Computers & Operations Research 78 (2017) 537–546

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Solving multifacility Huff location models on networks using metaheuristic and exact approaches

Sanja Grohmann [a], Dragan Urošević [a,*], Emilio Carrizosa [b], Nenad Mladenović [a]

[a] Mathematical Institute SANU, Belgrade, Serbia
[b] University of Seville, Seville, Spain

ARTICLE INFO

Article history:
Received 31 March 2015
Received in revised form
15 January 2016
Accepted 8 March 2016
Available online 15 March 2016

Keywords:
Location
Networks
Competitive location
Metaheuristics
Variable Neighborhood Search

ABSTRACT

In this paper we consider multifacility Huff facility location problem on networks. First, we introduce a slight modification of the existing mixed integer nonlinear mathematical model and confirm its validity by using the solver for nonlinear optimization, KNITRO. Second, since the problem is NP-hard, we develop three methods that are based on three metaheuristic principles: Variable Neighborhood Search, Simulated Annealing, and Multi-Start Local Search. Based on extensive computational experiments on large size instances (up to 800 customers and 100 potential facilities), it appears that VNS based heuristic outperforms the other two proposed methods.

1. Introduction

Location optimization problems on a network in a competitive environment have been extensively studied in operational research. Hakimi [4] formulated the competitive problem under the assumption that consumers deterministically choose the nearest store. In the real world, this assumption is not always acceptable because consumers do not usually choose the nearest store, they rather choose probabilistically among several stores. This probabilistic choice behavior is modeled by Huff, known as the Huff model [5]. Huff formulated a model for capturing market share, assuming that the probability of a consumer patronizing a shopping center is proportional to its attractiveness and inversely proportional to a power of the distance needed for a consumer to reach it. Although the original Huff model was based on an assumption that a market area is represented by a continuous plane with Euclidean distance, Okabe and Kitamura [10] extended it to the network Huff model by using the shortest path distance on a network. Ghosh et al. [3] considered the problem under the same assumption but for discrete demand (nodal demand). Okunuki and Okabe [11] considered link based demand with slightly changed objective function.

In this paper we apply the network Huff model to a competitive location problem, optimizing new facility locations on a network. We assume that new facilities can be located at any point on the network, and that the demand is generated in the vertices. We introduce a slight modification of the nonlinear mathematical model proposed earlier in [13]. As a step forward with respect to [13], we implemented the model. The implementation was performed by KNITRO software package for solving nonlinear optimization problems, and our computational experience is reported, as well. We considered three different metaheuristics for solving this problem: Variable Neighborhood Search, Simulated Annealing and Multi-Start Local Search metaheuristics for solving this problem. An ampler number of test instances than in [12] is considered and detailed results of the extensive computational testing are shown, as well.

2. Problem formulation

We assume that customers are located in the vertices of a network $\mathcal{N} = (V, E)$, $V = \{v_1, \ldots, v_n\}$, $E \subseteq V^2$. The customers make demand. Further, we assume that there are q facilities already located on the network. The facilities provide service and satisfy the demand. They are located at points $y_1, \ldots, y_q$ on network $\mathcal{N}$. Hence, the facility locations can be network vertices, as well as other points along the edges. Adopting the notation that $w_i = w(v_i)$

* Corresponding author.
E-mail addresses: sanja@mi.sanu.ac.rs (S. Grohmann), draganu@mi.sanu.ac.rs (D. Urošević), ecarrizosa@us.es (E. Carrizosa), nenad@mi.sanu.ac.rs (N. Mladenović).

http://dx.doi.org/10.1016/j.cor.2016.03.005
0305-0548/

is the demand associated with vertex v_i, $i \in \{1, \ldots, n\}$, we assume the following properties:

1. $w_i \geq 0$ and
2. $\sum_{i=1}^{n} w_i = 1$.

The demand w may vary from one vertex to another one. For instance, if the demand among the vertices is considered as a random variable, its distribution can be uniform.

Our goal is to *locate p new facilities* $x_1, \ldots, x_p$ on the network, which will respond to the demand made by customers, so that *the captured demand is maximal.*

To state the above location optimization problem more explicitly, let us formulate the network Huff model on $\mathcal{N}$. Firstly, let us introduce *facility attractiveness*, a property assigned to each facility in the system. Facility attractiveness of a specific facility is a scalar, defining the power of the facility to attract customers. It is not related to the location of a facility, yet, it reflects the rating of the facility. It may be measured by the floor area, by the number of services/items that specific facility offers, by the quality of service, by the level of service updating or in any other predefined way. Therefore, let us denote by $a_{y_1}, \ldots, a_{y_q}$ and $a_{x_1}, \ldots, a_{x_p}$ the attractiveness of the existing and new facilities, respectively. In order to unify the notations and to simplify formulas, let us denote by a_{f_j} either

- the attractiveness of the existing facility, when $f \equiv y$ and $j \in \{1, \ldots, q\}$, or
- the attractiveness of the new facility, when $f \equiv x$ and $j \in \{1, \ldots, p\}$,

located at point f_j. Let $d(v_i, f_j)$ be the distance from the customer located in vertex v_i to the facility at f_j on network $\mathcal{N}$. Let us now introduce *the distance deterrence function* $F(d(v_i, f_j))$ which, actually, involves the distance $d(v_i, f_j)$ between the customer in v_i and the facility at f_j. The distance deterrence function is a monotonically decreasing function with respect to $d(v_i, f_j)$. In his original model, Huff specified the distance deterrence function F as a power function, i.e.

$$F(d(v_i, f_j)) = d(v_i, f_j)^{-\lambda}, \quad \lambda > 0. \tag{1}$$

Eventually, let $P(v_i, f_j)$ be the probability of a customer in v_i choosing facility at f_j among the $q+p$ possible facilities. On these terms, the network Huff model is as follows:

$$P(v_i, f_j) = \frac{a_{f_j} d(v_i, f_j)^{-\lambda}}{\sum_{f_k} a_{f_k} d(v_i, f_k)^{-\lambda}}. \tag{2}$$

Using the network Huff model, we proceed with formulating a problem for obtaining the demand $D(f_j)$ captured by facility at f_j. Let $D(v_i, f_j)$ be the demand in v_i captured by facility at f_j. Since the Huff model gives the probability of the customer in v_i choosing the facility at f_j, $D(v_i, f_j)$ is obtained from multiplying the probability $P(v_i, f_j)$ by $w(v_i)$, i.e.

$$D(v_i, f_j) = P(v_i, f_j) w(v_i) = \frac{a_{f_j} d(v_i, f_j)^{-\lambda}}{\sum_{f_k} a_{f_k} d(v_i, f_k)^{-\lambda}} w(v_i). \tag{3}$$

To obtain the demand $D(f_j)$ captured by facility at f_j, we need to sum Eq. (3) over all vertices $v_i \in V$, i.e.

$$D(f_j) = \sum_{v_i \in V} D(v_i, f_j) = \sum_{v_i \in V} \frac{a_{f_j} d(v_i, f_j)^{-\lambda}}{\sum_{f_k} a_{f_k} d(v_i, f_k)^{-\lambda}} w(v_i). \tag{4}$$

With q existing facilities located at points $y_1, \ldots, y_q$ of network $\mathcal{N}$, we are supposed to locate p new facilities at points $x_1, \ldots, x_p$ in order to compete them and capture maximal demand. The total demand captured only by new facilities is given by the formula

$$\sum_{j=1}^{p} D(x_j) = \sum_{j=1}^{p} \sum_{v_i \in V} \frac{a_{x_j} d(v_i, x_j)^{-\lambda}}{\sum_{f_k} a_{f_k} d(v_i, f_k)^{-\lambda}} w(v_i), \tag{5}$$

where $f \in \{y, x\}$; $k \in \{1, \ldots, q\}$ if $f = y$, and $k \in \{1, \ldots, p\}$ if $f = x$. Since it has to be maximal, the problem we have to solve is

$$\max_{x_1, \ldots, x_p \in \mathcal{N}} \sum_{j=1}^{p} \sum_{v_i \in V} \frac{a_{x_j} d(v_i, x_j)^{-\lambda}}{\sum_{f_k} a_{f_k} d(v_i, f_k)^{-\lambda}} w(v_i). \tag{6}$$

3. A mathematical model for the Huff location problem

In this section we discuss the mathematical programming model for the Huff location problem. Let $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ be a vertex set and an edge set of a network, respectively. If $l : E \longrightarrow \mathbb{R}$ is a weight function defining edge lengths, let $l_i = l(e_i)$ be the length of edge e_i. Since the edge lengths of the graph are known in advance as input data, all pair shortest path distances can be precalculated and considered as input data, too. Therefore, let $d(v_i, v_j)$ be the shortest path distance between vertices v_i and v_j, $\forall i, j \in \{1, \ldots, n\}$.

The location of any point of the graph is given by a triple (v_j, v_k, y), where

- v_j and v_k are endpoints of edge containing the point,
- y is the relative position of the point on edge (v_j, v_k) with respect to edge end v_j.

Let us assign a point to every pair of vertex v and edge $e = (u_e, v_e)$, so that being on the edge e, it is on the largest distance from vertex v. In other words, the distance between the assigned point and vertex v is larger than the distance between vertex v and any other point on the edge e. Relative position M_{ve} of this point on the edge, with regard to preselected endpoint of the edge e, can be expressed as a number from $[0, 1]$. Denote with $dist_{ve}$ the distance between vertex v and the assigned point.

