. The lower-level problem of the BCompFLP is to maximize the Follower's profit in each of the scenarios.

Possible applications for the BCompFLP may arise when there are several projects of reformations. For instance, a modification of the transportation system alters an attainability of facilities. Reformation of the taxation scheme affects the prices and, consequently, the demand of customers. Thus, each reformation project specifies customers' behavior and characterize an individual scenario. In this situation, the probability of scenario implementation can be estimated by experts who are familiar with practices of decision-making in reformation processes. Another application case is the presence of different economic predictions provided by many experts. The buying power of the customers and their interest in a certain kind of products or services depends on the economic situation. The Leader may build scenarios based on parameters predicted by experts, and the probabilities of scenarios implementation can be coherent to the authority of the experts and their level of confidence. Finally, the scenarios are generated by sampling the demand parameters from some anticipated distribution when sample approximation (Emelogu et al. 2016) is applied to solve the Leader's problem with stochastic demand. All the scenarios are equiprobable in the sample approximation problem, so this special class of BCompFLP instances, where the probabilities of realization are equal, is of particular interest.

The current state of the competitive location area is shown in detail by the most recent reviews (Kress and Pesch 2012; Karakitsiou 2015; Ashtiani 2016; Aras and Küçükaydın 2017). To put the present paper into the context of existing studies, we list the most significant modeling and algorithmic approaches to competitive location. As it is noticed in Spoerhase (2010) and Ashtiani (2016), the ingredients of the competitive location models are the space to establish facilities in, the customer behavior rule used, the demand type, the possibility of redesign or relocate facilities, and other factors. Besides, we refer to a review (Farahani et al. 2010) observing objective function types used in multi-criteria location problems.

The underlying space where the parties locate their facilities is considered to be either continuous or discrete. In continuous formulations, a facility can be placed at any of an infinite number of points from the continuous space such as Euclidean plane (Davydov et al. 2014; Saiz et al. 2009) or edges of a graph representing a transportation network (Bandyapadhyay et al. 2015). The present paper considers a discrete location model where the set of potential location is finite and can be represented by an abstract finite set or a set of graph nodes.

Regarding the customers' behavior model, in most cases, an initial point is a definition of *utility* perceived by a customer from being served by a facility. Depending on aspects taken into account by the model, the utility is assumed to be a

function of traveling distance, quality of the facility, service prices, and other measurable factors. From the computed utilities, two kinds of behavior can be simulated (Santos-Peñate et al. 2007). In proportional behavior rule, a customer's demand is split among the open facilities in proportion to utility values. Models incorporating this rule are burdened by nonlinearities expressing the proportions and can be applied to problems of simultaneously choosing the location and design of new facilities. A short communication on models with a particular case of proportional rule called Huff-like or gravity-based rule is given in Fernández and Hendrix (2013). A cover-based proportional rule, where the demand is equally split among the facilities covering the customer by their spheres of influence, is presented in Drezner et al. (2015). Instead of visiting all the facilities producing nonzero utility, a customer might choose a single facility to get service from by maximizing the utility value perceived. In this case, we come to a binary behavior rule. Generally, additional assumptions must be made to break ties between facilities producing the same utility value. This issue is discussed in detail in Pelegrín et al. (2015). If the utility value depends only on the location of the facility and all the utility values perceived by a customer are different, we come to a situation when the customer is associated with an order defined on the set of potential facility locations. Based on the utility values, the customer may compare any couple of facilities and decide which one they would prefer to another. In the present paper, we assume the customers to be accompanied with linear-order relations representing their preferences. Binary behavior rule can be formulated with linear inequalities and is suitable for modeling the demand for homogeneous products. Polyhedral properties of formulations for this rule are studied in Vasilyev et al. (2013) and Canovas et al. (2007).

Due to the high computational complexity of bi-level programming problems, multi-objective formulations are studied rarely in the literature. In Uno and Katagiri (2008), the authors consider a problem to prevent an attacker from reaching important nodes of the network. The Leader is a protection planner who opens defensive facilities at some nodes, while the Follower, an attacker, minimizes an amount of energy consumed while moving toward the important nodes. In the multi-objective formulation, for each of the important nodes, a corresponding objective represents the amount of energy the attacker has when it reaches the node. The authors propose a heuristic approach to find satisfactory solutions. A series of works (Fernández and Tóth 2009; Redondo et al. 2015) focuses on single-level nonlinear continuous competitive location problems with two objectives: maximizing both the market share and the profit obtained. In Gang et al. (2015) and references therein, evolution-based approaches are utilized to solve some industrial problems formulated in terms of multi-objective bi-level programming. Finally, in Alekseeva et al. (2017), the authors consider a bi-level location problem where the lower-level problem has several objectives. That case is complicated even in terms of definitions since multiple incomparable solutions of the lower-level give different values to the Leader's objective function. The authors overcome terminological and computational difficulties and propose a hybrid procedure to obtain near-optimal solutions for the upper-level problem.

The papers (Beresnev 2014; Hemmati and Smith 2016) consider formulations, closely related to the one studied in the present paper. They propose exact algorithms to maximize the Leader's profit in a case when the demand scenario is known. The algorithms implement the branch-and-bound approach and the cut generation scheme,

respectively. In contrast to Beresnev (2014), in Hemmati and Smith (2016), another interpretation of the location model is given where competing parties choose products to place on the market. Customers buy the most preferable product, whereas, according to free choice of supplier rule used both in Beresnev (2014) and in the present paper, a party can serve the customer by any facility which is preferred to all the competitor's ones.

In the present paper, we suggest an ε -constraint method (Ehrgott 2006) to find an approximation of the Pareto frontier of BCompFLP. It consists in solving a series of single-objective bi-level subproblems where the value of the second Leader's objective function, representing a probability of getting a guaranteed profit, is bounded from below. Models of that type were considered in Ivanov and Morozova (2016) where a local search procedure is suggested to find a good Leader's solution. In Melnikov and Beresnev (2016), we have formulated an estimating problem in the form of MIP providing an upper bound for the single-objective subproblem. Two reformulations of the estimating problem have been suggested as well. In the present work, we incorporate the suggested upper bound procedures into an implicit enumeration scheme called as a subroutine in the ε -constraint method. To investigate the efficiency of elements of this scheme, we perform numerical experiments with artificial data. Besides studying the efficiency of the procedures developed, we explore connections of BCompFLP with stochastic competitive location model where profit values are random parameters with known distribution. In our experiments, we inspect the distribution of the Leader's objective function and try to answer the question if sample approximation of the stochastic problem can be utilized to get quality solutions of the initial problem.

The paper is organized as follows: In Sect. 2, we introduce all the necessary notations and formalize the BCompFLP in terms of bi-level integer programming. Concepts of feasibility are carefully discussed as well. The key ingredients of the proposed method such as the branch-and-bound scheme to solve a constrained Leader's problem and the upper bound procedures utilized are presented in Sect. 3. Computational study of the proposed procedures and experimental results are given in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Mathematical model

Before formalizing the Leader's problem, let us introduce some auxiliary terminology. When taking risks, the Leader may consider only some of the scenarios available and make a more specific decision leading to higher income. We call these scenarios *active*, and the other ones *inactive*, respectively. The least profitable active scenario determines the value of a *guaranteed income*, i.e., a tight lower bound for the value of income in a case when the scenario realized is an active one.

In the mathematical model of BCompFLP, we use the following index sets:

$I = \{1, \dots, m\}$ is a set of locations available to open facilities;

$S = \{1, \dots, l\}$ is a set of possible scenarios;

J_s is a set of customers' indexes in the scenario $s \in S$.

Without loss of generality, we assume that $J_{s_1} \cap J_{s_2} = \emptyset$ for all $s_1, s_2 \in S, s_1 \neq s_2$. This assumption is not restrictive since any customer appearing in several scenarios can

be duplicated so that all the copies would have different indexes. Let $J = \bigcup_{s \in S} J_s = \{1, \dots, n\}$ denote the set of all customers' indexes.

The list of input parameters for BCompFLP consists of

- f_i , the fixed cost of opening the Leader's facility $i \in I$;
- g_i , the fixed cost of opening the Follower's facility $i \in I$;
- c_{ij} , the income of the Leader's facility i from serving the customer $j \in J$;
- d_{ij} , the income of the Follower's facility i from serving the customer $j \in J$;
- p_s , the probability of realization of scenario $s \in S$.

Finally, we use the following binary variables.

- x_i is equal to one, if the Leader opens facility i , and zero otherwise;
- z_i^s is equal to one, if the Follower opens facility $i \in I$ in the scenario $s \in S$, and zero otherwise;
- x_{ij} is equal to one, if the Leader's facility $i \in I$ is assigned to serve the customer $j \in J$, and zero otherwise;
- z_{ij} is equal to one, if the Follower's facility $i \in I$ is assigned to serve the customer $j \in J$, and zero otherwise;
- δ_s is equal to one, if the scenario $s \in S$ is active, and zero otherwise.

We assume that each customer is served by a single facility chosen with respect to customer's preferences represented by a linear order on the set I . Given $i_1, i_2 \in I, j \in J$ the relation $i_1 \succeq_j i_2$ means that for the customer j , either i_1 is more preferable than i_2 or $i_1 = i_2$. If $i_1 \succeq_j i_2$ and $i_1 \neq i_2$, then we use the notation $i_1 \succ_j i_2$. The greatest element of a non-empty subset $I' \subseteq I$ according to the order \succeq_j is denoted with $\alpha_j(I') = \{i' \in I' | i' \succeq_j k \forall k \in I'\}$. Given a nonzero Boolean vector $v = (v_i), i \in I$, we set $\alpha_j(v) = \alpha_j(\{i \in I | v_i = 1\})$.

