

Research Article

An Approximation Algorithm for the Facility Location Problem with Lexicographic Minimax Objective

Ľuboš Buzna, Michal Koháni, and Jaroslav Janáček

Department of Transportation Networks, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 01026 Žilina, Slovakia

Correspondence should be addressed to Ľuboš Buzna; buzna@frdsa.uniza.sk

Received 30 January 2014; Accepted 24 April 2014; Published 17 June 2014

Academic Editor: Włodzimierz Ogryczak

Copyright © 2014 Ľuboš Buzna et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a new approximation algorithm to the discrete facility location problem providing solutions that are close to the lexicographic minimax optimum. The lexicographic minimax optimum is a concept that allows to find equitable location of facilities serving a large number of customers. The algorithm is independent of general purpose solvers and instead uses algorithms originally designed to solve the p -median problem. By numerical experiments, we demonstrate that our algorithm allows increasing the size of solvable problems and provides high-quality solutions. The algorithm found an optimal solution for all tested instances where we could compare the results with the exact algorithm.

1. Introduction

Our study is motivated by problems faced by public authorities when locating facilities, such as schools, branch offices, or libraries and ambulance, police, or fire stations. To find an efficient spatial design of these systems, various types of location and allocation models can be used [1–4]. Typically, these problems are seen as an example of the resource allocation problem with a central planner. Thus, the costs of the system construction and its maintenance are typically “shared” by everybody, though not all contributors to the system (tax payers) are enjoying the same access to services. When pursuing an economically efficient design, some customers may end up being located close to the located service centres, whereas others are placed far away. Self-interested customers could understand such situation as unfair.

It is clear that in a real-world situation it is impossible to reach completely equal access to services for all customers, however, considering fairness criteria can help distribute the accessibility of services among customers more evenly. The number of existing problems related to the fair division of scarce resources is overwhelming. We restrict our short overview to few examples only. A recent comprehensive

overview of models, algorithms, and applications is available in [5].

In manufacturing, the omnipresent problem is how to allocate limited resources among many competing processes. Solving methods for problems with the knapsack inequality type of constraints, including multiple resource constraints, multiperiod problems, and problems with substitutable resources, are discussed in [6]. In engineering, one of the most profound applications of fairness is sharing of capacities in communication networks [7–9]. The method for solving the maximum flows [10] often generates unfair flows in the sense of how the flows are distributed to sink nodes. Using the principle of max-min fairness [11], the sink- (source-) optimal flow was defined as a flow which lexicographically maximizes the flow vector from the point of view of sink (source) nodes, and an efficient algorithm to find such flows was proposed in [12, 13]. The same fairness scheme was studied in the context of various routing mechanisms. An overview of basic problems in communication networks, associated with the applications of the max-min fairness to the flow rate allocation, routing, and load balancing, is given in [14]. For example, fixed paths were studied in regular networks finding analytical expressions describing

the network throughput [15]. For multicommodity flows, LP formulation of the problem was used to represent the flexible routing [16]. When mixed or integer decision problems are to be solved, the basic sequential procedure is not applicable. The ordered outcomes approach and the ordered values approach [17] allow to overcome this difficulty. Application of these two approaches to the bandwidth allocation problem was demonstrated in [18].

One important element appearing in studies focusing on fairness is the efficiency-fairness tradeoff. A well-known limitation of the lexicographically maximal flow is the relatively large reduction in the network throughput. A possible solution can be found in the optimization of flows with respect to the efficiency, provided that some level of fairness is guaranteed. An example of such approach is (α, β) fairness introduced in [19]. A general approach, combining optimization of the minisum and the minimax criteria with the tunable size of the applicability area for the minimax criterion, was proposed in [20]. To demonstrate the applicability of the proposed approach, the authors analysed the problem of how to distribute limited resources among patients to pay for the costly medical treatments. Alternatively, one can use different fairness schemes. One such scheme known as the proportional fairness was generalized and presented as an optimization problem [21]. The efficiency of the max-min fairness and proportional fairness was studied and analytically evaluated for a general set-up, using a simple measure [22], finding analytical expressions describing bounds for a gap between fair and efficient solutions. Using the generalized objective function which encompasses both these schemes as special cases, the same authors studied the efficiency-fairness trade-off, proposing several managerial prescriptions for the selection of the objective [23].

The first attempts to consider equity considerations when solving location problems on networks date back to the influential paper [24]. In this paper, the problem of finding the minimum number of locations was addressed, considering that no customer is farther from an existing location than a given distance. Since then, the inclusion of equity in location models has been a recurring topic. The equity is usually quantified by an equity measure. The taxonomy of equity measures proposed specifically for location problems was given in [25]. Here the authors decided to organize the equity measures around three dimensions: the choice of reference distribution which represents the desired goal, the distance metric which determines the way of how to assess the distance to the desired goal, and the scaling function used to take into account different importance of customers (e.g., by considering population, land area, demand, or income). In addition, the authors summarize from the literature useful criteria which should be taken into account when selecting an equity measure. Recently, an application of the equity measure to the equitable facility location problem in a plane was described in [26]. The authors analysed properties of the Gini coefficient and proposed an algorithm that finds the optimal location of a facility in a bounded area.

The requirement of equitable distribution comes often combined with other objectives. Noteworthy is the close relation between fairness criteria and multicriterial optimization.

Typically, the equity is either formally represented by one out of several criteria or the interest of each individual is represented as a single objective function [27]. For the survey of multiple criteria facility location problems, including those considering the equity, see the recent paper by Farahani et al. [28].

In the literature we do not find many attempts to suggest algorithms for solving facility location problems considering the lexicographic minimax objective. Specialized algorithms, considering minimax and lexicographic minimax optimization, were proposed when locating single facility in a plane [29]. Another approach, applicable to planar problems simultaneously optimizing the equitable distribution of distances by minimizing the radius of the serviced area and ensuring equitable distribution of loads, is described in [30]. The problem of how to locate facilities with equitable loads using the minimax criterion was also studied on networks [31]. The work [32] is closely related to the issue of lexicographic minimax optimization on networks. The authors noticed the possibility of reducing the classical minimax to minisum for 0-1 programming problems by transforming the coefficients in the objective function using a power function [32]. As pointed out in [33], the concept of power functions can be also extended to the lexicographic minimax solution concept. However, in general, high powers may be necessary to generate large enough differences between distances. When solving practical problems, large differences between distance coefficients may cause serious computational problems. To overcome these difficulties, the ordered outcomes approach and the ordered values approach were proposed in [17]. The latter approach was found as more efficient, but the size of tested instances was rather small. A very convenient technique for interactive analysis, where facilities are located with respect to the objective function taking into account lexicographic minimax combined with the minisum term, was proposed in [34]. The approach is based on the reference distribution method which can be controlled by manipulating few parameters and allows to take into account aspiration values of assigned distances defined by the user.

The lexicographic minimax optimization problem can be converted to a problem, where the k th term in the objective function is the number of occurrences of the k th worst possible unique outcome. The optimal solution is then found by minimizing the first term followed by the minimization of the second term without worsening the first term and so on [5]. The approach to the facility location problem based on this concept was developed in [33]. The initial computational experience with this approach shows that the lexicographic minimax approach, in comparison to the standard minimax, selects the locations characterized by remarkably smaller mean distance and absolute difference. Although the algorithm requires to run a large number of stages for the general distance matrix, it performs very well for the cases when the matrix contains only few distinct distance values. This makes the algorithm a very good choice when searching for an approximate solution or when only rough estimates of distances are considered.

