

Matej Cebecauer - Lubos Buzna \*

---

## RE-AGGREGATION HEURISTICS FOR THE LARGE LOCATION PROBLEMS WITH LEXICOGRAPHIC MINIMAX OBJECTIVE

*We propose a new heuristic algorithm that provides solutions to the discrete lexicographic minimax location problem. The algorithm is applicable to large instances of the problem. The lexicographic minimax location problem is known to be NP-hard. Therefore, the large instances of the problem are not computable in reasonable time. An aggregation is a valuable tool that allows to adjust the size of the problem and approximate the problem by another one that can be solved. An inevitable consequence of aggregation is the loss of the precision. Typically, an aggregation method is used only once, in the initial phase of the solving process. Here, we propose iterative re-aggregating approach which adapts aggregated problem to achieve more precise location of facilities. To test the efficiency of the proposed method, we compute large location problems reaching 67 000 aggregated customers. The proposed approach is versatile and can be used for large range of location problems.*

**Keywords:** Location problem, heuristics, aggregation, lexicographic minimax.

### 1. Introduction

A location problem consists of finding a suitable set of facility locations from where services could be efficiently distributed to customers [1, 2 and 3]. Many location problems are known to be NP-hard. Consequently, the ability of algorithms to compute the optimal solution quickly decreases as the problem size is growing. There are two basic approaches how to deal with this difficulty. First approach is to use a heuristic method, which, however, does not guarantee that we find the optimal solution. Second approach is to use the aggregation that lowers the number of customers and candidate locations. The aggregated location problem (ALP) can be solved by exact methods or by heuristics. Aggregation, however, induces various types of errors. There is a strong stream of literature studying aggregation methods and corresponding errors [4 and 5]. Various sources of aggregation errors and approaches to minimise them are discussed by [4, 6 and 7].

Here, we are specifically interested in finding the fair design of a public service system that is serving spatially large geographical area with many customers. Customers are modelled by a set of demand points (DP) representing their spatial locations [5]. To include all possible locations of customers as DPs is often impossible and also unnecessary. In similar situations the aggregation is a valuable tool to obtain ALP of computable size.

It is well known that the solution provided by a heuristic method using more detailed data is often better than a solution

achieved by the exact method, when solving aggregated problem [8 and 9]. Often, aggregation is used only in the initial phase of the solving process, to match the problem size with the performance of the used solving method. In this paper, we propose simple re-aggregation heuristics, where the solved problem is modified to minimise the aggregation error in the following iterations. The aim of this article is to assess whether re-aggregation approach can be used for solving large location problem with lexicographic minimax objective. Our results show that the re-aggregation may provide better solutions than the solution obtained by the exact or heuristic method, when the aggregation is used only once on the large location problems.

The paper is organised as follows: section 2 briefly introduces the data processing procedure. In section 3 we summarise the lexicographic minimax problem. The re-aggregation heuristics is explained in section 4. Results of numerical experiments are reported in section 5. We conclude in section 6.

### 2. Data model

The OpenStreetMap (OSM) provides all necessary data to generate DP locations and to extract the road network. To estimate the position of demand points we use OSM layers describing positions of buildings, roads, residential, industrial and commercial areas. To generate DPs we use a simple procedure.

---

\* Matej Cebecauer, Lubos Buzna

Department of Mathematical Methods and Operations Research, Faculty of Management Science and Informatics, University of Zilina, Slovakia  
E-mail: matej.cebecauer@gmail.com

First, we generate a spatial grid which consists of uniform square cells with a size of 100 meters. For each cell, we extract elements from the OSM layers that are situated inside each cell. Second, DPs are located as centroids of the cells with a non-empty content. Third, generated DPs are connected to the road network and we compute the shortest paths distances between them. Finally, we calculate Voronoi diagrams, while using DP as generating points, and we associate with each DP a demand by intersecting Voronoi polygons with residential population grids produced by [10].

