Journal of the Nigerian Association of Mathematical Physics
Volume **22 (November, 2012), pp 579 - 586**

Erratum: Max-Linear Programming: Transformation From $\overline{\mathbb{R}}$ To $\mathbb{R}$

[J. Nig. Assoc. Math. Phys. Vol. 17, 215 – 222 (2010)]

A. Aminu
Department of Mathematical Sciences,
Kano University of Science and Technology, Wudil,
P.M.B 3244, Kano, Nigeria.

Some equations and symbols in this paper did not appear properly in the vol. 17 issue of the Journal of NAMP. The entire article is therefore reproduced below as it ought to appear in pages 215 – 222 (Vol. 17).

Abstract

Let $a \oplus b = \max(a,b)$ and $a \otimes b = a + b$ for $a, b \in \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$ and extend the pair of operations to matrices and vectors in the same way as in linear algebra. Max-linear programming is a problem of the form $f^T \otimes x \to \min(\text{or} \max)$ subject to $A \otimes x \oplus c = B \otimes x \oplus d$. Max-linear programs with finite entries have been considered in the literature and solution methods for both minimization and maximization problems have been developed. In this paper we consider max-linear programming problems with infinite entries and show that this problem can be transformed to the one with all input variables finite.

Keywords: - max-algebra; two-sided system; max-linear programming; pseudopolynomial.

Introduction

1.0 Max-algebra and its basic definitions

Let $a \oplus b = \max(a,b)$ and $a \otimes b = a + b$ for $a, b \in \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$ The element $-\infty$ plays the role of neutral for $\oplus$ and a null for $\otimes$. We will denote by ε the element $-\infty$ and for convenience we will use the same symbol to denote a vector or matrix whose elements are $-\infty$. For $a \in \mathrm{R}$, the symbol a^{-1} means $-a$.

Max-algebra is an analogue of linear algebra developed for the pair of operations $(\oplus, \otimes)$, extended to matrices and vectors in the same way as in linear algebra. That is if $A = (a_{ij}), B = (b_{ij})$ and $\mathrm{C} = (c_{ij})$ are matrices of compatible sizes with entries from $\overline{\mathbb{R}}$, we write $C = A \oplus B$ if $c_{ij} = a_{ij} \oplus b_{ij}$ for all i, j and $C = A \otimes B$ if $c_{ij} = \sum_k^{\oplus} a_{ik} \otimes b_{kj} = \max_k (a_{ik} \otimes b_{kj})$ for all i, j. One of the main advantages of using max-algebra is the possibility of dealing with a class of non-linear problems in a linear-like way.

Max-algebra has been studied by many authors for further reading the reader is referred to [1, 2, 4, 5, 6] and [11].

We will now summarize some standard properties of matrices and vectors in max-algebra. Identity matrix is a matrix whose all diagonal elements are 1 and all off the diagonal elements ε. We denote by I the diagonal matrix. The following holds for matrices (including vectors) A, B, C of compatible sizes over $\overline{\mathbb{R}}$ and $a \in \overline{\mathbb{R}}$

$$A \oplus B = B \oplus A$$
$$(A \oplus B) \oplus C = A \oplus (B \oplus C)$$
$$A \oplus \varepsilon = A = \varepsilon \oplus A$$
$$A \oplus B \geq A$$
$$A \otimes I = A = I \otimes A$$

$$A \otimes \varepsilon = \varepsilon = \varepsilon \otimes A$$
$$(A \oplus B) \otimes C = A \otimes C \oplus B \otimes C$$
$$A \otimes (B \oplus C) = A \otimes B \oplus A \otimes C$$
$$a \otimes (B \oplus C) = a \otimes B \oplus a \otimes C$$
$$a \otimes (B \otimes C) = B \otimes (a \otimes C)$$

Corresponding author: E-mail; abdulaamin77@yahoo.com , Tel. +234803518523

2.0 Problem formulation

Consider the following 'multi-processor interactive process' (MPIS).