By using the introduced notations, the BCompFLP can be written as the following a bi-objective bi-level program:

$$\max_{(x_i), (x_{ij}), (\delta_s)} \left(- \sum_{i \in I} f_i x_i + \min_{s \in S | \delta_s = 1} \sum_{i \in I} \sum_{j \in J_s} c_{ij} x_{ij} \right), \tag{1}$$

$$\max_{(x_i), (x_{ij}), (\delta_s)} \sum_{s \in S} p_s \delta_s, \tag{2}$$

$$x_i \geq x_{ij}, \quad i \in I, \quad j \in J; \tag{3}$$

$$\sum_{i \in I} x_{ij} \leq \delta_s, \quad s \in S, \quad j \in J_s; \tag{4}$$

$$\tilde{z}_i^s + \sum_{k \in I | i \succ_j k} x_{kj} \leq 1, \quad i \in I, \quad s \in S, \quad j \in J_s; \tag{5}$$

$$x_i, \delta_s, x_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J, \quad s \in S; \tag{6}$$

$$(\tilde{z}_i^s), (\tilde{z}_{ij}) \text{ solves the Follower's problem:} \tag{7}$$

$$\max_{(z_i^s), (z_{ij})} \sum_{s \in S} \left(- \sum_{i \in I} g_i z_i^s + \sum_{i \in I} \sum_{j \in J_s} d_{ij} z_{ij} \right), \tag{8}$$

$$z_i^s \geq z_{ij}, \quad i \in I, \quad s \in S, \quad j \in J_s; \quad (9)$$

$$x_i + \sum_{k \in I | i >_j k} z_{kj} \leq 1, \quad i \in I, \quad j \in J; \quad (10)$$

$$x_i + z_i^s \leq 1, \quad i \in I, \quad s \in S; \quad (11)$$

$$z_i^s, z_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J, \quad s \in S. \quad (12)$$

The first objective function (1) of the upper-level problem represents the Leader's guaranteed profit calculated as fixed costs subtracted from the guaranteed income. The second Leader's objective (2) is to maximize the total probability of active scenario realization. Inequalities (3) forbid to serve customers from closed facilities; (4) guarantee that customers from active scenarios cannot be served by more than one facility; (5) ensure that the customer is served with a facility which is more preferable than any competitor's one.

The lower-level objective function (8) is a sum of profit values obtained by the Follower in all the scenarios. Its maximization is equivalent to maximizing the profit for each scenario individually. The constraints (9) and (10) have the same meaning as the upper-level constraints (3) and (5), respectively. Finally, the inequalities (11) provide that the Follower does not open facility in locations occupied by the Leader.

For brevity, further, we use the following vector notations. Vectors of Leader's and Follower's location variables' values $(x_i), i \in I$, and $(z_i^s), i \in I, s \in S$, are denoted by \mathbf{x} and \mathbf{z} , correspondingly. Similarly, values (δ_s) are stored in the vector δ . Assignment matrices (x_{ij}) and $(z_{ij}), i \in I, j \in J$, are denoted by \mathbf{X} and \mathbf{Z} , respectively. Given \mathbf{x} , we denote the problem \mathcal{F} , where the corresponding values of $x_i, i \in I$, are introduced into the inequalities (10), (11), by $\mathcal{F}(\mathbf{x})$. Analogously, the problem \mathcal{L} with the value of \mathbf{z} in the constraints (5) is denoted by $\mathcal{L}(\mathbf{z})$. Whole model (1)–(12) is further referred to as $(\mathcal{L}, \mathcal{F})$.

2.1 Pessimistic feasible solutions of the problem $(\mathcal{L}, \mathcal{F})$ and efficiency

Consider a triplet (X, δ, Z) , where $X = (\mathbf{x}, \mathbf{X}), Z = (\mathbf{z}, \mathbf{Z})$. We say that it is a *feasible solution* of the problem $(\mathcal{L}, \mathcal{F})$ induced by a pair (\mathbf{x}, δ) if it satisfies constraints (3)–(6), (9)–(12) and Z is an optimal solution of the problem $\mathcal{F}(\mathbf{x})$. To simplify the presentation, we additionally assume that assignment variables (x_{ij}) and (z_{ij}) take their optimal values in all the feasible solutions.

When the lower-level problem has multiple optimal solutions, it is necessary to specify the rule to select the one to be incorporated in the upper-level problem. The most common approach is considering optimistic and pessimistic feasible solutions (Dempe 2002; Yanıkoğlu and Kuhn 2018). For a single-objective bi-level maximization problem, these concepts assume that the lower-level optimal solution is chosen which maximizes or minimizes the upper-level objective function, respectively. However, when the upper-level problem is multi-objective, it is unclear, which lower-level solution is most or least desirable for the upper-level objective functions. This difficulty does not apply to the BCompFLP since the Follower influences the Leader's profit only. The set of active scenarios, determining the value of the second Leader's

objective function, can be chosen independently when locations and assignments of the competing parties are known. Thus, we can base the rule of choosing the Follower’s optimal solution on the value of the Leader’s profit. Often, a pessimistic formulation of a bi-level problem is more difficult in terms of the existence of an optimal solution and computing a feasible solution than an optimistic one. Here, we discuss a pessimistic formulation, but an optimistic one can be covered with the same technique after a few natural modifications.

Let $L_1(X, \delta, Z)$ denotes the value of objective function (1) on the feasible solution (X, δ, Z) . Analogously, let $L_2(X, \delta, Z) = \sum_{s \in S} p_s \delta_s$ be the corresponding value of the objective function (2). A feasible solution (X', δ, \tilde{Z}) induced by the pair (x, δ) is called a *pessimistic feasible solution* of the problem $(\mathcal{L}, \mathcal{F})$ if $L_1(X', \delta, \tilde{Z}) \leq L_1(X, \delta, Z)$ for any feasible solution (X, δ, Z) induced by (x, δ) .

Now we see that the pessimistic BCompFLP can be considered as a problem to maximize an implicitly given vector-function $f : \{0, 1\}^m \times \{0, 1\}^l \rightarrow \mathbb{R}^2$ such that, for any pair of (0,1)-vectors (x, δ) , we have $f_r(x, \delta) = L_r(X, \delta, \tilde{Z})$, $r = 1, 2$, where (X, δ, \tilde{Z}) is a pessimistic feasible solution induced by (x, δ) .

To compute $f(x, \delta)$, we could find an optimal Follower’s location \tilde{z} minimizing the Leader’s profit in all the scenarios. Similarly to the single-scenario formulation (Beresnev 2014), the procedure consists of two steps. At the first step, we solve the problem $\mathcal{F}(x)$ and obtain its optimum F^* . At the second step, an auxiliary MIP provides a Follower’s solution minimizing the Leader’s objective function (1).

Having optimal location vector $\tilde{z}^s = (\tilde{z}_i^s)$ of the Follower, we compute the value of Leader’s income from serving each of the customers:

$$u_j = \max_{i | i > j \alpha_j(\tilde{z}^s)} (c_{ij} x_i), \quad s \in S, \quad j \in J_s.$$

When δ is chosen, we can construct a pessimistic feasible solution induced by (x, δ) . If $\delta_s = 0$ for some $s \in S$, then $x_{ij} = 0$ for any $i \in I, j \in J_s$. For $s \in S$ such that $\delta_s = 1$, and all $j \in J_s$ such that $u_j > 0$, let $i_j = \arg \max_{i \in I} c_{ij} (x_i - \sum_{k | k > j} z_k^s)$. Then, we set $x_{i_j j} = 1$. For all other indexes $i \in I, j \in J$ we set $x_{ij} = 0$. The triplet $(X, \delta, \tilde{Z}), X = (x, (x_{ij}))$ is a desired pessimistic feasible solution of the problem $(\mathcal{L}, \mathcal{F})$. Now, $f(x, \delta)$ can be computed straightforwardly:

$$f_1(x, \delta) = - \sum_{i \in I} f_i x_i + \min_{s | \delta_s = 1} \sum_{j \in J_s} u_j, \quad f_2(x, \delta) = \sum_{s \in S} p_s \delta_s.$$

We say that the pair (x, δ) *strongly dominates* (x', δ') if $f(x, \delta) > f(x', \delta')$. Additionally, (x, δ) *dominates* (x', δ') if $f(x, \delta) \geq f(x', \delta')$ and there exists $r \in \{1, 2\}$ such that $f_r(x, \delta) > f_r(x', \delta')$. The pair (x, δ) is called (*weakly*) *efficient* if there is no other pair that (strongly) dominates (x, δ) .

3 ε -Constraint method

We have shown that BCompFLP can be considered as a problem to maximize a vector-function $f : \{0, 1\}^m \times \{0, 1\}^l \rightarrow \mathbb{R}^2$ mapping an arbitrary pair of (0,1)-vectors (\mathbf{x}, δ) into the values of objective functions $L_1(X, \delta, \tilde{Z}), L_2(X, \delta, \tilde{Z})$ on the pessimistic feasible solution (X, δ, \tilde{Z}) induced by the pair. Under solution of this problem, we would mean the set of all efficient pessimistic feasible solutions or, in other terminology, a Pareto frontier (Ehrgott 2006). Other objects related to a multi-objective problem are so-called nadir points (Özpeynirci 2017) being efficient solutions with the worst values of an individual objective function. Such solutions show the boundaries for values of each of the objective functions. In our case, these are the “best-case” and the “worst-case” solutions, i.e., whose ones where the set of active scenarios contains the most profitable scenario and all the possible scenarios, respectively.

