BEZPOŚREDNIE METODY MINIMAKSYMALIZACJI LEKSYKOGRAFICZNEJ †

Włodzimierz OGRYCZAK*, Tomasz ŚLIWIŃSKI**

*Politechnika Warszawska, Instytut Automatyki i Informatyki Stosowanej
ul. Nowowiejska 15/19, 00-665 Warszawa e-mail:wogrycza@ia.pw.edu.pl

**Politechnika Warszawska, Instytut Automatyki i Informatyki Stosowanej
ul. Nowowiejska 15/19, 00-665 Warszawa e-mail:tsliwins@ia.pw.edu.pl

Streszczenie: Obok efektywności całego systemu, istotnym czynnikiem oceny decyzji jest minimalizacja nierówności (rozbieżności), czyli sprawiedliwe traktowanie wszystkich konkurujących użytkowników. Oparty na koncepcji sprawiedliwości Rawlsa model minimaksymalizacji leksykograficznej stanowi regularyzację standardowej minimaksymalizacji uwzgędniającą oprócz minimalizacji największej (najgorszej) oceny również minimalizację drugiej największej oceny (pod warunkiem minimalnych wartości największych ocen), minimalizację trzeciej największej (pod warunkiem minimalnych wartości dwóch największych ocen), itd. Celem niniejszej pracy jest weryfikacja możliwości skutecznego wykorzystania metodologii optymalizacji skumulowanych uporządkowanych ocen do bezpośrednich metod implementacji minimaksymalizacji leksykograficznej.

Słowa kluczowe: minimaksymalizacja leksykograficzna, efektywność, sprawiedliwość, porządkowa średnia ważona.

1. ZAGADNIENIE MINIMAKSYMALIZACJI LEKSYKOGRAFICZNEJ

Wiele zagadnień optymalnego projektowania lub sterowania systemem może być formułowanych w postaci zagadnień optymalizacji wielokryterialnej, gdzie poszczególne funkcje oceny wyrażają w jednolitej skali indywidualne oceny różnych użytkowników systemu i poszukuje się rozwiązania możliwie najbardziej zadowalającego wszystkich użytkowników. Obok efektywności całego systemu, istotnym czynnikiem oceny decyzji jest minimalizacja nierówności (rozbieżności), czyli sprawiedliwe (bezstronne i równe) traktowanie wszystkich konkurujących użytkowników (ang. fairness, equity) [9, 10, 15]. Tego typu modele wielokryterialne z jednolitymi (homogenicznymi) ocenami pojawiają się w wielu innych zagadnieniach, jak na przykład w licznych problemach dynamicznych, gdzie poszczególne funkcje oceny reprezentują tę samą ocenę w odniesieniu do różnych momentów czasu [8]. W problemach stochastycznych jednolite oceny mogą reprezentować różne możliwe wartości niedeterministycznej pojedynczej oceny ([14] i literatura tam cytowana). Ponadto, wiele technik modelowania problemów decyzyjnych wprowadza najpierw oceny jednolite, a następnie dokonuje ich (bezstronnej) agregacji, jak na przykład, agregacja relacji przynależności do zbioru rozmytego.

Dla ustalenia uwagi będziemy przyjmować, że funkcje oceny wyrażają negatywne własności systemu takie jak opóźnienia realizacji usługi, czas reakcji itp., czyli podlegają minimalizacji. W takim przypadku pewne wymagania sprawiedliwości mogą zostać osiągnięte przez minimaksymalizację, polegającą na minimalizacji największej (najgorszej) oceny:

$$\min\{ \max_{j=1,\ldots,m} f_j(\mathbf{x}) \ : \ \mathbf{x} \in Q \ \}$$

Faktycznie w przypadku bardzo prostego zbioru dopuszczalnego określonego jednym równaniem bilansującym rozwiązanie zadania minimaksymalizacji $\min\{\max_{j=1,\ldots,m} y_j \ : \ \sum_{j=1}^m y_j \ \leq \ b \ \}$ gwarantuje pełną równość wyników $\bar{y}_1 = \bar{y}_2 = \ldots = \bar{y}_m$. W ogólnym przypadku dla bardziej złożonych ograniczeń to nie jest już prawdą, a co gorsza, istnieje na ogół wiele rozwiązań zadania minimaksymalizacji i większość z nich jest zdominowana. Koncepcja optymalizacji minimaksowej może być rozciągnięta na minimalizację drugich co do wielkości współrzędnych wektorów ocen, trzecich co do wielkości itd. Prowadzi to do koncepcji minimaksymalizacji leksykograficznej, czyli minimalizacji leksykograficznej wektorów ze współrzędnymi uporządkowanymi nierosnąco (optymalizacji ocen uporządkowanych od najgorszej do najlepszej). Niech $\langle \mathbf{a} \rangle = (a_{\langle 1 \rangle}, a_{\langle 2 \rangle}, \ldots, a_{\langle m \rangle})$ oznacza wektor otrzymany z $\mathbf{a}$ w wyniku posortowania jego współrzędnych od najgorszej do najlepszej (w niemalejącym porządku). To znaczy, $a_{\langle 1 \rangle} \geq a_{\langle 2 \rangle} \geq \ldots \geq a_{\langle m \rangle}$ i istnieje permutacja π zbioru M taka, że $a_{\langle i \rangle} = a_{\pi(i)}$ dla $j = 1, \ldots, m$. Relacja nierówności leksykograficznej jest porządkiem liniowym i dlatego możliwe jest wyznaczenie najlepszego uporządkowanego wektora ocen w sensie tej relacji. Stosując porządek leksykograficzny do uporządkowanych wektorów ocen $\langle \mathbf{y} \rangle$ otrzymujemy zadanie minimaksymalizacji leksykograficznej:

$$\text{lex}\min\ \{(\theta_1(\mathbf{f}(\mathbf{x})), \ldots, \theta_m(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\},$$

† Praca wykonana w ramach grantu MNiI 3T11C 005 27: “Modele i algorytmy wspomagania efektywnego i sprawiedliwego rozdziału zasobów w złoónych systemach rozproszonych”.

gdzie $\theta_j(\mathbf{y}) = y_{\langle j\rangle}$.

Model minimaksymalizacji leksykograficznej stanowi regularyzację standardowej minimaksymalizacji uwzględniającą oprócz minimalizacji największej oceny również minimalizację drugiej największej oceny (pod warunkiem minimalnych wartości największych ocen), minimalizację trzeciej największej (pod warunkiem minimalnych wartości dwóch największych ocen), itd. Model minimaksymalizacji leksykograficznej stanowi formalizację koncepcji sprawiedliwości wprowadzonej przez Rawlsa [20]. Jest on też jedyną konsekwentną regularyzacją standardowego kryterium minimaksymalizacji zgodną z aksjomatami rozszerzania i zawężania zbioru ocen [12, 7].

Minimaksymalizacja leksykograficzna była najpierw analizowana w teorii gier jako koncepcja wyboru optymalnej (minimaksowej) strategii wykorzystującej dodatkowo złą (nie optymalną) grę przeciwnika [5] i ostatecznie sformalizowana jako jądro (nucleolus) gry [19]. Koncepcja ta była też rozpatrywana w teorii aproksymacji jednostajnej [21] jako dodatkowe uściślenie wyboru elementu aproksymującego w oparciu o drugie co do wielkości odchylenia, i dalsze. Podobne znaczenie ma wykorzystanie minimaksymalizacji leksykograficznej dla uściślania operatora przecięcia zbiorów rozmytych [6]. W zagadnieniach optymalizacji minimaksymalizacja leksykograficzna była najpierw rozpatrywana w odniesieniu do programowania liniowego [1, 11]. W zadaniach programowania liniowego, podobnie jak w grach macierzowych zbiór dopuszczalny jest wielościennym zbiorem wypukłym a oceny są funkcjami liniowymi. W takim przypadku zawsze istnieje blokująca (dominująca) funkcja oceny, stała (przyjmująca wartość optymalną) na całym zbiorze rozwiązań optymalnych zadania minimaksowego. Dlatego leksykograficzne rozwiązanie minimaksowe odpowiedniego zadania może być wyznaczone przez sekwencyjną optymalizację minimaksową z eliminacją blokujących funcji oceny. Umożliwia to zastosowanie minimaksymalizacji leksykograficznej do wieloetapowego (dynamicznego) zagadnienia rozdziału zasobów [8] i do zagadnień projektowania sieci telekomunikacyjnych [2, 18].