Each of the above mentioned approaches to the lexicographic minimax optimization [17, 33, 34] results in a specific

form of the mathematical model that is supposed to be solved by a general purpose solver. Our initial experience with the algorithm [33], implemented on the state of the art solver XPRESS [35], indicated that we are able to solve problems up to 900 customers and 900 candidate facility locations, while restricting the distances to integers and measuring them in kilometres. This limitation might be too tight for some real-world applications and therefore it is of interest to elaborate algorithms which can provide high-quality solutions to larger problems. Building on the same basic concept of the unique classes of distances [33], we propose an approximation algorithm providing high-quality solutions for large instances of solved problems. We use the resulting algorithm to perform an extensive study using the well-known benchmarks [36–39], and we enriched them by new benchmarks derived from the real-world road network data [40, 41]. Our main contribution is the set of rules which allow us (i) to take into account the multiplicities assigned to different customers; (ii) to detect whether for a given distance active customers can reach higher, equal, or smaller distance to the closest located facility; and (iii) to use methods customized for solving the p -median problem. Customized methods can handle larger problems than up-to-date general purpose integer programming solvers. Therefore, the applicability of the algorithm is enhanced, especially, when it is used to solve large instances of real-world problems.

The remainder of the paper is organized as follows: the lexicographic minimax approach to the facility location problem is briefly described in Section 2.1. Section 2.2 presents our algorithm. In Section 3 the benefits of the algorithm are demonstrated on the set of benchmarks derived from the real-world networks. To conclude, we summarize our findings in Section 4.

2. Materials and Methods

When solving a facility location problem, the goal is to find suitable positions of facilities that provide services to the set of customers distributed in a serviced area. Along the years, many variants of the facility location problem have been elaborated. As an archetypical example, we describe the weighted p -median problem.

2.1. The Equitable Facility Location Problem. We consider the set I of potential locations for facilities and the set of locations J representing aggregate customers. Each aggregate customer $j \in J$ is characterized by a unique geographical position and we associate an integer weight (multiplicity) b_j with each position. The weight b_j represents the number of individual customers situated in the location j . We denote the set of all individual customers by \bar{J} . In order to map individual customers to aggregate customers we define the function $j(k)$ for $k \in \bar{J}$, returning the element $j \in J$ if and only if the individual customer k is situated in the location j (see Figure 1).

The decisions to be made can be represented by a set of binary variables. The variable y_i equals 1 if the location $i \in I$ is used as a facility location and equals 0 otherwise. Allocation

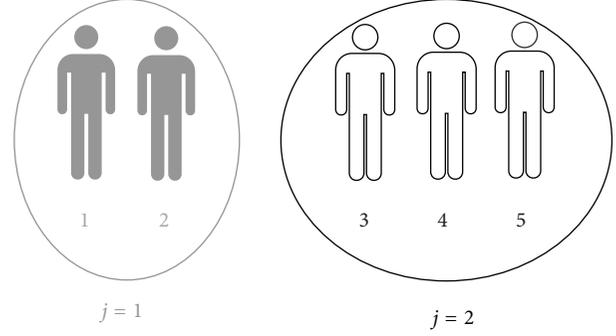


FIGURE 1: Schematic illustrating the definition of customers. The set of aggregate customers J is composed of two elements, that is, $J = \{1, 2\}$, and the set of individual customers is $\bar{J} = \{1, 2, 3, 4, 5\}$. The aggregate customer 1 includes individual customers 1 and 2, and the aggregate customer 2 stands for individual customers 3, 4, and 5. Thus, in this case the function $j(k)$, mapping the individual customer $k \in \bar{J}$ to the aggregate customer, returns these values $j(1) = 1$, $j(2) = 1$, $j(3) = 2$, $j(4) = 2$, and $j(5) = 2$, and the weights (multiplicities) assigned to aggregate customers are $b_1 = 2$ and $b_2 = 3$.

decisions are modelled by variables x_{ij} for $i \in I$ and $j \in J$, whereas $x_{ij} = 1$ if location i is serving the customer j and $x_{ij} = 0$ otherwise. In order to obtain a feasible solution, the decision variables have to satisfy the following set of constraints:

$$\sum_{i \in I} y_i = p, \quad (1)$$

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J, \quad (2)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J, \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J, \quad (4)$$

$$y_i \in \{0, 1\} \quad \forall i \in I, \quad (5)$$

where (1) specifies that the number of located facilities equals p . The constraints (2) make sure that each customer is assigned to exactly one facility, and the constraints (3) allow assigning a customer only to the located facilities. Following [33], we denote the set of all feasible location patterns, which satisfy the constraints (1)–(5), by the symbol Q . For each customer $j \in J$ we define the performance function $f_j(\mathbf{x})$ [27] where \mathbf{x} is a matrix representing the assignment decisions. This function estimates how customer j values the effect of located facilities. There is a strong stream of literature in location science studying measures that can be used to describe preferences of customers [42]. Typically, a measure called utility (disutility), u_{ij} , is defined to characterize preference of the customer j for the candidate location i . Utility is often defined as a function of the distance d_{ij} from the customer j to the facility location i . The performance function $f_j(\mathbf{x})$ is then a composite of utilities u_{ij} representing a model of the customer's choice behaviour [43]. The two most frequently used choice behavioural models are the minimum function, when the demand of a customer is assigned to one located

facility only (e.g., the closest one), and some kind of gravity model when the customer's demand is distributed among located facilities following a given mathematical prescription. Although the presented approach is able to handle the situation when u_{ij} is an arbitrary function of the distance d_{ij} , to simplify the text we choose $u_{ij} = d_{ij}$. The main focus of this paper is to extend the size of problems solvable by lexicographic approach, and, therefore, for simplicity reasons, we define the performance function that represents the distance from the customer to the closest located facility. Consequently, individual customers will get the same value of the performance function if they correspond to the same aggregate customer. Using constraints (2), the performance function for the individual customer $k \in \bar{J}$ can be formulated as

$$f_k(\mathbf{x}) = f_j(\mathbf{x}) = \sum_{i \in I} d_{i,j} x_{i,j}, \quad (6)$$

where $j = j(k)$ and d_{ij} is the distance from the aggregate customer $j \in J$ to the facility location $i \in I$.

The system optimum [22], frequently referred to as the minisum optimum or the utilitarian solution, corresponding to the p -median problem [1] is obtained when we minimize the expression (7) subject to $(\mathbf{x}, \mathbf{y}) \in Q$:

$$S(\mathbf{x}) = \sum_{j \in J} b_j f_j(\mathbf{x}) = \sum_{k \in \bar{J}} f_k(\mathbf{x}). \quad (7)$$

The corresponding optimal solutions we denote

$$(\mathbf{x}^{\text{SYS}}, \mathbf{y}^{\text{SYS}}) = \arg \min \{S(\mathbf{x}) \mid (\mathbf{x}, \mathbf{y}) \in Q\}. \quad (8)$$

