J Glob Optim (2015) 63:445–460
DOI 10.1007/s10898-013-0130-6

An exact method for the discrete $(r|p)$-centroid problem

Ekaterina Alekseeva · **Yury Kochetov** ·
Alexandr Plyasunov

Received: 31 January 2013 / Accepted: 6 December 2013 / Published online: 21 December 2013

Abstract This paper provides a new exact iterative method for the following problem. Two decision makers, a leader and a follower, compete to attract customers from a given market. The leader opens p facilities, anticipating that the follower will react to the decision by opening r facilities. Each customer patronizes the closest opened facility. The goal is to find p facilities for the leader to maximize his market share. It is known that this problem is Σ_2^P-hard and can be presented as an integer linear program with a large number of constraints. Based on this representation, we design the new iterative exact method. A local search algorithm is used at each iteration to find a feasible solution for a system of constraints. Computational results and comparison with other exact methods show that the new method can be considered as one of the alternative approaches among the most advanced exact methods for the problem.

Keywords Leader–follower problem · Stackelberg game · Voronoi game · Competitive facility location · Voting location

This research was partially supported by RFBR Grants 13-07-00016, 12-01-31090, 12-01-00077. The work of the first author was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement N246016.

E. Alekseeva · Y. Kochetov (✉) · A. Plyasunov
Sobolev Institute of Mathematics, Novosibirsk State University,
4 pr. Akademika Koptyuga, Novosibirsk, Russia
e-mail: jkochet@math.nsc.ru

E. Alekseeva
e-mail: ekaterina2@math.nsc.ru

A. Plyasunov
e-mail: apljas@math.nsc.ru

1 Introduction

In this paper we study one of the competitive facility location problems with two decision makers called a leader and a follower. They make a decision according to their level in a hierarchy. The leader is at the upper level and the follower is at the lower level. They compete to service customers from a given market by opening facilities. At first, the leader decides where to locate p facilities taking into account the follower's reaction. Later on, the follower opens r facilities. Each customer patronizes the nearest opened facility. In case of ties, the leader's facility is preferred. Each customer has a weight (purchasing power or demand). We assume that the weights are essential, that is goods must be consumed, and each customer visits one facility to get them. The weight of each customer is fixed and does not depend on how far from, or close to a facility, the customer is. The leader and the follower obtain a profit from serving the customer which coincides with the weight of the customer. Each decision maker maximizes his own profit or market share. The problem is to define p facilities which should be opened by the leader to maximize his market share. We assume that the number of customers is finite and facilities can be opened at the finite number of sites.

The history of the competitive location models is originated from the Hotelling work [15]. The formalization of this class of problems and fundamental complexity results were established by Hakimi [14]. Later the different groups of scientists studied the complexity status for some particular cases of the $(r|p)$-centroid problem [12,16,22,23,28]. Thus, this problem is a back-breaker due to its Σ_2^P-completeness even in the case of the Euclidean distances between customers and facilities. Nevertheless, some exact approaches based on enumeration ideas have been developed taking into account the combinatorial nature of the problem [10,13]. Due to the laboriousness of these approaches the new appeared heuristics based on local search procedures have became more prominent techniques. They have been applied successfully for some location problems. Moreover, they have opened a new wide class of algorithms to tackle bi-level mixed integer programming problems. The readers can find the heuristical approaches for the discrete $(r|p)$-centroid problem based on tabu search [3,6,27], genetic algorithm [3], particle swarm optimization [9], p-median solution heuristics [2]. Before developing a heuristic, one should remember that many heuristics are based on the fact that the objective function of the underlying problem is polynomially computable. This concerns many metaheuristics such as hill climbing, variable neighborhood search, tabu search, simulated annealing, genetic algorithms and others. For the discrete $(r|p)$-centroid problem, in order to calculate the leader's objective function, we have to solve the follower's problem which is NP-hard in the strong sense. Hence, we could not immediately apply these heuristic approaches. An approach based on using metaheuristics at each level might find only so called semi-feasible solutions. To overcome this obstacle, the hybridization of heuristics at the upper level with exact approaches at the lower level has been proposed in [3,7,11].

In this paper we present a method which is an improved version of the exact iterative method previously developed in [3]. We use a new single level reformulation of the discrete $(r|p)$-centroid problem suggested by Roboredo and Pessoa [25]. This reformulation has a polynomial number of variables and exponentially many constraints. If we extract a small family of the constraints (the follower strategies), we get an upper bound for the global maximum. Our method iteratively increases the family and terminates when the upper bound coincides with a lower bound. At each iteration we have a family of constraints and try to find the best leader's solution against this family. A local search heuristics is used to this end. To accelerate the search, we adapt the data structures developed for the p-median problem by Resende and Werneck [26]. The method is able to solve optimally the instances with

$p = r \in \{5, 10, 15\}$ for the *Euclidean* benchmarks and produces near optimal solutions for the case $p = r = 20$. Moreover, for the first time we present computational results for a class *Uniform*. It is more difficult class than the class *Euclidean* for the uncapacitated facility location problem [19]. We confirm the difficulty of this class for the discrete $(r|p)$-centroid problem.

The paper is organized as follows. In Sect. 2, we present a bi-level mixed integer formulation of the problem. It demonstrates the nature of the problem and allows us to define clearly the conceptions of optimal and feasible solutions. In Sect. 3, a single-level mixed integer linear reformulation is presented. We discuss the complexity status of this problem for the small number of constraints. In Sect. 4, we formulate a feasibility problem which we solve instead of the full single-level reformulation and develop a local search algorithm with a randomized neighborhood. In Sect. 5, a main framework of the method is presented. Finally, comparative computational results and conclusions are discussed in Sects. 6 and 7.

