

# Comparison of Two Heuristic Algorithms for a Location and Design Problem

Alexander Gnusarev

**Abstract** The article is devoted to the decision-making methods for the following location and design problem. The Company is planning to locate its facilities and gain profit at an already existing market. Therefore it has to take into consideration such circumstances as already placed competing facilities; the presence of several projects for each facility opening; the share of the served demand is flexible and depends on the facility location. The aim of the Company is to determine its new facilities locations and options in order to attract the biggest share of the demand. Modeling flexible demand requires exploiting nonlinear functions which complicates the development of the solution methods. A Variable Neighborhoods Search algorithm and a Greedy Weight Heuristic are proposed. The experimental analysis of the algorithms for the instances of special structure has been carried out. New best known solutions have been found, thus denoting the perspective of the further research in this direction.

## 1 Introduction

Among the numerous applications there is a great interest in location problems [1]. In many cases, it is necessary to place the facilities in several locations and to assign a number of customers to be served to them so, that the total expenses were the least. In the well-known classical models, such as Simple Plant Location Problem,  $p$ -median Problem, and others [1]; the decision is made by one person, not taking into consideration the customers' opinions and other circumstances.

However, the customers very often have their own preferences, and several rival companies struggle for serving them. Such situations are described in competitive market location problems which most accurately characterize the actual business environment. The paper deals with a discrete type of such a task in which the facilities can be opened at a finite set of possible locations and the clients are in a discrete set of points. The rival companies have already occupied some of them and are unable to change their decision. The new Company is aware of the existing competition. The

---

A. Gnusarev (✉)

Sobolev Institute of Mathematics SB RAS, Omsk Branch, Pevtsova str. 13, Omsk 644043, Russia  
e-mail: alexander.gnussarev@gmail.com

© Springer International Publishing AG 2017

V.A. Kalyagin et al. (eds.), *Models, Algorithms, and Technologies for Network Analysis*, Springer Proceedings in Mathematics & Statistics 197,  
DOI 10.1007/978-3-319-56829-4\_4

customers choose the facilities taking into account their attractiveness. This task is related to the Static Probabilistic Competitive Facility Location Problems [2] one of them was developed by Berman and Krass [3].

We take into consideration a special case of the model in [3] which was proposed in [4] and is called “facility location and design problem”. Aboolian et al. proposed a spatial interaction model for describing demand cannibalization. In [4], the demand is a function of the total utility and the objective function is nonlinear. This location problem is quite complicated both from the theoretical and the practical points of view. Obtaining optimal solutions for large instances of the problem using the exact algorithms, including software packages, can require significant time and computer resources. In [4], the adapted weighted greedy heuristic algorithm is proposed for the solution of the discrete competitive facility location and design problem. Therefore, it is interesting to develop modern decision methods for the considered problem.

A lot of attention has been given to the methods of finding approximate solutions recently [5], they include the class of local search algorithms [6]. Variable Neighborhood Search Algorithm (VNS) belongs to this class, it is successfully used for solution of many applied tasks. The basic idea of VNS is to explore a set of predefined neighborhoods successively in order to provide a better solution. The Variable Neighborhood Search Approach [7] for the location and design problem is developed in this paper. A new version of the VNS is proposed, the specific types of neighborhoods are described. The numerical experiments based on the specially generated instances [8] have been carried out, the comparison of Variable Neighborhood Search Algorithm and Greedy Weight Heuristic [4] have been executed. CoinBonmin is used for the analysis of the quality of the obtained solutions [9]. The results show that VNS can find new best known solutions to the large instances of the problem with a small relative error.

The rest of the paper is organized as follows. Section 2 contains the formulation of the problem. Section 3 describes the schemes of the Neighborhood Search Algorithm and Greedy Weight Heuristic. The new version of VNS for the considered problem is proposed in Sect. 3 as well. Section 4 presents the results of the numerical experiments. Section 5 concludes the paper.

## 2 Problem Formulation

The article deals with the situation when a new Company plans to enter the market of existing products and services. It makes a decision to open a supermarket chain, which will differ in size or the set of goods provided from the existing ones. Such differences are called “design options”. Customers select companies based on their attractiveness and distance from their location. The aim of the Company is to interest the greatest number of the customers thus serving the largest share of the demand. This percentage is not fixed for the company and depends on where and which option the new enterprise will be opened on.