The standard definition of the lexicographic minimax optimum [33] can be adjusted to the weighted problem, where the weights are representing the multiplicities of customers, as follows. We enlarge the set of aggregate customers J to the set of individual customers \bar{J} ; that is, each aggregate customer $j \in J$ is replaced by b_j individual customers situated in the same location. After this adjustment, we introduce the map $\Theta : R^{|\bar{J}|} \rightarrow R^{|\bar{J}|}$ which orders the values $f_k(\mathbf{x})$ for $k \in \bar{J}$ in a nonincreasing order. Thus, more formally,

$$\begin{aligned} \Theta(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{|\bar{J}|}(\mathbf{x})) \\ = (\bar{f}_1(\mathbf{x}), \bar{f}_2(\mathbf{x}), \dots, \bar{f}_{|\bar{J}|}(\mathbf{x})), \end{aligned} \quad (9)$$

if and only if there is the permutation τ such that $\bar{f}_k(\mathbf{x}) = f_{\tau(k)}(\mathbf{x})$ for all $k \in \bar{J}$ where $\bar{f}_1(\mathbf{x}) \geq \bar{f}_2(\mathbf{x}) \geq \dots \geq \bar{f}_{|\bar{J}|}(\mathbf{x})$. Let us denote

$$\begin{aligned} \mathbf{v} &= (\bar{f}_1(\mathbf{x}^1), \bar{f}_2(\mathbf{x}^1), \dots, \bar{f}_{|\bar{J}|}(\mathbf{x}^1)), \\ \mathbf{u} &= (\bar{f}_1(\mathbf{x}^2), \bar{f}_2(\mathbf{x}^2), \dots, \bar{f}_{|\bar{J}|}(\mathbf{x}^2)), \end{aligned} \quad (10)$$

where $(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2) \in Q$. We define the strong lexicographic inequality $<_{\text{LEX}}$ by $\mathbf{v} <_{\text{LEX}} \mathbf{u}$ if and only if there is an index $m \leq |\bar{J}|$ such that $v_n = u_n$ for all $n < m$ and $v_m < u_m$. The weak lexicographic inequality \leq_{LEX} is then defined as

$\mathbf{v} \leq_{\text{LEX}} \mathbf{u}$, if and only if either $\mathbf{v} = \mathbf{u}$ or $\mathbf{v} <_{\text{LEX}} \mathbf{u}$. Now we can define solution $(\mathbf{x}^{\text{LEX}}, \mathbf{y}^{\text{LEX}})$ to be the lexicographic minimax optimum if

$$\begin{aligned} \Theta(f_1(\mathbf{x}^{\text{LEX}}), f_2(\mathbf{x}^{\text{LEX}}), \dots, f_{|\bar{J}|}(\mathbf{x}^{\text{LEX}})) \\ \leq_{\text{LEX}} \Theta(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{|\bar{J}|}(\mathbf{x})), \end{aligned} \quad (11)$$

for all $(\mathbf{x}, \mathbf{y}) \in Q$.

We defined the lexicographic ordering and the lexicographic optimum using the set of individual customers. When we translate this definition to the set of aggregate customers, we can interpret the lexicographic optimization as two subsequent goals. Primarily, we are aiming to assign to facilities those customers whose performance function cannot be lowered any further; that is, the maximal distance from the relevant customers to the closest facility is minimized. Secondly, we minimize the number of relevant customers which are assigned to a facility at the considered stage. Please note that the number of individual customers, having assigned a distance value, is given by the sum of b_j values corresponding to those aggregate customers that are decided upon. In the next section we present the detailed description of the algorithm.

2.2. The Approximation Algorithm to the Facility Location Problem with the Lexicographic Minimax Objective. Similarly to the algorithm [33], our algorithm subsequently solves optimization problems corresponding to the distance values in stages. We order the set of all feasible distance values $d_{i,j}$ into the descending sequence of unique distance values D_k , for $k = 1, \dots, k_{\text{max}}$. At each stage $k > 1$ we consider a partitioning of the set J into the system of subsets $\{J_1, \dots, J_{k-1}, C_k\}$, where C_k is a set of active customers. We aim to identify the minimal subset of customers $J_k \subseteq C_k$, whose distance from the closest facility location equals the value D_k . We define the minimal subset as the set with the minimum number of individual customers, that is, the set where the sum of multiplicities $\sum_{j \in J_k} b_j$ is the smallest. For a given value of D_k , we find the minimal set J_k by solving the problem P_k :

$$\text{Minimize } g^k(\mathbf{x}) = \sum_{i \in I} \sum_{j \in \bar{J}} r_{ij}^k x_{ij} \quad (12)$$

$$\text{Subject to } (\mathbf{x}, \mathbf{y}) \in Q,$$

where r_{ij}^k are the costs defined for $j \in C_k$ and $i \in I$ in the following way:

$$r_{ij}^k = \begin{cases} 0, & \text{if } d_{ij} < D_k, \\ b_j, & \text{if } d_{ij} = D_k, \\ \left(1 + \sum_{u \in C_k} b_u\right), & \text{if } d_{ij} > D_k, \end{cases} \quad (13)$$

and for $j \in J_l$ where $l = 1, \dots, k-1$ and $i \in I$ according to the following prescription:

$$r_{ij}^k = \begin{cases} 0, & \text{if } d_{ij} \leq D_l, \\ \left(1 + \sum_{u \in C_k} b_u\right), & \text{otherwise.} \end{cases} \quad (14)$$

This setting of coefficients r_{ij}^k allows us to effectively distinguish three important situations, which can be directly used in the construction of the algorithm. Knowing the optimal solution $(\mathbf{x}^k, \mathbf{y}^k)$ of the problem P_k , the following implications denoted as cases (a), (b), and (c) can be derived.

- (a) If $g^k(\mathbf{x}^k) = 0$, then each customer $j \in C_k$ can be assigned to a facility whose distance from j is less than D_k , and each customer $j \in J_l$ for $l = 1, \dots, k-1$ can be assigned to a facility whose distance from j is less than or equal to D_l .
- (b) If $0 < g^k(\mathbf{x}^k) < 1 + \sum_{u \in C_k} b_u$, then each customer $j \in C_k$ can be assigned to a facility whose distance from j is less than or equal to D_k , and each customer $j \in J_l$ for $l = 1, \dots, k-1$ can be assigned to a facility whose distance from j is at most D_l . The minimal subset of customers $J_k \subseteq C_k$ whose distance from the closest facility locations equals the value D_k can be defined as $\{j \in C_k \mid \sum_{i \in I} r_{ij}^k x_{ij}^k = b_j\}$.
- (c) If $g^k(\mathbf{x}^k) > \sum_{u \in C_k} b_u$, then there exists either the customer $j \in C_k$, which is farther from the allocated facility than D_k or a customer in the subset J_l that is farther from the allocated facility than D_l . Thus, this case indicates nonexistence of a solution (\mathbf{x}, \mathbf{y}) to the problem P_k , for which $\sum_{i \in I} d_{ij} x_{ij} \leq D_l$ for $j \in J_l$, where $l = 1, \dots, k$.

We formulate the following algorithm, where we identify the customers whose distance from the closest facility location cannot be shorter than D_k , by embedding the problem P_k :

Algorithm A-LEX

Step 0. Initialize $k = 1$ and $C_1 = J$.

Step 1. Solve the problem P_k and denote the optimal solution by $(\mathbf{x}^k, \mathbf{y}^k)$.

Step 2. If $g^k(\mathbf{x}^k) = 0$, set $C_{k+1} = C_k$ and go to Step 4; otherwise if $(0 < g^k(\mathbf{x}^k) < 1 + \sum_{u \in C_k} b_u)$ go to Step 3.

Step 3. Set $J_k = \{j \in C_k \mid \sum_{i \in I} r_{ij}^k x_{ij}^k = b_j\}$; $C_{k+1} = C_k - J_k$.

Step 4. If $C_{k+1} = \emptyset$, then terminate and return $(\mathbf{x}^k, \mathbf{y}^k)$ as the solution; otherwise set $k = k + 1$ and continue with Step 1.

