Lexicographic Max-Min Optimization for Efficient and Fair Bandwidth Allocation

Włodzimierz Ogryczak, *Warsaw Univ. of Technology, Inst. of Control & Comput. Engg., Poland*
Tomasz Śliwiński, *Warsaw Univ. of Technology, Inst. of Control & Comput. Engg., Poland*

Keywords: lexicographic max-min, max-min fairness, multiple criteria, multiple commodity network flow

1. Introduction

In many multiple criteria optimization problems preserving fairness among individual outcomes becomes an important issue. This is particularly true for systems that are designed for multiple users or services like computer or telecommunication networks. In this paper we investigate a new approach to the resource allocation preserving the, so called, Max-Min Fairness (MMF) in the solution. We analyze its applicability to a hard discrete multiple commodity network flow problem, where the flow (bandwidth) for each pair of nodes is allocated to exactly one path chosen from a predefined set.

There is no common definition of fairness. Intuitively, one wants the uniform criteria to be treated equally and impartially. For example, in the multiple users system resources can be allocated in a way that each user gets the same outcome. Obviously, this approach becomes inefficient when, for some reason, one of the users can get only very small value because then everyone gets the same small value. One of the ways to overcome this difficulty is the application the so called Max-Min Fairness concept. This allows to achieve not only fair solutions but also decently efficient in terms of the system utilization.

Let us follow the idea of the MMF concept basing on the example of the resource allocation in a multiuser system. In the first step we allocate as much of the resources as possible, with the assumption that the outcome value for all users stays the same. This corresponds to the max-min optimization, i.e. maximization of the worst outcome. In the second step we maximize the second smallest outcome (provided that the smallest one remains as large as possible). In the third step we maximize the third smallest (provided that the two smallest remain as large as possible), and so on. Formally, Max-Min Fairness corresponds to the lexicographic Max-Min and it can be seen as searching for a vector lexicographically maximal in the space of the feasible vectors with components rearranged in the non-decreasing order.

The lexicographic Max-Min solution is known in the game theory as the nucleolus of a matrix game. It originates from an idea [4] to select from the optimal strategy set those which allow one to exploit mistakes of the opponent optimally. It has been later refined to the formal nucleolus definition [15]. The concept was early considered in the Tschebyscheff approximation [16] as a refinement taking into account the second largest deviation, the third one and further to be hierarchically minimized. Similar refinement of the fuzzy set operations has been recently analyzed [5]. Within the telecommunications or network applications the lexicographic Max-Min approach has appeared already in [2] and now under the name Max-Min Fairness (MMF) is treated as one of the standard fairness concepts [14]. The LMM approach has been used for general linear programming multiple criteria problems [1, 9], as well as for specialized problems related to (multiperiod) resource allocation [7, 8] or decision under risk ([12] and references therein). The LMM approach has been considered also for various discrete optimization problems [3, 6] including the location-allocation ones [11].

In the previous work in this area the definition of the LMM solution concept was expressed with a direct formula including some mixed integer variables. On the contrary, we show that introduction of some artificial criteria with auxiliary linear variables and inequalities allows to model and solve the LMM problem in a more efficient way.

2. Problem description

The generic problem, we consider here, may be stated as follows. One or many resources are to be allocated to a set $J = \{1, 2, \ldots, m\}$ of competing users, processes, services, etc. Let Q be a feasible set of resource allocation decisions. The effect of the decision $\mathbf{x} \in Q$ on the service $j \in J$ is measured by uniform individual objective functions $y_j = f_j(x)$. The goal is to preserve the fairness of the decisions while keeping the highest system utilization possible. The fairness is achieved by application of the lexicographic maximinimization assumed, without loss of generality, that higher value of the individual objective function means better.

As stated before, the lexicographic maximinimization can be seen as searching for a vector lexicographically maximal in the space of the feasible vectors with components rearranged in the non-decreasing order. This can be mathematically formalized as follows. Let $\langle \mathbf{a} \rangle = (a_{\langle 1 \rangle}, a_{\langle 2 \rangle}, \ldots, a_{\langle m \rangle})$ denote the vector obtained from $\mathbf{a}$ by rearranging its components in the non-decreasing order. That means $a_{\langle 1 \rangle} \leq a_{\langle 2 \rangle} \leq \ldots \leq a_{\langle m \rangle}$ and there exists a permutation π of set J such that $a_{\langle j \rangle} = a_{\pi(j)}$ for $j = 1, 2, \ldots, m$. Comparing lexicographically such ordered vectors $\langle \mathbf{y} \rangle$ one gets the so-called lex-min order. The lexicographic Max-Min problem can be then represented in the following way.

