Available online at www.sciencedirect.com
SCIENCE DIRECT®
LINEAR ALGEBRA AND ITS APPLICATIONS
ELSEVIER Linear Algebra and its Applications 369 (2003) 319–338
www.elsevier.com/locate/laa

The general trapezoidal algorithm for strongly regular max–min matrices

Martin Gavalec

Department of Information Technologies, Faculty of Informatics and Management, University of Hradec Králové, V. Nejedlého 573, Hradec Králové 50003, Czech Republic

Received 25 September 2001; accepted 18 December 2002

Submitted by R.A. Brualdi

Abstract

The problem of the strong regularity for square matrices over a general max–min algebra is considered. An $O(n^2 \log n)$ algorithm for recognition of the strong regularity of a given $n \times n$ matrix is proposed. The algorithm works without any restrictions on the underlying max–min algebra, concerning the density, or the boundedness.

AMS classification: Primary 08A72; Secondary 15A06, 15A33

Keywords: Strong regularity; Max–min algebra; Trapezoidal matrix; Fuzzy algebra

1. Introduction

In many applications, e.g. in the theory of discrete dynamic systems, fuzzy control systems, or knowledge engineering, fuzzy relation equations play an important role, see e.g. [10] (or [9], in a similar context). The solvability of fuzzy relation equations was studied in [17,18], and later in many other works, see e.g. [13,15,16].

The solvability and unique solvability of linear systems in the max–min algebra, which is one of the most important fuzzy algebras, and the related question of the strong regularity of max–min matrices was considered in [1,4,5,7]. The relation of the strong regularity to other types of regularity and linear independence is described in [2,8]. A number of interesting results were found for special cases of max–min

E-mail address: martin.gavalec@uhk.cz (M. Gavalec).

0024-3795/03/$ - see front matter © 2003 Elsevier Science Inc. All rights reserved.
doi:10.1016/S0024-3795(03)00369-0

algebras, such as discrete, dense, bounded, or unbounded algebras. The results were completed and generalized for general max–min algebras in [11,12].

An algorithm for checking strong regularity of matrices in the dense unbounded max–min algebra (bottleneck algebra) was first presented in [3]. Later, similar algorithms were described in [4,5,7], for other types of max–min algebra. Recognition of the strongly regular matrices, and of the closely related trapezoidal matrices, is important in solving the bottleneck assignment problem and the 3D axial assignment problem, see [2,6,14].

The aim of this paper is to solve the question of recognizing the strong regularity for square matrices in a general max–min algebra. An $\mathrm{O}(n^2 \log n)$ algorithm for recognition of the strong regularity of a given max–min matrix is described and its correctness is proved. The algorithm is based on the results presented in [12] and it works without any restrictions on the underlying max–min algebra, concerning the density, or the boundedness.

2. Strong regularity

By a max–min algebra we mean any linearly ordered set $(\mathscr{B}, \leqslant, \oplus, \otimes)$ together with the binary operations of maximum and minimum, denoted by $\oplus$ and $\otimes$. The quadruple will be abbreviated just to $\mathscr{B}$, if no confusion arises. The lower (upper) bound in $\mathscr{B}$ is denoted by O (by I). If B does not have the bound elements (or one of them), then the missing bound elements are formally added to $\mathscr{B}$. For any natural $n > 0$, $\mathscr{B}(n)$ denotes the set of all n-dimensional column vectors over $\mathscr{B}$, and $\mathscr{B}(m, n)$ denotes the set of all matrices of the type $m \times n$ over $\mathscr{B}$. For $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n) \in \mathscr{B}(n)$, we write $x \leqslant y$, if $x_i \leqslant y_i$ holds for all $i \in N = \{1, 2, \ldots, n\}$, and we write $x < y$, if $x \leqslant y$ and $x \neq y$. A vector $x \in \mathscr{B}(n)$ is called increasing (strictly increasing), if $x_i \leqslant x_j$ holds for every $i \leqslant j$ ($x_i < x_j$ holds for every $i < j$). The matrix operations over $\mathscr{B}$ are defined with respect to $\oplus, \otimes$, formally in the same manner as the matrix operations over any field.

A square matrix $A \in \mathscr{B}(n, n)$ in a max–min algebra $\mathscr{B}$ is *strongly regular* if there is $b \in \mathscr{B}(n)$ such that $A \otimes x = b$ is a uniquely solvable system of linear equations. The strong regularity over a general max–min algebra has been studied in [12], and its relation to the trapezoidal property was described. In this section, we present the basic notation and results, which will be later used.

For $x \in \mathscr{B}$, the *general successor* $\mathrm{GS}(x)$ of x is defined by

$$\mathrm{GS}(x) := \max\{y \in \mathscr{B};\ x \leqslant y \wedge \neg(\exists z) x < z < y\}.$$

Clearly, if x is equal to the greatest element $I \in \mathscr{B}$, then $\mathrm{GS}(x) = \mathrm{GS}(I) = I$. If $x < I$, then the set $S_x = \{y \in \mathscr{B};\ y > x\}$ is non-empty. If S_x has the least element y, then $\mathrm{GS}(x) = y$, otherwise $\mathrm{GS}(x) = x$. Thus, $\mathrm{GS}(x)$ is always well-defined. It is easy to verify that

$$\mathrm{GS}(x \oplus y) = \mathrm{GS}(x) \oplus \mathrm{GS}(y)$$

holds true for any $x, y \in \mathscr{B}$. The formula will be helpful in Section 4.

In the special case when the max–min algebra $\mathscr{B}$ is discrete, the above definition of general successor gives the same notion as in [4]. If $\mathscr{B}$ is dense, then $\mathrm{GS}(x) = x$ for every $x \in \mathscr{B}$. The above definition applies also in the case when $\mathscr{B}$ is neither discrete nor dense. We say that $x \in \mathscr{B}$ is an *upper density point* in $\mathscr{B}$, if $\mathrm{GS}(x) = x < I$.

The general successor of a vector $x \in \mathscr{B}(n)$ is a vector $y = \mathrm{GS}(x) \in \mathscr{B}(n)$ with $y_i = \mathrm{GS}(x_i)$, for all $i \in N$. If the vector $x \in \mathscr{B}(n)$ is increasing, then $\mathrm{GS}(x)$ is increasing as well. However, if $x \in \mathscr{B}(n)$ is strictly increasing, then $\mathrm{GS}(x)$ need not be strictly increasing, as we can see in the following example.

Example 1. Let us consider the max–min algebra

$$\mathscr{B} = \{0, 1, 2, 6, 7, 8\} \cup (3, 5\rangle \cup \left\{6 + \frac{1}{2^n}; n = 1, 2, 3, \ldots\right\}$$

Algebra $\mathscr{B}$ is bounded by elements $O = 0$ and $I = 8$. By the above definition, $\mathrm{GS}(x) > x$ for $x = 0, 1, 5, 7$ and for $x = 6 + 2^{-n}$, $n = 1, 2, 3, \ldots$ Namely, $\mathrm{GS}(0) = 1$, $\mathrm{GS}(1) = 2$, $\mathrm{GS}(5) = 6$, $\mathrm{GS}(7) = 8$, $\mathrm{GS}(6 + 1/2) = 7$, $\mathrm{GS}(6 + 1/4) = 6 + 1/2$, $\mathrm{GS}(6 + 1/8) = 6 + 1/4, \ldots$

For $x = 2, 6, 8$ and for $3 < x < 5$, we have $\mathrm{GS}(x) = x$. With the exception of 8, these are the upper density points in $\mathscr{B}$.

Let us put $n = 5$ and consider vectors $u = (0, 2, 4, 5, 7)^{\mathrm{T}}$, $v = (1, 2, 6, 7, 8)^{\mathrm{T}}$ in $\mathscr{B}(n)$. By definition, we get $\mathrm{GS}(u) = (1, 2, 4, 6, 8)^{\mathrm{T}}$, $\mathrm{GS}(v) = (2, 2, 6, 8, 8)^{\mathrm{T}}$. Although both vectors $u, v \in \mathscr{B}(n)$ are strictly increasing, only the general successor $\mathrm{GS}(u)$ is strictly increasing, and $\mathrm{GS}(v)$ is not.

For vectors $x, y \in \mathscr{B}(n)$, we say that x is strongly greater than y, and we write $x \sqsupset y$, when the strict inequality $x_i > y_i$ is fulfilled for every $i \in N$. Further, we say that x is almost strongly greater than y, and we write $x \sqsupseteq y$ when, for every $i \in N$, $x_i > y_i$ or $x_i = I$ holds true.

