Efficient Approximation Methods for Lexicographic Max-Min Optimization

Tomasz Śliwiński

Warsaw University of Technology, Warsaw, Poland

https://doi.org/10.26636/jtit.2024.1.1421

Abstract — Lexicographic max-min (LMM) optimization is of considerable importance in many fairness-oriented applications. LMM problems can be reformulated in a way that allows to solve them by applying the standard lexicographic maximization algorithm. However, the reformulation introduces a large number of auxiliary variables and linear constraints, making the process computationally complex. In this paper, two approximation schemes for such a reformulation are presented, resulting in problem size reduction and significant performance gains. Their influence on the quality of the solution is shown in a series of computational experiments concerned with the fair network dimensioning and bandwidth allocation problem.

Keywords — fairness, lexicographic max-min, lexicographic optimization, network dimensioning

1. Introduction

No commonly accepted definition of fairness exists. In the decision-making environment, if there are some uniform criteria, i.e. criteria applied to different entities of the same object type, one would like them to be treated equally and impartially. For example, in a system serving multiple users, resources could be allocated in a way ensuring that the outcomes are fairly distributed among users.

Let us formalize the generic decision problem of this type. There is a given set J of m clients (users, services), $J = \{1, 2, \ldots, m\}$. There is also a given set $\mathcal{Q} \subseteq R^n$ of feasible decisions $\mathbf{x}$, $\mathbf{x} \in \mathcal{Q}$. For each client $j \in J$, a real-valued function $f_j(\mathbf{x})$ of the decision $\mathbf{x}$ is defined. This function measures the outcome $y_j = f_j(\mathbf{x})$ of the decision for client j. An outcome may have the form of a service time, service cost, delay or a more subjective, case-specific utility level. Without loss of generality, we can assume that each individual outcome is to be maximized.

The basic notion of fairness would imply that the same maximal outcome is assigned to each client. This could be achieved by the max–min solution concept depending on the optimization of the worst outcome:

$$\max_{\mathbf{x}} \left\{ \min_{j \in J} f_j(\mathbf{x}) : \mathbf{x} \in \mathcal{Q} \right\} . \tag{1}$$

Obviously, this approach becomes inefficient when, due to the complex nature of the feasible set $\mathcal{Q}$, one of the clients gets only a very small outcome, because in such a scenario everyone obtains the same small outcome [1]–[3]. While allocating clients to service facilities, for instance, such a situation may be caused by the existence of an isolated client located at a considerable distance from all the facilities. Minimization of the maximum distance is then restricted to that single isolated client, leaving other allocation decisions unoptimized [4]. This clearly is an inefficient solution case, where other outcomes may still be improved while maintaining fairness (equitability), by leaving the worst outcome at its best possible value.

Various approaches to fairness-related problems differ in terms of the fairness/equality measures applied, ways of expressing preferences, and in solution algorithms deployed. A thorough review of fairness-related approaches and formulations may be found in [5].

One approach consists in the application of the so-called max-min fairness (MMF) concept. It allows to achieve not only fair solutions, but is also decently efficient in terms of system utilization. It gained wide acceptance in many optimization fields, for instance in network optimization [6], [7]. In MMF, fairness is accomplished by a simple max-min optimization with regularization, through maximization of the second-smallest outcome, provided that the smallest outcome remains as large as possible. Then, maximization of the third-smallest outcome is performed, provided that the two smallest outcomes remain as large as possible, and so on.

This approach is equivalent to the lexicographic max-min (LMM) solution. It prevents some services with structurally low outcomes from blocking/disabling the max-min function. Although LMM does not leave any room for the decision-maker's preferences regarding the distribution of outcomes, such as maximizing the worst versus maximizing the average outcome, it has also been widely used in practical applications [8]–[10].

One needs to note that the lexicographic maximization in LMM is not applied to any specific order of the original criteria, i.e. we do not assume any priority of the objective functions to be maximized. Still, in the case of convex optimization, such as simple linear programming (LP), there exists an objective function that is constant (blocked) on the entire optimal set of the max-min problem from Eq. (1) [11]. Thus, after solving the max-min problem, it is possible to identify and remove the blocked objective from the new restricted max-min problem defined on the previously optimal set. Additionally, in the case of linear problems and by using the simplex method for solving the max-min problem, the set of blocking objectives can be easily determined [12], [13].

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

In discrete models, due to the lack of convexity, no blocked outcomes may exist [2], thus eliminating the possibility of applying the sequential max-min algorithm. Such problems can be solved effectively with the use of the ordered cumulated outcomes methodology introduced in [14], [15] by sequential optimization of directly defined criteria. In [16], two alternative approaches relying on this methodology were developed and analyzed, namely ordered outcomes and ordered values. They enable to form lexicographic sequential procedures for various non-convex (possibly discrete) LMM problems and are based on directly introduced criteria, with a certain LP expansion of the original model.

However, both approaches suffer from fast growth of the problem's size in subsequent iterations of the standard lexicographic algorithm, which results in implementation-related difficulties. This is particularly evident in interactive decision support systems, where the result should be computed within a short timeframe. Hence, the need for accurate and efficient approximation techniques.

