UNIVERSITÄT KAISERSLAUTERN

Lexicographic Max-Ordering - A Solution Concept for Multicriteria Combinatorial Optimization

Matthias Ehrgott

Preprint Nr. 268

ISSN 0943-8874

FACHBEREICH MATHEMATIK

Lexicographic Max-Ordering – A Solution Concept for Multicriteria Combinatorial Optimization

Matthias Ehrgott*
Universität Kaiserslautern
and
Zentrum für Techno- und Wirtschaftsmathematik

Abstract

In this paper we will introduce the concept of lexicographic max-ordering solutions for multicriteria combinatorial optimization problems. Section 1 provides the basic notions of multicriteria combinatorial optimization and the definition of lexicographic max-ordering solutions.

In Section 2 we will show that lexicographic max-ordering solutions are pareto optimal as well as max-ordering optimal solutions. Furthermore lexicographic max-ordering solutions can be used to characterize the set of pareto solutions. Further properties of lexicographic max-ordering solutions are given.

Section 3 will be devoted to algorithms. We give a polynomial time algorithm for the two criteria case where one criterion is a sum and one is a bottleneck objective function, provided that the one criterion sum problem is solvable in polynomial time. For bottleneck functions an algorithm for the general case of Q criteria is presented.

1 Introduction

In this paper we consider combinatorial optimization problems with multiple criteria. First we introduce the general framework.
We are given a ground set $E = \{e_1, \ldots, e_m\}$ of elements (e.g. edges of a graph) and a set $\mathcal{F} \subseteq \mathcal{P}(E)$ of feasible solutions (e.g. spanning trees of a graph). We have a vector-valued weight-function $w : E \to \mathbb{N}_0^Q$ and for each $q \in \{1, \ldots, Q\}$ an objective function $f_q : \mathcal{F} \to \mathbb{N}$ such that $f(F) = (f_1(F), \ldots, f_Q(F))^t \in \mathbb{N}^Q$.
In this paper we shall consider two types of objective functions, namely sum ($f_q(F) = \sum_{e \in F} w_q(e)$) and bottleneck ($f_q(F) = \max_{e \in F} w_q(e)$) functions. We always consider minimization problems.
The most common solution concepts in multicriteria optimization are the following.

1. Pareto optimality

   A feasible solution $F \in \mathcal{F}$ is pareto optimal (a pareto solution) if there is no $F' \in \mathcal{F}$ such that $f_q(F') \leq f_q(F) \quad q = 1, \ldots, Q$ and strict inequality holds in at least one case.

*Partially supported by a grant of the Deutsche Forschungs Gemeinschaft and the European HC&M Programme

The main disadvantage of this concept is the possibly large cardinality of the set of pareto solutions. It is easy to construct examples such that every feasible solution is pareto optimal and all solutions have different objective values, see e.g. [3] or [4].

2. Max-ordering optimality

   A solution $F \in \mathcal{F}$ is max-ordering optimal (a max-ordering solution) if

   $$\max_{q\in\{1,\ldots,Q\}} f_q(F) \leq \max_{q\in\{1,\ldots,Q\}} f_q(F')$$

   for all $F' \in \mathcal{F}$.

   Compared to pareto optimality this concept has the advantage of a unique optimal solution value, i.e. $\max_{q\in\{1,\ldots,Q\}} f_q(F)$ is the same for each max-ordering optimal solution F. But on the other hand only one of the Q objectives is taken into account for the optimal solution, i.e. only one objective function is needed to determine if a feasible solution is optimal, once the optimal value is known.

By introducing the concept of lexicographic max-ordering we shall combine both features: Consideration of all objective values simultanously and unique optimal solution value.