Minimaksymalizacja leksykograficzna była też rozpatrywana w modelach dyskretnych [3, 7, 4], w tym w zagadnieniach lokalizacyjnych [13]. W modelach dyskretnych brak wypukłości zbioru dopuszczalnego powoduje, że nie ma gwarancji istnienia oceny blokującej [12], co wyklucza stosowanie metodologii opartej na stopniowej eliminacji ocen blokujących. Celem niniejszej pracy jest weryfikacja możliwości skutecznego wykorzystania rozwiniętej wcześniej metodologii optymalizacji skumulowanych uporządkowanych ocen [15] do bezpośrednich metod implementacji minimaksymalizacji leksykograficznej. Zauważmy, że najnowowsze wyniki [17, 16] pokazują możliwości bezpośredniego wyrażenia w terminach programowania liniowego (konstrukcji z prostymi dodatkowymi nierównościami liniowymi) sumy ustalonej liczby najgorszych ocen i co za tym idzie ustalonej sekwencji funkcji liniowych do leksykograficznej optymalizacji dla realizacji minimaksymalizacji leksykograficznej.

2. MODELE BEZPOŚREDNIE

2.1. Uporządkowane oceny

Uporządkowane oceny wykorzystywane w minimaksymalizacji leksykograficznej bezpośrednio z definicji można wyrazić za pomocą zmiennych dyskretnych [22]:

$$\begin{aligned} y_{\langle k\rangle} \quad = \quad & \min\ t_k \\ \text{p.o.} \quad & \\ & t_k - y_j \ge -C z_{kj},\ z_{kj} \in \{0,1\}\ \forall j \\ & \sum_{j=1}^{m} z_{kj} \le k-1 \end{aligned}$$

gdzie C jest dostatecznie dużą stałą, która pozwala wymuszać nierówność $t_k \ge y_j$ dla $z_{kj} = 0$ jednocześnie ignorując ją dla $z_{kj} = 1$. Dla $k = 1$ wszystkie zmienne binarne z_{1j} są równe 0 co prowadzi do standardowego sformułowania minimaksymalizacji. Dla $k > 1$ wszystkie m zmiennych binarnych z_{kj} jest istotną częścią modelu, co prowadzi do złożonych zadań obliczeniowych.

Rozważmy skumulowane kryteria $\bar{\theta}_k(\mathbf{y}) = \sum_{i=1}^{k} y_{\langle i\rangle}$ wyrażające odpowiednio najgorszą (największą) wartość oceny, sumę dwóch najgorszych ocen, sumę trzech najgorszych ocen, itd. Przy stosowaniu leksykograficznej optymalizacji taka kumulacja ocen nie wpływa na rozwiązanie zadania, czyli minimaksymalizacja leksykograficzna jest równoważna zadaniu

$$\operatorname{lex}\min\ \{(\bar{\theta}_1(\mathbf{f}(\mathbf{x})),\ldots,\bar{\theta}_m(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\},$$

gdzie $\bar{\theta}_k(\mathbf{y}) = \sum_{i=1}^{k} y_{\langle i\rangle}$. Pozwala to istotnie uprościć problem dzięki bezpośredniemu wyrażeniu skumulowanych uporządkowanych ocen za pomocą zależności programowania liniowego (PL). Zauważmy, że dla dowolnego ustalonego wektora $\mathbf{y}$, jego skumulowane uporządkowane współrzędne stanowią wartości optymalne następujących zadań PL:

$$\begin{aligned} \bar{\theta}_k(\mathbf{y}) \quad = \quad & \max \sum_{j=1}^{m} y_j u_{kj} \\ \text{p.o.} \quad & \sum_{j=1}^{m} u_{kj} = k \\ & 0 \le u_{kj} \le 1 \ \ \text{dla}\ j = 1,\ldots,m. \end{aligned}$$