### 3. Problem formulation

We consider the set of potential locations of facilities  $I$  and the set of customers  $J$ . Each customer  $j \in J$  is characterised by a unique geographical position and by an integer weight  $b_j$ . The weight  $b_j$  represents the number of individual customers situated in the location  $j$ . The decisions to be made can be represented by a set of binary variables. The variable  $y_i$  equals to 1 if the location  $i \in I$  is used as a facility location and 0 otherwise. Allocation decisions are modelled by variables  $x_{ij}$  for  $i \in I$  and  $j \in J$ , where  $x_{ij} = 0$  if location  $i$  is not serving the customer  $j$  and otherwise. In order to obtain a feasible solution, the decision variables have to satisfy the following set of constraints:

$$\sum_{i \in I} y_i = p, \tag{1}$$

$$\sum_{i \in I} x_{ij} = 1 \text{ for all } j \in J \tag{2}$$

$$x_{ij} \leq y_i \quad \text{for all } i \in I, j \in J \tag{3}$$

$$x_{ij} \in \{0, 1\} \text{ for all } i \in I, j \in J \tag{4}$$

$$y_i \in \{0, 1\} \text{ for all } i \in I \tag{5}$$

where the equation (1) specifies that the number of located facilities equals to  $p$ . The constraints (2) make sure that each customer is assigned to exactly one facility, and the constraints (3) allow to assign a customer only to the located facilities. Following reference [11], we use the symbol  $Q$  to denote the set of all feasible location patterns, which satisfy the constraints (1)-(5).

By the symbol  $d_{ij}$  we denote the distance from customer  $j \in J$  to the potential facility location  $i \in I$ . We identify all unique distance values  $D_k$ ,  $k = 1, \dots, k_{max}$  in the set of all feasible distance values  $d_{ij}$ , where  $k_{max}$  is the number of unique  $d_{ij}$  values and we order the set of  $D_k$  values into the descending sequence. Thus  $D_1$  denotes the maximal possible distance between a customer and a facility. Each feasible solution in the set  $Q$  can be associated with a sequence of subsets  $[J_1, J_2, \dots, J_{k_{max}}]$  and with a vector  $[B_1, B_2, \dots, B_{k_{max}}]$ . Each customer  $j \in J$  is assigned to one of the subsets  $J_1, J_2, \dots, J_{k_{max}}$ . The distance between customers in the set  $J_k$

and the assigned facility is exactly  $D_k$ . The component  $B_k$  is a value defined as  $B_k = \sum_{j \in J_k} b_j$ , which denotes the number of individual customers situated in the subset  $J_k$ . If a set is empty, then the associated value  $B_k$  is zero.

The lexicographically minimal solution in the set is a solution that corresponds to the lexicographically minimal vector  $[B_1, B_2, \dots, B_{k_{max}}]$  [12].

To solve this problem we use the algorithm A-LEX [13] which similarly to the algorithm [11] solves optimisation problems in stages corresponding to individual distance values. Correctness and finiteness of the algorithm A-LEX, including the optimality conditions, are in more details analysed in reference [13]. Alternatively, there is also another fairness criterion known as a proportional fairness. Solutions provided with respect to this criterion tend to achieve higher system efficiency, for more details see reference [14].

### 4. Re-aggregation heuristics

In this section we describe our re-aggregation approach. The main goal is to re-aggregate solved problem in each iteration, to achieve more precise locations of facilities in the following iterations. Aggregation is an essential part of the heuristics and it leads to locations errors [4 and 5]. To minimise the effect of aggregation errors we need to understand the possible sources of errors. Therefore, we start by a brief summary of known sources of aggregation errors that are related to the input data. These sources of errors are denoted as  $A, B, C$  and  $D$  in the literature.

#### 4.1 Aggregation errors

Aggregation errors are caused by the loss of information when DPs are replaced by aggregated demand points (ADP). In [7] are introduced  $A, B$  and  $C$  source errors. Elimination of the source  $A$  and  $B$  errors was studied in [6]. Minimisation of the source  $C$  error was analysed in [15]. Source  $D$  error and possibilities how it can be minimised were studied in [16]. Source errors  $A$  and  $B$  can be eliminated by pre-processing the input data, while source errors  $C$  and  $D$  can be eliminated by post-processing the results. We summarise source errors in Table 1.

Some errors are also often made by designers or decision makers who are preparing the input data or evaluating the aggregation errors. Examples of such errors are: use of uniform demand distribution, aggregation method that ignores population clusters, or incorrect methods used to measure the aggregation error [4].

Types of source errors.