Definition 1 *1. For $x \in \mathbb{R}^Q$ define $\Theta(x)$ as the permutation $(\Theta_1(x),\ldots,\Theta_Q(x))$ of x for which $\Theta_1(x) \geq \ldots \geq \Theta_Q(x)$ holds.*

2. If $x, y \in \mathbb{R}^Q$ x is lexicographically smaller *than y if there is $0 < q < Q$ such that $x_i = y_i$, $i \leq q-1$ and $x_q < y_q$. We write $x <_{lex} y$.*

3. A feasible solution $F \in \mathcal{F}$ of a combinatorial optimization problem is lexicographic max-ordering optimal (lex-MO optimal, lex-MO solution) *if $\Theta(f(F)) \leq_{lex} \Theta(f(F'))$ for all $F' \in \mathcal{F}$.*

Such solutions have been introduced for location problems as lexicographic centers in [6] and for multicriteria linear programming in [5]. In the latter lex-MO solutions are called *nucleolar solutions.* The concept of the nucleolus is taken from game-theory. The relations of game-theory and muliobjective linear programming have been investigated in [1].

Example 1 *Let $\mathcal{F} = \{a, b, c, d, e\}$ and $f(F), \Theta(f(F))$ be given as in Table 1.*

F	$f(F)$	$\Theta(f(F))$
a	(1,3,8,2,4)	(8,4,3,2,1)
b	(4,3,8,1,1)	(8,4,3,1,1)
c	(7,5,4,6,1)	(7,6,5,4,1)
d	(3,7,4,6,5)	(7,6,5,4,3)
e	(4,7,5,6,5)	(7,6,5,5,4)

Table 1: f and $\Theta(f)$ in Example 1

Hence a, b, c, d are pareto optimal, c, d and e are max-ordering optimal. But the lex-MO solution c is unique.

Example 1 is illustrated in Figure 1.

2 Properties of Lexicographic Max-Ordering Solutions

In the following we will usually write $\Theta(F)$ instead of $\Theta(f(F))$.
The next result proves that lex-MO solutions really combine the concepts of pareto and max-ordering optimality.

Proposition 1 *1. If F is lex-MO optimal then F is pareto optimal.*

2. If F is lex-MO optimal then F is max-ordering optimal.

Proof:
Suppose that F is a lex-MO solution.

1. If there is some feasible F' such that $f_q(F') \leq f_q(F)$ for $q = 1, \ldots, Q$ and inequality holds in least one case it follows that $\Theta(F') <_{lex} \Theta(F)$, contradicting the choice of F.
2. If there is some F' such that $\max_{q=1,\ldots,Q} f_q(F') < \max_{q=1,\ldots,Q} f_q(F)$ we have that $\Theta_1(F') < \Theta_1(F)$. Hence obviously $\Theta(F') <_{lex} \Theta(F)$, again a contradiction.

□

Therefore we have the inclusion as in Figure 1, which due to Example 1 is strict in general.

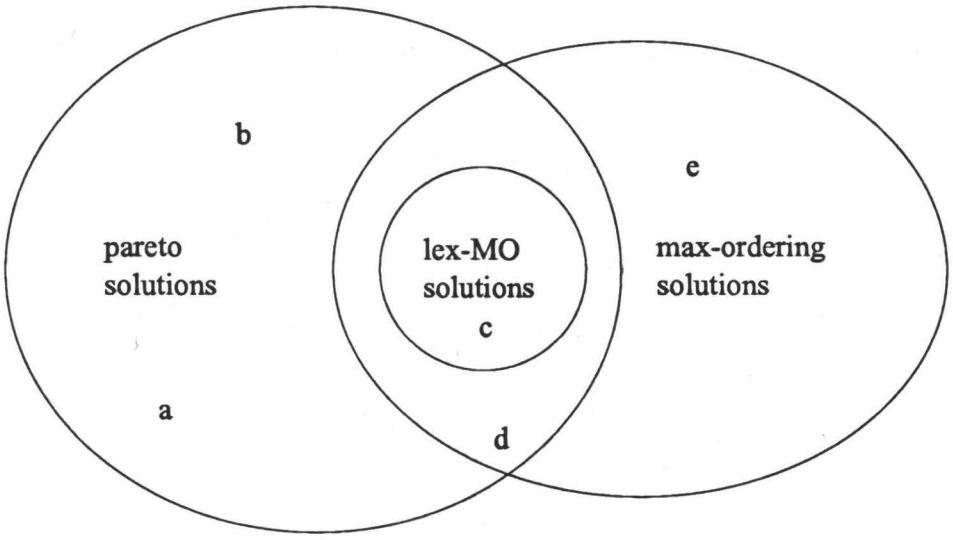

Figure 1: Lex-MO, pareto and max-ordering solutions

Lex-MO solutions can further be applied to parametrize the set of pareto solutions. Thus, besides the interest in lex-MO solutions due to combination of unique solution value and simultanous consideration of all objective functions, lex-MO solutions are of interest for determining the set of pareto solutions.