Powyższy model jest zadaniem PL dla ustalonego danego wektora ocen $\mathbf{y}$, ale staje się nieliniowym zadaniem w typowym przypadku wektora ocen $\mathbf{y}$ reprezentującego zmienne. Ta trudność może być pokonana, przez wykorzystanie dualnego modelu PL, gdzie wektor ocen $\mathbf{y}$ występuje jako prawa strona ograniczeń nierównościowych:

$$\begin{aligned} \bar{\theta}_k(\mathbf{y}) \quad = \quad & \min\ k t_k + \sum_{j=1}^{m} d_{kj} \\ \text{p.o.} \quad & t_k + d_{kj} \ge y_j,\ d_{kj} \ge 0 \quad \forall\, j. \end{aligned}$$

lub w zwartym zapisie

$$\bar{\theta}_k(\mathbf{f}(\mathbf{x})) = \min\ \{k t_k + \sum_{j=1}^{m} (f_j(\mathbf{x}) - t_k)_+ : \mathbf{x} \in Q\ \},$$

gdzie $(.)_+$ stanowi nieujemną część liczby, a t_k jest dodatkową zmienną pomocniczą (nieograniczoną co do znaku). Tak określony model umożliwia dla dowolnego zadania optymalizacji wprowadzenie minimaksymalizacji leksykograficznej w postaci dodatkowych ograniczeń PL. Tak więc w przypadku oryginalnego zadania PL otrzymany model stanowi leksykograficzne zadanie PL. Podejście to nie wymaga jednak spełnienia założeń wypukłości. Może być zatem stosowane do modeli dyskretnych tworząc ustalony ciąg zadań optymalizacji dyskretnej do rozwiązania.

$$\begin{array}{rl} \text{lex min} & [t_1 + \sum_{j=1}^{m} d_{1j}, \ldots, mt_m + \sum_{j=1}^{m} d_{mj}] \\ \text{p.o.} & t_k + d_{kj} \geq f_j(\mathbf{x}),\ d_{kj} \geq 0 \quad \forall\, j,k \\ & \mathbf{x} \in Q. \end{array}$$

2.2. Uporządkowane wartości

Dla pewnych szczególnych zagadnień optymalizacji dyskretnej (kombinatorycznej) możliwe jest wykorzystanie skończoności zbioru wszystkich możliwych wartości funkcji ocen f_j (wynikającej ze skończonej liczby rozwiązań dopuszczalnych). Wektor uporządkowanych ocen określa rozkład wartości ocen wynikających z decyzji $\mathbf{x}$. W przypadku gdy istnieje skończony zbiór możliwych wartości ocen (lub gdy możemy się ograniczyć do odpowiedniej skończonej aproksymacji tego zbioru, np. wartościami rozmytymi), można operować bezpośrednim opisem rozkładu wartości przez krotności przyjmowania poszczególnych wartości. Niech $V = \{v_1, v_2, \ldots, v_r\}$ (gdzie $v_1 > v_2 > \cdots > v_r$) oznacza zbiór wszystkich osiągalnych wartości ocen, czyli zbiór wszystkich możliwych wartości funkcji f_j dla $\mathbf{x} \in Q$. Możemy wprowadzić funkcje całkowite $h_k(\mathbf{y})$ wyrażające liczbę wystąpień wartości v_k w wektorze $\mathbf{y}$. Mając zdefiniowane funkcje h_k możemy wprowadzić skumulowane funkcje rozkładu: $\bar{h}_k(\mathbf{y}) = \sum_{l=1}^{k} h_l(\mathbf{y})$. Funkcja $\bar{h}_k$ wyraża liczbę współrzędnych wektora $\mathbf{y}$ większych lub równych v_k. Ponieważ dążymy do minimalizacji ocen, to jesteśmy również zainteresowani minimalizacją funkcji $\bar{h}_k$. Minimaksymalizacja leksykograficzna może być wtedy wyrażona jako standardowa minimalizacja leksykograficzna funkcji $\bar{h}_k(\mathbf{f}(\mathbf{x}))$:

$$\text{lex min}\ \{(\bar{h}_1(\mathbf{f}(\mathbf{x})), \ldots, \bar{h}_r(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\}$$

Zauważmy, że analogiczne wyrażenie nie byłoby możliwe dla funkcji h_k nie zawierających kumulacji, ponieważ nie można z góry ustalić, które tych funkcji powinny być minimalizowane, a które maksymalizowane. Niestety funkcje $\bar{h}_k$, podobnie jak h_k, nie mają prostej postaci analitycznej. Ta trudność może być pokonana przez dalszą kumulację funkcji z odpowiednimi współczynnikami wagowymi. Operacje (dodatniego) ważenia i kumulacji fukcji celu nie mają wpływu na wynik odpowiedniej optymalizacji leksykograficznej. Dlatego można wprowadzić funkcje:

$$\begin{aligned} \hat{h}_k(\mathbf{y}) &= \sum_{l=1}^{k-1} (v_l - v_{l+1}) \bar{h}_l(\mathbf{y}) \\ &= \sum_{j=1}^{m} (y_j - v_k)_+ = \sum_{j=1}^{m} \max\{y_j - v_k, 0\}, \end{aligned}$$

gdzie $\hat{h}_1(\mathbf{y}) = 0$ dla dowolnego wektora ocen. Tak zdefiniowane funkcje są przedziałami liniowe i wypukłe, co gwarantuje ich łatwą implementację w zadaniach minimalizacji za pomocą zależności liniowych:

$$\begin{array}{l} \hat{h}_k = \sum_{j=1}^{m} h_{kj} \\ h_{kj} \geq y_j - v_k,\ h_{kj} \geq 0 \ \text{ dla } j = 1, \ldots, m. \end{array}$$

Dlatego zadanie minimalizacji leksykograficznej:

$$\text{lex min}\ \{(\hat{h}_2(\mathbf{f}(\mathbf{x})), \ldots, \hat{h}_r(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\}$$

stanowi alternatywny model obliczeniowy dla problemu minimaksymalizacji leksykograficznej. Podobnie jak model minimalizacji leksykograficznej skumulowanych uporządkowanych ocen, powyższy model umożliwia dla dowolnego zadania optymalizacji wprowadzenie minimaksymalizacji leksykograficznej w postaci dodatkowych nierówności liniowych. Tak więc w przypadku oryginalnego zadania PL otrzymany model stanowi leksykograficzne zadanie PL. Podejście to nie wymaga jednak spełnienia założeń wypukłości. Może być zatem stosowane do zagadnień dyskretnych tworząc ustalony ciąg zadań optymalizacji dyskretnej do rozwiązania.

$$\begin{array}{rl} \text{lex min} & [\sum_{j=1}^{m} h_{2j}, \sum_{j=1}^{m} h_{3j}, \ldots, \sum_{j=1}^{m} h_{rj}] \\ \text{p.o.} & h_{kj} \geq f_j(\mathbf{x}) - v_k,\ h_{kj} \geq 0\ \forall\, j,k \\ & \mathbf{x} \in Q. \end{array}$$