For $A \in \mathscr{B}(n, n)$, the *diagonal vector* $d(A) \in \mathscr{B}(n)$ and the *overdiagonal maximum vector* $a^{\star}(A) \in \mathscr{B}(n)$ are defined by

$$d_i(A) := a_{ii}, \quad a_i^{\star}(A) := \sum_{k=1}^{i}{}^{\oplus} \sum_{j=k+1}^{n}{}^{\oplus} a_{kj}.$$

We shall sometimes abbreviate $d(A)$, $a^{\star}(A)$ to d, $a^{\star}$, if the matrix A is clear from the context.

If $A \in \mathscr{B}(n, n)$ with $n \geqslant 2$, then we define the *overdiagonal delimiter* $\alpha(A)$ as the least element in the partially ordered set consisting of all vectors $\alpha \in \mathscr{B}(n)$ with properties

(i) $\alpha \geqslant \mathrm{GS}(a^\star)$
(ii) $i \leqslant j \Rightarrow \alpha_i \leqslant \alpha_j$
(iii) $j < i, \alpha_j \leqslant a_{ij} \Rightarrow \mathrm{GS}(\alpha_j) \leqslant \alpha_i$

for all $i, j \in N$. We say that the overdiagonal delimiter $\alpha(A)$ is *strict* in A, if for any $j, k \in N$, $j \neq k$, the equalities $\alpha_j(A) = \alpha_k(A) = I$ imply $a_{jk} < I$. Similarly as above, we shall sometimes use a shorter notation α, instead of $\alpha(A)$.

If $A \in \mathscr{B}(n, n)$ with $n = 1$, then we put $\alpha_1(A) = O$. It is easy to verify that, in the case $n \geqslant 2$, the vector α computed by the following recursion

$$\alpha_1 := \mathrm{GS}(a_1^\star),$$
$$\alpha_i := \alpha_{i-1} \oplus \mathrm{GS}(a_i^\star) \oplus \max\{\mathrm{GS}(\alpha_k);\ k < i,\ \alpha_k = \alpha_{i-1} \leqslant a_{ik}\} \quad \text{for } i > 1$$

satisfies the properties (i)–(iii) in the definition of the overdiagonal delimiter. Moreover, the recursively defined vector α is the least vector with these properties, i.e., α is the overdiagonal delimiter $\alpha(A)$. Therefore, the overdiagonal delimiter is always well-defined, by the above recursion.

We say that a matrix $A \in \mathscr{B}(n, n)$ is *generally trapezoidal*, if the overdiagonal delimiter $\alpha(A)$ is strict in A and $d(A) \sqsupseteq \alpha(A)$.

Example 2. Let $\mathscr{B}$ be the max–min algebra described in Example 1, and let us consider the following matrix $A \in \mathscr{B}(5, 5)$:

$$A = \begin{bmatrix} 7 & 0 & 0 & 0 & 0 \\ 8 & 6 & 0 & 0 & 0 \\ 4 & 5 & 8 & 5 & 6.5 \\ 5 & 6 & 7 & 8 & 1 \\ 8 & 4 & 4 & 2 & 8 \end{bmatrix}.$$

Applying the above definitions, we get $a^\star(A) = (0, 0, 6.5, 6.5, 6.5)^{\mathrm{T}}$ and $\mathrm{GS}(a^\star(A)) = (1, 1, 7, 7, 7)^{\mathrm{T}}$. The vector $\mathrm{GS}(a^\star(A))$ fulfills the conditions (i) and (ii) in the definition of the overdiagonal delimiter, but it does not satisfy the condition (iii). When the underdiagonal elements $a_{21} = 8$ and $a_{43} = 7$ are considered, then the condition (iii) gives $\alpha(A) = (1, 2, 7, 8, 8)^{\mathrm{T}}$. The overdiagonal delimiter $\alpha(A)$ is strict in A, and the diagonal vector $d(A) = (7, 6, 8, 8, 8)^{\mathrm{T}}$ is almost strongly greater than $\alpha(A)$. Therefore, the matrix A is generally trapezoidal.

For the special case of a discrete max–min algebra $\mathscr{B}$, the overdiagonal delimiter $\alpha(A)$ was defined in [4] (without any special name) and a generally trapezoidal matrix in the sense of the above definition was called strongly trapezoidal. For a dense algebra $\mathscr{B}$, the overdiagonal delimiter is equal to the overdiagonal maximum vector $\alpha(A) = a^\star(A)$ and our definition of a generally trapezoidal matrix coincides with the definition of a trapezoidal matrix in a dense algebra [2,5]. Thus, the above definition is a generalization of both cases and applies for any general case of max–min algebra $\mathscr{B}$.

Theorem 2.1 describes a close relation between generally trapezoidal matrices and strongly regular matrices over a given max–min algebra $\mathscr{B}$. For a square matrix $A \in \mathscr{B}(n, n)$ and for permutations φ, ψ on N, we denote by $A_{\varphi\psi} \in \mathscr{B}(n, n)$ the result of applying the permutation φ to the rows and the permutation ψ to the columns of the matrix A. We say that matrices A, B are equivalent if there are permutations φ, ψ with $B = A_{\varphi\psi}$.

Theorem 2.1 [12]. *Let $A \in \mathscr{B}(n, n)$. Then the following statements are equivalent*

(i) *A is strongly regular*;
(ii) *A is equivalent to a generally trapezoidal matrix, i.e. there are permutations φ, ψ such that $A_{\varphi,\psi}$ is generally trapezoidal.*

Remark 2.1. It should be emphasized, that the notion of the generalized successor heavily depends on the max–min algebra in consideration. As a consequence, the notions of the overdiagonal delimiter, general trapezoidality and the strong regularity of a matrix, are also dependent on the underlying max–min algebra.

3. The general trapezoidal algorithm

Theorem 2.1 leads to a natural question: can the strong regularity of a given square matrix over a given max–min algebra be recognized in a polynomial time? In this section we give a positive answer to this question. Namely, we describe an $\mathrm{O}(n^2 \log n)$ algorithm recognizing whether a given matrix $A \in \mathscr{B}(n, n)$ is strongly regular, or not.

Analogous results were published earlier for max–min algebras with special properties. The case when $\mathscr{B}$ is dense and unbounded was solved in [2] for matrices of type $m \times n$ and a connection with the strong linear independence of columns was described there. The results of [2] were extended to the dense and bounded case (the max–min algebra on the closed unit interval $\langle 0, 1\rangle$) in [5]. An algorithm for verifying the strong regularity of square matrices over discrete unbounded max–min algebras was presented in [4] and for discrete bounded algebras in [7]. In this paper, we consider the general case, i.e. we assume that $\mathscr{B}$ is an arbitrary max–min algebra without any restrictions.

The algorithms mentioned above verify the strong regularity of a matrix by finding permutations of rows and columns of the given matrix so that the permuted matrix contains a square submatrix, of type $n \times n$, which is “trapezoidal” in some sense (the algorithms are called trapezoidal, too). The notion of the trapezoidality is modified according to the special properties of the algebra $\mathscr{B}$.

The basic idea of all the trapezoidal algorithms is the same: chose one or several suitable entries in the given matrix, shift the corresponding rows and columns to the first positions and repeat the process with the reduced matrix without the chosen rows and columns. If this procedure can be continued for every obtained matrix, then at the end the algorithm gives a permuted matrix in a trapezoidal form. If no suitable entry can be found at some level of the recursion, then the algorithm stops with a negative result. Our version of the trapezoidal algorithm uses a generalization of the same basic idea, allowing to handle the problem over any possible max–min algebra.

The notion of the overdiagonal delimiter $\alpha(A)$ defined in Section 2 will be generalized by adding a parameter. Let $e \in \mathscr{B}$, $A \in \mathscr{B}(n, n)$. If $n \geqslant 2$, then the *e-overdiagonal delimiter* of A, notation $\alpha = \alpha(A, e)$, is defined by the following recursion

$$\begin{aligned}
&\alpha_1 := \mathrm{GS}(e \oplus a_1^{\star}(A)),\\
&\alpha_i := \alpha_{i-1} \oplus \mathrm{GS}(a_i^{\star}(A)) \oplus \max\{\mathrm{GS}(\alpha_k);\ k < i,\ \alpha_k = \alpha_{i-1} \leqslant a_{ik}\}\\
&\qquad \text{for } i > 1.
\end{aligned}$$

If $n = 1$, then we set $\alpha_1(A, e) = e$.

