NORTH-HOLLAND

Extremal Eigenproblem for Bivalent Matrices

R. A. Cuninghame-Green and P. Butkovič

School of Mathematics and Statistics
The University of Birmingham
Birmingham B15 2TT, United Kingdom

Submitted by Richard A. Brualdi

ABSTRACT

For a general $n \times n$ real matrix (a_{ij}), standard $O(n^3)$ algorithms exist to find $\lambda, x_1, \ldots, x_n$ such that

$$\max_{j=1,\ldots,n} (a_{ij} + x_j) = \lambda + x_i \qquad (i = 1, \ldots, n).$$

It is known that λ is unique and equals the maximum cycle mean of (a_{ij}). We consider the case when all the elements a_{ij} take values in the real binary set $\{0, 1\}$, and we present algorithms which determine $\lambda, x_1, \ldots, x_n$ in $O(m + n)$ time, where m is the number of nonzero elements of (a_{ij}). We show that these algorithms may in fact be applied to bivalent matrices over any linearly ordered, commutative radicable group.

1. BACKGROUND TO THE PROBLEM

Let $\mathcal{G} = (G, \otimes, \leq)$ be a linearly ordered, commutative group with neutral element e. We suppose that $\mathcal{G}$ is radicable, i.e., for every integer $t \geq 1$ and $a \in G$ there exists a (necessarily unique) $b \in G$ such that $b^t = a$. We write $b = a^{1/t}$.

Throughout the paper $n \geq 1$ is a given integer. Denote the set of $n \times n$ matrices over G by M_n. We introduce a further binary operation on $\mathcal{G}$ by the formula

$$a \oplus b = \max(a, b) \qquad \text{for all} \qquad a, b \in G.$$

Let $\oplus, \otimes$ be extended to the matrix-vector algebra over G by direct analogy to conventional linear algebra.

LINEAR ALGEBRA AND ITS APPLICATIONS 222:77–89 (1995)

0024-3795/95/$9.50
SSDI 0024-3795(93)00285-8

Let $A = (a_{ij}) \in M_n$. The problem of finding $x \in G^n$, $\lambda \in G$ satisfying

$$A \otimes x = \lambda \otimes x \tag{1.1}$$

is called an *extremal eigenproblem* corresponding to the matrix A; here λ and x are usually called the *extremal eigenvalue* and *extremal eigenvector* of A, respectively, but the word "extremal" will be omitted throughout the paper. This problem was treated by several authors during the sixties [2, 10]: a survey of the results concerning this and similar eigenproblems can be found for example in [11]. We summarize next some of the main results.

First we introduce the necessary notation. Let $N = \{1, 2, \dots, n\}$, and C_n be the set of cyclic permutations of nonempty subsets of N. For $\sigma = (i_1, i_2, \dots, i_\ell) \in C_n$ we denote ℓ by $\ell(\sigma)$ and

$$\begin{aligned} w_A(\sigma) &= a_{i_1 i_2} \otimes a_{i_2 i_3} \otimes \cdots \otimes a_{i_\ell i_1}, \\ \mu_A(\sigma) &= [w_A(\sigma)]^{1/\ell(\sigma)} \\ \lambda(A) &= \sum_{\sigma \in C_n}^{\oplus} \mu_A(\sigma). \end{aligned}$$

where $\sum^{\otimes}$ denotes iterated use of the operation $\oplus$.

THEOREM 1.1. *Let $A \in M_n$. Then $\lambda(A)$ is the unique eigenvalue of A.*

PROOF. Can be found in [3]. ∎

The problem of finding $\lambda(A)$ is called *the maximum-cycle-mean problem* and has been studied by several authors (e.g. [1–7]). Various algorithms are known, that of Karp [6] having the best worst-case performance $[O(n^3)]$.

For $B \in M_n$ we denote by $\Delta(B)$ the matrix

$$B \oplus B^2 \oplus \cdots \oplus B^n,$$

where B^s stands for the s-fold iterated product $B \otimes B \otimes \cdots \otimes B$.