Products $P_1,...,P_m$ are prepared using n processors, every processor contributing to the completion of each product by producing a partial product. It is assumed that every processor can work on all products simultaneously and that all these actions on a processor start as soon as the processor is ready to work. Let a_{ij} be the duration of the work of the jth processor needed to complete the partial product for P_i (i = 1, . . . ,m; j = 1, . . . , n). Let us denote by x_j the starting time of the jth processor (j = 1, . . . , n). Then, all partial products for P_i (i = 1, . . . ,m; j = 1, . . . , n) will be ready at $\max(a_{i1}+x_1,...,a_{in}+x_n)$. Now, suppose that independently k other processors prepare partial products for products $Q_1,...,Q_m$ and the duration and starting times are b_{ij} and y_j, respectively. Then the synchronization problem is to find starting times of all $n+k$ processors so that each pair (P_i,Q_i) (i = 1, . . . ,m) is completed at the same time. This task is equivalent to solving the system of equations

$$\max(a_{i1}+x_1,...,a_{in}+x_n)=\max(b_{i1}+y_1,...,b_{in}+y_n),(i=1,...,m).$$

It may also be required that P_i is not completed before a particular time c_i and similarly Q_i not before time d_i. Then, the equations are

$$\max(a_{i1}+x_1,...,a_{in}+x_n,c_i)=\max(b_{i1}+y_1,...,b_{in}+y_n,d_i),(i=1,...,m).$$

This system is called 'two-sided max-linear system with separated variables' and can be written in matrix-vector notation as follows:

$$A\otimes x\oplus c=B\otimes y\oplus d \qquad (2.1)$$

It is shown [6] that (1) can be transformed to the one with non separated variable

$$A\otimes x\oplus c=B\otimes x\oplus d \qquad (2.2)$$

In applications it may be required that the starting times are optimized with respect to a given criterion. In [3] the objective function is considered to be max-linear, that is

$$f(x)=f^T\otimes x=\max(f_1+x_1,...,f_n+x_n)$$

and developed a method for solving this problem for both minimization and maximization. But all the entries for both objective function and constraints are finite. If in the MPIS some processor j does not produce some partial product i then the duration a_{ij} of work of the jth processor needed to complete these partial products for $P_i(\text{or } Q_i)$ (i = 1, . . . ,m; j = 1, . . . , n) is set to ε. If under this assumption we have a non-homogeneous two-sided constraints and the objective function is max-linear then the problem is called 'max-linear programming problem over $\overline{\mathbb{R}}$'. In this paper we will consider 'max-linear programming problem over $\overline{\mathbb{R}}$ and show that this problem can be transformed to the one whose all input variables are finite and hence methods developed in [3] can be applied to find solution to this problem.

General two-sided max-linear systems have been investigated in several articles e.g [4], [6], [7], [10]. A general solution method was presented in [10], however, no complexity bound was given. In [6] an algorithm with a pseudopolynomial complexity (Alternating method) has been developed. In [4] it was shown that the solution set is generated by a finite number of vectors. An iterative method suggested in [10] assumes that finite upper and lower bounds for all variables are given. The iterative method presented in [10] makes it possible to find an approximation of the maximum solution of the given system, which satisfies the given lower and upper bounds or to find out that no such solution exists

3. Max-linear programming: existing results

Here we give a short description of methods for solving max-linear programming, for details, see [3]. We will describe minimization problem only since the method we present in paper is for the minimization problem. *Max-linear program* (MLP) is of the form $f^T\otimes x\to\min$ subject to

$$A\otimes x\oplus c=B\otimes x\oplus d, \qquad (3.1)$$

where $f=(f_1,...,f_n)^T\in\mathrm{R}^n, c=(c_1,...,c_m)^T\in\mathrm{R}^m, d=(d_1,...,d_m)^T\in\mathrm{R}^m,\quad A=(a_{ij})$ and $B=(b_{ij})\in\mathrm{R}^{m\times n}$

are given matrices and vectors with finite entries. This problem is denoted by $\mathrm{MLP}^{\min}$.