Below, ordered outcomes and ordered values approaches are briefly introduced. Next, we show how both approaches may be effectively approximated for significant performance gains, with only a minor deterioration of the quality of the solution. Finally, we introduce an application example and show the results of the numerical experiments performed.

2. Direct LMM Models

2.1. Ordered Outcomes Approach

The LMM problem becomes much more difficult in the case of non-convex optimization, in particular with the mixed-integer scenario applied. [17] showed that for any non-convex LMM problem, the k-th ordered outcome, denoted by $y_{\langle k \rangle}$, can be expressed using the following formula:

$$y_{\langle k \rangle} = \max_{\mathbf{x}, t_k, z_{kj}} \Big\{ t_k : t_k - y_j \leqslant M z_{kj},\ z_{kj} \in \{0,1\}\ \forall j, \quad \sum_{j=1}^{m} z_{kj} \leqslant k-1, \quad \mathbf{y} \in \mathcal{A} \Big\}, \tag{2}$$

where

$$\mathcal{A} = \big\{ \mathbf{y} = (y_1, y_2, \ldots, y_m) : y_j = f_j(\mathbf{x})\ \forall j,\ \mathbf{x} \in \mathcal{Q} \big\}$$

is a set of all attainable outcome vectors, i.e. outcomes of feasible decisions $\mathbf{x} \in \mathcal{Q}$ and M is a sufficiently large constant. The model given by Eq. (2) restricts the number of violated constraints $t_k \leqslant y_k$ to $k-1$. As t_k is the objective being maximized, it effectively evaluates to the k-th worst (smallest) element of $\mathbf{y}$. Equation (2) can be used to state any LMM problem as a standard lexicographic maximization, namely:

$$\begin{aligned} \operatorname{lex} \max_{\mathbf{x}, t_k, z_{jk}} \Big\{ & (t_1, t_2, \ldots, t_m) \\ \text{s.t } & t_k - y_j \leqslant M z_{kj},\ z_{kj} \in \{0,1\}\ \forall j,k \\ & \sum_{j \in J} z_{kj} \leqslant k-1\ \forall k \\ & \mathbf{y} \in \mathcal{A} \Big\} . \end{aligned} \tag{3}$$

However, the existence of the binary variables z_{kj} makes this representation computationally demanding.

The above problem can be solved using the standard lexicographic maximization algorithm which, for the general problem of

$$\operatorname{lex} \max \Big\{ \big(g_1(\mathbf{s}), g_2(\mathbf{s}), \ldots, g_m(\mathbf{s}) \big) : \mathbf{s} \in S \Big\} ,$$

can be stated as follows:

0: Put $k := 1$,
1: Solve the problem P_k defined as:
$g_k^* = \max_{\mathbf{s} \in Q} \{ g_k(\mathbf{s}) : g_j(\mathbf{s}) \geqslant g_j^* \quad \forall j < k \}$,
2: If $k = m$, stop, otherwise put $k := k+1$ and go to 1.

The optimal solution in the last step of the algorithm is the optimal solution of the LMM problem. Note that in each iteration, the problem solved in step 1 is expanded by additional constraints applied to the objectives from previous steps.

2.2. Cumulated Ordered Outcomes

The main drawback of the LMM model from Eq. (3) is the need for binary variables z_{kj}. The recently developed modeling approach uses a cumulation of criteria with the help of linear expressions [3], [18]. Let us consider a cumulative operator expressing the total of k worst outcomes. As lexicographic optimization of the cumulated criteria does not influence the optimal solution of the original problem, the LMM problem can be stated as a standard lexicographic maximization of cumulated criteria:

$$\operatorname{lex} \max \Big\{ \big(\bar{\theta}_1(\mathbf{y}), \bar{\theta}_2(\mathbf{y}), \ldots, \bar{\theta}_m(\mathbf{y}) \big) : \mathbf{y} \in \mathcal{A} \Big\} . \tag{4}$$

The cumulative operator used in the above formulation can be modeled with the help of an auxiliary objective and a number of criteria. For *any* given vector $\mathbf{y}$, the cumulative operator $\bar{\theta}_k(\mathbf{y})$ can be found as the optimal value of the following LP problem:

$$\bar{\theta}_k(\mathbf{y}) = \min \left\{ \sum_{j \in J} y_j u_{kj} : \sum_{j \in J} u_{kj} = k,\ 0 \leqslant u_{kj} \leqslant 1\ \forall j \in J \right\}. \tag{5}$$

While this problem becomes non-linear for the variable vector $\mathbf{y}$, its LP dual preserves linearity even for the variable vector $\mathbf{y}$. Let t_k and d_{kj} denote the dual variables corresponding to the equation:

$$\sum_{j=1}^{m} u_{kj} = k$$

and to the upper bounds on u_{kj}, respectively. Now, one may write the following LP dual problem to Eq. (5):

$$\bar{\theta}_k(\mathbf{y}) = \max\left\{kt_k - \sum_{j\in J} d_{kj} : t_k - y_j \leqslant d_{kj},\ d_{kj} \geqslant 0\ \forall j \in J\right\}. \tag{6}$$

