



NORTH-HOLLAND

Strong Regularity of Matrices in a Discrete Bounded Bottleneck Algebra

Katarína Cechlárová and Karol Kolesár

Katedra geometrie a algebry PF UPJŠ

Jesenná 5

041 54 Košice, Slovakia

Submitted by Richard A. Brualdi

ABSTRACT

The results concerning strong regularity of matrices over bottleneck algebras are reviewed. We extend the known conditions to the discrete bounded case and modify the known algorithms for testing strong regularity. © Elsevier Science Inc., 1997

1. INTRODUCTION

The notion of linear independence plays a crucial role in the classical linear algebra. However, in extremal structures the situation is somewhat more complicated and needs a special treatment, e.g. from the point of view of solution sets of systems of linear equations. The notions of *strong linear independence* and *strong regularity* were introduced in [8] for a lattice ordered group and further studied in [1] and [4]. In linearly ordered sets with addition replaced by maximum and multiplication by minimum (called *bottleneck algebras*), strong linear independence was studied in [2], [3], [5], [6], and [7]. It turns out that other properties of the ordering need to be taken into account, namely density and the existence or nonexistence of universal bounds. We review the known conditions for strong regularity over various types of bottleneck algebras and extend them to the discrete bounded case.

LINEAR ALGEBRA AND ITS APPLICATIONS 256:141–152 (1997)

Let us note here that a special case of a bottleneck algebra based on the closed interval $[0, 1]$ of reals is called by some authors the *fuzzy algebra*, and an extensive study of linear systems of equations over lattices (called *fuzzy relations*) can be found in [9].

2. DEFINITIONS AND BASIC PROPERTIES

The quadruple $\mathcal{B} = (B, \oplus, \otimes, \leq)$ is called a *bottleneck algebra* (BA for short) if (B, \leq) is a nonempty, linearly ordered set and \oplus, \otimes are binary operations on B defined by the formulas

$$a \oplus b = \max\{a, b\},$$

$$a \otimes b = \min\{a, b\}.$$

A BA \mathcal{B} is called

- unbounded* if for every $x \in B$ there exist $y, z \in B$ with $y < x < z$;
- bounded* if there exist $\delta \in B$ and $\Delta \in B$ such that $\delta \leq x \leq \Delta$ for every $x \in B$;
- dense* if for every $x, y \in B$ such that $x < y$ there exists $z \in B$ with $x < z < y$;
- discrete* if every $x \in B$ (in the bounded case $x < \Delta$) possesses a *successor* $S(x)$ such that $x < S(x)$ and the open interval $(x, S(x))$ is empty. In the bounded case we set $S(\Delta) = \Delta$.

The best-known examples of bottleneck algebras are the various number sets with the natural order: the reals \mathbf{R} are a dense and unbounded BA, a closed interval $[u, v]$ of reals is a dense bounded BA with $\delta = u$ and $\Delta = v$; the integers \mathbf{Z} play the role of a discrete unbounded BA; and finally a closed interval of integers is a discrete bounded BA.

The set of all m -by- n matrices over a BA \mathcal{B} will be denoted by $B(m, n)$. B_m is short for the set of column vectors $B(m, 1)$. For $\mathbf{b} \in B_m$ the notation $\mathbf{b} > \delta$ means that $b_i > \delta$ for all $i = 1, \dots, m$.

P_n will denote the set of all permutations of the set $\{1, 2, \dots, n\} = N$; id stands for the identical permutation. Two matrices are called *equivalent* if one of them can be obtained from the other by row and/or column permutations only.

Let $A \in B(m, n)$, $\mathbf{b} \in B_m$ be given. $T(A, \mathbf{b})$ will denote the set of all solutions $\mathbf{x} \in B_n$ of the linear system of equations $A \otimes \mathbf{x} = \mathbf{b}$. A matrix $A \in B(m, n)$ is said to have *strongly linearly independent columns* if

$|T(A, \mathbf{b})| = 1$ for some $\mathbf{b} \in B_m$ such that $\mathbf{b} > \delta$. (It is easy to see that equations with the right-hand side $b_i = \delta$ force all x_j with $a_{ij} > \delta$ to be equal to δ ; hence such a system can be reduced w.l.o.g. to the previous case by dropping those equations and the corresponding unknowns.) In the case $m = n$ the matrix is called *strongly regular*. $\text{Ir}(A)$ denotes the set of all $\mathbf{b} \in B_m$, $\mathbf{b} > \delta$ with $|T(A, \mathbf{b})| = 1$.