Table 1

Error type	Description	Elimination
source <i>A</i> error	This error is a result of wrongly estimated distance between ADPs <i>a</i> and <i>b</i> , when measuring the distance only between corresponding centroids.	Replace the distance by the sum of distances from all DPs aggregated in the ADP <i>a</i> to the centroid of ADP <i>b</i> .
source <i>B</i> error	It is a specific case of source <i>A</i> error. If ADP <i>a</i> is a candidate location for a facility, and at the same time it represents a customer, the distance between facility location <i>a</i> and customer <i>a</i> is incorrectly set to zero value.	Replace the zero distance by the sum of all distances from DPs aggregated in the ADP <i>a</i> to the centroid of the ADP <i>a</i> .
source <i>C</i> error	All DPs aggregated in the same ADP are assigned to the same facility.	Re-aggregate ADPs and find the closest facility for all DPs.
Source <i>D</i> error	It is consequence of establishing facilities in ADPs and not in DPs.	Find the facility location by disaggregating ADPs in the close neighbourhood of located facilities.

### 4.2 Aggregation method

To aggregate DP, we use a row-column aggregation method proposed by [8 and 17]. We slightly modified this approach by applying it to each individual administrative zone separately. This allows us to approximate population clusters more precisely and thus it helps to minimise the aggregation error.

### 4.3 Re-aggregation algorithm

In order to characterise the size of the aggregated problem, we define the relative reduction coefficient:

$$\alpha = \left(1 - \frac{\text{number of ADPs}}{\text{number of original DPs}}\right) 100\% \quad (6)$$

Thus, for the unaggregated problem we recover the value  $\alpha = 0$ .

Used notation and main parameters of the algorithm are described in Table 2.

Implementation of the algorithm A-LEX requires that the distance matrix has zero values on the diagonal [13]. This limitation does not allow us to fully eliminate source errors *A* and

*B* by pre-processing the input data. Therefore, we eliminate only source errors *C* and *D*. Re-aggregation algorithm is composed of phases that are executed in the following order:

**Phase 0: Initialisation:**

Set  $i=1$  and prepare ALP, of size corresponding to the reduction coefficient  $\alpha(1)$  by aggregating the input data and compute the distance matrix.

**Phase 1: Location of facilities:**

Solve ALP using *A-LEX* algorithm. As a result, we obtain  $p$  located facilities.

**Phase 2: Elimination of source *C* and *D* errors:**

- To minimise the source *C* errors reallocate DPs to the closest facilities.
- To minimise the source *D* errors decompose the problem into  $p$  location problems each consisting of one facility location and of all associated DPs. Solution of each problem is given by locating one facility that minimises maximal distance to the associated DPs. As a result we obtain  $p$  new locations of facilities.

**Phase 3: Re-aggregation:**

If every facility is established in a DP that cannot be further de-aggregated or if  $i > i_{max}$  then terminate. Output the best

Notations used in the description of the re-aggregation algorithm.

Table 2

Symbol	Description
$i$	Iteration counter
$\alpha(i)$	Value of the relative reduction coefficient corresponding to the solved ALP in the iteration $i$ . Hence, $\alpha(1)$ is the initial number of ADPs.
$\alpha_{max}$	Maximal allowed size of the solved problem.
$i_{max}$	Maximal number of iterations.
$\varepsilon$	Radius of ADP neighbourhood. This parameter divides the set $U$ of all ADPs into two subsets $E, F \subset U$ , where $E \cap F = \emptyset$ and $E \cup F = U$ . Subset $E$ includes all ADPs that are located from the closest facility at distance less than $\varepsilon$ . Subset $F$ is defined as $F = U - E$ . If the $\varepsilon = 0$ then in the set $E$ are situated only ADPs with established facility.
$\lambda$	Maximal number of newly created ADPs when de-aggregating an ADP.

found solution as the final result. Otherwise, considering the parameter  $\mathcal{E}$ , divide the set of ADPs into two subsets  $E$  and  $F$ . For each located facility  $f$  we define the maximum distance  $d_f^{max}$  as the maximum distance from all ADPs assigned to the facility  $f$ . Then we move from the subset  $F$  into the subset  $E$  all ADPs that include at least one DP that has distance to the closest facility  $f$  larger than  $d_f^{max}$ . Increment  $i$  by 1 and de-aggregate each ADP in the set  $E$  to at maximum  $\lambda$  new ADPs, using the aggregation method and update value  $\alpha(i)$ . If  $\alpha(i) > \alpha_{max}$  then terminate, otherwise go to phase 1.