Using the above dual formulation, we can express the lex–max problem – Eq. (4) – as follows:

$$\begin{aligned} \operatorname{lex\,max} & \left(t_1 - \sum_{j\in J} d_{1j}, 2t_2 - \sum_{j\in J} d_{2j}, \ldots, mt_m - \sum_{j\in J} d_{mj}\right) \\ & \text{s.t. } \mathbf{y} \in \mathcal{A},\ d_{kj} \geqslant t_k - y_j,\ d_{kj} \geqslant 0 \\ & \quad \forall i, j \in J, k = 1, \ldots, m\ . \end{aligned} \tag{7}$$

In needs to be noted that despite the auxiliary objectives, constraints, and variables all being linear, the above problem formulation remains valid for non-convex feasible sets $\mathcal{A}$. Unfortunately, its size grows quickly with each iteration of the lexicographic algorithm, reaching – in the final iteration $m+m^2$ auxiliary variables and m^2 constraints. Still, as shown by our previous experiments, the above formulation performs much better than the direct formulation (3).

2.3. Ordered Values Approach

In many applications, the set of attainable outcome values is finite, allowing for quite a different formulation of the lexicographic max-min optimization. Let us express the distribution of outcomes using outcome frequencies. Let $V = \{v_1, v_2, \ldots, v_r\}$ be a set of all possible outcome values, i.e. objective values for each client, corresponding to feasible decision vectors $x \in Q$, and $v_1 < v_2 < \cdots < v_r$. Furthermore, let the function $h_k(\mathbf{y})$ express the number of outcomes in the vector $\mathbf{y}$, taking the value v_k, i.e.

$$h_k(\mathbf{y}) = |\{j : y_j = v_k, j \in J\}|\ .$$

We can now define a cumulative distribution function as:

$$\bar{h}_k(\mathbf{y}) = \sum_{l=1}^{k} h_l(\mathbf{y}). \tag{8}$$

Function $\bar{h}_k(\mathbf{y})$ expresses the number of outcomes smaller than or equal to v_k. In particular, $\bar{h}_r(\mathbf{y}) = m$ for any outcome vector $\mathbf{y}$. As we intend to maximize all the outcomes, we are interested in minimizing all the functions $\bar{h}_k$ for $k = 1, 2, \ldots, r-1$. Hence, we can express the LMM optimization problem (3) as a standard lexicographic minimization problem with objectives $\bar{h}_k(y)$ [2]:

$$\operatorname{lex\,min}\left\{\left(\bar{h}_1(\mathbf{y}), \bar{h}_2(\mathbf{y}), \ldots, \bar{h}_r(\mathbf{y})\right) : \mathbf{y} \in \mathcal{A}\right\}. \tag{9}$$

Now, taking advantage of the fact that the cumulation of consecutive outcomes does not affect lexicographic optimization and that values v_k are strictly increasing, we can rewrite the above problem using cumulated $\bar{h}_k(\mathbf{y})$ values, weighted with the differences among consecutive v_k values:

$$\operatorname{lex\,min}\left\{\left(\hat{h}_1(\mathbf{y}), \ldots, \hat{h}_r(\mathbf{y})\right) : \mathbf{y} \in \mathcal{A}\right\}, \tag{10}$$

where

$$\hat{h}_k(\mathbf{y}) = \sum_{l=1}^{k} (v_{l+1} - v_l)\bar{h}_l(\mathbf{y})\ ,$$

for $k = 2, \ldots, r$, and $\hat{h}_1 = 0$. One can interpret the $\hat{h}_k(\mathbf{y})$ value as the total shortage of outcome values to the value v_k, and express it in an alternative fashion as:

$$\hat{h}_k(\mathbf{y}) = \sum_{j\in J} \max\{v_k - y_j, 0\}\ .$$

This enables the formulation of the LMM as a standard lexicographic minimization with auxiliary constraints and variables [18], [19]:

$$\begin{aligned} \operatorname{lex\,min} & \left(\sum_{j\in J} h_{2j}, \sum_{j\in J} h_{3j}, \ldots, \sum_{j\in J} h_{rj}\right) \\ \text{s.t. } & h_{kj} \geqslant v_k - y_j,\ \ h_{kj} \geqslant 0 \\ & \forall j \in J,\ \ k = 2, \ldots, r,\ \ \mathbf{y} \in \mathcal{A}. \end{aligned} \tag{11}$$

3. Approximation Methods

In the following section, two different approximation methods and their variations are considered. The first one is based on the reduction of the number of lexicographic iterations. In the cumulated ordered outcomes approach, this is achieved by reducing the number of lexicographic levels, while in the ordered values approach, by reducing the number of distinguished target values. The latter approximation strategy preserves the number of lexicographic steps but prevents the problem from increasing in size by aggregating the auxiliary constraints from the previous iterations, in the form of simple bounds applied to the outcome variables.