The following characterization of unique solvability was originally proved in [2] for the unbounded case and extended to bounded bottleneck algebras in [6].

THEOREM 1. *Let $A \in B(n, n)$, $\mathbf{b} \in B_n$. Then $|T(A, \mathbf{b})| = 1$ if and only if the relations*

$$a_{i\pi(i)} \geq b_i > \bigoplus_{j \in N, j \neq i} a_{i\pi(j)} \otimes b_{\pi(j)} \tag{1}$$

are satisfied by at least one $\pi \in P_n$ and equality can hold only for $b_i = \Delta$.

If the inequalities in Equation (1) are fulfilled by $\pi = \text{id}$, we say that A is in a *\mathbf{b} -normal form*. For any $\mathbf{b} \in \text{Ir}(A)$ the matrix A can easily be transformed to its \mathbf{b} -normal form using only column permutations. For $A \in B(n, n)$ and $i \in N$ let the symbol A_i stand for

$$A_i = \bigoplus_{k=1}^i \bigoplus_{j=k+1}^n a_{kj}.$$

Strongly regular matrices over a dense BA can be characterized by their equivalence to the so-called trapezoidal matrices. This was proved in [2] for the unbounded case and extended to the bounded case in [6].

DEFINITION 1. $A \in B(n, n)$ is called trapezoidal if $a_{ii} > A_i$ holds for all $i \in N$.

THEOREM 2. *Let $A \in B(n, n)$. Then a necessary condition for A to be strongly regular is the existence of a trapezoidal matrix equivalent to A . Moreover, if \leq is dense, this condition is also sufficient.*

In [3] an $O(n^2 \log n)$ algorithm was proposed for recognition of matrices equivalent to trapezoidal ones.

3. STRONGLY TRAPEZOIDAL MATRICES

From now on suppose that a bounded discrete BA \mathcal{B} is given.

DEFINITION 2. A vector $(\alpha_1, \dots, \alpha_n) \in B_n$ is called the *overdiagonal vector* of a given $A \in B(n, n)$ if

$$(i) \alpha_1 = S(A_1)$$

and for $i = 1, 2, \dots, n - 1$

- (ii) $\alpha_{i+1} = \alpha_i$ if $A_{i+1} < \alpha_i$ and $(\forall j \leq i)(\alpha_j = \alpha_i \Rightarrow a_{i+1j} < \alpha_i)$,
- (iii) $\alpha_{i+1} = S(\alpha_i)$ if $A_{i+1} < \alpha_i$ and $(\exists j \leq i)(\alpha_j = \alpha_i \ \& \ a_{i+1j} \geq \alpha_i)$,
- (iv) $\alpha_{i+1} = S(A_{i+1})$ if $A_{i+1} \geq \alpha_i$.

The overdiagonal vector $\alpha(A)$ or simply α of a matrix $A \in B(n, n)$ was defined in [5] for an unbounded discrete BA. The successor is used to ensure that the entries of α strictly increase in cases when this is necessary for the existence of a suitable right-hand-side vector \mathbf{b} fulfilling (1). In a bounded BA, when α_i or A_{i+1} is equal to Δ , this increase will not happen, so that case needs to be considered separately.

EXAMPLE 1. We compute the overdiagonal vectors for matrices over $([0, 10] \cap \mathbf{Z}, \oplus, \otimes, \leq)$. If

$$A = \begin{pmatrix} 10 & 4 & 0 \\ 4 & 8 & 6 \\ 3 & 6 & 9 \end{pmatrix},$$

then $\alpha(A) = (5, 7, 7)$, and in computing the entries of $\alpha(A)$ points (i), (iv), and (ii), respectively, of Definition 2 were used. For

$$C = \begin{pmatrix} 6 & 7 & 3 & 5 \\ 8 & 4 & 6 & 2 \\ 4 & 4 & 1 & 9 \\ 3 & 1 & 10 & 0 \end{pmatrix}$$

the overdiagonal vector is $\alpha(C) = (8, 9, 10, 10)$, using successively points (i),

(iii), (iv), and (iii) of Definition 2. And finally, for

$$D = \begin{pmatrix} 10 & 8 & 2 & 3 \\ 10 & 10 & 4 & 0 \\ 0 & 3 & 10 & 8 \\ 6 & 1 & 9 & 10 \end{pmatrix}$$

we have $\alpha(D) = (9, 10, 10, 10)$. Note that as soon as $\alpha_2 = 10 = \Delta$, all the remaining entries of α must be equal to Δ too.