To approximate the set of efficient solutions, we develop an ε -constraint method iteratively finding weakly efficient solutions of the Leader. On each iteration, the objective function (1) is maximized while the function (2) is bounded from below. Note that depending on δ , the function f_2 takes value from the set $\mathcal{P} = \{p' | p' = \sum_{s \in S'} p_s, S' \subseteq S\}$. It implies that the set of weakly efficient solutions may contain up to 2^l elements. We start from some initial value of the probability threshold p_0 and solve the problem

$$\max_{\mathbf{x}, \delta} f_1(\mathbf{x}, \delta), \quad (13)$$

$$f_2(\mathbf{x}, \delta) \geq p_0 \quad (14)$$

by an implicit enumeration algorithm. Schematically, it is a parallel depth-first search method with greedy branching function applied to location variables \mathbf{x} .

The obtained solution (\mathbf{x}^*, δ^*) is weakly efficient since, for any feasible solution (\mathbf{x}', δ') such that $f_2(\mathbf{x}', \delta') \geq p_0$, it holds $f_1(\mathbf{x}', \delta') \leq f_1(\mathbf{x}^*, \delta^*)$. On the next iteration, we increase the value of p_0 and repeat the procedure while $p_0 \leq 1$.

Since the left-hand side of the constraint (14) takes values from the set \mathcal{P} , it is sufficient to consider all the elements of \mathcal{P} as the probability threshold p_0 . Note that if the pair (\mathbf{x}_0, δ_0) , where $\delta_0 = 0$, is weakly efficient, then $f_1(\mathbf{x}, \delta) = 0$ for any weakly efficient pair (\mathbf{x}, δ) . Thus, computing the set of weakly efficient solutions, one may start from the initial value of p_0 equal to $\min_{s \in S} p_s$. Having the solution (\mathbf{x}^*, δ^*) of the problem (13)–(14), the next value of the threshold to be considered is $\min\{p \in \mathcal{P} | p > \sum_{s \in S} p_s \delta_s^*\}$. It is easy to see that the problem to find this value is NP-hard since it is relative to a (0,1)-knapsack problem with weights of items equal to their costs. Based on additional assumptions about the values of p_s , efficient approaches to modify the threshold value p_0 can be suggested. We apply a strategy to increase p_0 by some predefined discretization parameter Δ after each iteration.

3.1 Solving a subproblem by implicit enumeration

To get an advantage of the power of modern computers, we have developed a parallel branch-and-bound algorithm to solve a subproblem (13), (14). The algorithm is orga-

nized in a master–worker framework. A master thread keeps all the information about the computational process and maintains exchanging data between worker threads. An essential role of the master thread is balancing the load of the workers. It is performed by splitting the unexplored part of the feasible region and assigning idling workers to perform the searching process in the subregions obtained. Each of the workers performs a depth-first search and informs the master about computational progress.

During the branching process, some components of the vector \mathbf{x} become fixed to a given value, 0 or 1. A worker stores the information about these components in a *partial solution* $\mathbf{y} = (y_i)$ in a following way. A component y_i , $i \in I$, is set to zero or one if x_i is fixed to take the corresponding value. Otherwise, y_i is set to be equal to an undetermined value “*”. Further, we use additional notations $I^\theta(\mathbf{y}) = \{i \in I | y_i = \theta\}$, $\theta \in \{0, 1, *\}$, and $P(\mathbf{y}) = \{\mathbf{x} \in \{0, 1\}^m | x_i = y_i \text{ for all } i \in I^0(\mathbf{y}) \cup I^1(\mathbf{y})\}$. The problem \mathcal{L} augmented with constraints $x_i = y_i$, $i \in I^1(\mathbf{y}) \cup I^0(\mathbf{y})$, is further referred to as $L(\mathbf{y})$.

Having T working threads, we initialize a computation by generating T partial solutions $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T$ such that $\bigcup_{t \in \{1, \dots, T\}} P(\mathbf{y}_t) = \{0, 1\}^m$ and $P(\mathbf{y}_{t_1}) \cap P(\mathbf{y}_{t_2}) = \emptyset$ for $t_1 \neq t_2$. Further, the worker t is assigned to solve the problem $\mathcal{L}(\mathbf{y}_t)$. When a subset $P(\mathbf{y}_t)$ is explored, one of the workers, having a relatively large subset of unexplored solutions, receives a command to split its workload. For a partial solution \mathbf{y} encoding the unexplored subset, two partial solutions \mathbf{y}^0 and \mathbf{y}^1 are generated. These partial solutions differ from \mathbf{y} in a single component $i \in I^*(\mathbf{y})$ in a such a way that $y_i^0 = 0$ and $y_i^1 = 1$. A worker–donor proceeds with the partial solution \mathbf{y}^0 , whereas a worker–recipient initializes computations with \mathbf{y}^1 .

An implicit enumeration of a subset $P(\mathbf{y})$ relies on the upper bound for a maximum of the function f_1 , which is computed by solving a specially constructed *estimating problem* proposed in Melnikov and Beresnev (2016). The problem has a form of MIP and can be tackled with a common MIP solver. Two alternative ways of computing an upper bound, based on the mentioned model, are proposed as well. Further, we describe all these procedures.

3.2 Estimating problem $\mathcal{B}(\mathbf{y})$

Similarly to the single-scenario competitive location model considered in Beresnev (2014), the upper bound for the optimum of $\mathcal{L}(\mathbf{y})$ can be obtained by solving a specially constructed MIP. To formulate it, we compute a system of estimating subsets $\{I_j(\mathbf{y})\}$, $j \in J$. The algorithm of constructing the estimating subsets is given in Beresnev (2013). The key property of these subsets is formulated in the following lemma.

Lemma 1 (Beresnev 2013) *Given $j \in J$, let $\mathbf{x} \in P(\mathbf{y})$ and $\alpha_j(\mathbf{x}) \notin I_j(\mathbf{y})$. Then, $\sum_{k \in I | k > \alpha_j(\mathbf{x})} \tilde{z}_k \geq 1$.*

Further, we transform the income matrix (c_{ij}) into a matrix (c'_{ij}) which is monotone with respect to the preferences of customers and majorizes (c_{ij}) . It means that for every $j \in J$, given $i_1, i_2 \in I$, the relation $i_1 \succeq_j i_2$ implies that $c'_{i_1 j} \geq c'_{i_2 j}$. Additionally, $c'_{ij} \geq c_{ij}$ for all $i \in I$, $j \in J$. Such a matrix can be constructed by setting $c'_{ij} = \max_{k | i \succeq_j k} c_{kj}$ for all $i \in I$ and $j \in J$.

Lemma 2 For every $j \in J$ and every pessimistic feasible solution (X, δ, \tilde{Z}) of the problem $(\mathcal{L}(y), \mathcal{F})$, the following inequality holds:

$$\sum_{i \in I} c_{ij}x_{ij} \leq \max_{i \in I_j} c'_{ij}x_i, \tag{15}$$

where maximum over the empty set equals zero.

Proof If $x = 0$, then the inequality holds. Assume that the condition (15) is violated for some pessimistic feasible solution (X, δ, \tilde{Z}) of the problem $(\mathcal{L}(y), \mathcal{F})$ induced by a pair (x, δ) with $x \neq 0$. In this case, $k = \arg \max_{i \in I} c_{ij}x_{ij} \notin I_j$. Moreover, for any $i \in I_j$ such that $x_i = 1$ we have $k \succ_j i$ due to monotonicity of the matrix c'_{ij} . It implies that $\alpha_j(x) \notin I_j$, but from Lemma 1 and conditions (5), $x_{ij} = 0$ for any $i \in I$ since there exists a Follower’s facility which is more preferable for the customer j than $\alpha_j(x)$. Thus, we come to a contradiction. \square

By setting

$$c''_{ij} = \begin{cases} c'_{ij}, & \text{if } i \in I_j(y) \\ 0, & \text{otherwise} \end{cases}$$

for all $i \in I, j \in J$, we get the following estimating problem:

$$\max_{(x_i), (x_{ij}), (\delta_s), C} \left(- \sum_{i \in I^1(y)} f_i - \sum_{i \in I^*(y)} f_i x_i + C \right), \tag{16}$$

$$x_i \geq x_{ij}, \quad i \in I^*(y), \quad j \in J; \tag{17}$$

$$\sum_{i \in I \setminus I^0(y)} x_{ij} \leq \delta_s, \quad s \in S, \quad j \in J_s; \tag{18}$$

$$C \leq \sum_{i \in I \setminus I^0(y)} \sum_{j \in J_s} c''_{ij}x_{ij} + M_s(1 - \delta_s), \quad s \in S; \tag{19}$$

$$\sum_{s \in S} p_s \delta_s \geq p_0; \tag{20}$$

$$x_i = y_i, \quad i \in I^0(y) \cup I^1(y); \tag{21}$$

$$x_i, x_{ij}, \delta_s \in \{0, 1\}, \quad i \in I^*(y), \quad s \in S; \tag{22}$$

The model (16)–(22) is further referred to as $\mathcal{B}(y)$. Inequalities (17) and (18) are taken from the problem $\mathcal{L}(y)$. Constraints (19) are the corollary of Lemma 2. Finally, the inequality (20) bounds from below the value of the second objective function of the Leader.