Let $A_\lambda = [\lambda(A)]^{-1} \otimes A$. It is known [3] that the matrix $\Delta(A_\lambda)$ contains at least one column the diagonal element of which is e. Every such column is called a *fundamental eigenvector* of the matrix A. The set of all fundamental eigenvectors will be denoted by F_A, and $q = |F_A|$. We say that x, $y \in F_A$ are *equivalent* if $x = \alpha \otimes y$ for some $\alpha \in G$. In what follows $\mathrm{sp}(A)$ denotes the set of all eigenvectors of A. This set is called the *eigenspace* of A.

THEOREM 1.2. *Let $A \in M_n$. Then*

$$\mathrm{sp}(A) = \left\{ \sum_{i=1}^{q} {}^{\oplus} \alpha_i \otimes g_i;\ \alpha_i \in G,\ g_i \in F_A,\ i = 1, \ldots, q \right\}.$$

PROOF. Can be found in [3]. ∎

It follows from the definition of equivalent fundamental eigenvectors that F_A in Theorem 1.2 can be replaced by any maximal set F'_A of fundamental eigenvectors such that no two of them are equivalent. Every such set F'_A will be called a *complete set of generators* (of the eigenspace).

The symbol D_A stands for a complete, arc-weighted digraph associated with A. Its node set is N, and the weight of the arc (i, j) is a_{ij}. Throughout the paper, *cycle* in a digraph means elementary cycle or loop, and *path* means nontrivial path, i.e., containing at least one arc. Evidently, there is a one-to-one correspondence between cycles of D_A and elements of C_n. Therefore, we will use the same notation as well as the concept of weight for both.

A cycle $\sigma \in C_n$ is called *optimal* if $\mu_A(\sigma) = \lambda(A)$; a node in D_A is called an *eigennode* if it is on at least one optimal cycle; E_A stands for the set of eigennodes in D_A.

Let $g_1, g_2, \ldots, g_n$ denote the columns of the matrix $\Delta(A_\lambda)$.

THEOREM 1.3.

(a) *$j \in E_A$ if and only if $g_j \in F_A$.*

(b) *g_i, g_j are equivalent members of F_A if and only if eigennodes i, j lie in the same strongly connected component of the digraph arising from D by deleting all arcs and nodes not belonging to an optimal cycle.*

PROOF. Can be found in [3]. ∎

Let the elements of $\Delta(A_\lambda)$ be (ξ_{ij}). It follows from the definition of $\Delta(A_\lambda)$ that ξ_{ij} is the weight of a heaviest path from i to j in D_{A_λ}. Hence $\Delta(A_\lambda)$ can be computed in $O(n^3)$ operations using the Floyd-Warshall algorithm [7]. By trivial search and comparisons one can then find a complete set of fundamental eigenvectors among the columns of $\Delta(A_\lambda)$ in $O(n^3)$ operations.

2. THE EIGENVALUE IN THE BIVALENT CASE

The aim of this paper is to investigate the above eigenproblem in the case when A is a bivalent matrix, i.e., $a_{ij} \in \{a, b\} \subseteq G$ for all $i, j \in N$. It turns out, however, that we may immediately specialize the discussion to the case where $\mathcal{G} = (\mathbb{R}, +, \leq)$ and $\{a, b\} = \{0, 1\}$. The proof of this is most conveniently left until Section 5. In the present and following two sections, therefore, M_n will denote the set of real zero-one matrices taken over $(\mathbb{R}, +, \leq)$.

Hence for $A = (a_{ij}) \in M_n$, $\sigma = (i_1, i_2, \ldots, i_\ell) \in C_n$ we have

$$\begin{aligned} w_A(\sigma) &= a_{i_1 i_2} + a_{i_2 i_3} + \cdots + a_{i_\ell i_1}, \\ \mu_A(\sigma) &= \frac{w_A(\sigma)}{\ell(\sigma)} \end{aligned}$$

and

$$\lambda(A) = \max_{\sigma \in C_n} \frac{w_A(\sigma)}{\ell(\sigma)}.$$

Clearly, $0 \leq \lambda(A) \leq 1$ for all $A \in M_n$, and $\lambda(A) = 0$ if and only if A is a zero matrix.