The properties of the overdiagonal vector, stated in the following lemma, are implied directly by its definition.

LEMMA 1. For every $A \in B(n, n)$:

- (a) $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$. If $\alpha_r = \dots = \alpha_s$ for some $r \leq s$, then $\alpha_r = \Delta$ or for all $i, j, i \neq j, r \leq i \leq s, j \geq r$, one has $a_{ij} < \alpha_r$.
- (b) If $\alpha_r = \dots = \alpha_s < \alpha_{s+1}$, then there exists $j \geq r$ such that $a_{s+1j} > \alpha_r$.
- (c) $\alpha_i \geq A_i$ for all $i \in N$, and equality can occur only for $A_i = \Delta$.

DEFINITION 3. $A \in B(n, n)$ is said to be *strongly trapezoidal* (ST for short) if $a_{ii} \geq \alpha_i(A)$ for all $i \in N$. Equality can hold only if both $\alpha_i(A) = \Delta$ and $a_{kj} < \Delta$ for all k, j such that $k \neq j$ and $\alpha_k(A) = \alpha_j(A) = \Delta$.

Notice that obviously in the unbounded case the inequality $a_{ii} > \alpha_i(A)$ is always strict. On the other hand, in the bounded case, the ST matrix fulfills $A_i < \Delta$ for all $i \in N$.

EXAMPLE 1 (Continued). For the given matrices one can easily see that A and D are ST while C is not.

LEMMA 2. If $A \in B(n, n)$ is ST, then $\alpha(A) \in \text{Ir}(A)$.

Proof. It is sufficient to show that the inequality (1) is fulfilled with $\mathbf{b} = \alpha(A)$ and $\pi = \text{id}$.

The inequality $a_{ii} \geq \alpha_i$ follows from Definition 3. If it is not strict, then by Definition 3 $\alpha_i = \Delta$. The second part of the inequality (1) can be split into two parts. The first,

$$\alpha_i > \bigoplus_{j < i} a_{ij} \otimes \alpha_j,$$

holds because by Lemma 1 $\alpha_j \leq \alpha_i$ for $j < i$, and if there is equality, then $a_{ij} < \alpha_i$. For the second part,

$$\alpha_i > \bigoplus_{j>i} a_{ij} \otimes \alpha_j,$$

realize that $\bigoplus_{j>i} a_{ij} \otimes \alpha_j \leq \bigoplus_{i>j} a_{ij} \leq A_i$, and by Lemma 1(c) $\alpha_i > A_i$ unless $\alpha_i = \Delta$. If $\alpha_i = \Delta$, then for all $j > i$ we have $\alpha_j = \Delta$ and Definition 3 implies $a_{ij} < \Delta$, hence the inequality holds too. ■

The following lemmas are proved identically as in the discrete unbounded case in [5].

LEMMA 3. *Let $A \in B(n, n)$ and $\mathbf{b} = (b_1, \dots, b_n) \in \text{Ir}(A)$ be given. If A is in a \mathbf{b} -normal form and $b_1 \leq b_2 \leq \dots \leq b_n$, then $b_i > A_i$ for all $i \in N$.*

LEMMA 4. *Let $A \in B(n, n)$ and $\mathbf{b} \in \text{Ir}(A)$ with $b_1 \leq b_2 \leq \dots \leq b_n$ be given. If A is in a \mathbf{b} -normal form, then $b_i \geq \alpha_i(A)$ for $i = 1, 2, \dots, n$.*

LEMMA 5. *If $A \in B(n, n)$ is SR, then A is equivalent to a ST matrix.*

Now Lemmas 2 and 5 yield also for the bounded discrete case

THEOREM 3. *Let \mathcal{B} be a discrete bounded BA, $A \in B(n, n)$. Then A is strongly regular if and only if A is equivalent to a strongly trapezoidal matrix.*

4. AN ALGORITHM FOR CHECKING STRONG REGULARITY OF MATRICES IN A DISCRETE BOUNDED BOTTLENECK ALGEBRA

In this section our aim is to develop an algorithm for checking the strong regularity of matrices (i.e. for finding a trapezoidal matrix equivalent to the tested one) similar to those which have been obtained in [3] and [5] (the dense and discrete unbounded cases). The main idea of the algorithm is to try to choose the appropriate entry of the matrix for the position in the left upper corner and then repeat the whole task for the reduced matrix. Previous results suggest that this is possible if the left-upper-corner entries are chosen in such a way that the overdiagonal vector of the obtained matrix is lexicographically minimal. Here, the approach from [5] needs to be only slightly modified to take into account the existence of Δ .