### 5. Results

To evaluate the proposed heuristics, we analyse the optimality error and the computational time consumed by the heuristics, when it is applied to three real geographical areas. More details about geographical areas are given in Table 3.

Basic information about the selected geographical areas that constitute our benchmarks.

Table 3

Area	Number of DPs	Size [km <sup>2</sup> ]	Population
Partizanske	4 336	301	47 801
Kosice	8 472	240	235 251
Zilina	66 372	6 809	690 420

To quantify the importance of the elimination of the source errors in phase 2 we formulate two different versions, denoting them as V1 (including phases 0, 1 and 3) and V2 (including phases 0, 1, 2 and 3).

We start by investigating the performance of the re-aggregation algorithms using benchmarks Partizanske and Kosice that can be also directly solved by the algorithm A-LEX. To evaluate the efficiency of the proposed heuristics, when it is applied to extremely large problems, we use the benchmark Zilina. This benchmark is too large to be solved directly by the algorithm A-LEX.

#### 5.1 Performance analysis

We investigate the relation between the quality of the solution and the computational time that is consumed by individual phases of the re-aggregation algorithm by means of numerical experiments.

Because we need to compare the degree of fairness for two solutions, to evaluate an objective function value is not possible. Therefore, adopting the formulation described in [4], we define as a proxy measure the relative error  $\Delta_{MAX}$  comparing the equity of two solutions:

$$\Delta_{MAX}(x_\alpha, y) = \frac{f(y) - f(x_\alpha)}{f(x_\alpha)} 100\%, \tag{7}$$

where,  $f()$  is the value of the maximum distance from an unaggregated DPs to the closest facility;  $x_\alpha$  is the optimal solution of the ALP problem that corresponds to the relative reduction  $\alpha$  and  $y$  is the solution provided by our re-aggregation algorithm. Thus,  $x_0$  denotes the optimal solution of the original (unaggregated) problem. Further, we define the relative difference between two solutions as:

$$\Delta_{GINI}(x_\alpha, y) = \frac{g(y) - g(x_\alpha)}{g(x_\alpha)} 100\%, \tag{8}$$

where,  $g()$  is the value of gini coefficient computed by taking all distances from unaggregated DPs to their closest facility. The gini coefficient is commonly used in the economics to gauge the level of equity and [11] use it to evaluate fairness of the facility location. We denote the distance from the customer  $j$  to the closest facility  $j$  in the solution  $x$  as the  $u_j$  and the mean of the all  $u_j$  values as the  $\mu$ . For the set  $(u_j, j=1, 2, \dots, n)$ , the gini coefficient  $G$ , may be estimated by a sample mean

$$G = \frac{\sum_{k=1}^n \sum_{l=1}^n |u_k - u_l|}{2n^2 \mu}. \tag{9}$$

The gini coefficient is used as a measure of inequality, because a sample where the only non-zero value is  $\sum_{j=1}^n u_j$  has  $\mu = \sum_{j=1}^n u_j / n$  and hence  $G = (n - 1) / n \rightarrow 1$  as  $n \rightarrow \infty$ , whereas  $G = 0$  when all data points have the same value.

Finally, using similar notation, we define the relative time efficiency  $\delta$  as:

$$\delta(x_\alpha, y) = \frac{t(y) - t(x_\alpha)}{t(x_\alpha)} 100\%, \tag{10}$$

where  $t()$  is the time spent by computing the solution.

It is important to note that the aim of this paper is to test the performance of the re-aggregation heuristic and not to investigate in details the impact of the parameter values.

#### 5.2 Medium sized location problems

First, we start by evaluating the quality of solutions that are provided by our algorithm by comparing them with the solutions that are provided directly by the algorithm A-LEX. In Table 4, we present results where the initial size of ALP is  $\alpha(1) = 10\%$ . The parameter  $\alpha_{max}$  is fixed to the value of DPs, because the size of the problems is small and thus the limitation of the ALP size by parameter  $\alpha_{max}$  or use of parameter  $i_{max}$  is not necessary. We consider the value of the parameter  $\lambda = 4$  as the suitable to avoid the large grow of ALP size in the iterations. We investigate two values of parameter  $\mathcal{E}$ :  $\mathcal{E} = 0$  where only ADPs with established facility are de-aggregated and  $\mathcal{E} = 1$ , thus the

Results of numerical experiments for the geographical areas of Partizanske and Kosice where  $\lambda = 4$ ,  $\alpha(1) = 10\%$  of unaggregated DPs and here denote the reduction coefficient achieved in the last iteration.

