ISSN 1990-4789, Journal of Applied and Industrial Mathematics, 2014, Vol. 8, No. 2, pp. 177–189. © Pleiades Publishing, Ltd., 2014.
Original Russian Text © V.L. Beresnev, A.A. Mel'nikov, 2014, published in Diskretnyi Analiz i Issledovanie Operatsii, 2014, Vol. 21, No. 2, pp. 3–23.

The Branch-and-Bound Algorithm for a Competitive Facility Location Problem with the Prescribed Choice of Suppliers

V. L. Beresnev[1,2*] and A. A. Mel'nikov[1,2]**

[1]*Sobolev Institute of Mathematics, pr. Akad. Koptyuga 4, Novosibirsk, 630090 Russia*

[2]*Novosibirsk State University, ul. Pirogova 2, Novosibirsk, 630090 Russia*

Received July 4, 2013; in final form, October 15, 2013

Abstract—In the mathematical model under study, the two competing sides consecutively place their facilities aiming to capture consumers and maximize profits. The model amounts to a bilevel integer programming problem. We take the optimal noncooperative solutions as optimal to this problem. To find approximate and optimal solutions, we propose a branch-and-bound algorithm. Simulations show that the algorithm can be applied to solve the individual problems of low and medium dimension.

DOI: 10.1134/S1990478914020045

Keywords: *bilevel programming, optimal noncooperative solution, pseudo-Boolean function, branch-and-bound method*

INTRODUCTION

We consider some consecutive competitive facility location problem generalizing the well-known facility location problem [3, 16]. In this model, two competing sides consecutively place their facilities aiming to capture consumers and maximize profits. We can regard decision-making by the competing sides as a Stackelberg game [18]. Following the terminology of this game, we call the sides the *leader* and the *follower*. The mathematical problems resulting from the formalization of this game amount to bilevel $(0, 1)$-programming problems [8–10, 13, 14]. They include an upper level problem (the leader's problem) and a lower level problem (the follower's problem). These are similar to the facility location problem with orders [6, 7, 12], but their form depends on the assumptions of the model. These assumptions are concerned first of all with the accepted rule for capturing consumers by one of the sides, as well as the rules used by the leader and follower when choosing an open facility to serve a certain consumer.

In competitive facility location problems we assume that each consumer has personal preferences that enable us to rank the open facilities. The side opening the most preferable facility captures the concrete consumer. The choice of a supplier to serve a consumer captured by one side is prescribed in the model under consideration: it is the most preferable facility opened by the chosen side. The competitive facility location problem is studied in [1, 11] in the case of free choice by the follower of a facility to serve the consumer. That model assumes that the follower makes a decision on the choice of a facility to serve each captured consumer.

In this article, for the model of consecutive competitive facility location under consideration, we propose a branch-and-bound algorithm [3, 17] to search for approximate and optimal solutions. An important element of the algorithm is the proposed method for calculating an upper bound on the optimal values of certain pseudo-Boolean functions. The idea of the method as applies to a particular case of the problem is presented in [2]. For the problem of searching for an optimal noncooperative solution, an algorithm is constructed in [5] to calculate the upper bound on the entire set of solutions on assuming

*E-mail: beresnev@math.nsc.ru
**E-mail: melnikov@math.nsc.ru

that the profits are monotone. Similar estimates are constructed in [1] for the general case of consecutive competitive facility location problem, in which the follower uses the rule of free choice of a facility to serve each captured consumer. For the last problem in the case of searching for optimal noncooperative solution, an upper bound is constructed in [11] for the values of the corresponding pseudo-Boolean functions on subsets specified by partial $(0, 1)$-vectors. In this article, for the problem of searching for optimal noncooperative solution we propose an algorithm to calculate an upper bound on the subset of solutions specified by a partial $(0, 1)$-vector on assuming that the profits are monotone. Furthermore, we find new properties of optimal solutions, which enables us to improve the resulting upper bounds substantially.

This article consists of five sections. In Section 2, we state the consecutive competitive facility location problem as a bilevel $(0, 1)$-programming problem. We introduce the concept of optimal noncooperative solutions and show that the problem of searching for a solution of this type can be reduced to maximizing certain pseudo-Boolean functions. In Section 3, we sketch a branch-and-bound algorithm for the problem of maximizing pseudo-Boolean functions. We consider a method for specifying subsets of $(0, 1)$-vectors with the use of partial $(0, 1)$-vectors. In Section 4, we propose a method for calculating an upper bound for the pseudo-Boolean functions under consideration. In Section 5, we present a branch-and-bound algorithm for searching for optimal and approximate (with an a priori estimate of accuracy) noncooperative solutions of the consecutive competitive facility location problem. We discuss the results of simulations on the test instances in the *Discrete location problems* library[1)].

1. THE CONSECUTIVE COMPETITIVE FACILITY LOCATION PROBLEM

We can regard the consecutive competitive facility location problem as the leader's problem in the Stackelberg game with the leader and the follower consecutively placing their facilities. The leader's problem consists in determining the set of facilities to be placed to maximize the profit under the condition that the follower will capture some consumers while also tending to maximize profit.

As in the classical facility location problem, to formalize the problem denote the set of facilities (possible facility locations) by $I = \{1, \dots, m\}$, and the set of consumers, by $J = \{1, \dots, n\}$. Assume that both the leader and the follower can open facility $i \in I$. Therefore, for each $i \in I$, we know the quantities f_i and g_i equal to the fixed costs of the leader and the follower to open facility i. If, for some reason, the leader or the follower cannot open this facility then we put $f_i = +\infty$ or $g_i = +\infty$.

Assume that an opened facility to serve consumer $j \in J$ is chosen by the preferences of consumer j as indicated by a linear order $\succ_j$ on the set I. For $i, k \in I$, the relation $i \succ_j k$ means that of two opened facilities i and k consumer $j \in J$ prefers facility i. The relation $i \succeq_j k$ means that either $i \succ_j k$ or $i = k$. For all $i \in I$ and $j \in J$, denote by p_{ij} the profit received by serving consumer j at facility i opened by the leader or the follower. Assume that, for each $j \in J$, the profits p_{ij} for $i \in I$ are monotone in $\succ_j$; that is, $p_{ij} \geq p_{kj}$ for all $i, k \in I$ with $i \succ_j k$.

Take $I_0 \subset I$. Given $j \in J$, let $i_j(I_0)$ denote the element $i_0 \in I_0$ with $i_0 \succeq_j i$ for all $i \in I_0$. If $I_0 = \{i \in I \mid w_i = 1\}$, where $w = (w_i)$ for $i \in I$ is a $(0, 1)$-vector, then we also write $i_j(w)$ instead of $i_j(I_0)$. Given $i \in I$, denote $J_i(w) = \{j \in J \mid i \succeq_j i_j(w)\}$. Given two $(0, 1)$-vectors $x = (x_i)$ and $z = (z_i)$, let $x \vee z$ denote the $(0, 1)$-vector $w = (w_i)$ with $w_i = \max(x_i, z_i)$ for $i \in I$.