For $A = (a_{ij}) \in M_n$ we denote by G_A the (unweighted) digraph with node set N in which $j \in \Gamma(i)$ if and only if $a_{ij} = 1$, where Γ denotes the usual *successor map* on the nodes of a digraph, i.e., $j \in \Gamma(i)$ if and only if there is an arc directed from i to j. Evidently G_A is a subgraph of D_A, so if π is any path in G_A, we may regard π also as a path in D_A. According to weight, arcs in D_A will be called *zero arcs* or *unity arcs* respectively.

One can easily see that $\lambda(A) < 1$ if and only if G_A is acyclic, and in this case we say also that A is *acyclic*; otherwise A is called *cyclic*. Hence a standard algorithm for checking the acyclicity of a digraph (see for example [7]) can be used in order to decide whether $\lambda(A) = 1$ or $\lambda(A) < 1$. By a slight modification, this algorithm can be used for a fast computation of the actual value of $\lambda(A)$. The algorithm is based on the evident property that in every acyclic digraph at least one node without a successor exists. Its unmodified version (for checking the acyclicity) consists in successive removals of such nodes, whereas our modification removes *all* such nodes at once. More precisely, define inductively

$$N_1 = \{i \in N;\ \Gamma(i) = \emptyset\}, \tag{2-1}$$

and for $k > 1$ for which N_{k-1} is defined and $\bigcup_{j=1}^{k-1} N_j \neq N$ define

$$N_k = \left\{ i \in N - \bigcup_{j=1}^{k-1} N_j;\ \Gamma(i) \subseteq \bigcup_{j=1}^{k-1} N_j \right\}. \tag{2-2}$$

Clearly, if G_A is acyclic, N_k is never empty, and

$$N = \bigcup_{j=1}^{d} N_j \qquad \text{for some} \quad d \geq 1.$$

And if $d > 1$,

$$N_i \bigcap N_j = \emptyset \qquad \text{for} \quad 1 \leq i < j \leq d. \tag{2-3}$$

The number d, which we also denote by $d(A)$, will be called the *decomposition number* of the acyclic matrix A. Clearly, $d(A) = 1$ if and only if $A = 0$ or, equivalently, G_A has no arcs.

We now reproduce the algorithm, as presented in [1].

ALGORITHM DECOMPOSITION
Input: $A = (a_{ij}) \in M_n$.
Output: Sets $N_1, \ldots, N_d$.

1. (Initialization).

$$N := \{1, 2, \ldots, n\}, \quad I := \emptyset, \quad k := 1, \quad z_i := \sum_{j \in N} a_{ij}\ (i = 1, \ldots, n).$$

2. $N_k := \{i \in N - I;\ z_i = 0\}$.
3. If $N_k = \emptyset$ then stop. [$\lambda(A) = 1$; A is cyclic.]
4. $I := I \cup N_k$.
   If $I = N$ then stop. [A is acyclic; $d(A) = k$.]
5. For all $i \in N - I$ do $z_i := z_i - \sum_{j \in N_k} a_{ij}$.
6. $k := k + 1$.
   Go to 2^0.

The following propositions are straightforward and have been proved in [1].

PROPOSITION 2.1. *Algorithm* DECOMPOSITION *is correct and has an implementation which terminates after $O(n^2)$ operations.* ∎

PROPOSITION 2.2. *Let $A \in M_n$ be acyclic and nonzero, and let $k \in \{1, \dots, d-1\}$. Then $\Gamma(i) \cap N_k \neq \emptyset$ for all $i \in N_{k+1}$.* ■

(Proposition 2.2 can be interpreted as follows: Every row in the shaded submatrices in Figure 1 is nonzero.)

PROPOSITION 2.3. *Let $A \in M_n$ be acyclic and nonzero. Then*

(i) *the greatest path length attained in G_A is $d(A) - 1$,*
(ii) *at least one path of the maximum length $d(A) - 1$ runs from each node in N_d and each such path runs to a node in N_1,*
(iii) $\lambda(A) = [d(A) - 1]/d(A)$. ■

As a corollary, for every $A \in M_n$ we have

$$\lambda(A) \in \left\{0, \frac{1}{2}, \frac{2}{3}, \dots, \frac{n-1}{n}, 1\right\}.$$

This result was first proved by J. Plávka [8].