Table 4

Area	$\varepsilon$ [km]		p=5		p=10		p=20	
			V1	V2	V1	V2	V1	V2
Partizanske	0	$\alpha$	86.65%	88.42%	86.28%	86.07%	80.24%	78.94%
		$\Delta_{MAX}(x_0, y)$	1.1%	1.1%	1.9%	0%	0%	0%
		$\Delta_{GINI}(x_0, y)$	3.3%	3.1%	1.3%	0.6%	-1.7%	-0.6%
		$\delta(x_0, y)$	-87.9%	-83.9%	-65.3%	-60.1%	-47.7%	-22.8%
	1	$\alpha$	80.35%	82.75%	69.03%	67.5%	45.74%	48.55%
		$\Delta_{MAX}(x_0, y)$	0%	0%	0%	0%	0%	0%
$\Delta_{GINI}(x_0, y)$		1.31%	0.45%	0.39%	1.3%	4.13%	0.05%	
Kosice	0	$\alpha$	89.00%	88.94%	87.59%	87.30%	84.10%	83.75%
		$\Delta_{MAX}(x_0, y)$	0%	0%	1.6%	0%	0%	0%
		$\Delta_{GINI}(x_0, y)$	-1.1%	-1.7%	-1.33%	-0.2%	-1.3%	1.2%
		$\delta(x_0, y)$	-91.6%	-85.7%	-91.4%	-89.3%	-65.1%	-57%
	1	$\alpha$	81.62%	82.74%	72.45%	73.91%	59.38%	62.29%
		$\Delta_{MAX}(x_0, y)$	0%	0%	0%	0%	0%	0%
$\Delta_{GINI}(x_0, y)$		-0.43%	-1.68%	0%	-1.14%	-2.27%	-2.51%	
		$\delta(x_0, y)$	-89.9%	-84.7%	-64.3%	-73.4%	8.82%	5.13%

area within the radius of 1 kilometre from each located facility is de-aggregated to  $\lambda$  new ADPs (see phase 3 of the algorithm).

In most of the cases, our heuristic algorithm finds the solution with the same maximum distance to the closest facility as is the solution  $x_0$  provided by the algorithm A-LEX on the unaggregated problem. Also the gini coefficient is similar to the solution  $x_0$ , thus the level of equity is also very similar. It is important to note that the reduction of the original problem is between 89% - 78%, which is significant value.

For the cases when our heuristic algorithm performs slightly worse than algorithm A-LEX ( $p = 10$  for Kosice and  $p = 5, 10$  for Partizanske). When  $\varepsilon = 1$  we found in all cases  $\Delta_{MAX}(x_0, y) = 0\%$ . Thus the parameter  $\varepsilon > 0$  has impact on the quality of the solution. The price for this extra operation is drop in the reduction coefficient to the value 82% - 45% and the computational time is now larger, but it is still smaller than in the case of the algorithm A-LEX, except for case Kosice and  $p=20$ . We observe that the time efficiency of the heuristic algorithm is more significant for smaller values of  $p$ .

Based on the results we can conclude that heuristics is able to provide equally fair solution using only 11%-25% DPs of the original problem and with better time efficiency.

In the next subsection, we present results obtained for large benchmarks derived from the problem instance Zilina. Here, in contrast to small problems, we compute the shortest path distances during the computational process while doing the re-aggregations. If we have large computer memory it is not necessary but it has to be done always, when the size of the

problem does not allow to store the complete distance matrix in the computer memory. This leads to larger computational times and makes impossible comparison of the computational time between small and larger problem instances.

### 5.3 Large location problem

Here, we compute the large location problem Zilina using the versions of the re-aggregation algorithm V1 and V2. In the experiments, we fixed the parameter  $\alpha_{max}$  to value that corresponds to the 85%,  $\lambda = 4$  and  $\varepsilon = 0$ . Furthermore, we do not apply the parameter  $i_{max}$ .