Proposition 2 *Let $\lambda_q > 0 \quad q = 1, \ldots, Q$ and let F^* be a lex-MO solution with respect to $f' = (f'_1, \ldots, f'_Q)$, where $f'_q(F) = \lambda_q f_q(F)$. Then F^* is pareto optimal with respect to f.*

Proof:
Suppose F^* is not pareto optimal. Then there exists some $F \in \mathcal{F}$ such that $f_q(F) \leq f_q(F^*)$ $q = 1, \ldots, Q$ and $f_{q^*}(F) < f_{q^*}(F^*)$ for some q^*. Hence $\lambda_q f_q(F) \leq \lambda_q f_q(F^*)$ $q = 1, \ldots, Q$ and $\lambda_{q^*} f_{q^*}(F) < \lambda_{q^*} f_{q^*}(F^*)$. Therefore

$$\Theta(\lambda_1 f_1(F), \ldots, \lambda_Q f_Q(F)) <_{lex} \Theta(\lambda_1 f_1(F^*), \ldots, \lambda_Q f_Q(F^*))$$

contradicting lex-MO optimality of F^*.

□

Note that in Proposition 2 it is important that $\lambda_q > 0$.

Proposition 3 *If F^* is pareto optimal then there exist $\lambda_q > 0$ $q = 1, \ldots, Q$ such that F^* is lex-MO optimal with respect to f', where f' is defined as in Proposition 2*

Proof:
Let $\lambda_q = \begin{cases} \frac{1}{f_q(F^*)} & f_q(F^*) \neq 0 \\ 2 & \text{otherwise} \end{cases}$.
Then $\lambda_q f_q(F^*) = \begin{cases} 1 & f_q(F^*) \neq 0 \\ 0 & \text{otherwise} \end{cases}$ and we have $\Theta(F^*) = (1, \ldots, 1, 0, \ldots, 0)$. Since F^* is pareto optimal for all $F \in \mathcal{F}$ such that $f(F) \neq f(F^*)$ there exists a q^* such that $f_{q*}(F) > f_{q^*}(F^*)$.
We consider two cases.

Case 1: $f_{q*}(F^*) \neq 0$
$\Rightarrow \lambda_{q^*} f_{q^*}(F) > \lambda_{q^*} f_{q^*}(F^*) = 1$

Case 2: $f_{q^*}(F^*) = 0$
$\Rightarrow f_{q^*}(F) > 0$
$\Rightarrow \lambda_{q^*} f_{q^*}(F) = 2 f_{q^*}(F) > 1$

The last inequality of Case 2 follows from integrality of weights $w_q(e)$. In both cases we conclude that $\Theta(F) >_{lex} \Theta(F^*)$ and the result follows.

□

We shall now mention two other properties of lex-MO solutions.

Proposition 4 *Let π be a permutation of $\{1, \ldots, Q\}$, let $f_\pi(F) = (f_{\pi(1)}(F), \ldots, f_{\pi(Q)}(F))$. Then*

$$\text{lexmin}_{F \in \mathcal{F}} \Theta(f_\pi(F)) = \text{lexmin}_{F \in \mathcal{F}} \Theta(f(F))$$

Proof:
For all $F \in \mathcal{F} : \Theta(f(F)) = \Theta(f_\pi(F))$.

□

This property is called *anonymity*. Note that this property (obviously) does not hold for usual lexicographic optimization. The next results considers a property concerning invariance under (positive) affine transformations of objective functions and weights.

