ISSN 1990-4789, Journal of Applied and Industrial Mathematics, 2015, Vol. 9, No. 3, pp. 392–401. © Pleiades Publishing, Ltd., 2015.
Original Russian Text © Yu.A. Kochetov, A.A. Panin, A.V. Plyasunov, 2015, published in Diskretnyi Analiz i Issledovanie Operatsii, 2015, Vol. 22, No. 3, pp. 36–51.

Comparison of Metaheuristics for the Bilevel Facility Location and Mill Pricing Problem

Yu. A. Kochetov[1,2*], A. A. Panin[1], and A. V. Plyasunov[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 March 14, 2015; in final form, April 6, 2015

Abstract—Under study is the bilevel nonlinear facility location and mill pricing problem. It is shown that the problem belongs to the class $Poly$-APX. We present the two hybrid algorithms that are based on local search: variable neighborhood descent (VND-metaheuristics) and genetic local search. These algorithms are compared with available algorithms and $CPLEX$ software and show their competitiveness. Computational experiments are conducted on the instances from the benchmark library "Discrete Location Problems." The results show high efficiency of the developed methods and possibility of solving the problems of large dimension.

DOI: 10.1134/S1990478915030102

Keywords: *bilevel optimization, facility location, pricing, local improvement with variable neighborhoods, genetic local search method, bilevel metaheuristics*

INTRODUCTION

Facility location and/or pricing constitutes a broad spectrum of mathematical models, methods, and applications in operation research [6, 13, 17, 18, 20, 21, 30]. In most models the aspects of location and allocation are considered without pricing. The processes of facility location and pricing are usually studied separately and independently of each other because they belong to different planning horizons. The location represent a long-term decision, while the pricing represent a short-term decision [19]. It turns out that at first we usually select location and only later prices [26]. However, separation of the location and pricing decisions may be inacceptable, for example, in the cases where the locations are chosen in dependence on the client's demand which, in turn, depends on prices [22]. Moreover, the separation of location and pricing is not reasonable when there is no need to know the exact prices, and only an interval of prices acceptable for the market is desired [8]. Thus, up-to-date approaches in the location and pricing problems are usually based on simultaneous analysis of both aspects in the same model [8–10, 14, 28, 29]. However, in order to estimate the quality of decisions we must be able to analyze the market reaction on the suggested location of facilities as well as the prices. To this aim it is convenient to use the bilevel model of the whole process [5, 15].

Based on [24], many mathematical models for location and pricing decisions under competition have been studied (e.g., see [19, 22, 26]). In this paper, we assume that some pre-existing facilities are fixed to their current prices and location [16]. They cannot adjust easily. A new company tries to open its own facilities and charge prices in order to maximize total revenue. Each client knows the transportation cost of servicing from each facility and has his own budget (threshold defined by the pre-existing facilities). The client selects a facility with minimal total payment (price and transportation cost). He buys the product if his payment does not exceed his budget. That is, we consider the Stackelberg type leader–follower game. The company is a leader. The clients form a set of followers. The objective of the leader is to find r facilities and assign the prices for them in order to maximize total revenue. We also assume

*E-mail: jkochet@math.nsc.ru
**E-mail: arteam1897@gmail.com
***E-mail: apljas@math.nsc.ru

that the client demand is concentrated at a finite set of discrete points. If there are several facilities with minimal total payment then the client selects the facility with minimal transportation cost.

There are three spatial pricing strategies identified by [22]: mill pricing when each facility may charge a different price; uniform pricing when facilities charge identical price; and discriminatory pricing when each client may be charged a different price. In this paper, we consider the mill pricing strategy, but the transportation costs may be different for any client and any facility. Similar problems with uniform and discriminatory strategies were considered in [19, 22, 26].

Thus, we consider the following facility location and pricing problem: A company wishes to open some facilities and assigns prices for the product of each opened facility. Each client knows the transportation cost of servicing from each facility and has his own budget (more precisely, this budget is a threshold defined by the leader facilities). The client selects a facility with minimal total payment: price and transportation cost. He buys the product if his payment does not exceed his budget. The objective is to locate r facilities for the company and assign the prices for each opened facility in order to maximize overall revenue.

In Section 1, we present a bilevel nonlinear programming model. In Section 2, we describe our hybrid heuristics for solving this problem. Since the pricing subproblem for fixed location is NP-hard in the strong sense, we developed a bilevel approach: firstly, local search for the facility locations under fixed price vector and, secondly, local search for price vector under fixed location. For the pricing subproblem, we use Genetic and VND metaheuristics. For the location subproblem, we apply a local search method. In Section 3, we discuss the computational results obtained for the data from the library of test problems "Discrete location Problems." The proposed algorithms are compared with some available approximation algorithms and exact methods of the CPLEX software.