The location of these points are graph properties, therefore, they can be precalculated and considered as input data, as well as their distances $dist_{ve}$ from the corresponding vertex v.

Let us now introduce binary variables x_{fe} (where f is a facility and $e \in E$ is an edge) whose meaning is given with:

$$x_{fe} = \begin{cases} 1, & \text{if facility with index } f \text{ is on edge } e, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Also, we introduce variables y_f whose value is the relative position of facility f on an edge chosen for the facility to be located on. In this context, the shortest path distance $d_{v,f}$ between facility f on edge e and vertex v is:

$$d_{v,f} = dist_{ve} - |M_{ve} - y_f| \, l(e). \tag{8}$$

On the other hand, if facility f is not located on edge e' then, the distance between vertex v and facility f can be described with the inequality:

$$d_{v,f} \geq dist_{ve'} - |M_{ve'} - y_f| \, l(e') - (1 - x_{fe'}) S, \tag{9}$$

where S is a very big number (for example, greater than the sum of lengths of all edges in the graph).

Also, we must bound from above these distances in the following way:

$$d_{v,f} \leq dist_{ve'} - |M_{ve'} - y_f| \, l(e') + (1 - x_{fe'}) S. \tag{10}$$

Finally, we can formulate our problem in the following way:

$$\max_{f_1^n,\dots,f_p^n \in \mathcal{N}} \sum_{j=1}^{p} \sum_{v_i \in V} \frac{a_{f_j^n} d(v_i, f_j^n)^{-\lambda}}{\sum_f a_f d(v_i, f)^{-\lambda}} w(v_i), \tag{11}$$

where f_i^n, $i \in \{1,\dots,p\}$, are locations of new facilities on the graph, $a_{f_j^n}$, $j \in \{1,\dots,p\}$, related attractiveness, while f and a_f are facility locations (either the existing or new ones) on the graph and related attractiveness, respectively.

We have the following constraints:

$$\sum_{e \in E} x_{f_j^n e} = 1 \quad j = 1,2,\dots,p \tag{12}$$

$$d_{v f_j^n} \geq dist_{ve} - | M_{ve} - y_{f_j^n} | l(e) - (1 - x_{f_j^n e}) S, \quad v \in V, e \in E, j = 1,2,\dots,p \tag{13}$$

$$d_{v f_j^n} \leq dist_{ve} - | M_{ve} - y_{f_j^n} | l(e) + (1 - x_{f_j^n e}) S, \quad v \in V, e \in E, j = 1,2,\dots,p \tag{14}$$

$$x_{f_j^n e} \in \{0,1\}, \quad e \in E, j = 1,2,\dots,p \tag{15}$$

$$y_{f_j^n} \in [0,1], \quad j = 1,2,\dots,p. \tag{16}$$

Constraints (12) ensure that every facility is located. By using constraints (13) and (14), we define the lower and the upper bound on the distance between vertices and facilities.

Regarding complexity of the model we have the following facts:

- We have $p \times |E| + p + |V| \times p$ variables ($p \times |E|$ of them are binary, while the others are continuous).
- There are $p + 2 \times p \times |V| \times |E|$ constraints.

4. The metaheuristics and the applications to the Huff location problem

4.1. Variable Neighborhood Search

Variable Neighborhood Search (VNS) [6,7,9] is a well known metaheuristic method. It is designed for solving various optimization problems: continuous as well as combinatorial. The basic idea of VNS metaheuristic is to use more than one neighborhood structure and to proceed with their systematic change within a local search. Unlike many other metaheuristics based on local search methods, VNS does not follow a trajectory, but explores increasingly distant neighborhoods of the current incumbent solution. The search is recentered around a new solution if and only if an improvement has been made with respect to the global best solution. A local search routine is applied repeatedly to find local optima, starting from these neighboring solutions.

Neighborhoods are usually ranked in such a way that intensification of the search around the current solution is followed naturally by diversification. The level of intensification and the level of diversification can be controlled by a few (easy to set) parameters. We may view the VNS as a “shaking” process, where a movement to a neighborhood further from the current solution corresponds to a harder shake. Unlike random restart, the VNS allows a controlled increase in the level of the shake.

Therefore, to construct different neighborhood structures and to perform a systematic search, there must be a way for finding the distance between any two solutions, i.e., the solution space must be supplied with some metric (or quasi-metric) and then, neighborhood structures are derived (induced) from it. In the following sections we answer this problem-specific question for our particular problem.

4.2. The application of VNS to the Huff network model

In order to implement VNS for the specific variant of the Huff location problem, we need to define a solution representation, as well as neighborhood structures and a local search strategy.

4.2.1. Solution space

A particular solution consists of the location set for the p new facilities on the given network. The location of each facility is uniquely determined by the edge, i.e. by the pair of vertices, and the position on the edge. The position on the edge is given by 1-dimension coordinate belonging to the $[0,1]$ interval with respect to one of the vertices of the edge. Therefore, the location of the particular facility is given by the ordered pair $(x,(u,v))$, where the first entry of the pair refers to the position on the edge given by the second entry. The position x is calculated with respect to the first vertex of the pair referring to the edge. As an example, Fig. 1 shows facility F_i located with coordinate x=0.25 on edge (u, v). Since a particular solution consists of p facility locations, it will be presented as a list $[(x_1,(u_1,v_1)),\dots,(x_p,(u_p,v_p))]$ of p ordered pairs where the ith pair corresponds to the ith facility location.

4.2.2. Neighborhood structures

Let us now define a neighborhood structure in the solution space we introduced. If $s = [(x_1,(u_1,v_1)),\dots,(x_p,(u_p,v_p))]$ is a solution, we may choose at random one of p facilities and move it to some of the adjacent edges. Then, we perform local search on the new edge by some of the well known line search techniques (Dichotomous search, Fibonacci search, Golden-section search (see more details of these methods in [1]), etc.) in order to reach the location that improves the objective function the most. We call this operation *rank* 1 *stepping*. Fig. 2 demonstrates a step of facility F_i from edge (u_{i_1}, u_{i_2}) to the adjacent edge (u_{i_2}, u_{i_7}). If we repeat this operation k times, $k \leq p$, we call it *rank k stepping*. We say that a solution s' is at the *step-distance k* from the solution s if s can be transformed into s' by applying the rank k stepping.

In order to improve the implementation performance, we have introduced another type of neighborhood structures. If $s = [(x_1,(u_1,v_1)),\dots,(x_p,(u_p,v_p))]$ is a solution, we may chose at random two of p new facilities of the solution and swap their locations. We call this operation *rank* 1 *swapping*. Fig. 3, for instance, demonstrates a swap of facilities F_i on (u_{i_1}, u_{i_2}), and F_j on (u_{i_1}, u_{i_3}). If we repeat this operation k times, $k < \lfloor p/2 \rfloor$, we call it *rank k swapping*. We say that a solution s' is at the *swap-distance k* from the solution s if s can be transformed into s' by applying the rank k swapping. The best results are obtained by combining these two types of neighborhood structures.

4.2.3. Local search strategy

To complete the VNS implementation, we have to define a local search strategy.

The *first improvement* local search strategy is performed: starting from a solution s, we move a particular new facility from its current position on an edge to an adjacent edge, while performing a line search on the new edge. This process is repeated for

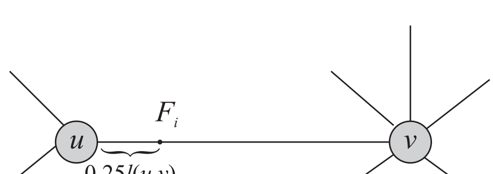

Fig. 1. Physical representation of facility F_i given with a pair $(0.25,(u,v))$.

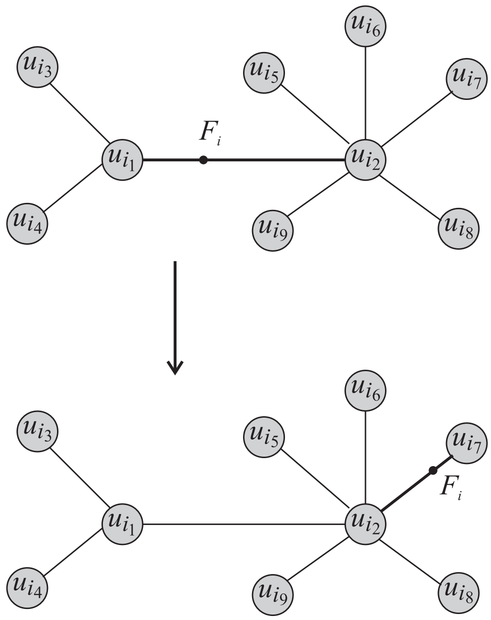

Fig. 2. Rank 1 stepping of facility F_i.