Any system of the form (3) is called "non-homogeneous max-linear system" and the set of solution of this system will be denoted by S. Any system of the form

$$E \otimes x = F \otimes x \tag{3.2}$$

is called "homogeneous max-linear system" and the set of solution of this system will be denoted by S_h.

Proposition 3.1 [3] Let $E = (A|0)$ and $F = (B|0)$ be matrices arising from A and B, respectively, by adding a zero column. If $x \in S$, then $(x|0) \in S_h$ and conversely, if $z = (z_1, z_2, ..., z_{n+1})^T \in S_h$, then $z_{n+1}^{-1} \otimes (z_1, z_2, ..., z_n)^T \in S$

Denote by $K = \max\{|a_{ij}|, |b_{ij}|, |c_i|, |d_i|, |f_j|; i \in M, j \in N\}$

Theorem 3.1 [6] Let $E = (e_{ij}), F = (f_{ij}) \in \mathrm{R}^{m \times n}$ and Γ be the greatest of the values $|e_{ij}|, |f_{ij}|, i \in M, j \in N$. There is an algorithm of complexity O($mn(m + n)\ \Gamma$)that finds an x satisfying (4) or decides that no such x exists.

Proposition 3.1 and Theorem 3.1 show that feasibility question for $\mathrm{MLP}^{\min}$ can be answered in pseudopolynomial time.

The following proposition shows that the problem of attainment of an optimal value for $\mathrm{MLP}^{\min}$ is converted to a feasibility question.

Proposition 3.2 [3] f (x) = α, for some $x \in S$ if and only if the following non-homogenous max-linear system has a solution:

$$A \otimes x \oplus c = B \otimes x \oplus d$$

$$f(x) \oplus \alpha' = f'(x) \oplus \alpha$$

where $\alpha' < \alpha$ and $f'^T(x) < f^T(x)$, where $f_j' < f_j$, for every $j \in N$.

Based on this a bisection method for finding an optimal solution to $\mathrm{MLP}^{\min}$ was developed. Before we give the algorithm we need to show the criteria for the existence of an optimal solution. Denote $\inf_{x \in S} f(x) = f^{\min}$.

Let $M^{>} = \{i \in M; c_i \geq d_i\}$ for $r \in M^{>}$ denote

$$L_r = \min_{k \in N} f_k \otimes c_r \otimes b_{rj}^{-1} \text{ and}$$

$$L = \max_{r \in M^{>}} L_r$$

Lemma 3.1 [3] If $c > d$, then $f(x) > L$ for every $x \in S$.

Theorem 3.2 [3] $f^{\min} = \varepsilon$ if and only if $c = d$

Algorithm 3.1 MAXLINMIN (max-linear minimization)

Input: $f = (f_1, ..., f_n)^T \in \mathrm{R}^n, c = (c_1, ..., c_m)^T, d = (d_1, ..., d_m)^T \in \mathrm{R}^m, c \geq d, c \neq d$, $A = (a_{ij}), B = (b_{ij}) \in \mathrm{R}^{m \times n}, \delta > 0$

Output: $x \in S$ such that $f(x) - f^{\min}(x) \leq \delta$.

1. If $L = f(x)$ for some $x \in S$, then stop ($f^{\min} = L$).
2. Find an $x^0 \in S$.
3. $L(0) := L, U(0) :=$ $f(x^0)$, $r := 0$.
4. $\alpha := \frac{1}{2}(L(r) + U(r))$
5. Check whether $f(x) = \alpha$ is satisfied by some $x \in S$ and in the positive case find one.

If yes, then $U(r + 1) := \alpha$, $L(r + 1) := L(r)$.

If not, then $U(r + 1) := U(r)$, $L(r + 1) := \alpha$.

6. $r := r + 1$.

7. If $U(r) - L(r) \le \delta$, then stop else go to 4.

Theorem 3.3 [3] Algorithm MAXLINMIN is correct and the number of iterations before termination is $O\left(\log_2 \frac{U-L}{\delta}\right)$.