The problem has several applications. One of them appeared in Toronto (Canada) and is described by Kraas, Berman, Aboolian. The mathematical model has been formulated in [4]. Let  $R$  be the set of facility designs,  $r \in R$ . There are  $w_i$  customers at the point  $i$  of discrete set  $N = \{1, 2, \dots, n\}$  in the problem. All the customers have the same demands, so each item can be considered as one client with the weight  $w_i$ . Let the distance  $d_{ij}$  between the points  $i$  and  $j$  be measured in Euclidean metric or it equals to the shortest part length in the corresponding graph. Let  $P \subseteq N$  be the set of potential facility locations. It is assumed that  $C \subset P$  is the set of preexisting rival facilities. The Company may open its branches in  $S = P \setminus C$  taking into account the available budget  $B$ , attractiveness  $a_{jr}$ , and the cost of opening  $c_{jr}$  facility  $j \in S$  with design  $r \in R$ .

Such flexible choice of customer is represented in the gravity-type spatial interaction models. These models are known as the brand share models in the marketing literature [10]. According to these models, the utility  $u_{ij}$  of a facility at location  $j \in S$  for a customer at point  $i \in N$  can be written as exponential function. Let  $x_{jr} = 1$ , if the facility  $j$  is opened with the design variant  $r$  and  $x_{jr} = 0$  otherwise,  $j \in S, r \in R$ .

To determine the usefulness  $u_{ij}$  of the facility  $j \in S$  for the customer  $i \in N$  the coefficients  $k_{ijk}$  have been introduced:  $k_{ijk} = a_{jr}(d_{ij} + 1)^{-\beta}$ . They depend on the sensitivity  $\beta$  of the customers to the distance from the facility.

The utility  $u_{ij} = \sum_{r=1}^R k_{ijr} x_{jr}$ . The total utility of the customer  $i \in N$  received from the competitive facilities is

$$U_i(C) = \sum_{j \in C} u_{ij}.$$

The demand function is

$$g(U_i) = 1 - \exp\left(-\lambda_i U_i\right),$$

where  $\lambda_i$  is the flexible demand characteristic in point  $i$ ;  $U_i$  is the total utility for a customer at  $i \in N$  from all open facilities:

$$U_i = \sum_{j \in S} \sum_{r=1}^R k_{ijr} x_{jr} + U_i(C).$$

The total share of the company in the facility  $i \in N$ :

$$MS_i = \frac{U_i(S)}{U_i(S) + U_i(C)} = \frac{\sum_{j \in S} \sum_{r=1}^R k_{ijr} x_{jr}}{\sum_{j \in S} \sum_{r=1}^R k_{ijr} x_{jr} + \sum_{j \in C} u_{ij}}.$$

The mathematical model looks like:

$$\sum_{i \in N} w_i \cdot g(U_i) \cdot MS_i \rightarrow \max \quad (1)$$

$$\sum_{j \in S} \sum_{r=1}^R c_{jr} x_{jr} \leq B, \quad (2)$$

$$\sum_{r=1}^R x_{jr} \leq 1, \quad (3)$$

$$x_{jr} \in \{0, 1\}, \quad r \in R, j \in S. \quad (4)$$

Based on the above notation, the objective function (1) looks as

$$\sum_{i \in N} w_i \left( 1 - \exp \left( - \lambda_i \left( \sum_{j \in S} \sum_{r=1}^R k_{ijr} x_{jr} + U_i(C) \right) \right) \right). \quad (5)$$

$$\cdot \left( \frac{\sum_{j \in S} \sum_{r=1}^R k_{ijr} x_{jr}}{\sum_{j \in S} \sum_{r=1}^R k_{ijr} x_{jr} + \sum_{j \in C} u_{ij}} \right) \rightarrow \max.$$

The objective function (5) reflects the Company's aim to maximize the volume of the customers' demand. Inequality (2) takes into account the available budget. Condition (3) shows that only one variant of the design can be selected for each facility.

### 3 Algorithms

The of well-known software GAMS (CoinBonmin) for the location and design problem can calculate an approximate solution which is not guaranteed to be optimal [9]. Solving such problem requires a significant investment of time and computing resources. In this regard, one of the approaches to its solution is in employment of the approximate heuristic methods. In this paper, the Variable Neighborhoods Search algorithm [7, 11] has been constructed and compared to the Greedy Weight Heuristic one for the considered problem, the scheme of the Variable Neighborhood Search algorithm (VNS) is given below.