1. MATHEMATICAL MODEL

Let us introduce the notations: $I = \{1, ..., n\}$ is the set of potential facility locations, $J = \{1, ..., m\}$ is the set of clients, r is the number of facilities to open, $b_j \geq 0$ is the budget of client j, and $c_{ij} \geq 0$ is the transportation cost for each pair of client j and facility i.

Now we define the decision variables: $p_i \geq 0$ is the price of facility i, $y_i = 1$ if facility i is opened and $y_i = 0$ otherwise, $x_{ij} = 1$ if client j is serviced from facility i, and $x_{ij} = 0$ otherwise.

With these variables we can present the facility location and mill pricing problem (*Problem* FLMP) as following:

$$\max_{y,p,x} \sum_{i \in I} p_i \sum_{j \in J} x_{ij},$$

$$\sum_{i \in I} y_i = r, \qquad x \in F(y,p), \qquad p_i \geq 0, \qquad y_i \in \{0,1\}, \qquad i \in I, \qquad j \in J,$$

where $F(y,p)$ is the set of optimal solutions of the inner problem:

$$\max_{x} \sum_{i \in I} \sum_{j \in J} (b_j - c_{ij} - p_j) x_{ij},$$

$$\sum_{i \in I} x_{ij} \leq 1, \qquad j \in J,$$

$$x_{ij} \leq y_i, \qquad i \in I, \quad j \in J,$$

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

The objective function of the bilevel problem defines the total revenue of the company, and the company constraint guarantees that exactly r facilities will be opened. The objective function of the inner problem describes the strategy of each client: to minimize the total payment according to his budget. Constraints of the inner problem guarantee that each client can be served by at most one opened facility.

Let us define the optimal solution of this bilevel problem as a feasible solution with the maximal value of total revenue. However, when the inner problem has several optimal solutions which are equivalent for the clients, such definition might cause troubles. In this case, the company may lose some income. This happens if the clients choose behavior that is optimal for the inner problem (i.e., each client tries to save as much as possible), but at least one of them choose the facility with the price less than the company expected. Thus, this choice is not optimal for the company.

In the case of several optimal solutions for the inner problem, we suppose that each client chooses the facility (among all open) which is nearest to him. Conceptually, it means that the clients select the serving facilities with the maximal income for the company. In other words, we consider a cooperative statement of the problem. An analogues agrement was used in [6, 7]. Note that in the problem discussed in [3] such a situation does not appear since, by a specific choice of the transport expenses, there is no difference between the cooperative and noncooperative statements of the problem.

By the above assumptions, we can write this bilevel problem as the quadratic programming problem with mixed variables:

$$\sum_{i\in I} p_i \sum_{j\in J} x_{ij} \;\rightarrow\; \max_{p,x,y},$$

$$\sum_{i\in I}(b_j - c_{ij} - p_i)x_{ij} \geq 0, \qquad j \in J,$$

$$\sum_{i\in I}(c_{ij} + p_i)x_{ij} \leq (c_{kj} + p_k)y_k, \qquad k \in I, \qquad j \in J,$$

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

$$\sum_{i\in I} x_{ij} \leq 1, \qquad j \in J,$$

$$x_{ij} \leq y_i, \qquad i \in I, \qquad j \in J,$$

$$p_i \geq 0, \qquad x_{ij}, y_i \in \{0,1\}, \qquad i \in I, \qquad j \in J.$$

The objective function defines the total revenue of the company. The first group of constraints ensure that the clients stay within their budgets. The fulfillment of the second group of constraints leads to minimization of the total expenses of each client on purchase and transportation. The third constraint guarantees that exactly r facilities will be opened. The fourth group of constraints means that each client can be served by at most one facility. The last group of constraints imply that clients can be served only at the opened facilities. We will keep the same notation FLMP for this reformulation of the problem.

Let us introduce some more conventions and notation:

Assume henceforth that all initial data b_j and c_{ij} are rational numbers, OPT(FLMP) is the optimal value of the objective function of Problem FLMP, and $g(y, p, x)$ is the value of the objective function of Problem FLMP at a feasible solution (y, p, x).

Let (y, p) be a vector satisfying the constraints of the leader problem, and let $f(y, p)$ denote the optimal objective function value of the problem for given y and p. Note that the value of $f(y, p)$ is computable in polynomial time [6].

2. LOCAL SEARCH HEURISTICS

It is known [6, 16] that Problem FLMP is strongly NP-hard even for the given facility location. We show that it is possible to obtain more precise hardness properties of this problem. To this end we consider the corresponding decision version of the problem known as Problem $\mathrm{OPT}_{\mathrm{FLMP}}$. This problem is constructed as follows: Add an integer parameter k to the instance of Problem FLMP and decide wether k is an optimal value of the goal function on a set of feasible solutions of Problem FLMP.