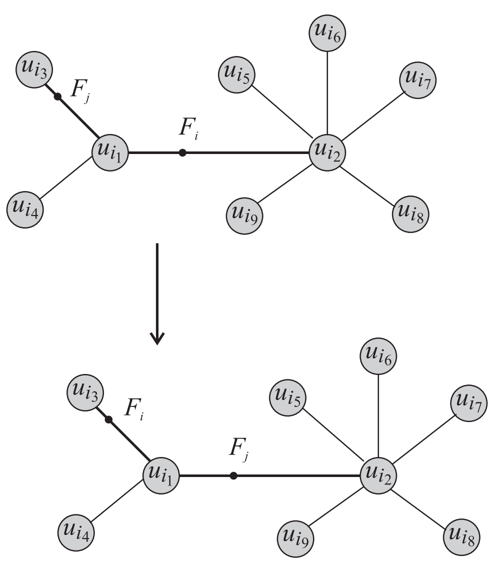

Fig. 3. Rank 1 swapping of facilities F_i and F_j.

all adjacent edges, or until on some edge, the improvement of the objective function value is encountered. After repeating this step for each of the p new facilities, the new solution corresponds to the choice of p facilities where the best objective function value (out of p) is found, and the local search continues from the so obtained solution. We also introduce the *best improvement* local search strategy: starting from a solution s, we move a particular new facility from its current position on an edge to an adjacent edge while performing some line search on the new edge. This process is repeated for all the adjacent edges and the best improvement of the objective function value (if there is any) is stored. After repeating this for each of the p new facilities, the new solution is defined by the best of p stored objective function values and the corresponding positions of facilities. The local search continues from this solution.

We have implemented and tested both local search strategies. The experiments showed that there was not a significant difference between solution improvements obtained either by the first or the second strategy, but execution time for the first strategy was notably shorter. Thus, we continued with the first strategy since it is more suitable for our problem.

To make our algorithms more efficient, we introduced an additional improvement in the search strategies. Namely, while performing the experiments, we have noticed that the optimal facility locations are close to the vertices of graphs. Therefore, while performing line search on an edge, we first exploit the small areas around vertices. In case the local optimum is not found, we extend the searching area, and repeat the procedure, otherwise, we stop. In the worst case, the whole edge is searched, decreasing the time required for obtaining local optimum.

4.2.4. VNS algorithm for the Huff location model

Let us denote by N_k, $k = 1, \ldots, k_{max}$ a finite sequence of pre-selected neighborhood structures, and by $N_k(x)$ the set of feasible solutions corresponding to the neighborhood structure N_k at the point x, where x is a solution. Algorithm 1 demonstrates the application of the basic VNS heuristic to the multifacility Huff location model on a network.

Algorithm 1. Basic VNS algorithm for the Huff location model.

```
Procedure VNSforHuff (k_max)
    x ← Initial Solution();
    Choose a stopping criterion;
    repeat
        k ← 1;
        while k ≤ k_max do
            x' ← Shaking(N_k(x));
            if Random() ≤ Prob_swap then
                x'' ← SwapFirstImprovementLS(x');
            else
                x'' ← MoveFirstImprovementLS(x');
            end
            if x'' is better than x then
                x ← x'';
                k ← 1;
            else
                k ← k + 1;
            end
        end
    until stopping criterion is met;
    return x;
```

Usually, the initial solution is determined by some constructive heuristic, and then improved by local search before the beginning of actual VNS procedure. In our case, the initial solution is generated randomly, and then improved by Fibonacci local search method (for more details regarding Fibonacci search, see [1]). The stopping criterion may be, e.g., a predetermined maximal allowed CPU time, a maximal number of all iterations, or the iterations between two improvements. Here the stopping criterion is maximal allowed CPU time. Often, successive neighborhoods N_k are nested, but not necessarily. Let us note that the point x' is generated at random in order to avoid cycling, which might occur if any deterministic rule was used. Basic VNS is a simple metaheuristic and its only parameter is k_{max}, the preselected number of neighborhoods. However, for each particular problem, the solution representation, the number and order of neighborhoods, and stopping criterion should be defined in a way to ensure efficient execution of the search.

The results obtained by VNS are compared with the results obtained by Simulated Annealing (SA) and Multi-Start Local Search metaheuristics (MSLS).

4.3. Simulated Annealing

Simulated Annealing [8,2] is a stochastic metaheuristic approach for solving optimization problems. It is inspired by the annealing in metallurgy, a technique involving heating and controlled cooling of a material in order to improve its crystal structure and reduce their defects. Each point s of the search space is analogous to a state of some physical system, and the function $E(s)$ to be minimized is analogous to the internal energy of the system in the particular state. The goal is to bring the system from the initial state to a state with the minimum possible energy. At each step, the SA metaheuristic considers some neighboring state s' of the current state s, and probabilistically decides whether to move the system to state s' or to stay in state s. These probabilities ultimately lead the system to states of lower energy. Typically, this step is repeated until the stopping criterion is met, which can be reaching a state that is good enough for the application or the exhaustion of the given computation budget. In order to apply the SA to the Huff location model, we have to decide how to find an initial solution, create the cooling schedule, choose the type of neighborhood, predefine the acceptance probability and the stopping criterion. The initial solution is generated at random. After performing a series of preliminary experiments, we have obtained a rough estimate of potentially good parameter values. Therefore, we define the cooling schedule with decreasing the temperature of 20 by multiplying it by 0.5 after each 50 iterations. The neighboring solution is chosen by applying rank 1 *stepping strategy*. The acceptance probability is given with $e^{(E(s')-E(s))/T}$, where T is current temperature. Eventually, the stopping criterion is the maximal number of iterations.

4.4. Multi-Start Local Search

Multi-Start Local Search [14] is an iterative approach where a single iteration consists of generating a random solution, and performing a local search strategy with the random solution as a starting point. In case there was the improvement of the objective function value, the incumbent is updated. Initial solution is generated randomly.

We apply a first improvement local search strategy. The best improvement local search is time consuming:

- Neighborhoods are relatively big (especially Move neighborhood).
- Exploring each neighbor is also relatively time consuming.

So, in the same time limit, the number of performed first improvement local search may be much greater than the number of best improvement local search. Intuitively, the chance to obtain better solutions increases as the number of performed local search increases.

5. Computational results

Since there is no set of benchmark problems for the Huff location model, we have chosen the problems from the TSPLIB library, where network dimension (number of nodes – customers) varies from 100 to 800. The number q of the existing facilities depends on the number of customers, so we generate instances with different number of existing facilities ($q \in Q = \{\lfloor\frac{n}{20}\rfloor, \lfloor\frac{n}{15}\rfloor, \lfloor\frac{n}{10}\rfloor, \lfloor\frac{n}{8}\rfloor\}$). The number p of new facilities, belonging to the set $P = \left\{\lfloor\frac{q}{2}\rfloor, \left\lfloor\frac{3q}{4}\right\rfloor, q\right\}$, depends on the number of existing facilities, so we define instances with different number of new facilities. In this way, for each network and a set of existing facilities, three test cases are made. The locations of the existing facilities are created in the following way. Firstly, they were chosen randomly. Then, the VNS method was applied with 10% of total running time planned for the VNS algorithm execution for the particular test instance, and with p set to $\left\lfloor\frac{2q}{3}\right\rfloor$. In the end, randomly chosen p, out of q

Table 1
Comparison of VNS with first improvement local search and best improvement local search.

Instance	n	q	p	VNS-fi				VNS-bi				Dev.
				Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	
rat99	99	12	6	33.451	33.214	0.124	98.67	33.356	32.562	0.581	98.67	0.28
rat195	195	24	12	40.733	40.309	0.255	194.35	32.933	31.491	0.939	194.43	19.15
rat575	575	71	35	41.156	40.711	0.352	506.56	28.029	27.482	0.355	574.96	31.90
rat783	783	97	48	40.472	39.728	0.490	716.59	25.717	25.099	0.338	703.12	36.46
rat99	99	12	8	46.576	46.479	0.077	98.67	46.022	44.900	1.156	98.67	1.19
rat195	195	24	16	48.588	47.917	0.389	194.35	35.628	34.072	0.900	194.35	26.67
rat575	575	71	47	46.586	45.841	0.401	505.87	31.716	31.338	0.199	502.04	31.92
rat783	783	97	64	46.873	46.215	0.386	713.26	32.456	32.165	0.184	706.48	30.76
rat99	99	12	9	50.413	50.217	0.119	98.67	49.548	48.625	0.711	98.67	1.72
rat195	195	24	18	52.104	51.055	0.788	194.36	35.520	34.442	0.679	194.35	31.83
rat575	575	71	53	48.651	47.837	0.556	506.89	34.022	33.270	0.584	502.78	30.07
rat783	783	97	72	49.042	48.423	0.404	782.00	35.190	34.445	0.516	706.80	28.25
rat195	195	24	19	53.835	52.234	0.844	194.36	37.230	35.440	0.832	194.35	30.84
rat575	575	71	56	50.541	49.148	0.873	511.35	34.743	34.384	0.394	503.41	31.26
rat783	783	97	77	51.309	49.954	0.798	782.08	36.278	35.710	0.519	707.35	29.30
rat99	99	12	10	55.286	53.590	1.306	98.67	53.766	51.725	1.421	98.67	2.75
rat195	195	24	20	54.949	53.208	0.981	194.36	37.375	36.310	0.614	194.35	31.98
rat575	575	71	59	51.943	50.241	0.967	526.76	36.196	35.397	0.671	503.52	30.32
rat783	783	97	80	52.437	50.746	0.816	782.15	37.438	36.636	0.536	708.47	28.60
rat99	99	12	12	63.732	58.655	2.159	98.67	61.035	55.947	2.405	98.67	4.23
rat195	195	24	24	59.183	57.241	1.420	194.36	40.497	38.233	1.569	194.35	31.57
rat575	575	71	71	55.553	53.517	1.209	539.12	40.699	39.030	1.010	503.62	26.74
rat783	783	97	97	57.253	55.494	0.820	782.47	42.132	40.806	0.784	707.88	26.41

existing facility locations, were switched with new facility locations, obtained by the VNS algorithm. The attractiveness of each facility was chosen randomly.