Correctness and finiteness of the algorithm A-LEX are justified by the following propositions.

Proposition 1. *The optimal solution of the problem P_k cannot satisfy the inequality $g^k(\mathbf{x}^k) > \sum_{u \in C_k} b_u$.*

Proof. For $k = 1$, the sum of all coefficients r_{ij}^1 does not exceed the value $\sum_{u \in C_1} b_u$ and thus $g^1(\mathbf{x}^1) \leq \sum_{u \in C_1} b_u$.

For $k > 1$, let us assume that the solution $(\mathbf{x}^{k-1}, \mathbf{y}^{k-1})$ of the problem P_{k-1} complies either with the case (a) or with the case (b) and $C_k \neq \emptyset$. The solution $(\mathbf{x}^{k-1}, \mathbf{y}^{k-1})$ assigns all customers from the set $C_k = C_{k-1} - J_{k-1}$ to facilities that are distant by at most D_k .

For $l = 1, \dots, k-1$, the customer $j \in J_l$ is assigned to facility at the distance D_l and therefore the following inequality $g^k(\mathbf{x}^{k-1}) < 1 + \sum_{u \in C_k} b_u$ is valid. Solution $(\mathbf{x}^k, \mathbf{y}^k)$, as a minimizer of the problem P_k , must fulfil inequality $g^k(\mathbf{x}^k) \leq g^k(\mathbf{x}^{k-1})$ and consequently $g^k(\mathbf{x}^k) \leq \sum_{u \in C_k} b_u$. \square

To assure the consistent termination of the algorithm A-LEX, the set C_k must be emptied for $k \leq k_{\max} + 1$.

Proposition 2. *If $C_{k_{\max}} \neq \emptyset$, then $C_{k_{\max}+1} = \emptyset$.*

Proof. If $C_{k_{\max}} \neq \emptyset$, then the solution $(\mathbf{x}^{k_{\max}}, \mathbf{y}^{k_{\max}})$ assigns all customers in the set $C_{k_{\max}}$ to facilities that are at the distance $D_{k_{\max}}$. As $D_{k_{\max}}$ is the minimal distance value, $J_{k_{\max}} = C_{k_{\max}}$ and thus $C_{k_{\max}+1} = \emptyset$. \square

To investigate under what circumstances the algorithm A-LEX provides the optimal solution, we start by noting that each feasible solution in the set Q is associated with a sequence of sets $[J_1, J_2, \dots, J_{k_{\max}}]$ and with a vector $[B_1, B_2, \dots, B_{k_{\max}}]$. The distance between customers in the set J_k and the assigned facility is exactly D_k . The component B_k is a number defined as $B_k = \sum_{j \in J_k} b_j$. If a set J_k is empty, then the associated value B_k is zero. The lexicographically minimal solution in the set Q corresponds to the lexicographically minimal vector $[B_1, B_2, \dots, B_{k_{\max}}]$ [5].

Proposition 3. *If algorithm A-LEX does not find the lexicographically minimal solution, then there must exist a problem P_k having at least two optimal solutions $(\mathbf{x}^A, \mathbf{y}^A)$ and $(\mathbf{x}^*, \mathbf{y}^*)$ associated with two different sets J_k^A and J_k^* , respectively.*

Proof. Let us consider that the algorithm A-LEX found solution $(\mathbf{x}^A, \mathbf{y}^A)$, which is not lexicographically optimal; then the optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$ induces vector $[B_1^*, B_2^*, \dots, B_{k_{\max}}^*]$ that is lexicographically smaller than the vector $[B_1^A, B_2^A, \dots, B_{k_{\max}}^A]$. Let us denote by k_0 the smallest subscript, for which the inequality $B_{k_0}^* < B_{k_0}^A$ holds.

Assuming that no pair $[J_k^*, J_k^A]$ consists of different sets for $k = 1, \dots, k_0 - 1$, then we obtain $C_{k_0}^A = J - \bigcup_{k=1}^{k_0-1} J_k^A = J - \bigcup_{k=1}^{k_0-1} J_k^* = C_{k_0}^*$.

As A-LEX solves P_{k_0} to optimality, the assumption of identity for J_k^A and J_k^* for $k = 1, \dots, k_0 - 1$ contradicts the inequality $B_{k_0}^* < B_{k_0}^A$. \square

Corollary 4. *If there is only one optimal solution of the problem P_k concerning variables \mathbf{x}^k at each stage of the algorithm, then the solution provided by the algorithm A-LEX is lexicographically optimal.*

It is important to note that the algorithm A-LEX yields an optimal solution with only minor exceptions concerning the ambiguous determination of the set J_k . A necessary condition for obtaining an approximate solution is that at the stage k there exist at least two optimal solutions of the problem P_k . Thus, there exist at least two different sets of customers J_k^* and J_k^A such that $\sum_{j \in J_k^*} b_j = \sum_{j \in J_k^A} b_j$, and each set of customers is assigned the distance D_k . Our algorithm is able to identify only one of these sets and it treats all such sets as being equivalent. An approximation error may arise when the algorithm chooses the set, which in the future steps leads to the distribution of distances that is lexicographically less efficient than the distribution that would be reached, when continuing with the alternative set of customers. The size of the approximation error depends on how likely the multiple solutions of the problem P_k are and whether multiple solutions have the tendency to appear in the early stages of the algorithm or close to the end. Thus, the resulting error depends on the distribution of b_j and d_{ij} values. This drawback is compensated by several advantages, which enable speeding up significantly the algorithm and thus enable solving larger problems. As the sequence of problems P_k keeps the same size and structure of constraints, we can easily replace the general purpose IP solver by an algorithm customized for the p -median problem.

When processing D_k values in the descending order one by one, we may observe that sets J_k are empty for a range of k values, especially at the beginning of the solving process (for large D_k values). Such “empty” iterations do not have any impact on the resulting solutions and could be skipped. As the processing of the problem P_k allows us to identify situations in which there are customers that require to be assigned larger distance value than D_k , values D_k do not have to be processed sequentially. We can use various searching schemes (e.g., bisection method) to find the next value D_k where $J_k \neq \emptyset$. This can help reduce the number of times the problem P_k is solved and thus reduce the overall computational time.

To initialize the algorithm, we do not need to consider the entire sequence of values D_k for $k = 1, \dots, k_{\max}$. Instead, we can start the solving process by finding the optimum corresponding to the objective function (7). Then it is enough to initialize the variable k in Step 0 to the value m that is determined by the equality $D_m = \max\{d_{ij} \mid x_{ij}^{\text{SYS}} = 1\}$. This typically allows to skip many processing steps when no customer is assigned a distance value. Another useful operation (for $k > 1$) is to check whether $g^k(\mathbf{x}^{k-1}) = 0$ in Step 1 before the problem P_k is solved. If that is the case, we can use \mathbf{x}^{k-1} as the optimal solution of the problem P_k [33].

3. Results and Discussion

To study the efficiency of the algorithm A-LEX, we performed a computational study. Our two main goals were to evaluate the quality of the solutions by comparing them to the exact algorithm [33] (algorithm O-LEX hereafter) and to test the limits of the algorithm regarding the size of the solvable problems. The algorithm O-LEX adds new constraints to

the solved problem. Consequently, the use of algorithms customized for the p -median problem is impossible. Therefore we used a general purpose IP solver to implement it. To be able to compare the algorithm A-LEX with the algorithm O-LEX, we implemented them in the Xpress-Mosel language (version 3.4.0) and we executed them using the Xpress-Optimizer (version 23.01.05) [35]. To keep both algorithms comparable, we did not use any searching scheme to process values D_k in the algorithm A-LEX, and thus we processed all values D_k sequentially. To explore the properties of the algorithm A-LEX beyond the limits of the general purpose integer solvers, we implemented the algorithm A-LEX in the Microsoft Visual C++ 2010. To solve the problem (12) we used the exact algorithm ZEBRA, the state of the art solver for the p -median problem [44]. The implementation of the algorithm ZEBRA is publicly available on the author’s web page [45]. To distinguish both versions of the algorithm A-LEX, we denote the version based on the Xpress by A-LEX^X and the version which uses the algorithm ZEBRA by A-LEX^Z.