Notice that an appropriate choice of values $M_s, s \in S$, in the inequalities (19) reduces the integrality gap of the model $\mathcal{B}(y)$ and speeds up the computation of the upper bound. Given $s' \in S$, we choose the value of $M_{s'}$ by the following rule. Firstly, for

each $s \in S$, we compute a primitive upper bound for the sum $\sum_{i \in I \setminus I^0(y)} \sum_{j \in J_s} c''_{ij} x_{ij}$ as $\bar{C}_s = \sum_{j \in J_s} \max_{i \in I \setminus I^0(y)} c''_{ij}$. Without loss of generality, let us assume that scenarios are ordered in such a way that $s' = l$ and $\bar{C}_1 \geq \bar{C}_2 \geq \dots \geq \bar{C}_{l-1}$. Let $r \in S$ be such an index that $\sum_{s < r} p_s < p_0$ and $\sum_{s \leq r} p_s \geq p_0$. If $r = l$, then all the scenarios must be active, and the big- M term is unnecessary. Otherwise, we set $M_{s'} = \bar{C}_r$. In the following lemma, we show that this value is the tightest one which does not affect the set of feasible solutions of the estimating problem.

Lemma 3 *For any feasible solution (x, X, δ, C) of the problem $\mathcal{B}(y)$ such that $\delta_{s'} = 0$, it holds $C \leq \bar{C}_r$. There exists a feasible solution (x', X', δ', C') of the problem $\mathcal{B}(y)$ such that $\delta_{s'} = 0$ and $C' = \bar{C}_r$.*

Proof From feasibility of (x, X, δ, C) , we have $\sum_{s \in S} p_s \delta_s \geq p_0$. Let $C_s = \sum_{i \in I} \sum_{j \in J_s} c_{ij} x_{ij}$. The following relations hold:

$$C \leq \min_{s|\delta_s=1} C_s \leq \min_{s|\delta_s=1} \bar{C}_s \leq \bar{C}_r.$$

The first inequality follows from (19). The second one is true due to $C_s \leq \bar{C}_s$ for any $s \in S$. And the last relation results from $\delta_{s'} = 0$ and the choice of r .

The solution (x', X', δ', C') from the second part of the lemma statement can be constructed directly. Let $x'_i = 1$ for all $i \in I \setminus I^0(y)$. Further, we set $\delta'_s = 1$ if $s \leq r$ and $\delta'_s = 0$ otherwise. For $s \leq r$ and any given $j \in J_s$, we set $x'_{k_j} = 1$, where $k = \arg \max_{i \in I \setminus I^0(y)} c''_{ij}$. It is easy to see that $C' = \min_{s|\delta'_s=1} \sum_{j \in J_s} \sum_{i \in I} c''_{ij} x'_{ij} = \bar{C}_r$. □

The lemmas above imply the following theorem.

Theorem 1 *The optimum of the problem $\mathcal{B}(y)$ is an upper bound for the value $\max_{x, \delta} f_1(x, \delta)$, where $x \in P(y)$ and $f_2(x, \delta) \geq p_0$.*

3.3 Relaxation of a large MIP and a cutting-plane method

The model $\mathcal{B}(y)$ can be reformulated as a MIP with an exponential number of variables. Denote the set of all subsets of S with \mathcal{R} . For each $R \in \mathcal{R}$, we introduce a new boolean variable u_R equal to one, if the set of active scenarios is equal to R , and zero otherwise. Additionally, we define a $(0, 1)$ -matrix (a_{sR}) , $s \in S$, $R \in \mathcal{R}$ such that a_{sR} is equal to one, if $s \in R$, and zero otherwise. By using the above definitions, we can replace the variables δ_s , $s \in S$, as $\delta_s = \sum_{R \in \mathcal{R}} a_{sR} u_R$. An additional constraint $\sum_{R \in \mathcal{R}} u_R = 1$, ensuring that the only set of active scenarios must be chosen, finalizes the reformulation.

Numerical experiments show that the introduced large MIP is tighter than $\mathcal{B}(y)$ in terms of integrality gap (Melnikov and Beresnev 2016). Direct solving such a large model is likely to be inefficient, but its linear relaxation can be considered as an alternative way to calculate an upper bound in a branch-and-bound scheme. We obtain an optimum of the relaxation by solving its dual problem by a cutting-plane method (Briant et al. 2008). The dual program is written as follows:

$$\min_{(\alpha_s), (\beta_j), (\gamma_{ij}), \eta, \lambda} \left(- \sum_{i \in I^1(\mathbf{y})} f_i + \lambda - p_0 \eta + \sum_{s \in S} M_s \alpha_s + \sum_{i \in I^1(\mathbf{y})} \sum_{j \in J} \gamma_{ij} \right) \tag{23}$$

$$\sum_{j \in J} \gamma_{ij} \leq f_i, \quad i \in I^*(\mathbf{y}); \tag{24}$$

$$\beta_j + \gamma_{ij} \geq c''_{ij} \alpha_s, \quad i \in I \setminus I^0(\mathbf{y}), \quad s \in S, \quad j \in J_s; \tag{25}$$

$$\sum_{s \in S} \alpha_s = 1; \tag{26}$$

$$\lambda \geq \sum_{s \in S} a_{sR} \left(p_s \eta + \sum_{j \in J_s} \beta_j - M_s \alpha_s \right), \quad R \in \mathcal{R}; \tag{27}$$

$$\alpha_s, \beta_j, \gamma_{ij}, \eta \geq 0, \quad i \in I \setminus I^0(\mathbf{y}), \quad s \in S, \quad j \in J. \tag{28}$$

Let $D(\mathcal{R}') = ((\alpha_s), (\beta_j), (\gamma_{ij}), \eta, \lambda)$ be an optimal solution of the problem (23)–(28), where an exponentially large index set \mathcal{R} in (27) is replaced by its relatively small subset $\mathcal{R}' \subseteq \mathcal{R}$. If $D(\mathcal{R}')$ satisfies (27) for all $R \in \mathcal{R}$, it is an optimal solution of the dual problem and provides a required upper bound. Otherwise, there exists a $(0, 1)$ -vector δ such that

$$\sum_{s \in S} \delta_s \left(p_s \eta + \sum_{j \in J_s} \beta_j - M_s \alpha_s \right) > \lambda. \tag{29}$$

To check if there exists such a vector, we solve the following problem:

$$\max_{\delta} \sum_{s \in S} w_s \delta_s \tag{30}$$

$$\sum_{s \in S} \delta_s p_s \geq p_0 \tag{31}$$

$$\delta_s \in \{0, 1\}, \quad s \in S, \tag{32}$$

where $w_s = p_s \eta + \sum_{j \in J_s} \beta_j - M_s \alpha_s$.

Given an optimal solution δ^* of the problem (30)–(32), if the inequality $\sum_{s \in S} w_s \delta_s^* \leq \lambda$ holds, then the solution $D(\mathcal{R}')$ satisfies (27) for any $R \in \mathcal{R}$. Otherwise, one of constraints violated by $D(\mathcal{R}')$ corresponds to the set of scenarios $\{s \in S \mid \delta_s^* = 1\}$. We include it into \mathcal{R}' and get back to solving the dual problem with a new constraint of type (27).

Thus, the cutting-plane (CP) scheme to calculate an upper bound for the value $\max_{x, \delta} f_1(x, \delta)$ is an iterative process. On each iteration, we solve a restricted dual problem and check if its optimal solution is feasible in the initial dual problem. In the case of an affirmative answer, a valid upper bound is obtained, and the procedure terminates. Otherwise, a cut is generated by solving a knapsack-type problem (30)–(32), and a new iteration begins.

To guarantee feasibility of the restricted dual problem at the first iteration, we set \mathcal{R}' to contain a randomly chosen subset $S' \subseteq S$ such that $\sum_{s \in S'} p_s \geq p_0$.

3.4 Equiprobable scenarios and estimating problem $\mathcal{B}'(\mathbf{y})$

The case of equiprobable scenarios, i.e., when $p_s = 1/l$ for all $s \in S$, is an important case of BCompFLP arising, for instance, in sample approximations of stochastic programming problems (Shapiro 2008). Given \mathbf{x} , a set of active scenarios maximizing $f_1(\mathbf{x}, \delta)$ contains exactly $\lceil lp_0 \rceil$ elements with the greatest income. Let us introduce variables (C_s) , $s \in S$, storing the value of income in the corresponding scenario. The set of most profitable active scenarios can be chosen by solving the following knapsack problem:

$$\max_{(\delta_s)} \sum_{s \in S} C_s \delta_s, \tag{33}$$

$$\sum_{s \in S} \delta_s = \lceil lp_0 \rceil; \tag{34}$$

$$\delta_s \in \{0, 1\}, \quad s \in S. \tag{35}$$

Notice that this model is equivalent to its linear relaxation since the simplex method obtains an optimal solution with no fractional components. Its linear relaxation can be equivalently rewritten in the form of complementary slackness conditions (Audet et al. 1997) which are nonlinear initially. After linearizing these expressions, we introduce new constraints into the estimating problem and obtain its another reformulation $\mathcal{B}'(\mathbf{y})$ looking as follows:

$$\max_{(x_i), (x_{ij}), (\delta_s), (C_s), C} \left(- \sum_{i \in I^1(\mathbf{y})} f_i - \sum_{i \in I^*(\mathbf{y})} f_i x_i + C \right), \tag{36}$$

$$x_i \geq x_{ij}, \quad i \in I^*(\mathbf{y}), \quad j \in J; \tag{37}$$

$$\sum_{i \in I \setminus I^0(\mathbf{y})} x_{ij} \leq \delta_s, \quad s \in S, \quad j \in J_s; \tag{38}$$

$$C_s = \sum_{i \in I \setminus I^0(\mathbf{y})} \sum_{j \in J_s} c''_{ij} x_{ij}, \quad s \in S; \tag{39}$$

$$C \leq C_s + M_s(1 - \delta_s), \quad s \in S; \tag{40}$$

$$\sum_{s \in S} \delta_s = \lceil lp_0 \rceil; \tag{41}$$

$$u_s \leq M \delta_s, \quad s \in S; \tag{42}$$

$$u_s + w \geq C_s, \quad s \in S; \tag{43}$$

$$u_s + w \leq C_s + M(1 - \delta_s), \quad s \in S; \tag{44}$$

$$x_i = y_i, \quad i \in I^0(\mathbf{y}) \cup I^1(\mathbf{y}); \tag{45}$$

$$x_i, \delta_s, x_{ij} \in \{0, 1\}; \quad C_s, C \geq 0, \quad i \in I, \quad j \in J, \quad s \in S. \tag{46}$$

Here, w and (u_s) , $s \in S$ are dual variables for constraints (34) and (35), respectively. Variables (δ_s) , $s \in S$ are boolean in this model due to linearization purposes.