Our experience shows that the best results are obtained if the probability of choosing either stepping or swapping shaking strategy is set to 0.5. Parameter k_{max} was set to $p/2$. Execution time (in seconds) was set to the number of locations (customers).

5.1. Comparison of first and best improvement strategy for local search

The first set of experiment is dedicated to comparing impact of local search strategy (first improvement and best improvement) on performance of complete Variable Neighborhood Search. In order to compare, we execute 10 times two variants of Variable Neighborhood Search (VNS with first improvement local search and VNS with best improvement local search) on four instances of different sizes (with different value of parameter p). Obtained results are presented in Table 1. The first column of the table contains instance name. The next three columns contain value for n, q, and p, respectively. Columns 4–7 contain summary results for VNS with first improvement local search (best results, average results, standard deviation and average time). Next four columns contain results obtained by VNS with best improvement local search (the same order). The best and average results are presented as the percentage of the total demand serviced by new facilities. The last column contains percentage deviation of best result obtained by VNS based on best improvement from best result obtained by VNS based on first improvement, for corresponding instance. Percentage deviation is calculated by the

Table 2
Comparison of VNS, SA and MSLS on test instances with $q = \lfloor \frac{n}{20} \rfloor$ existing facilities.

Name	n	q	p	VNS				SA				MSLS				% dev.	
				Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	SA	MSLS
gr120	120	6	3	34.92	34.78	0.07	119.00	34.82	34.53	0.46	119.40	31.82	30.99	0.48	120.02	0.30	8.88
gr229	229	11	5	41.48	41.38	0.07	228.00	41.38	40.67	0.57	228.24	34.89	33.68	0.58	229.15	0.23	15.88
gr431	431	21	10	36.76	36.47	0.20	430.01	36.96	36.67	0.23	430.14	25.91	25.66	0.25	432.59	−0.54	29.52
gr666	666	33	16	43.11	42.59	0.36	665.03	41.80	41.02	0.65	600.39	27.58	26.56	0.67	715.41	3.05	36.03
lin105	105	5	2	36.56	36.52	0.02	104.00	36.56	36.49	0.10	104.47	36.00	35.55	0.10	105.01	0.00	1.52
lin318	318	15	7	33.17	32.92	0.17	317.00	33.29	32.91	0.26	317.21	24.03	23.48	0.27	318.31	−0.36	27.55
pcb442	442	22	11	42.21	41.91	0.25	441.01	42.39	41.93	0.28	441.12	33.10	32.18	0.29	444.19	−0.44	21.58
pr124	124	6	3	37.60	37.54	0.17	123.00	37.60	37.11	0.33	123.38	34.75	33.83	0.33	124.02	0.00	7.58
pr152	152	7	3	40.41	40.34	0.16	151.00	40.38	39.90	0.35	151.24	37.34	36.30	0.37	152.05	0.06	7.59
pr226	226	11	5	38.65	38.61	0.03	225.00	38.61	38.18	0.28	225.25	32.78	32.27	0.29	226.17	0.10	15.19
pr264	264	13	6	38.84	38.69	0.16	263.00	38.64	38.04	0.44	263.12	30.34	29.85	0.47	264.58	0.52	21.89
pr299	299	14	7	37.35	37.14	0.16	298.00	37.21	36.89	0.26	298.25	29.64	28.82	0.26	299.33	0.39	20.64
pr439	439	21	10	35.16	34.80	0.29	438.01	35.15	34.76	0.23	438.12	26.02	25.22	0.23	441.60	0.04	25.99
rat99	99	4	2	36.40	36.37	0.05	98.00	36.38	36.32	0.07	98.48	35.68	34.98	0.07	99.01	0.06	1.98
rat195	195	9	4	34.82	34.82	0.01	194.00	34.76	34.26	0.21	194.35	31.13	30.60	0.23	195.04	0.18	10.60
rat575	575	28	14	41.04	40.48	0.39	574.01	40.54	40.21	0.16	500.27	28.83	28.31	0.18	578.37	1.22	29.74
rat783	783	39	19	41.76	40.97	0.36	782.02	39.20	38.90	0.21	700.71	28.76	28.45	0.21	794.82	6.14	31.14
gr120	120	6	4	41.45	41.16	0.21	119.00	41.13	40.78	0.28	119.40	36.27	35.29	0.30	120.03	0.77	12.51
gr229	229	11	8	47.97	47.50	0.22	228.00	47.81	46.99	0.66	228.24	38.90	37.49	0.71	229.23	0.33	18.91
gr431	431	21	15	49.75	49.25	0.33	430.01	50.31	49.81	0.27	430.14	35.86	34.92	0.29	435.04	−1.13	27.92
gr666	666	33	24	49.71	48.11	0.73	665.03	47.71	45.81	1.31	600.68	30.85	29.07	1.39	677.97	4.02	37.93
lin105	105	5	3	40.61	40.51	0.16	104.00	40.61	39.73	0.80	104.47	37.41	36.51	0.83	105.02	0.00	7.88
lin318	318	15	11	51.79	51.29	0.38	317.01	52.44	51.88	0.29	317.21	37.75	37.33	0.30	318.83	−1.25	27.12
pcb442	442	22	16	49.63	48.23	1.04	441.02	49.37	47.95	0.95	441.12	34.26	32.83	0.96	443.80	0.52	30.97
pr124	124	6	4	42.29	42.26	0.08	123.00	42.21	41.71	0.29	123.38	38.55	37.60	0.31	124.03	0.19	8.84
pr152	152	7	5	49.74	49.64	0.13	151.00	49.50	48.82	0.41	151.24	41.22	40.02	0.44	152.09	0.47	17.13
pr226	226	11	8	57.37	57.10	0.17	225.00	57.08	56.40	0.46	225.25	45.11	44.01	0.49	226.80	0.50	21.37
pr264	264	13	9	45.83	45.56	0.27	263.00	46.16	45.38	0.47	263.12	35.94	34.73	0.47	264.58	−0.74	21.57
pr299	299	14	10	44.57	44.30	0.15	298.01	44.61	44.17	0.23	298.25	32.51	31.85	0.25	299.74	−0.09	27.06
pr439	439	21	15	47.09	46.43	0.39	438.01	47.32	46.50	0.46	438.12	31.72	31.02	0.51	442.01	−0.49	32.65
rat99	99	4	3	48.50	48.39	0.05	98.00	49.46	49.20	0.13	98.67	46.37	46.02	0.13	99.02	−1.97	4.39
rat195	195	9	6	46.23	46.10	0.09	194.00	46.20	45.64	0.39	194.35	38.59	37.51	0.41	195.21	0.08	16.53
rat575	575	28	21	53.03	51.25	0.93	574.02	50.72	48.95	1.10	500.44	38.82	37.18	1.15	595.91	4.35	26.80
rat783	783	39	29	51.31	49.57	0.99	782.03	49.86	47.77	0.97	701.37	36.52	34.57	1.05	803.00	2.82	28.82
gr120	120	6	6	54.87	51.59	2.33	119.00	54.54	51.07	2.62	119.40	43.92	40.83	2.85	120.09	0.61	19.96
gr229	229	11	11	63.51	57.40	2.89	228.00	63.44	57.39	3.03	228.24	38.45	34.40	3.21	229.62	0.11	39.46
gr431	431	21	21	59.44	56.99	1.87	430.02	59.99	56.61	2.11	430.14	40.11	37.24	2.28	437.77	−0.93	32.52
gr666	666	33	33	56.47	54.73	1.77	665.04	55.42	52.95	1.94	601.18	28.14	26.56	2.12	739.52	1.87	50.18
lin105	105	5	5	57.43	52.68	3.63	104.00	56.96	51.78	3.57	104.47	40.30	36.23	3.57	105.04	0.81	29.83
lin318	318	15	15	61.68	58.40	1.83	317.01	61.10	57.21	1.78	317.21	47.58	44.31	1.94	321.95	0.93	22.85
pcb442	442	22	22	57.74	55.48	1.77	441.01	55.87	53.50	1.95	441.12	36.88	34.94	2.10	454.84	3.24	36.13
pr124	124	6	6	54.85	51.89	2.30	123.00	54.61	51.27	2.54	123.38	43.19	40.51	2.64	124.07	0.44	21.26
pr152	152	7	7	61.10	57.05	2.29	151.00	61.18	56.49	2.29	151.24	44.50	40.72	2.45	152.24	−0.13	27.17
pr226	226	11	11	75.71	69.43	2.92	225.01	75.13	68.98	2.97	225.25	52.63	47.45	3.01	227.43	0.77	30.49
pr264	264	13	13	63.15	57.21	2.83	263.01	62.46	57.53	2.69	263.12	33.45	30.29	2.74	265.56	1.11	47.04
pr299	299	14	14	60.08	55.13	2.41	298.01	60.58	55.41	2.66	298.25	34.24	31.06	2.66	299.80	−0.84	43.00
pr439	439	21	21	59.42	56.64	1.96	438.02	58.45	54.96	2.19	438.12	43.30	40.04	2.40	459.92	1.63	27.13
rat99	99	4	4	64.45	61.89	1.92	98.00	58.08	57.81	0.23	98.67	52.82	52.09	0.25	99.03	9.88	18.05
rat195	195	9	9	58.92	55.61	1.87	194.00	59.08	55.49	1.97	194.35	46.94	43.27	2.09	195.36	−0.29	20.32
rat575	575	28	28	58.05	56.74	1.55	574.03	56.04	54.58	1.59	500.85	44.67	43.32	1.59	598.31	3.46	23.06
rat783	783	39	39	58.33	56.50	1.74	782.02	57.42	55.03	1.72	702.07	38.66	36.70	1.81	810.64	1.55	33.73
Average				48.87	47.42	0.92	320.60	48.40	46.75	1.03	307.97	36.76	35.19	1.08	327.71	0.85	23.60