$$\operatorname{lex\,max} \{(y_{\langle 1 \rangle}, y_{\langle 2 \rangle}, \ldots, y_{\langle m \rangle}) : \mathbf{y} \in A\} \tag{1}$$

where $A = \{\mathbf{y} : \mathbf{y} = \mathbf{f}(\mathbf{x}),\ \mathbf{x} \in Q\}$. The problems considered here are not necessarily convex, which means that we cannot apply the approach common for the LP problems [10] where dominating objective functions constant on the entire optimal set of the max-min problem were identified and eliminated in the successive steps of the sequential max-min optimization.

3. Direct models

In the classic approach introduced by Yager [17], each k-th value $y_{\langle k \rangle}$ can be expressed by the mixed-integer programming model:

$$y_{\langle k \rangle} = \max \{t_k : t_k - y_j \leq C z_{kj},\ z_{kj} \in \{0, 1\} \quad \forall j,\ \textstyle\sum_{j \in J} z_{kj} \leq k - 1\}$$

where C is a sufficiently large constant (larger than any possible difference between various individual outcomes y_j) which allows us to enforce inequality $t_k \leq y_j$ for $z_{kj} = 0$ while ignoring it for $z_{kj} = 1$. Hence, the lexicographic Max-Min problem (1) can be written as a standard lexicographic maximization:

$$\begin{array}{ll} \operatorname{lex\,max} & (t_1, t_2, \ldots, t_m) \\ \text{s.t.} & t_k - y_j \leq C z_{kj},\ z_{kj} \in \{0, 1\} \quad \forall\, j, k \\ & \sum_{j \in J} z_{kj} \leq k - 1 \quad \forall\, k \\ & \mathbf{y} \in A \end{array} \tag{2}$$

Unfortunately, binary variables z_{kj} in the auxiliary constraints contribute to implementation difficulties of this model. There is, however, a way to reformulate the above model so that only linear variables are used.

Let us consider cumulated criteria $\bar{\theta}_k(\mathbf{y}) = \sum_{i=1}^{k} y_{\langle i \rangle}$ expressing, respectively: the worst (smallest) outcome, the total of the two worst outcomes, the total of the three worst outcomes, etc. Within the lexicographic optimization a cumulation of criteria does not affect the optimal solution. Hence, the LMM problem can be formulated as the standard lexicographic maximization with cumulated ordered outcomes:

$$\operatorname{lex\,max} \{(\bar{\theta}_1(\mathbf{y}), \ldots, \bar{\theta}_m(\mathbf{y})) : \mathbf{y} \in A\}$$

This allows us to simplify dramatically the optimization problem since $\bar{\theta}_k(\mathbf{y}) = \min \{\sum_{j \in J} y_j u_{kj} : \sum_{j \in J} u_{kj} = k,\ 0 \leq u_{kj} \leq 1 \quad \forall j\}$ and taking advantages of the LP duality, allows us to find $\bar{\theta}_k(\mathbf{y})$ as the optimal value of the following LP problem (without use of any integer variables):

$$\bar{\theta}_k(\mathbf{y}) = \max \{k t_k - \textstyle\sum_{j \in J} d_{kj} : t_k - d_{kj} \leq y_j,\ d_{kj} \geq 0 \quad \forall\, j\}. \tag{3}$$

It follows from (3) that $\bar{\theta}_k(\mathbf{y}) = \max \{kt_k - \sum_{j\in J}(t_k - f_j(\mathbf{x}))_+ : \mathbf{x} \in Q\}$, where $(.)_+$ denotes the nonnegative part of a number and t_k is an auxiliary (unbounded) variable. The latter, with the necessary adaptation to the minimized outcomes in location problems, is equivalent to the computational formulation of the k–centrum model introduced in [13].

Now, every optimal solution to the LMM problem (1) can be found as an optimal solution to a standard lexicographic optimization problem with predefined linear criteria:

$$\begin{array}{rl} \operatorname{lex}\max & [t_1 - \sum_{j\in J} d_{1j}, 2t_2 - \sum_{j\in J} d_{2j}, \dots, mt_m - \sum_{j\in J} d_{mj}] \\ \text{s.t.} & t_k - d_{kj} \le y_j,\ d_{kj} \ge 0 \quad \forall\, j,k \\ & \mathbf{y} \in A. \end{array} \tag{4}$$

This direct lexicographic formulation remains valid for nonconvex (e.g. discrete) feasible sets Q, where the standard sequential approaches [8, 9] are not applicable [11]. Note that model (4) does not use integer variables and it can be considered as an LP expansion of the original max-min problem. Thus, this model preserves the problem's convexity if the original problem is defined with a convex feasible set Q and concave objective functions f_j. The size of the problem is quadratic with respect to the number of outcomes ($m^2 + m$ auxiliary variables and m^2 constraints).