3. EKSPERYMENTY OBLICZENIOWE

Przeprowadzone zostały eksperymenty obliczeniowe dla realizacji minimaksymalizacji leksykograficznej przy pomocy metod bezpośrednich. Za punkt wyjścia wybrano dyskretny problem lokalizacyjny. Problem ten można sformułować następująco. Dany jest zbiór m klientów (jednostek przestrzennych) oraz zbiór n potencjalnych lokalizacji obiektów. W szczególności może to być podzbiór (lub cały zbiór) punktów reprezentujących klientów. Ponadto dana jest liczba p ($p \leq n$) obiektów do lokalizacji. Decyzję można tu opisać poprzez zmienne binarne x_j ($j = 1, 2, \ldots, n$) równe 1, gdy ma być użyta j–ta lokalizacja, a 0 w przeciwnym przypadku. Następnie zakłada się, że dla każdego klienta $i = 1, 2, \ldots, m$ jest zdefiniowana funkcja $f_i(\mathbf{x})$ rozlokowania obiektów $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Jest ona miarą satysfakcji i–tego klienta z danego rozlokowania obiektów. W typowych sformułowaniach problemów lokalizacyjnych funkcja ta jest zwykle związana z odległością i dlatego jej mniejsza wartość oznacza wyższą satysfakcję klienta, a więc każda funkcja f_i jest minimalizowana. Decyzje przydziału są modelowane przy użyciu dodatkowych zmiennych decyzyjnych x'_{ij} ($i = 1, 2, \ldots, m; j = 1, 2, \ldots, n$) równych 1, gdy lokalizacja j–ta jest użyta do obsługi i–tego klienta, a 0 w przeciwnym przypadku.

$$\min \mathbf{f}(\mathbf{x}) \tag{1}$$

$$\text{p.o.} \sum_{j=1}^{n} x_j = p \tag{2}$$

$$\sum_{j=1}^{n} x'_{ij} = 1 \quad i = 1, 2, \ldots, m \tag{3}$$

$$x'_{ij} \leq x_j \quad i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n \tag{4}$$

$$x_j,\, x'_{ij} \in \{0, 1\} \quad i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n \tag{5}$$

Poszczególne funkcje f_i są zapisane w postaci liniowej $f_i = \sum_{j=1}^{n} d_{ij} x'_{ij}$ dla $i = 1, 2, \ldots, m$, gdzie współczynnik d_{ij} wyraża odległość i-tego klienta od lokalizacji j. Celem jest wyznaczenie lokalizacji dla minimaksymalizacji leksykograficznej wektora ocen $\mathbf{f}(\mathbf{x})$.

W zadaniach testowych za lokalizacje klientów przyjęto punkty na osi X o współrzędnych będących wielokrotnością liczby 5 generowanych jako liczby losowe o rozkładzie jednostajnym z przedziału $[0, 100]$. Wszystkie eksperymenty zostały przeprowadzone na komputerze PC klasy Pentium IV, 1.7 GHz z wykorzystaniem pakietu CPLEX 6.0.

Tablica 1. Czasy obliczeń (w sekundach) algorytmu minimaksymalizacji leksykograficznej z uporządkowanymi ocenami.

liczba	liczba obiektów (p)						
klientów (m)	1	2	3	5	7	10	15
2	0.0						
5	0.1	0.0	0.0				
10	0.2	0.3	0.3	0.3	0.1		
15	0.9	2.1	1.8	2.3	1.8	0.5	
20	3.2	8.3	11.0	16.2	15.7	23.2	0.5
25	10.2	29.4	46.5	52.7	–	–	10.2

Tablica 2. Czasy obliczeń (w sekundach) algorytmu minimaksymalizacji leksykograficznej z uporządkowanymi wartościami.

liczba	liczba obiektów (p)						
klientów (m)	1	2	3	5	7	10	15
2	0.0						
5	0.0	0.1	0.0				
10	0.1	0.1	0.0	0.1	0.0		
15	0.4	0.2	0.1	0.1	0.1	0.0	
20	1.0	0.6	0.4	0.2	0.2	0.0	0.0
25	1.8	1.5	1.7	0.8	0.6	0.1	0.1