formula:

$$dev(f_{fi},f_{bi}) = \frac{f_{fi} - f_{bi}}{f_{fi}} \times 100$$

where f_{fi} is the best solution obtained by VNS based on first improvement local search and f_{bi} is the best solution obtained by VNS based on best improvement.

From this table we can conclude:

- VNS based on first improvement produces better solution on all instances.
- Percentage deviation increases with increasing the size of an instance.
- We suppose that with increasing the size of an instance, the size of the neighborhood (which must be completely explored in case of best improvement local search) also increases.
- For example,
  - network (instance) rat575 contains 575 vertices and 32848 edges,
  - average degree of vertex is 114,
  - each new facility contains in average $2 \times 114 = 228$ neighbors in Move neighborhood,
  - there are between 35 and 71 new facilities (depending on case), so number of neighbors in Move neighborhood is between 8000 and 16,000.

Based on this results, we decided to use the first improvement local search with VNS in the rest of our experiments.

Table 3
Comparison of VNS, SA and MSLS on test instances with $q = \lfloor \frac{n}{15} \rfloor$ existing facilities.

Name	n	q	p	VNS				SA				MSLS				% dev.	
				Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	SA	MSLS
gr120	120	8	4	31.26	31.22	0.13	119.00	31.25	30.83	0.39	119.00	28.49	28.11	0.42	120.04	0.03	8.88
gr229	229	15	7	39.12	38.94	0.16	228.00	39.01	38.70	0.29	228.00	30.95	30.21	0.30	229.37	0.27	20.89
gr431	431	28	14	39.37	38.67	0.45	430.01	39.33	39.05	0.17	430.00	24.10	23.60	0.19	432.54	0.11	38.78
gr666	666	44	22	40.22	39.08	0.63	665.04	37.30	36.63	0.51	665.00	23.85	22.86	0.54	690.99	7.26	40.70
lin105	105	7	3	23.40	22.96	0.29	104.00	22.81	22.54	0.36	105.00	20.35	19.69	0.38	105.02	2.53	13.06
lin318	318	21	10	42.48	42.18	0.24	317.00	42.76	42.39	0.23	317.00	30.04	29.40	0.23	318.57	−0.66	29.30
pcb442	442	29	14	37.50	37.17	0.24	441.01	37.09	36.52	0.42	441.00	26.09	25.77	0.46	447.01	1.12	30.42
pr124	124	8	4	38.74	38.47	0.22	123.00	38.74	38.31	0.29	123.00	34.97	34.14	0.31	124.03	0.00	9.74
pr152	152	10	5	41.52	41.40	0.14	151.00	41.50	40.72	0.42	151.00	34.26	34.14	0.43	152.07	0.04	17.49
pr226	226	15	7	42.42	42.22	0.11	225.00	42.21	41.59	0.51	225.00	31.87	31.13	0.55	226.26	0.48	24.88
pr264	264	17	8	41.24	40.82	0.26	263.00	41.12	40.74	0.29	263.00	30.37	29.82	0.30	264.53	0.30	26.36
pr299	299	19	9	43.08	42.88	0.11	298.01	43.25	42.81	0.38	298.00	32.75	32.57	0.39	300.94	−0.41	23.97
pr439	439	29	14	38.55	37.81	0.40	438.01	36.40	35.75	0.45	438.00	24.64	23.83	0.45	442.53	5.57	36.06
rat99	99	6	3	39.59	39.59	0.00	98.00	39.59	39.39	0.43	99.00	36.20	35.84	0.44	99.02	0.00	8.55
rat195	195	13	6	36.65	36.21	0.76	194.00	36.36	36.02	0.21	194.00	29.74	29.16	0.22	195.13	0.80	18.86
rat575	575	38	19	42.10	41.38	0.40	574.01	39.98	39.27	0.30	574.00	29.56	28.52	0.32	586.87	5.05	29.80
rat783	783	52	26	40.31	39.72	0.43	782.02	38.88	38.08	0.52	782.00	26.10	25.38	0.55	841.61	3.55	35.25
gr120	120	8	6	47.21	46.96	0.21	119.00	47.06	46.32	0.46	119.00	39.38	38.75	0.46	120.02	0.31	16.59
gr229	229	15	11	50.01	49.64	0.23	228.01	49.81	49.27	0.49	228.00	39.00	38.05	0.50	229.88	0.39	22.00
gr431	431	28	21	51.62	50.20	1.02	430.03	49.87	47.70	1.36	430.00	30.33	29.03	1.42	437.21	3.37	41.23
gr666	666	44	33	47.34	46.15	0.92	665.04	45.97	44.71	1.03	665.00	28.69	27.93	1.07	768.88	2.89	39.40
lin105	105	7	5	53.84	53.78	0.13	104.00	53.83	52.78	0.59	104.00	45.98	45.31	0.63	105.02	0.01	14.60
lin318	318	21	15	50.33	49.74	0.34	317.01	50.66	49.43	0.70	317.00	35.72	35.24	0.73	315.64	−0.66	29.02
pcb442	442	29	21	49.86	49.38	0.40	441.02	47.45	46.47	0.73	441.00	35.26	34.59	0.74	456.93	4.82	29.29
pr124	124	8	6	46.96	46.73	0.17	123.00	46.59	46.15	0.35	123.00	39.67	39.44	0.37	124.06	0.78	15.52
pr152	152	10	7	47.18	46.97	0.28	151.00	46.77	46.31	0.41	151.00	37.06	36.88	0.44	152.03	0.87	21.44
pr226	226	15	11	54.65	54.00	0.42	225.01	54.51	53.65	0.51	225.00	39.65	38.73	0.53	226.50	0.26	27.45
pr264	264	17	12	53.87	53.07	0.47	263.01	54.23	53.28	0.37	263.00	38.29	37.09	0.39	265.07	−0.67	28.92
pr299	299	19	14	53.02	51.37	1.30	298.01	53.81	51.46	1.55	298.00	34.64	33.07	1.55	302.16	−1.48	34.67
pr439	439	29	21	52.06	50.88	0.72	438.01	48.49	47.51	0.78	438.00	34.76	33.40	0.86	441.68	6.85	33.23
rat99	99	6	4	49.64	49.42	0.34	98.00	49.64	49.04	0.48	99.00	43.00	42.49	0.51	99.02	0.00	13.39
rat195	195	13	9	48.76	47.52	1.75	194.00	48.51	48.22	0.20	194.00	37.22	36.01	0.21	195.11	0.50	23.67
rat575	575	38	28	51.46	50.07	0.60	574.03	50.24	48.48	0.87	574.00	33.67	32.75	0.92	595.96	2.36	34.56
rat783	783	52	39	49.63	48.35	0.73	782.02	48.89	47.22	1.15	782.00	31.12	30.13	1.24	913.06	1.47	37.29
gr120	120	8	8	60.58	55.64	2.84	119.00	60.80	55.03	3.19	119.00	46.54	42.02	3.49	120.24	−0.36	23.17
gr229	229	15	15	60.60	57.12	1.74	228.00	60.56	57.09	1.94	228.00	41.92	39.04	1.98	234.87	0.06	30.82
gr431	431	28	28	58.85	56.40	1.81	430.05	57.05	54.20	2.39	430.00	32.72	31.03	2.62	442.43	3.06	44.41
gr666	666	44	44	53.06	51.97	0.75	665.03	53.37	51.71	1.21	665.00	29.61	28.44	1.31	671.56	−0.58	44.20
lin105	105	7	7	66.73	62.49	2.55	104.00	66.60	61.52	2.78	104.00	56.57	52.18	3.03	105.20	0.19	15.23
lin318	318	21	21	58.48	56.85	1.57	317.02	57.73	54.82	2.30	317.00	41.74	40.53	2.41	317.34	1.28	28.63
pcb442	442	29	29	57.85	56.05	1.51	441.02	55.40	53.20	1.81	441.00	43.59	41.42	1.82	462.80	4.25	24.65
pr124	124	8	8	59.31	54.72	2.69	123.00	59.45	53.95	2.90	123.00	43.96	40.32	3.09	124.02	−0.24	25.88
pr152	152	10	10	62.79	57.03	2.64	151.01	62.05	56.88	2.45	151.00	38.82	34.71	2.68	152.50	1.17	38.17
pr226	226	15	15	67.65	64.09	2.10	225.02	67.15	63.53	2.24	225.00	45.96	42.76	2.39	227.86	0.74	32.06
pr264	264	17	17	63.01	60.03	1.66	263.02	62.99	59.65	1.70	263.00	44.62	41.67	1.71	265.65	0.04	29.20
pr299	299	19	19	61.47	58.85	1.87	298.02	60.78	57.67	1.90	298.00	36.59	34.97	2.09	304.08	1.12	40.48
pr439	439	29	29	59.51	57.77	1.73	438.03	56.09	54.45	1.56	438.00	38.97	37.66	1.59	472.97	5.75	34.51
rat99	99	6	6	63.67	61.35	1.81	98.00	63.12	60.63	1.88	98.00	53.13	51.05	1.98	99.16	0.87	16.56
rat195	195	13	13	58.51	55.58	1.53	194.01	59.64	55.79	1.70	194.00	45.87	43.24	1.79	196.94	−1.93	21.61
rat575	575	38	38	57.11	55.76	1.45	574.03	55.76	54.12	1.46	574.00	36.31	34.83	1.58	703.19	2.37	36.42
rat783	783	52	52	56.94	54.24	1.22	782.03	54.96	52.24	1.43	782.00	33.39	31.53	1.53	901.99	3.48	41.37
Average				49.83	48.45	0.91	320.60	49.16	47.53	1.04	320.65	35.84	34.52	1.10	335.83	1.36	27.50