The scheme of the VNS algorithm

*Initialization.* Select the set of neighborhood structures  $N_k, k = 1, \dots, k_{max}$ , that will be used in the search; find the initial solution  $x$ ; choose the stopping condition.

Repeat the following steps until the stopping condition is met.

- (1) Set  $k := 1$ .
- (2) Until  $k = k_{max}$ , repeat the following steps.
  - (a) *Shaking*. Generate a point  $x'$  at random from the  $k$ -th neighborhood of  $x$  ( $x' \in N_k(x)$ );
  - (b) *Local search*. Apply a local search method with  $x'$  as the initial solution; denote  $x''$  as the obtained local optimum;
  - (c) *Move or not*. If this local optimum is better than the incumbent, move there  $x := x''$ , and continue the search with  $N_1$ ,  $k := 1$ ; otherwise, set  $k := k + 1$ .

The new types of neighborhoods used for the algorithm are described below. Let the vector  $z = (z_i)$  be such that  $z_i$  corresponds to facility  $i$ :  $z_i = r$  iff  $x_{ir} = 1$ . The feasible initial solution  $z$  is obtained using a special deterministic procedure.

*Neighborhood 1* (N1). Feasible solution  $z'$  is called neighboring for  $z$  if it can be obtained with the following steps:

- (a) choose randomly one of the open facilities  $p$  with the scenario  $z_p$  and close it;
- (b) select the facility  $q$  which is closed and has the highest attractiveness; then open the facility  $q$  with the scenario  $z_p$ .

*Neighborhood 2* (N2). Feasible solution  $z'$  is called neighboring for  $z$  if it can be obtained with the following operations:

- (a) choose randomly one of the open facilities  $p$  with the scenario  $z_p$  and reduce the number of the scenario;
- (b) select randomly the facility  $q$  and increase the number of its scenario.

*Neighborhood 3* (N3). Unlike the Neighborhood 2 at step (b) select randomly the facility  $q$  which is closed; then open the facility  $q$  with the scenario  $z_p$ .

Lin–Kernighan neighborhood was applied as *Neighborhood 4* [12].

In order to describe Greedy Weight Heuristic let us introduce the following terms:  $L$  is a set of facility design pairs  $(j, r)$ , where  $j \in S$  is the location of the facility,  $r \in R$  is the design scenario chosen for that location.  $Z(L)$  is the objective function value associated with the location-design set  $L$ . Let  $\rho_{jr}(L) = Z(L \cup (j, r)) - Z(L)$  be the improvement of the objective function obtained by adding the location-design pair  $(j, r)$  to the location-design set  $L$ . Set  $T$  is the location-design pairs that should be excluded from further consideration. The outline of Greedy Weight Heuristic is as follows:

*Step 1*:  $L^0 = \emptyset, T^0 = \emptyset, t = 1$

*Step 2*: Let  $(j(t), r(t)) = \arg \max_{(j,r) \notin T^{t-1}} \left\{ \frac{\rho_{jr}(L^{t-1})}{c_{jr}} \right\}$

If  $\sum_{(j,r) \in L^{t-1}} c_{jr} + c_{j(t)r(t)} \leq B$  then

set  $L^t = L^{t-1} + \{(j(t), r(t))\}$ ,  $T^t = T^{t-1} + \{(j(t), r)|r \in R\}$

and set  $t = t + 1$ .

Return to *Step 2*.

Else go to *Step 3*.

Endif

Step 3: If  $Z(L^{t-1}) \geq Z(j(t), r(t))$ , then  
 $L^H = L^{t-1}$  is the adapted greedy solution with value  $Z(L^H)$ .  
 Else set  $L^1 = \{(j(t), r(t))\}$ ,  $T^1 = \{(j(t), r) | r \in R\}$ ,  $t = 2$   
 Return to Step 2.  
 Endif  
 Stop

The described algorithms have been programmed on a computer and experimentally investigated. The results are contained in the following section.

### 4 Experimental Study

The validation of the VNS algorithm has been conducted for the following data: the neighborhoods N1, N2, N3 and Lin–Kernighan are used, local descent has been carried out with the help of the neighborhood of Lin–Kernighan with 9 points. Stopping criteria for the VNS has been the complete neighborhood exploration without an improvement of the solution.