To determine the side capturing the consumer $j \in J$, we accept the rule: Suppose that the 1s of the $(0, 1)$-vector $x = (x_i)$ for $i \in I$ mean the facilities opened by the leader and the 1s of the $(0, 1)$-vector $z = (z_i)$ for $i \in I$ mean the facilities opened by the follower. Then consumer $j \in J$ will be captured by the leader whenever $i_j(x) \succeq_j i_j(z)$, and by the follower, whenever $i_j(z) \succ_j i_j(x)$. To choose the facility to serve the captured consumer $j \in J$, assume that the leader and the follower use the *strict choice* rule: this will be facility $i_j(x)$ when the leader captures the consumer and $i_j(z)$ when the follower does. This rule models the situation in which each consumer independently chooses a facility.

Introduce the following variables similar to those of the classical facility location problem:

- x_i equals 1 if the leader opens facility $i \in I$, and zero otherwise;
- x_{ij} equals 1 if facility $i \in I$ opened by the leader turns out the most preferable for consumer $j \in J$ among all facilities opened by the leader, and zero otherwise;

---

1) http://www.math.nsc.ru/AP/benchmarks/

- z_i equals 1 if the follower opens facility $i \in I$, and zero otherwise;
- z_{ij} equals 1 if facility $i \in I$ opened by the follower turns out the most preferable for consumer $j \in J$ among all facilities opened by both the leader and the follower, and zero otherwise.

Using these variables, we formulate the consecutive competitive facility location problem as the bilevel integer programming problem:

$$\max_{(x_i),(x_{ij})} \Big(-\sum_{i\in I} f_i x_i + \sum_{j\in J} \Big(\sum_{i\in I} p_{ij} x_{ij} \Big) \Big(1 - \sum_{i\in I} \widetilde{z}_{ij} \Big) \Big), \tag{1}$$

$$x_i + \sum_{k\in I | i \succ_j k} x_{kj} \leqslant 1, \qquad i \in I, \quad j \in J, \tag{2}$$

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

$$x_i, x_{ij} \in \{0,1\}, \qquad i \in I, \quad j \in J, \tag{4}$$

$\big((\widetilde{z}_i),(\widetilde{z}_{ij})\big)$ is an optimal solution to (5)–(8):

$$\max_{(z_i),(z_{ij})} \Big(-\sum_{i\in I} g_i z_i + \sum_{j\in J}\sum_{i\in I} p_{ij} z_{ij} \Big), \tag{5}$$

$$x_i + z_i + \sum_{k\in I | i \succ_j k} z_{kj} \leqslant 1, \qquad i \in I, \quad j \in J, \tag{6}$$

$$z_i \geqslant z_{ij}, \qquad i \in I, \quad j \in J, \tag{7}$$

$$z_i, z_{ij} \in \{0,1\}, \qquad i \in I, \quad j \in J. \tag{8}$$

As a bilevel mathematical programming problem, (1)–(8) includes the upper level problem (1)–(4) and the lower level problem (5)–(8). Denote the upper level problem by $\mathfrak{L}$ and the lower level problem by $\mathfrak{F}$. The objective function (1) of $\mathfrak{L}$ expresses the profit made by the leader taking into account the cost of facilities opening and losing part of consumers which are captured by the follower. Conditions (2)–(4) are the constraints of location problem with orders. The inequalities (2) realize the rule of strict choice of a facility opened by the leader to serve each consumer. The same inequalities guarantee that only one facility opened by the leader can be chosen to serve a certain consumer. The objective function (5) of problem $\mathfrak{F}$ expresses the profit made by the follower. The inequalities (6) realize the conditions of the follower capturing consumers for the specified facilities opened by the leader. In particular, these conditions show that if a facility is opened by the leader then it cannot be opened by the follower. The remaining conditions of problem $\mathfrak{F}$ are the constraints of the classical facility location problem. Denote problem (1)–(8) as a whole by $(\mathfrak{L},\mathfrak{F})$, and regard the objective function (1) of problem $\mathfrak{L}$ as the objective function of problem $(\mathfrak{L},\mathfrak{F})$.

Let X denote an admissible solution $((x_i),(x_{ij}))$ to problem $\mathfrak{L}$, and let Z be an admissible solution $((z_i),(z_{ij}))$ to $\mathfrak{F}$. Refer to a pair $(X,\widetilde{Z})$, where X is an admissible solution to $\mathfrak{L}$ and $\widetilde{Z}$ is an optimal solution to $\mathfrak{F}$, as an *admissible solution* to problem $(\mathfrak{L},\mathfrak{F})$.

Take an admissible solution $(X,\widetilde{Z})$ to problem $(\mathfrak{L},\mathfrak{F})$ with $X = ((x_i),(x_{ij}))$ and $\widetilde{Z} = ((\widetilde{z}_i),(\widetilde{z}_{ij}))$. Assume that if X is the zero solution then the optimal solution $\widetilde{Z}$ to problem $\mathfrak{F}$ is nonzero. Thus, we assume henceforth that the admissible solutions $(X,\widetilde{Z})$ to problem $(\mathfrak{L},\mathfrak{F})$ are nonzero. Denote by $L(X,\widetilde{Z})$ the value of the objective function of problem $(\mathfrak{L},\mathfrak{F})$ on an admissible solution $(X,\widetilde{Z})$ and by $F(Z)$ the value of the objective function of problem $\mathfrak{F}$ on an admissible solution Z.

Since, for some admissible solution X to problem $\mathfrak{L}$, an optimal solution $\widetilde{Z}$ to $\mathfrak{F}$ need not be unique, let us state the rule the follower uses to make a decision: Assume that, among the optimal solutions, the follower chooses the least profitable for the leader. Refer to an admissible solution $(X,\overline{Z})$ to problem

$(\mathfrak{L}, \mathfrak{F})$ as an *admissible noncooperative solution* to $(\mathfrak{L}, \mathfrak{F})$ whenever $L(X, \overline{Z}) \leq L(X, \widetilde{Z})$ for every admissible solution $(X, \widetilde{Z})$ to $(\mathfrak{L}, \mathfrak{F})$, and refer to an admissible noncooperative solution $(X^*, \overline{Z}^*)$ as an *optimal noncooperative solution* if $L(X^*, \overline{Z}^*) \geq L(X, \overline{Z})$ for every admissible noncooperative solution $(X, \overline{Z})$.