4. Ordered values

For some discrete optimization problems one may try to reformulate the above model taking advantage of the finiteness of the set of all possible outcome values. Let $V = \{v_1, v_2, \dots, v_r\}$ (where $v_1 < v_2 < \dots < v_r$) denote the set of all attainable outcomes (all possible values of the individual objective functions f_j for $\mathbf{x} \in Q$). Now, instead of maximizing the worst value, then maximizing the second worst value (provided that the worst one remains as large as possible), and so on, one may try to minimize the number of the outcomes equal to v_1, then to minimize the number of the outcomes equal to v_2 (provided that the number of outcomes equal to v_1 remains as small as possible), and so on.

Let $h_k(\mathbf{y})$ ($k = 1, 2, \dots, r$) be the number of values v_k in the outcome vector $\mathbf{y}$. The LMM solution concept can be expressed in terms of the standard lexicographic minimization problem with objectives $h_k(\mathbf{y})$ [10]:

$$\operatorname{lex}\min \{(h_1(\mathbf{y}), \dots, h_r(\mathbf{y})) : \mathbf{y} \in A\} \tag{5}$$

Unfortunately, for functions h_k there is no simple analytical formula allowing to minimize them without use of some auxiliary integer variables. This difficulty can be overcome by taking advantage of possible weighting and cumulating achievements in lexicographic optimization, one may eliminate auxiliary integer variables from the achievement functions. For this purpose we introduce $\hat{\mathbf{h}}(\mathbf{y})$ which denotes the total shortage of individual objective functions to v_k

$$\hat{h}_k(\mathbf{y}) = \sum_{l=1}^{k-1} (v_{l+1} - v_l)\bar{h}_l(\mathbf{y}) = \sum_{j\in J} (v_k - y_j)_+$$

where $\bar{h}_k = \sum_{l=1}^{k} h_l(\mathbf{y})$ and $\hat{h}_1(\mathbf{y}) = 0$ for any outcome vector. Due to positive differences $v_{l+1} - v_l > 0$, the lexicographic minimization problem (5) is equivalent to the lexicographic problem with objectives $\hat{h}_k(\mathbf{y})$:

$$\operatorname{lex}\min \{(\hat{h}_2(\mathbf{y}), \dots, \hat{h}_r(\mathbf{y})) : \mathbf{y} \in A\}$$

Moreover, such defined criteria are piecewise linear convex function [10] which allow to compute them directly by the minimization:

$$\hat{h}_k(\mathbf{y}) = \min\{\textstyle\sum_{j\in J} h_{kj} : h_{kj} \ge v_k - y_j,\ h_{kj} \ge 0\ \forall\, j\}$$

Therefore, every optimal solution to the LMM problem (1) can be found as an optimal solution to a standard lexicographic optimization problem with predefined linear criteria:

$$\begin{array}{rl} \operatorname{lex}\min & [\sum_{j\in J} h_{2j}, \sum_{j\in J} h_{3j}, \dots, \sum_{j\in J} h_{rj}] \\ \text{s.t.} & h_{kj} \ge v_k - y_j,\ h_{kj} \ge 0\ \forall\, j,k \\ & \mathbf{y} \in A. \end{array} \tag{6}$$

One may notice that the above approach can also be applied to problems with the infinite set of all possible outcome values. In this case, however, the resulting vector $\mathbf{y}$ is only the approximation of the LMM solution achieved by solving model (4). Certainly, its accuracy depends on the applied grid of the v_k values. The advantage of this approach is the ability to control accuracy versus computational efficiency. Additionally the distribution of v_k values in the outcome space can be freely chosen – it can be uniform but also, for example, exponential.

5. Computational experiments

We used multiple commodity network flow problem to perform some initial tests. Let us consider a network G consisting of a set V of nodes and of a set E of undirected links, each with given capacity c_e ($e \in E$). 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 the given pair of nodes. The flow can be routed on exactly one path chosen from a given set P_j predefined for each service. The objective is to allocate bandwidth to competing services preserving Max-Min Fairness. We do not make any assumptions about the set of possible bandwidth values that can be allocated. It means that in case of the ordered values approach the resulting allocation will only be an approximation to the exact solution.