The integer modification of Algorithm MAXLINMIN is given as follows as follows

Algorithm 3.2 INTEGER MAXLINMIN (integer max-linear minimization)

Input: $f = (f_1,..., f_n)^T \in \mathrm{R}^n, c = (c_1,...,c_m)^T, d = (d_1,...,d_m)^T \in \mathrm{R}^m, c \ge d, c \ne d$, $A = (a_{ij}), B = (b_{ij}) \in \mathrm{R}^{m \times n}$

Output: $x \in S^{\min} \cap \mathrm{R}^n$.

1. If $L = f(x)$ for some $x \in S^{\min} \cap \mathrm{R}^n$, then stop ($f^{\min} = L$).

2. Find an $x^0 \in S^{\min} \cap \mathrm{R}^n$.

3. $L(0) := L, U(0) :=$ $f(x^0)$, $r := 0$.

4. $\alpha := \left\lceil \frac{1}{2}\left(L(r) + U(r)\right) \right\rceil$

5. Check whether $f(x) = \alpha$ is satisfied by some $x \in S^{\min} \cap \mathrm{R}^n$ and in the positive case

   find one.

   If x exists, then $U(r + 1) := \alpha$, $L(r + 1) := L(r)$.

   If it does not, then $U(r + 1) := U(r)$, $L(r + 1) := \alpha$.

6. $r := r + 1$.

7. If $U(r) - L(r) = 1$, then stop $(U(r) = f^{\min})$ else go to 4.

Theorem 3.4 [3] Algorithm INTEGER MAXLINMIN is correct and terminates after using $O(mn(m + n)K \log K)$ operations and hence pseudopolynomial.

4.0 Max-linear programming over $\overline{\mathrm{R}}$

We consider the following problem:

$$f^T \otimes x \to \min(\text{or max}) \text{ subject to } A \otimes x \oplus c = B \otimes x \oplus d, \qquad (4.3)$$

where $f = (f_1,..., f_n)^T \in \overline{\mathrm{R}}^n, c = (c_1,...,c_m)^T \in \mathrm{R}^m, d = (d_1,...,d_m)^T \in \mathrm{R}^m$, $A = (a_{ij})$ and $B = (b_{ij}) \in \overline{\mathrm{R}}^{m \times n}$ are given matrices and vectors. We assume that $c \ge d$ otherwise swap the equations appropriately. The aim is to show that this problem can be transformed to the one with all input variables finite. We will deal with problems whose objective function is to be minimized and denote this problem by MLP.

We assume without loss of generality that $M^> \ne \emptyset$, where $M^> = \{i \in M; c_i \ge d_i\}$ (otherwise by Theorem 3.3 $f^{\min} = \varepsilon$) and denote

$$L_r = \min_{k \in N} f_k \otimes c_r \otimes b_{rj}^{-1}, b_{rj} \ne \varepsilon$$

$$L = \max_{r \in M^>} L_r$$

Theorem 4.1

If $M^{>} \neq \emptyset$ then $f(x) \geq L$ for every $x \in S$.

Proof.

If $M^{>} \neq \emptyset$ then the statement follows trivially since $L = \varepsilon$. Let $x \in S$ and $r \in M^{>}$. Then we have $(B \otimes x)_r \geq c_r$

Since $c_r \in \mathrm{R}$ for all $r \in M^{>}$, Therefore we have $x_k \geq c_r \otimes b_{rk}^{-1}, b_{rk} \neq \varepsilon$ for some $k \in N$. It follows that, $f(x) \geq f_k \otimes x_k \geq f_k \otimes c_r \otimes b_{rk}^{-1} \geq L_r$ and hence $f(x) \geq L$.

A variable x_j will be called *active* if $x_j = f(x)$, for some $j \in N$. Also, a variable will be called active on the constraint equation if the value $\max_{i \in M}(a_{ij} + x_j), \max_{i \in M}(b_{ij} + x_j)$ is attained at the term x_j respectively. We may similarly say that a coefficient is active if its corresponding variable is active.