We say that the e-overdiagonal delimiter $\alpha(A, e)$ is *strict* in A, if for any $j, k \in N$, $j \neq k$, the equalities $\alpha_j(A, e) = \alpha_k(A, e) = I$ imply $a_{jk} < I$. We say that a matrix $A \in \mathscr{B}(n, n)$ is *e-generally trapezoidal* (e-GT for short), if the e-overdiagonal delimiter $\alpha(A, e)$ is strict in A and $d(A) \sqsupseteq \alpha(A, e)$.

Example 3. We consider the max–min algebra $\mathscr{B}$, described in Example 1, and the matrix $A \in \mathscr{B}(n, n)$ with $n = 5$, described in Example 2. For $e = 4$, we compute the e-overdiagonal delimiter $\alpha(A, e) = \alpha$. We get, by definition,

$$\begin{aligned}
\alpha_1 &= \mathrm{GS}(e \oplus a_1^{\star}(A)) = \mathrm{GS}(4 \oplus 0) = \mathrm{GS}(4) = 4,\\
\alpha_2 &= \alpha_1 \oplus \mathrm{GS}(a_2^{\star}(A)) \oplus \max\{\mathrm{GS}(\alpha_k);\ k < 2,\ \alpha_k = \alpha_1 \leqslant a_{2k}\}\\
&= 4 \oplus \mathrm{GS}(0) \oplus \max\{\mathrm{GS}(\alpha_k);\ k = 1\} = 4 \oplus 1 \oplus 4 = 4,\\
\alpha_3 &= \alpha_2 \oplus \mathrm{GS}(a_3^{\star}(A)) \oplus \max\{\mathrm{GS}(\alpha_k);\ k < 3,\ \alpha_k = \alpha_2 \leqslant a_{3k}\}\\
&= 4 \oplus \mathrm{GS}(6.5) \oplus \max\{\mathrm{GS}(\alpha_k);\ k = 1, 2\} = 4 \oplus 7 \oplus \max\{4, 4\} = 7,\\
\alpha_4 &= \alpha_3 \oplus \mathrm{GS}(a_4^{\star}(A)) \oplus \max\{\mathrm{GS}(\alpha_k);\ k < 4,\ \alpha_k = \alpha_3 \leqslant a_{4k}\}\\
&= 7 \oplus \mathrm{GS}(6.5) \oplus \max\{\mathrm{GS}(\alpha_k);\ k = 3\} = 7 \oplus 7 \oplus 8 = 8,\\
\alpha_5 &= \alpha_4 \oplus \mathrm{GS}(a_5^{\star}(A)) \oplus \max\{\mathrm{GS}(\alpha_k);\ k < 5,\ \alpha_k = \alpha_4 \leqslant a_{5k}\}\\
&= 8 \oplus \mathrm{GS}(6.5) \oplus \max\emptyset = 8 \oplus 7 \oplus 0 = 8.
\end{aligned}$$

Thus, for $e = 4$, the 4-overdiagonal delimiter is $\alpha(A, 4) = (4, 4, 7, 8, 8)$. The delimiter $\alpha(A, 4)$ is strict in A and $d(A) = (7, 6, 8, 8, 8) \sqsupseteq (4, 4, 7, 8, 8) = \alpha(A, 4)$, therefore A is 4-generally trapezoidal (4-GT). It can easily be verified in a similar way, that A is e-GT for every $e \in \mathscr{B}$, $e < 5$, but A is not e-GT for $e \geqslant 5$.

The following notation was introduced in [4]. Let $A \in \mathscr{B}(n, n)$, $n \geqslant 2$. For $i \in N$, we denote

$$M_i(A) := \sum_{j=1}^{n}{}^{\oplus} a_{ij} \quad \text{and} \quad m_i(A) := \sum_{k \neq j}{}^{\oplus} a_{ik},$$

where j is one of the indices satisfying $a_{ij} = M_i(A)$. Further, we denote

$$\mathscr{U}(A) := \{i \in N; m_i(A) < M_i(A)\},$$

and we put

$$\mu(A) := \min\{m_i(A); i \in \mathscr{U}(A)\}.$$

Remark 3.1. $M_i(A)$ is the greatest and $m_i(A)$ is the second greatest value in row i. Value $\mu(A)$ is the minimum of all second greatest values in the rows containing unique maximum entry. Clearly, $\mu(A) = I$ if and only if $\mathscr{U}(A) = \emptyset$, by definition of the minimum of an empty set.

Remark 3.2. It is clear that, for a given $A \in \mathscr{B}(n, n)$ with $n \geqslant 2$ and $e \leqslant \mu(A)$, A is generally trapezoidal if and only if A is e-GT. Similarly, for $n \geqslant 2$ and $e \leqslant \mu(A)$, the matrix A is e-GT if and only if A is $\mu(A)$-GT. For $n = 1$, the value $\mu(A)$ is not defined, and A is e-GT if and only if $a_{11} > e$ or $a_{11} = I$.

For $e \in B$, $A \in \mathscr{B}(n, n)$, $n \geqslant 2$, we define

$$R(A, e) := \{i \in \mathscr{U}(A); m_i(A) \leqslant e\},$$
$$C(A, e) := \{j \in N; (\exists i \in R(A, e)) a_{ij} = M_i(A)\}.$$

Remark 3.3. $R(A, e)$ is the set of all the rows containing unique maximum entry, for which the second greatest entry does not exceed the value e. $C(A, e)$ is the set of all the columns containing the maximum entries in rows of $R(A, e)$.

Remark 3.4. It is easy to see that $|C(A, e)| \leqslant |R(A, e)|$. Moreover, if $\mu(A) \leqslant e < I$, then $R(A, e) \neq \emptyset$.

Let $e \in B$, $A \in \mathscr{B}(n, n)$, $n \geqslant 2$ and $\mu(A) \leqslant e < I$. We say that A is *e-reducible*, if

(i) $|C(A, e)| = |R(A, e)|$
(ii) $M_i(A) > \mathrm{GS}(e)$ for every $i \in R(A, e)$.

We say that A is in an *e-reduced diagonal form*, if A is e-reducible and all entries $M_i(A)$ with $i \in R(A, e)$ are placed in the first $|R(A, e)|$ diagonal positions.

If A is e-reducible and if $|R(A,e)| < n$, then the matrix A' which remains after deleting all the rows with indices in $R(A,e)$ and all the columns with indices in $C(A,e)$, is called the *e-reduction* of A.

Example 4. Let us consider the same max–min algebra $\mathscr{B}$ and the matrix $A \in \mathscr{B}(n,n)$ with $n = 5$, as in Example 2. We compute the vectors $M(A)$, $m(A)$, the set $\mathscr{U}(A)$, and the value $\mu(A)$. The vector $M(A)$ contains the greatest row values, and $m(A)$ contains the second greatest row values, i.e. $M(A) = (7, 8, 8, 8, 8)$, $m(A) = (0, 6, 6.5, 7, 8)$. The set of indices with unique row maximum is $\mathscr{U}(A) = \{1, 2, 3, 4\}$, and the minimum of the second greatest row values, which are not the greatest ones, is $\mu(A) = \min\{0, 6, 6.5, 7\} = 0$.

We shall demonstrate the e-reducibility of the matrix A on the values $e = 5$ and $e = 7$. For $e = 5$, we get $R(A, 5) = \{1\}$ and $C(A, 5) = \{1\}$. Thus, $|R(A, 5)| = |C(A, 5)|$, and $M_1(A) = 7 > \mathrm{GS}(5) = 6$, which means that the matrix A is 5-reducible. Moreover, A is in an 5-reduced diagonal form. The corresponding e-reduction of A is the matrix

$$A' = \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 8 & 5 & 6.5 \\ 6 & 7 & 8 & 1 \\ 4 & 4 & 2 & 8 \end{bmatrix}$$

obtained from A by deleting the first row and the first column.

For $e = 7$, we get $R(A, 7) = \{1, 2, 3, 4\}$ and $C(A, 7) = \{1, 3, 4\}$ with $|R(A, 7)| \neq |C(A, 7)|$. Therefore, the matrix A is not 7-reducible.

The following three lemmas form a theoretical background for our trapezoidal algorithm.