3.1. Reduction of Lexicographic Steps (RLS)

Let us consider the cumulated ordered outcomes model and a sequence of indices $I = \{i_1, i_2, \ldots, i_q\} \subset J$, where $i_1 < i_2 < \cdots < i_q$. For example, I may consist of every n-th index from the original sequence J. We can now formulate the approximation of the original LMM problem in the following manner:

$$\operatorname{lex\,max}\left\{\left(\bar{\theta}_{i_1}(\mathbf{y}), \bar{\theta}_{i_2}(\mathbf{y}), \ldots, \bar{\theta}_{i_q}(\mathbf{y})\right) : \mathbf{y} \in \mathcal{A}\right\}. \tag{12}$$

In such an approach, we can expect a reasonably fair solution, and the only unfairness may be related to the distribution of outcomes within classes of the skipped criteria. The model's efficiency and approximation error will be related to the number and distribution of indices within sequence I.

A similar methodology can be applied to the ordered values approach with the number of the distinguished target values being restricted.

Let us consider a sequence of indices $K = \{k_1, k_2, \ldots, k_q\}$, where $v_{k_i} < v_{k_2} < \cdots < v_{k_q}$. Now, the corresponding approximate lexicographic optimization problem can be stated as follows:

$$\operatorname{lex\,min}\left\{\left(\hat{h}_{k_1}(\mathbf{y}), \hat{h}_{k_2}(\mathbf{y}), \ldots, \hat{h}_{k_q}(\mathbf{y})\right) : \mathbf{y} \in \mathcal{A}\right\}. \tag{13}$$

The original problem with the full set of attainable target values allows us to generate the optimal LMM solution.

However, the reduction in the number of lexicographic steps certainly introduces approximation errors resulting from the unfair distribution of outcomes within classes of the skipped criteria, and depends on the distribution of indices in sequence K.

3.2. Aggregation of Auxiliary Constraints (AAC)

The basic idea behind this approach is to replace the constraints applied to objective functions from previous iterations by the bounds imposed on the outcomes.

Let us go back to the standard lexicographic maximization algorithm. In each iteration k, an additional constraint in the form of $g_{k-1}(\mathbf{s}) \geqslant g^*_{k-1}$, where g^*_{k-1} is the optimal objective value from step $k-1$, is included in the maximization problem. This constraint is kept in all subsequent iterations and prevents the outcome of the previous iteration from worsening its values, which is consistent with the definition of LMM.

In the case of the cumulated ordered outcomes approach (7), in iteration k of the standard lexicographic maximization algorithm, objective

$$g_k(\mathbf{s}) = kt_k - \sum_{j\in J} d_{kj}$$

together with constraints $d_{kj} \geqslant t_k - y_j \ \forall j \in J$ expresses the cumulated value of k worst outcomes. In subsequent iterations, it is lower-bounded by the optimal value g^*_k from iteration k, i.e.

$$kt_k - \sum_{j\in J} d_{kj} \geqslant g^*_k \ ,$$

which effectively prevents the total of k lowest outcomes from decreasing. All constraints applied to the objectives in previous iterations, together with auxiliary variables and constraints, are kept in the model, thus increasing its size by m variables d_{kj}, one variable t_k and m auxiliary constrains in the form of $d_{kj} \geqslant t_k - y_j$ after one k-th iteration.

At the end of the standard lexicographic maximization algorithm, the size of the model may be large enough to pose a significant challenge even to modern solvers.

Let t^*_k and y^*_j be the values of t_k and y_j in an optimal solution in iteration k, when solving model (7) with the use of the standard lexicographic maximization algorithm. The value of the k-th objective denoted by $g(\mathbf{s})$:

$$g(\mathbf{s}) = kt_k - \sum_{j\in J} d_{kj} \ ,$$

which expresses the total of k worst (smallest) outcomes

$$g(\mathbf{s}) = \sum_{i=1}^{k} y_{\langle i \rangle}$$

will obviously not decrease in subsequent iterations if none of the outcomes worsens, i.e. if we put $y_j \geqslant y^*_j \ \ \forall j \in J$.

Actually, for some outcomes, this lower bound may be relaxed, i.e. $y_j \geqslant t^*_k$ for $j \in J$ such that $y^*_j > t^*_k$, as this does not affect the total of k worst outcomes. Hence, in iterations following iteration k, the constraint

$$kt_k - \sum_{j\in J} d_{kj} \geqslant g^*_k$$

together with m auxiliary constraints $d_{kj} \geqslant t_k - y_j \ \forall j \in J$ is replaced with simple lower bounds put on the outcomes: $y_j \geqslant \min\{y^*_j, t^*_k\}$.

Obviously, this constrains the solutions in subsequent iterations of the lexicographic maximization algorithm, leading to non-optimal solutions, but simultaneously greatly reduces the size of the problem.

Some balance between performance and solution quality may be achieved if in the subsequent iterations $k+1, \ldots, m$ only $1, \ldots, k-n$ previous objective constraints are aggregated in the above way, and the remaining objective constraints $k-n+1, \ldots, k$ are formulated to the full extent, as defined in problem (7).