2 Problem formulation

Let I be a set of m potential facility locations and J be a set of n customers' locations. Each element of matrix (d_{ij}) defines the distance between customer $j \in J$ and facility $i \in I$. Each component of positive vector (w_j) defines the weight of customers j.

Let X and Y be the sets of locations occupied by the leader and the follower, respectively. Denote as $d(j, X)$ and $d(j, Y)$ the distance between customer j and its nearest facility from X and Y, respectively. Customer j prefers Y over X if $d(j, Y) < d(j, X)$ and prefers X over Y otherwise. Let $J(Y \prec X)$ be a set of customers that prefer Y over X. The total weight of the customers that prefer Y to X is denoted as $W(Y \prec X)$. Therefore, $W(Y \prec X) = \sum_{j \in J(Y \prec X)} w_j$.

For each X, the follower's strategy is a set of others locations Y that provides the follower with the maximal market share. This market share, denoted as $W^*(X)$, is a solution of the following problem:

$$W^*(X) = \max_{Y, |Y|=r} W(Y \prec X).$$

This problem is called the follower's problem. The leader is interested in a strategy that maximizes his (or her) market share or, in other words, minimizes the follower's market share. This strategy denoted as X^* is a solution to the following problem:

$$W^*(X^*) = \min_{X, |X|=p} W^*(X).$$

This problem is called the leader's problem. Once strategy X^* is found, the leader's market share is calculated as $\sum_{j \in J} w_j - W^*(X^*)$. In the $(r|p)$-centroid problem we need to find X^* and the corresponded leader's market share [4,10,11].

Now we present a 0–1 bi-level reformulation of the problem that we use in solving approach. Let us introduce the decision variables:

$$x_i = \begin{cases} 1 & \text{if facility } i \text{ is opened by the leader,} \\ 0 & \text{otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if facility } i \text{ is opened by the follower,} \\ 0 & \text{otherwise,} \end{cases}$$

$$z_j = \begin{cases} 1 & \text{if customer } j \text{ is served by the leader,} \\ 0 & \text{if customer } j \text{ is served by the follower.} \end{cases}$$

Denote $x = (x_i)$, $y = (y_i)$, $i \in I$, and $z = (z_j)$, $j \in J$, for short. Remember that each customer is served either by the leader or the follower. Define the set of facilities $I_j(x)$ which allows the follower to capture customer j if the leader uses a solution x:

$$I_j(x) = \left\{ i \in I \mid d_{ij} < \min_{l \in I \mid x_l = 1} d_{lj} \right\}, \quad j \in J.$$

Note that we deal with conservative customers [14]. It means that if a customer has the same distances to the closest leader's and the closest follower's facilities, he (or she) prefers the leader's facility. Hence, the follower never opens a facility at the same site where the leader has already occupied. Now the problem can be rewritten as follows [3,20]:

$$\max_x \sum_{j \in J} w_j z_j^* \tag{1}$$

subject to

$$\sum_{i \in I} x_i = p, \tag{2}$$

$$x_i \in \{0, 1\}, \quad i \in I, \tag{3}$$

where $z^* = z(x)$ is a component of the optimal solution to the follower's problem:

$$\max_{y,\, z} \sum_{j \in J} w_j (1 - z_j) \tag{4}$$

subject to

$$\sum_{i \in I} y_i = r, \tag{5}$$

$$1 - z_j \le \sum_{i \in I_j(x)} y_i, \quad j \in J, \tag{6}$$

$$x_i + y_i \le 1, \quad i \in I, \tag{7}$$

$$y_i, z_j \in \{0, 1\}, \quad i \in I, \quad j \in J. \tag{8}$$

The objective function (1) defines the market share of the leader. Equation (2) guarantees that the leader opens exactly p facilities. The objective function (4) defines the market share of the follower. Equation (5) guarantees that the follower opens exactly r facilities. Constraints (6) determine the market shares of the decision makers. If the follower has no facility in the set $I_j(x)$, then customer j is served by the leader. Constraints (7) guarantee that each facility can be opened by at most one decision maker. Actually, these constraints are redundant. Nevertheless, we use them to reduce the feasible domain of the follower's problem. Note that we can drop the integrality constraints for the variables z. The optimal value is not changed in this case. Thus, the formulation (1)–(8) can be rewritten as a mixed integer bi-level program. Note that the follower problem is linear for each leader solution x and we can use, for example, CPLEX software for finding the best strategy for the follower.

Let us remind some definitions which help to understand the nature of the bi-level problem deeper.

Definition 1 [5] The triple (x, y, z) is called a *semi-feasible solution* to the bi-level problem (1)–(8) if and only if x satisfies the constraints (2)–(3) and the pair (y, z) satisfies the constraints (5)–(8).

Definition 2 [5] The semi-feasible solution (x, y, z) is called a *feasible solution* to the bi-level problem (1)–(8) if and only if the pair (y, z) is an optimal solution to the follower's problem (4)–(8).

Note that semi-feasible solutions can be found in polynomial time. To find feasible solutions, we have to solve the follower's problem, which is NP-hard in the strong sense even for Euclidean distances between facilities and customers [12].

For the feasible solution (x, y, z), let us denote the value of the leader's objective function as $W(x, y, z)$. Note that the follower's problem may have several optimal solutions for a given x. As a result, the leader's problem can be ill-posed. Fortunately, the sum of leader's and follower's objective functions is $\sum_{j\in J} w_j$, i.e. a constant. Thus, two feasible solutions (x, y^1, z^1) and (x, y^2, z^2) give the same objective values for the leader and the follower. We deal with a well-defined two players constant-sum game. Hence, we can define the optimal solution as the best feasible solution.