Lemma 3.1. *Let $e \in \mathscr{B}$, $\mathrm{GS}(e) < I$. Let $A \in \mathscr{B}(n,n)$ be e-GT, let $n \geqslant 2$ and let $i \in R(A,e)$. If the matrix A is transformed to C by shifting row i and column i to the first positions, then the matrix C is e-GT, as well.*

Proof. The proof is presented in Section 4. □

Lemma 3.2. *Let $e \in \mathscr{B}$, $\mathrm{GS}(e) < I$, $n \geqslant 2$ and $A \in \mathscr{B}(n,n)$ be such that $\mu(A) \leqslant e$. Then A is equivalent to an e-GT matrix if and only if A is e-reducible and the e-reduction A' is equivalent to an e'-GT matrix with $e' = \mathrm{GS}(e)$, or if $|R(A,e)| = n$.*

Proof. The proof is presented in Section 4. □

Lemma 3.3. *Let $A \in \mathscr{B}(n,n)$, $n \geqslant 2$ and $\mathrm{GS}(e) = I$. Then A is equivalent to an e-GT matrix if and only if every row and every column of A contains exactly one value I.*

Proof. As a direct consequence of the definition of e-general trapezoidality, a matrix D is e-GT with $\mathrm{GS}(e) = I$ if and only if all diagonal elements in D are equal to I and all other elements are less than I. □

The general trapezoidal algorithm will be first described in a simpler version, which works in $\mathrm{O}(n^3)$ time. Later we show that the computational complexity of the algorithm can be reduced to $\mathrm{O}(n^2 \log n)$.

General trapezoidal algorithm—GenTrap

Input: A matrix $A \in \mathscr{B}(n, n)$.

Outputs: A Boolean variable SR with SR = true, if A is strongly regular, and SR = false otherwise. A permuted matrix $A \in \mathscr{B}(n, n)$ which is generally trapezoidal if and only if SR = true.

Step 1 (initialization). Set SR := true, $e' := O$, $k := n$, $A' := A$ (the reduced matrix).

Step 2 (case $k = 1$). If $k = 1$, then check whether $a'_{11} \leqslant e'$ and $a'_{11} < I$. If *yes*, then set SR := false. Go to Step 7.

Step 3 (case $k > 1$). Set $e' := e' \oplus \mu(A')$.

Step 4 (subcase $\mathrm{GS}(e') = I$). If $\mathrm{GS}(e') = I$, then check whether every row and every column of A' contains exactly one entry I. If *yes*, then permute the last k rows of the matrix A, so that the I entries of A' will be on the diagonal. If *not*, then set SR := false. Go to Step 7.

Step 5 (subcase $\mathrm{GS}(e') < I$). If $\mathrm{GS}(e') < I$, then set $r := |R(A', e')|$ and check from the definition whether A' is e'-reducible. If *yes*, then permute the last k rows and columns of the matrix A, so that A' will be transformed to an e'-reduced diagonal form, and set $k := k - r$. If *not*, then set SR := false and go to Step 7.

Step 6 (main loop condition). If $k > 0$, then set A' to the e'-reduction of A', set $e' := \mathrm{GS}(e')$ and go to Step 2.

Step 7. Stop.

The correctness of the algorithm GenTrap will be first proved in a more general formulation. For $e \in \mathscr{B}$, we denote by e-GenTrap such a modification of the algorithm GenTrap, in which the Step 1 sets the initial value $e' := e$, instead of $e' := O$. Further, the properties of the outputs of e-GenTrap are modified, as described in detail in the following theorem. Clearly, GenTrap is the same as O-GenTrap.

Theorem 3.4. *For any $e \in \mathscr{B}$, the algorithm e-GenTrap is correct in the sense that, for any input matrix $A \in \mathscr{B}(n, n)$, e-GenTrap stops and gives the output* SR = true *if and only if the input matrix A is equivalent to some e-GT matrix. In the positive case, the output matrix A is e-GT and it is equivalent to the input matrix.*

Proof. The assertion of the theorem will be proved by induction on n.

Induction step $n = 1$.

By the definition of $\alpha(A, e)$, we have $\alpha_1(A, e) = e$. Therefore, the matrix A is e-GT if and only if $a_{11} > e$, or if $a_{11} = I$. As, for $n = 1$, the only permutation of the index set $N = \{1\}$ is the identical one, the algorithm has to give the answer *false* if and only if $a_{11} \leqslant e$ and $a_{11} < I$. This is done in Step 2 of e-GenTrap.

Induction step $n - 1 \to n$.

We assume that $n \geqslant 2$ and the assertion of the theorem holds true for all input matrices of order $\leqslant n - 1$. Let $A \in \mathscr{B}(n, n)$ be fixed. Then the algorithm e-GenTrap sets $e' := e$, $k := n$ and $A' := A$. By assumption $n \geqslant 2$, Step 2 is skipped, and in Step 3 we get $e' := e' \oplus \mu(A')$, i.e. $e' := \max(e, \mu(A))$. In Remark 3.2, we have mentioned that, for $e \leqslant \mu(A)$, the matrix A is e-GT if and only if A is $\mu(A)$-GT. Therefore, A is e-GT if and only if it is e'-GT.

If $\mathrm{GS}(e') = I$ then, by Lemma 3.3, A is equivalent to an e'-GT matrix if and only if every row and every column of A contains exactly one entry equal to I. The equivalent e'-GT matrix contains the entries I in the diagonal positions. The corresponding row and column permutations are performed in Step 4, and if this is not possible, then the variable SR is set to *false*. Then the algorithm stops in Step 7.

If $\mathrm{GS}(e') < I$, then Step 4 is skipped, and Lemma 3.2 can be used, because after Step 3 we have $e' \geqslant \mu(A)$. By Lemma 3.2, the matrix A is equivalent to an e'-GT matrix if and only if A is e'-reducible and its e'-reduction is equivalent to an e''-GT matrix with $e'' = \mathrm{GS}(e')$, or if $|R(A, e')| = n$. The corresponding procedure is described in Step 5. If A is not e'-reducible, then the variable SR is set to *false* and the algorithm stops. If A is e'-reducible, then the rows and columns of A are permuted, so that the permuted matrix will be in an e'-reduced diagonal form, i.e. all entries $M_i(A)$ with the row indices $i \in R(A, e')$ are placed in the first $r = |R(A, e')|$ diagonal positions. Such permutations always exist, because the condition (i) from the definition of e'-reducibility gives $|C(A, e')| = |R(A, e')|$, i.e. all the uniquely maximal entries $M_i(A)$ with the row indices $i \in R(A, e')$ are placed in different columns. Moreover, by Remark 3.4 and by the assumptions $\mu(A) \leqslant e'$, $\mathrm{GS}(e') < I$, we have $R(A, e') \neq \emptyset$, i.e. $r \geqslant 1$. After setting $k := k - r$, the previous value $k = n$ is replaced by a lower value $k \leqslant n - 1$.

In Step 6 we either have $k = 0$, $r = n$, or $k > 0$, $r < n$. In the case $k = 0$, Step 6 is skipped and the algorithm stops in Step 7 with the answer SR = true, and with the matrix A permuted to an e'-GT form. In the case $k > 0$, the last k rows and the last k columns form the e'- reduction of A, which is denoted by A', and the algorithm goes back to Step 2 with the variable e' set to the new value $e'' = \mathrm{GS}(e')$.

In the rest of the proof, e' will denote the old value $e' = \max(e, \mu(A))$ used in the first run of the main loop. In the second run of Step 2, the algorithm e-GenTrap with the input value A starts the computation of the algorithm e''-GenTrap with the e'-reduction A' as the input matrix of order $k \leqslant n - 1$. The permutations in this computation, which should be performed on the e'-reduction A', are performed on the last k rows and the last k columns of the original input matrix A. Therefore, the first $n - k$ rows and columns of the original matrix A are not mutually permuted (they can be changed, however, by permutations of their last k entries, but these changes have no influence on the results of the previous computation).

Thus, from the beginning of the second run of the main loop, the algorithm e-GenTrap with the input matrix A gives an output $\mathrm{SR} = \mathrm{true}$ if and only if the algorithm e''-GenTrap with the input matrix A', equal to the e'-reduction of A, gives the output $\mathrm{SR} = \mathrm{true}$. By the induction assumption, this is equivalent to the statement that A' is equivalent to an e''-GT matrix with $e'' = \mathrm{GS}(e')$ and, by Lemma 3.2, it is further equivalent to the statement that the matrix A is equivalent to an e'-GT matrix, i.e. A is e'-GT. This holds if and only if A is e-GT, as we have mentioned at the beginning of the proof. Thus, we have verified that the algorithm e-GenTrap gives a correct result in all cases.

Finally, it can easily be observed, that in every return to Step 2, the value of the variable k is lowered by at least 1. As k takes only non-negative integer values, the algorithms e-GenTrap always stops with one of the output values $k = 0$, $k = 1$, or with $\mathrm{SR} = \mathrm{false}$. □

Theorem 3.5. *The algorithm GenTrap with an input matrix $A \in \mathscr{B}(n, n)$ stops after at most n loops, and gives the output* $\mathrm{SR} = \mathrm{true}$ *if and only if A is strongly regular. In the positive case, the output matrix A is generally trapezoidal and equivalent to the input matrix.*

Proof. The assertion is a direct consequence of Theorem 2.1 and Theorem 3.4. □

The work of the algorithm GenTrap will be demonstrated by the following example.