Since all the variables x_j corresponding to ε coefficients cannot be active on any side of any equation or in the objective function while searching for an optimal solution. Therefore we can replace these ε coefficients by some sufficiently small finite values, so that the matrices A and B in (4.3) can be transformed to another one with finite elements and have the same solution set. If the ε coefficients are replaced we will therefore call the matrix a transformed-matrix. If A and B are transformed we can use the Alternating method for finding a feasible solution to (4.3) and the algorithms for solving the max-linear programs with two-sided constraints for finite entries to find an optimal solution. To do this transformation we denote for all $j \in N$:

$$h_j = \min\left(\min_{\substack{r \in M \\ a_{rj \neq \varepsilon}}} a_{rj}^{-1} \otimes c_r, \min_{\substack{r \in M \\ b_{rj \neq \varepsilon}}} b_{rj}^{-1} \otimes d_r, \underset{f_{j \neq \varepsilon}}{f_j^{-1} \otimes L} \right) \tag{4.1}$$

and $h = (h_1, h_2, ..., h_n)^T$.

Proposition 4.1

For any $x \in S$ there is an $x' \in S$ such that $x' \geq h$ and $f(x) = f(x')$.

Proof.

Let $x \in S$. It is sufficient to set $x' = x \oplus h$ since if $x_j < h_j \;\; j \in N$ then x_j is not active on any side of any equation or in the objective function. Therefore for such $j \in N$ changing x_j to h_j will not affect any of the equations or the objective function value.

It follows from Proposition 4.1 that any $x_j < h_j \;\; j \in N$ is not active on any side of any constraint equation or in the objective function. We now show how to transform matrices A and B to A' and B' respectively. Let us denote $U = |f(x)|$, where *x* is the initial feasible solution to a given problem.

$$a'_{ij} \equiv \begin{cases} a_{ij}, & \text{if } a_{ij} > \varepsilon \\ h_j - U, & \text{if } a_{ij} = \varepsilon \end{cases} \tag{4.2}$$

$$b'_{ij} \equiv \begin{cases} b_{ij}, & \text{if } \mathrm{b}_{ij} > \varepsilon \\ h_j - U, & \text{if } \mathrm{b}_{ij} = \varepsilon \end{cases} \tag{4.3}$$

Therefore the *transformed* system is $A' \otimes x \oplus c = B' \otimes x \oplus d$, where $A' = (a'_{ij}), B' = (b'_{ij}) \in \mathrm{R}^{m \times n}$ are defined in (7) and (8), $c = (c_1, c_2, ..., c_m)^T$ and $\mathrm{d} = (d_1, d_2, ..., d_m)^T \in \mathrm{R}^m$.

Recall that

$S = \{x; A \otimes x \oplus c = B \otimes x \oplus d\}$ and define $S' = \{x; A' \otimes x \oplus c = B' \otimes x \oplus d\}$.

Theorem 4.2

The sets S and S' are equal.

Proof.

To show that $S = S'$, we show $S \subseteq S'$ and $S' \subseteq S$. Let $x \in S$ and it follows that $A \otimes x \oplus c = B \otimes x \oplus d$.

Claim: $x \in S'$.

Proof of the claim:

It is clear that any row r such that $a'_{rj} = a_{rj}$ for all $j \in N$ (and therefore $b'_{rj} = b_{rj}$) will be satisfied by x. Therefore we consider rows $r \in M$ such that there exist $j \in N$ where $a'_{rj} \neq a_{rj}$ for all $j \in N$ (and therefore $b'_{rj} \neq b_{rj}$) and show that the coefficients $a'_{rj} \neq a_{rj}$ (and therefore $b'_{rj} \neq b_{rj}$) are not active. It follows from Proposition 4.1 that x_j will not be active on any side of any equation or in the objective function if $x_j < h_j$. Therefore $x \geq h$. Now we have for all $r \in M$

$$\max_{\substack{j \in N \\ a'_{rj} \neq a_{rj}}} \left(a'_{rj} + x_j\right) = \max_{\substack{j \in N \\ b'_{rj} \neq b_{rj}}} \left(b'_{rj} + x_j\right) \leq h_j - U < h_j \qquad (4.4)$$

Hence for all $r \in M$ the coefficients $a'_{rj} \neq a_{rj}$ (and therefore $b'_{rj} \neq b_{rj}$) are not active. Thus, $x \in S'$ and so $S \subseteq S'$.