There are three options for the enterprises development ( $R = 3$ ): the small one with the cost of opening  $c_{j1} = 1$ ; the medium one with the cost of opening  $c_{j2} = 2$ ; the large one with the cost of opening  $c_{j3} = 3$  for all  $j \in S$ . At each point of the

**Table 1** Best known solutions

Tests	Arbitrary distances			Euclidean distances		
	GAMS	VNS	GWH	GAMS	VNS	GWH
300.3.1	36.15	36.143	23.190	—	35.183	35.183
300.3.2	54.12	57.158	44.283	—	54.446	54.446
300.3.3	74.41	75.513	70.392	—	73.053	73.053
300.3.4	94.47	96.097	86.411	—	91.081	91.081
300.5.1	30.02	30.334	21.419	—	30.899	30.899
300.5.2	51.04	51.051	38.392	—	50.514	50.515
300.5.3	66.99	67.503	54.510	—	69.360	69.360
300.5.4	81.93	86.312	70.427	—	87.756	87.848
300.7.1	36.43	36.427	20.095	—	36.154	36.154
300.7.2	53.69	55.627	39.798	—	57.568	57.568
300.7.3	74.62	74.610	58.637	—	77.670	77.787
300.7.4	92.83	95.266	76.566	—	96.992	97.119
300.9.1	31.83	31.823	20.322	—	32.093	32.093
300.9.2	48.65	51.274	38.876	—	51.503	51.503
300.9.3	67.09	69.000	56.537	—	70.294	70.295
300.9.4	85.87	85.603	73.210	—	88.947	88.947

**Table 2** CPU time (sec)

Tests $ N $	Arbitrary distances						Euclidean distances					
	VNS			GWH			VNS			GWH		
	min	av	max	min	av	max	min	av	max	min	av	max
60	10.94	21.92	46.16	0.06	0.12	0.19	12.29	20.52	39.26	0.10	0.19	0.36
80	21.32	34.16	86.34	0.14	0.23	0.32	23.04	36.49	83.92	0.29	0.46	0.63
100	38.01	60.95	152.16	0.25	0.52	0.99	32.58	48.47	100.74	0.43	0.72	1.14
150	76.11	97.48	141.28	0.54	1.08	2.61	75.41	145.97	451.60	1.51	2.91	4.23
200	225.05	183.43	295.40	1.65	2.66	4.55	109.51	222.46	447.04	3.44	5.65	8.71
300	265.45	438.38	643.23	3.96	9.14	13.75	268.04	614.21	1408.92	10.68	18.24	25.85

demand a business can be opened, i.e.,  $P = N$ . The budget varies from 3 to 9 in the increments of 2 units. For example, having the budget of 9 units the Company can open 3 large enterprises or 9 small or it can combine them. The problem has been considered for random distances ( $d_{ij} \in [0, 30], i \neq j, d_{ij} = d_{ji}$ ) and satisfying the triangle inequality (coordinate  $x \in [0, 100]$ , coordinate  $y \in [0, 150]$ ). The number of obtained locations is 60, 80, 100, 150, 200, 300. It has been assumed that the problem possesses a high sensitivity to the distance ( $\beta = 2$ ) and a nonfixed demand, i.e.,  $\lambda = 1$ .

Experiments were carried out on PC Intel i5–2450M, 2.50 GHz, 4GB memory. The test cases with Euclidean distances are proved to be difficult for the CoinBonmin solver. In particular, the maximum CPU time for the test problems with  $|N| = 60$  was more than 63 h. Therefore, CoinBonmin was given 10 min of CPU time for each example of higher dimensions. Solver CoinBonmin found the best known solutions in 13 cases out of 80 (see Table 1). The VNS and GWH algorithms found new best known solutions for all the test problems with Euclidean distances with dimensions from 80 to 300 in less time. The average time of the VNS algorithm until the stopping criterion was triggered is 181.35 s. (GWH 2.29 s). In all the cases, the best solutions of the VNS were equal to the best solutions of GWH.

Table 2 contains the information about minimal (min), average (av), and maximal (max) CPU time (in seconds) of the proposed algorithms for the test problems. GAMS found records for all the tasks in the test cases with arbitrary distances. The average improvement of the VNS from GAMS was 1.55%, the average deviations of the GWH from GAMS was 12.5%. The results are presented in Figs. 1 and 2. The VNS algorithm improved the record values found by GWH in all test instances with arbitrary distances.