Example 5. Let $\mathscr{B}$ denote the max–min algebra $\mathscr{B}$ described in Example 1 and let us take the matrix $A \in \mathscr{B}(n, n)$, with $n = 5$, as an input matrix

$$A = \begin{bmatrix} 4 & 8 & 8 & 4 & 2 \\ 0 & 8 & 0 & 6 & 0 \\ 8 & 4 & 6.5 & 5 & 5 \\ 0 & 7 & 0 & 0 & 0 \\ 7 & 5 & 1 & 6 & 8 \end{bmatrix}.$$

Step 1. The algorithm GenTrap with the input A sets $\mathrm{SR} := \mathrm{true}$, $e' := O = 0$, $k := n = 5$, $A' := A$.

Step 2. The step is skipped, because $k \neq 1$.

Step 3. It is set $e' := e' \oplus \mu(A') = 0 \oplus 0 = 0$.

Step 4. The step is skipped, because $\mathrm{GS}(e') = \mathrm{GS}(0) = 1 \neq I = 8$.

Step 5. The algorithm has to verify, whether A' is e'-reducible. First, the sets $R(A', e') = \{4\}$ and $C(A', e') = \{2\}$ are computed, and $r := |R(A', e')| = 1$ is found. As $|R(A', e')| = |C(A', e')|$ and $M_4(A') = 7 > \mathrm{GS}(e')$ holds true, the matrix A' is e'-reducible. By shifting the row 4 and the column 2 to the first positions in A, the matrix $A = A'$ is transformed to the form

$$A := \begin{bmatrix} 7 & 0 & 0 & 0 & 0 \\ 8 & 4 & 8 & 4 & 2 \\ 8 & 0 & 0 & 6 & 0 \\ 4 & 8 & 6.5 & 5 & 5 \\ 5 & 7 & 1 & 6 & 8 \end{bmatrix}$$

and a new value $k := k - r = 5 - 1 = 4$ is assigned.

Step 6. As $k > 0$, the variable A' is set to the e'-reduction of A' and $e' := \mathrm{GS}(e') = \mathrm{GS}(0) = 1$. Thus, we get

$$A' := \begin{bmatrix} 4 & 8 & 4 & 2 \\ 0 & 0 & 6 & 0 \\ 8 & 6.5 & 5 & 5 \\ 7 & 1 & 6 & 8 \end{bmatrix}.$$

Then the algorithm begins the second run of the main loop by returning back to Step 2, with variables e', k and A' set to new values.

In the second run, Step 2 is skipped again, because $k = 4 \neq 1$. In Step 3, the algorithm sets $e' := e' \oplus \mu(A') = 1 \oplus 0 = 1$. Step 4 is skipped, because $\mathrm{GS}(e') = \mathrm{GS}(1) = 2 \neq I = 8$.

Step 5 (second run). The algorithm verifies, whether A' is 1-reducible. The sets $R(A', e') = \{2\}$ and $C(A', e') = \{3\}$ are found, and the variable r is set to $|R(A', e')| = 1$. As $|R(A', e')| = |C(A', e')|$ and $M_2(A') = 6 > \mathrm{GS}(1) = 2$, the matrix A' is 1-reducible. By permuting the rows and the columns of the matrix A in such a way that the row 2 and the column 3 in A' are shifted to the first positions in A', we get

$$A := \begin{bmatrix} 7 & 0 & 0 & 0 & 0 \\ 8 & 6 & 0 & 0 & 0 \\ 8 & 4 & 4 & 8 & 2 \\ 4 & 5 & 8 & 6.5 & 5 \\ 5 & 6 & 7 & 1 & 8 \end{bmatrix}, \quad A' := \begin{bmatrix} 6 & 0 & 0 & 0 \\ 4 & 4 & 8 & 2 \\ 5 & 8 & 6.5 & 5 \\ 6 & 7 & 1 & 8 \end{bmatrix}.$$

Then a new value $k := k - r = 4 - 1 = 3$ is assigned.

Step 6 (second run). As $k > 0$, the variable A' is set to the 1-reduction of A'. Therefore, the third run of the main loop begins with values $k = 3$, $e' := \mathrm{GS}(1) = 2$ and

$$A' := \begin{bmatrix} 4 & 8 & 2 \\ 8 & 6.5 & 5 \\ 7 & 1 & 8 \end{bmatrix}.$$

In the third run, similarly as above, Step 2 is skipped, then the algorithm sets $e' := e' \oplus \mu(A') = 2 \oplus 4 = 4$ in Step 3 and it skips Step 4.

Step 5 (third run). Verifying, whether A' is 4-reducible, the algorithm finds $R(A', e') = \{1\}$ and $C(A', e') = \{2\}$, and assigns the value $|R(A', e')| = 1$ to the variable r.

The formulas $|R(A', e')| = |C(A', e')|$ and $M_1(A') = 8 > \mathrm{GS}(4) = 4$ imply that the matrix A' is 4-reducible. The corresponding permutation of the last three columns of the matrix A shifts the column 2 in A' to the first position in A', with the result

$$A := \begin{bmatrix} 7 & 0 & 0 & 0 & 0 \\ 8 & 6 & 0 & 0 & 0 \\ 8 & 4 & 8 & 4 & 2 \\ 4 & 5 & 6.5 & 8 & 5 \\ 5 & 6 & 1 & 7 & 8 \end{bmatrix}, \quad A' := \begin{bmatrix} 8 & 4 & 2 \\ 6.5 & 8 & 5 \\ 1 & 7 & 8 \end{bmatrix}$$

and $k := k - r = 3 - 1 = 2$ is set.

Step 6 (third run). The variable k is still positive, therefore A' is set to the 4-reduction of A'. The fourth run of the main loop begins with values $k = 2$, $e' := \mathrm{GS}(4) = 4$ and

$$A' := \begin{bmatrix} 8 & 5 \\ 7 & 8 \end{bmatrix}.$$

In the fourth run, the algorithm skips Step 2, sets $e' := e' \oplus \mu(A') = 4 \oplus 5 = 5$ in Step 3 and skips Step 4.

Step 5 (fourth run). The algorithm finds $R(A', e') = \{1\}$ and $C(A', e') = \{1\}$, and sets $r = |R(A', e')| = 1$. By $|R(A', e')| = |C(A', e')|$ and $M_1(A') = 8 > \mathrm{GS}(5) = 6$, we see that the matrix A' is 5-reducible. Moreover, A' is in an 5-reduced form, therefore, no permutations of rows and columns of A are necessary, and the algorithm only sets $k := k - r = 2 - 1 = 1$.

Step 6 (fourth run). The variable A' is set to the 5-reduction of A'. Thus, for the fifth run of the main loop we have the values $k = 1$, $e' := \mathrm{GS}(5) = 6$ and $A' := [8]$.

In the fifth run, Step 2 is used, because $k = 1$. As $a'_{11} = 8 = I$, the value SR = true remains unchanged and the algorithm goes to Step 7, where it stops.

The result of the computation shows that the input matrix is strongly regular and the permuted output matrix A is generally trapezoidal.

Remark 3.5. We may notice that, besides the output matrix computed in Example 5, there exist further matrices which are generally trapezoidal and are equivalent to the input matrix. One of them is, e.g., the matrix considered in Example 2. Clearly, all such matrices must be mutually equivalent.

Theorem 3.6. *There is an* $\mathrm{O}(n^2 \log n)$ *algorithm which, for every matrix* $A \in \mathscr{B}(n, n)$, *recognizes whether A is strongly regular, or not. Moreover, if the input matrix A is strongly regular, the algorithm finds a matrix which is generally trapezoidal and is equivalent to the matrix A.*

Proof. We shall use the method described in [2] and used also in [4,7]. Taking Theorem 3.5 as a base, we modify the work of the algorithm GenTrap so that it avoids the repeated searches for the maximum entry and the second greatest entry in every row of the reduced matrix. This can be simply done by arranging first the elements in each row of the input matrix in a non-increasing order, so that the first two greatest elements are always situated at the beginning of the row. The arranging of n rows takes $\mathrm{O}(n^2 \log n)$ time. The orderings of all rows are stored separately using pointers, and the input matrix A will not be changed. The permuted matrix is stored in the form of the actual row and column permutation.

The ordering of rows will be updated at the end of the main loop, according to the shifting of the rows and columns of the reduced matrix. The updating in each loop is made, in every row of the reduced matrix, by switching the pointer that corresponds to the column containing the diagonal element, which is left out of the reduced matrix. These are $\mathrm{O}(n)$ operations for each diagonal element, i.e. $\mathrm{O}(n^2)$ operations in total.