Observe that, given an admissible solution X to problem $\mathfrak{L}$, we can construct the corresponding admissible noncooperative solution $(X, \overline{Z})$ by a two-stage algorithm. At Stage 1, for a fixed solution X, we solve problem $\mathfrak{F}$ and calculate the optimal value F^* of its objective function. At Stage 2, for the fixed solution X, we solve the auxiliary problem

$$\max_{(z_i),(z_{ij})} \sum_{j \in J} \sum_{i \in I} p_{i_j(x)j} z_{ij}, \tag{9}$$

$$x_i + z_i + \sum_{k \in I | i \succ_j k} z_{kj} \leqslant 1, \qquad i \in I, \quad j \in J, \tag{10}$$

$$z_i \geqslant z_{ij}, \qquad i \in I, \quad j \in J, \tag{11}$$

$$z_i, z_{ij} \in \{0, 1\}, \qquad i \in I, \quad j \in J, \tag{12}$$

$$-\sum_{i \in I} g_i z_i + \sum_{j \in J} \sum_{i \in I} p_{ij} z_{ij} \geqslant F^*. \tag{13}$$

An optimal solution $\overline{Z} = ((\overline{z}_i, \overline{z}_{ij}))$ to this problem yields a required admissible noncooperative solution $(X, \overline{Z})$ to problem $(\mathfrak{L}, \mathfrak{F})$. Furthermore, the quantity $L(X, \overline{Z})$ will be the same for every optimal solution $\overline{Z}$ to the auxiliary problem (9)–(13). Observe also that the admissible solution $X = ((x_i), (x_{ij}))$ to problem $\mathfrak{L}$ is itself uniquely determined by the $(0,1)$-vector $x = (x_i)$. Thus, every $(0,1)$-vector x uniquely determines some value $L(X, \overline{Z})$ of the objective function at the corresponding admissible noncooperative solution $(X, \overline{Z})$. This implies that we can express the problem of searching for an optimal noncooperative solution to the consecutive competitive facility location problem $(\mathfrak{L}, \mathfrak{F})$ as the problem of maximizing some pseudo-Boolean function $f(x)$ of $x \in B^m$. This function is defined implicitly; to calculate its values, we have to find F^*, which is the optimal value of the objective function of the lower level problem $\mathfrak{F}$, and then find an optimal solution to the auxiliary problem (9)–(13).

2. THE BRANCH-AND-BOUND METHOD

Let us consider the general scheme of the branch-and-bound method with depth-first search for the problem of maximizing a pseudo-Boolean function $f(x)$ of $x \in B^m$. Denote an optimal solution to this problem by x^*. Owing to [3], assume that the following functions are specified on the subsets $s \subseteq B^m$. Refer to a function $b(s)$ defining a proper subset of s as the *branching function*. Refer to a function $H(s)$ with $H(s) \geq f(x)$ for all $x \in s$ as an *upper bound*. Assume that $H(s) = f(x)$ whenever $s = \{x\}$. Consider also a function $x(s)$ determining a solution in s.

The algorithm realizing the branch-and-bound method with depth-first search consists of finitely many similar steps. On each step we consider: the set $D \subset B^m$ called the *set of candidate solutions*; a subset $d \subset D$ called the *set of testing solutions*; and the best solution $x^0 \in B^m$ known on this step, called the *record solution*. At the first step, $D = B^m$ and $d = B^m$, while x^0 is an arbitrary element of B^m. Suppose that before the next step we have the set D of candidate solutions and the set $d \subset D$ of testing solutions, as well as a record solution x^0. The step starts with the evaluation of $H(d)$ and $x(d)$. If $f(x(d)) > f(x^0)$ then we put $x^0 = x(d)$. The step consists in checking whether d contains a solution better than the current record. To this end, we check the inequality $(1 - \epsilon)H(d) \leq f(x^0)$, where $\epsilon \in [0, 1)$ is a parameter determining the accuracy of the solution x^0 found by the algorithm. If the inequality holds then we discard d by putting $D = D \backslash d$. If $D = \varnothing$ then the algorithm stops; otherwise, we put $d = b(D)$ and start the next step. However, if the inequality is violated then we put $d = b(d)$ and start the next step. The algorithm stops after finitely many steps, and the resulting solution x^0 satisfies $(1-\epsilon)f(x^*) \le f(x^0)$.

It is convenient to define subsets of B^m using the so-called partial solutions. Put $I = \{1, \dots, m\}$. Refer to a vector $y \in \{0, 1, *\}^m$ as a *partial* $(0,1)$-*vector* or a *partial solution*. A partial solution splits the variables of a pseudo-Boolean function into the variables with a specified value 0 or 1 and free variables. Given a partial $(0,1)$-vector $y = (y_i)$ for $i \in I$, define the sets

$$I^0(y) = \{i \in I \mid y_i = 0\}, \qquad I^1(y) = \{i \in I \mid y_i = 1\}, \qquad I^*(y) = \{i \in I \mid y_i = *\}.$$

Refer to $x \in B^m$ as an *extension* of a partial solution y whenever

$$I^0(y) \subseteq I^0(x), \qquad I^1(y) \subseteq I^1(x).$$

Denote the set of all extensions of a partial solution y by $P(y)$. Call a partial solution y *ordered* whenever a vector of order $\{i_1, \dots, i_q\} = I^0(y) \cup I^1(y)$ is specified for y, indicating the order in which the components of y were assigned the values of 0 or 1. To an ordered partial solution $y = (y_i)$ and a vector of order $(i_1, \dots, i_q)$, we can associate, apart from the set $P(y)$, another set $Q(y)$ with $P(y) \subset Q(y)$. In order to define $Q(y)$ for every k with $1 \le k \le q$ and $y_{i_k} = 1$, construct a partial solution $y(k) = (y_i(k))$ for $i \in I$ such that

$$I^0(y(k)) = (I^0(y) \cap \{i_1, \dots, i_{k-1}\}) \cup \{i_k\},$$

$$I^1(y(k)) = I^1(y) \cap \{i_1, \dots, i_{k-1}\}.$$

The union of the sets $P(y(k))$ for the constructed partial solutions $y(k)$ and the set $P(y)$ constitutes $Q(y)$. Ordered partial solutions are useful since the branching function can be defined so that, on each step of the branch-and-bound algorithm, the sets D and d are determined by an ordered partial solution y and coincide respectively with $Q(y)$ and $P(y)$ [3].

3. THE BOUNDING AND BRANCHING FUNCTIONS FOR THE CONSECUTIVE COMPETITIVE FACILITY LOCATION PROBLEM

Consider a method for calculating the upper bound $H(y)$ for the pseudo-Boolean function $f(x)$ on the subset $P(y)$ of extensions of a partial solution y. To this end, given a fixed partial solution $y = (y_i)$, consider problem (1)–(4) with the additional restriction

$$x_i = y_i, \qquad i \in I^0(y) \cup I^1(y). \tag{14}$$

Denote problem (1)–(4), (14) by $\mathfrak{L}(y)$ and problem (1)–(4), (14), (5)–(8) by $(\mathfrak{L}(y), \mathfrak{F})$. Problem $(\mathfrak{L}(y), \mathfrak{F})$ corresponds to the problem of maximizing the pseudo-Boolean function $f(x)$ on $P(y)$, while the upper bound on the value of the objective function of $(\mathfrak{L}(y), \mathfrak{F})$ on an optimal noncooperative solution is the value $H(y)$ of the required upper bound on $P(y)$. Recall that, for every $j \in J$, we assume that the profit p_{ij} for $i \in I$ is monotone with respect to the order $\succ_j$; that is, $p_{ij} \ge p_{kj}$ for every $i, k \in I$ with $i \succ_j k$.