Let $x \in S'$. This implies that $A' \otimes x \oplus c = B' \otimes x \oplus d$. It follows from Proposition 4.1 that $x'_j \geq h_j$ for all $j \in N$. Also it follows from (9) that $x \in S$, thus $S' \subseteq S$.

Since we have $S \subseteq S'$ and $S' \subseteq S$ therefore $S' = S$.

Corollary 4.1.

The following hold:

(a) $S^{\min} \neq \emptyset$ if and only if $S'^{\min} \neq \emptyset$

(b) $\min_{x \in S} f(x) = \min_{x \in S'} f(x)$.

Proof.

Follow straightforwardly from Theorem 4.2.

Since all entries for the matrices A' and B' are finite therefore we can use the same idea for solving max-linear programs over R.

5. An example

Consider the following max-linear programming problem (minimisation) in which

$f = (3,1,4,2,0)^T$,

$$A = \begin{bmatrix} \varepsilon & \varepsilon & 15 & 2 & 18 \\ \varepsilon & 12 & \varepsilon & 7 & 14 \\ 1 & \varepsilon & 12 & \varepsilon & \varepsilon \end{bmatrix}, \; B = \begin{bmatrix} 14 & \varepsilon & 0 & \varepsilon & 14 \\ \varepsilon & 14 & 10 & \varepsilon & 5 \\ 7 & 14 & \varepsilon & 14 & \varepsilon \end{bmatrix},$$

$$c = \begin{bmatrix} 18 \\ 5 \\ 16 \end{bmatrix}, d = \begin{bmatrix} 17 \\ 5 \\ 5 \end{bmatrix}$$

and the starting vector is $x^0 = (5,1,4,2,1)^T$.

It follows that $f(x^0) = 8$, $M^{>} = \{1,3\}$ the corresponding upper and lower bounds are

$$L = \max_{r\in M^{>}} \min_{k\in N} f_k \otimes c_r \otimes b_{rj}^{-1}, b_{rj} \neq \varepsilon$$
$$= \max(\min(7,22,4), \min(12,3,4)) = 4,$$

$$U = f(x^0) = 8.$$

Now we transform the matrices A and B as follows

$$h_j = \min\left(\min_{\substack{r\in M \\ a_{rj\neq\varepsilon}}} a_{rj}^{-1} \otimes c_r, \min_{\substack{r\in M \\ b_{rj\neq\varepsilon}}} b_{rj}^{-1} \otimes d_r, \underset{f_{j\neq\varepsilon}}{f_j^{-1}} \otimes L \right)$$

$$h_1 = \min(15, \min(3,-2), 1) = -2$$
$$h_2 = \min(-7, \min(-9,-9)) = -9$$
$$h_3 = \min(\min(3,6), \min(17,-5)) = -5$$
$$h_4 = \min(\min(16,-2), -9) = -9$$
$$h_5 = \min(\min(0,-9), \min(3,0)) = -9$$

Using (7) and (8) the transformed matrices are

$$A = \begin{bmatrix} -10 & -17 & 15 & 2 & 18 \\ -10 & 12 & -13 & 7 & 14 \\ 1 & -17 & 12 & -17 & -17 \end{bmatrix}, \quad B = \begin{bmatrix} 14 & -17 & 0 & -17 & 14 \\ -10 & 14 & 10 & -17 & 5 \\ 7 & 14 & -13 & 14 & -17 \end{bmatrix}$$

and their corresponding vectors $f = (3,1,4,2,0)^T$ and $c = \begin{bmatrix} 18 \\ 5 \\ 16 \end{bmatrix}, d = \begin{bmatrix} 17 \\ 5 \\ 5 \end{bmatrix}$ remain unchanged. The starting vector here is also $x^0 = (5,1,4,2,1)^T$. Using Algorithm MAXLINMIN, presented in Section 3 we find that the optimal solution to this transformed problem is $x = (4,0,3,2,0)^T$ with corresponding optimal value $f^{\min} = 7$