The initialization takes a constant time, and the remaining operations in each loop can be performed in $\mathrm{O}(n)$ time, i.e. in $\mathrm{O}(n^2)$ time, for all loops. Thus, the total computational complexity of the modified GenTrap is $\mathrm{O}(n^2 \log n)$. $\square$

Remark 3.6. When the max–min algebra $\mathscr{B}$ is discrete, i.e. when every element $x \in \mathscr{B}$, $x < I$ has a successor $S(x) > x$, then we can use the algorithm STRTP described in [4] (for unbounded $\mathscr{B}$), or StrTrp in [7] (if $\mathscr{B}$ is bounded). However, correct results require a slight modification of these algorithms to ensure proper processing of the case when the reduced matrix has order 1.

4. Proofs of two lemmas

The rather technical proofs of two key lemmas are presented in this section.

Proof of Lemma 3.1. Let $e \in \mathscr{B}$, $\mathrm{GS}(e) < I$. Let $A \in \mathscr{B}(n, n)$ be e-GT, let $n \geqslant 2$ and let $i \in R(A, e)$. We assume that the matrix A is transformed to C by shifting row i and column i to the first positions. Our aim is to prove that the matrix C is e-GT, too.

The case $i = 1$ is trivial. For the rest of the proof, let $i \in N$ be fixed with $1 < i \leqslant n$, $m_i(A) < M_i(A)$ and $m_i(A) \leqslant e$.

It is evident that values $M_j(A)$ and $m_j(A)$ are not influenced by column permutations. For example, we have $M_1(C) = M_i(A)$ and $m_1(C) = m_i(A)$. The proof will be presented in five claims.

Claim 1. $\alpha_1(C, e) = \mathrm{GS}(e)$.

Proof. By assumption, A is e-GT, which implies that $d_i(A) = I \geqslant M_i(A) > m_i(A)$, or $d_i(A) > \alpha_i(A, e) \geqslant \mathrm{GS}(e) \geqslant m_i(A)$. In both cases we have $d_i(A) > m_i(A)$, i.e. $d_1(C) > m_1(C)$. Therefore, $a_1^\star(C) = m_1(C)$, which implies $\alpha_1(C, e) = \mathrm{GS}(e \oplus a_1^\star(C)) = \mathrm{GS}(e)$. $\square$

Claim 2. *$\alpha_j(C,e) \leqslant \alpha_{j-1}(A,e)$ for all j, $1 < j \leqslant i$.*

Proof. We can easily see that inequalities

$$a^{\star}_j(C) \leqslant m_i(A) \oplus a^{\star}_{j-1}(A), \tag{4.1}$$

$$\mathrm{GS}(m_i(A)) \leqslant \mathrm{GS}(e) \leqslant \alpha_{j-1}(C,e) \tag{4.2}$$

hold true for all j, $1 < j \leqslant i$. Claim will be proved by induction on j.
Induction step $j = 2$.

We consider two cases:

(a) $c_{21} < \alpha_1(C,e)$, i.e. $a_{1i} < \mathrm{GS}(e)$, and
(b) $c_{21} \geqslant \alpha_1(C,e)$, i.e. $a_{1i} \geqslant \mathrm{GS}(e)$.

In case (a) we have $\alpha_2(C,e) = \alpha_1(C,e) \oplus \mathrm{GS}(a^{\star}_2(C)) \leqslant \mathrm{GS}(e) \oplus \mathrm{GS}(m_i(A)) \oplus \mathrm{GS}(a^{\star}_1(A)) = \alpha_1(A,e) \oplus \mathrm{GS}(m_i(A)) = \alpha_1(A,e)$, in view of (4.1) and (4.2).

In case (b), the inequality $a_{1i} \geqslant \mathrm{GS}(e)$ implies $a^{\star}_1(A) \geqslant a_{1i} \geqslant \mathrm{GS}(e) \geqslant e$, i.e. $\mathrm{GS}(a^{\star}_1(A)) \geqslant GS^{(2)}(e) \geqslant \mathrm{GS}(e)$. Thus, using Claim 1 and (4.1), we have $\alpha_2(C, e) = \mathrm{GS}(\alpha_1(C,e)) \oplus \mathrm{GS}(a^{\star}_2(C)) \leqslant GS^{(2)}(e) \oplus \mathrm{GS}(m_i(A)) \oplus \mathrm{GS}(a^{\star}_1(A)) = \mathrm{GS}(a^{\star}_1(A)) = \mathrm{GS}(e) \oplus \mathrm{GS}(a^{\star}_1(A)) = \alpha_1(A,e)$.
Induction step $j-1 \to j$.

Let $2 < j \leqslant i$ and let us assume that

$$\alpha_k(C,e) \leqslant \alpha_{k-1}(A,e) \tag{4.3}$$

holds true for all k with $1 < k < j$. We consider two cases:

(a) $c_{jk} < \alpha_k(C,e)$ for all $k < j$ with $\alpha_k(C,e) = \alpha_{j-1}(C,e)$, and
(b) $c_{jk} \geqslant \alpha_k(C,e) = \alpha_{j-1}(C,e)$ for some k with $k < j$.

By (4.1) and (4.2), the inequalities

$$\mathrm{GS}(a^{\star}_j(C)) \leqslant \mathrm{GS}(m_i(A)) \oplus \mathrm{GS}(a^{\star}_{j-1}(A)) \leqslant \alpha_{j-1}(A,e) \tag{4.4}$$

hold in both cases.

In case (a) we have $\alpha_j(C,e) = \alpha_{j-1}(C,e) \oplus \mathrm{GS}(a^{\star}_j(C)) \leqslant \alpha_{j-2}(A,e) \oplus \alpha_{j-1}(A,e) = \alpha_{j-1}(A,e)$, by (4.3) and (4.4).

In case (b) we have $\alpha_j(C,e) = \mathrm{GS}(\alpha_{j-1}(C,e)) \oplus \mathrm{GS}(a^{\star}_j(C))$. In view of (4.4), it remains to show that

$$\mathrm{GS}(\alpha_{j-1}(C,e)) \leqslant \alpha_{j-1}(A,e). \tag{4.5}$$

Due to the assumption (4.3), $\alpha_{j-1}(C,e) \leqslant \alpha_{j-2}(A,e) \leqslant \alpha_{j-1}(A,e)$ holds true. If at least one of the inequalities is strict, then (4.5) is fulfilled. Thus, we may assume that $\alpha_{j-1}(C,e) = \alpha_{j-2}(A,e) = \alpha_{j-1}(A,e)$. We shall consider two subcases: (b1) $k = 1$, and (b2) $1 < k < j$.

In subcase (b1) we use inequalities $a^\star_{j-1}(A) \geqslant a_{j-1i} = c_{j1} \geqslant \alpha_1(C, e) = \alpha_{j-1}(C, e)$, which imply $\mathrm{GS}(\alpha_{j-1}(C, e)) \leqslant \mathrm{GS}(a^\star_{j-1}(A)) \leqslant \alpha_{j-1}(A, e)$.

In subcase (b2) we have $a_{j-1k-1} = c_{jk} \geqslant \alpha_k(C, e) = \alpha_{j-1}(C, e) = \alpha_{j-2}(A, e) \geqslant \alpha_{k-1}(A, e) \geqslant \alpha_k(C, e)$, which implies $a_{j-1k-1} \geqslant \alpha_{k-1}(A, e) = \alpha_{j-2}(A, e)$.

The definition of the diagonal delimiter $\alpha(A, e)$ gives $\alpha_{j-1}(A, e) = \mathrm{GS}(\alpha_{j-2}(A, e)) \oplus \mathrm{GS}(a^\star_{j-1}(A))$. Using (4.3) we get $\mathrm{GS}(\alpha_{j-1}(C, e)) = \mathrm{GS}(\alpha_k(C, e)) \leqslant \mathrm{GS}(\alpha_{k-1}(A, e)) = \mathrm{GS}(\alpha_{j-2}(A, e)) \leqslant \alpha_{j-1}(A, e)$. □

Claim 3. *$\alpha_j(C, e) \leqslant \alpha_j(A, e)$ for all j, $i \leqslant j \leqslant n$.*

Proof. Similarly as in the proof of Claim 2, we can easily see that inequalities

$$a^\star_j(C) \leqslant m_i(A) \oplus a^\star_j(A), \tag{4.6}$$

$$\mathrm{GS}(m_i(A)) \leqslant \mathrm{GS}(e) \leqslant \alpha_j(A, e) \tag{4.7}$$

hold true for all j, $i \leqslant j \leqslant n$. Claim will be proved by induction on j.
Induction step $j = i$.