Let us point out some properties of feasible and optimal solutions to problem $(\mathfrak{L}(y), \mathfrak{F})$. To simplify exposition, consider the set $I \cup \{0\}$ and assume that $p_{0j} = 0$ for each $j \in J$ and $i \succ_j 0$ for $i \in I$. Take a partial solution $y = (y_i)$. Given $j \in J$, put $i_j(y) = i_j(I^1(y))$ whenever $I^1(y) \ne \varnothing$, and $i_j(y) = 0$ otherwise. Given some $i \in I$, denote the set $\{j \in J \mid i \succ i_j(y)\}$ by $J_i(y)$. Consider the sets

$$R(y) = \Big\{ i \in I^0(y) \cup I^*(y) \mid \sum_{j \in J_i(y)} p_{ij} < g_i \Big\},$$

$$S(y) = \Big\{ i \in I^*(y) \mid \sum_{j \in J_i(y)} (p_{ij} - p_{i_j(y)j}) \le f_i \text{ and for each } j \in J_i(y) \text{ if } i \succeq_j k \succ_j i_j(y) \text{ then } k \in R(y) \Big\}.$$

The following elucidates the meaning of $R(y)$ and $S(y)$:

Lemma 1. *Given a partial solution y, every feasible solution* $(X, \widetilde{Z})$ *to problem* $(\mathfrak{L}(y), \mathfrak{F})$ *with* $X = ((x_i), (x_{ij})$ *and* $\widetilde{Z} = ((\widetilde{z}_i), (\widetilde{z}_{ij}))$ *satisfies* $R(y) \cap I^1(\widetilde{z}) = \varnothing$ *with* $\widetilde{z} = (\widetilde{z}_i)$. *There exists an optimal noncooperative solution* $(X, \overline{Z})$ *with* $X = ((x_i), (x_{ij}))$ *and* $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$ *to problem* $(\mathfrak{L}(y), \mathfrak{F})$ *such that*

$$S(y) \cap I^1(x) = \varnothing, \qquad x = (x_i).$$

Proof. To verify the first relation, suppose that $\widetilde{z}_i = 1$ for some $i \in R(y)$. Since $J_i(x \vee \widetilde{z}) \subset J_i(y)$, we have

$$\sum_{j \in J_i(x \vee \widetilde{z})} p_{ij} \le \sum_{j \in J_i(y)} p_{ij} < g_i,$$

which contradicts the fact that $\widetilde{Z}$ is an optimal solution to problem $\mathfrak{F}$.

To verify the second claim, assume that, in the optimal noncooperative solution $(X, \widetilde{Z})$ under consideration, $x_{i_0} = 1$ for some $i_0 \in S(y)$. Take the solution $X' = ((x'_i), (x'_{ij}))$ to problem $\mathfrak{L}(y)$ in which the vector $x' = (x'_i)$ differs from $x = (x_i)$ only in $x'_{i_0} = 0$. Denote problem $\mathfrak{F}$ for fixed solutions X and X' by $\mathfrak{F}(X)$ and $\mathfrak{F}(X')$ respectively. To verify that the sets of optimal solutions to $\mathfrak{F}(X)$ and $\mathfrak{F}(X')$ coincide, rearrange the constrains (6) of problem $\mathfrak{F}(X)$ as

$$z_i + \sum_{k \in I | i \succ_j k} z_{kj} \leqslant 1, \qquad i \in I, \quad j \in J, \tag{15}$$

$$\sum_{k | i_j(x) \succeq_j k} z_{kj} = 0, \qquad j \in J. \tag{16}$$

By above, every optimal solution $\widetilde{Z} = ((\widetilde{z}_i), (\widetilde{z}_{ij}))$ to problem $\mathfrak{F}(X)$ satisfies $\widetilde{z}_i = 0$ for $i \in R(y)$; therefore, we can also add the equalities

$$z_i = 0, \qquad i \in R(y)$$

to its constrains. Problem $\mathfrak{F}(X')$ differs from $\mathfrak{F}(X)$ only in constrains (16), which we replace by

$$\sum_{k | i_j(x') \succeq k} z_{kj} = 0, \qquad j \in J. \tag{17}$$

Verify that the sets of optimal solutions to problems $\mathfrak{F}(X)$ and $\mathfrak{F}(X')$ coincide. Indeed, if Z is a feasible solution to problem $\mathfrak{F}(X)$ then (17) holds since $i_j(x) \succeq_j i_j(x')$ for every $j \in J$. Conversely, take a feasible solution Z to problem $\mathfrak{F}(X')$, and suppose that $i_j(x) \succ_j i_j(x')$ for some $j \in J$. Since

$$i_0 = i_j(x) \succ_j i_j(x') \succeq_j i_j(y), \qquad i_0 \in S(y);$$

therefore, $k \in R(y)$ for every k with $i_0 \succ_j k \succ_j i_j(x')$. Hence, $z_k = 0$ and (16) hold. Thus, the sets of optimal solutions to problems $\mathfrak{F}(X)$ and $\mathfrak{F}(X')$ coincide; moreover, if $\widetilde{Z}$ is an optimal solution then $(X, \widetilde{Z})$ and $(X', \widetilde{Z})$ is a feasible solution to problem $(\mathfrak{L}(y), \mathfrak{F})$. By the monotonicity of profit, these solutions satisfy

$$L(X, \widetilde{Z}) - L(X', \widetilde{Z}) = -f_{i_0} + \sum_{j \in J_{i_0}(x)} (p_{i_0 j} - p_{i_j(x') j}) \le -f_{i_0} + \sum_{j \in J_{i_0}(y)} (p_{i_0 j} - p_{i_j(y) j}) \le 0.$$

Consequently, given a feasible noncooperative solution $(X, \overline{Z})$, there is a feasible noncooperative solution $(X', \overline{Z})$ with

$$S(y) \cap I^1(x') = \varnothing, \qquad L(X, \overline{Z}) \le L(X', \overline{Z}).$$

This completes the proof of Lemma 1. □

Henceforth, as we consider optimal noncooperative solutions, we assume that they enjoy the properties established in Lemma 1. The method we propose for calculating $H(y)$ rests on a construction of a system of subsets $\{I_j(y)\}$ with $I_j(y) \subset I$ for $j \in J$. Using it, we manage to state sufficient conditions for the capture of consumers by the follower.