5.2. Comparison with other methods

All three algorithms for solving the specific Huff location model were implemented in C programming language on Linux platform. The experiments have been run on the computer with the i686 Intel Core 2 Duo CPU E6750 at 2.66 GHz and 8 GB RAM. The summary results based on ten executions of each method for each test instance are presented in Tables 2 (instances with $q = \lfloor \frac{n}{20} \rfloor$ existing facilities), 3 (instances with $q = \lfloor \frac{n}{15} \rfloor$ existing facilities), 4 (instances with $q = \lfloor \frac{n}{10} \rfloor$ existing facilities), and 5 (instances with $q = \lfloor \frac{n}{8} \rfloor$ existing facilities). The first column of these tables contains the instance name. Next three columns contain information about the instance (the number of vertices/customers (n), the number of existing facilities (q), and the number of new facilities (p)). Next four columns contain summary information for results obtained by Variable Neighborhood Search (best result, average result, standard deviation, and average time for ten executions). The best and average results are presented as the percentage of the total demand captured by new facilities. Next four columns contain summary information about results obtained by Simulated Annealing (SA in the rest). Next four columns contain information about results obtained by Multistart local search (MSLS in the rest). Two last columns contain percentage deviation of best solutions obtained by SA and MSLS from best solution obtained by VNS, for corresponding instance.

All three methods were given the same total execution time for a particular instance, depending on the size of the test instance. The number of seconds allowed for an instance to run is set to the number of vertices of the graph.

From these tables, we conclude the following:

Table 4
Comparison of VNS, SA and MSLS on test instances with $q = \lfloor \frac{n}{10} \rfloor$ existing facilities.

Name	n	q	p	VNS				SA				MSLS				% dev.	
				Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	SA	MSLS
gr120	120	12	6	41.29	41.11	0.12	119.00	41.07	40.76	0.25	119.00	33.46	32.55	0.30	120.13	0.55	18.98
gr229	229	22	11	39.98	39.77	0.19	228.01	40.26	39.84	0.40	228.00	27.39	26.95	0.46	230.05	−0.69	31.50
gr431	431	43	21	42.60	41.48	0.62	430.02	39.26	38.77	0.42	430.00	25.50	24.97	0.46	444.76	7.85	40.15
gr666	666	66	33	42.68	41.54	0.58	665.03	41.98	40.59	0.75	665.00	24.00	22.80	0.88	704.55	1.64	43.77
lin105	105	10	5	42.50	42.19	0.25	104.00	42.03	41.49	0.38	104.00	35.67	34.54	0.40	105.08	1.13	16.09
lin318	318	31	15	44.31	43.92	0.31	317.01	44.11	42.92	0.81	317.00	29.49	28.54	0.90	323.14	0.46	33.45
pcb442	442	44	22	41.22	40.65	0.39	441.01	38.81	38.31	0.36	441.00	28.38	27.92	0.42	458.68	5.85	31.15
pr124	124	12	6	35.71	35.49	0.20	123.00	35.63	34.87	0.50	123.00	28.31	27.71	0.55	124.13	0.24	20.73
pr152	152	15	7	36.06	35.75	0.22	151.00	35.68	35.11	0.40	151.00	27.02	26.28	0.40	152.26	1.05	25.08
pr226	226	22	11	45.66	45.40	0.18	225.01	45.45	44.52	0.59	225.00	27.84	26.91	0.63	227.63	0.48	39.03
pr264	264	26	13	42.07	41.36	0.41	263.01	41.70	41.06	0.45	263.00	25.42	24.99	0.51	265.76	0.88	39.57
pr299	299	29	14	42.77	42.24	0.38	298.01	42.41	41.66	0.58	298.00	28.01	27.15	0.70	308.67	0.85	34.51
pr439	439	43	21	43.40	42.62	0.48	438.03	40.03	39.23	0.47	438.00	25.17	24.44	0.48	460.68	7.76	42.00
rat99	99	9	4	35.38	35.21	0.10	98.00	35.51	35.12	0.19	99.00	32.01	31.11	0.19	99.03	−0.38	9.54
rat195	195	19	9	40.10	39.80	0.21	194.00	40.48	39.77	0.42	194.00	29.70	28.90	0.46	195.40	−0.94	25.93
rat575	575	57	28	43.47	42.87	0.38	574.02	42.00	41.03	0.40	574.00	27.93	27.03	0.45	622.93	3.39	35.75
rat783	783	78	39	38.59	36.81	1.58	782.03	38.32	35.97	1.44	782.00	22.61	21.00	1.55	1117.47	0.70	41.41
gr120	120	12	9	48.68	48.55	0.12	119.00	50.57	50.25	0.21	119.00	40.14	39.57	0.24	120.29	−3.89	17.54
gr229	229	22	16	46.89	46.42	0.26	228.01	46.31	44.95	1.02	228.00	36.37	34.99	1.06	232.64	1.25	22.44
gr431	431	43	32	51.57	50.74	0.60	430.02	49.89	47.04	1.26	430.00	28.30	26.58	1.32	478.74	3.25	45.12
gr666	666	66	49	52.70	51.89	0.73	665.04	49.60	47.76	1.39	665.00	28.96	27.55	1.61	991.32	5.89	45.05
lin105	105	10	7	51.92	51.86	0.14	104.00	52.57	51.92	0.44	104.00	43.08	42.33	0.49	105.19	−1.24	17.03
lin318	318	31	23	52.12	51.55	0.26	317.01	49.75	47.64	1.16	317.00	32.75	31.07	1.20	329.74	4.55	37.16
pcb442	442	44	33	50.48	49.99	0.45	441.03	47.77	46.13	1.02	441.00	31.16	30.09	1.14	472.87	5.37	38.27
pr124	124	12	9	46.48	46.36	0.10	123.00	43.99	43.13	0.65	123.00	31.81	30.83	0.73	124.48	5.36	31.56
pr152	152	15	11	47.42	46.98	0.23	151.00	47.56	46.71	0.38	151.00	37.64	36.43	0.43	152.47	−0.31	20.62
pr226	226	22	16	54.55	54.03	0.27	225.01	53.96	52.37	0.98	225.00	33.64	32.21	1.10	228.98	1.09	38.33
pr264	264	26	19	47.55	47.16	0.32	263.01	47.17	45.21	1.02	263.00	30.22	28.87	1.09	270.30	0.79	36.45
pr299	299	29	21	52.73	51.85	0.43	298.02	46.98	45.96	0.82	298.00	33.06	31.97	0.90	302.93	10.92	37.31
pr439	439	43	32	52.68	52.18	0.47	438.03	48.57	47.55	0.71	438.00	31.97	30.71	0.80	481.22	7.79	39.31
rat99	99	9	6	44.96	44.78	0.17	98.00	45.71	44.27	0.76	98.00	38.42	37.17	0.82	99.09	−1.67	14.54
rat195	195	19	14	52.98	52.70	0.21	194.00	52.53	51.03	0.97	194.00	38.06	36.25	1.09	196.40	0.85	28.16
rat575	575	57	42	54.03	53.27	0.41	574.03	49.91	47.98	0.99	574.00	31.26	29.82	1.08	695.14	7.64	42.15
rat783	783	78	58	48.61	46.68	1.71	782.03	46.09	43.88	1.40	782.01	27.75	26.23	1.40	1683.53	5.18	42.92
gr120	120	12	12	60.19	58.10	1.97	119.00	59.65	54.00	2.59	119.00	44.87	40.24	3.05	120.44	0.89	25.45
gr229	229	22	22	59.84	57.82	2.12	228.01	58.43	55.50	1.97	228.00	40.16	37.68	2.01	241.79	2.36	32.89
gr431	431	43	43	57.34	54.71	2.21	430.03	55.42	53.10	2.00	430.00	28.33	27.07	2.18	512.57	3.34	50.59
gr666	666	66	66	53.21	52.27	0.60	665.10	52.94	50.81	1.42	665.01	29.91	28.26	1.58	883.45	0.52	43.80
lin105	105	10	10	61.62	59.95	1.30	104.00	61.45	56.33	2.19	104.00	52.71	48.11	2.49	105.46	0.27	14.46
lin318	318	31	31	60.34	58.40	1.93	317.05	57.35	54.78	1.73	317.00	35.55	33.46	1.76	341.81	4.96	41.09
pcb442	442	44	44	55.05	53.34	1.45	441.02	53.43	51.95	1.19	441.00	30.87	29.45	1.27	548.33	2.95	43.93
pr124	124	12	12	54.84	52.30	1.99	123.01	59.78	53.06	2.81	123.00	33.63	29.44	2.95	124.49	−9.00	38.68
pr152	152	15	15	59.29	57.26	1.61	151.01	60.40	57.34	1.64	151.00	40.18	37.57	1.89	152.40	−1.87	32.23
pr226	226	22	22	63.84	61.74	1.86	225.03	58.16	56.76	1.80	225.00	41.57	40.32	2.15	229.77	8.90	34.89
pr264	264	26	26	58.73	57.31	1.67	263.02	55.34	53.53	1.54	263.00	37.74	36.23	1.56	285.81	5.78	35.75
pr299	299	29	29	59.50	57.05	2.08	298.02	55.58	53.05	2.01	298.00	41.51	39.26	2.09	340.01	6.59	30.23
pr439	439	43	43	57.42	55.99	1.31	438.03	55.59	52.63	1.69	438.00	41.76	38.97	1.77	499.31	3.18	27.27
rat99	99	9	9	63.98	59.60	2.69	98.00	64.94	59.07	3.07	98.00	50.20	45.06	3.46	99.34	−1.49	21.54
rat195	195	19	19	59.65	57.80	1.81	194.01	59.14	56.41	1.66	194.00	38.98	36.53	1.97	202.96	0.87	34.66
rat575	575	57	57	55.97	54.09	1.32	574.03	55.84	52.96	1.39	574.01	32.73	30.77	1.40	779.75	0.23	41.52
rat783	783	78	78	55.00	52.96	1.28	782.01	49.65	47.31	1.37	782.01	31.89	29.91	1.38	2398.68	9.72	42.02
Average				50.00	48.94	0.84	320.60	48.76	46.93	1.10	320.61	33.43	31.86	1.20	407.69	2.39	32.82