At this initial step, the assertion is a consequence of Claim 2.
Induction step $j - 1 \to j$.

Let $i < j \leqslant n$ and let us assume that

$$\alpha_k(C, e) \leqslant \alpha_k(A, e) \tag{4.8}$$

holds true for all k with $i \leqslant k < j$. We shall consider two cases:

(a) $c_{jk} < \alpha_k(C, e)$ for all $k < j$ with $\alpha_k(C, e) = \alpha_{j-1}(C, e)$, and
(b) $c_{jk} \geqslant \alpha_k(C, e) = \alpha_{j-1}(C, e)$ for some $k < j$.

By (4.6) and (4.7), the inequalities

$$\mathrm{GS}(a^\star_j(C)) \leqslant \mathrm{GS}(m_i(A)) \oplus \mathrm{GS}(a^\star_j(A)) \leqslant \alpha_j(A, e) \tag{4.9}$$

hold in both cases.

In case (a) we have $\alpha_j(C, e) = \alpha_{j-1}(C, e) \oplus \mathrm{GS}(a^\star_j(C)) \leqslant \alpha_{j-1}(A, e) \oplus \alpha_j(A, e) = \alpha_j(A, e)$, by (4.8) and (4.9).

In case (b) we have $\alpha_j(C, e) = \mathrm{GS}(\alpha_{j-1}(C, e)) \oplus \mathrm{GS}(a^\star_j(C))$. By (4.9), it is sufficient to prove that

$$\mathrm{GS}(\alpha_{j-1}(C, e)) \leqslant \alpha_j(A, e). \tag{4.10}$$

In view of the assumption (4.8), $\alpha_{j-1}(C, e) \leqslant \alpha_{j-1}(A, e) \leqslant \alpha_j(A, e)$ holds true. If at least one of the inequalities is strict, then (4.10) is fulfilled. Thus, we may assume that $\alpha_{j-1}(C, e) = \alpha_{j-1}(A, e) = \alpha_j(A, e)$. We shall consider three subcases: (b1) $k = 1$, (b2) $1 < k \leqslant i$, and (b3) $i < k < j$.

In subcase (b1) we have $a_{ji} = c_{j1} \geqslant \alpha_1(C, e) = \alpha_{j-1}(C, e) = \alpha_{j-1}(A, e) \geqslant \mathrm{GS}(e) = \alpha_1(C, e)$, which implies $\alpha_{j-1}(A, e) = \mathrm{GS}(e)$ and $\alpha_{j-1}(A, e) \geqslant \alpha_i(A, e) \geqslant \mathrm{GS}(e) = \alpha_{j-1}(A, e)$. Therefore, $a_{ji} \geqslant \alpha_i(A, e) = \alpha_{j-1}(A, e)$.

In subcase (b2) we have $a_{jk-1} = c_{jk} \geqslant \alpha_k(C, e) = \alpha_{j-1}(C, e) = \alpha_{j-1}(A, e) \geqslant \alpha_{k-1}(A, e) \geqslant \alpha_k(C, e)$, which implies $a_{jk-1} \geqslant \alpha_{k-1}(A, e) = \alpha_{j-1}(A, e)$.

Finally, in subcase (b3) we have $a_{jk} = c_{jk} \geqslant \alpha_k(C, e) = \alpha_{j-1}(C, e) = \alpha_{j-1}(A, e) \geqslant \alpha_k(A, e) \geqslant \alpha_k(C, e)$, which implies $a_{jk} \geqslant \alpha_k(A, e) = \alpha_{j-1}(A, e)$.

In all three subcases the definition of diagonal delimiter $\alpha(A, e)$ gives $\alpha_j(A, e) = \mathrm{GS}(\alpha_{j-1}(A, e)) \oplus \mathrm{GS}(a_j^\star(A))$. Using (4.8) we get $\mathrm{GS}(\alpha_{j-1}(C, e)) \leqslant \mathrm{GS}(\alpha_{j-1}(A, e)) \leqslant \alpha_j(A, e)$. $\square$

Claim 4. $d(C) \sqsupseteq \alpha(C, e)$.

Proof. If $d_1(C) < I$, then we have $d_1(C) = d_i(A) > \alpha_i(A, e) \geqslant \mathrm{GS}(e) = \alpha_1(C, e)$, using Claim 1. Further, if $d_j(C) < I$, for some j with $1 < j \leqslant i$, then we have $d_j(C) = d_{j-1}(A) > \alpha_{j-1}(A, e) \geqslant \alpha_j(C, e)$, using Claim 2. Finally, if $d_j(C) < I$, for some j with $i < j \leqslant n$, then $d_j(C) = d_j(A) > \alpha_j(A, e) \geqslant \alpha_j(C, e)$ holds true, by Claim 3. $\square$

Claim 5. $\alpha(C, e)$ *is strict in* C.

Proof. We shall show that for any $j, k \in N$, $j < k$, the equalities $\alpha_j(C, e) = \alpha_k(C, e) = I$ imply $c_{jk} < I$ and $c_{kj} < I$. If $j = 1$, then we have $\mathrm{GS}(e) = \alpha_1(C, e) = I$, which is in contradiction with the assumption $\mathrm{GS}(e) < I$.

We shall consider three cases:

(a) $1 < j < k \leqslant i$,
(b) $1 < j \leqslant i < k$, and
(c) $i < j < k$.

In case (a) we have $\alpha_j(C, e) = \alpha_{j-1}(A, e) = I$ and $\alpha_k(C, e) = \alpha_{k-1}(A, e) = I$, by Claim 2. As $\alpha(A, e)$ is strict in A, we get $c_{jk} = a_{j-1k-1} < I$ and $c_{kj} = a_{k-1j-1} < I$.

In case (b) we have $\alpha_j(C, e) = \alpha_{j-1}(A, e) = I$ and $\alpha_k(C, e) = \alpha_k(A, e) = I$, by Claims 2 and 3. Thus, we get $c_{jk} = a_{j-1k} < I$ and $c_{kj} = a_{kj-1} < I$.

Finally, in case (c) we get $\alpha_j(C, e) = \alpha_j(A, e) = I$ and $\alpha_k(C, e) = \alpha_k(A, e) = I$, by Claim 3. Similarly as above, we get $c_{jk} = a_{jk} < I$ and $c_{kj} = a_{kj} < I$. $\square$

Proof of Lemma 3.2. Let $e \in \mathscr{B}$, $\mathrm{GS}(e) < I$, let $n \geqslant 2$. Let $A \in \mathscr{B}(n, n)$ with $\mu(A) \leqslant e$. We have to prove that the matrix A is equivalent to an e-GT matrix if and only if A is e-reducible and the e-reduction A' is equivalent to an e'-GT matrix with $e' = \mathrm{GS}(e)$, or if $|R(A, e)| = n$.

Let A be equivalent to an e-GT matrix. As row and column permutations preserve properties (i), (ii) in the definition of e-reducibility, we may assume without any loss of generality, that A itself is e-GT. By definition of the e-overdiagonal delimiter $\alpha(A, e)$, we have $\alpha_i(A, e) \geqslant \mathrm{GS}(e)$ for every i, $1 \leqslant i \leqslant n$. This implies

$d_i(A) > \alpha_i(A, e) \geqslant \mathrm{GS}(e)$, or $d_i(A) = I > \mathrm{GS}(e)$, i.e. $d_i(A) > \mathrm{GS}(e)$, for every i, $1 \leqslant i \leqslant n$. Then $m_i(A) \leqslant \mathrm{GS}(e) < d_i(A)$ holds true for all $i \in R(A, e)$. Due to the strict inequality $m_i(A) < d_i(A)$ and the fact that $m_i(A)$ is the second greatest value in row i, the diagonal element $d_i(A)$ must be equal to $M_i(A)$. Therefore, every $i \in R(A, e)$ belongs to $C(A, e)$, which implies $R(A, e) \subseteq C(A, e)$ and $|R(A, e)| \leqslant |C(A, e)|$. On the other hand, the converse inequality $|C(A, e)| \leqslant |R(A, e)|$ holds by Remark 3.4. Thus, $|C(A, e)| = |R(A, e)|$ holds true. We have shown that the matrix A is e-reducible.