Definition 3 The solution (x^*, y^*, z^*) is called an *optimal solution* if and only if $W(x^*, y^*, z^*) \geq W(x, y, z)$ for each feasible solution (x, y, z).

In case of elastic or non-essential demands we deal with inconstant-sum game. Hence, we should distinguish cooperative and noncooperative follower's behaviors and modify these definitions accurately. The readers can find the corresponded definitions in [2] and how to work out in these cases of the competitive facility locations, for example, in [7].

3 A single-level reformulation

In spite of its complexity status, the problem admits a single level linear programming formulation with polynomially many variables and exponentially many constraints. Nevertheless, neither a polynomial formulation nor a formulation where the constraints can be separated in polynomial time is possible unless NP $= \Sigma_2^P$.

Originally, a single-level reformulation with an exponential number of constraints and variables was presented by Campos-Rodríguez and Moreno Pérez [8]. At that time it was the first single-level reformulation for the discrete $(r|p)$-centroid problem. Alekseeva et al. [3] presented another reformulation with an exponential number of constraints and variables. Later on, a new improved reformulation (9)–(14) with an exponential number of constraints and polynomially many variables was proposed by Roboredo and Pessoa [25]. This formulation is important for our method. We present and discuss it here.

Suppose that the leader has opened facilities but the follower has not opened yet. Introduce the new binary variables:

$$z_{ij} = \begin{cases} 1 & \text{if customer } j \text{ patronizes the leader facility } i, \\ 0 & \text{otherwise,} \end{cases}$$

and a positive variable W which means the leader's market share after the follower's reaction. Let $\mathscr{F}$ be the set of all possible follower's solutions. Each $y^f \in \mathscr{F}$ defines a set of r facilities opened by the follower. For each y^f we define a set of the facilities

$$I_j(y^f) = \left\{ i \in I \mid d_{ij} \leq \min_{l \in I, y_l^f = 1} d_{lj} \right\}, \quad j \in J$$

which allows the leader to keep the customer j if the follower uses the solution y^f. The discrete $(r|p)$-centroid problem can be reformulated as a single-level mixed integer linear problem:

$$\max_{W,\ x,\ z} W \tag{9}$$

subject to

$$\sum_{i \in I} x_i = p, \tag{10}$$

$$0 \leq z_{ij} \leq x_i, \quad i \in I, j \in J, \tag{11}$$

$$\sum_{i \in I} z_{ij} = 1, \quad j \in J, \tag{12}$$

$$W \leq \sum_{j \in J} \sum_{i \in I_j(y^f)} w_j z_{ij}, \quad y^f \in \mathscr{F}, \tag{13}$$

$$W \geq 0, \quad x_i, \quad z_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J. \tag{14}$$

The objective in (9) is to maximize the market share of the leader. Constraint (10) indicates as before that the leader opens p facilities. Constraints (11) ensure that the leader can serve customers from opened facilities only. Constraints (12) indicate that each customer patronizes exactly one leader's facility. Finally, constraints (13) ensure that the follower uses the best own solution to minimize the market share for the leader. Note that we can remove the integrality constraint for the variables (z_{ij}) without loss of generality.

Reformulation (9)–(14) contains a polynomial number of variables and an exponential number of constraints. Unfortunately, it is NP-hard problem even when the follower uses small subset of the set $\mathscr{F}$.

Theorem 1 *The problem* (9)–(14) *is NP-hard problem even when the follower uses only two strategies from the set* $\mathscr{F}$.

Proof Consider the following *Set partitioning problem*. Given a finite set V, $|V| = m$, and a positive integer weight $a(i)$ for each element $i \in V$. Is there a partition of V into two subsets V_1 and V_2, such that $\sum_{i \in V_1} a(i) = \sum_{i \in V_2} a(i) = A$?

We reduce this NP-hard problem to problem (9)–(14) constrained to two follower's solutions. For each element $j \in V$ we introduce three customers j, j', and j'' with weights $w_{j'} = w_{j''} = a_j$ and $w_j = a_{max} = \max_{i \in V} a_i$. The total set of customers is $J \cup J' \cup J''$. Each set J, J', and J'' has cardinality m. Denote $J \cup J' \cup J''$ as $\overline{J}$. The set of candidate facility locations is I such that $I = \overline{J}$. The distances d_{ik}, for all $i \in I$, $k \in \overline{J}$ we define as follows:

$$d_{ik} = \begin{cases} 0 & \text{if } i = k, \\ 1 & \text{if } i \neq k, i = j' \text{ or } i = j'', \\ 2 & \text{otherwise.} \end{cases}$$

Put $p = r = m$ and define two follower's solutions y^1 and y^2:

$$y_i^1 = \begin{cases} 1 & \text{if } i \in J' \\ 0 & \text{otherwise,} \end{cases} \qquad y_i^2 = \begin{cases} 1 & \text{if } i \in J'' \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that the desired partition of the set V exists if and only if the leader market share is $(ma_{max} + 3A)$. In this case he (or she) opens the facilities in J' for customers from the V_1 and in J'' for the customers from V_2. The solutions y^1 and y^2 give the same market share A to the follower. The leader has got $(ma_{max} + 3A)$. It is global leader's maximum.