Let δ_{ejp} ($e \in E$, $j \in J$, $p \in P_j$) be input parameter defining available paths for all services (δ_{ejp} equals 1 if and only if link e belongs to path p of the service j). Let x_{jp} ($j \in J$, $p \in P_j$) denote the bandwidth allocated to path p of the service j and let u_{jp} be the binary flag of that allocation. Using auxiliary variable $\mathbf{y} = (y_j)$ ($j \in J$) that denotes vector of individual objective values we can express the considered problem as

$$\begin{array}{ll} y_j = \sum_{p \in P_d} x_{jp}, & j \in J \\ x_{jp} \leq M u_{jp}, & j \in J,\ p \in P_d \\ \sum_{p \in P_d} u_{jp} = 1, & j \in J \\ \sum_{j \in J} \sum_{p \in P_d} \delta_{ejp} x_{jpj} \leq c_e, & e \in E \\ x_{jp} \geq 0, u_{jp} \in \{0, 1\} & j \in J,\ p \in P_d \end{array} \tag{7}$$

with the objective lex $\max\,(y_{\langle 1 \rangle}, y_{\langle 2 \rangle}, \ldots, y_{\langle m \rangle})$, where M is a sufficiently big constant.

For the experiments we used a set of 10 randomly generated problems for each tested size. The problem generation procedure was following. First, we created random but consistent network structure. Then, we chose random node pairs to define services. For each service 3 different possible flow routes between the two nodes were generated. Two of them were fully random and one was the shortest path between the nodes (with smallest number of links). We decided to use the integer grid of the v_k values in the ordered values approach, that is to check each integer value from the feasible set of objective values. In this case the number of steps depends on the range of the feasible objective values. Although it is not possible to fully control this parameter in randomly generated problems, we managed to restrict the number of steps to the range of 5 to 10 applying different link capacities for different problem sizes.

We analyzed the performance of the three models, i.e. classic (2), ordered outcomes (4) and ordered values (6) with the condition $\mathbf{y} \in A$ replaced by (7). Each model was computed using standard algorithm [17] for lexicographic optimization with predefined objective functions (lex $\max\{(g_1(\mathbf{y}), \ldots, g_m(\mathbf{y})) : \mathbf{y} \in Y\}$):

Step 0: Put $k := 1$.
Step 1: Solve problem P_k:
$\max\limits_{\mathbf{y} \in Y} \{\tau_k;\ \tau_k \leq g_k(\mathbf{y}),\ \tau_j^0 \leq g_j(\mathbf{y})\ \forall j < k\}$
and denote the optimal solution by $(\mathbf{y}^0, \tau_k^0)$.
Step 2: If $k = m$, then **STOP** ($\mathbf{y}^0$ is optimal solution).
Otherwise, put $k := k + 1$ and go to **Step 1**.

For example, the algorithm for the ordered outcomes approach worked according to the above scheme with functions g_k defined as $kt_k - \sum_{j \in J} d_{kj}$. Let $k = 1$. Following (4), we built initial problem P_1 with the objective $\tau_1 = t_1 - \sum_{j \in J} d_{1j}$ being maximized and m constraints of the form $t_1 - d_{1j} \leq y_j,\ j = 1 \dots m$. The expression $\mathbf{y} \in A$ of (4) was replaced by (7). Each new problem P_k in subsequent iterations ($k > 1$) was built by adding new constraints $\tau^0_{k-1} \leq t_{k-1} - \sum_{j \in J} d_{k-1,j}$ and $t_k - d_{kj} \leq y_j,\ j = 1 \dots m$ to problem P_{k-1}, where τ^0_{k-1} was the optimal objective value of P_{k-1}. Similar algorithm was performed for the classic as well as for the ordered values approach. The difference was in the objective and auxiliary constraints, as defined in (2) and (6), respectively. All the tests were performed on the Pentium IV 1.7GHz computer employing the CPLEX 9.1 package.