We now show that an alternative implementation of Algorithm DECOMPOSITION leads to a computational complexity $O(n+m)$, where

$$m = |\{(i,j);\ a_{ij} = 1\}|.$$

For this purpose, we use a helpful data structure. Specifically, we assume that A is given in the form of n *column lists* C_j $(j = 1, \dots, n)$. The first entry in the list C_j is c_j; the number of nonzero elements in column j of A. If $c_j > 0$, then there follow c_j entries i_s^j $(s = 1, \dots, c_j)$, one for each such nonzero element, denoting that column j has a unity on row i_s^j.

Thus if

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix},$$

then the lists are

$$C_1 = \{2; 2, 3\}, \qquad C_2 = \{1; 3\}, \qquad C_3 = \{0;\}.$$

From the specification $\{C_1, \dots, C_n\}$ we may derive the n *rowsums* r_i $(i = 1, \dots, n)$ by the following straightforward procedure.

PROCEDURE ROWSUMS
Input: Parameter n; Lists $C_1, \ldots, C_n$.
Output: Rowsums $r_1, \ldots, r_n$.

1. for $i = 1$ to n do $r_i := 0$;
2. for $j = 1$ to n do
   if $c_j > 0$ then do
      for $s = 1$ to c_j do
         $i := i_s^j, \; r_i := r_i + 1$;
3. end

PROPOSITION 2.4. *The computational complexity of Procedure* ROWSUMS *is* $O(m + n)$.

PROOF. In step 2, each nonzero element of A, and each column, is considered exactly once, while step 1 is clearly $O(n)$. ∎

The following modification of Algorithm DECOMPOSITION uses the output of Procedure ROWSUMS. The parameters $n_1, \ldots, n_d$ give the cardinalities of the sets $N_1, \ldots, N_d$, whilst the parameter n_0 counts the nodes so far dealt with.

ALGORITHM FASTDECOMPOSITION
Input: Parameter n; lists $C_1, \ldots, C_n$; rowsums $r_1, \ldots, r_n$.
Output: Parameters d, λ; sets $N_1, \ldots, N_d$; parameters $n_1, \ldots, n_d$.

1. (Initialization). $d := 0, \; k := 1, N_1 := \emptyset, n_1 := 0, n_0 := 0$;
2. for $i = 1, \ldots, n$ do
   if $r_i = 0$ then do
      $N_1 := N_1 \cup \{i\}, \; n_1 := n_1 + 1$;
3. if $n_k = 0$ then do
   $\lambda := 1$, stop;
4. $n_0 := n_0 + n_k$;
   if $n_0 = n$ then do
      $d := k, \; \lambda := (k-1)/k$, stop;
5. $N_{k+1} := \emptyset, \; n_{k+1} := 0$,
   for all $j \in N_k$ do
      if $c_j > 0$ then do
         for all $s = 1$ to c_j do
            $i := i_s^j, \; r_i := r_i - 1$,
            if $r_i = 0$ then do
               $N_{k+1} := N_{k+1} \cup \{i\}, \; n_{k+1} := n_{k+1} + 1$;
6. $k := k + 1$, goto 3;

PROPOSITION 2.5. *Algorithm* FASTDECOMPOSITION *is correct and terminates after* $O(m+n)$ *operations.*

PROOF. Correctness follows as for Algorithm DECOMPOSITION. Step 2 is clearly $O(n)$. Total complexity $O(m+n)$ follows from the fact that each nonzero element of A, and each node, is considered exactly once in step 5, while steps 3 and 4 are clearly executed at most n and $d \leq n$ times each. ■

3. EIGENSPACE OF ACYCLIC MATRICES

We will show that the sets $N_1, \ldots, N_d$ produced by Algorithm FASTDECOMPOSITION can also be used for describing the whole eigenspace of A. The construction (2-1), (2-2), together with Proposition 2.2, characterizes the structure of G_A for nonzero acyclic A: each arc is directed from a higher- to a lower-indexed N_k, and each node in each N_{k+1} has at least one arc directed from it to a node in N_k (see Figure 2).