The computational study was carried out on an Intel (R) Core TM i7-3610 QM CPU with four 2.3 GHz cores each composed of two threads (although C++ code used just one thread) and 8 GB RAM.

3.1. Benchmarks. Three sets of testing problems organized by the size were used to perform the computational study. In all cases, customers’ sites are considered to be also possible facility locations; that is, the sets I and J are identical. As there are no standard test problems for the facility location problem with the lexicographic minimax objective, we used the problems originally proposed for the capacitated p -median problem while interpreting the demands as b_j values (multiplicities of customers). Twenty small instances *pmedcap1–pmedcap20* were taken from the OR-library [36]. We also included into this set the smallest testing problem *SJC1* used in [37]. Three larger problems taken from the same source, *SJC2*, *SJC3*, and *SJC4*, together with two instances derived from the network of 737 Spanish cities [38] constitute the medium-sized instances. The largest test problems include the problem *p3038* originally proposed for the TSP [39] and later adjusted to the capacitated p -median problem [37]. Furthermore, considering the population data as b_j values, we created large-sized benchmarks from the road network of the Slovak Republic [40] and the road network of six southeastern US states [41] (for more details see the caption in Figure 2). All the benchmarks and the source code are available for download as a supplemental material (<http://frdsa.uniza.sk/~buzna/>).

3.2. Numerical Experiments. We summarized the computational results in Tables 1, 2, and 3. The abbreviations in Tables 1–3 have the following meanings:

- instance: the problem name,
- $|I|$: number of facility/customer locations,
- p : number of facilities to be located,
- k_{\max} : number of D_k values,

TABLE 1: Computational results for the algorithm A-LEX: small instances.

Instance	$ I $	p	k_{\max}	O-LEX		A-LEX ^x		Δ
				Time [s]	k_s	Time [s]	k_s	
<i>pmedcap1</i>	50	5	30	1.88	11	0.64	11	0
<i>pmedcap2</i>	50	5	42	3.23	8	0.67	7	0
<i>pmedcap3</i>	50	5	28	1.60	9	0.46	7	0
<i>pmedcap4</i>	50	5	43	2.97	15	0.91	15	0
<i>pmedcap5</i>	50	5	33	2.11	13	0.74	10	0
<i>pmedcap6</i>	50	5	38	2.51	11	1.21	12	0
<i>pmedcap7</i>	50	5	40	3.15	13	0.98	13	0
<i>pmedcap8</i>	50	5	39	2.75	7	0.80	6	0
<i>pmedcap9</i>	50	5	36	2.80	9	0.80	8	0
<i>pmedcap10</i>	100	10	30	1.98	10	0.59	10	0
<i>pmedcap11</i>	100	10	21	4.29	2	1.04	1	0
<i>pmedcap12</i>	100	10	26	5.21	4	1.74	3	0
<i>pmedcap13</i>	100	10	30	5.65	2	1.76	1	0
<i>pmedcap14</i>	100	10	24	4.89	3	1.91	2	0
<i>pmedcap15</i>	100	10	27	6.21	2	1.89	1	0
<i>pmedcap16</i>	100	10	23	4.59	1	1.20	0	0
<i>pmedcap17</i>	100	10	25	6.83	2	1.50	1	2
<i>pmedcap18</i>	100	10	25	4.67	2	1.70	1	0
<i>pmedcap19</i>	100	10	30	6.83	2	2.46	1	0
<i>pmedcap20</i>	100	10	20	5.95	3	0.90	2	0
<i>SJC1</i>	100	5	732	47.90	588	28.54	569	0
<i>SJC1</i>	100	10	426	22.99	283	10.82	282	0
<i>SJC1</i>	100	15	427	24.51	261	10.65	269	0
<i>SJC1</i>	100	20	150	5.31	64	2.91	65	0

TABLE 2: Computational results for the algorithm A-LEX: medium instances.

Instance	$ I $	p	k_{\max}	O-LEX		A-LEX ^x		Δ
				Time [s]	k_s	Time [s]	k_s	
<i>SJC2</i>	200	10	426	131.4	238	50.9	237	0
<i>SJC2</i>	200	20	306	64.4	132	37.4	128	0
<i>SJC2</i>	200	30	218	32.2	79	17.4	71	0
<i>SJC2</i>	200	40	169	20.3	39	9.7	40	0
<i>SJC3</i>	300	15	445	461.6	207	357.7	189	0
<i>SJC3</i>	300	30	267	145.1	70	68.8	58	0
<i>SJC3</i>	300	45	226	71.1	46	37.9	41	0
<i>SJC3</i>	300	60	215	53.3	50	29.8	42	0
<i>SJC4</i>	402	20	461	1371.2	161	1205.8	140	0
<i>SJC4</i>	402	40	342	1207.5	74	1052.5	58	0
<i>SJC4</i>	402	60	229	158.7	33	87.2	29	0
<i>SJC4</i>	402	80	193	144.9	25	56.2	24	0
<i>Spain_737_1</i>	737	37	467	116838.0	92	81185.1	65	0
<i>Spain_737_1</i>	737	50	348	196000.0	53	27296.2	49	0
<i>Spain_737_1</i>	737	185	108	12367.4	5	279.5	5	0
<i>Spain_737_1</i>	737	259	59	430.4	2	32.2	2	0
<i>Spain_737_2</i>	737	37	467	35590.7	88	29185.6	65	0
<i>Spain_737_2</i>	737	50	348	64005.7	59	27806.1	38	0
<i>Spain_737_2</i>	737	185	108	3182.3	5	232.4	5	0
<i>Spain_737_2</i>	737	259	59	72.5	1	43.2	1	2

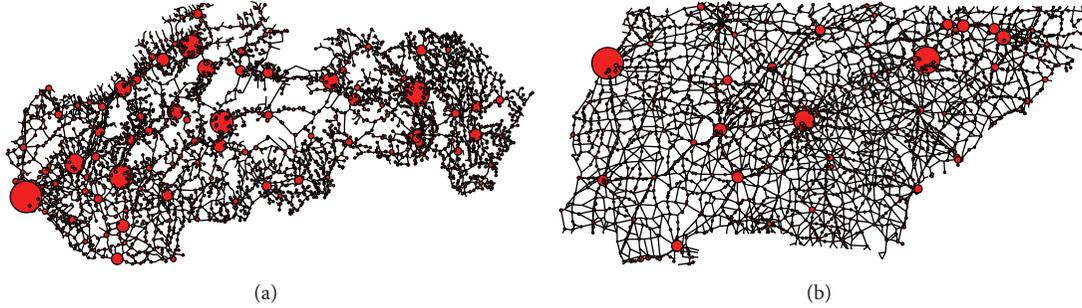


FIGURE 2: Road networks used to test the algorithm A-LEX. All inhabited nodes (marked red) are considered as aggregate customers and possible locations of facilities. The diameter of nodes is scaled proportionally to the population. (a) The road network of the Slovak Republic consists of $|I| = 2928$ aggregate customers representing the locations of $|\bar{I}| = 5\,384\,909$ individuals. (b) The joined road network of six southeastern US states (Tennessee, North Carolina, South Carolina, Georgia, Alabama, and Mississippi). All inhabitants included in the dataset are considered to be customers ($|I| = 2398$, $|\bar{I}| = 14\,830\,101$).