In these experiments, we also investigate how the solutions are improving and what is the computational time when we are increasing the number of iterations. The size of the Zilina problem does not allow us to compute the solution by using the algorithm A-LEX without aggregating the data. Therefore, instead of the solution  $x_0$ , we use in formulas (7) and (8) the solution provided by the algorithm A-LEX on the ALPs. We prepared three versions of problem Zilina corresponding to three values of reduction coefficient  $\alpha$ : 95%, 90% and 85% and we denote corresponding solutions as:  $x_{95}$ ,  $x_{90}$  and  $x_{85}$ . Here, we also used the initialization phase 0.

The results summarised in Table 5 confirm that versions V1 and V2 result in equitable service systems for the unaggregated problem Zilina. Version V1 can find with using only 1 368 ADPs a solution where the maximum distance of customer to the closest

Relative errors  $\Delta_{MAX}(x_{\alpha}, y)$  and  $\Delta_{GINI}(x_{\alpha}, y)$  obtained by our heuristics, considering  $x_{99}$ ,  $x_{95}$  and  $x_{90}$ ;  $i$  denotes the iteration step of the heuristic algorithm;  $\alpha(i)$  represents the reduction coefficient in the iteration  $i$  and  $t$  is the computational time in hours. Table 5

	i	$\alpha(i)$ [%]	Number of ADPs	t [h]	$\Delta_{MAX}(x_{\alpha}, y)$ [%]			$\Delta_{GINI}(x_{\alpha}, y)$ [%]		
					$x_{95}$	$x_{90}$	$x_{85}$	$x_{95}$	$x_{90}$	$x_{85}$
version V1	1	99.00	633	0.56	0.5	9.8	8.7	-0.2	-2.5	-1.7
	3	98.65	894	2.01	-0.3	4.1	3.1	3.3	0.8	1.6
	5	98.36	1 086	4.35	-2.1	2.4	1.4	1.2	-1.2	-0.4
	10	98.02	1 311	9.71	-7.4	-3.2	-4.1	2.1	-0.3	0.5
	15	97.94	1 368	15.61	-7.4	-3.2	-4.1	2.2	-0.2	0.6
version V2	1	99.00	663	25.19	-5.7	-1.4	-2.4	-1.9	-4.2	-3.4
	3	98.64	901	72.25	-5.1	-0.7	-1.7	1.2	-1.2	-0.4
	5	98.33	1 109	122.77	-6.7	-2.5	-3.5	0.1	-2.1	-1.3
	6	98.22	1 183	144.42	-6.7	-2.5	-3.5	-1.3	-3.6	-2.9
	7	98.14	1 237	166.08	-7.4	-3.2	-4.1	-1.1	-3.5	-2.7

facility is shorter by 2.2 kilometre compared to the solution that was obtained using 9 945 ADPs. In other words, it is possible to find about 4% better solutions for original unaggregated problem, by considering only 14% of the 9 945 customers and candidate locations.

The time complexity for version V2 is higher than for version V1. This is mainly due to the phase 2 where the distance matrices are computed. Version V2 provides equal solution than version V1 when it comes to the value  $\Delta_{MAX}$ , however, the value of the gini coefficient is significantly smaller.

## 6. Conclusions

When a location problem is too large to be solved by the available solving method, the aggregation is commonly used to lower the size of the problem. Typically, solving methods do not re-adjust the input data and the aggregation is done at the beginning of the process and it is kept separated from the solving method. In this paper, we proposed a method which adapts the granularity of input data in each iteration of the solving process to aggregate less in areas where locating facilities and in the neighbourhood of the most distant DPs. The proposed method is versatile and it can be used for a wide range of location problems.

To test the proposed method we used large real-world problems derived from the geographical areas that consist of many municipalities. It is worth noting that it is not very common to work with such large problems in the location analysis. We found only two examples in the literature where problems with approximately 80 000 DPs were solved [18 and 19]. In contrast to our study, these references used only randomly generated benchmarks and, in addition, simpler location problem than lexicographic minimax is solved.

Our heuristic algorithm provides solutions with smaller maximum distance (about 3%-7%) and better gini coefficient (about 1%-4%) with using only 3% of DPs in the case of a very large lexicographic minimax problem when it is compared to the classical method with a fixed aggregated problem with 5%, 10% and 15% of the DPs. This approach allows us to find high quality solutions of very large location problems, which are not computable to optimality.