Also, we make a correspondence between Problem FLMP and the standard decision version D(FLMP) of this problem. The latter is to decide whether there exists a feasible solution with the value of the goal function greater than or equal to k, where k is an integer parameter.

Further we use the three basic classes P, NP, and co-NP of the decision problems which represent the first level of the polynomial hierarchy of complexity classes. Also we use Δ_2^p class as an element of the second level of the above hierarchy [1, 11]. This class is defined as follows: The decision problem D belongs to Δ_2^p if there exists a deterministic oracle Turing machine which verifies problem D in polynomial time by some NP language as an oracle. The class Δ_2^p is frequently denoted by P^{NP}. We call D a *nontrivial* Δ_2^p*-problem* [27] if

$$D \in \Delta_2^p, \qquad D \notin \Sigma_1^p \cup \Pi_1^p.$$

Theorem 1. *If* $\mathrm{NP} \neq \mathrm{co} - \mathrm{NP}$ *then* $\mathrm{OPT}_{\mathrm{FLMP}}$ *is a nontrivial* Δ_2^p*-problem.*

Proof. Let us show that the following are satisfied: $\mathrm{OPT}_{\mathrm{FLMP}} \in \Delta_2^p$ and if $\mathrm{NP} \neq \mathrm{co} - \mathrm{NP}$ then

$$\mathrm{OPT}_{\mathrm{FLMP}} \notin \mathrm{NP} \cup \mathrm{co} - \mathrm{NP}.$$

Is is easy to see that the decision problem $D(\mathrm{FLMP})$ is in NP class. Modifying the reduction of [6] and [16], it is possible to show the strong NP-completeness of $D(\mathrm{FLMP})$. Owing to the structure of the goal function of Problem FLMP, the optimal value of the latter is bounded by a polynomial of the input length. Applying the binary search and Problem $D(\mathrm{FLMP})$ as an oracle, we find the solution of $\mathrm{OPT}_{\mathrm{FLMP}}$ in polynomial time. By the definition of Δ_2^p-class, we see that $\mathrm{OPT}_{\mathrm{FLMP}} \in \Delta_2^p$.

Since Problem $D(\mathrm{FLMP})$ is NP-complete in the strong sense and there is a trivial polynomial reduction to $\mathrm{OPT}_{\mathrm{FLMP}}$, from [27] it follows that

$$\mathrm{OPT}_{\mathrm{FLMP}} \notin \mathrm{NP} \cup \mathrm{co} - \mathrm{NP}.$$

Thus, if $\mathrm{NP} \neq \mathrm{co} - \mathrm{NP}$ then $\mathrm{OPT}_{\mathrm{FLMP}}$ is a nontrivial Δ_2^p-problem.

The proof of Theorem 1 is complete. □

Theorem 1 implies that if $\mathrm{NP} \neq \mathrm{co} - \mathrm{NP}$ finding an optimal solution of Problem FLMP becomes harder with the growth of the size of an instance. Note that $NP \neq co - NP$ hypothesis is stronger than usual $P \neq NP$ [1, 11]. So it is reasonable to focus the efforts on finding some "good" feasible solution. Usually, in that case the problem is considered from the point of complexity of constructing an algorithm with approximation ratio; i.e., to find the position of the optimization problem in the approximation hierarchy [12]

$$PO \subseteq \mathrm{FPTAS} \subseteq \mathrm{PTAS} \subseteq \mathrm{APX} \subseteq \mathrm{Log} - \mathrm{APX} \subseteq \mathrm{Poly} - \mathrm{APX} \subseteq \mathrm{Exp} - \mathrm{APX} \subseteq \mathrm{NPO}.$$

Further we use the class Poly-APX. It is a class of optimization problems such that there exist a polynomial algorithm with approximation ratio which is polynomially bounded with respect to the length of the input. All notions and definitions can be found in [12].

With [6, 7], we can show the following

Theorem 2. *Problem* FLMP *belongs to* Poly-APX.