If the desired partition does not exist, then one of the solutions y^1 or y^2 guarantees that the follower has the market share greater than A. □

The case of one follower's solution is not interesting. The leader opens facilities at the same sites and gets the whole market if $p \geq r$. In case of several follower's solutions, the problem is NP-hard.

4 Feasibility problem and heuristics

Let us replace the set $\mathscr{F}$ by a subset F and solve problem (9)–(14) with F instead of $\mathscr{F}$. The optimal solution provides an upper bound $W(F)$ at the global maximum for the leader's market share. Our method iteratively increases the subset F and terminates when the upper bound coincides with a lower bound which derived by the best found solution. Thus, the method is exact and finite. To improve the upper bound, we apply the lifted inequalities suggested by Roboredo and Pessoa [25].

4.1 Lifted inequalities

The main idea of these inequalities comes from the following observation. When we compute the upper bound $W(F)$, some of the constraints (13) make the upper bound weaker if the leader locates a facility opened in y^f. To avoid it, a function $H : y^f \to I$ is defined. It gives an alternative place for each facility opened by the follower, if the original place has already been used by the leader. Then, in addition to the follower's solution y^f, the upper bound takes into account the solution that replaces some facilities i such that $y_i^f = 1$ by $H(i)$. Now the new family of inequalities is as follows:

$$W \leq \sum_{j \in J} \sum_{i \in I_j(y^f) \cup \widetilde{I}_j(y^f)} w_j z_{ij}, \quad y^f \in F, \tag{15}$$

where

$$I_j(y^f) = \left\{ i \in I \mid y_i^f = 0 \text{ and } d_{ij} \leq \min_{k \in I}(d_{kj} | y_k^f = 1) \right\},$$

$$\widetilde{I}_j(y^f) = \left\{ i \in I \mid y_i^f = 1 \text{ and } d_{ij} \leq \min \left\{ \min_{k \in I}(d_{kj} | y_k^f = 1),\ d_{H(i)j} \right\} \right\}, \quad j \in J.$$

Set $I_j(y^f)$ contains the locations where no follower's facility is opened for solution y^f and leader would capture customer j if he (or she) opens a facility at one of these locations. Set $\widetilde{I}_j(y^f)$ contains the locations where the follower has opened a facility and a facility would keep customer j for the leader even if the follower moves the facility i to its alternative location $H(i)$. In other words, for each follower's solution y^f, the set $I_j(y^f) \cup \widetilde{I}_j(y^f)$ contains the locations that allow the leader to keep customer j if the follower moves its facility from i to the alternative location $H(i)$.

Roboredo and Pessoa have shown that the lifted inequalities (15) are valid. For finding some violated cuts, they have defined a separation problem. Given a fractional solution $(\bar{W}, \bar{x}, \bar{z})$ that satisfies (10)–(12), and some of the constraints (15), the separation problem is to find the follower's solution $y^f \in \mathscr{F}$ and a corresponding function $H : y^f \to I$ that minimizes the market share of the leader. But other functions might be adapted as well. To avoid the separation problem at each step of the method, we use a static function H that is an alternative closest site for each facility.

4.2 Feasibility problem

Denote a lower bound for bi-level problem (1)–(8) as W' and consider the following feasibility problem:

$$W > W', \tag{16}$$

$$W \le \sum_{j\in J} \sum_{i\in I_j(y^f)\cup\widetilde{I}_j(y^f)} w_j z_{ij}, \quad y^f \in F, \tag{17}$$

$$\sum_{i\in I} x_i = p, \tag{18}$$

$$z_{ij} \le x_i, \quad i \in I, \quad j \in J, \tag{19}$$

$$\sum_{i\in I} z_{ij} = 1, \quad j \in J, \tag{20}$$

$$W \ge 0, x_i, z_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J. \tag{21}$$

If we have a feasible solution $x(F)$ to this system then the leader's a market share is greater than W' when the follower reacts selecting his best strategy from the family F. In this case, we solve follower's problem (4)–(8) for $x(F)$ and include the optimal solution $y(F)$ into the set F. Otherwise, when this problem is infeasible, W' is the global maximum to problem (1)–(8). Note that the feasibility problem is easier than the optimization one because we do not need to prove the optimality. We can apply IP solvers or heuristics for finding a feasible solution. We use a local search procedure with a randomized neighborhood to this end.

4.3 Local search procedure for the feasibility problem

Here we describe a local search procedure to the feasibility problem. It provides us with a new leader's solution $x(F)$ against the set F. We apply the randomized local search although other meta-heuristics, for example, Variable Neighborhood Search, Simulated Annealing, or Genetic Algorithms [17,30], can be adapted as well.

The local search procedure focuses on the leader's variables. The basic attribute of the local search is a neighborhood. We adapt the well-known swap neighborhood. It contains all the leader's solutions which can be obtained from the current solution by closing one leader's facility and opening another one. The size of the neighborhood is $p(m-p)$. For each neighboring solution we have to calculate the market share of the leader. It is time-consuming for large F. Thus, we use randomization. This procedure independently includes each element of the swap neighborhood with a fixed probability q in the randomized neighborhood denoted N_q to prevent the local search from coming back to the previously visited solutions and cycling.

Figure 1 shows the framework of the local search. As an initial solution at Step 1 we can take the best solution from the previous iteration with smaller set F. At Step 2.2 we have to compute the objective function value for each element of N_q. To reduce the running time, in addition to randomization, we adapt the procedure developed for the p-median problem by Resende and Werneck [26]. This procedure finds the most prominent pair of the opened-closed facilities for a current leader's solution and results in the best neighboring solution. Due to the special data structures, it is significantly faster than the straightforward calculations.