A similar aggregation may be formulated for the ordered values approach to LMM. The objective in iteration k, i.e.

$$\sum_{j\in J} h_{kj}$$

together with auxiliary constraints in the form of

$$h_{kj} \geqslant v_k - y_j \ \ \forall j \in J$$

express the total shortage of outcome values to v_k. In subsequent iterations, it is upper-bounded by including constraint

$$\sum_{j\in J} h_{kj} \leqslant g^*_k$$

to the optimization problem. However, the total shortage expressed by the objective may be preserved in subsequent iterations if the above constraints are aggregated in the form of lower bounds put on the outcome values: $y_j \geqslant \min\{y^*_j, v^*_k\}$. This once again accelerates the solution time with a trade-off in terms of the solution's quality. The right balance may be achieved by applying those aggregations to earlier iterations only.

Tab. 1. Example network parameters.

Network	Nodes	Links
pdh	11	34
newyork	16	49
ta1	24	55
france	25	45
norway	27	51
cost266	37	57

4. Application Example and Numerical Experiments

One of the basic application areas of the MMF concept is the optimization of systems that serve many users. In the following example, network dimensioning and routing are considered. This problem has an important prerequisite of preserving the right level of fairness when allocating bandwidth to competing services (users).

Let us consider a network G consisting of a set V of nodes and a set E of undirected links. There is also a set $J = \{1, 2, \ldots, m\}$ of services defined in the network. Each service $j \in J$ depends on a flow between a given pair of nodes. The flow may be routed on exactly one path chosen from set P_j of paths allowed for each service. The paths are represented by given binary matrices

$$\Delta_e = (\delta_{ejp})_{j \in J; p \in P_j},$$

assigned to each link $e \in E$, where $\delta_{ejp} = 1$ if link e belongs to path $p \in P_j$.

The services may use any bandwidth assigned thereto. Examples can be found in Voice over IP or Video on Demand services, where a higher bandwidth translates into higher audio or video quality. The objective is to allocate link capacities (common resource) to competing services, maximizing total throughput, and taking care of the fairness of the allocation.

The above traffic routing problem is extended by allowing some network dimensioning, performed by expanding the existing capacities. Each link $e \in E$ is assumed to have an existing capacity. It can be expanded to $a_e + \xi_e$, where the expansion value is bounded in the range of 0 to $\bar{a}_e$. A limited budget B constrains the overall expansion of the network. The unit expansion cost of each link $e \in E$ is given and denoted by c_e.

The formal model may be stated as follows:

$$0 \leqslant x_{jp} \leqslant M b_{jp},\ b_{jp} \in \{0, 1\} \quad j \in J;\ p \in P_j \quad (14)$$

$$\sum_{p \in P_j} x_{jp} = y_j,\ \sum_{p \in P_j} b_{jp} = 1 \quad j \in J \quad (15)$$

$$\sum_{j \in J} \sum_{p \in P_j} \delta_{ejp} x_{jp} \leqslant a_e + \xi_e \quad \forall e \in E \quad (16)$$

$$0 \leqslant \xi_e \leqslant \bar{a}_e \quad \forall e \in E \quad (17)$$

$$\sum_{e \in E} c_e \xi_e \leqslant B, \quad (18)$$

where x_{jp} is the flow for service j on path $p \in P_j$ and y_j – being the total flow for service j – represents the model outcome for each service. The single-path flow requirement is enforced with binary variables b_{jp} and multiple-choice constraints (14)–(15), with a sufficiently large constant M.

The outcomes y_j for each service $j \in J$ are to be distributed fairly, using lexicographic max-min optimization. The application of one of the previously defined LMM models to the above problem consists in extending model (14)–(18) by a number of auxiliary linear constraints, variables, and objectives embedded in the standard lexicographic optimization algorithm.

We analyzed the time performance and the quality of the proposed LMM approximations, namely RLS and AAC. For the experiments, we used six network topologies from the survivable network design library (SNDLib) [20]. Their respective parameters (number of nodes and links) are listed in Tab. 1. For each topology, we generated 10 random problems following the algorithm introduced in [18]. First, for each link, we generated its current capacity a_e and unit expansion cost c_e, as numbers from the range of 2 to 10 and 1 to 1.5, respectively. Based on the current capacity, the maximum expansion capacity $\bar{a}$ is also generated as a number in the range of $0.2\, a_e$ to $0.6\, a_e$. Budget B for the network expansion was set to 130% of the current network value, i.e.

$$B = 1.3 \sum_{e \in E} (c_e a_e).$$

All random numbers referred to above were generated with uniform distribution applied.

Two problem sizes were tested with respect to the number of services – problems with 30 and 50 services were considered. Each service is defined by a random node pair, and the service flow can be realized on one of three different, potential paths. Two of them are fully random, and one is the shortest path between the end nodes (with the smallest number of links).

The exact and approximate algorithms were tested in diverse configurations. As a reference for all experiments, we used the cumulated ordered outcomes approach (COO) to compute exact LMM results.

As for the RLS approximations applied to the COO approach, two configurations were analyzed:

1) COO2 – utilizing every second index from the original outcomes sequence plus the last index,
2) COO4 – utilizing every fourth index from the original outcomes sequence plus the last index.

As for the AAC approximation applied to the COO approach, we tested an aggregation of constraints from all previous iterations (COOa) and an aggregation of constraints from all previous iterations except the last iteration (COOa1).