Table 5
Comparison of VNS, SA and MSLS on test instances with $q = \lfloor \frac{n}{8} \rfloor$ existing facilities.

Name	n	q	p	VNS				SA				MSLS				% dev.	
				Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	Best	Avg.	Std.	Time	SA	MSLS
gr120	120	15	7	37.26	37.18	0.07	119.40	37.00	36.47	0.27	119.40	29.83	29.37	0.29	119.40	0.71	19.95
gr229	229	28	14	41.01	40.60	0.28	228.24	41.26	40.59	0.39	228.24	28.67	27.65	0.43	228.24	−0.61	30.08
gr431	431	53	26	40.41	39.26	0.54	430.15	37.22	36.74	0.41	430.14	22.02	21.33	0.44	430.17	7.91	45.52
gr666	666	83	41	42.99	42.40	0.53	617.05	42.18	40.99	0.80	602.93	24.52	23.54	0.93	665.09	1.89	42.96
lin105	105	13	6	41.63	41.52	0.07	104.48	41.30	40.81	0.32	104.48	33.89	33.29	0.36	104.48	0.80	18.59
lin318	318	39	19	41.52	40.94	0.31	317.22	37.50	36.55	0.61	317.21	24.87	23.79	0.71	317.22	9.69	40.09
pcb442	442	55	27	41.76	41.29	0.27	441.13	39.43	38.85	0.41	441.12	25.94	25.28	0.47	441.16	5.58	37.88
pr124	124	15	7	29.77	29.62	0.12	123.38	29.45	29.05	0.37	123.38	23.65	22.97	0.38	123.38	1.07	20.54
pr152	152	19	9	41.05	40.83	0.26	151.24	40.84	40.30	0.35	151.24	28.44	27.99	0.39	151.24	0.52	30.71
pr226	226	28	14	44.64	44.24	0.22	225.26	44.38	43.47	0.59	225.25	27.52	26.55	0.62	225.26	0.56	38.34
pr264	264	33	16	42.33	41.56	0.44	263.14	37.67	37.25	0.26	263.12	22.63	21.95	0.30	263.14	11.02	46.53
pr299	299	37	18	43.56	43.01	0.42	298.27	42.71	41.21	0.83	298.25	27.67	26.39	0.97	298.27	1.95	36.48
pr439	439	54	27	41.33	40.28	0.40	438.15	38.52	37.39	0.56	438.12	23.84	22.75	0.57	438.15	6.80	42.32
rat99	99	12	6	33.45	33.21	0.12	98.67	33.08	32.47	0.39	98.67	27.59	26.74	0.42	98.67	1.12	17.52
rat195	195	24	12	40.73	40.31	0.26	194.35	40.40	40.09	0.38	194.35	29.11	28.88	0.39	194.35	0.81	28.53
rat575	575	71	35	41.16	40.71	0.35	506.56	41.61	40.07	0.66	501.88	25.25	23.99	0.68	539.45	−1.10	38.65
rat783	783	97	48	40.47	39.73	0.49	716.59	39.73	38.32	0.90	703.28	27.41	26.15	0.96	782.06	1.84	32.28
gr120	120	15	11	49.63	49.42	0.12	119.40	49.27	48.74	0.34	119.40	38.17	37.09	0.38	119.40	0.71	23.10
gr229	229	28	21	52.03	50.11	0.98	228.25	50.89	47.86	1.52	228.24	30.84	28.82	1.57	228.26	2.20	40.72
gr431	431	53	39	49.90	47.95	0.78	430.16	47.90	45.85	1.63	430.14	29.11	27.73	1.77	430.28	4.02	41.67
gr666	666	83	62	49.91	48.29	0.67	652.61	47.44	45.13	1.34	604.56	32.07	29.95	1.54	665.43	4.96	35.75
lin105	105	13	9	47.93	47.41	0.29	104.48	47.23	46.68	0.39	104.48	36.59	35.98	0.45	104.48	1.47	23.66
lin318	318	39	29	51.56	50.43	0.73	317.24	48.39	46.62	1.42	317.21	30.67	29.21	1.42	317.25	6.14	40.51
pcb442	442	55	41	50.01	48.86	0.54	441.15	47.91	46.74	0.66	441.12	30.87	30.00	0.69	441.20	4.20	38.26
pr124	124	15	11	51.42	50.87	0.32	123.38	50.32	49.82	0.46	123.38	37.10	36.55	0.52	123.38	2.14	27.85
pr152	152	19	14	52.32	51.09	0.89	151.25	51.70	50.68	0.73	151.24	33.66	32.50	0.84	151.25	1.18	35.67
pr226	226	28	21	56.17	54.74	1.00	225.26	52.95	50.55	1.47	225.25	30.48	28.76	1.50	225.27	5.73	45.73
pr264	264	33	24	52.72	51.44	0.86	263.13	48.71	46.95	0.88	263.12	30.43	28.93	0.94	263.21	7.60	42.28
pr299	299	37	27	53.11	50.83	1.25	298.28	49.75	47.22	1.09	298.25	32.50	30.33	1.21	298.26	6.34	38.81
pr439	439	54	40	49.58	48.44	0.61	438.15	47.50	45.65	1.13	438.13	30.06	28.81	1.26	438.16	4.20	39.36
rat99	99	12	9	50.41	50.22	0.12	98.67	49.78	48.93	0.74	98.67	39.45	38.07	0.76	98.67	1.26	21.76
rat195	195	24	18	52.10	51.06	0.79	194.36	51.94	50.37	0.84	194.35	36.13	34.94	0.96	194.36	0.32	30.67
rat575	575	71	53	48.65	47.84	0.56	506.89	47.87	45.37	1.37	504.00	30.86	28.82	1.56	574.32	1.61	36.58
rat783	783	97	72	49.04	48.42	0.40	782.00	45.13	43.55	0.88	710.88	35.63	34.34	1.01	782.62	7.98	27.34
gr120	120	15	15	62.01	58.35	1.69	119.40	62.06	58.14	1.85	119.40	38.17	35.55	2.17	119.40	−0.08	38.45
gr229	229	28	28	58.01	56.66	1.38	228.24	54.57	53.12	1.54	228.24	30.84	30.00	1.67	228.26	5.93	46.83
gr431	431	53	53	55.06	54.02	1.13	430.18	55.10	51.65	2.26	430.14	29.11	27.15	2.32	430.28	−0.07	47.14
gr666	666	83	83	57.12	55.19	1.03	665.41	49.00	47.05	1.13	607.30	32.07	30.45	1.23	665.43	14.21	43.85
lin105	105	13	13	62.68	58.01	2.05	104.48	61.41	57.22	2.03	104.48	36.59	33.50	2.22	104.48	2.04	41.62
lin318	318	39	39	57.72	56.39	1.46	317.22	55.00	52.79	1.39	317.21	30.67	29.25	1.61	317.25	4.71	46.86
pcb442	442	55	55	56.62	54.77	1.10	441.17	53.35	51.15	1.25	441.12	30.87	29.39	1.42	441.20	5.79	45.48
pr124	124	15	15	62.49	59.70	1.53	123.38	61.97	59.03	1.93	123.38	37.10	34.70	2.29	123.38	0.82	40.63
pr152	152	19	19	60.58	58.12	1.48	151.25	60.13	56.49	2.18	151.24	33.66	31.13	2.51	151.25	0.75	44.44
pr226	226	28	28	62.68	61.31	1.57	225.28	56.39	54.29	2.00	225.25	30.48	28.99	2.12	225.27	10.04	51.37
pr264	264	33	33	59.93	57.94	1.07	263.14	54.90	52.99	1.23	263.12	30.43	29.05	1.40	263.21	8.38	49.23
pr299	299	37	37	60.33	58.47	1.72	298.28	56.93	55.19	1.26	298.25	32.50	31.29	1.47	298.26	5.65	46.14
pr439	439	54	54	56.41	54.93	1.04	438.22	53.87	51.30	1.62	438.13	30.06	28.44	1.69	438.16	4.49	46.70
rat99	99	12	12	63.73	58.66	2.16	98.67	61.88	57.37	2.14	98.67	39.45	35.94	2.45	98.67	2.90	38.11
rat195	195	24	24	59.18	57.24	1.42	194.36	58.03	54.46	2.06	194.35	36.13	33.32	2.09	194.36	1.94	38.96
rat575	575	71	71	55.55	53.52	1.21	539.12	52.33	48.63	1.93	506.66	30.86	28.55	2.30	574.32	5.80	44.46
rat783	783	97	97	57.25	55.49	0.82	782.47	48.43	45.73	1.35	714.38	35.63	33.53	1.55	782.62	15.41	37.76
Average				50.17	48.88	0.78	315.06	48.12	46.35	1.07	308.92	31.02	29.60	1.18	320.22	3.948	37.348