Take a partial solution $y = (y_i)$ and fix $j_0 \in J$. Let us state the conditions enabling us for every $i_0 \in I$ to determine whether $i_0 \in I_{j_0}(y)$ or $i_0 \notin I_{j_0}(y)$. If $y_{i_0} = 0$ then $i_0 \notin I_{j_0}(y)$. Assume that $y_{i_0} \neq 0$. Consider the set $N(i_0) = \{i \in I \mid i \succ_{j_0} i_0\}$. If $N(i_0) = \varnothing$ then $i_0 \in I_{j_0}(y)$. Assume that $N(i_0) \neq \varnothing$. If $N(i_0) \cap I^1(y) \neq \varnothing$ then $i_0 \notin I_{j_0}(y)$. Assume that $N(i_0) \neq \varnothing$ and $N(i_0) \cap I^1(y) = \varnothing$. Take the partial solution $y' = (y'_i)$ with $y'_i = y_i$ for $i \neq i_0$ and $y'_{i_0} = 1$. Consider the sets $R(y')$, $S(y')$, and

$$J(y, i_0) = \left\{ j \in J \mid \text{ if } i \succ i_j(y') \text{ then } i \in N(i_0) \cup R(y') \right\}.$$

Observe that $J(i_0) \neq \varnothing$ since $j_0 \in J(i_0)$. Given $k \in N(i_0)$, consider the set

$$J(y, k, i_0) = \left\{ j \in J(y, i_0) \mid \text{ if } \ i \succ_j k \text{ then } i \in N(i_0) \cup I^0(y') \cup S(y') \right\}.$$

Assume that $i_0 \in I_{j_0}(y)$ if, for each $k \in N(i_0)$, we have

$$g_k > \sum_{j \in J(y,k,i_0)} p_{kj},$$

and $i_0 \notin I_j(y)$ if there is $k \in N(i_0)$ for which the inequality fails. Refer to this system of subsets $\{I_j(y)\}$ for $j \in J$ as the system of subsets *defined by strict inequalities*. The following lemma elucidates the meaning of $I_{j_0}(y)$ for $j_0 \in J$ and establishes that if the leader plans to profit by serving consumer j_0 at facility $i_0 \notin I_{j_0}(y)$ then consumer j_0 will be captured by the follower.

Lemma 2. *Given a partial solution y, take a system of subsets $\{I_j(y)\}$ for $j \in J$ defined by strict inequalities and a feasible noncooperative solution $(X, \overline{Z})$ to problem $(\mathfrak{L}(y), \mathfrak{F})$ with $X = ((x_i), (x_{ij}))$ and $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$. Then every $j_0 \in J$ with $p_{i_0 j_0} x_{i_0 j_0} > 0$ satisfies the equality $\sum_{i \in I} \overline{z}_{ij_0} = 1$ for some $i_0 \notin I_{j_0}(y)$.*

Proof. Consider the $(0,1)$-vectors $x = (x_i)$ and $\overline{z} = (\overline{z}_i)$. Suppose that $p_{i_0 j_0} x_{i_0 j_0} > 0$ for some $j_0 \in J$ and $i_0 \notin I_{j_0}(y)$, but the required equality fails. Consider the partial solution $y' = (y'_i)$ with $y'_i = y_i$ for $i \neq i_0$ and $y'_{i_0} = 1$, as well as the set $N(i_0) = \{i \in I \mid i \succ_{j_0} i_0\}$. Since $i_0 \notin I_{j_0}(y)$, it follows that $N(i_0) \neq \varnothing$. Note that $x_i = 0$ and $\overline{z}_i = 0$ for $i \in N(i_0)$. Consider also the sets $R(y')$ and $S(y')$. Lemma 1 implies that $\overline{z}_i = 0$ for $i \in R(y')$, and we may assume that $x_i = 0$ for $i \in S(y')$. Put

$$J(y, i_0) = \{ j \in J \mid \ \text{ if } \ i \succ_j i_j(y') \ \text{ then } \ i \in N(i_0) \cup R(y') \}$$

and note that $i_j(x) \succ_j i_j(\overline{z})$ for $j \in J(y, i_0)$. Since $i_0 \notin I_{j_0}(y)$, there is $k \in N(i_0)$ for which the set

$$J(y, k, i_0) = \{ j \in J(i_0) \mid \ \text{ if } \ i \succ_j k \ \text{ then } \ i \in N(i_0) \cup I^0(y') \cup S(y') \}$$

satisfies

$$g_k \leqslant \sum_{j \in J(y,k,i_0)} p_{kj}.$$

Observe that $k \succ_j i_j(x) \succ_j i_j(\overline{z})$ for $j \in J(y, k, i_0)$. Consider the sets

$$J_L(k) = \{ j \in J \mid k \succ_j i_j(x) \succ_j i_j(\overline{z}) \}, \qquad J_F(k) = \{ j \in J \mid k \succ_j i_j(\overline{z}) \succ_j i_j(x) \},$$

and the solution $Z = ((z_i), (z_{ij}))$ to problem $\mathfrak{F}$ differing from $\overline{Z}$ in $z_k = 1$, $z_{kj} = 1$ for $j \in J_L(k)$ and $z_{kj} = 1$, $z_{i_j(\overline{z})j} = 0$ for $j \in J_F(k)$. By the monotonicity of profit, $J(y, k, i_0) \subset J_L(k)$ entails

$$F(Z) - F(\overline{Z}) = -g_k + \sum_{j \in J_L(k)} p_{kj} + \sum_{j \in J_F(k)} (p_{kj} - p_{i_j(\overline{z})j}) \geq -g_k + \sum_{j \in J(y,k,i_0)} p_{kj} \geq 0.$$

Consequently, Z is an optimal solution to problem $\mathfrak{F}$, while (X, Z) is a feasible solution to $(\mathfrak{L}(y), \mathfrak{F})$. The feasible solutions (X, Z) and $(X, \overline{Z})$ satisfy

$$L(X, \overline{Z}) - L(X, Z) \geq \sum_{j \in J(y,k,i_0)} p_{i_j(x)j} \geq p_{i_0 j_0} > 0.$$

This contradicts the fact that $(X, \overline{Z})$ is a feasible noncooperative solution.

The proof of Lemma 2 is complete. □

Lemma 3. *Given a partial solution y, take a system of subsets $\{I_j(y)\}$ for $j \in J$ defined by strict inequalities and a feasible noncooperative solution $(X, \overline{Z})$ to problem $(\mathfrak{L}(y), \mathfrak{F})$ with $X = ((x_i), (x_{ij}))$ and $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$. Then every $j \in J$ satisfies*

$$\Big(\sum_{i \in I} p_{ij} x_{ij}\Big)\Big(1 - \sum_{i \in I} \overline{z}_{ij}\Big) = \Big(\sum_{i \in I_j(y)} p_{ij} x_{ij}\Big)\Big(1 - \sum_{i \in I} \overline{z}_{ij}\Big).$$