We have shown that the re-aggregation approach may bring significant improvements of the solution also for a very large lexicographic minimax problem. Both versions of the algorithm V1 and V2 provide solutions with very good level of fairness. It is also important to underline that to our knowledge, there is no previous study exploring the role of the aggregation when solving the lexicographic minimax problem. We demonstrated that better understanding of aggregation errors, in this case, can help to provide more appropriated solutions with relatively large reduction coefficient  $\alpha$ .

It is also important to note that this is an initial study of the proposed approach and additional investigations using also other types of location problems will be done in the near future.

## Acknowledgements

This work was supported by the research grants VEGA 1/0339/13 Advanced microscopic modeling and complex data sources for designing spatially large public service systems and APVV-0760-11 Designing Fair Service Systems on Transportation Networks.

## References

- [1] DASKIN, M. S.: *Network and Discrete Location: Models, Algorithms and Applications*. John Wiley & Sons, 1995.
- [2] DREZNER, Z.: *Facility Location: A survey of Applications and Methods*. Springer Verlag, 1995.
- [3] EISELT, H. A., MARIANOV, V.: *Foundations of Location Analysis*. International Series in Operations Research and Management Science. Springer: Science + Business, 2011.
- [4] ERKUT, E., BOZKAYA, B.: Analysis of Aggregation Errors for the p-median Problem. *Computers & Operations Research*, 26(10-11):1075-1096, 1999.
- [5] FRANCIS, R., LOWE, T., RAYCO, M., TAMIR, A.: Aggregation Error for Location Models: Survey and Analysis. *Annals of Operations Research*, 167(1):171-208, 2009.
- [6] CURRENT, J. R., SCHILLING, D. A.: Elimination of Source A and B Errors in p-Median Location Problems. *Geographical Analysis*, 19(8):95-110, 1987.
- [7] HILLSMAN, E., RHODA, R.: Errors in Measuring Distances from Populations to Service Centers. *The Annals of Regional Science*, 12(3):74-88, 1978.
- [8] ANDERSSON, G., FRANCIS, R. L., NORMARK, T., RAYCO, M.: Aggregation Method Experimentation for Large-scale Network Location Problems. *Location Science*, 6(1-4):25-39.
- [9] HODGSON, M., HEWKO, J.: Aggregation and Surrogation Error in the p-Median Model. *Annals of Operations Research*, 123(1-4):53-66, 2003.
- [10] BATISTA, E., SILVA, F., GALLEGGO, J., LAVALLE, C.: A high-resolution Population Grid Map for Europe. *J. of Maps*, 9(1):16-28, 2013.
- [11] OGRYCZAK, W.: On the Lexicographic Minmax Approach to Location Problems. *European J. of Operational Research*, 100:566-585, 1997.
- [12] LUSS, H.: *Equitable Resources Allocation: Models, Algorithms and Applications*, John Wiley & Sons, ISBN 978-1-118-05468-0, 2012.
- [13] BUZNA, L., KOHANI, M., JANACEK, J.: An Approximation Algorithm for the Facility Location Problem with Lexicographic Minimax Objective. *J. of Applied Mathematics*, 2014.
- [14] BUZNA, L., KOHANI, M., JANACEK, J.: Proportionally Fairer Public Service Systems Design, *Communications - Scientific Letters of the University of Zilina*, 15(1):14-18, 2013
- [15] HODGSON, M. J., NEUMAN, S.: A GIS Approach to Eliminating Source C Aggregation Error in p-median Models. *Location Science*, 1:155-170, 1993.
- [16] HODGSON, M. J., SHMULEVITZ, F., KORKEL, M.: Aggregation Error Effects on the Discrete-space p-median Model: The case of Edmonton, *Canadian Geographer / Le Geographe Canadien*, 41(4):415-428, 1997.
- [17] FRANCIS, R. L., LOWE, T. J., RAYCO, M. B.: Row-Column Aggregation for Rectilinear Distance p-Median Problems. *Transportation Science*, 30(2):160-174, 1996.
- [18] AVELLA, P., BOCCIA, M., SALERNO, S., VASILYEV, I.: An Aggregation Heuristic for Large Scale p-median Problem. *Computers and Operations Research*, 39:1625-1632, 2012.
- [19] GARCIA, S., LABBE, M., MARIN, A.: Solving Large p-Median Problems with a Radius Formulation. *INFORMS - J. on Computing*, 23(4):546-556, 2011.