Proof. Denote by FLMP^1 some Problem FLMP with at most one opened facility ($r = 1$). It is shown in [7] that $rmFLMP^1$ is polynomially solvable. We use an optimal solution to this problem in order to build a feasible solution of the original problem. Let (y^1, p^1, x^1) be an optimal solution of Problem FLMP^1 and let $y_{i_1}^1$ be the unique unit component of the vector y^1. Choose $r - 1$ different indices $i_2, \dots, i_r$ from the set I such that $i_1 \notin \{i_2, \dots, i_r\}$. Put

$$y_i^{i_1,r} = \begin{cases} 1, & \text{if } \ i \in \{i_1, i_2, \dots, i_r\}, \\ 0, & \text{otherwise,} \end{cases} \qquad x_{ij}^{i_1,r} = \begin{cases} x_{ij}^1, & \text{if } \ i = i_1, \\ 0, & \text{otherwise,} \end{cases}$$

$$p_i^{i_1,r} = \begin{cases} p_i^1, & \text{if } \ i = i_1, \\ 1 + \max\limits_{j \in J} b_j, & \text{otherwise.} \end{cases}$$

Obviously, $(y^{i_1,r}, p^{i_1,r}, x^{i_1,r})$ is a feasible solution to Problem FLMP and

$$g(y^{i_1,r}, p^{i_1,r}, x^{i_1,r}) = g(y^1, p^1, x^1).$$

In order to verify that

$$\mathrm{OPT(FLMP)} \le rg(y^{i_1,r}, p^{i_1,r}, x^{i_1,r}).$$

take an arbitrary feasible solution $(\widetilde{y}, \widetilde{p}, \widetilde{x})$ to Problem FLMP. Now put

$$i^* = \arg \max_{i \in I \mid \widetilde{y}_i = 1} \Big\{ \sum_j \widetilde{p}_i \widetilde{x}_{ij} \Big\},$$

$$y_i^{i^*} = \begin{cases} 1, & \text{if } i = i^*, \\ 0 & \text{otherwise,} \end{cases} \qquad x_{ij}^{i^*} = \begin{cases} \widetilde{x}_{ij}, & \text{if } i = i^*, \\ 0 & \text{otherwise.} \end{cases}$$

Then $(y^{i^*}, \tilde{p}, x^{i^*})$ is the feasible solution to Problem FLMP^1. Thus we have

$$g(\tilde{y}, \tilde{p}, \tilde{x}) = \sum_{i \in I} \tilde{p}_i \sum_{j \in J} \tilde{x}_{ij} \le r \sum_{i \in I} \tilde{p}_i \sum_{j \in J} x_{ij}^{i^*} \le r \cdot g(y^{i_1,r}, p^{i_1,r}, x^{i_1,r})$$

and hence

$$\mathrm{OPT(FLMP)} \le r \cdot g(y^{i_1,r}, p^{i_1,r}, x^{i_1,r}).$$

This completes the proof of Theorem 2. □

In other words, we can find an approximate solution in polynomial time with deviation from the optimum at most $O(l(z))$, where $l(z)$ is a polynomial in the length z of Problem FLMP. Unfortunately, the deviation can be large. Therefore, we developed the two hybrid algorithms based on local search method: Variable Neighborhood Descent (VND) and Genetic Local Search (GLS) [13, 22, 25].

For solving Problem FLMP, in [16] was developed the bilevel heuristics:

- local search for the facility location with the decision variables y_i and fixed price vector p_i,
- for the case of given p facilities, local search for the pricing problem with the decision variables p_i.

In this framework, the pricing problem has small dimension, and we can quickly evaluate the total revenue for a given location. The VNS metaheuristics [23] is applied to this end. We used the neighborhoods N_k, $k = 1, \dots, K$, where the prices p_i of at most k open facilities are changed.

To obtain a solution of the facility location problem, we applied the local search again but with another decision variables. The SA and VNS metaheuristics are used at this stage. For a given vector with components y_i, the neighboring solutions are generated by the k-swap and *Lin–Kernighan neighborhoods* [25]. In the k-swap neighborhood, we move at most k facilities to new locations or, in other words, we open some $k' \leqslant k$ new facilities and close exactly k' previously opened facilities.

Put $S = \{i \mid y_i = 1\}$ and let χ_S stand for the characteristic function of S, so $y_i = \chi_S(i)$, $i = 1, \dots, m$. Further, for the sake of simplicity, we identify a Boolean vector and the appropriate characteristic function.

A Lin–Kernighan neighborhood of y_i is constructed by the iterative procedure:

Step 1. Take a pair of the elements $i_1 \in S$ and $i_2 \in \overline{S}$ such that $f(\chi_{S \cup \{i_2\} \setminus \{i_1\}}, p)$ is maximal among all these pairs. If we have several pairs then take one of them arbitrarily.

Step 2. Choose an element i_1 of S and add i_2.

Step 3. Repeat Steps 1 and 2 k times. Note that, at each iteration, we are allowed to use as the indices $i_1 \in S$ and $i_2 \in \overline{S}$ only those that did not appear at all previous iteration steps.

This procedure defines k neighbors of an available location y_i. The parameter k is chosen from the interval $[\min(r, m-r), \max(r, m-r)]$. The local search in this neighborhood is used in the VNS metaheuristics when we did not manage to improve the record solution in several iterations.