Proof. If $p_{ij} x_{ij} = 0$ for every $i \in I$ then the equality holds. Suppose that $p_{i_0 j} x_{i_0 j} > 0$ for some $i_0 \in I$. If $i_0 \in I_j(y)$ then the equality holds. If $i_0 \notin I_j(y)$ then the equality holds because

$$\sum_{i \in I} \overline{z}_{ij} = 1$$

by Lemma 1. The proof of Lemma 3 is over. □

Given a fixed partial solution $y = (y_i)$, consider the following problem which we call the *estimating* problem:

$$\max_{(x_i),\,(x_{ij})} \Big(- \sum_{i \in I} f_i x_i + \sum_{j \in J} \sum_{i \in I_j(y)} p_{ij} x_{ij} \Big),$$

$$x_i + \sum_{k \in I | i \succ_j k} x_{kj} \leqslant 1, \qquad i \in I, \quad j \in J,$$

$$x_i \geqslant x_{ij}, \qquad i \in I, \quad j \in J,$$

$$x_i = y_i, \qquad i \in I^0(y) \cup I^1(y),$$

$$x_i, x_{ij} \in \{0, 1\}, \qquad i \in I, \quad j \in J.$$

Let $\mathfrak{B}(y)$ denote the estimating problem with a system of subsets $I_j(y)$ for $j \in J$ defined by strict inequalities; and $B(X)$, the value of its objective function on a solution $X = ((x_i), (x_{ij}))$.

Theorem. *Given a partial solution y, take a system of subsets $\{I_j(y)\}$ for $j \in J$ defined by strict inequalities and an optimal noncooperative solution $(X, \overline{Z})$ to problem $(\mathfrak{L}(y), \mathfrak{F})$ with $X = ((x_i), (x_{ij}))$ and $\overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$. Denote by X^0 an optimal solution to problem $\mathfrak{B}(y)$. Then*

$$L(X, \overline{Z}) \leq B(X^0).$$

Proof. By Lemma 3, the optimal values of the objective function of problem $(\mathfrak{L}(y), \mathfrak{F})$ satisfy

$$L(X, \overline{Z}) = - \sum_{i \in I} f_i x_i + \sum_{j \in J} \Big(\sum_{i \in I} p_{ij} x_{ij}\Big)\Big(1 - \sum_{i \in I} \overline{z}_{ij}\Big) \leq - \sum_{i \in I} f_i x_i + \sum_{j \in J} \sum_{i \in I_j(y)} p_{ij} x_{ij}.$$

Since X is a feasible solution to problem $\mathfrak{B}(y)$, we infer

$$L(X, \overline{Z}) \leq B(X) \leq B(X^0).$$

The proof of Theorem 1 is complete. □

Take an optimal solution X^0 to problem $\mathfrak{B}(y)$ and the corresponding feasible noncooperative solution $(X^0, \overline{Z})$ to $(\mathfrak{L}(y), \mathfrak{F})$. As a corollary to Theorem 1, we can state a sufficient condition for $B(X^0)$ to be a sharp bound and for $(X^0, \overline{Z})$ to be an optimal noncooperative solution to $(\mathfrak{L}(y), \mathfrak{F})$:

Corollary. *A feasible noncooperative solution $(X^0, \overline{Z})$ with*

$$X^0 = \big((x_i^0), (x_{ij}^0)\big), \qquad \overline{Z} = ((\overline{z}_i), (\overline{z}_{ij}))$$

is optimal for problem $(\mathfrak{L}(y), \mathfrak{F})$ whenever, for all $j \in J$,

$$\Big(\sum_{i \in I_j(y)} p_{ij} x_{ij}^0\Big) \sum_{i \in I} \overline{z}_{ij} = 0.$$

This implies in particular that if $\overline{Z}$ is the zero solution then $(X^0, \overline{Z})$ is an optimal noncooperative solution to problem $(\mathfrak{L}(y), \mathfrak{F})$. Using the test for the optimality of $(X, \overline{Z})$, we can indicate a rule for choosing $i(y) \in I^*(y)$ which increases the set of 1s of the partial solution y in the best possible way. Put

$$\overline{z} = (\overline{z}_i), \qquad J_0 = \Big\{ j \in J \;\Big|\; \Big(\sum_{i \in I_j(y)} p_{ij} x_{ij}^0\Big) \sum_{i \in I} \overline{z}_{ij} \neq 0 \Big\} \neq \varnothing.$$

For $i \in I^*(y)$, put $J_0(i) = \{j \in J_0 \cap J_i(\overline{z}) \mid i \in I_j(y)\}$ and denote by $i(y)$ the element $i \in I^*(y)$ with the greatest value of $\sum_{j \in J_0(i)} p_{ij}$. Define the function calculating the upper bound $H(y)$ as follows: If $y \in B^m$ then consider the feasible noncooperative solution $(X, \overline{Z})$ to problem $(\mathfrak{L}, \mathfrak{F})$ corresponding to the $(0,1)$-vector y. Put

$$H(y) = L(X, \overline{Z}), \qquad x(y) = y.$$

If $y \notin B^m$ then take the $(0,1)$-vector $x^0 = \big(x_i^0\big)$ from an optimal solution X^0 to the estimating problem $\mathfrak{B}(y)$ and put $H(y) = B(X^0)$ and $x(y) = x^0$.

Let us use the functions $i(y)$ and $x(y)$ defined in this section, as well as $H(y)$, to construct an algorithm for searching for a $(1-\varepsilon)$-approximate solution to the consecutive competitive location problem.

4. THE BRANCH-AND-BOUND ALGORITHM. RESULTS OF SIMULATIONS

We showed above how to express $(\mathfrak{L}, \mathfrak{F})$ as a problem of maximizing a certain pseudo-Boolean function. The branch-and-bound algorithm for the problem of maximizing the pseudo-Boolean functions $f(x)$ under consideration includes an initial step and a finitely many main steps.

Initial Step. We have an ordered partial solution $y{=}(y_i)$ with $y_i = *$ for $1 \le i \le m$ and $q{=}0$. Calculate the upper bound $H(y)$ and the solution $x(y)$. Then apply to the vector $x(y)$ the local search procedure; it yields a locally optimal solution, which we take as the initial record solution x^0. If

$$(1-\epsilon)H(y) \le f(x^0)$$

then STOP, otherwise GOTO *move down*.

Main Step. We have an ordered partial solution $y = (y_i)$ and an order vector $(i_1, \dots, i_q)$ with $1 \le q \le m$. If $q = m$ then calculate $f(x(y))$. If $f(x(y)) > f(x^0)$ then $x^0 := x(y)$ and GOTO *move up*. If $q < m$ then calculate $H(y)$ and $x(y)$. If $(1-\epsilon)H(y) \le f(x^0)$ then GOTO *move up*, otherwise GOTO *move down*.

Move up. Put $r := 0$ and calculate the greatest index r with $1 \le r \le q$ and $y_{i_r} = 1$. If $r = 0$ then STOP, otherwise $y_{i_k} := *$ for $r < k \le q$, $y_{i_r} := 0$, $q := r$, and GOTO *Main Step*.