Following Resende and Werneck, we define four components: $gain(y^f, i_o)$, $loss(y^f, i_c)$, $extra(y^f, i_o, i_c)$ and $profit(y^f, i_o, i_c)$, where i_o (i_c) is a candidate facility for opening (closing) in the leader's solution. We calculate $gain(y^f, i_o)$ to estimate potential leader's

1 Choose an initial leader's solution.
2 Repeat the following until the stopping condition is met:
  2.1 Generate the randomized neighborhood N_q;
  2.2 If $N_q \neq \emptyset$ then
    Find a neighbor from N_q with the best market share against F
    else go to Step 2.1;
  2.3 Move to the best neighboring solution;
3 Return the best found leader's solution.

Fig. 1 Local search procedure

gain from opening i_o and $loss(y^f, i_c)$ to estimate potential leader's losses from closing i_c if the follower uses y^f. We have

$$profit\left(y^f, i_o, i_c\right) = gain\left(y^f, i_o\right) - loss\left(y^f, i_c\right) + extra\left(y^f, i_o, i_c\right),$$

where $extra(y^f, i_o, i_c)$ intends to correct the estimation due to some particular cases arising after closing i_c and opening i_o. The value $profit(y^f, i_o, i_c)$ means the effect of opening-closing facilities under a given the follower's solution y^f. The follower's decision is a solution which delivers $\min_{y^f \in F} profit(y^f, i_o, i_c)$.

Resende and Werneck have shown that we can avoid calculating these four components straightforwardly. We calculate them only once for each candidate facility and use these values during one step of the neighborhood search.

As the stopping condition, we use the total number of iterations or inequality (16). If we cannot find a feasible solution by the local search, we call the branch-and-bound method from an IP solver. Our computational experiments demonstrate that the local search allows us to decrease calling the IP solver significantly.

5 Framework of the method

Figure 2 presents the framework of the exact iterative method for bi-level problem (1)–(8). It combines metaheuristics for the feasibility problem and mathematical programming tools for the follower's problem. The main idea is to find a subset F so that the corresponded feasibility problem becomes infeasible. To create an initial subset F, we apply the alternating heuristics and heuristics based on the p-median solution [2]. For finding a tight lower bound W' at Step 1, we use the hybrid memetic algorithm [3]. The efficiency of the method strongly depends on the size of the final set F. Ideally, if we find the set F at Step 1 that contains the reaction of follower to the best strategy of the leader and corresponded value W'. Because in this case we need to check the feasibility of the system only once. Otherwise, we have to solve the system many times improving the set F.

1 Create an initial subset F and find a lower bound W'.
2 Find a feasible solution $x(F)$ to system (16)–(21).
  If it is infeasible then return the best found solution and stop.
3 Solve the follower's problem for $x(F)$ and find optimal solution $y(F)$
  and lower bound $LB(F)$.
4 If $W' < LB(F)$ then $W' := LB(F)$.
5 Include $y(F)$ into the subset F and go to Step 2.

Fig. 2 The iterative exact method

At Step 2 we apply the local search procedure for the feasibility system. If we cannot find a feasible solution $x(F)$, we call a branch-and-bound method, for example, CPLEX software to check infeasibility. If the feasible solution is found, we need a corresponded follower's response $y(F)$ and the leader's market share $LB(F)$ for this solution $x(F)$. To this end, we solve the follower's problem at Step 3 exactly by the IP solver. At Step 4 we improve the best found value W' for the leader if it is possible. Finally, at Step 5, we update the set F and continue computations from Step 2.

Let us consider a positive parameter ε and replace inequality (16) in the feasibility problem by the following:

$$W > (1+\varepsilon)W'.$$

In this case our method allows us to find approximate solutions with at most ε relative gap of the optimum. Our computational experiments presented in the next section demonstrate that we are able to find optimal and approximate solutions within a reasonable time.

6 Computational experiments

The method has been tested and compared on the instances from the benchmark library *Discrete Location Problems* [1]. This electronic library contains the test instances for the facility location problems including the competitive ones. For all instances, customers and facilities are at the same sites. For the discrete $(r|p)$-centroid problem we have two classes of benchmarks which are differed in the way of generating of the matrix (d_{ij}). For the first class called *Euclidean* the sites are chosen in $7{,}000 \times 7{,}000$ square uniformly and the elements of matrix (d_{ij}) are Euclidean distances between sites i and j. For the second class called *Uniform* the elements of the matrix (d_{ij}) are chosen from (0, 10,000) interval uniformly and $d_{ji} = d_{ij}$ for all $j \in J, i \in I$. Each class has 10 test instances. There are two cases for each instance with respect to the customer's weight. In the first case, all customers are identical and $w_j = 1$, for all $j \in J$. In the second case, all customers are different and each weight w_j is chosen from (0, 200) interval uniformly for all $j \in J$. The size of instances is $m = n = 100$ and $p = r \in \{5, 10, 15, 20\}$.

Our experiments have been carried out on a PC Intel Xeon X5675, 3 GHz, RAM 96 Gb, running under the Windows Server 2008 operating system. We have used CPLEX 12.3. as an optimization solver. Below we present comparative computational results which have been carried out on a computer with other specifications. Namely, the branch-and-cut method by Roboredo and Pessoa [25] has been performed on a PC Pentium Intel Core 2 duo, 2.13 GHz, RAM 2 Gb and used CPLEX 12.1. We are not familiar with the best way to compare these two machines' efficiency. Thus, without being exact, we consider that PC Intel Xeon X5675 is 1.5 times more efficient than PC Pentium Intel Core 2 duo.