TABLE 3: Computational results for the algorithm A-LEX: large instances.

Instance	$ I $	p	k_{\max}	A-LEX ^Z	
				Time [s]	k_s
<i>p3038</i>	3038	2500	33	2520.8	1
<i>p3038</i>	3038	2000	35	4204.3	2
<i>p3038</i>	3038	1500	38	8915.9	2
<i>p3038</i>	3038	900	73	190092.9	7
<i>p3038</i>	3038	700	110	17902.9	8
<i>p3038</i>	3038	100	*	*	*
<i>p3038</i>	3038	50	*	*	*
<i>p3038</i>	3038	10	1188	201165.3	36
<i>SR</i>	2928	2500	3	786.8	0
<i>SR</i>	2928	2000	4	1021.6	0
<i>SR</i>	2928	1500	5	1083.4	0
<i>SR</i>	2928	1000	8	1622.9	0
<i>SR</i>	2928	900	9	1988.5	0
<i>SR</i>	2928	700	11	2954.2	0
<i>SR</i>	2928	100	48	9624.7	0
<i>SR</i>	2928	50	65	10509.8	0
<i>SR</i>	2928	10	92	11888.4	0
<i>US</i>	2398	2000	10	1006.6	0
<i>US</i>	2398	1500	15	1203.7	0
<i>US</i>	2398	1000	22	1702.7	0
<i>US</i>	2398	900	25	1694.7	0
<i>US</i>	2398	700	33	11022.5	0
<i>US</i>	2398	100	*	*	*
<i>US</i>	2398	50	*	*	*
<i>US</i>	2398	10	219	9038.2	0

Table cells filled with the symbol “*” indicate the instances when the algorithm did not terminate within 3 days.

time: CPU time (in seconds) used to solve the problem,

k_s : number of skipped k values for which we did not solve the problem P_k because $g^k(\mathbf{x}^{k-1}) = 0$,

Δ : number of facilities that the algorithms placed differently, calculated using the formula:

$$\Delta = \sum_{i \in I} \left| y_i^{\text{O-LEX}} - y_i^{\text{A-LEX}^X} \right|. \quad (15)$$

Due to problems with the computer memory, XPRESS solver was not able to solve large instances successfully. Therefore, in Table 3 we show the results obtained by the solver ZEBRA only. Comparison of results reveals that the algorithm A-LEX^X outperforms the algorithm O-LEX on all tested instances. A-LEX^X computed all small instances in time which accounts for 42.5% and all medium instances for 47.6% of the time needed by the algorithm O-LEX. In order to compare the quality of the solution, we evaluate

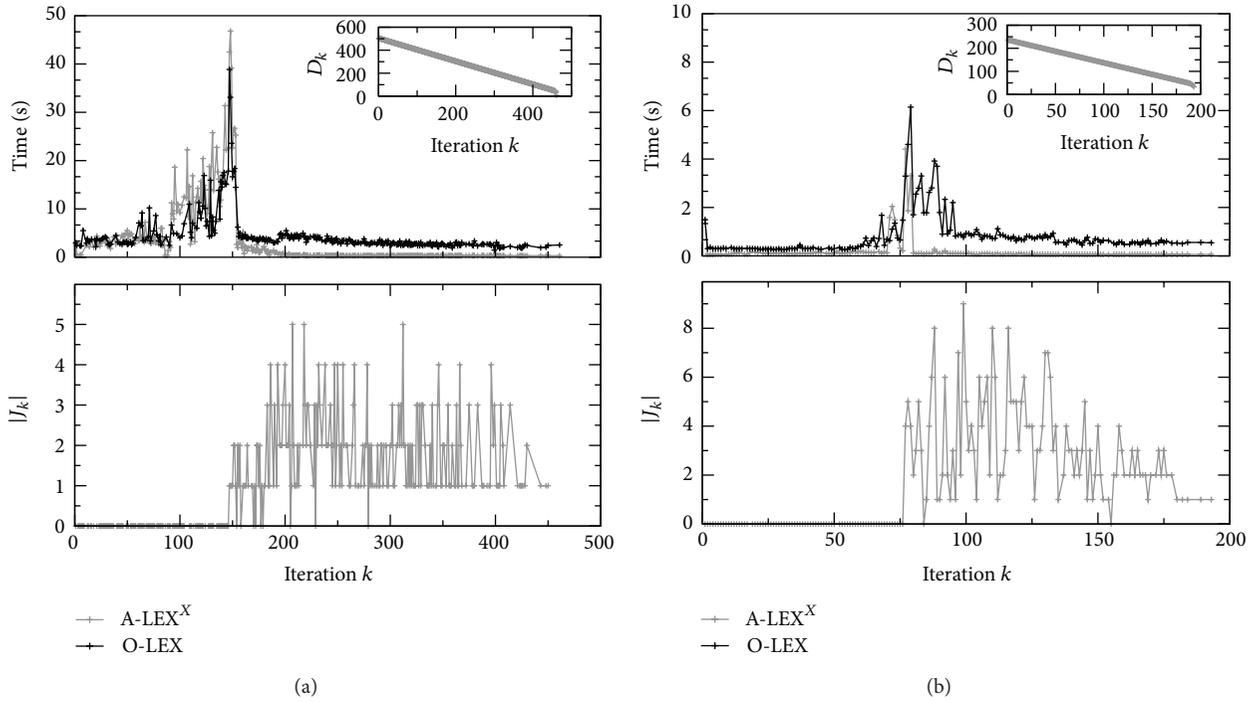


FIGURE 3: Computational time in seconds and the number of fixed aggregate customers $|J_k|$ as a function of the iteration step k . The results were obtained for the benchmark *SJC4* in (a) $p = 20$ and in (b) $p = 80$. The values $|J_k|$ are identical for both algorithms, and the datasets are overlapping.

the Manhattan distance (15) between the location vectors y . Occasionally, we find small differences in the location of facilities. However, we checked and we found that customers have the same distance to the closest located facility in all tested instances. Thus, the algorithm A-LEX found an optimal solution in all cases where we were able to compare it with the algorithm O-LEX.

To explain why the algorithm A-LEX^X is faster than the algorithm O-LEX, we need to take more detailed view on both algorithms. In the algorithm A-LEX^X, the structure of constraints in P_k is independent on k and only the structure of coefficients r_{ij}^k varies with the step k . These coefficients may take three possible values for a given customer (see the expression (13)). On the contrary, the algorithm O-LEX has simpler structure of the objective function but a new constraint is added at each iteration. New constraints make sure that the objective function values reached in the previous iterations cannot deteriorate in the following iterations [33]. Thus, the algorithm A-LEX^X outperforms O-LEX if IP solver can handle more easily the constant set of constraints with a more complex objective function and, vice versa, the algorithm O-LEX is faster if the IP solver can process faster simpler structure of the objective function with the growing set of constraints.