Proposition 5 *1. Let $f'_q = af_q + b \quad q = 1,\ldots,Q \quad a > 0, b \geq 0$. Define $f' = (f'_1,\ldots,f'_Q)$. Then $\Theta(f'(F)) = a\Theta(f(F)) + b$ and therefore lex-MO solutions are invariant under (positive) affine transformations of objective functions.*

2. Let $w'_q(e) = aw_q(e) \quad q = 1,\ldots,Q \quad a > 0$. Then $f'_q = af_q \quad q = 1,\ldots,Q$. Hence lex-MO solutions are invariant under (positive) linear transformations of weights. In case the Q objectives are all bottleneck functions the invariance holds for (positive) affine transformations of weights.

Proof:

1. Obviously $f_q(F) \leq f_r(F) \Leftrightarrow af_q(F) + b \leq af_r(F) + b$ for $a > 0, b \geq 0$.

2. If f_q is a sum function:

$$f'_q(F) = \sum_{e\in F}(aw_q(e)) = a\sum_{e\in F} w_q(e) = af_q(F)$$

If f_q is a bottleneck function:

$$f'_q(F) = \max_{e\in F}(aw_q(e) + b) = a\max_{e\in F} w_q(e) + b = af_q(F) + b$$

The claim follows from these equations and the first part.

□

For sum functions invariance under (positive) affine transformations of weights depends on the cardinalities of feasible solutions, since $f'_q(F) = af_q(F) + |F|b$. Hence it holds if $|F| = const.\forall F \in \mathcal{F}$, e.g. if $\mathcal{F}$ is the set of bases of a matroid.

3 Algorithms

Now we will turn our attention to algorithms for determining lex-MO solutions. We first restrict ourselves to the case $Q = 2$ and consider two cases.
We will first shortly describe the case where f_1 and f_2 are sum functions. By Proposition 1 every lex-MO solution is also a max-ordering solution. But since the max-ordering problem for sum objectives is NP-hard, so is the lex-MO problem. This was shown for unconstrained problems in [2] and for spanning tree problems in [3]. The same is obviously also true for $Q > 2$ sum functions.

3.1 One bottleneck and one sum objective

Without loss of generality we assume that f_1 is a sum objective and f_2 is a bottleneck objective. In this case we have no more than $m = |E|$ different values of pareto solutions, since f_2 cannot attain more than m values. This fact enables us to give the following polynomial time algorithm.

Algorithm 1

Step 1 Renumber E such that $w_2(e_1) \geq \ldots \geq w_2(e_m)$
Let $t = 0, Opt = \infty, k = 0$

Step 2 $t = t + 1$

Step 3 $k = k + 1$

a) If $k \geq 2$ and $w_2(e_k) = w_2(e_{k-1})$ goto 3

b) Let $w'(e_j) = \begin{cases} \infty & j = 1, \ldots, k-1 \\ w_1(e_j) & j = k, \ldots, m \end{cases}$

c) Solve $\min_{F \in \mathcal{F}} \sum_{e \in F} w'(e)$
Denote the solution by F_t
Let $f = \max\{f_1(F_t), f_2(F_t)\}$

d) If $f < Opt$
$Opt = f, t^* = t$, goto 2
If $f > Opt, \tilde{t} = t - 1$, goto 4
If $f = Opt$ goto 2

Step 4 Solution is $F_{\tilde{t}}$ with objective value $\Theta(F_{\tilde{t}})$

Theorem 1 *Algorithm 1 is correct and calculates a lex-MO solution in $\mathcal{O}(mS(\mathcal{F}))$ steps, where $S(\mathcal{F})$ is the time complexity to solve a one criterion sum problem for feasible set $\mathcal{F}$.*

Proof:
Due to the reordering of elements in Step 1) and Step 3a) $f_2(F_t)$ has to be strictly decreasing in t. By restricting the ground set and therefore the set of feasible solutions, $f_1(F_t)$ is increasing. Therefore there exists some $t^* \geq 0$ such that