To complete the proof of the 'only if' implication, it remains to prove that either $|R(A, e)| = n$, or the e-reduction A' is equivalent to an e'-GT matrix with $e' = \mathrm{GS}(e)$. By assumption $\mu(A) \leqslant e < I$, the set $R(A, e)$ is non-empty, i.e. $r := |R(A, e)| \geqslant 1$. We have shown above that $d_i(A) = M_i(A)$ holds for every $i \in R(A, e)$. By Lemma 3.1, the elements of $R(A, e)$ can be shifted to first r positions, by simultaneous permutations on rows and columns of the matrix A, while preserving the property of being e-GT. In other words, there is a permutation φ in N such that the matrix $D := A_{\varphi\varphi}$ is e-GT, $R(D, e) = \{1, 2, \ldots, r\}$ and φ preserves the ordering of rows and columns with indices not belonging to $R(A, e)$. Then the matrix D has an e-reduced diagonal form. If $r = n$, then $|R(A, e) = n|$. If $r < n$, then the matrix D can be written in a block form

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix},$$

where $D_{11} \in \mathscr{B}(r, r)$, and the submatrix D_{22} is equal to the e-reduction A' of the matrix A. Now, it remains only to show that D_{22} is equivalent to an e'-GT matrix with $e' = \mathrm{GS}(e)$. In fact, we shall even show that the matrix D_{22} itself is e'-GT (it is a consequence of our assumption that A itself is e-GT).

Case 1. If $\alpha_r(D, e) > e'$, then $\alpha_r(D, e) \geqslant \mathrm{GS}(e')$, and $\alpha_i(D, e) \geqslant \mathrm{GS}(e')$ holds for every i, $r + 1 \leqslant i \leqslant n$. Thus, D_{22} is an e'-GT matrix.

Case 2. If $\alpha_r(D, e) = e'$, then $\alpha_1(D, e) = \alpha_2(D, e) = \cdots = \alpha_r(D, e) = e'$. By assumption on D, we have $m_{r+1}(D) > e$, i.e. $M_{r+1}(D) \geqslant m_{r+1}(D) \geqslant e'$. Let us choose $k, l \in N$, $k \neq l$, with $d_{r+1k} = M_{r+1}(D)$, $d_{r+1l} = m_{r+1}(D)$. If $\max(k, l) > r + 1$, then $a^\star_{r+1}(D) \geqslant e'$, which implies $\alpha_{r+1}(D, e) \geqslant \mathrm{GS}(a^\star_{r+1}(D)) \geqslant \mathrm{GS}(e')$. On the other hand, if $h := \min(k, l) < r + 1$, then we have $d_{r+1h} \geqslant \alpha_h(D, e) = e'$ and, by definition of the e-overdiagonal delimiter $\alpha(D, e)$, we get $\alpha_{r+1}(D, e) \geqslant \mathrm{GS}(\alpha_r(D, e)) = \mathrm{GS}(e')$. Thus, for any choice of k, l we have $\alpha_{r+1}(D, e) \geqslant \mathrm{GS}(e')$, which implies $\alpha_i(D, e) \geqslant \mathrm{GS}(e')$ for every i, $r + 1 \leqslant i \leqslant n$. Therefore, D_{22} is an e'-GT matrix.

For the proof of the converse implication, let us assume first that the matrix A is e-reducible and its e-reduction A' is equivalent to an e'-GT matrix of order $n - r$ with $e' = \mathrm{GS}(e)$ and $r := |R(A, e)| < n$. Applying Lemma 3.1 at most r times, we can show that A is equivalent to a matrix D in an e-reduced diagonal form, containing A' as a right-lower corner submatrix D_{22}. From the definition of the e-reducibility and from the definition of $R(A, e)$ we have $d_i(D) = M_i(D) > \mathrm{GS}(e) \geqslant$

$\max_{j \leqslant i} m_j(A) \geqslant a_i^{\star}(D)$. Then the definition of the parametrized overdiagonal delimiter gives $\alpha_i(D, e) = \mathrm{GS}(e) = e'$, for every $i = 1, 2, \ldots, r$.

Further, the submatrix $D_{22} = A'$ is equivalent to an e'-GT matrix of order $n - r$. Therefore, the matrix D can be transformed, by suitable permutations on the last $n - r$ rows and columns, to a matrix C in an e-reduced diagonal form, in which the right-lower corner submatrix C_{22} of order $n - r$ is e'-GT. Then, for every $i = 1, 2, \ldots, r$, the inequality $d_i(C) = d_i(D) > \mathrm{GS}(e) = e' = \alpha_i(D, e) = \alpha_i(C, e)$ holds true, and, for $i = r + 1, \ldots, n$, either the inequality $d_i(C) = d_{i-r}(C_{22}) > \alpha_{i-r}(C_{22}, e') = \alpha_i(C, e)$, or the equality $d_i(C) = d_{i-r}(C_{22}) = I$ is fulfilled. The above equality $\alpha_{i-r}(C_{22}, e') = \alpha_i(C, e)$ follows directly from the definition of the parametrized overdiagonal delimiter and from the equalities $\alpha_1(C, e) = \alpha_2(C, e) = \cdots = \alpha_r(C, e) = e'$ (in some cases, the inequality is a strict one, e.g. if $\alpha_1(C_{22}, e') = \mathrm{GS}(e') > e'$ and $\alpha_{r+1}(C, e) = \alpha_r(C, e) = e'$). Thus, C is an e-GT matrix equivalent to A.

On the other hand, let us assume that the matrix A is e-reducible and $|R(A, e)| = n$. Similarly as above, A is equivalent to an e-reduced diagonal matrix D with $|R(D, e)| = n$. Then we have $d_i(D) = M_i(D) > \mathrm{GS}(e) = \alpha_i(D, e)$ for every $i \in N$, i.e. D is an e-GT matrix equivalent to A. $\square$

Acknowledgements

This work was supported by Slovak Scientific Grant Agency, # 1/6055/99. I express my gratitude to the referee for his valuable comments which helped to improve the presentation of this paper.

References

[1] P. Butkovič, Strong regularity of matrices—a survey of results, Discrete Appl. Math. 48 (1994) 45–68.

[2] P. Butkovič, K. Cechlárová, P. Szabó, Strong linear independence in bottleneck algebra, Linear Algebra Appl. 94 (1987) 133–155.

[3] P. Butkovič, P. Szabó, An algorithm for checking strong regularity of matrices in the bottleneck algebra, in: Proceedings of the International Symposium on Numerical Analysis, Polytechnic University of Madrid, Spain, 1985.

[4] K. Cechlárová, Strong regularity of matrices in a discrete bottleneck algebra, Linear Algebra Appl. 128 (1990) 35–50.

[5] K. Cechlárová, Unique solvability of max–min fuzzy equations and strong regularity of matrices over fuzzy algebra, Fuzzy Sets and Systems 75 (1995) 165–177.

[6] K. Cechlárová, Trapezoidal matrices and the bottleneck assignment problem, Discrete Appl. Math. 58 (1995) 111–116.

[7] K. Cechlárová, K. Kolesár, Strong regularity of matrices in a discrete bounded bottleneck algebra, Linear Algebra Appl. 256 (1997) 141–152.

[8] K. Cechlárová, J. Plávka, Linear independence in bottleneck algebras, Fuzzy Sets and Systems 77 (1996) 337–348.
[9] R.A. Cuninghame-Green, Minimax algebra, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, 1979.
[10] A. Di Nola, S. Sessa, W. Pedrycz, E. Sanchez, Fuzzy Relation Equations and Their Applications to Knowledge Engineering, Kluwer, Dordrecht, 1989.
[11] M. Gavalec, Solvability and unique solvability of max–min fuzzy equations, Fuzzy Sets and Systems 124 (2001) 385–393.
[12] M. Gavalec, J. Plávka, Strong regularity of matrices in general max–min algebra, Linear Algebra Appl., to appear.
[13] S. Gottwald, Fuzzy Sets and Fuzzy Logic, Vieweg Verlag, Braunschweig, 1993.
[14] B. Klinz, G.J. Woeginger, A new efficiently solvable special case of the three-dimensional axial bottleneck assignment problem, in: Combinatorics and Computer Science, Lecture Notes in Computer Science, Springer, Berlin, 1996 (Brest, 1995, pp. 150–162).
[15] I. Perfilieva, A. Tonis, Consistency of models for approximate reasoning based on fuzzy logic, in: Proceedings of the 4th European Congress on IT & Soft Computing, Aachen, 1996, pp. 651–655.
[16] I. Perfilieva, A. Tonis, Compatibility of systems of fuzzy relation equations, Int. J. General Systems 29 (2000) 511–528.
[17] W. Pedrycz, Identification in fuzzy systems, IEEE Trans. Systems, Man Cybernetics 14 (1984) 361–366.
[18] E. Sanchez, Resolution of composite fuzzy relation equations, Inform. Control 30 (1976) 38–48.