https://doi.org/10.26552/com.C.2005.2.9-12

KOMUNIKÁCIE COMMUNICATIONS

Ľuboš Buzna *

THE EXACT AND NEAR OPTIMAL SOLUTION OF THE COMPETITIVE UNCAPACITATED LOCATION PROBLEM

This paper is dedicated to the recent unprecedented boom of new supermarkets and hypermarkets in middle Europe. The motivation is to provide for the newcomers the tool for decision support, and to help answer the questions: "Is still economically advantageous to build new shopping malls and where to locate them?" We introduce the four versions of data correcting algorithm and three heuristics: simple exchange heuristic, exchange heuristics enhanced by simulated annealing metaheuristic, and genetic algorithm. All these methods were examined at the benchmarks of practical nature. This research showed the data correcting method as useable for practical instances of this problem. There were also identified interesting dependencies between the computational time and the number of competitors.

1. Introduction

In the area of middle Europe enormous boom of supermarkets and hypermarkets in the last years can be observed. They quickly get a favour of customers and are very popular. In this context arises a question where the limits for supermarket chains extensions are. Before the newcomer comes or before the existing supermarket chain is extended, it is essential to estimate if these steps can bring a profit. The future success of a shopping center depends on the quality of services, competitive advantages, sale management, and location of the store. The location of the store is considered as the most important because the neighbourhood of shopping center determines the level of competition, costs for supply management, labour costs, volume of demand, etc. In this paper we deal with the quantitative approach which allows performing a simple analysis of the supermarket chain extension. This approach can be used as a decision support tool for this difficult problem.

This paper is organized as follows. Section 2 presents a mathematical model of this problem. Section 3 is dedicated to the data correcting method, which allows solving this model to optimality. In section 4 the used heuristic methods are briefly described and in section 5 the results of computational experiments are presented. The set of benchmarks was generated from the digital road network of the Slovak Republic (Cenek, Jánošíková, 2000). To conclude, section 6 attempts to identify an applicability area of presented exact methods.

2. The Competitive Uncapacitated Location Problem

Let us assume that we can neglect some secondary indicators like decreased proneness to save money, possibility to shop cheaper, and subsequent economic grow. Thereafter, the investor's decisions to built a new shopping center do not have substantive influence on incomes and expenses of the customers. The customers still need to satisfy their demand and must manage their budget. For a short time period we can suppose that the cost of a shopping center location, the operational cost, and the appropriate profit rate have to be covered by the profit of existing competitors. The success of a newcomer depends on competitive advantage and, as we can often see in practice, on appropriate location of the new shopping center.

This problem can be modelled by using the graph $G(V, E)$. Let $V = \{1, 2, 3, ..., n\}$ be a set of nodes where competitors, customers or potential newcomers are situated. We assume the aggregated customer, it means that one populated municipality represents one customer and the size of demand is proportional to the number of dwellers. Consider a set of arcs $E = \{(i, j): i,j \in V\}$ and let $d_{ij} \in \Re^+$ be the length of arc (i, j) in kilometres. The average purchase ability of the aggregated customer is denoted $w_i \in \Re^+$ for every $i \in V - (X \cup Y)$. The set $Y \subset V$ contains the nodes where the competitors are placed, and the set of potential locations of newcomers is denoted $X \subset V$. The deterministic costs for a shopping center location is denoted $f_j \in \Re^+$ for every $j \in X$. We set up the constants a_{ij} to have a possibility to estimate the proportion of customers' demand attracted by newcomers. The constants a_{ij} represent the shopping center $i \in X \cup Y$ attractiveness for the customer $j \in V - (X \cup Y)$. These constants should respect the preferences of customers, the service quality, and price rate, which all are very difficult to quantify. To keep a reasonable simplicity of the model we decided to use in a_{ij} calculation only the length d_{ij} between the shopping center i and the customer j. In accordance with (Benati, 2003) we used the formula $a_{ij} = \exp(-\beta d_{ij})$, where the β is the coefficient of spatial friction. This coefficient was statistically estimated in (Benati, 2003). Let x_j be a binary variable taking on value 1, if a new facility is located in j for every $j \in X$ and 0 otherwise. The size of demand, which has been taken over by all newcomers then can be expressed by the formula:

* **Ľuboš Buzna**
Faculty of Management Science and Informatics, University of Žilina, E-mail: lubos.buzna@fri.utc.sk

$$w_i = \frac{\sum_{j \in X} a_{ij} x_j}{\sum_{j \in X} a_{ij} x_j + \sum_{j \in Y} a_{ij}}.$$

Consecutively, we can define the solved problem as the following combinatorial problem:

$$\max_{Q \subseteq X} f(Q) = \sum_{i \in V-(X \cup Y)} w_i \frac{\sum_{j \in Q} a_{ij}}{\sum_{j \in Q} a_{ij} + b_i} - \sum_{j \in Q} f_i, \qquad (1)$$

where $b_i = \sum_{k \in Y} a_{ik}$ and $Q = \{i: i \in X \wedge x_i = 1\}$. The objective function (1), see (Benati, Hansen, 2002) has a useful property, $f(Q)$ is a submodular function of Q. The submodular function is defined by the following property.