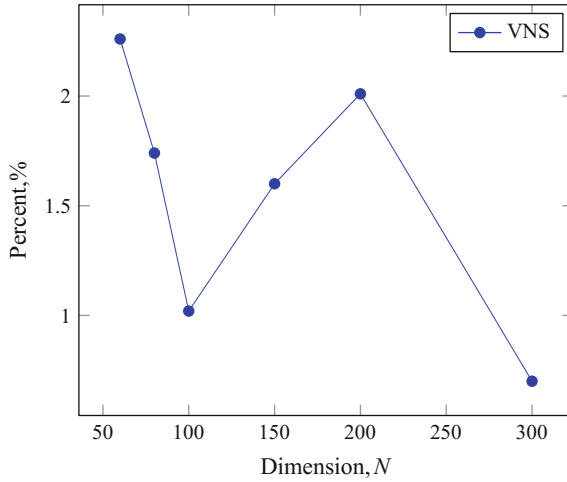


Fig. 1 The average improvement of new best known solutions obtaining by VNS upon CoinBonmin

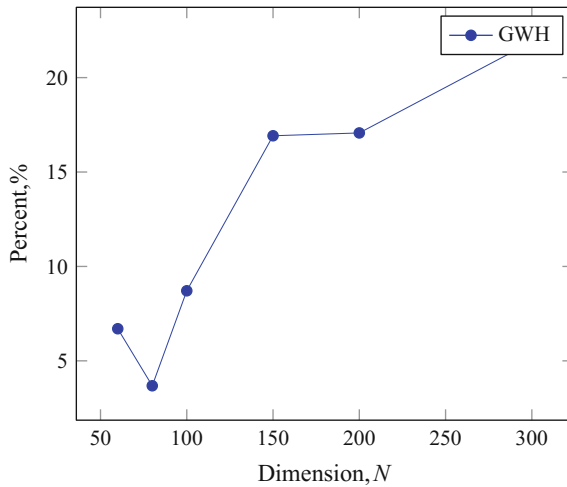


Fig. 2 The average deviation of GWH from the CoinBonmin results (It is information about how many percent on average GWH results worse than CoinBonmin results)

## 5 Conclusion

In this paper, we have created a new version of the Variable Neighborhood Search algorithm and implemented the Greedy Weight Heuristic for the location and design problem. New neighborhoods of a special type have been proposed, the experimental tuning of parameters for both algorithms have been carried out. Two sets of specially structured test examples have been generated. The proposed algorithms have been



able to gain new best known solutions or solutions with small relative error. Therefore, the obtained results have indicated the usefulness of the proposed algorithms for solving the problem.

**Acknowledgements** This research was supported by the Russian Foundation for Basic Research, grant 15-07-01141.

## References

1. Mirchandani, B.P., Francis, R.L. (eds.): *Discrete Location Theory*. Wiley-Interscience, New York (1990)
2. Karakitsion, A.: *Modeling Discrete Competitive Facility Location*. Springer, Heidelberg (2015)
3. Berman, O., Krass, D.: Locating multiple competitive facilities: spatial interaction models with variable expenditures. *Ann. Oper. Res.* **111**, 197–225 (2002)
4. Aboolian, R., Berman, O., Krass, D.: Competitive facility location and design problem. *Eur. J. Oper. Res.* **182**, 40–62 (2007)
5. Gendreau, M., Potvin, J.Y.: *Handbook of Metaheuristics*. Springer, New York (2010)
6. Yannakakis, M.: Computational complexity. In: Aarts, E., Lenstra, J.K. (eds.) *Local Search in Combinatorial Optimization*, pp. 19–55. Wiley, Chichester (1997)
7. Hansen, P., Mladenovic, N.: Variable neighborhood search: principles and applications (invited review). *Eur. J. Oper. Res.* **130**(3), 449–467 (2001)
8. Levanova, T., Gnusarev, A.: Variable neighborhood search approach for the location and design problem. In: Kochetov, Y et al. (eds.) *DOOR-2016*. LNCS, vol. 9869, pp. 570–577. Springer, Heidelberg (2016)
9. Bonami, P., Biegler, L.T., Conn, A.R., Cornuejols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wachter, A.: An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* **5**(2), 186–204 (2008)
10. Huff, D.L.: Defining and estimating a trade area. *J. Mark.* **28**, 34–38 (1964)
11. Hansen, P., Mladenovic, N., Moreno-Perez, J.F.: Variable neighbourhood search: algorithms and applications. *Ann. Oper. Res.* **175**, 367–407 (2010)
12. Kochetov, Y., Alekseeva, E., Levanova, T., Loresh, M.: Large neighborhood local search for the p-median problem. *Yugoslav J. Oper. Res.* **15**(2), 53–64 (2005)