4 Numerical experiments

In this section, we study performance of the suggested procedures and investigate some properties of the solutions obtained by the ε -constraint method. Firstly, we consider a family of randomly generated instances of BCompFLP, where scenarios correspond to different variations of the transportation network. These instances are used to compare branch-and-bound methods equipped with different upper bound procedures and study a Pareto frontier of the BCompFLP. Further, we come to an important application of the BCompFLP related to the issue of stability of competitive location model's solution with respect to perturbations of numerical data, and, namely, perturbations of the profit matrix. We concentrate on studying the question about does consideration of multiple demand scenarios help to obtain more "stable" solutions in a situation of uncertain demand parameters values.

All the calculations are performed by a workstation with two six-core processors Intel Xeon X5675 3.07 GHz and 96 GB RAM. The methods are implemented in C# programming language and get solutions of arising optimization problems from Gurobi 8.0 MIP solver (Gurobi Optimization 2016).

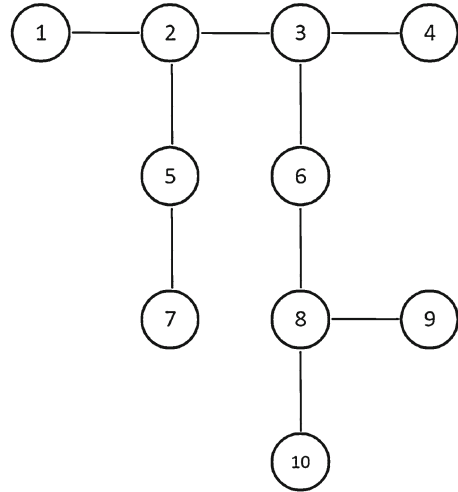
4.1 Instances based on transportation network variations

Consider a graph with a set of vertices divided into two disjoint non-empty subsets, V_1 and V_2 . Nodes of the subset V_1 form a connected subgraph and represent potential locations to open facilities. The set of the customers is represented by vertices of the set V_2 . Each vertex $v \in V_2$ has exactly four edges connecting it with randomly chosen vertices from V_1 . Lengths of these edges are generated randomly for each of the scenarios. Preferences of the customer located in the vertex $v \in V_2$ are based on distances from v to vertices from V_1 : The shorter the distance, the more preferable the facility located in the corresponding vertex. Thus, scenarios are different from each other only in the preferences of the customers.

Each customer $j \in J$ is assumed to have a budget b_j which is the same for all the scenarios. Income from serving the customer j does not depend on the serving facility and equals b_j for both the Leader and the Follower, i. e. $c_{ij} = d_{ij} = b_j$, for all $i \in I$. Values of the parameters are independently chosen with uniform distribution from the corresponding integer segments: f_i from $\{7, 8, \dots, 13\}$; g_i from $\{4, 5, \dots, 10\}$; b_j from $\{1, 2, \dots, 5\}$.

For all the generated test instances, we have $|V_2| = 30$ and 15 scenarios in total. Thus, we have a set J containing 450 customers, where each of the customers is represented by fifteen copies having different preferences resulting from variations of distance matrix corresponding to each of the scenarios.

Fig. 1 Network with $m = 10$ vertices



We consider four classes of instances that differ from each other in a topology of the network formed by vertices from V_1 . These classes are named in accordance with the topology of the network: Tree, Plane_{ℓ_1} , Sphere, and Grid.

The class Tree contains instances where connections between vertices from V_1 form a tree. Each of the connections is represented either by a *short* or a *long* edge. The length of a short edge is an integer between 1 and 15, while the long edge's length varies between 100 and 150. An edge is short with probability 0.9 and is long otherwise. Experiments with relative competitive location models show that this topology specifies a relatively simple structure of preferences allowing to solve instance faster (Mel'nikov 2014).

The second class considered is called Plane_{ℓ_1} because the set V_1 is represented by points of the unit square. The coordinates of points are chosen independently with uniform distribution. All the vertices are connected, and the length of connection is computed by ℓ_1 -metric as a sum of absolute values of coordinate differences.

Similarly to the class Plane_{ℓ_1} , in the class Sphere, all the vertices from V_1 are connected with each other. Being represented by points of a sphere, whose polar and azimuthal angles are chosen randomly with uniform distribution, these vertices are less likely to contain “peripheral” and “internal” ones. A distance between two vertices is computed along a great circle passing through the points representing them.

Finally, in the class **Grid**, the network organized by vertices from V_1 is a rectangular grid. In this structure, the vertices can be thought to be elements of a matrix. Each element is connected with its nearest neighbors located at the same column or the same row. Thus, all the vertices have up to four edges. The edges' lengths are randomly chosen integers from 1 to 3.

To illustrate the model's outcomes on instances generated, let us consider a transportation network shown in Fig. 1. It represents the connections between the ten vertices available to establish facilities forming the set V_1 . On the base of the network,

Table 1 Weakly efficient solutions of the instance with $m = 10$ vertices

s	p_s	p_0				
		0.2	0.4	0.6	0.8	1
1	0.02	-3	-5	-10	-1	0
2	0.05	4	10	3	7	0
3	0.1	1	2	5	10	0
4	0.05	1	18	1	14	0
5	0.07	9	11	5	2	0
6	0.02	4	7	8	-1	0
7	0.1	-6	-7	-12	4	0
8	0.07	9	6	6	1	0
9	0.02	5	14	20	14	0
10	0.05	17	3	9	7	0
11	0.07	19	16	11	6	0
12	0.1	0	4	3	3	0
13	0.1	-1	0	7	3	0
14	0.1	19	13	19	10	0
15	0.07	6	19	11	8	0
$f_1(\mathbf{x}, \delta)$		17	10	5	3	0
$f_2(\mathbf{x}, \delta)$		0.22	0.44	0.61	0.8	1
$i \in I x_i = 1$		3, 10	2, 3, 8, 10	1, 3, 8, 10	3, 8	-

we obtain an instance from the class Tree. All the parameters' values of the instance are generated in the manner described earlier.

Table 1 provides the characteristics of weakly efficient solutions obtained for different values of the threshold p_0 . The right part of the table has five columns corresponding to the solutions with the value of p_0 given in the second row of the table header. The main part of the table is divided into 15 rows representing the demand scenarios. For each scenario s , the probability of its realization, p_s , and the Leader's profit for each of the solutions are given. The profit values in active scenarios are shown in bold. In the bottom part of the table, the values of the Leader's objective functions are given. The last row provides the vertex numbers where the Leader's facilities are established.

Note that the Leader has no location decision that guarantees a positive profit in all the scenarios since the solution for $p_0 = 1$ is an empty one. More risky solutions may result in a higher guaranteed profit. At the same time, possible losses in non-active scenarios are not taken into account, so more risky solutions can perform better in a worst-case scenario than the less risky ones. We observe this for the solutions with $p_0 = 0.4$, where the Leader gets a negative profit of -7 , and $p_0 = 0.6$ with the profit of -12 in the worst scenario.

4.2 Choosing the upper bound procedure

In this subsection, we aim to compare the performances of a branch-and-bound algorithm equipped with different upper bound procedures. For brevity, we refer these algorithms as $A_{\mathcal{B}}$, $A_{\mathcal{B}'}$, and A_{CP} , respectively. In Melnikov and Beresnev (2016), we studied the procedures independently and concluded that the model \mathcal{B}' performs better than \mathcal{B} on instances with equiprobable scenarios. Additionally, the upper bound provided by CP takes less time than solving \mathcal{B} and \mathcal{B}' , while the value obtained is more accurate than optimums of these models' relaxations. It is necessary to mention that an optimization library Microsoft Solver Foundation 3.1 used in Melnikov and Beresnev (2016) to solve MIPs turned up to be significantly slower than the latest versions of commercial optimization software. Thus, our interest is to investigate how the suggested models perform when being installed into the branch-and-bound framework and solved by very recent optimization tools.

In Tables 2 and 3, we present results of numerical experiments with instances where scenarios have equal and randomly generated probabilities, correspondingly. The experiment was organized in the following way. For each of the classes considered, instances with $|I| = m = 9, 16, \text{ and } 25$ were generated. Instances of the class *Grid* are based on 3×3 , 4×4 , and 5×5 grids formed by vertices from V_1 , respectively. For each instance, four values of threshold p_0 were considered: 0.25, 0.5, 0.75, and 1. Computations had been stopped when reaching the time limit of 30 min. The tables contain the following columns:

- $T_{\mathcal{B}}$, $T_{\mathcal{B}'}$, and T_{CP} are for computation times of $A_{\mathcal{B}}$, $A_{\mathcal{B}'}$, and A_{CP} , respectively. In a case when the time limit is reached and the computations are not finished, we provide the progress of the algorithm written in italics. The progress is computed as a percentage of leaves of a branching tree that are explored by the algorithm.
- $N_{\mathcal{B}}$, $N_{\mathcal{B}'}$, and N_{CP} are for the total number of nodes of the branching tree explored by the corresponding algorithm.
- Opt is for the optimal value of the function f_1 ; for instances which are not solved by some of the algorithms, we give the best objective value obtained. The modification of the method delivered this value is given in brackets when the methods were terminated with different results.
- B is an initial upper bound value computed by $A_{\mathcal{B}}$ and $A_{\mathcal{B}'}$.
- CP is an initial upper bound value computed by A_{CP} .