Move down. Construct

$$I^1(x(y)) \backslash I^1(y) = \{i_{q+1}, \dots, i_{q+r}\}, \qquad r \ge 0.$$

If $r = 0$ then calculate the index $i(y)$ and put $i_{q+1} = i(y)$, $y_{i_{q+1}} = 1$, and $q := q + 1$; otherwise, put $y_{i_{q+k}} := 1$ for $1 \le k \le r$ and $q := q + r$. Construct the set

$$S(y) = \{i_{q+1}, \dots, i_{q+r}\}, \qquad r \ge 0.$$

If $r > 0$ then $y_{i_{q+k}} := 0$ for $1 \le k \le r$ and $q := q + r$. GOTO *the Main Step*.

We studied the properties of the method by the randomly generated test instances of the competitive facility location problem on a tree network taken from the class of examples $TreeNE$ of the *Discrete location problems* library of test instances. For these examples, the sets I and J coincide with the set V of vertices of a random tree. The iterative procedure constructing the tree starts with the trivial tree with a single vertex. At every iteration, a new dangling vertex is added to the tree already constructed and joined to a random vertex of the tree with either a long edge with probability 0.1 or a short edge with probability 0.9. The lengths of short edges are random numbers in the interval $[1, 15]$, while those of long edges lie in the interval $[100, 150]$. The fixed costs f_i and g_i of opening a facility at $i \in V$ for the leader and the follower are randomly chosen integer numbers in the intervals $[30, 39]$ and $[20, 29]$ respectively. The consumers' preferences are determined by the distance in the tree. For each consumer $j \in V$, we choose randomly in the integer interval $[5, 9]$ the value b_j of budget. Put the profit p_{ij} by serving consumer j at facility i equal to b_j in the case that the distance from consumer j to facility i is at most 100.

For the values of m equal to 20, 30, 40, 50, and 60, we generated 20 tuples of input data of competitive location problem. A parallel (multi-thread) implementation of the algorithm is written in the C# programming language. We ran tests on a computer controlled by the operating system Windows Server 2008 R2 with two six-core processors Intel Xeon X5675 at 3.07 GHz and 96 Gb of RAM. To solve integer programming problems, we used the class library Microsoft Solver Foundation 3.1 based on the Gurobi 4.5 solver. As the initial approximation we chose a solution obtained using the algorithm of [4]. The running time of the branch-and-bound algorithm was less than one hour in each instance.

Tables 1 and 2 show the key indicators of the capability of the algorithm to search for a $(1 - \epsilon)$-approximate solution for various numbers of worker threads:

m is the number of points on the plane in the instances of this group;

P is the number of worker threads;

ϵ is the guaranteed relative error of the solution;

S is the number of solved instances in this group out of 20 possible;

p_{avg} is the share, in percent, of discarded solutions averaged over all instances in this group;

$t_{\min}$, t_{avg}, and $t_{\max}$ are the minimal, average, and maximal times in seconds the algorithm took to solve an instance in this group;

$N_{\min}$, N_{avg}, and $N_{\max}$ are the minimal, average, and maximal numbers of vertices of the branching tree inspected by the algorithm when solving instances in this group;

$GAP_{\min}$, GAP_{avg}, and $GAP_{\max}$ are the minimal, average, and maximal values of H/L^{rec} for all instances in this group, where H is the value of the upper bound over the entire set of feasible solutions and L^{rec} is the record value of the objective function when the algorithm stops;

VL_{avg} is the average value, over all examples in this group, of the objective function of the leader on the best noncooperative solution found;

VF_{avg} is the average value, over all instances in this group, of the objective function of the follower on the best noncooperative solution found;

$|x|_{\text{avg}}$ is the average number, over all instances of this group, of facilities opened by the leader in the best noncooperative solution found;

$|z|_{\text{avg}}$ is the average number, over all instances in this group, of facilities opened by the follower in the best noncooperative solution found.

It is clear from Table 1 that the test instances of dimension 20 turn out simple: for all values of ϵ under consideration, $(1 - \epsilon)$-approximate noncooperative solutions were found in less than 10 seconds. Furthermore, the algorithm running in a single thread tackles the problem faster, and, as ϵ grows, its running time decreases as expected. In the case of 12 threads, the running time increases: additional work goes into balancing the load of running threads, which turns out superfluous. Already in dimension 30, work in parallel threads is advantageous, and single-thread implementation begins to lose

Table 1. Results of simulations

m	P	ϵ	S	p_{avg}	$t_{\min}$	t_{avg}	$t_{\max}$	$N_{\min}$	N_{avg}	$N_{\max}$
20	1	0	20	100	0	2	6	1	230	1041
20	1	0.3	20	100	0	1	6	1	177	959
20	1	0.5	20	100	0	1	6	1	125	727
20	12	0	20	100	0	4	11	12	247	1208
20	12	0.3	20	100	1	5	14	12	202	1143
20	12	0.5	20	100	3	6	14	16	150	903
30	1	0	20	100	9	261	807	668	6462	15548
30	1	0.3	20	100	1	202	751	54	4864	14404
30	1	0.5	20	100	0	131	700	1	3247	12450
30	12	0	20	100	15	81	199	836	5854	16120
30	12	0.3	20	100	17	60	154	126	4438	14873
30	12	0.5	20	100	14	55	164	71	2943	13032
40	1	0	3	69	301	3291	3601	3954	55018	165895
40	1	0.3	8	82	2	2738	3601	98	43082	164913
40	1	0.5	14	86	0	1595	3601	1	26781	164742
40	12	0	13	98	125	2306	3602	3490	112770	270576
40	12	0.3	13	98	87	1748	3605	260	78577	266396
40	12	0.5	16	99	70	1210	3603	218	47733	268737
50	1	0	0	27	3600	3600	3607	16279	84001	273633
50	1	0.3	6	50	335	2761	3601	2098	61266	269407
50	1	0.5	11	66	1	1933	3601	1	39936	199281
50	12	0	2	79	1405	3429	3608	47392	320299	1114543
50	12	0.3	8	86	91	2402	3603	2181	216383	1104509
50	12	0.5	12	93	54	1655	3602	124	175120	1016613
60	1	0	0	5	3600	3600	3602	9433	69233	298759
60	1	0.3	1	14	415	3441	3602	1435	47094	118441
60	1	0.5	3	26	3	3065	3601	1	24551	52917
60	12	0	0	34	3600	3601	3603	119133	295625	527817
60	12	0.3	1	46	2145	3455	3603	95273	191101	461580
60	12	0.5	4	77	134	3052	3604	134	116379	355659