Definition: Let V be a set of elements and $f: 2^{|V|} \to \Re$ a set function, then f is submodular if and only if

$$f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$$

$$\forall S \subset T \subset V, j \in V - T. \qquad (2)$$

3. Data correcting method

The mentioned property (2), submodularity of objective function, is used in data correcting method (DCM), proposed in (Benati, 2003) for the effective solution of the problem (1). This method is made up of three phases:

- Pre – processing phase,
- VNS heuristic,
- Binary search.

The purpose of the pre-processing phase used is the reduction of the size of the solved problem employing special inequalities. The essence of this algorithm is the branch and bound method. While the branching process is running, the current sub-problem is described by a pair of the sets S_0 and S_1, $S_0 \cap S_1 = \emptyset$. The set $S_0 \subset X$ contains variables that are fixed at value 1, $S_0 = \{ j \in X \mid x_j = 1\}$ and the set $S_1 \subset X$ contains variables that are free, $S_0 = \{ j \in X \mid x_j$ is free variable$\}$. The submodularity of the objective function allows constructing several rules for x_j variables fixations at value 1 or 0. The data correcting method employs the inequalities (3) and (4) for the variables fixation. If we have the node of a branching tree determined by the sets S_0, S_1, and $j \in S_1$, then the variable x_j can be fixed at value 0, if the following inequality holds:

$$f(S_0 \cup j) - f(S_0) \leq 0. \qquad (3)$$

Likewise, if the node of the branching tree is determined by a pair of sets S_0, S_1 and $j \in S_1$, then in the corresponding sub-tree the variable x_j can be fixed at value 1, if the inequality holds:

$$f(S_0 \cup S_1) - f(S_0 \cup S_1 - j) \geq 0. \qquad (4)$$

The variable fixation is processed at the beginning of algorithm before the branching phase and also consecutively in every node of branching tree. We used the simple exchange heuristic method VNS (variable neighbourhood search) to decrease the number of processed branches during the tree searching and to improve the starting solution. The VNS improves the randomly generated starting solution. This process uses the permissible operations and is not completed until the algorithm gets stuck in a local optimum. This procedure works as follows: consider all feasible solutions obtained by moving a shopping center to an unoccupied location, by adding one shopping center to an empty node, or by removing a location. The algorithm computes the changes in an objective function and accepts the most suitable one. If no improvement has been found, then a local optimum is determined. This procedure can be intensified by metaheuristics, as well.

We built four versions of the data correcting algorithm. The first one is the *DCM B*, here we used the basic branch and bound method without the VNS heuristic. The second version, DCM ExH, employs the simple exchange heuristics and the third DCM SA1 is the version with the exchange heuristics enhanced by metaheuristic simulated annealing with the first-fit search strategy. The last version DCM SA2 applies the metaheuristic simulated annealing, which uses the best-fit strategy for neighbourhood search.

The computation of the upper bound of the objective function for every branch can intensify the branch and bound method. According to (Benati, 2003), as the upper bound for branch S_0, S_1, can be used the value $f^U(S_0, S_1) = \min[f_1^U(S_0, S_1) \cdot f_2^U(S_0, S_1)]$, where:

$$f_1^U(S_0, S_1) = f(S_0) + \sum_{j \in S_1} (f(S_0 \cup j) - f(S_0)) \text{ and}$$

$$f_2^U(S_0, S_1) = f(S_0 \cup S_1) - \sum_{j \in S_1} (f(S_0 \cup S_1) - (f(S_0 \cup S_1 - j)).$$

We used the depth first strategy for searching the branching tree.

4. Simple heuristics

We adapted several simple heuristics to solve this problem. These heuristics allow evaluating the effectiveness of the exact methods. We used the exchange heuristic (ExH) and simulated annealing (SA), which were used in the data correcting method. They both employ the best-fit strategy for neighbourhood search. We modified the genetic algorithm (GA) (Kratica, Tošic, Filipovič, Ljubić, 2000), which was originally addressed to the Uncapacitated Facility Location Problem. The good computational efficiency of the algorithms was achieved by the parameter tuning.