Surely, the finding of the best neighboring solution with respect of every of these neighborhoods is a time consuming procedure for k large. Thus, we used k small: $k \le 3$ for the VNS metaheuristics and $k = 1$ for the SA heuristics. All needed details are available in [16].

2.1. Neighborhood Structures

Let us describe the neighborhood structures that are used in hybrid algorithms.

Let $Flip(p \mid y)$ denote the neighborhood of the price vector p for a given location y. This neighborhood consists of all vectors that differ from p exactly in one component. If $\overline{p} \in Flip(p \mid y)$ and for some k we have $\overline{p}_i = p_i$ for all $i \neq k$, then to calculate the value of the objective function and find the component $\overline{p}_k$ we solve Problem FLMP with a fixed location y, prices $\overline{p}_i$ for $i \neq k$, and the only variable p_k. This problem is solvable in polynomial time [7].

The neighborhood Swap$(y \mid p)$ is defined for a given price vector p. It contains all vectors $\overline{y}$ such that $\sum \overline{y}_i = r$, with the Hamming distance from $\overline{y}$ to y equal to 2. Consider such a vector $\overline{y}$. Then, for some i_0 and i_1 such that $y_{i_0} = 0$ and $y_{i_1} = 1$, we have

$$\overline{y}_i = y_i, \quad i \neq i_0, i_1, \qquad \overline{y}_{i_0} = 1, \qquad \overline{y}_{i_1} = 0.$$

Since some facility is closed and a new facility appeared, it is necessary to recalculate the prices for the components with indices i_0 and i_1. Let $\overline{p}_i = p_i$ for $i \neq i_0, i_1$, and let $\overline{p}_{i_1} = 0$; then the value of the component $\overline{p}_{i_0}$ is obtained by solving Problem FLMP with a given facility location $\overline{y}$, prices $\overline{p}_i$ for all $i \neq i_0$, and one variable p_{i_0}.

Let $LS^{\mathrm{pr}}(p \mid y)$ denote the local search algorithm with neighborhood Flip$(p \mid y)$, while $LS^{\mathrm{loc}}(y \mid p)$, the local search algorithm with neighborhood Swap$(y \mid p)$.

2.2. The Metaheuristics Gen *and* VND

Using algorithm $LS^{\mathrm{pr}}(p \mid y)$, we describe the VND and Genetic metaheuristics. The evolution strategies, evolution programming, and the genetic algorithms are famous approaches in combinatorial optimization. They have sustained many modifications according to the variety of problems.

In this paper, we use the genetic local search algorithm (GLS) that is interesting both from the theoretical and practical points of view. It is a variant of a Memetic algorithm in which we apply different greedy strategies and crossover operators [5]. Algorithm GLS is an iterative method. At each iteration, we have a set of local optima within the prescribed neighborhoods. This set constitutes a *population*. The population evolves during a succession of iterations until some termination criterion is satisfied. The framework of our metaheuristic can be presented as follows:

Algorithm Gen

Input: y is the facility location, and I_{max} is the total number of iterations.

Output: The best found price vector p.

Step 0: $i \leftarrow 0$. Randomly generate a starting population of price vectors. For each vector p from population, apply Algorithm $LS^{\mathrm{pr}}(p \mid y)$.

Step 1. Randomly select two elements from population as parents. Using uniform crossover, create an offspring solution p' for the parents. Apply local search for the new price vector p' and find local optimum $p^* := LS^{\mathrm{pr}}(p' \mid y)$.

Step 2. Update population $i \leftarrow i + 1$. If $i \leq I_{\max}$ then go to Step 1, else Stop.

The algorithm Variable Neighborhood Descend (VND) performs several descents with different neighborhoods until a local optimum is reached for all considered neighborhoods. Let $N_1, N_2, \ldots, N_K$ denote a set of K neighborhood structures. Starting with the first structure N_1, VND algorithm performs local search until no further improvements are possible. From this local optimum, it continues local search with neighborhood structure N_2. If an improved solution could be found with this structure then VND returns to N_1 again; otherwise, it continues with N_3; and so forth.

Thus, if the last structure N_K has been applied and no further improvements are possible then the solution represents a local optimum with respect to all neighborhood structures; and VND terminates. The neighborhood structures are explored in order typically from the smallest and fastest to evaluate, to the slower and bigger one.

We present our implementation of this heuristics:

Algorithm VND

Input: y is the facility location, $I_{\max}$ is the maximal number of iterations, d is the number of improvements in Procedure Improve, and k is the parameter of neighborhood $k\text{-}Flip(p \mid y)$.

Output: The best found price vector p.

Step 0. $i \leftarrow 0$. Generate randomly a starting price vector p.