In Figure 3, we show the computational time needed to solve the individual instances of the problem P_k . For small values of k , where no customers are fixed (i.e., $J_k = \emptyset$), both algorithms perform comparably well. For larger values of k , the algorithm A-LEX^X systematically outperforms

the algorithm O-LEX. To gain more insights into the solving process, we plotted in Figure 4 the number of branch-and-bound nodes processed by the IP solver and the number of simplex iterations required to solve the optimization problem in the root node of the branch-and-bound method. We found that the algorithm A-LEX^X needs a much smaller number of simplex iterations than the algorithm O-LEX^X. We conjecture that the number of simplex iterations is smaller in the algorithm A-LEX^X because the large values of coefficients r_{ij}^k for $d_{ij} > D_k$ allow excluding the variables taking zero values in the optimal solution faster from the basic feasible solution.

In Figure 3, we can also see that few iterations preceding and few iterations following the value k , where the first customer is fixed, take the largest portion of the computational time. This increase can be explained by the need to search through a larger number of nodes in the searching tree before the optimal integer solution is found (see Figure 4). From this point of view it could be beneficial to use a searching scheme in the algorithm A-LEX to find the value of k when the first customer is fixed and then to continue by incrementing k sequentially. This could reduce the number of iterations and could avoid processing some time-demanding iterations. In Figures 3 and 4, we showed the results obtained for the selected benchmark *SJC4*. However, it should be noted that we found qualitatively similar results with all other benchmarks.

As the algorithm A-LEX preserves the problem P_k in the form of the p -median problem, we can replace the general

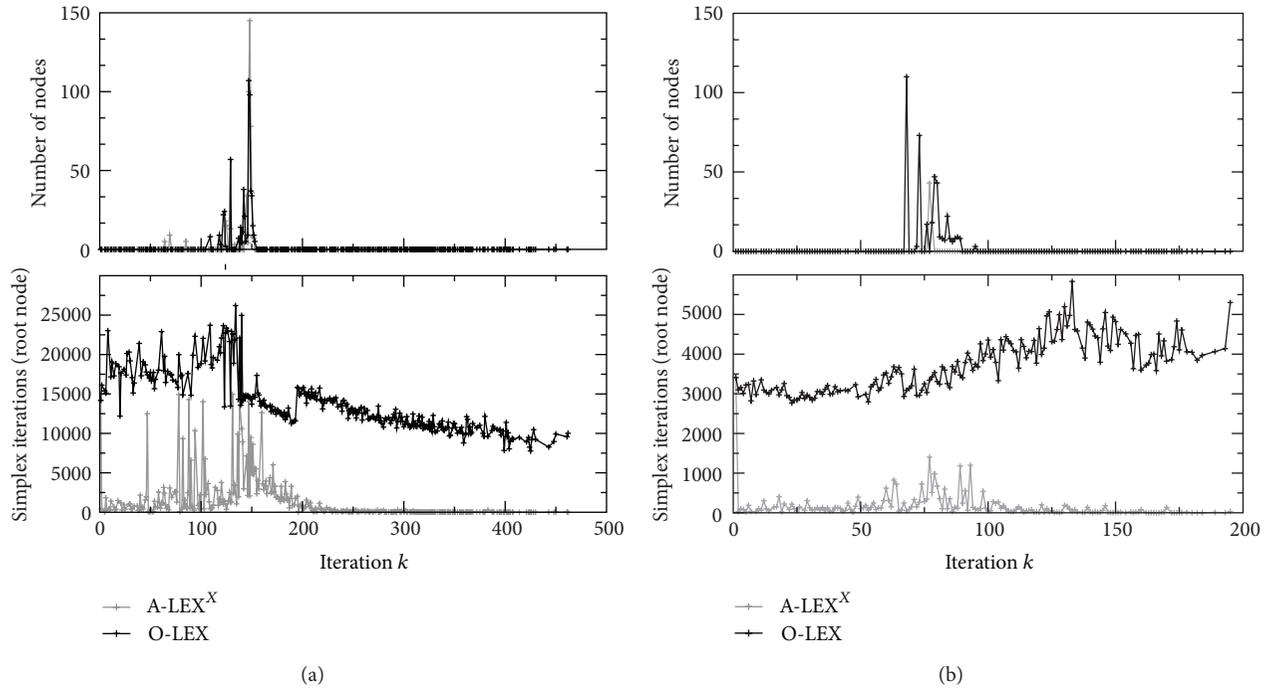


FIGURE 4: The number of nodes processed by the branch-and-bound method and the number of simplex iterations needed to process the root node as a function of the iteration step k . The results were obtained for the benchmark *SJC4* in (a) $p = 20$ and in (b) $p = 80$.

purpose IP solver by the customized p -median solver. To solve the p -median problem, we used the algorithm ZEBRA [44]. The size of instances in Table 3 approaches the size of large p -median instances used to test the performance of the algorithm ZEBRA [44]. Due to the problems with the computer memory, the algorithms O-LEX and A-LEX^X could not be used to solve this set of problems. Facility location problems are strategic planning problems and the computational time is not necessarily a core issue. Therefore we set the time limit for all computations to 3 days. The problems, where p is large, can be solved relatively fast. The computational time rapidly grows by lowering the value p . We were not able to solve the problems *p3038* and *US* for the intermediate p values (50 and 100) within the time limit of 3 days. Please note that similar limitations of the algorithm ZEBRA are also reported in [44].

4. Conclusions

The algorithm A-LEX, proposed in this paper, preserves the structure of constraints in the form of the p -median problem, which allows solving larger instances of problems than can be solved by the algorithm O-LEX. Moreover, the computational experiments showed that the algorithm A-LEX provides high-quality solutions. Therefore, it can be concluded that A-LEX is competitive with the existing state of the art algorithm O-LEX for solving the facility location problem with the lexicographic minimax objective function.

The proposed approximation approach is also applicable to other types of similar combinatorial optimization problems with lexicographic minimax objective. Values assigned

to individual customers at different stages of the algorithm need to be included in the objective function so that we can construct rules which allow detecting whether customers can be assigned equal, smaller, or larger value than the outcome value tested on a given stage of the algorithm. To gain a computational advantage compared to all purpose IP solvers, it is needed that, for a given problem, we can use a customized exact algorithm to find the system optimum. Concrete example where this approach could be used is the maximum generalized assignment problem.

We compared the approximation algorithm A-LEX to the exact algorithm O-LEX. We are aware of the fact that it is not a standard practice to compare exact algorithms with heuristics in terms of the computational time. However, we believe that this decision can be well justified by the similarity of both algorithms, and such comparison highlights better the advantages of our approach. To our best knowledge there are no attempts in the literature to construct (meta) heuristics for the facility location problem with the lexicographic minimax objective, and such comparison could be considered as a topic for future research.

Moreover, the algorithm A-LEX can be directly applied to an arbitrary utility function dependent on the distance; therefore, it allows comparing different measures expressing how customers perceive the suitability of facility locations. The more challenging task consists of finding out how the algorithm could be extended to the composite measures representing the customer's choice behavioural model, for example, how to compute the lexicographic optimum when customers are not assigned to the closest facility, but their demand is distributed within a subset of k -nearest located facilities.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to Sergio García for making the code of the algorithm ZEBRA publicly available and for his valuable comments which helped them run the code. They thank P. Cenek, L. Jánošíková, and M. Kvet for thorough reading of the paper and useful suggestions. This work was supported by the Research Grants VEGA 1/0339/13 “Advanced Microscopic Modelling and Complex Data Sources for Designing Spatially Large Public Service Systems” and APVV-0760-11 “Designing Fair Service Systems on Transportation Networks.”