As we can see from Tables 2 and 3, the structure of the graph specifying customers preferences significantly affects computational time needed to solve the instance. Similarly to the previously studied competitive location models (Mel'nikov 2014), instances based on trees are simpler than the ones based on graphs with higher connectivity. Oppositely, grids and similar regular graph structures induce instances that are more difficult for our approach because the intricate structure of preferences makes the upper bound computed by using the models based on estimating subsets overly optimistic.

Another general observation considers the role of the parameter p_0 . For all the considered instances, the case $p_0 = 1$ is the simplest one. One of the reasons for this is that big-M terms vanish from the estimating problem formulations in this case. The

Table 2 Branch-and-bound: equiprobable scenarios

m	p_0	$T_{\mathcal{B}}$	$T_{\mathcal{B}'}$	T_{CP}	$N_{\mathcal{B}}$	$N_{\mathcal{B}'}$	N_{CP}	Opt	B	CP
<i>Tree</i>										
9	0.25	4	4	30	73	82	357	34	43	75
9	0.5	3	4	19	48	54	267	24	35	59
9	0.75	3	3	14	91	84	261	14	31	43
9	1	3	3	2	67	67	77	6	20	25
16	0.25	7	8	222	89	74	2k	21	27	70
16	0.5	4	5	120	46	47	1k	14	23	51
16	0.75	12	11	73	132	136	829	7	15	32
16	1	11	12	12	75	76	91	0	4	9
25	0.25	490	502	45%	4k	4k	8k	12	25	66
25	0.5	690	757	73%	5k	6k	10k	5	14	46
25	0.75	315	328	98%	2k	2k	11k	0	8	26
25	1	21	20	21	74	74	92	0	2	6
<i>Plane$_{\ell_1}$</i>										
9	0.25	12	14	52	255	279	548	21	38	69
9	0.5	10	11	29	214	222	443	15	30	55
9	0.75	11	10	21	267	220	351	10	24	41
9	1	6	6	6	145	157	184	6	22	27
16	0.25	121	120	827	1k	1k	5k	14	31	65
16	0.5	235	216	592	2k	2k	5k	7	26	50
16	0.75	204	192	394	2k	2k	4k	2	19	35
16	1	42	37	33	385	368	413	0	14	19
25	0.25	56%	52%	24%	13k	13k	7k	16 (CP)	37	75
25	0.5	37%	38%	25%	11k	12k	9k	8 (CP)	31	58
25	0.75	65%	65%	28%	12k	12k	11k	2	25	41
25	1	99%	99%	1533	11k	11k	13k	0	21	25
<i>Sphere</i>										
9	0.25	15	12	61	359	227	537	22	38	69
9	0.5	11	12	33	278	281	513	15	29	55
9	0.75	8	9	22	217	212	451	10	26	42
9	1	6	6	5	143	143	156	6	20	27
16	0.25	80	94	549	876	975	3k	18	36	67
16	0.5	92	93	384	1k	1k	3k	11	32	53
16	0.75	87	86	212	854	881	2k	6	23	39
16	1	36	32	37	351	349	472	1	15	20
25	0.25	40%	34%	12%	12k	12k	7k	16 (\mathcal{B})	43	77
25	0.5	28%	22%	19%	13k	12k	9k	9 (\mathcal{B}')	36	62
25	0.75	32%	32%	19%	12k	12k	12k	0	32	47
25	1	87%	86%	95%	11k	12k	16k	0	25	31

Table 2 continued

m	p_0	$T_{\mathcal{B}}$	$T_{\mathcal{B}'}$	T_{CP}	$N_{\mathcal{B}}$	$N_{\mathcal{B}'}$	N_{CP}	Opt	B	CP
<i>Grid</i>										
9	0.25	15	14	83	323	320	800	17	37	68
9	0.5	16	18	52	369	388	698	10	26	53
9	0.75	13	13	25	266	265	535	5	22	38
9	1	5	5	5	127	127	144	0	14	23
16	0.25	215	219	1490	2k	2k	9k	15	34	68
16	0.5	320	327	976	3k	3k	7k	8	28	53
16	0.75	331	346	469	3k	3k	4k	4	23	38
16	1	101	108	116	1k	987	1k	0	17	20
25	0.25	89%	93%	25%	11k	13k	7k	11($\mathcal{B}, \mathcal{B}'$)	33	70
25	0.5	84%	85%	55%	12k	12k	9k	7(CP)	29	54
25	0.75	1021	951	73%	7k	7k	10k	4	23	38
25	1	472	456	393	3k	3k	3k	0	17	22

enumeration speeds up due to a faster computation of the upper bound. Additionally, the method, using CP to calculate the upper bound, shows itself more competitive for these instances.

When analyzing Table 2, one would notice that the difference between the upper bounds presented in columns B and CP is significant. The value provided by CP is around two times higher than the one provided by MIP. It results in a weak performance of the branch-and-bound equipped with CP. For **Grid** instances with $m = 9$ and $p_0 = 0.25, 0.5$, A_{CP} explores around seven hundreds of nodes which is more than the total number of possible location decisions. Another notable effect is that the number of nodes explored by CP is significantly higher than the one explored by \mathcal{B} and \mathcal{B}' for the class Tree. For other classes, which are more difficult in terms of computational time, the situation is the opposite. It results from the behavior of the cutting-plane method which needs fewer iterations to converge when the structure of preferences is relatively simple. $A_{\mathcal{B}}$ and $A_{\mathcal{B}'}$ perform similarly, and the data do not give priority to any of them.

In Table 3, scenarios with different probabilities of realization were considered. To generate the probabilities, we choose at random a set of integers $\{\rho_s\}$ from the interval $\{1, \dots, 4\}$. For the scenario $s \in S$, we set $p_s = \rho_s / \sum_{s \in S} \rho_s$. The model \mathcal{B}' cannot be applied in this situation, and our goal is to compare \mathcal{B} and CP with each other. As we can see, the key points regarding performances of the methods noted from the analysis of Table 2 stay the same in a case of different probabilities. Thus, the modification $A_{\mathcal{B}}$ showed itself the most universal and capable among the considered ones, and the following computations are performed using this tool.

Table 3 Branch-and-bound: scenarios with different probabilities

m	p_0	T_B	T_{CP}	N_B	N_{CP}	Opt	B	CP
<i>Tree</i>								
9	0.25	7	39	103	368	34	45	76
9	0.5	7	25	121	324	27	37	61
9	0.75	3	15	77	246	17	32	46
9	1	3	3	67	85	6	20	25
16	0.25	13	326	185	2k	21	27	71
16	0.5	14	257	196	2k	11	21	53
16	0.75	15	181	196	2k	6	15	35
16	1	11	11	75	101	0	4	9
25	0.25	450	44%	3k	7k	12 (B)	21	66
25	0.5	356	67%	2k	8k	6 (B)	13	47
25	0.75	272	93%	2k	10k	1	9	29
25	1	21	21	74	110	0	2	6
<i>Planeℓ_1</i>								
9	0.25	10	60	207	480	21	36	69
9	0.5	9	33	212	481	15	30	55
9	0.75	7	20	163	342	11	25	41
9	1	6	5	161	145	6	22	27
16	0.25	277	1229	3k	8k	10	29	66
16	0.5	278	655	3k	5k	7	24	52
16	0.75	206	505	2k	4k	2	20	37
16	1	32	35	379	422	0	14	19
25	0.25	60%	19%	12k	6k	15 (B)	42	76
25	0.5	57%	20%	12k	8k	9	31	61
25	0.75	54%	27%	12k	9k	4 (CP)	28	46
25	1	1788	1616	11k	12k	0	21	25
<i>Sphere</i>								
9	0.25	9	46	186	495	22	34	69
9	0.5	7	38	155	450	16	29	55
9	0.75	8	17	147	307	12	26	42
9	1	6	6	141	178	6	20	27
16	0.25	109	663	1k	4k	16	36	67
16	0.5	106	494	1k	3k	11	31	54
16	0.75	109	280	1k	2k	6	25	41
16	1	35	36	375	442	1	15	20
25	0.25	36%	16%	12k	6k	17 (B)	47	78
25	0.5	59%	13%	12k	8k	11 (B)	37	64
25	0.75	29%	18%	11k	9k	1 (B)	32	50
25	1	87%	87%	11k	13k	0	25	31

Table 3 continued

m	p_0	T_B	T_{CP}	N_B	N_{CP}	Opt	B	CP
<i>Grid</i>								
9	0.25	16	63	370	652	18	33	68
9	0.5	15	58	375	736	10	26	54
9	0.75	20	37	435	649	4	22	40
9	1	5	5	127	165	0	14	23
16	0.25	217	933	2k	5k	16	38	69
16	0.5	280	841	2k	6k	9	28	55
16	0.75	309	743	3k	6k	4	23	41
16	1	103	110	957	1k	0	17	20
25	0.25	66%	25%	12k	7k	6 (B)	33	70
25	0.5	77%	44%	12k	8k	7 (CP)	29	56
25	0.75	1153	58%	8k	9k	4 (B)	23	40
25	1	447	398	3k	3k	0	17	22

4.3 Dealing with uncertainty in the CompFLP model

The model BCompFLP provides a rich set of instruments for making a decision in a situation of uncertainty. Let us back to the basic CompFLP model (Beresnev 2014), which can be regarded as BCompFLP with a single scenario. In the benchmark library (Discrete location problems 2018), a series of instances of CompFLP with known optimal solutions is collected. Our interest in this subsection is to investigate what happens with these solutions when the elements of the profit matrix slightly change, and how the BCompFLP model can support decisions when profits are stochastic.