$$\max\{f_1(F_t), f_2(F_t)\} = f_2(F_t) \quad t = 1, \ldots, t^*$$

$$\max\{f_1(F_t), f_2(F_t)\} = f_1(F_t) \quad t \geq t^* + 1$$

(See Figure 2.) To see this, suppose that $\max\{f_1(F_1), f_2(F_1)\} = f_2(F_1)$. Otherwise we are done, setting $t^* = 0$ due to monotonicity of the sequences.
Now suppose there are $t_2 > t_1$ and solutions F_{t_1}, F_{t_2} such that $f_1(F_{t_1}) \geq f_2(F_{t_1}), f_1(F_{t_2}) \leq f_2(F_{t_2})$ but $f_1(F_{t_1}) \leq f_1(F_{t_2})$ i.e. for t_1 the maximum is attained for f_1 and for t_2 it is attained for f_2. $t_2 > t_1$ implies $f_2(F_{t_1}) > f_2(F_{t_2})$. But this yields

$$f_2(F_{t_1}) \leq f_1(F_{t_1}) \leq f_1(F_{t_2}) \leq f_2(F_{t_2}) < f_2(F_{t_1})$$

Due to this contradiction t^* must exist and the algorithm stops after at most m iterations. The minimal value of $\max\{f_1(F_t), f_2(F_t)\}$ can only be attained for $t \in \{t^*, \ldots, \tilde{t}\}$. It is therefore clear that the algorithm computes a max-ordering solution. But since every lex-MO solution is max-ordering optimal and due to the strictly decreasing values of $f_2(F_t)$ only $\Theta(F_{\tilde{t}})$ can be the optimal value of a lex-MO solution.

□

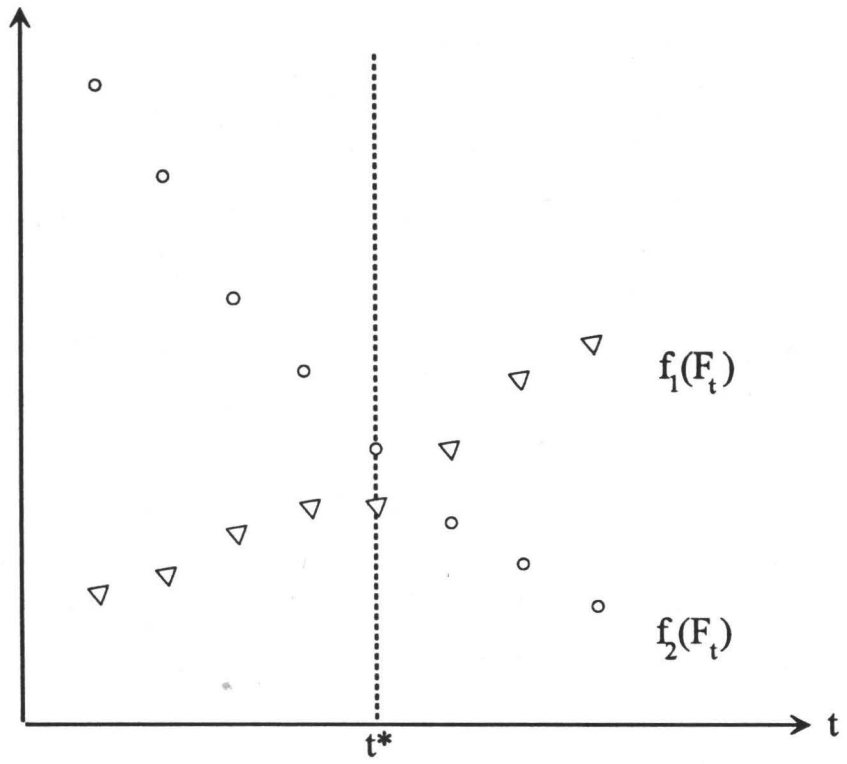

Figure 2: Sequences $f_1(F_t), f_2(F_t)$

3.2 Bottleneck functions

Here we will give an algorithm for the general case of Q bottleneck functions. It will be useful to introduce some notation for restricted max-ordering bottleneck problems (MO-bottleneck problems).

Definition 2 *Let* $i_1, \ldots, i_k \quad k \leq Q$ *be indices in* $\{1, \ldots, Q\}$ *such that* $i_j \neq i_l \quad j \neq l$. *Let* $v_1, \ldots, v_k$ *be nonnegative integers. The* restricted MO-bottleneck problem

$$MOB((i_1, \ldots, i_k), (v_1, \ldots, v_k))$$

is defined as follows

$$\min_{F \in \mathcal{F}} \max_{q \in \{1,\ldots,Q\} \setminus \{i_1,\ldots,i_k\}} w_q(e)$$

under the restriction that $f_{i_j}(F) \leq v_j \quad j = 1, \ldots, k$.