For this aim we need the following notation:

$$M_i(A) = \bigoplus_{j=1}^n a_{ij},$$

$$m_i(A) = \bigoplus_{k \neq j} a_{ik},$$

where j is one of the indices satisfying $a_{ij} = M_i(A)$, and

$$\mu(A) = \min\{m_i(A) \mid M_i(A) > m_i(A)\}.$$

Definition 3 and Lemma 1 imply the following properties of ST matrices:

LEMMA 6. *Let $A \in B(n, n)$ be strongly trapezoidal. Then*

- (a) $m_i(A) < \Delta$ for all $i \in N$.
- (b) $a_{11} \geq S(m_1(A))$. If this inequality is fulfilled as equation, then $a_{11} = a_{22} = \dots = a_{nn} = \Delta$ and $a_{ij} < \Delta$ for all i, j such that $i \neq j$.
- (c) *Let A be written blockwise as*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

with $A_{11} \in B(r, r)$. Then both A_{11} and A_{22} are ST, and $a_{ii} > A_r$ for all $i > r$.

Lemma 6 includes also the assertion that $\mu(A) < \Delta$ for every ST matrix. Hence the following lemma can be proved identically to Lemma 8 in [5] for the discrete unbounded case.

LEMMA 7. *Let A be equivalent to a strongly trapezoidal matrix C . Then A is equivalent to a strongly trapezoidal matrix D such that $\alpha_1(D) = S(\mu(A))$.*

For the elimination algorithm we need some more notions. First, the d -overdiagonal vector $\alpha(A, d)$ of the given matrix and the given $d \in B$ is obtained when in Definition 2 of the overdiagonal vector $\alpha(A)$ point (i) is replaced by setting $\alpha_1 = S(A_1) \oplus d$. Then, when in Definition 3 of the strongly trapezoidal matrix the overdiagonal vector $\alpha(A)$ is replaced by the

d -overdiagonal vector $\alpha(A, d)$, the matrix is said to be d -strongly trapezoidal. Clearly, if A is d -strongly trapezoidal for an arbitrary d , it is also strongly trapezoidal.

The algorithm works as follows. As soon as the last defined $\alpha_i(A)$ is equal to Δ , all the remaining entries of $\alpha(A)$ must be equal to Δ too, and it remains only to find a permutation of entries equal to Δ such that outside it everything is less than Δ . If this is not the case, for the reduction we need to delete all the rows with the same $\alpha(A)$ and their corresponding columns simultaneously, see point (ii) in Definition 2 and Lemma 1(a). Therefore also for the following lemma the proof of Lemma 10 stated in [5] is also valid *mutatis mutandis*.

LEMMA 8. *Let $d \in B$, and suppose that $A \in B(n, n)$ can be written blockwise in the form*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where $A_{11} \in B(r, r)$, and that

- (i) $m_1(A), m_2(A), \dots, m_r(A) \leq d$,
- (ii) $m_i(A) > d$ for all $i = r + 1, \dots, n$,
- (iii) $a_{ii} > \alpha_1(A, d)$ for all $i = 1, 2, \dots, r$.

Then A is equivalent to a d -ST matrix if and only if A_{22} is equivalent to a d' -ST matrix with $d' = \alpha_1(A, d)$.

As in [3] and [5], we ensure the low complexity of the recognition algorithm for the ST matrices by ordering all the rows nonincreasingly at the beginning, which helps to avoid the repeated searching of the matrix. After reordering the rows, in the obtained matrix, say $AC = (b_{ij})$, pointers $G(i)$ and $g(i)$ show the positions of $M_i(A)$ and $m_i(A)$ in AC . At first, $G(i) = 1$ and $g(i) = 2$ for all $i \in N$. In the run of the algorithm these pointers need to be “shifted to the right” as more rows and columns are deleted, so as to still render the information valid for the reduced matrix. To ensure the correctness of this update, we shall maintain two sets R and C indicating which rows and columns are still considered in the reduced matrix. Initially, $R = C = N$. In an array of vectors $P = (p_i(j))$ we store the information concerning the reordering of A , so that $p_i(j) = k$ means that the entry b_{ij} of matrix AC corresponds to the k th entry in row i in the matrix A .

Finally, the vectors \mathbf{r} , \mathbf{c} indicate the order of the rows and columns of A , producing the desired strongly trapezoidal matrix; the variable s counts the number of entries of α defined so far.

ALGORITHM StrTrp.