Consider the class of instances A20 from Discrete location problems (2018). There, similarly to previously introduced instances, both the set of customers and the set of potential facility locations are represented by vertices of a randomly generated graph. The Leader's fixed costs are set to 40 for all $i \in I$, whereas the Follower's ones are randomly chosen from 25 to 35. Instances have names of a form 'a20-xx,' where the postfix 'xx' is formed by an individual number of instance.

In the class A20, profit matrices of both players are equal, i. e. $c_{ij} = d_{ij}$ for all $i \in I, j \in J$, and their elements are taken from the integer interval $\{10, \dots, 20\}$. Let us assume these elements are determined with an accuracy of 10%. To take into account possible variations of these parameters, we generate a list of scenarios where the same set of customers is attributed to different realizations of the profit matrix. Given the facility i and the customer j , we set the corresponding value of the profit matrix to a random positive integer from the interval $[0.9c_{ij}, 1.1c_{ij}]$ independently for each of the scenarios.

In our experiments, we considered instances of BCompFLP with 10, 15, and 20 realizations of the profit matrix (scenarios), whereas the total number of possible variations is sufficiently higher. Each of these variations determines the value of the Leader's profit provided that the Follower's reaction is rational. Thus, each Leader's decision on facilities' location is characterized by a distribution of the profit value. To

express this distribution, given the Leader's location, we sample $\nu = 1000$ variations of the profit matrix and compute the corresponding value of the Leader's profit.

The graphs in Fig. 2 show the profit distribution for five solutions computed by the ϵ -constraint method for $p_0 = 0.2, 0.4, 0.6, 0.8,$ and 1 . The value of p_0 is shown on the horizontal axis, whereas the vertical axis corresponds to the profit value.

A distribution is visualized by both a box and a violin-type plots. Boxes' whiskers show minimal and maximal value among ν computed values of the profit. Lower and upper sides of a box show the first and the third quartile for this set of values. Finally, a band inside a box represents a median value. A general look of a distribution function is shown by shaded vertically oriented shapes, where thicker parts of a shape correspond to more probable values of profit.

On the figures, two line graphs are depicted as well. The light-gray line shows the value $f_1^*(p_0)$ which is an optimal value of f_1 in the BCompFLP model with $f_2 \geq p_0$. To compare this value with the real value of the profit guaranteed with a probability p_0 ,

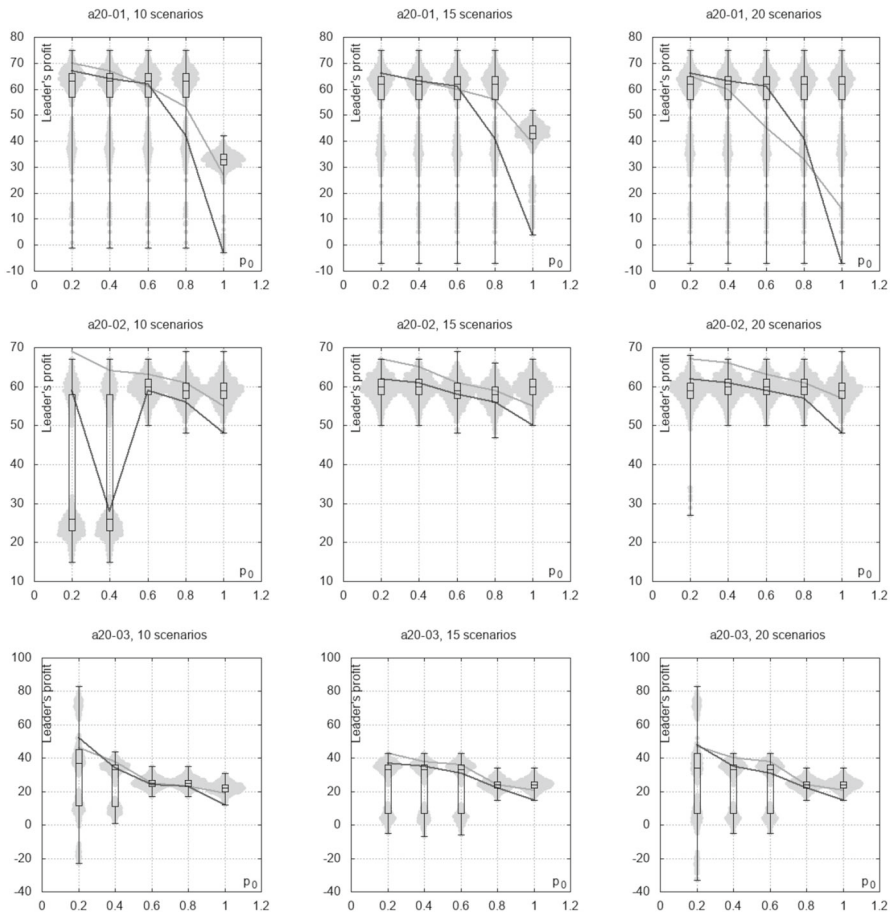


Fig. 2 Distribution of the Leader's profit for solutions obtained for different values of p_0

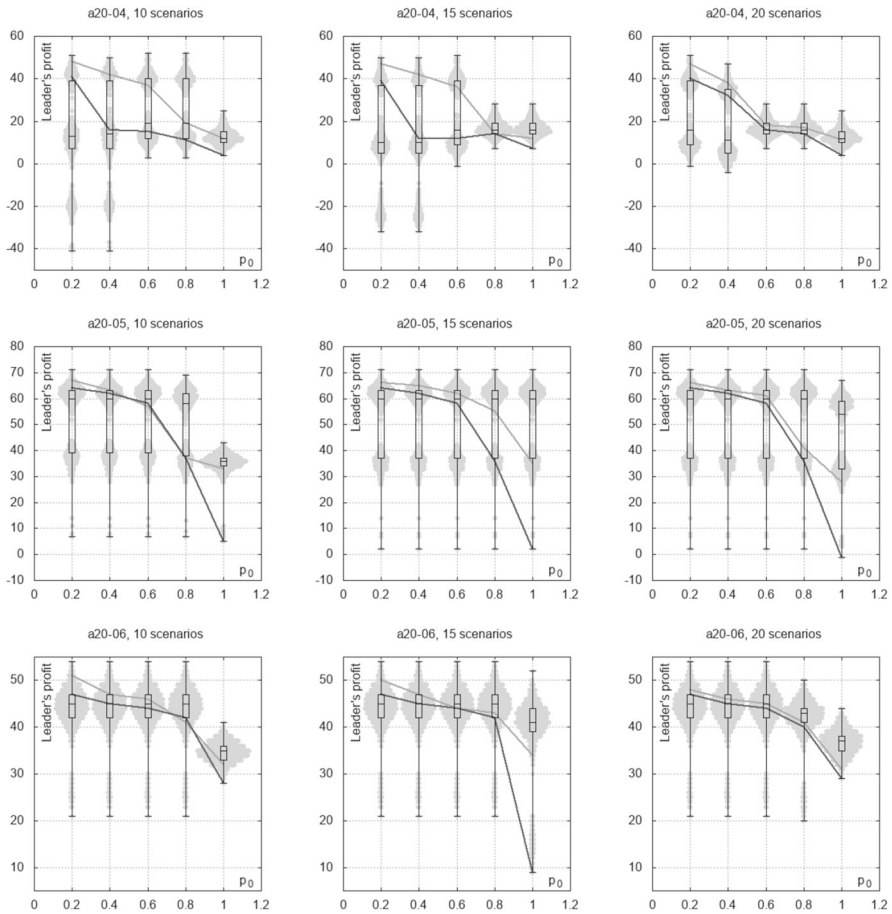


Fig. 2 continued

we added a dark-gray line showing the value $f_1^v(p_0)$ which is the $\lceil(1 - p_0)\nu\rceil$ -th-order statistic for a set of profit values computed for ν sampled profit matrices.

From the graphs presented, we notice that the density function of f_1 is often multimodal. Given the Leader's location, the optimal solution of the Follower and, consequently, the set of customers captured by the Leader depend on the realization of the profit matrix. For the numerical data under consideration, the function f_1 takes similar values for variations of the profit matrix inducing the same optimal solution of the Follower. At the same time, changing the Follower's solution may influence f_1 heavily. It results in that the shape of the distribution is represented by several clusters. Values from the same cluster correspond to a single location of the parties and differ from each other due to deviations of the profit matrix.

Distribution of the profit value tends to be more concentrated for solutions obtained with higher p_0 . At the same time, these solutions are often less preferable than ones computed for lower p_0 . It is true for the instance a20-01, 10 scenarios, $p_0 = 1$, for

a20-04, 20 scenarios, $p_0 = 1$, and in several other cases. We suppose that here we observe a phenomenon similar to so-called overfitting in the machine learning area. It manifests itself in that the solution strongly adapts to the specific of the numeric data appeared in the scenarios considered and performs worse on the data unseen before.