5. Computational experiments

The algorithms were implemented in Delphi and were tested on the set of 120 benchmarks randomly generated from the Slovak road network. The used PC is a machine equipped with a Pentium IV processor with a 2400 MHz internal clock.

The benchmarks are organized in six groups according to their size 15×30, 20×50, 30×50, 40×100 a 60×200 a 60×2100. Each group encompasses 20 instances of the problem. The first number in the name of each benchmark group is the robustness of set $Y \cup X$ and the second represents the number of aggregated customers. The last class of benchmarks, 60×2100, has a privileged place because this size of the problem could be considered as a size of real problems in the Slovak Republic. As candidates for a new shopping center location are considered here almost all district towns. The values of all coefficients in the objective function were randomly generated from the pre-defined intervals. The coefficient β was set at the value $\beta = 0.195$.

The characteristics of all the mentioned heuristics can be seen in Figs. 1 and 2. Here we can see the total computational time (Fig. 1) and successfulness of heuristics in the optimal solution search (Fig. 2).

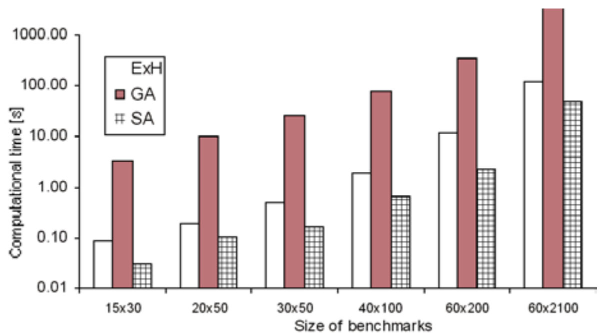

Fig. 1. Total computational time of heuristics

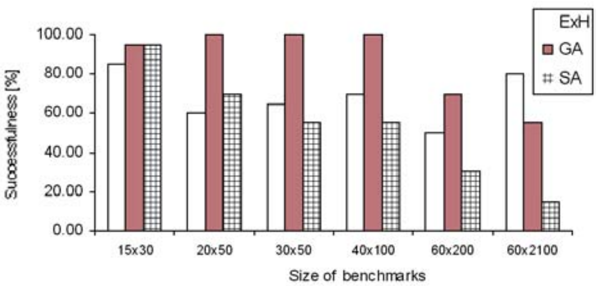

Fig. 2. Successfulness of heuristics

The total computational time for exact algorithms can be seen in table Tab. 1. The graphical representation is depicted in Fig. 3. Here, we can see the growing trend of the computational time and benefit, which was brought by using the VNS heuristic. The comparison of exact algorithms (Tab. 1) showed that the use of the simple exchange heuristics fits small-sized problems. The more sophisticated VNS heuristics (DCM SA1, DCM SA2) are more appropriate for large-sized problems.

Total computational time of exact algorithms Tab. 1

Size	DCM	DCM	DCM	DCM
15×3	0.01	0.01	0.03	0.65
20×5	0.35	0.33	0.37	2.96
30×5	0.58	0.55	0.65	5.99
40×10	21.24	20.24	20.60	40.56
60×20	13383.19	12578.75	29511.54	12043.71