Step 1. Apply local search for the price vector p and find a local optimum $p^* \leftarrow LS^{\mathrm{pr}}(p \mid y)$.

Step 2. $i \leftarrow i+1$. If $i > I_{max}$ then Stop, else $p \leftarrow Improve(k, p^* \mid y)$. If $f(y,p) > f(y,p^*)$ then go to Step 1, else Stop.

In the $Improve$ procedure we explore only the following neighborhood structures: 2-Flip$(p \mid y)$ and 3-Flip$(p \mid y)$. Searching for the best neighboring solution is a time consuming procedure for k large, so we perform each move within the neighborhood structure k-Flip$(p \mid y)$ as k consecutive moves within Flip$(p \mid y)$ neighborhood.

Each of these neighborhood structures are explored by d-improvement search strategy; i.e., if we find d improvements in the current neighborhood then the search is terminated [23].

2.3. The Hybrid Algorithms LS+GEN *and* LS+VND

Given the current solution (y, p), at each iteration of the algorithms we first apply the local search with Flip$(p \mid y)$ neighborhood and then apply the local search with Swap$(y \mid \overline{p})$ neighborhood. The price vector $\overline{p}$ at the second step is obtained from the first step. As the stopping criterion we use some given number of iterations. The result of the algorithm is the best found solutions.

The first stage procedure is realized on the Genetic framework in the first algorithm. In the second algorithm, this procedure is based on the VND approach. Using the algorithms of Section 2.2 and local search algorithm $LS^{\mathrm{loc}}(y \mid p)$, we now describe the hybrid VND and Genetic heuristics. Let Y denote the set of all vectors $\overline{y}$, where $\sum \overline{y}_i = r$, and let L denote the tabu list.

Algorithm LS+Gen

Input: $I_{\max}$ is the maximum total number of iterations.

Output: The best found solution.

Step 0. Put $i \leftarrow 0$ and $L \leftarrow \varnothing$. Generate randomly a starting location y.

Step 1. If $i > I_{\max}$ then Stop, else $L := L \cup \{y\}$, $i \leftarrow i+1$, apply Genetic metaheuristic, and find local optimum p.

Step 2. Find local optimum $y^* := LS^{\mathrm{loc}}(y \mid p)$. Let p^* be the price vector corresponding to the location y^*. If $f(y^*, p^*) \leq f(y,p)$ then select an arbitrary $y \in Y \setminus L$, else $y := y^*$; and go to Step 1.

To obtain Algorithm LS+VND we replace the Genetic algorithm by Algorithm VND.

3. COMPUTATIONAL RESULTS

We tested our hybrid methods on PC Intel Core i7-3612QM, RAM 4Gb. We compared the two developed algorithms LS+Gen and LS+VND with the already known method VNS+VNS [16] and CPLEX software on the instances from the benchmark library "Discrete Location Problems" with dimension $n = 40, 100$, $m = 100$, and $r = 5$ (see Table 1).

Moreover, we compared Algorithms LS+Gen and LS+VND (see Table 2) on these instances but only with $n = 40$, $m = 100$, and $r = 10, 15$. In [16], the heuristics SA+VNS was under study together with the bilevel method VNS+VNS. Details of the method SA+VNS are omitted since this algorithm shows the same results as Algorithm VNS+VNS, but requires significantly more computing time.

In Tables 1 and 2, the first column denotes the code of the instances. The column $n(r)$ represents the quantity of possible places to open a facility and the number of opened facilities. Columns "Revenue," "Time," and "Iteration" denote respectively the best found revenue, the running time, and the iteration at which the best solution was found. The CPLEX software was stopped after 24 hours, and the best found solution was presented as the result for $n = 40$. For higher dimension (for example, $n = 100$) the CPLEX software cannot find any feasible solution.

Table 1. Comparison of Algorithms LS+Gen, LS+VND, and VNS+VNS with CPLEX software