Input: A matrix $A \in B(n, n)$, $n \geq 2$.

Output: $Strong = \mathbf{true}$ if A is equivalent to a strongly trapezoidal matrix, and $Strong = \mathbf{false}$ otherwise. If $Strong = \mathbf{true}$, then the vectors \mathbf{r} and \mathbf{c} contain the desired row and column permutation.

```

1. begin create  $AC, P$ ;
2.    $d := \delta$ ;  $Strong := \mathbf{false}$ ;  $R := N$ ;  $C := N$ ;
3.   for  $i := 1$  to  $n$  do begin  $G(i) := 1$ ;  $g(i) := 2$  end;
4.    $s := 0$ ; (comment: initialize)
5.   while  $n - s > 1$  do
6.     begin  $M := \{i \in R \mid b_{iG(i)} > b_{ig(i)}\}$ ;
7.     if  $M \neq \emptyset$  then  $m := \min\{b_{ig(i)} \mid i \in M\} \oplus d$  else stop;
8.      $find := \mathbf{false}$ ;
9.     for all  $i \in R$  do
10.    begin
11.      if  $b_{ig(i)} \leq m$  and  $p_i(G(i)) \in C$  and  $\{b_{iG(i)} > S(m)$ 
12.        or  $[S(m) = \Delta$  and  $b_{iG(i)} = \Delta]\}$  then
13.        begin  $R := R - \{i\}$ ;  $C := C - \{p_i(G(i))\}$ ;  $s := s + 1$ ;  $r(s) := i$ ;
14.           $c(s) := p_i(G(i))$ ;  $\alpha(s) := S(m)$ ;  $find := \mathbf{true}$ 
15.        end
16.      end;
17.    if  $n - s = 0$  then begin  $Strong := \mathbf{true}$ ; stop end;
18.    if  $S(m) = \delta$  or not( $find$ ) then stop;
19.    for all  $i \in R$  do
20.    begin
21.      while  $p_i(G(i)) \notin C$  do
22.      begin  $G(i) := G(i) + 1$ ;  $g(i) := \max\{G(i) + 1, g(i)\}$  end;
23.      while  $p_i(g(i)) \notin C$  do  $g(i) := g(i) + 1$ 
24.    end;
25.     $d := \alpha(s)$ 
26.  end;
27.  if  $n - s = 1$  then (comment:  $R = \{i\}$ ;  $C = \{j\}$ )
28.  if  $b_{ij} > S(\alpha(s))$  then
29.  begin  $\alpha(s + 1) = S(\alpha(s))$ ;  $r(s + 1) := i$ ;  $c(s + 1) := j$ ;  $Strong :=$ 
30.    true end
31.  else stop;
32. end.

```

THEOREM 4. *Algorithm StrTrp is correct and terminates after using $O(n^2 \log n)$ arithmetical operations.*

Proof. The correctness of the algorithm follows from Lemmas 6, 7, and 8. For the estimation of its computational complexity, first recall that for ordering n rows of the matrix nonincreasingly (line 1) $O(n^2 \log n)$ operations are needed. The remaining initialization (lines 2–4) takes $O(n)$ operations. The main **while** loop in lines 5–24 is repeated at most n times, because s strictly increases after each run. The number of operations in each **while** loop is $O(n)$ except only for the evaluation of $G(i)$ and $g(i)$. However, these variables rise monotonically, and so the total number of operations needed for their computation throughout the algorithm is at most $O(n^2)$. Finally, the number of operations in lines 25–28 is constant. Thus the complexity of the whole algorithm is $O(n^2 \log n)$. ■

EXAMPLE 2. We demonstrate the application of the algorithm to the matrix

$$A = \begin{pmatrix} 4 & 10 & 5 \\ 8 & 0 & 3 \\ 6 & 1 & 2 \end{pmatrix}$$

over $([0, 10] \cap \mathbf{Z}, \oplus, \otimes, \leq)$.

We have

$$AC = \begin{pmatrix} 10 & 5 & 4 \\ 8 & 3 & 0 \\ 6 & 2 & 1 \end{pmatrix}, \quad P = \begin{pmatrix} 2 & 3 & 1 \\ 1 & 3 & 2 \\ 1 & 3 & 2 \end{pmatrix}$$

The initialization gives

$$R = \{1, 2, 3\}, \quad C = \{1, 2, 3\}, \quad d = 0, \\ G(i) = 1, \quad g(i) = 2 \quad \text{for all } i \in \{1, 2, 3\}.$$

In the first run of the **while** loop we obtain

$$M = \{1, 2, 3\}, \quad m = \min\{5, 3, 2\} \oplus 0 = 2, \\ r(1) = 3, \quad c(1) = 1, \quad \alpha(1) = 3,$$

and after that $R = \{1, 2\}$, $C = \{2, 3\}$.