The method uses matheuristics for finding strong initial solution to the leader's problem and a corresponding tight lower bound for the global optimum. Our method tries to improve it and then prove its optimality. Actually, the test instances are not difficult for the metaheuristics. They are able to find the optimal solutions [3,25]. However, proving of its optimality is a burdensome problem for exact method.

At the first experiment, we have explored the influence of parameters p and r on the total number of iterations of the method. We have considered the instances from the class *Euclidean* with $m = n = 50$, $p = r$ and solved them with the different values p and r. Figure 3 shows the cardinality of the final set F that is the total number of iterations. The problem is easy and the total number of iterations is reasonable when p and r are less than

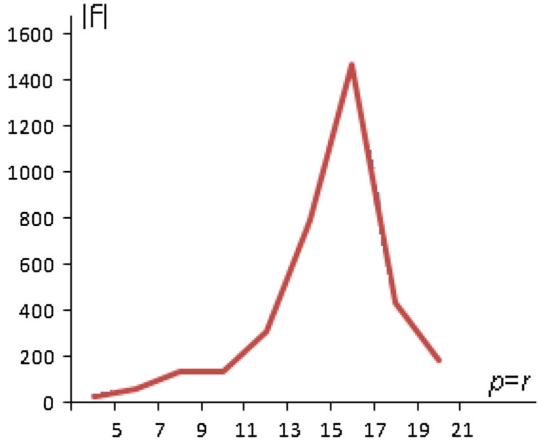

Fig. 3 The total number of iterations depending on p and r, $n = m = 50$, class *Euclidean*

15 or greater than 17. The problem becomes hard when $15 \leq p = r \leq 17$ and the method needs more than one thousand follower's solutions at the final iteration. We conclude that the case $p = r = [m/3]$ might be the most difficult for the method even if we have the optimal value at the first step of the method. Checking the feasibility of the system with a large set F is the main reason why we cannot solve the large scale instances.

Tables 1, 2, 3, 4, 5 and 6 present computational results: Tables 1, 2, 3 and 4 for class *Euclidean*; Tables 5, 6 for class *Uniform*. In each Table the column $Instance$ indicates a code name of a test instance. The column Opt contains the leader's market share for the optimal solution, the columns IM and BC show the total CPU time in minutes consumed by the iterative method addressed in this paper and the branch-and-cut method by Roboredo and Pessoa, respectively. The column $5\,\% \cdot Opt$ contains a leader's market share with a gap of at most 5 % of the optimal value for the IM. The columns $|F^{opt}|$ and $|F|$ present the total number of iterations consumed by the IM to prove optimality or find an approximate solution, respectively.

Table 1 $m = n = 100$, $p = r = 5$, class *Euclidean*

Instance	$w_j = 1,\ j \in J$				$w_j \in (0{,}200),\ j \in J$			
	Opt	Time (min)		$\lvert F^{opt}\rvert$	Opt	Time (min)		$\lvert F^{opt}\rvert$
		IM	BC			IM	BC	
111	47	6	44.2	108	4,139	1	27.4	91
211	48	1	58.7	85	4,822	4	159.9	148
311	45	26	202.8	219	4,215	38	313.7	233
411	47	2	52.0	103	4,678	69	74.1	378
511	47	1	39.4	97	4,594	16	469.0	201
611	47	3	51.0	102	4,483	2	25.7	127
711	47	4	53.4	99	5,153	5	130.9	167
811	48	1	35.7	68	4,404	2	195.5	178
911	47	2	44.8	94	4,700	13	290.3	208
1,011	47	2	66.2	127	4,923	1	18.3	72

Table 2 $m = n = 100$, $p = r = 10$, class *Euclidean*

Instance	$w_j = 1,\ j \in J$				$w_j \in (0, 200),\ j \in J$			
	Opt	Time (min)		$\|F^{opt}\|$	Opt	Time (min)		$\|F^{opt}\|$
		IM	BC			IM	BC	
111	50	13	38.1	485	4,361	60	170.3	678
211	49	20	78.5	488	5,310	42	158.1	798
311	48	195	222.8	959	4,483	146	317.9	1,209
411	49	135	188.6	873	4,994	33	229.1	679
511	48	270	315.4	1,231	4,906	399	1,340.2	1,288
611	47	900	381.8	1,443	4,595	143	859.7	1,503
711	51	12	42.2	382	5,586	73	339.2	773
811	48	145	259.9	888	4,609	152	446.8	1,198
911	49	102	187.3	877	5,302	6	39.6	345
1,011	49	180	237.1	733	5,005	97	562.7	1,301

Table 3 $m = n = 100$, $p = r = 15$, class *Euclidean*

Instance	$w_j \in (0{,}200),\ j \in J$		
	Opt	Time (min)	
		IM	BC
111	4,596	72	162.53
211	5,373	3,845	1,349.27
311	4,800	395.00	461.78
411	5,064	1,223	1,402.33
511	5,131	2,120	1,318.32
611	4,881	2,293	472.37
711	5,827	1,320	810.00
811	4,675	4,570	1,919.73
911	5,158	>600	>600
1,011	5,195	>600	1,200.57

The columns IM and BC in Tables 1 and 2 show that the iterative method might be better for test instances with $m = n = 100$ and $p = r = 5$, 10 than the branch-and-cut method in comparison with the computational time.