Table 1: Computation times (in seconds) for different solution approaches.

	number of of nodes	number of of links	number of services 5	10	20	30	45
classic	5	10	0.0	1.2			
approach	10	20	0.0	6.8	–	–	–
model (1)	15	30	0.0	3.9	–	–	–
ordered	5	10	0.0	0.2			
outcomes	10	20	0.0	1.3	^{3}62.9	–	–
model (4)	15	30	0.1	1.0	^{5}78.0	–	–
ordered	5	10	0.1	0.1			
values	10	20	0.0	0.3	4.1	^{2}35.0	^{7}100.5
model (6)	15	30	0.1	0.3	7.1	^{4}72.4	^{8}105.7

Table 1 presents solution times for the three approaches being analyzed. The times are averages of 10 randomly generated problems. The upper index denotes the number of tests out of 10 for which the timeout of 120 seconds occurred. The minus sign '–' shows that the timeout occurred for all 10 test problems. One can notice that while for smaller problems with number of services equal 5 all three approaches perform very well, for bigger problems only the ordered values approach gives acceptable results.

Table 2: Computation times (in seconds) for problems with increased link capacities.

	number of of nodes	number of of links	number of services 5	10	20	30	45
ordered	5	10	0.1	0.1			
values	10	20	0.1	1.3	23.8	^{4}74.3	–
model (6)	15	30	0.1	1.2	33.9	^{8}108.0	–

To check how the number of steps in the ordered values approach influences the test results we also performed similar experiments increasing this time the capacities of the links. The number of steps was restricted to the range of 15 to 25. In Table 2 we present the computing times only for the ordered values approach as for the other two methods no significant change in performance was observed. As one can see, although the computing times are bigger than in the previous tests, it still outperforms classical and ordered outcomes approaches.

6. Concluding remarks

As lexicographic minimization in the lexicographic Max-Min optimization is not applied to any specific order of the original criteria, the LMM optimization can be very hard to implement in general nonconvex (possibly discrete) problems. We have shown that introduction of some artificial criteria with auxiliary linear variables and inequalities allows one to model and solve the LMM problems in a very efficient way. We have tested computational performance of the presented models for certain multiple commodity network flow problem.

Acknowledgment

The research was supported by the Ministry of Science and Information Society Technologies under grant 3T11C 005 27. “Models and Algorithms for Efficient and Fair Resource Allocation in Complex Systems”.

References

[1] Behringer F.A. (1981) A simplex based algorithm for the lexicographically extended linear maxmin problem. Eur. J. Opnl. Res., 7, 274–283.

[2] Bertsekas D., Gallager R. (1987) Data Networks. Prentice-Hall, Englewood Cliffs.

[3] Burkard R.E., Rendl F. (1991) Lexicographic bottleneck problems. Operations Research Letters, 10, 303–308.

[4] Dresher M. (1961) Games of Strategy. Prentice-Hall, Englewood Cliffs.

[5] Dubois D., Fortemps Ph., Pirlot M., Prade H. (2001) Leximin optimality and fuzzy set-theoretic operations. Eur. J. Opnl. Res., 130, 20–28.

[6] Ehrgott M. (1998) Discrete decision problems, multiple criteria optimization classes and lexicographic max-ordering. Trends in Multicriteria Decision Making, T.J. Stewart, R.C. van den Honert (red.), Springer, Berlin, 31–44.

[7] Klein R.S., Luss H., Smith D.R. (1992) A lexicographic minimax algorithm for multiperiod resource allocation. Mathematical Programming, 55, 213–234.

[8] Luss H. (1999) On equitable resource allocation problems: A lexicographic minimax approach. Operations Research, 47, 361–378.

[9] Marchi E., Oviedo J.A. (1992) Lexicographic optimality in the multiple objective linear programming: the nucleolar solution. Eur. J. Opnl. Res., 57, 355–359.

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

[11] Ogryczak W. (1997a) On the lexicographic minimax approach to location problems. Eur. J. Opnl. Res., 100, 566–585.

[12] Ogryczak W. (2002) Multiple criteria optimization and decisions under risk. Control and Cybernetics, 31, 975–1003.

[13] Ogryczak W., Tamir A. (2003) Minimizing the sum of the k largest functions in linear time, Information Processing Letters, 85, 117–122.

[14] Pióro M., Medhi D. (2004) Routing, Flow and Capacity Design in Communication and Computer Networks. Morgan Kaufmann, San Francisco.

[15] Potters J.A.M., Tijs S.H. (1992) The nucleolus of a matrix game and other nucleoli. Mathematics of Operations Research, 17, 164–174.

[16] Rice J.R. (1962) Tschebyscheff approximation in a compact metric space. Bull. Amer. Math. Soc., 68, 405–410.

[17] Yager R.R. (1997) On the analytic representation of the Leximin ordering and its application to flexible constraint propagation. Eur. J. Opnl. Res., 102, 176–192.