- VNS and SA significantly outperform MSLS (for example, average percentage deviation of results obtained by MSLS from results obtained by VNS presented in Table 2 is 23.61).
- Results obtained by VNS are in average better than the results obtained by SA (0.85 for instances with $q = \lfloor \frac{n}{20} \rfloor$ existing facilities, 2.39 for instances with $q = \lfloor \frac{n}{10} \rfloor$ existing facilities).
- In general, the results do not depends on the number of existing facilities (for example average of best results obtained by VNS for instances with $q = \lfloor \frac{n}{20} \rfloor$ is 48.868, while average of best results obtained by VNS for instances with $q = \lfloor \frac{n}{10} \rfloor$ is 49.999).
- Note that percentage of total demand assigned to new facilities are greater than the percentage participation in total number of facilities: for example average of best results obtained by VNS for test instances with $q = \lfloor \frac{n}{20} \rfloor$ and $p = \lfloor \frac{q}{2} \rfloor$ is 38.249 while percentage participation of new facilities is 33.333.

5.3. Statistical test

In order to confirm the superiority of the method based on VNS over the method based on SA (taking into account that results obtained by MSLS are significantly worse), we perform a statistical test known as the Wilcoxon signed-rank test [16]. For this purpose we compute the differences between the solutions obtained by the two compared algorithms in each instance and then rank them according to their absolute values. The sum of ranks for the instances in which the first algorithm (i.e. algorithm based on VNS) outperforms the second algorithm (algorithm based on SA) is denoted as R^+, while R^- denotes the sum of ranks for the reverse case. Ranks corresponding to zero differences are split evenly among the sums. If $\min\{R^+, R^-\}$ is less than or equal to the critical

Table 6
Statistical comparison on used instances grouped according to proportion of the existing facilities (critical value=454).

Instance group	Num. of inst.	R^+	R^-	Sign.
$q=\lfloor\frac{n}{20}\rfloor$	51	992	334	+
$q=\lfloor\frac{n}{15}\rfloor$	51	1108	218	+
$q=\lfloor\frac{n}{10}\rfloor$	51	1113	213	+
$q=\lfloor\frac{n}{8}\rfloor$	51	1305	21	+

value, this test detects significant differences between the algorithms, which means that an algorithm outperforms its opponent.

Detailed results of this statistical test are given in Table 6. The first column of Table 6 contains description of the group of instances. The second column contains the number of instances in the corresponding group. Columns 3 and 4 contain corresponding sum of ranks (column 3 for VNS metaheuristic and column 4 for SA). The last column indicates whether the Wilcoxon test found statistical differences between these algorithms (+ if a significant difference is found, and − otherwise).

Tests are performed on all instances examined above, with significance level $\alpha=0.05$. The critical value is taken from statistical tables. Critical value for each test is given in the caption of the table. The results from Table 6 clearly confirm the superiority of VNS approach over the SA approach.

5.4. Getting an exact solution

In order to check whether the proposed model for solving the multifacility Huff location problem with nodal demands is correct, we have implemented it in KNITRO, a software package for solving nonlinear optimization problems exactly. It is a commercial software developed by Ziena Optimization LLC. We have used version 8.1.1.

We first tried to solve exactly a 20-node test instance derived randomly from the 55-node data set of Swain [15]. The number of edges, the number of existing facilities, and the number of new facilities are set to 50, 5, and 3, respectively. After a day and a half of execution, KNITRO solver finished its work with a message that the node limit of the search tree has been reached. Thus, the problem that could not be solved had the total number of variables $3\times 50+3+20\times 3=213$, while the total number of constraints was 6003.

In order to get some conclusions regarding the size of instances solvable by the solver, and also to check the correctness of our mathematical model, we have tried with the smaller instance: the number of nodes equals to 15; the number of edges equals to 37; the number of existing facilities equals to 5, and the number of new facilities equals to 3. This instance has 159 variables (111 binary ones) and 3333 constraints. KNITRO managed to get the optimal solution value of 57.0360% captured demands after 11,224 s (more than 3 h). As a comparison, the VNS finds the same optimal solution in just 1.73 s. The last experiment confirms that our model is correct. Execution time by commercial solver (more than 3 h) is really very large. Thus, developing heuristics for this problem appears to be a good idea.

6. Conclusion and future lines

Although there are other approaches in modeling real world competitive location situations, the advantage of the Huff location model is its consumer psychology orientation. It allows taking into consideration facility specific features, therefore, providing us with the more refined model, which eventually implies increasing the demand satisfiability. We have shown that VNS performs very well in solving the multifacility Huff location problem with nodal demand. We have shown in practice that the model we have proposed is correct. Yet, obtaining exact solutions for the proposed test instances is time consuming and, therefore, while dealing with the nonlinear optimization problem with large scale solution spaces, metaheuristics, as opposed to exact solving, are an inevitable approach.

Acknowledgments

This research has been supported by Serbian Ministry of Sciences, project #174010.

References

[1] Antoniou A, Lu WS. Practical optimization: algorithms and engineering applications. New York: Springer; 2007.
[2] Černý V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J Optim Theory Appl 1985;45:41–51.
[3] Ghosh A, McLafferty S, Craig CS. Multifacility retail networks. In: Drezner Z, editor. Facility locations: a survey of applications and methods. New York: Springer; 1995. p. 301–30.
[4] Hakimi SL. On locating new facilities in a competitive environment. Eur J Oper Res 1983;12:29–35.
[5] Huff DL. A probabilistic analysis of shopping center trade areas. Land Econ 1963;39:81–90.
[6] Hansen P, Mladenović N. Variable neighborhood search. In: Burke EK, Kendall G, editors. Search methodologies: introductory tutorials in optimization and decision support techniques. New York: Springer; 2005. p. 211–38.
[7] Hansen P, Mladenović N. Variable neighborhood search methods. In: Floudas C, Pardalos P, editors. Encyclopedia of optimization. New York: Springer; 2009. p. 3975–89.
[8] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220:671–80.
[9] Mladenović N, Hansen P. Variable neighborhood search: principles and applications. Eur J Oper Res 1997;130:449–67.
[10] Okabe A, Kitamura M. A computational method for market area analysis on a network. Geogr Anal 1996;28:330–49.
[11] Okunuki KI, Okabe A. Solving the Huff-based competitive location model on a network with link-based demand. Ann Oper Res 2002;111:239–52.
[12] Roksandić S, Carrizosa E, Urošević D, Mladenović N. Solving multifacility Huff location models on networks using variable neighborhood search and multi-start local search metaheuristics. In: Proceedings of EURO mini conference XVIII on variable neighborhood search, Herceg Novi, Montenegro; 2012. p. 121–8.
[13] Roksandić S, Carrizosa E, Mladenović N, Urosević D. Solving multifacility Huff location models on networks. In: Proceedings of international conference on industrial engineering and systems management (IEEE-IESM'2013), Rabat, Morocco; 2013. p. 420–5.
[14] Snyman JA, Fatti LP. A multi-start global minimization algorithm with dynamic search trajectories. J Optim Theory Appl 1987;54:121–41.
[15] Toregas C, Swain R, ReVelle C, Bergman L. The location of emergency service facilities. Oper Res 1971;19:1363–73.
[16] Wilcoxon F. Individual comparisons by ranking methods. Biometrics 1945;1:80–3.