Table 3 shows that the IM and BC methods take a lot of computational efforts for the case $p = r = 15$. It means that these instances become difficult for both methods. Figure 3 illustrates the reason why this case is so hard. It deals with the growth of the subfamily F as the values of p and r increase. The problem becomes the most difficult when the values of p and r equal to about a one-third of m. The computational time for the BC algorithm increases as both the number of branch-and-bound nodes and the number of cuts generated expand [25]. Thus, for more difficult instances we have to develop other exact approaches or apply approximate methods.

Table 4 shows the comparative results concerning the approximate solutions. The column $GapBC$ shows a root gap between the best known lower bound and the best upper bound

Table 4 $m = n = 100$, $p = r = 20$, class *Euclidean*

Instance	$w_j \in (0, 200)$, $j \in J$			
	$5\,\% \cdot Opt$ IM	Time IM (min)	Gap (%) BC	Time BC (min)
111	4,737.6	1	3.83	>600
211	5,703.6	185	3.35	>600
311	5,137.65	248	3.05	>600
411	5,677.81	5	1.99	>600
511	5,600.7	110	1.61	>600
611	5,199.6	190	1.90	>600
711	6,187.65	97	7.28	>600
811	5,100.9	570	2.46	>600
911	5,731.95	165	2.41	>600
1,011	5,668.95	130	1.64	>600

Table 5 $m = n = 100$, $p = r = 5$, class *Uniform*

Instance	$w_j = 1$, $j \in J$		$w_j \in (0{,}200)$, $j \in J$	
	Opt	$\|F^{opt}\|$	Opt (market share in %)	$\|F^{opt}\|$
111	48	95	4,901 (46.6)	194
211	49	66	4,963 (47.2)	153
311	47	143	4,807 (45.7)	315
411	47	136	4,941 (46.9)	146
511	48	79	5,017 (47.7)	125
611	48	143	4,840 (46.0)	223
711	51	35	5,224 (49.7)	66
811	47	138	4,963 (47.2)	103
911	47	121	5,010 (47.6)	123
1,011	48	102	4,931 (46.9)	173

Table 6 $m = n = 100$, $p = r = 10$, class *Uniform*

Instance	$w_j = 1$, $j \in J$		$w_j \in (0{,}200)$, $j \in J$	
	$5\,\% \cdot Opt$	$\|F\|$	$5\% \cdot Opt$ (market share in %)	$\|F\|$
111	48	379	5,125 (48.7)	373
211	54	41	5,157 (49.0)	339
311	50	280	5,374 (51.1)	236
411	48	499	5,022 (47.7)	578
511	48	464	5,220 (49.6)	261
611	49	486	5,273 (50.1)	407
711	51	115	5,288 (50.3)	197
811	48	326	4,949 (47.0)	486
911	50	214	5,164 (49.1)	409
1,011	51	115	5,241 (49.8)	394

obtained by the branch-and-cut algorithm [25]. We can see that even for the problems where the gap has been about 2 %, the running time has reached 10 hours. It is unlikely that IM would be able to find solutions with a lesser gap for admissible computational efforts but to find solutions within 5 % of the optimum by the IM takes a reasonable time on a powerful computer.

The instances for the class *Uniform* have been solved for the first time in this paper. According to the related works this class of test instances is more difficult than the class *Euclidean* for the uncapacitated facility location problem [19]. In case of the discrete $(r|p)$-centroid problem we have observed the same. The columns $|F^{opt}|$ in Tables 1 and 5 support this class characteristic on the average. Thus we were able to optimally solve the *Uniform* class instances with $p = r = 5$ but IM did not optimally solve the same instances for $p = r = 10, 15$.

We can conclude that the existing exact methods IM and BC are good enough since they are able to tackle previously open instances with up to 100 customers, 100 potential facilities and $p = r = 15$ in the Euclidean case for a reasonable time. They are alternative approaches. The IM can be faster for relatively small instances and well suited to obtain approximate solutions with a priori gap. The BC runs faster on the relatively large instances. Nevertheless, both of them leave room for further improvements.

7 Conclusions

Since the nineties of the previous century, the competitive facility location models become more and more asked-for and create an active field of research. In this paper we have discussed the fundamental model in this field formalized by Hakimi, so-called the discrete $(r|p)$-centroid problem. We have considered a discrete case, one decision criterion based on the travel distances, the binary oriented customers' behavior, and essential demands. Future research direction could be connected with several aspects such as the introduction of the quality measures for the facilities, different customers' decision rules, non-essential demands (the customers not necessarily use all their buying power, a part of the demand could be not satisfied), etc. Some ideas may be found in [18,21,24,29].

The multicriteria models in competitive facility location could be considered included in the design of new models. In some models the service quality is incorporated in the model as a decision variable, the quality has a cost and the profit is a non-increasing function of this variable. In this case, a bi-criteria optimization problem may be considered, one of the objectives is to maximize the income and the other one is to minimize the cost associated to the quality. Normally, the objective of maximizing the profit and the objective of maximizing the service quality are in conflict, in this case the multicriteria optimization techniques are useful.

Our local search procedure shows excellent results. Sure, it is time consuming approach but we have got the global optimum by the relatively small number of steps. We guess that the matheuristics are useful for bi-level optimization. We can apply these methods for the discrete bi-level problems but have to spend a lot of efforts for computing the objective function values. We have proposed an exact approach and compared it with the branch-and-cut method by Roboredo and Pessoa. They use different ideas but they are based on the same single level reformulation. Such formulation is easy to create for the min-max problems. Is it possible to adopt these methods for other competitive location models with multiple criteria? We believe that exact methods and matheuristics for this case are very interesting for further research.

Acknowledgments Many thanks to Alexander Kononov for the fruitful discussions.