Instance		VNS+VNS		CPLEX	LS+Gen			LS+VND		
	$n(r)$	Revenue	Time	Revenue	Revenue	Iteration	Time	Revenue	Iteration	Time
1	40(5)	2245	45 m	2226	2245	5	36 s	2245	20	4 s
2	40(5)	2259	51 m	2259	2259	14	97 s	2259	72	16 s
3	40(5)	2019	41 m	2019	2019	8	47 s	1984	12	3 s
4	40(5)	1533	42 m	1508	1533	63	347 s	1552	22	5 s
5	40(5)	2386	46 m	2313	2386	11	77 s	2346	34	8 s
6	40(5)	1960	60 m	1949	1956	9	57 s	1987	17	4 s
7	40(5)	2179	60 m	2142	2179	55	415 s	2178	58	14 s
8	40(5)	2139	51 m	2139	2139	30	224 s	2140	31	7 s
9	40(5)	1895	59 m	1877	1904	17	115 s	1900	45	10 s
10	40(5)	2209	37 m	2209	2209	4	32s	2252	4	1 s
11	100(5)	2235	1 h 9 m		2230	7	48 s	2235	24	14 s
12	100(5)	2240	3 h 12 m		2240	295	2015 s	2233	31	18 s
13	100(5)	1923	1 h 19 m		1923	16	107 s	1957	13	8 s
14	100(5)	2133	1 h 48 m		2133	466	3194 s	2118	15	9 s
15	100(5)	2099	1 h 58 m		2099	27	197 s	2153	25	15 s
16	100(5)	2237	1 h 10 m		2237	108	806 s	2182	126	75 s
17	100(5)	1888	1 h 15 m		1893	31	202 s	1921	108	62 s
18	100(5)	1825	2 h 51 m		1825	48	312 s	1871	4	2 s
19	100(5)	1767	1 h 43 m		1767	8	48 s	1767	4	3 s
20	100(5)	2363	1 h 2 m		2368	55	369 s	2368	139	84 s

It follows from Table 1 that Algorithms LS+Gen and LS+VND show similar results as VNS+VNS and CPLEX, but spend less running time. Moreover, Table 2 shows that *LS+VND* method dominates LS+Gen only in running time for r large.

4. CONCLUSION

In this paper, we proposed the two metaheuristics based on local search: variable neighborhood descent and genetic local search, for solving the bilevel facility location and mill pricing problem. Also we discussed experimental comparison of those heuristics with the double VNS heuristic [16], and the branch and bound method (CPLEX).

In future it would be interesting to apply other neighborhoods, for example, the Lin-Kernighan neighborhood, and math-heuristics.

Table 2. Comparison of Algorithm LS+Gen with Algorithm LS+VND

Instance		LS+Gen			LS+VND		
	$n(r)$	Revenue	Iteration	Time	Revenue	Iteration	Time
21	40(10)	2650	43	2157 s	2626	421	635 s
22	40(10)	2617	39	2266 s	2621	182	282 s
23	40(10)	2351	31	1601 s	2327	73	106 s
24	40(10)	1888	55	2305 s	1895	78	112 s
25	40(10)	2827	24	942 s	2793	88	141 s
26	40(10)	2261	53	2902 s	2289	349	538 s
27	40(10)	2481	15	498 s	2480	528	855 s
28	40(10)	2447	56	1842 s	2443	192	311 s
29	40(10)	2245	47	2629 s	2246	758	1250 s
30	40(10)	2497	58	3054 s	2596	112	180 s
31	40(15)	2804	26	3162 s	2796	794	3413 s
32	40(15)	2758	26	3635 s	2797	346	1616 s
33	40(15)	2515	13	1908 s	2477	146	620 s
34	40(15)	1952	29	3161 s	1983	453	2046 s
35	40(15)	2908	11	1881 s	2870	9	45 s
36	40(15)	2286	23	3129 s	2436	25	114 s
37	40(15)	2529	16	2144 s	2571	288	1369 s
38	40(15)	2532	8	545 s	2558	302	1510 s
39	40(15)	2355	6	630 s	2372	659	3091 s
40	40(15)	2424	24	3431 s	2698	52	261 s

ACKNOWLEDGMENTS

The authors were supported by the Russian Foundation for Basic Research (project no. 13-07-00016) and the Institute of Informatics and Computational Technologies of the Ministry of Education and Science of the Republic Kazakhstan (project no. 0115PK00546).

REFERENCES

1. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979; Mir, Moscow, 1982).
2. I. A. Davydov, Yu. A. Kochetov, N. Mladenović, and D. Urošević, "Fast Metaheuristics for the Discrete $(r|p)$-Centroid Problem," Avtomat. i Telemekh. No. 4, 106–119 (2014) [Automat. Remote Control **75** (4), 677–687 (2014)].
3. V. T. Dement'ev and Yu. V. Shamardin, "The Problem of Price Selection for Production under the Condition of Obligatory Satisfaction of Demand," Diskret. Anal. Issled. Oper. Ser. 2, **9** (2), 31–40 (2002).