6. Conclusion

A method for converting max-linear programs with infinite entries to the one whose all entries are finite is proposed. The method can be applied to the minimization problems only, this happens due to the difficulty in finding the upper bound for the maximization case. We give a simple example that demonstrates this difficulty as follows. Suppose we want to maximize $f^T \otimes x$ subject to $A \otimes x \oplus c = B \otimes x \oplus d$ in which

$$f = (8,3,4)^T \quad A = \begin{pmatrix} \varepsilon & 0 & 0 \\ \varepsilon & 0 & -1 \\ 5 & 1 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} \varepsilon & -1 & 1 \\ \varepsilon & -1 & -2 \\ 0 & 4 & 0 \end{pmatrix}, \quad c = (5,6,7)^T \text{ and } d = (3,6,4)^T$$

and a feasible solution $x = (2,3,4)^T$

The optimal objective function value for this problem is shown to be finite (see [3]). It is clear that $f(x) = 10$. Following the idea used in [3], one could think that the upper bound can be determined as follows

$$U = \max_{i\in M} \max_{j\in N} f_j \otimes a_{ij}^{-1} \otimes c_i, a_{ij} \neq \varepsilon$$
$$= \max(\max(8, 9), \max(9, 11), \max(10, 9, 8)) = 11.$$

But this is not the case because $x = (5,6,5)^T$ is another feasible solution to this problem in which $f(x) = 13 > 11 = U$.

References

[1] F. L. Bacelli, G. Cohen, G. J. Olsder, J. P. Quadrat, Synchronization and Linearity, An Algebra for Discrete Event Systems, Wiley, Chichester (1992).

[2] P. Butkovic, Max-algebra: the linear algebra of combinatorics?, Linear Algebra & Appl., 367 (2003) 313-335.

[3] P. Butkovic, A. Aminu, Introduction to Max-linear programming, IMA Journal of Management Mathematics 20 Number 3 (2009), 233-249.

[4] P. Butkovic, G. Hegedus, An elimination method for finding all solutions of the system of linear equations over an extremal algebra, Ekonom. mat. Obzor 20 (1984) 2003-215.

[5] R. A. Cuninghame-Green, Minimax Algebra, Lecture Notes in Economics and Mathematical Systems, vol.166, Springer, Berlin, (1979).

[6] R. A. Cuninghame-Green, P. Butkoviˇc, The equation Ax = B y over (max,+), Theoret. Comput. Sci., 293 (1991) 3-12.

[7] R. A. Cuninghame-Green, K. Zimmermann, Equation with residual functions, Comment. Math. Uni. Carolina. 42(2001) 729-740.

[8] B. Heidergott, G. J. Olsder, J. van der Woude, Max-plus at work, Modelling And Analysis of Synchronized Systems: A course on Max-Plus Algebra and Its Applications, Princeton University Press, New Jersey (2006).

[9] N. N. Vorobyov, Extremal algebra of positive matrices, Elektronische Datenverarbeitung und Kybernetik 3 (1967) 39-71 (in Russian).

[10] E. A. Walkup, G. Boriello, A general linear max-plus solution technique, in: Gunawardena(Ed.), Idempotency, Cambridge, (1988) 406-415.

[11] K. Zimmermann, (1976) *Extrem'aln'ı algebra. V'yzkumna publikace 46. Ekonomicko—matematick'a laboratoˇre pˇri Ekonomicke'm u'staveˇ CˇSAV*, vol. 46. Praha. (In Czech.)