W celu zbadania efektywności obliczeniowej algorytmów minimaksymalizacji leksykograficznej dla problemu (1)–(5) przeprowadzona została seria testów dla różnych jego rozmiarów. Wyniki przedstawione w tab. 1 i 2 reprezentują średnie czasy obliczeń 10 losowych zadań. Znak '-' oznacza, że przekroczony został maksymalny czas wykonania jednego kroku algorytmu (ustalony na poziomie 60s). Zauważmy, że algorytm uporządkowanych wartości charakteryzuje się znacznie krótszymi czasami obliczeń niż algorytm uporządkowanych ocen, co staje się szczególnie widoczne wraz ze wzrostem liczby klientów. Faktycznie liczba kroków algorytmu uporządkowanych wartości nie zależy od liczby klientów tylko od liczby różnych wartości odległości, a ta jest stała (nie przekracza 20 w generowanych przykładach). Co więcej, algorytm uporządkowanych wartości wymaga coraz mniejszej liczby zadań (kroków) przy większej liczbie lokalizowanych obiektów, jako że największa możliwa odległość w generowanych eksperymentach nie przekracza wartości $100/p$.

Tablica 3. Czasy obliczeń (w sekundach) poszczególnych kroków algorytmu ($m = 25$).

numer	uporządkowane oceny			uporządkowane wartości		
kroku	$p = 1$	$p = 3$	$p = 5$	$p = 1$	$p = 3$	$p = 5$
1	0.1	0.6	1.0	0.0	1.4	0.6
2	0.2	1.4	3.0	0.2	0.1	0.1
3	0.2	1.3	2.8	0.0	0.1	0.1
4	0.3	1.7	1.8	0.2	0.0	0.0
5	0.2	2.1	2.1	0.1	0.1	
6	0.3	1.8	1.8	0.1	0.0	
7	0.2	1.8	2.5	0.1		
8	0.3	1.8	2.1	0.1		
9	0.3	2.0	1.9	0.1		
10	0.3	2.1	3.1	0.0		
11	0.4	1.8	2.8	0.1		
12	0.3	2.0	2.6	0.0		
13	0.4	2.1	3.3	0.1		
14	0.3	2.2	3.0	0.1		
15	0.5	2.2	3.2	0.1		
16	0.4	2.5	3.0	0.1		
17	0.3	2.6	3.3	0.1		
18	0.5	2.3	1.7	0.2		
19	0.5	2.3	2.1	0.1		
20	0.5	2.2	1.0	0.0		
21	0.6	2.0	1.2			
22	0.6	1.8	0.9			
23	0.9	1.3	0.7			
24	0.7	1.3	0.8			
25	0.9	1.3	1.0			

Przeprowadzone zostały także testy pokazujące wpływ wprowadzania do modelu dodatkowych ograniczeń liniowych w kolejnych krokach algorytmów, tj. podczas wyznaczania kolejnych wartości $\bar{\theta}_k$ i odpowiednio $\hat{h}_k$. Tab. 3 i 4 zawierają czasy obliczeń pojedynczych kroków algorytmów będące średnią z 10 losowych zadań. Zauważmy, że algorytm uporządkowanych wartości generuje zadania rozwiązywane zwykle znacznie szybciej niż odpowiednie zadania algorytmu uporządkowanych ocen, pomimo podobnej struktury obu typów zadań.

Tablica 4. Czasy obliczeń (w sekundach) poszczególnych kroków algorytmu ($m = 20$).