References

- [1] H. A. Eiselt and V. Marianov, *Foundations of Location Analysis, International Series in Operations Research and Management Science*, Springer Science + Business, Berlin, Germany, 2011.
- [2] M. S. Daskin, *Network and Discrete Location: Models, Algorithms and Applications*, John Wiley & Sons, New York, NY, USA, 1995.
- [3] Z. Drezner, *Facility Location: A Survey of Applications and Methods*, Springer, New York, NY, USA, 1995.
- [4] S. Müller, K. Haase, and S. Kless, “A multiperiod school location planning approach with free school choice,” *Environment and Planning A*, vol. 41, no. 12, pp. 2929–2945, 2009.
- [5] H. Luss, *Equitable Resources Allocation: Models, Algorithms and Applications*, John Wiley & Sons, New York, NY, USA, 2012.
- [6] H. Luss, “On equitable resource allocation problems: a lexicographic minimax approach,” *Operations Research*, vol. 47, no. 3, pp. 361–378, 1999.
- [7] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, New York, NY, USA, 2 edition, 1992.
- [8] R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhäuser, Boston, Mass, USA, 2004.
- [9] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan Kaufmann, San Francisco, Calif, USA, 2004.
- [10] L. R. Ford Jr. and D. R. Fulkerson, “A simple algorithm for finding maximal network flows and an application to the Hitchcock problem,” *Canadian Journal of Mathematics. Journal Canadien de Mathématiques*, vol. 9, pp. 210–218, 1957.
- [11] J. Rawls, *A Theory of Justice*, Harvard University Press, Cambridge, Mass, USA, 1971.
- [12] N. Megiddo, “Optimal flows in networks with multiple sources and sinks,” *Mathematical Programming*, vol. 7, pp. 97–107, 1974.
- [13] N. Megiddo, “A good algorithm for lexicographically optimal flows in multi-terminal networks,” *Bulletin of the American Mathematical Society*, vol. 83, no. 3, pp. 407–409, 1977.
- [14] D. Nace and M. Pióro, “Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial,” *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 5–17, 2008.
- [15] R. Carvalho, L. Buzna, W. Just, D. Helbing, and D. K. Arrow-smith, “Fair sharing of resources in a supply network with constraints,” *Physical Review E-Statistical, Nonlinear, and Soft Matter Physics*, vol. 85, no. 4, Article ID 046101, 2012.
- [16] D. Nace, L. N. Doan, O. Klopfenstein, and A. Bashllari, “Max-min fairness in multi-commodity flows,” *Computers & Operations Research*, vol. 35, no. 2, pp. 557–573, 2008.
- [17] W. Ogryczak and T. Śliwiński, “On direct methods for Lexicographic Min-Max optimization,” in *Proceedings of the Computational Science and Its Applications (ICCSA '06)*, M. Gavrilova, O. Gervasi, V. Kumar et al., Eds., vol. 3982 of *Lecture Notes in Computer Science*, pp. 802–811, Springer, Berlin, Germany, 2006.
- [18] W. Ogryczak and T. Śliwiński, “Sequential algorithms for exact and approximate max-min fair bandwidth allocation,” in *Proceedings of the Telecommunications Network Strategy and Planning Symposium (NETWORKS '12)*, pp. 1–6, 2012.
- [19] M. Zukerman, M. Mammadov, L. Tan, I. Ouveysi, and L. L. H. Andrew, “To be fair or efficient or a bit of both,” *Computers and Operations Research*, vol. 35, no. 12, pp. 3787–3806, 2008.
- [20] J. N. Hooker and H. P. Williams, “Combining equity and utilitarianism in a mathematical programming model,” *Management Science*, vol. 58, no. 9, pp. 1682–1693, 2012.
- [21] F. P. Kelly, A. K. Maulloo, and D. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.
- [22] D. Bertsimas, V. F. Farias, and N. Trichakis, “The price of fairness,” *Operations Research*, vol. 59, no. 1, pp. 17–31, 2011.
- [23] D. Bertsimas, V. F. Farias, and N. Trichakis, “On the efficiency-fairness trade-off,” *Management Science*, vol. 58, no. 12, pp. 1–17, 2012.
- [24] S. L. Hakimi, “Optimum distribution of switching centers in a communication network and some related graph theoretic problems,” *Operations Research*, vol. 13, pp. 462–475, 1965.
- [25] M. T. Marsh and D. A. Schilling, “Equity measurement in facility location analysis: a review and framework,” *European Journal of Operational Research*, vol. 74, no. 1, pp. 1–17, 1994.
- [26] T. Drezner, Z. Drezner, and J. Guyse, “Equitable service by a facility: minimizing the Gini coefficient,” *Computers and Operations Research*, vol. 36, no. 12, pp. 3240–3246, 2009.
- [27] W. Ogryczak, “Inequality measures and equitable locations,” *Annals of Operations Research*, vol. 167, pp. 61–86, 2009.
- [28] R. Z. Farahani, M. SteadieSeifi, and N. Asgari, “Multiple criteria facility location problems: a survey,” *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 34, no. 7, pp. 1689–1709, 2010.
- [29] M. Ehrgott, H. W. Hamacher, and S. Nickel, “Geometric methods to solve max-ordering location problems,” *Discrete Applied Mathematics. The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, vol. 93, no. 1, pp. 3–20, 1999.
- [30] A. Suzuki and Z. Drezner, “The minimum equitable radius location problem with continuous demand,” *European Journal of Operational Research*, vol. 195, no. 1, pp. 17–30, 2009.
- [31] O. Berman, Z. Drezner, A. Tamir, and G. O. Wesolowsky, “Optimal location with equitable loads,” *Annals of Operations Research*, vol. 167, pp. 307–325, 2009.
- [32] J. Krarup and P. M. Pruzan, “Reducibility of minimax to minisum 0 – 1 programming problems,” *European Journal of Operational Research*, vol. 6, no. 2, pp. 125–132, 1981.
- [33] W. Ogryczak, “On the lexicographic minimax approach to location problems,” *European Journal of Operational Research*, vol. 100, no. 3, pp. 566–585, 1997.

- [34] W. Ogryczak, "On the distribution approach to location problems," *Computers and Industrial Engineering*, vol. 37, no. 3, pp. 595–612, 1999.
- [35] <http://www.fico.com/>.
- [36] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.
- [37] L. A. Lorena and E. L. Senne, "A column generation approach to capacitated p -median problems," *Computers & Operations Research*, vol. 31, no. 6, pp. 863–876, 2004.
- [38] J. A. Díaz and E. Fernández, "Hybrid scatter search and path relinking for the capacitated p -median problem," *European Journal of Operational Research*, vol. 169, no. 2, pp. 570–585, 2006.
- [39] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*, vol. 840, Springer, Berlin, Germany, 1994.
- [40] Data describing the road network of the Slovak Republic were purchased from the publisher MAPA Slovakia Plus s.r.o., 2014, <http://www.mapaslovakia.sk/>.
- [41] United States Census Database 2000, 2014, <http://www.nationalatlas.gov/mld/ce2000t.html>.
- [42] Y. Zhang, O. Berman, and V. Verter, "The impact of client choice on preventive healthcare facility network design," *OR Spectrum. Quantitative Approaches in Management*, vol. 34, no. 2, pp. 349–370, 2012.
- [43] T. Drezner and Z. Drezner, "The gravity p -median model," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1239–1251, 2007.
- [44] S. García, M. Labbé, and A. Marín, "Solving large p -median problems with a radius formulation," *INFORMS Journal on Computing*, vol. 23, no. 4, pp. 546–556, 2011.
- [45] 2014, <https://sites.google.com/site/sergiogarciaquiles/Home>.