4. Yu. A. Kochetov and A. V. Plyasunov, "A Polynomially Solvable Class of Two-Level Linear Programming Problems," Diskret. Anal. Issled. Oper. Ser. 2, **1** (2), 23–33 (1997).
5. Yu. A. Kochetov and A. V. Plyasunov, "Genetic Local Search the Graph Partitioning Problem under Cardinality Constraints," Zh. Vychisl. Mat. Mat. Fiz. **52** (1), 164–176 (2012.) [Comput. Math. Math. Phys.**52** (1), 157–167 (2012)].
6. A. A. Panin and A. V. Plyasunov, "On Complexity of the Bilevel Location and Pricing Problems," Diskret. Anal. Issled. Oper. **21** (5), 54–66 (2014) [J. Appl. Indust. Math. **8** (4), 574–581 (2014)].
7. A. V. Plyasunov and A. A. Panin, "The Pricing Problem. Part I: Exact and Approximate Algorithms," Diskret. Anal. Issled. Oper. **19** (5), 83–100 (2012) [J. Appl. Indust. Math. **7** (2), 241–251 (2013)].
8. R. Aboolian, O. Berman, and D. Krass, "Competitive Facility Location Model with Concave Demand," European J. Oper. Res. **181** (2), 598–619 (2007).
9. R. Aboolian, O. Berman, and D. Krass, "Optimizing Pricing and Location Decisions for Competitive Service Facilities Charging Uniform Price," J. Oper. Res. Soc. **59** (11), 1506–1519 (2008).
10. N. Adler and K. Smilowitz, "Hub-and-spoke Network Alliances and Mergers: Price-Location Competition in the Airline Industry," Transp. Res. Part B, **41** (4), 394–409 (2007).
11. *Algorithms and Theory of Computation Handbook*, Ed. by M. Attallah (CRC Press LLC, Boca Raton, 1998).
12. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties* (Springer, Berlin, 1999).
13. M. Bouhtou, A. Grigoriev, S. van Hoesel, A. F. van Der Kraaij, F. C. R. Spieksma, and M. Uetz, "Pricing Bridges to Cross a River," Naval. Res. Logist. **54** (4), 411–420 (2007).
14. A. Dasci and G. Laporte, "Location and Pricing Decisions of a Multistore Monopoly in a Spatial Market," J. Region Sci. **44** (3), 489–515 (2004).
15. S. Dempe, *Foundations of Bilevel Programming* (Kluwer Acad. Publ., Dordrecht, 2002).
16. Z. Diakova and Yu. Kochetov, "A Double VNS Heuristic for the Facility Location and Pricing Problem," Electron. Notes Discrete Math. **39** (4), 29–34 (2012).
17. *Facility Location: A Survey of Applications and Methods*, Ed. by Z. Drezner (Springer, New York, 1995).
18. T. Drezner and H. A. Eiselt, "Consumers in Competitive Location Models," in *Facility Location: Applications and Theory*, Ed. by Z. Drezner and H. W. Hamacher (Springer, Berlin, 2004), pp. 151–178.
19. H. A. Eiselt, G. Laporte, and J.-F. Thisse, "Competitive Location Models: A Framework and Bibliography," Transp. Sci. **27** (1), 44–54 (1993).
20. H. A. Eiselt and V. Marianov, *Foundations of Location Analysis* (Springer, New York, 2011) [Internat. Ser. Oper. Res. Management Sci., Vol. 155].
21. A. Grigoriev, J. van Loon, M. Sviridenko, M. Uetz, and T. Vredeveld, "Optimal Bundle Pricing with Monotonicity Constraint," Oper. Res. Lett. **36** (5), 609–614 (2008).
22. P. Hanjoul, P. Hansen, D. Peeters, and J.-F. Thisse, "Uncapacitated Plant Location under Alternative Spatial Price Policies," Management Sci. **36** (1), 41–57 (1990).
23. P. Hansen and N. Mladenović, "Variable Neighborhood Search," European J. Oper. Res. **130** (3), 449–467 (2001).
24. H. Hotelling, "Stability in Competition," Econom. J. **39** (153), 41–57 (1929).
25. Yu. A. Kochetov, E. V. Alekseeva, T. V. Levanova, and M. A. Loresh, "Large Neighborhood Local Search for the p-Median Problem," Yugosl. J. Oper. Res. **15** (1), 53–63 (2005).
26. Ph. J. Lederes and J.-F. Thisse, "Competitive Location on Network under Delivered Pricing," Oper. Res. Lett. **9** (3), 147–154 (1990).
27. E. W. Leggett Jr. and D. J. Moore, "Optimization Problems and the Polynomial Hierarchy," Theor. Comput. Sci. **15** (3), 279–289 (1981).
28. A. Lüer-Villagra and V. Marianov, "A Competitive Hub Location and Pricing Problem," European J. Oper. Res. **231** (3), 734–744 (2013).
29. D. Serra and Ch. ReVelle, "Competitive Locations and Pricing on Networks," Geograph. Anal. **31** (2), 109–129 (1999).
30. X. Vives, *Oligopoly Pricing: Old Ideas and New Tools* (MIT Press, Cambridge, MA, 2001).