numer kroku	uporządkowane oceny $p=1$	$p=3$	$p=5$	uporządkowane wartoci $p=1$	$p=3$	$p=5$
1	0.1	0.1	0.3	0.1	0.2	0.1
2	0.1	0.4	0.9	0.0	0.1	0.1
3	0.1	0.4	1.0	0.0	0.0	0.0
4	0.1	0.5	0.9	0.0	0.0	0.0
5	0.1	0.6	1.1	0.1	0.1	
6	0.1	0.6	1.0	0.0	0.0	
7	0.2	0.6	1.0	0.1		
8	0.1	0.7	1.1	0.0		
9	0.1	0.7	1.0	0.0		
10	0.2	0.6	0.9	0.1		
11	0.1	0.7	1.0	0.0		
12	0.2	0.6	0.9	0.0		
13	0.2	0.7	1.0	0.1		
14	0.1	0.8	0.9	0.0		
15	0.2	0.6	0.8	0.1		
16	0.2	0.6	0.6	0.1		
17	0.2	0.5	0.5	0.0		
18	0.3	0.4	0.4	0.1		
19	0.2	0.5	0.5	0.0		
20	0.3	0.4	0.4	0.1		

Ma to dodatkowe znaczenie dla efektywności całego procesu minimaksymalizacji leksykograficznej (por. tab. 2), powiększając efekt wymagania przez algorytm uporządkowanych wartości mniejszej liczby zadań (kroków) przy większej liczbie lokalizowanych obiektów. Jednocześnie wskazuje to na potencjalną atrakcyjność algorytmu uporządkowanych wartości nawet w przypadku zadań o bardzo dużej liczbie różnych możliwych wartości wynikowych.

DIRECT METHODS FOR LEXICOGRAPHIC MIN-MAX OPTIMIZATION

Abstract: The approach called the Lexicographix Min-Max or the Min-Max Fairness (MMF) problem depends on searching for solutions minimal according to the lex-max order. MMF is a refinement (regularization) of the standard min-max optimization, but in the former, in addition to the largest outcome, we minimize also the second largest outcome (provided that the largest one remains as small as possible), minimize the third largest (provided that the two largest remain as small as possible), and so on. The (point-wise) ordering of outcomes causes that the lexicographic min-max problem is, in general, hard to implement. Nevertheless, for convex problems it is possible to use iterative algorithms solving a sequence of properly defined min-max problems by eliminating some blocking criteria. In general, it may not exist any blocking criterion allowing for iterative min-max processing. In this paper we discuss two optimization models allowing to form lexicographic sequential procedures for various nonconvex (possibly discrete) MMF problems. Both the approaches are based on some LP expansion of the original model thus they remain relatively simple computationally.

Literatura

[1] Behringer F.A. (1981) A simplex based algorithm for the lexicographically extended linear maxmin problem. European Journal of Operational Research, 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] Della Croce F., Paschos V.T., Tsoukias A. (1999) An improved general procedure for lexicographic bottleneck problem. Operations Research Letters, 24, 187–194.

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

[6] Dubois D., Fortemps Ph., Pirlot M., Prade H. (2001) Leximin optimality and fuzzy set-theoretic operations. European Journal of Operational Research, 130, 20–28.

[7] 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.

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

[9] Kostreva M.M., Ogryczak W. (1999) Linear optimization with multiple equitable criteria. RAIRO Oper. Res., 33, 275–297.

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

[11] Marchi E., Oviedo J.A. (1992) Lexicographic optimality in the multiple objective linear programming: the nucleolar solution. European Journal of Operational Research, 57, 355–359.

[12] Ogryczak W. (1997) Wielokryterialna optymalizacja liniowa i dyskretna. Wydawnictwa Uniwersytetu Warszawskiego, Warszawa.

[13] Ogryczak W. (1997a) On the lexicographic minimax approach to location problems. European Journal of Operational Research, 100, 566–585.

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

[15] Ogryczak W., Śliwiński T. (2002) On equitable approaches to resource allocation problems: the conditional minimax solution, J. Telecommunications and Information Technology, 3/2002, 40–48.

[16] Ogryczak W., Śliwiński T. (2003) On solving linear programs with the ordered weighted averaging objective. European Journal of Operational Research, 148, 80–91.

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

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

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

[20] Rawls J. (1971) The Theory of Justice. Harvard University Press, Cambridge.

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

[22] Yager R.R. (1997) On the analytic representation of the Leximin ordering and its application to flexible constraint propagation. European Journal of Operational Research, 102, 176–192.