No bandwidth granulation and thus no grid of possible bandwidth values were assumed. Therefore, in the ordered values approach (OV), the resulting bandwidth allocation is always an approximation to the exact LMM solution. However, to compare its computational effectiveness to that of COO and COO2, we decided for the OV to use the same number of LMM steps (and distinct v_k values) as in COO or COO2, respectively. The OV approach with the number of v_k values reduced by factor 2 in the RLS approximation will be denoted by OV2. The assumed attainable outcome values v_k are computed with formula:

$$v_k = \underline{z} + \frac{(k-1)}{(|J|-1)} (\bar{z} - \underline{z}),$$

where $k = 1, 2, 3, \ldots, |J|$ for OV and $k = 1, 3, \ldots, |J|$ for OV2, $\underline{z}$ is the worst outcome (precomputed with max-min) and $\bar{z}$ is a simple estimation of the highest possible flow for any of the services.

It is computed as $\max_{j \in J, p \in P_j} \min_{e \in p} (a_e + \bar{a}_e)$, which expresses the least burdensome bottleneck over all possible service paths. One needs to not that the real objective value may exceed $\bar{z}$. In our preliminary experiments, this happened only occasionally and by a small margin.

The AAC approximation was also applied to the ordered values approach and the only tested case was a configuration aggregating all the constraints from previous iterations (OVa).

Tab. 2. Average computing times [s].

Network	No. of srv	Algorithm							
		COO	COO2	COO4	COOa	COOa1	OV	OV2	OVa
pdh	30	2.2	0.6	0.3	0.3	0.3	1.8	0.5	0.2
newyork		6.4	1.4	0.5	0.3	0.5	3.3	0.9	0.3
ta1		5.2	1.3	0.4	0.3	0.4	2.8	1.0	0.3
france		3.0	0.7	0.3	0.3	0.4	2.0	0.7	0.3
norway		8.0	1.9	0.7	0.8	0.8	3.2	0.6	0.4
cost266		4.9	1.1	0.3	0.4	0.4	2.3	0.6	0.3
pdh	50	211.9	31.9	5.3	1.7	2.0	30.1	6.8	1.0
newyork		207.9	47.6	12.1	1.3	2.0	122.6	18.8	1.0
ta1		235.2	55.0	36.5	4.1	4.0	107.7	10.1	2.0
france		154.2	22.5	5.6	5.4	3.7	15.8	3.0	0.8
norway		164.2	23.7	5.5	2.9	3.7	22.8	7.3	1.3
cost266		174.7	28.7	5.9	1.8	2.1	18.7	4.3	0.7

Tab. 3. Relative deviation for $\beta = 10\%$ of the worst outcomes [%].

Network	No. of srv	Algorithm						
		COO2	COO4	COOa	COOa1	OV	OV2	OVa
pdh	30	0.0	–1.9	0.0	0.0	–0.9	–1.3	–1.0
newyork		0.0	–1.5	0.0	0.0	–0.9	–2.0	–1.0
ta1		0.0	–0.2	0.0	0.0	–0.2	–0.3	–0.3
france		0.0	0.0	0.0	0.0	0.0	0.0	–0.3
norway		0.0	0.0	0.0	0.0	–0.3	–0.4	–0.5
cost266		0.0	0.0	0.0	0.0	0.0	0.0	–0.3
pdh	50	0.0	0.0	0.0	0.0	–1.1	-2.2	–1.3
newyork		0.0	0.0	0.0	0.0	–1.4	–1.4	–1.6
ta1		0.0	0.0	0.0	0.0	–0.6	-0.8	–1.0
france		0.0	0.0	0.0	0.0	–0.1	–0.1	–0.8
norway		0.0	0.0	0.0	0.0	–0.2	–0.2	–0.8
cost266		0.0	0.0	0.0	0.0	–0.1	–0.1	–0.9

All the experiments were performed on the Intel Core i7 3.4 GHz microprocessor using the CPLEX 12.1 optimization library for MIP optimization problems. All the results are the average of 10 randomly generated problems. The results of the experiments are shown in Tabs. 2–5. Table 2 presents the average computing times, and Tabs. 3–5 show the estimated approximation errors. The first two columns of each table describe the network name and the number of services defined over the networks. The remaining columns denote the results for different lexicographic max-min approaches (COO or OV) and approximation algorithms (RLS – columns COO2, COO4, OV2 or AAC – columns COOa, COOa1, OVa).

As one may notice in Tab. 2, the computing time increases rapidly along with the problem size and depends heavily on the number of services defined over the network (columns COO and OV). As for the approximation methods, the newly introduced AAC approximation performs exceptionally well, as compared to the basic lexicographic max-min approaches, meaning that when compared with the exact cumulated ordered outcomes approach (column COO), the computing time of the AAC approximation (columns COOa and COOa1) is reduced by a factor of 30–120 in the case of 50 services. A similar reduction is observed when the AAC approximation (column OVa) is applied to the ordered values approach (column OV).

The RLS approximation does not offer as spectacular results as achieved in the case of AAC approximation. However, when using the RLS approximation, the computing times can be reduced by a significant factor (columns COO2, COO4, and OV2).