To solve restricted MO-bottleneck problems is not harder than to solve a bottleneck problem with one objective function as Proposition 6 shows.

Proposition 6 $MOB((i_1, \ldots, i_k), (v_1, \ldots, v_k))$ *can be solved by solving*

$$\min_{F \in \mathcal{F}, F \subseteq E'} \max_{e \in F} w'(e)$$

where $E' = E \setminus \{e \in E | \exists j \in \{1,\ldots,k\} : f_{i_j}(e) > v_j\}$ *and* $w'(e) = \max_{q \in \{1,\ldots,Q\}\setminus\{i_1,\ldots,i_k\}} w_q(e)$.

Proof:
For $F \in \mathcal{F}$

$$\max_{q\in\{1,\ldots,Q\}\setminus\{i_1,\ldots,i_k\}} \max_{e\in F} w_q(e) = \max_{e\in F} \max_{q\in\{1,\ldots,Q\}\setminus\{i_1,\ldots,i_k\}} w_q(e) = \max_{e\in F} w'(e)$$

Since $F \in \mathcal{F}$ is feasible for $MOB((i_1,\ldots,i_k),(v_1,\ldots,v_k))$ if and only if it does not contain any element of $\{e \in E | \exists j \in \{1,\ldots,k\} : f_{i_j}(e) > v_j\}$ the proof is complete.

□

For $k = 0$ $MOB((),())$ is the unrestricted MO-bottleneck problem. For simplicity we will use $MOB((i_1,\ldots,i_k),(v_1,\ldots,v_k))$ also to denote the optimal value of the problem.
Now we can formulate the algorithm. It is understood that $\Theta_q = -\infty$ if $q \leq 0$.

Algorithm 2

Step 1 q=1
Step 2 I $\Theta_q = \infty$
II For all $q-1$ tuples $(i_1,\ldots,i_{q-1})$ of $\{1,\ldots,Q\}$ $i_j \neq i_l, j \neq l$
If $MOB((i_1,\ldots,i_{q-2}),(\Theta_1,\ldots,\Theta_{q-2})) > \Theta_{q-1}$ next tuple
$\Theta_q = \min(\Theta_q, MOB((i_1,\ldots,i_{q-1}),(\Theta_1,\ldots,\Theta_{q-1})))$
next tuple
III $q = q+1$
IV If $q \leq Q$ goto Step 2 else STOP

Theorem 2 *Algorithm 2 is correct. At termination* $(\Theta_1,\ldots,\Theta_Q)$ *is the value of a lex-MO solution and any solution with* $MOB((i_1,\ldots,i_{Q-1}),(\Theta_1,\ldots,\Theta_{Q-1})) = \Theta_Q$ *found in the last iteration is a lex-MO solution.*

Proof:
Let $(\Theta_1,\ldots,\Theta_Q)$ be the output of the algorithm and $(\Theta_1^*,\ldots,\Theta_Q^*)$ be the optimal solution value for the lex-MO problem. We show that $\Theta_q = \Theta_q^*$ by induction on q.
The case $q = 1$ is obvious by definition of $MOB((),())$ and Proposition 1. Now suppose that for some q $\Theta_i = \Theta_i^*$ $i \leq q$. We have to show that $\Theta_{q+1} = \Theta_{q+1}^*$.
First note that any solution with $MOB((i_1,\ldots,i_{q-1}),(\Theta_1,\ldots,\Theta_{q-1})) > \Theta_q = \Theta_q^*$ cannot be optimal since the algorithm must already have encountered a solution F such that $\Theta_q(F) = \Theta_q = \Theta_q^*$. Obviously we have $\Theta_{q+1} \geq \Theta_{q+1}^*$. Now let $F \in \mathcal{F}$ be such that $(\Theta_1(F),\ldots,\Theta_{q+1}(F)) = (\Theta_1^*,\ldots,\Theta_{q+1}^*) = (f_{j_1}(F),\ldots,f_{j_{q+1}}(F))$, e.g. any optimal solution. Hence F is feasible in $MOB((j_1,\ldots,j_q),(\Theta_1^*,\ldots,\Theta_q^*)) = MOB((j_1,\ldots,j_q),(\Theta_1,\ldots,\Theta_q))$, the optimal value of which is therefore Θ_{q+1}^*. Since Θ_{q+1} is the minimum of all values $MOB((i_1,\ldots,i_q),(\Theta_1,\ldots,\Theta_q))$ computed in Step 2 II it follows that $\Theta_{q+1} = \Theta_{q+1}^*$.
Now since every solution F computed in the last iteration has by the above arguments $\Theta(F) = (\Theta_1^*,\ldots,\Theta_Q^*)$ the rest of the theorem follows.