One could suggest that $f_1^*(p_0)$ converges to $\lim_{v \rightarrow +\infty} f_1^v(p_0)$ when the number of scenarios considered in the BCompFLP model tends to infinity. The graphs support this suggestion. Comparison of the graphs built for a different number of scenarios shows that the convergence rate decreases when p_0 grows. The values $f_1^*(p_0)$ and $f_1^v(p_0)$, represented by light and dark lines, respectively, tend to be closer for smaller p_0 and could differ significantly for p_0 close to one. The question of convergence of the sample average approximation to the actual solution of the stochastic model with expectation objective function is considered in Shapiro (2008). To our knowledge, the case of quantile objective function remains unstudied.

5 Conclusion

In this paper, we considered a competitive location model BCompFLP where the Leader makes a decision in a situation where multiple demand scenarios are possible. It allows taking into account uncertainty in all the aspects characterizing the demand such as the number of customers, their preferences, and income from serving them. We formulate the Leader's problem as a bi-objective bi-level program aiming to maximize both the Leader's profit and the probability to get it. Efficient solutions of the model provide information about trade-offs between revenues and risks as well as about regrets which are possible in concrete circumstances. Thus, BCompFLP is a powerful tool supporting a decision-making process in situations meeting assumptions of the model.

To approximate the Pareto frontier of the problem, we developed an ε -constraint method providing a subset of weakly efficient solutions. In the method, a series of bi-level problems with a single objective is solved by a branch-and-bound algorithm. Three upper bound procedures built on the basis of a specially constructed estimating problem in a form of MIP were built in the algorithm and compared with each other in numerical experiments.

Further, we performed a series of experiments aiming to investigate relationships between the stochastic competitive location problem with uncertain profits and its sample approximation that can be written as a BCompFLP. The data show that the value of guaranteed profit in the sample approximation BCompFLP converges to the corresponding profit value in the stochastic model. At the same time, the number of scenarios that are to be included in the BCompFLP model depends on the probability p_0 of obtaining the guaranteed profit. The higher p_0 , the more scenarios are needed to get a good approximation of the guaranteed profit.

Another effect observed in our experiments is analogous to overfitting in the machine learning area. It manifests itself in that solutions obtained for high probability p_0 are sometimes worse in terms of profit distribution than the ones corresponding to lower p_0 . As an instance, a location obtained for $p_0 = 1$ exploits all the specific of

scenarios when aiming to optimize the profit in all of them. It leads to a decrease in the profit in unseen scenarios and worsening the profit distribution.

Our future research is oriented on developing models for stochastic competitive location and methods to solve them.

Acknowledgements This research is supported by the Russian Science Foundation (Grant 17-11-01021).

References

- Alekseeva E, Kochetov Y, Talbi EG (2017) A matheuristic for the discrete bilevel problem with multiple objectives at the lower level. *Int Trans Oper Res* 24(5):959–981. <https://doi.org/10.1111/itor.12268>
- Aras N, Küçükaydın H (2017) Bilevel models on the competitive facility location problem. In: Mallozzi L, D'Amato E, Pardalos PM (eds) *Spatial interaction models: facility location using game theory*. Springer, Berlin, pp 1–19. <https://doi.org/10.1007/978-3-319-52654-6>
- Ashtiani M (2016) Competitive location: a state-of-art review. *Int J Ind Eng Comput* 7(1):1–18. <https://doi.org/10.5267/j.ijiec.2015.8.002>
- Audet C, Hansen P, Jaumard B, Savard G (1997) Links between linear bilevel and mixed 0–1 programming problems. *J Optim Theory Appl* 93(2):273–300
- Bandyapadhyay S, Banik A, Das S, Sarkar H (2015) Voronoi game on graphs. *Theor Comput Sci* 562(C):270–282. <https://doi.org/10.1016/j.tcs.2014.10.003>
- Beresnev V (2013) Branch-and-bound algorithm for competitive facility location problem. *Comput Oper Res* 40:2062–2070
- Beresnev VL (2014) On the competitive facility location problem with a free choice of suppliers. *Autom Remote Control* 75(4):668–676. <https://doi.org/10.1134/S0005117914040079>
- Briant O, Lemaréchal C, Meurdesoif P, Michel S, Perrot N, Vanderbeck F (2008) Comparison of bundle and classical column generation. *Math Program* 113(2):299–344. <https://doi.org/10.1007/s10107-006-0079-z>
- Canovas L, Garcia S, Labbe M, Marin A (2007) A strengthened formulation for the simple plant location problem with order. *Oper Res Lett* 35:141–150
- Davydov I, Kochetov Y, Plyasunov A (2014) On the complexity of the $(r|p)$ -centroid problem in the plane. *Top* 22(2):614–623. <https://doi.org/10.1007/s11750-013-0275-y>
- Dempe S (2002) *Foundations of Bilevel Programming*. Kluwer Academic Publisher, Dordrecht
- Discrete location problems. *Benchmarks library* (2018). <http://math.nsc.ru/AP/benchmarks/english.html>
- Drezner T, Drezner Z, Kalczyński P (2015) A leader-follower model for discrete competitive facility location. *Comput Oper Res* 64:51–59. <https://doi.org/10.1016/j.cor.2015.04.012>
- Ehrgott M (2006) *Multicriteria optimization*. Springer, Berlin
- Emelogu A, Chowdhury S, Marufuzzaman M, Bian L, Eksioğlu B (2016) An enhanced sample average approximation method for stochastic optimization. *Int J Prod Econ* 182:230–252. <https://doi.org/10.1016/j.ijpe.2016.08.032>
- Farahani RZ, SteadieSeifi M, Asgari N (2010) Multiple criteria facility location problems: a survey. *Appl Math Model* 34(7):1689–1709. <https://doi.org/10.1016/j.apm.2009.10.005>
- Fernández J, Hendrix EM (2013) Recent insights in huff-like competitive facility location and design. *Eur J Oper Res* 227:581–584
- Fernández J, Tóth B (2009) Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound methods. *Comput Optim Appl* 42(3):393–419. <https://doi.org/10.1007/s10589-007-9135-8>
- Gang J, Tu Y, Lev B, Xu J, Shen W, Yao L (2015) A multi-objective bi-level location planning problem for stone industrial parks. *Comput Oper Res* 56:8–21. <https://doi.org/10.1016/j.cor.2014.10.005>
- Gurobi Optimization I (2016) *Gurobi optimizer reference manual*. <http://www.gurobi.com>. Accessed 2018
- Hemmati M, Smith JC (2016) A mixed-integer bilevel programming approach for a competitive prioritized set covering problem. *Discrete Optim* 20:105–134
- Ivanov SV, Morozova MV (2016) Stochastic problem of competitive location of facilities with quantile criterion. *Autom Remote Control* 77:451–461
- Karakitsiou A (2015) *Modeling discrete competitive facility location*. Springer, Berlin

- Krarup J, Pruzan PM (1983) The simple plant location problem: survey and synthesis. *Eur J Oper Res* 12(1):36–81
- Kress D, Pesch E (2012) Sequential competitive location on networks. *Eur J Oper Res* 217:483–499
- Mel'nikov A (2014) Randomized local search for the discrete competitive facility location problem. *Autom Remote Control* 75:700–714
- Melnikov A, Beresnev V (2016) Upper bound for the competitive facility location problem with quantile criterion. In: Kochetov Y, Khachay M, Beresnev V, Nurminski E, Pardalos P (eds) *Discrete optimization and operations research*. Springer, Cham, pp 373–387
- Özpeynirci Ö (2017) On nadir points of multiobjective integer programming problems. *J Glob Optim* 69(3):699–712. <https://doi.org/10.1007/s10898-017-0534-9>
- Pelegrín B, Fernández P, García Pérez MD (2015) On tie breaking in competitive location under binary customer behavior. *Omega (UK)* 52:156–167. <https://doi.org/10.1016/j.omega.2014.10.010>
- Redondo JL, Fernández J, Domingo Álvarez Hervás J, Gila Arrondo A, Ortigosa PM (2015) Approximating the Pareto-front of a planar bi-objective competitive facility location and design problem. *Comput Oper Res* 62:337–349. <https://doi.org/10.1016/j.cor.2014.02.013>
- Rockafellar RT, Uryasev S (2002) Optimization of conditional value-at-risk. *J Risk* 26(7):1443–1471
- Saiz ME, Hendrix EMT, Fernandez J, Pelegrín B (2009) On a branch-and-bound approach for a huff-like stackelberg location problem. *OR Spectr* 31:679–705
- Santos-Peñate DR, Suarez-Vega R, Dorta-Gonzalez P (2007) The leader–follower location model. *Netw Spat Econ* 7:45–61
- Shapiro A (2008) Stochastic programming approach to optimization under uncertainty. *Math Program* 112(1):183–220. <https://doi.org/10.1007/s10107-006-0090-4>
- Spoerhase J (2010) Competitive and voting location. Ph.D. Thesis, Julius-Maximilians-Universität Würzburg
- Stackelberg H (1952) *The theory of the market economy*. Oxford University Press, Oxford
- Uno T, Katagiri H (2008) Single- and multi-objective defensive location problems on a network. *Eur J Oper Res* 188:76–84
- Vasilyev I, Klimentova X, Boccia M (2013) Polyhedral study of simple plant location problem with order. *Oper Res Lett* 41(2):153–158. <https://doi.org/10.1016/j.orl.2012.12.006>
- Yanıkoglu I, Kuhn D (2018) Decision rule bounds for two-stage stochastic bilevel programs. *SIAM J Optim* 28(1):198–222. <https://doi.org/10.1137/16M1098486>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.