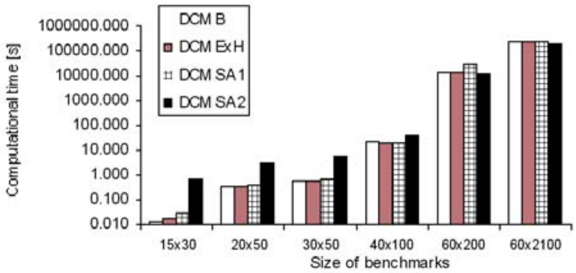

Fig. 3. Total computational time of exact algorithms

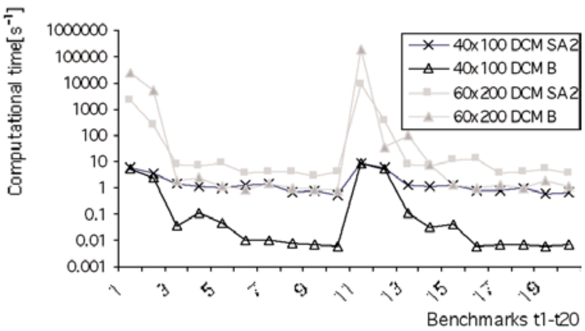

Fig. 4. Computational time of exact algorithms

As we mentioned, each class of benchmarks (t1–t20) was generated almost in the same way. The difference was only in the number of the already placed competitors. We divided the group of benchmarks belonging to one class (20 benchmarks) to two subgroups (t1–t10 and t11–t20, both 10 benchmarks). The number of competitors within each subgroup is distributed regularly. For the first benchmark the number of competitors is $|Y \cup X| / 10 - 1$ and for the tenth benchmark the number of competitors is equal to $|Y \cup X| - 1$. Thus, the first and eleventh benchmarks in each class have equal numbers of the placed competitors. As it can be observed in Tab. 2 and also in Fig. 4, the computational time of exact algorithm strongly depends on the number of competitors. The enormous computational time can be seen on the first benchmarks of each subgroup and then it rapidly goes down. The benchmarks t1, t2, t11 and t12 show the most adverse computational time in each group. The number of nodes where some competitors are placed is here less than 20 % of the all-possible shopping center

The computational time of exact algorithms Tab. 2

Benchmark	40 × 100		60 × 200		60 × 2100	
	DCM B	DCM SA2	DCM B	DCM SA2	DCM B	DCM SA2
t1	5.017	6.039	2235.66	12166.224	25923.924	19526.161
t2	2.409	3.613	270.471	255.388	5116.966	4958.842
t3	0.034	1.397	1.614	7.448	1.962	46.299
t4	0.110	1.182	0.241	7.128	2.536	149.956
t5	0.049	0.940	0.069	9.133	1.098	135.135
t6	0.009	1.340	0.045	3.651	0.892	44.451
t7	0.010	1.440	0.063	4.308	1.503	98.819
t8	0.008	0.712	0.028	4.133	0.982	46.392
t9	0.007	0.809	0.025	2.670	0.866	49.563
t10	0.006	0.520	0.026	3.969	0.790	52.970
t11	8.339	9.059	10485.07	9159.76	1198356.147	182900.021
t12	5.023	6.064	387.717	365.059	37.256	181.564
t13	0.116	1.296	1.722	7.894	97.224	234.092
t14	0.031	1.095	0.062	7.157	8.107	162.974
t15	0.041	1.268	0.101	10.720	1.296	88.031
t16	0.006	0.752	0.141	12.436	1.060	81.829
t17	0.006	0.764	0.030	3.383	1.273	96.511
t18	0.007	0.942	0.029	3.993	0.939	84.630
t19	0.006	0.620	0.032	5.540	1.989	87.633
t20	0.006	0.713	0.043	3.725	1.200	85.331

locations. This interesting phenomenon was observed for all exact algorithms and for all sizes of benchmarks.

6. Conclusions

The described experiments have shown that the present exact solving methods are able to compute the Competitive Uncapacitated Location Problem to optimality in acceptable computational time for instances of problems, which correspond to reality. This model and algorithms can be used as a tool for decision support and allows analysing the possibilities of the shopping malls chains extension. The notably long computational time was measured for problems with a small number of competitors, but fortunately this cases do not correspond to any practical application of the discussed model. The supermarkets or hypermarkets have been already placed almost in every district town. The district towns were considered as the candidates for further extension of the shopping mall chains.

The real operation costs of the shopping malls, purchasing power of the population and etc., were not at our disposal. We estimated the lower and upper bounds for each interval and all the input data were randomly generated from these intervals. This fact could have an impact on the obtained results.

This approach can be used as a complementary tool in a decision-making process. An expert using several feasible solutions and his own knowledge and experience can choose a final solution. The interactive system for decision support (Bednár, 2002) can be a good help for experts.

Acknowledgement
This work has been supported by the institutional grant FRI 9/2004

References

[1] BEDNÁR, P.: *Interactive system for cargo train routing.* In proceedings of the international symposium for PhD students and young researchers, Perner's Contact 2002, Pardubice 5. – 6. February 2002, p. 11
[2] BENATI, S.: *An improved branch & bound method for the uncapacitated competitive location problem.* Annals of Operations Research 122, 2003, p. 43–58,
[3] CENEK, P., JÁNOŠÍKOVÁ, Ľ.: *Transportation service model for regions.* Horizonty dopravy, No. 1, 2000, p. 10–12,
[4] GOLDENGORIN, B., TJISSEN, J., GHOSH, D., SIERKSMA, G.: *Solving the simple plant location problem using a data correcting approach.* Journal of Global Optimization 25, 2003, p. 377–406,
[5] KRATICA, J., TOŠIC, D., FILIPOVIČ, V., LJUBIĆ, I.: *Solving the plant location problem by genetic algorithm.* RAIRO Operations Research 35, 2001, p. 127–142.