The update of β and arrays G and g is

$$d = 3, \quad G(1) = 1, \quad G(2) = 2, \quad g(1) = 2, \quad g(2) = 3.$$

The second run:

$$M = \{1, 2\}, \quad m = \min\{5, 0\} \oplus 3 = 3,$$

but the cycle in lines 9–14 ends with $find = \mathbf{false}$, so the algorithm terminates on line 16 with a negative answer.

EXAMPLE 3. Let

$$A = \begin{pmatrix} 4 & 10 & 9 & 7 \\ 7 & 9 & 10 & 8 \\ 6 & 7 & 6 & 9 \\ 10 & 8 & 9 & 5 \end{pmatrix}$$

over $([0, 10] \cap \mathbf{Z}, \oplus, \otimes, \leq)$. Then

$$AC = \begin{pmatrix} 10 & 9 & 7 & 4 \\ 10 & 9 & 8 & 7 \\ 9 & 7 & 6 & 6 \\ 10 & 9 & 8 & 5 \end{pmatrix}, \quad P = \begin{pmatrix} 2 & 3 & 4 & 1 \\ 3 & 2 & 4 & 1 \\ 4 & 2 & 1 & 3 \\ 1 & 3 & 2 & 4 \end{pmatrix},$$

$$R = \{1, 2, 3, 4\}, \quad C = \{1, 2, 3, 4\}, \quad d = 0,$$

$$G(i) = 1, \quad g(i) = 2 \quad \text{for all } i \in \{1, 2, 3, 4\}.$$

At first, we obtain

$$M = \{1, 2, 3, 4\}, \quad m = \min\{9, 9, 7, 9\} \oplus 0 = 7,$$

$$r(1) = 3, \quad c(1) = 4, \quad \alpha(1) = 8;$$

hence $R = \{1, 2, 4\}$, $C = \{1, 2, 3\}$. We update:

$$d = 8, \quad G(1) = G(2) = G(4) = 1, \quad g(1) = g(2) = g(4) = 2.$$

Further,

$$M = \{1, 2, 4\}, \quad m = \min\{9, 9, 9\} \oplus 8 = 9,$$

$$r(2) = 1, \quad c(2) = 2, \quad \alpha(2) = 9,$$

$$r(3) = 2, \quad c(3) = 3, \quad \alpha(3) = 9,$$

$$r(4) = 4, \quad c(4) = 1, \quad \alpha(4) = 9,$$

and the algorithm terminates on line 15 with a positive answer.

REFERENCES

- 1 P. Butkovič and F. Hervey, A condition for the strong regularity of matrices in the minimax algebra, *Discrete Appl. Math.* 11:209–222 (1985).
- 2 P. Butkovič and P. Szabó, An algorithm for checking strong regularity of matrices in the bottleneck algebra, in *Proceedings of the International Symposium on Numerical Analysis*, Polytechnic Univ. of Madrid, Spain, 1985.
- 3 P. Butkovič, K. Cechlárová, and P. Szabó, Strong linear independence in bottleneck algebra, *Linear Algebra Appl.* 94:133–155 (1987).
- 4 P. Butkovič, Strong regularity of matrices—a survey of results, *Discrete Appl. Math.* 48:45–68 (1994).
- 5 K. Cechlárová, Strong regularity of matrices in a discrete bottleneck algebra, *Linear Algebra Appl.* 128:35–50 (1990).
- 6 K. Cechlárová, Unique solvability of max-min fuzzy equations and strong regularity of matrices over fuzzy algebra, *Fuzzy Sets and Systems*, 75:165–177 (1995).
- 7 K. Cechlárová and J. Plávka, Linear independence in bottleneck algebras, *Fuzzy Sets and Systems*, 77:337–348 (1996).
- 8 R. A. Cuninghame-Green, *Minimax Algebra*, Lectures Notes in Econom. and Math. Systems 166, Springer-Verlag, Berlin, 1979.
- 9 A. Di Nola, S. Sessa, W. Pedrycz, and E. Sanchez, *Fuzzy Relation Equations and Their Applications to Knowledge Engineering*, Kluwer, Dordrecht, 1989.

Received 18 May 1995; final manuscript accepted 6 December 1995