If $i \in N$, we denote by $h(i)$ the index k for which $i \in N_k$. By inserting the instruction $h(i) := k+1$ at the end of step 5, it is clear Algorithm FASTDECOMPOSITION will furnish the values of $h(i)$.

PROPOSITION 3.1. *Let A be acyclic and nonzero. If $i, j \in N$ and a path from i to j exists in G_A, then $h(i) > h(j)$. If $r > s$, then the longest path in G_A from a node in N_r to a node in N_s has length $r - s$. The length of the longest path in G_A from any node $i \notin N_1$, is $h(i)-1$, and the greatest path length attained in G_A is $d(A) - 1$.*

PROOF. Immediate from the discussion in Section 2. ■

The inequality $r > s$ does not imply, in general, that for every $i \in N_s$ there exists a path from some node in N_r to i of length $r - s$, nor the existence of such a path of any length at all.

For $r, s \in \{1, 2, \ldots, d\}$, $r > s$, and $V \subseteq N_r$, we define

$$N_s(V) = \{i \in N_s; \text{ a path form a node in } V \text{ to } i \text{ of length } r-s \text{ in } G_A \text{ exists}\}. \quad (3\text{-}1)$$

Further, we extend this notation to the case $r{=}s$ by defining $N_r(V){=}V$.

PROPOSITION 3.2. *Let A be acyclic and nonzero, and let $i \in N - N_d$. Then the greatest length of a path in G_A terminating in i does not exceed $d - h(i)$. Moreover, its length is $d - h(i)$ if and only if $i \in N_{h(i)}(N_d)$ or, equivalently, i belongs to a longest path in G_A. Every node in N_d begins at* *least one path of the maximum length $d-1$, and each such path ends at a node in N_1.*

PROOF. Immediate from the foregoing discussion. ■

For convenience we abbreviate $N_t(N_r)$ to N_t^r.

PROPOSITION 3.3. *Let $A \in M_n$ be acyclic. Then*

$$E_A = \bigcup_{r=1}^{d} N_r^d.$$

PROOF. The case of zero matrices is trivial, so suppose $A \neq 0$, i.e., $d > 1$. Any cycle σ in D_A contains at least one zero arc, because A is acyclic. We can decompose σ into t (say) disjoint paths each of which contains a zero arc as the first arc, possibly followed by one or more unity arcs. For $h = 1, \ldots, t$ let $u_h \geq 0$ be the number of such unity arcs. Then

$$\mu_A(\sigma) = \frac{\sum_{h=1}^{t} u_h}{\sum_{h=1}^{t}(u_h+1)}. \tag{3-2}$$

Proposition 3.1 implies that $u_h \leq d(A) - 1$ for all $h = 1, \ldots, t$. It easily follows that $\mu_A(\sigma)$ achieves the value $[d(A)-1]/d(A)$ only if every summand u_h has the (maximal) value $d(A) - 1$. By Propositions 3.1 and 3.2 this implies that every node i of σ lies in $N_{h(i)}^d$.

Conversely, if $i \in N_{h(i)}^d$, then by Proposition 3.2 node i lies on some path π of maximal length $d-1$ in G_A. Closing π by a (necessarily zero) arc gives a cycle σ having $\mu_A(\sigma) = (d-1)/d$. ■

THEOREM 3.1. *Let $A \in M_n$ be acyclic. Then all eigenvectors of A are equivalent to the particular eigenvector, given by*

$$X^0 = \frac{(h(1), \ldots, h(n))^T}{d(A)}.$$

PROOF. The case of zero matrices is trivial, so suppose $A \neq 0$, i.e., $d > 1$. Let $k \in N$, $i \in E_A$. Then we first show

$$x_k^i = \frac{h(k) - h(i)}{d(A)}, \tag{3-3}$$

where x_k^i stands for the weight of a heaviest path from node k to node i in D_{A_λ}. Now, ω_{ki}, the weight of arc (k, i) in D_{A_λ}, is $a_{ki} - \lambda$, where $a_{ki} = 1$ or 0. Hence, using Proposition 2.3 (iii),

$$\omega_{ki} = \frac{1}{d(A)} \text{ or } \frac{1 - d(A)}{d(A)}, \qquad \text{respectively.} \tag{3-4}$$