□

Algorithm 2, however, is not necessarily polynomial. In the worst case it is necessary to solve $Q!$ restricted MO-bottleneck problems. This worst case occurs if for each q every $MOB((i_1,\ldots,i_{q-1}),(\Theta_1,\ldots,\Theta_{q-1}))$ has the same value, and Θ_q is different for each q.
On the other hand in the best case we only have to solve $\binom{Q}{2}$ such problems, namely in the case that for each q there is only one $MOB((i_1,\ldots,i_{q-1}),(\Theta_1,\ldots,\Theta_{q-1}))$ with optimal value Θ_q.
Some improvement of Algorithm 2 is easily possible by not considering any elements $e \in E$ with $w_q(e) > \Theta_{q-1}$ when solving the restricted MO-bottleneck problems.
We will now give an example for algorithm 2.

Example 2 *We consider the spanning tree problem for K_4 with edge weights given as in Figure 3.*

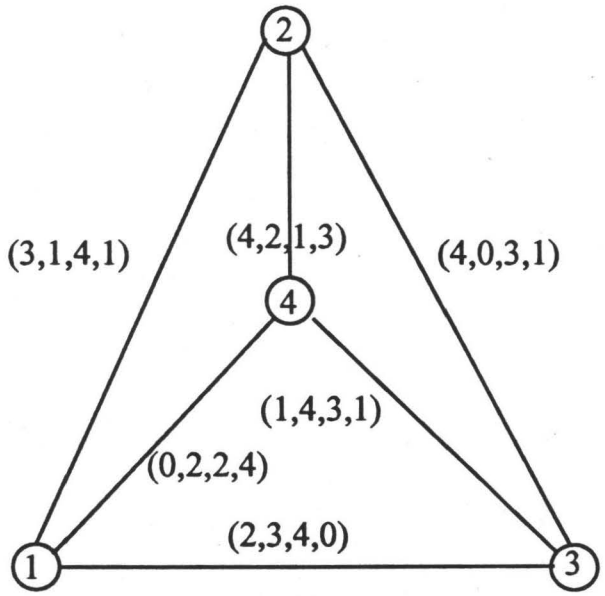

Figure 3: Graph for Example 2

In Table 2 the performance of Algorithm 2 is summarized. Edge weights $w'(e)$ of Proposition 6 are noted in the sequence $[1,2],[1,3],[1,4],[2,3],[2,4],[3,4]$ with "-" indicating edges not feasible in the restricted MO-bottleneck problem. MOB denotes the optimal values for the corresponding problem. Tuples not considered in Step 2 II as well as tuples that yield subproblems MOB that have been considered before are not recorded.