Table 2. Numerical characteristics of instances

m	$GAP_{\min}$	GAP_{avg}	$GAP_{\max}$	VL_{avg}	VF_{avg}	$\lvert x\rvert_{\text{avg}}$	$\lvert z\rvert_{\text{avg}}$
20	1.0	2.7	8.3	9.3	23.8	1.6	1.6
30	1.6	5.3	17.7	12.5	57.3	2.3	2.2
40	1.7	4.7	13.6	17.6	84.0	3.1	3.1
50	1.8	4.6	14.0	25.1	128.2	3.5	3.3
60	1.6	3.7	6.4	27.7	145.7	4.7	3.6

its efficiency. However, the speedup is not 12-fold: the running time is decreased roughly by a factor of three, which again is a consequence of nonoptimal balance of loads. The time required to prove the optimality of a solution for instances in this class could be from few seconds to several minutes depending on the instance, but at most 15 minutes for $P = 1$ and 4 minutes for $P = 12$. We observe a turning point at $m = 40$. The single-thread implementation of the algorithm proves the optimality of the solution found only for three out of 20 proposed instances. However, the increase in the number of worker threads, as well as the increase of ϵ, lead to substantial growth of efficiency. To convince ourselves, pay attention to the value of p_{avg} showing the share, averaged over the instances in this class, of the total number of solutions inspected by the algorithm in the allocated time. The examples of low dimensions $m = 20$ and $m = 30$ are completely solved, and for them $p_{\text{avg}} = 100\%$. For $m = 40$, $\epsilon = 0$, and $P = 1$, on average 69% of solutions end up inspected, while, for $P = 12$, so are 98% of solutions on average, while the number of instances in which the optimality of the solution found is proved increases from 3 to 13 in comparison with the single-thread implementation. Instances of high dimension $m = 50$ admit searching for $\frac{1}{2}$-approximate solutions corresponding to $\epsilon = 0.5$: more than half of these instances are solved for both $P = 1$ and $P = 12$. It is worth noting that in the latter case the quantity p_{avg} is 93%. A relatively small increase in the allocated time would enable the algorithm to complete the inspection of the set of feasible solutions. It is clear from the table that the generated instances of dimension $m = 60$ include simple cases. For $P = 1$ and $\epsilon = 0.3$, one instance took about 7 minutes. However, the burst in complexity prevents us from solving the majority of instances in this collection.

Table 2 shows some numerical characteristics concerning the instances generated and the noncooperative solutions found. The values of $GAP_{\min}$ and $GAP_{\max}$ for various m show that each group includes the test instances in which the upper bound is a quite precise estimate for the optimal value of the objective function: for all presented m, we have $GAP_{\min} < 2$. However, for all m under consideration, there are some instances for which the estimate for the upper bound turns out much too high. These instances require more time to solve since unpromising subsets of feasible solutions are discarded on later stages, when they are relatively small. On average the estimate for the upper bound exceeds F^{rec} in 2.7–5.3 times. As we can observe seeing the data on VL_{avg} and VF_{avg}, as well as $|x|_{\text{avg}}$ and $|z|_{\text{avg}}$, on average the leader makes smaller profit than the follower even though opening the same number of facilities.

ACKNOWLEDGMENTS

The authors were supported by the Russian Foundation for Basic Research (projects nos. 12-01-00077-a, 11-07-00474-a, and 12-01-31090) and the Ministry of Education and Science of the Russian Federation (Agreements nos. 8227 and 8230).

REFERENCES

1. V. L. Beresnev, "Local Search Algorithms for the Problem of Competitive Location of Enterprises," Avtomat. Telemekh. No. 3, 12–27 (2012) [Automation and Remote Control **73** (3), 425–439 (2012)].
2. V. L. Beresnev, "Upper Bounds for Objective Functions of Discrete Competitive Facility Location Problems," Diskretn. Anal. Issled. Oper. **15** (4), 3–24 (2008) [J. Appl. Indust. Math. **3** (4), 419–432 (2009)].

3. V. L. Beresnev, *Discrete Competitive Location Problems and Polygons of Boolean Functions* (Inst. Mat., Novosibirsk, 2005) [in Russian].
4. V. L. Beresnev, E. N. Goncharov, and A. A. Mel'nikov, "Local Search with a Generalized Neighborhood in the Optimization Problem for Pseudo-Boolean Functions," Diskretn. Anal. Issled. Oper. **18** (4), 3–16 (2011) [J. Appl. Indust. Math. **6** (1) 22–30 (2012)].
5. V. L. Beresnev and A. A. Mel'nikov, "Approximate Algorithms for the Competitive Facility Location Problems," Diskretn. Anal. Issled. Oper. **17** (6), 3–19 (2010) [J. Appl. Indust. Math. **5** (2), 180–190 (2011)].
6. I. L. Vasil'ev and K. B. Klimentova, "The Branch and Cut Method for the Facility Location Problem with Clients' Preferences," Diskretn. Anal. Issled. Oper. **16** (2), 21–41 (2009) [J. Appl. Indust. Math. **4** (3), 441–454 (2010)].
7. I. L. Vasil'ev, K. B. Klimentova, and Yu. A. Kochetov, "New Lower Bounds for the Facility Location Problem with Clients' Preferences," Zh. Vychisl. Mat. Mat. Fiz. **49** (6), 1055–1066 (2009) [Comput. Math. Math. Phys. **49** (6), 1010–1020 (2009)].
8. A. V. Kononov, Yu. A. Kochetov, and A. V. Plyasunov, "Competitive Facility Location Models," Zh. Vychisl. Mat. Mat. Fiz. **49** (6), 1037–1054 (2009) [Comput. Math. Math. Phys. **49** (6), 994–1009 (2009)].
9. A. V. Plyasunov and A. A. Panin, "The Pricing Problem. Part 1: Exact and Approximate Algorithms," Diskretn. Anal. Issled. Oper. **19** (5), 83–100 (2012) [J. Appl. Indust. Math. **7** (2), 241–251 (2013)].
10. A. V. Plyasunov and A. A. Panin, "The Pricing Problem. Part 2: Computational Complexity," Diskretn. Anal. Issled. Oper. **19** (6), 56–71 (2012) [J. Appl. Indust. Math. **7** (3), 420–430 (2013)].
11. V. Beresnev, "Branch-and-Bound Algorithm for Competitive Facility Location Problems," Comput. Oper. Res. **40**, 2062–2070 (2013).
12. L. Canovas, S. Garcia, M. Labbe, and A. Marin, "A Strengthened Formulation for the Simple Plant Location Problem with Order," Oper. Res. Lett. **35**, 141–150 (2007).
13. I. Davydov, Yu. Kochetov, and A. Plyasunov, "On the Complexity of the $(r|p)$-Centroid Problem in the Plane," TOP (accepted). DOI: 10.1007/s11750-013-0275-y.
14. S. Dempe, *Foundations of Bilevel Programming* (Kluwer Acad. Publ., Dordrecht, 2002).
15. P. L. Hammer and S. Rudeanu, "Pseudo-Boolean Programming," Oper. Res. **17**, 233–261 (1969).
16. J. Krarup and P. M. Pruzan, "The simple plant location problem: survey and synthesis," Eur. J. Oper. Res. No. 12, 36–81 (1983).
17. L. G. Mitten, "Branch-and-Bound Method: General Formulation and Properties," Oper. Res. **18**, 24–34 (1970).
18. H. Stackelberg, *The Theory of the Market Economy* (Oxford Univ. Press, Oxford, 1952).