In the first case, $h(k) - h(i) > 1$ by Proposition 3.1, because (k, i) is a path in G_A. Hence *a fortiori*

$$\omega_{ki} \leq \frac{h(k) - h(i)}{d(A)}. \tag{3-5}$$

But (3-5) also holds in the second case of (3-4), because $h(k) \geq 1$ and $h(i) \leq d$. So if $(j_1 = k, j_2, \ldots, j_p = i)$ is a heaviest path from k to i in D_{A_λ}, its weight is

$$x_k^i = \sum_{r=1}^{p-1} \omega_{j_r j_{r+1}} \leq \sum_{r=1}^{p-1} \frac{h(j_r) - h(j_{r+1})}{d(A)} = \frac{h(k) - h(i)}{d(A)}. \tag{3-6}$$

But, using Proposition 3.1, we may find in G_A a path π_1 of length $h(k) - 1$ from k to some node $k' \in N_1$ and a path π_2 of length $d(A) - h(i)$ to i from some node $i' \in N_d$, and hence in D_{A_λ} construct the path

$$\pi = \pi_1(k', i')\pi_2. \tag{3-7}$$

[If $k \in N_1$, take $k' = k$ and drop π_1 from (3-7); if $i \in N_d$, take $i' = i$ and drop π_2 from (3-7).]

The total weight of π in D_{A_λ} is

$$[h(k) - 1]\frac{1}{d(A)} + \frac{1 - d(A)}{d(A)} + [d(A) - h(i)]\frac{1}{d(A)} = \frac{h(k) - h(i)}{d(A)}. \tag{3-8}$$

Hence the bound in (3-6) is achieved, and (3-3) follows.

Now every $j \in N_d$ is an eigennode. By choosing $i = j$ in (3-3) we find X^j with components

$$x_k^j = \frac{h(k) - d(A)}{d(A)},$$

which differs only by unity from the eigenvector proposed in the theorem statement, and only by an additive constant $h(i)/d(A) - 1$ from X^i given by (3-3) for arbitrary $i \in E_A$. ∎

COROLLARY. *Let $A \in M_n$ be acyclic. Then*

$$\mathrm{sp}(A) = \{c + X^0;\ c \in \mathbb{R}\}. \tag{3-9}$$

THEOREM 3.2. *Let $A \in M_n$ be acyclic. Then $\lambda(A)$ and the (unique) generator of* sp(A) *can be found in $O(m + n)$ operations.*

PROOF. Immediate from the foregoing results. ■

4. EIGENSPACE OF CYCLIC MATRICES

A cycle in D_A will be called a *unity cycle* if it consists of unity arcs unity. If $\lambda(A) = 1$, then a node in D_A is an eigennode if and only if it lies on a unity cycle, or equivalently, it belongs to a strongly connected component of G_A. Hence E_A is now a union of the sets of nodes of strongly connected components of G_A, and two eigennodes are equivalent if and only if they belong to the same strongly connected component. Strongly connected components of a digraph can be found in $O(m + n)$ operations [9].

The arcs of D_{A_λ} now have weight either -1 or 0. Hence the weight x_i^j of a heaviest path from i to j in D_{A_λ} is 0 if and only if j can be reached from i by a path consisting only of arcs with weight 0, and is otherwise -1 using arc (i, j). However arcs with weight 0 in D_{A_λ} correspond precisely to (nonweighted) arcs of G_A. Hence we have: $x_i^j \in \{-1, 0\}$, and $x_i^j = 0$ if and only if a path from i to j in G_A exists.

In particular, for any $j \in E_A$, all nodes $i \in N$ for which a path from i to j exists in G_A can be found by a standard reachability algorithm in $O(m)$ operations [7].