q	tuple	w'	E'	MOB	Θ_q
1	()	4 4 4 4 4 4	E	4	4
2	(1)	4 4 4 3 3 4	E	4	4
	(2)	4 4 4 4 4 3	E	4	4
	(3)	3 3 4 4 4 4	E	4	4
	(4)	4 4 2 4 4 4	E	4	4
3	(1,2)	4 4 4 3 3 3	E	4	4
	(1,3)	1 3 4 1 3 4	E	3	3
	(1,4)	4 4 2 3 2 4	E	3	3
	(2,3)	3 2 4 4 4 1	E	3	3
	(2,4)	4 4 2 4 4 3	E	4	3
	(3,4)	3 3 2 4 4 4	E	3	3
4	(1,3,2)	1 0 4 1 3 -	$E \setminus \{[3,4]\}$	3	3
	(1,3,4)	1 3 - 0 2 4	$E \setminus \{[1,4]\}$	2	2
	(1,4,2)	4 4 2 3 1 -	$E \setminus \{[3,4]\}$	3	2
	(1,4,3)	- - 2 0 2 4	$E \setminus \{[1,2],[1,3]\}$	2	2
	(2,3,1)	1 0 4 - - 1	$E \setminus \{[2,3],[2,4]\}$	1	1
	(2,3,4)	3 2 - 4 4 1	$E \setminus \{[1,4]\}$	3	1
	(3,4,1)	1 3 2 - - 4	$E \setminus \{[2,3],[2,4]\}$	3	1
	(3,4,2)	3 2 0 4 4 -	$E \setminus \{[3,4]\}$	3	1

Table 2: Algorithm 2 for the graph of Figure 3

Due to Theorem 2 any optimal solution of $MOB((2,3,1)(4,4,3))$ *is lex-MO solution. So by Proposition 6 we have to solve a bottleneck spanning tree problem for the graph of Figure 4. Hence* $F = \{[1,2],[1,3],[3,4]\}$ *is the unique optimal solution with* $\Theta(F) = (4,4,3,1)$. *Note that each of the 16 spanning trees of* K_4 *is max-ordering optimal in this example and that there are 4 pareto optimal spanning trees, namely:*

$$F = \{[1,2],[1,3],[1,4]\}, f(F) = (3,3,4,4)$$
$$F = \{[1,2],[2,3],[2,4]\}, f(F) = (4,2,4,3)$$
$$F = \{[1,2],[1,3],[3,4]\}, f(F) = (3,4,4,1)$$
$$F = \{[1,4],[2,3],[2,4]\}, f(F) = (4,2,3,4)$$

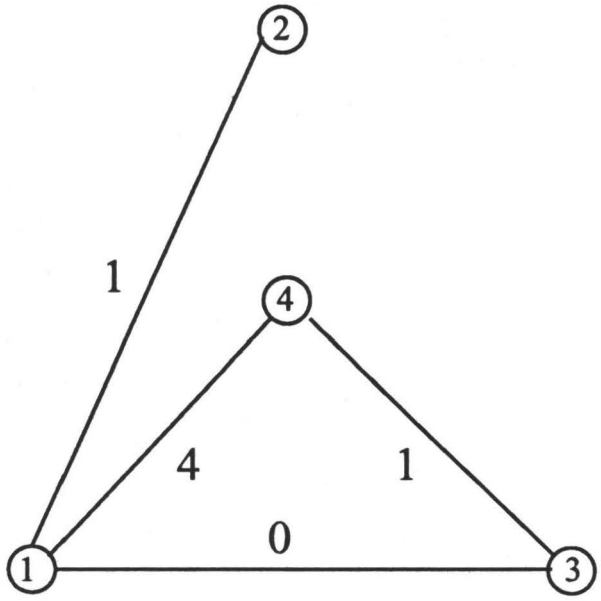

Figure 4: Finding an optimal solution in Example 2

References

[1] F. Christensen, M. Lind, and J. Tind. On the nucleolus of ntu-games defined by multiple objective linear programs. Technical Report 95/7, University of Aarhus, Department of Operations Research, 1995.

[2] S. Chung, H.W. Hamacher, F. Maffioli, and K.G. Murty. Note on combinatorial optimization with max-linear objective functions. *Discrete Applied Mathematics*, 42:139–145, 1993.

[3] H.W. Hamacher and G. Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52:209–230, 1994.

[4] P. Hansen. Bicriterion path problems. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, number 177 in Lecture Notes in Economics and Mathematical Systems, pages 109–127, 1979.

[5] E. Marchi and J.A. Oviedo. Lexicographic optimality in the multiple objective linear programming: The nucleolar solution. *European Journal of Operations Research*, 57:355–359, 1992.

[6] W. Ogryczak. Location problems from the multiple criteria perspective: Efficient solutions. Technical Report 94-019, Université Libre de Bruxelles, 1994.