Tab. 4. Relative deviation for $\beta = 50\%$ of the worst outcomes [%].

Network	No. of srv	Algorithm						
		COO2	COO4	COOa	COOa1	OV	OV2	OVa
pdh	30	0.0	0.0	–0.1	0.0	0.3	0.1	0.1
newyork		0.0	0.0	–0.3	–0.5	0.2	–0.3	–0.6
ta1		0.0	0.0	–0.4	–0.1	–0.1	–0.2	–0.3
france		0.0	–0.3	0.0	–0.1	–0.3	–1.0	–0.6
norway		0.0	–0.3	–0.3	–0.2	0.0	–1.7	–0.1
cost266		0.0	0.1	0.0	0.0	0.8	0.4	0.5
pdh	50	0.1	0.1	–0.2	0.0	–0.3	–0.5	–0.5
newyork		0.1	0.1	–0.8	–0.3	0.3	0.3	–0.4
ta1		–0.3	–0.4	–0.75	–0.3	–0.2	0.7	–0.8
france		0.0	0.1	–0.8	–0.4	–0.2	–0.7	–0.4
norway		0.0	0.0	–0.1	–0.6	0.4	–0.9	0.3
cost266		0.0	0.0	0.0	0.0	0.5	–0.1	0.0

Tab. 5. Relative deviation for $\beta = 100\%$ of the worst outcomes [%].

Network	No. of srv	Algorithm						
		COO2	COO4	COOa	COOa1	OV	OV2	OVa
pdh	30	0.0	0.3	–0.1	–0.3	0.1	0.6	–0.3
newyork		0.0	0.1	0.1	–0.4	0.5	1.2	–0.4
ta1		–0.4	0.1	–0.4	–0.2	–0.2	0.4	–0.4
france		0.5	1.0	–0.6	–0.6	0.7	1.2	–0.3
norway		0.1	0.8	–1.0	–1.0	0.6	0.9	–0.4
cost266		0.1	0.8	0.1	–0.2	0.7	1.7	0.8
pdh		0.0	0.2	–0.6	–0.3	0.1	0.4	–1.0
newyork	50	–0.1	–0.1	–0.1	–0.4	0.3	0.5	0.0
ta1		0.9	1.1	–0.2	–0.2	1.1	1.2	0.2
france		0.2	0.3	0.3	0.2	1.1	1.7	0.4
norway		0.1	0.3	–0.3	–0.3	0.6	1.0	0.1
cost266		0.1	0.4	–0.1	–0.2	1.5	0.8	0.7

The lexicographic max-min applied to the multiple criteria problem results in a certain distribution of outcomes. Each approximation method introduces some deviations from this distribution. To show the quality of RLS and AAC approximation methods, we compare three parameters of the resulting distribution: the total of the 10%, 50%, and 100% of the worst outcomes. The worst outcome is assumed to be correctly computed by each approximation approach. The actual number of the worst outcomes is upper rounded to the nearest integer.

To make the results independent of the absolute values, the relative deviations were computed as:

$$\frac{x_\beta - \mathrm{COO}_\beta}{\mathrm{COO}_\beta} \cdot 100\%, \tag{19}$$

where x_β is the total of β (β equals 10%, 50%, or 100%) worst outcomes in one of the approximation approaches, and COO_β is the total of β worst outcomes in the exact solution computed with use of the COO algorithm.

The results are presented in Tabs. 3–5, for the total of 10%, 50% and 100% of the worst outcomes, respectively.

For the COO approach, in the case of 10% of the worst outcomes, the COO2, COOa and COOa1 approximations fail to generate any noticeable approximation errors. For the 50% scenario, one may notice a slight deterioration of the quality of the AAC approximations (COOa and COOa1), as compared to RLS approximations (COO2 and COO4). The COOa1 approximation performs much better than COOa with only a minor computation time penalty. When the total of all outcomes (100%) is considered, the quality of COO2, COO4, COOa and COOa1 solutions is similar, but does not deviate from the exact solution by more than 1.1%.

Also, the approximations for the ordered outcomes (OV) approach perform quite well, although, for obvious reasons, the relative deviations are more significant than for COO approaches. However, values in the range $[-1.0\%, 1.7\%]$ may still be qualified as a very good result.

5. Conclusions

We developed approximate optimization methods that contribute to the previously known max-min fairness principle and the underlying lexicographic max-min optimization algorithms. They allow high-speed and accurate computations and can be applied for both convex and non-convex solution spaces. The numerical experiments for the fair bandwidth allocation and network dimensioning problem proved the exceptional performance and high solution quality of the proposed methods. With short computing times and minor solution quality losses, they allow to apply the lexicographic max-min approach in the context of interactive decision support systems, where short system response times are preferred.

References

[1] H. Luss, *Equitable Resource Allocation: Models, Algorithms, and Applications*, Hoboken: John Wiley & Sons, 376 p., 2012 (https://doi.org/10.1002/9781118449189).

[2] W. Ogryczak, *Linear and Discrete Optimization with Multiple Criteria: Preference Models and Applications to Decision Support*, Warsaw University Press, 1997 (ISBN: 9788323099413) [in Polish].