References

1. Alekseeva, E., Beresnev, V., Kochetov, Y. et al.: Benchmark library: Discrete Location Problems (Russia), http://math.nsc.ru/AP/benchmarks/Competitive/p_me_comp_eng.html
2. Alekseeva, E., Kochetov, Y.: Matheuristics and exact methods for the discrete (r|p)-centroid problem. In: Talbi, E.-G. (ed.) Metaheuristics for bi-level optimization. Studies in Computational Intelligence, vol. 482, pp. 189–219. Springer, Berlin (2013)
3. Alekseeva, E., Kochetova, N., Kochetov, Yu., Plyasunov, A.: A heuristic and exact methods for the discrete $(r|p)$-centroid problem. LNCS 6022, pp. 11–22 (2010)
4. Banik, A., Das, S., Maheshwari, A., Smid, M.: The discrete Voronoi game in a simple polygon. LNCS 7936, pp. 197–207 (2013)
5. Ben-Ayed, O.: Bilevel linear programming. Comput. Oper. Res. **20**(5), 485–501 (1993)
6. Benati, S., Laporte, G.: Tabu search algorithms for the $(r|X_p)$-medianoid and $(r|p)$-centroid problems. Locat. Sci. **2**, 193–204 (1994)
7. Beresnev, V., Melnikov, A.: Approximate algorithms for the competitive facility location problem. J. Appl. Ind. Math. **5**(2), 180–190 (2011)
8. Campos-Rodríguez, C.M., Moreno Pérez, J.A.: Multiple voting location problems. Eur. J. Oper. Res. **191**, 437–453 (2008)
9. Campos-Rodríguez, C.M., Moreno Pérez, J.A., Santos-Peñate, D.R.: Particle swarm optimization with two swarms for the discrete $(r|p)$-centroid problem. LNCS 6927, pp. 432–439 (2012)
10. Campos-Rodríguez, C.M., Santos-Peñate, D.R., Moreno-Pérez, J.A.: An exact procedure and LP formulations for the leader-follower location problem. TOP **18**(1), 97–121 (2010)
11. Davydov, I., Kochetov, Y., Carrizosa, E.: VNS heuristic for the $(r|p)$-centroid problem on the plane. Electron. Notes Discrete Math. **39**, 5–12 (2012)
12. Davydov I., Kochetov Y., Plyasunov A.: On the complexity of the $(r|p)$-centroid problem in the plane. TOP (2013). doi:10.1007/s11750-013-0275-y
13. Ghosh, A., Craig, C.: A Location allocation model for facility planning in a competitive environment. Geogr. Anal. **16**, 39–51 (1984)
14. Hakimi, S.L.: Locations with spatial interactions: competitive locations and games. In: Mirchandani, P.B., Francis, R.L. (eds.): Discrete Location Theory, pp. 439–478. Wiley-Interscience, New York (1990)
15. Hotelling, H.: Stability in competition. Econ. J. **39**, 41–57 (1929)
16. Hansen, P., Labbé, M.: Algorithms for voting and competitive location on a network. Transp. Sci. **22**(4), 278–288 (1988)
17. Hansen, P., Mladenović, N.: Variable neighborhood search: principles and applications. Eur. J. Oper. Res. **130**, 449–467 (2001)
18. Küçükaudin, H., Aras, N., Altinel, I.K.: Competitive facility location problem with attractiveness adjustment of the follower: a bilevel programming model and its solution. Eur. J. Oper. Res. **208**, 206–220 (2011)
19. Kochetov, Y., Ivanenko, D.: Computationally difficult instances for the uncapacitated facility location problem. In: Ibaraki, T., et al. (eds.) Metaheuristics: Progress as Real Solvers, pp. 351–367. Springer, Berlin (2005)
20. Kochetov, Y.A.: Facility location: discrete models and local search methods. In: Chvatal, V. (ed.) Combinatorial Optimization, Methods and Applications, pp. 97–134. IOS Press, Amsterdam (2011)
21. Kress, D., Pesch, E.: Sequential competitive location on networks. Eur. J. Oper. Res. **217**(3), 483–499 (2012)
22. Megiddo, N., Zemel, E., Hakimi, S.: The maximum coverage location problem. SIAM J. Algebraic Discrete Methods **4**, 253–261 (1983)
23. Noltemeier, H., Spoerhase, J., Wirth, H.: Multiple voting location and single voting location on trees. Eur. J. Oper. Res. **181**, 654–667 (2007)
24. Plastria, F., Carrizosa, E.: Optimal location and design of a competitive facility. Math. Program. Ser. A. **100**, 247–265 (2004)
25. Roboredo, M.C., Pessoa, A.A.: A branch-and-cut algorithm for the discrete $(r|p)$-centroid problem. Eur. J. Oper. Res. **224**(1), 101–109 (2013)
26. Resende, M.G.C., Werneck, R.F.: A hybrid multistart heuristic for the uncapacitated facility location problem. Eur. J. Oper. Res. **174**(1), 54–68 (2006)

27. Serra, D., ReVelle, C.: Market capture by two competitors: the pre-emptive capture problem. J. Reg. Sci. **34**(4), 549–561 (1994)
28. Spoerhase, J., Wirth, H.: (r, p)-centroid problems on paths and trees. J. Theor. Comput. Sci. Arch. **410**(47–49), 5128–5137 (2009)
29. Saidani, N., Chu, F., Chen, H.: Competitive facility location and design with reactions of competitors already in the market. Eur. J. Oper. Res. **219**, 9–17 (2012)
30. Talbi, E.-G.: Metaheuristics: From Design to Implementation. Wiley, Chichester (2009)