We summarize the foregoing discussion:

THEOREM 4.1. *If $\lambda(A) = 1$, then this property can be recognized in $O(m + n)$ operations using Algorithm* FASTDECOMPOSITION. *Moreover,*

(a) *eigennodes of A correspond exactly to the nodes of strongly connected components of G_A whereby two eigennodes are equivalent if and only if they belong to the same component; thus a maximal set of nonequivalent eigennodes can be found in $O(m + n)$ operations;*
(b) *a fundamental eigenvector $X^j = (x_1^j, x_2^j, \ldots, x_n^j)$ corresponding to eigennode j can be found in $O(m + n)$ operations using a standard reachability algorithm and the formula*

$$x_i^j = \begin{cases} 0 & \text{if } j \text{ is reachable from } i, \\ -1 & \text{otherwise;} \end{cases}$$

(c) *a complete set of generators can be found in* $O(q'(m+n))$ *operations, where* $q' = |F'_A|$.

5. EXTENSION TO GENERAL $\mathcal{G}$

Let us now return to the case where $\mathcal{G} = (G, \otimes, \leq)$ is a general linearly ordered, commutative radicable group and A is an $n \times n$ matrix all of whose elements lie in $\{a, b\}$, where $a, b \in G$ and $a < b$. Let $B = a^{-1} \otimes A$. Then

$$\begin{aligned} \lambda(A) &= a \otimes \lambda(B), \\ \mathrm{sp}(A) &= \mathrm{sp}(B). \end{aligned} \tag{5-1}$$

Hence it suffices to consider B whose elements lie in $\{e, \alpha\}$, where $\alpha = b \otimes a^{-1} > e$.

Let $\mathbb{Q} = (Q, +, \leq)$ denote the rational numbers, and let $\mathcal{H}$ denote the subgroup of $\mathcal{G}$ given by

$$\mathcal{H} = \big(\{\alpha^p; p \in Q\}, \otimes, \leq\big). \tag{5-2}$$

Then there is an isotonic isomorphism between $\mathcal{H}$ and $\mathbb{Q}$:

$$\theta : \alpha^p \leftrightarrow p. \tag{5-3}$$

Under θ, the elements e, α correspond to $0, 1$ respectively. Hence to carry out the computations necessary to solve the eigenproblem for general bivalent matrices over $\mathcal{G}$, it suffices to apply the (rational) algorithms of the preceding sections to the corresponding zero-one real matrices.

REFERENCES

1 P. Butkovič and R. A. Cuninghame-Green, An $O(n^2)$ algorithm for the maximum cycle mean of an $n \times n$ bivalent matrix, *Discrete Appl. Math.*, 35:157–162 (1992).

2 R. A. Cuninghame-Green, Describing industrial processes with interference and approximating their steady-state behavior, *Oper. Res. Quart.* 13:95–100 (1962).

3 R. A. Cuninghame-Green, *Minimax Algebra*, Lecture Notes in Econom. and Math. Systems 166, Springer-Verlag, Berlin, 1979.

4 G. B. Dantzig, E. D. Blattner, and M. R. Rao, Finding a cycle in a graph with minimum cost to time ratio, with application to a shiprouting problem, in *Theory of Graphs* (P. Rosenstiehl, Ed.), Gordon and Breach, New York, 1967.

5 M. von Golitschek, Optimal cycles in doubly weighted graphs and approximation of bivariate functions by univariate ones, *Numer. Math.* 39:65–84 (1982).

6 R. M. Karp, A characterization of the minimum cycle mean in a digraph, *Discrete Math.* 23:309–311 (1978).

7 E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.

8 J. Plávka, On the complexity of the least maximum cycle mean problem with respect to row and column permutations, in preparation.

9 R. E. Tarjan, Depth-first search and linear graph algorithms, *SIAM J. Comput.* 1:146–160 (1972).

10 N. N. Vorobyev, Ekstremal'naya algebra polozhitel'nykh matrits (in Russian), *Elektron. Informationsverarbeitung und Kybernetik* 3:39–71 (1967).

11 U. Zimmermann, *Linear and Combinatorial Optimization in Ordered Algebraic Structures*, Ann. Discrete Math. 10, North-Holland, Amsterdam, 1981.

Received April 22, 1992; final manuscript accepted October 6, 1993