[3] W. Ogryczak, M. Pióro, and A. Tomaszewski, "Telecommunication Network Design and Max-min Optimization Problem", *Journal of Telecommunication and Information Technology*, no. 3, pp. 43–56, 2005 (https://jtit.pl/jtit/article/view/326).

[4] W. Ogryczak, "On the Lexicographic Minimax Approach to Location Problems", *European Journal of Operational Research*, vol. 100, no. 3, pp. 566–585, 1997 (https://doi.org/10.1016/S0377-2217(96)00154-3).

[5] V.X. Chen and J.N. Hooker, "A Guide to Formulating Fairness in an Optimization Model", *Annals of Operations Research*, vol. 326, no. 1, pp. 581–619, 2023 (https://doi.org/10.1007/s10479-023-05264-y).

[6] J. Kleinberg, Y. Rabani, and E. Tardos, "Fairness in Routing and Load Balancing", *Journal of Computer and System Sciences*, vol. 63, no. 1, pp. 2–21, 2001 (https://doi.org/10.1006/jcss.2001.1752).

[7] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, San Francisco: Morgan Kaufman, 800 p., 2004 (https://doi.org/10.1016/B978-0-12-557189-0.X5000-8).

[8] J. Jaffe, "Bottleneck Flow Control", *IEEE Transactions on Communications*, vol. 29, no. 7, pp. 954–962, 1981 (https://doi.org/10.1109/TCOM.1981.1095081).

[9] D. Nace, L.N. Doan, O. Klopfenstein, and A. Bashllari, "Max-min Fairness in Multi-commodity Flows", *Computers and Operations Research*, vol. 35, no. 2, pp. 557–573, 2008 (https://doi.org/10.1016/j.cor.2006.03.020).

[10] M. Pióro, M. Dzida, E. Kubilinskas, and P. Nilsson, "Applications of the Max-min Fairness Principle in Telecommunication Network Design", in: *Next Generation Internet Network*s, *2005*, Rome, Italy, pp. 219–225, 2005 (https://doi.org/10.1109/NGI.2005.1431669).

[11] W. Ogryczak, "Location Problems from the Multiple Criteria Perspective: Efficient Solutions", *Archives of Control Sciences*, vol. 7, no. 3, pp. 161–180, 1998 (https://www.ia.pw.edu.pl/~wogryczа/publikacje/artykuly/mcdmlocn.pdf).

[12] F.A. Behringer, "A Simplex Based Algorithm for the Lexicographically Extended Linear Maxmin Problem", *European Journal of Operational Research*, vol. 7, no. 3, pp. 274–283, 1981 (https://doi.org/10.1016/0377-2217(81)90349-0).

[13] R.S. Klein, H. Luss, and U.G. Rothblum, "Minimax Resource Allocation Problems with Resource-substitutions Represented by Graphs", *Operations Research*, vol. 41, no. 5, pp. 959–971, 1993 (https://doi.org/10.1287/opre.41.5.959).

[14] W. Ogryczak and T. Śliwiński, "On Equitable Approaches to Resource Allocation Problems: The Conditional Minimax Solution", *Journal of Telecommunications and Information Technology*, no. 3, pp. 40–48, 2002 (https://jtit.pl/jtit/article/view/134).

[15] W. Ogryczak and A. Tamir, "Minimizing the Sum of the *k* Largest Functions in Linear Time", *Information Processing Letters*, vol. 85, no. 3, pp. 117–122, 2003 (https://doi.org/10.1016/S0020-0190(02)00370-8).

[16] W. Ogryczak and T. Śliwiński, "On Direct Methods for Lexicographic Min-Max Optimization", *Lecture Notes in Computer Science*, vol. 3982, pp. 802–811, 2006 (https://doi.org/10.1007/11751595_85).

[17] R.R. Yager, "On the Analytic Representation of the Leximin Ordering and Its Application to Flexible Constraint Propagation", *European Journal of Operational Research*, vol. 102, no. 1, pp. 176–192, 1997 (https://doi.org/10.1016/S0377-2217(96)00217-2).

[18] W. Ogryczak and T. Śliwiński, "Sequential Algorithms for Exact and Approximate Max-min Fair Bandwidth Allocation", in: *2012 15th International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, Rome, Italy, 2012 (https://doi.org/10.1109/NETWKS.2012.6381707).

[19] W. Ogryczak and T. Śliwiński, "Sequential Algorithms for Max-min Fair Bandwidth Allocation", in: *Proceedings of the European Computing Conference*, vol. 1, pp. 511–522, 2009 (https://doi.org/10.1007/978-0-387-84814-3_51).

[20] O. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0-Survivable Network Design Library", *Network*, vol. 55, no. 3, pp. 276–286, 2010 (https://doi.org/10.1002/net.20371).

Tomasz Śliwiński, Ph.D., Assistant Professor
Institute of Control and Computation Engineering
https://orcid.org/0000-0002-5111-1830
E-mail: tomasz.sliwinski@pw.edu.pl
Warsaw University of Technology, Warsaw, Poland
https://www.pw.edu.pl