Maximal acyclic subgraphs and closest stable matrices

Aleksandar Cvetković, †Vladimir Yu. Protasov ‡

Abstract

We consider the following weakened version of the Maximal Acyclic Subgraph (MAS) problem: given a directed graph, find minimal $r$ such that one can make the graph acyclic by cutting at most $r$ incoming edges from each vertex. We present an efficient algorithm of solution for this problem and for some of its modifications: when each vertex is assigned with its own maximal number $r_i$ of cut edges, when some of edges are “untouchable”, etc. The same algorithm can be modified for approximate solution of the classical MAS problem. Numerical results are provided for random graphs with the number of vertices from 50 to 1500. They show the rate of approximation for the MAS problem around 0.6.

The main idea is based on finding the closest non-negative Schur stable matrix. In particular, recent methods of minimising spectral radius over special families of matrices are put to good use in those problems.

Keywords: acyclic graph, non-negative matrix, spectral radius, relaxation algorithm, closest stable matrix, spectrum of a graph

AMS 2010 subject classification: 05C50, 15B48, 90C26

1. Introduction

Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with the set of vertices $\mathcal{V} = \{g_1, \ldots, g_n\}$ and with the set of edges $\mathcal{E}$. The vertices will be identified with the corresponding numbers $\{1, \ldots, n\}$. The Maximal Acyclic Subgraph (MAS) of $G$ is a graph $\hat{G} = (\mathcal{V}, \hat{\mathcal{E}})$ such that $\hat{\mathcal{E}} \subset \mathcal{E}$, $\hat{G}$ has no cycles, and the number of edges $|\hat{\mathcal{E}}|$ is maximal possible. The MAS problem (in short MAS) is to find a maximal acyclic subgraph of a given graph $G$. This problem is NP-hard [18]. It is not known if efficient algorithms exist to obtain an approximate solution, when the number
of edges of the obtained acyclic subgraph is at least \( \delta |\hat{E}| \) with \( \delta > \frac{1}{2} \), and most likely the answer is negative \[15\].

It is interesting that the simplest algorithm for an approximate solution of MAS gives the best known result \( \delta = \frac{1}{2} \). Indeed, take an arbitrary renumbering of vertices of \( G \) (a permutation \( \sigma \) of the set \( \{1, \ldots, n\} \)), then the set of forward edges in the new enumeration and the set of backward edges are both acyclic and one of them contains at least \( \frac{1}{2} |E| \geq \frac{1}{2} |\hat{E}| \) edges. Hence, every permutation gives an approximate solution to MAS with the factor \( \delta \geq \frac{1}{2} \). In fact, the exact solution of MAS is equivalent to finding an optimal permutation \( \sigma \) with the biggest number of edges \( |\hat{E}| \). See \[6, 8, 15, 17\] for algorithms of approximate solutions and for discussions of the complexity.

We consider a matrix approach to MAS. First we reformulate this problem as finding the closest (in the Frobenius norm) non-negative matrix \( X \) to a given matrix \( A \) with the condition \( \rho(X) = 0 \). One can note that this problem is related to finding the closest Schur stable matrix to a given matrix \( A \). That stabilisation problem is formulated in the same way but with the condition \( \rho(X) = 1 \). Some methods for the Schur stabilization were elaborated in recent literature, but all of them find only local minima, which does not give much for the MAS problem. Nevertheless, replacing the Frobenius norm by the \( L_\infty \)-norm, we obtain the Schur stabilization problem that can be effectively solved \[20\]. In the graph terms this replacement leads to the following problem: find minimal \( r \) such that one can make the graph acyclic by cutting at most \( r \) incoming edges from each vertex. In Section 3 we present an algorithm of solution. For graphs with 100 vertices, the complete solution takes usually a few seconds in a standard laptop, for 1000 vertices it takes about 10 minutes (see Section 6 for numerical results). The algorithm is based on minimising the Perron eigenvalue on special sets of matrices. In Section 4 we formulate several generalisations of our problem which admit similar methods of solution. In Section 5 we come back to the (classical) MAS problem. We show that a slight modification of our method can be used to finding an approximate solution of MAS. The rate of approximation is estimated from below by the value \( \gamma = n^{-2}|E'| \), where \( E' \) is the set of edges of the obtained acyclic subgraph. Obviously \( \delta \geq \gamma \). In the numerical experiments, our method gives the rate of approximation \( \gamma \approx 0.6 \). Finally, in Section 6 we present numerical results and discuss the complexity issue.

Throughout the paper, by the graph of a non-negative \( n \times n \) matrix \( A \) we mean the graph \( G \) with \( n \) vertices \( \{1, \ldots, n\} \) such that there is an edge from a vertex \( i \) to a vertex \( j \) if and only if \( A_{ij} > 0 \). We use a standard componentwise ordering of real matrices: \( A \geq (>\))B if \( A_{ij} \geq (>\)B_{ij} for all \( i, j \); the same for vectors; \( \rho(A) \) denotes the spectral radius of a matrix \( A \), i.e., the largest modulus of its eigenvalues. By the Perron-Frobenius theorem \[10\], chapter 8], if \( A \geq 0 \), then there is an eigenvalue \( \lambda_{\text{max}} = \rho(A) \) with a non-negative eigenvector. This eigenvalue and eigenvector are referred to as leading. The vectors will be denoted by bold letters and their components by standard letters: \( x = (x_1, \ldots, x_n) \). Matrices (vectors) with entries 0 and 1 are Boolean. In particular, an adjacency matrix of a graph \( (A_{ji} = 1 \Leftrightarrow \text{there is an edge from } i \text{ to } j; \text{ otherwise } A_{ji} = 0) \) is Boolean.

We use the notation \( \mathcal{V} = \{1, \ldots, n\} \). This set will be identified with the set of vertices of the graph \( G \). For an arbitrary finite set \( K \), \( |K| \) denotes its cardinality. Let \( X \) be an
$n \times n$ matrix. For a subset $S \subset \mathcal{V}$, we denote by $X|_S$ the $|S| \times |S|$ principal submatrix of $X$, which is a restriction of $X$ to the index set $S$. By $x|_S$ we denote the restriction of the vector $x \in \mathbb{R}^n$ to the $|S|$-dimensional space of vectors with all non-zero components in $S$.

The Frobenius norm is the Euclidean norm on the set of matrices understood as vectors in the $n^2$ dimensional space: $\|X\| = \left[ \sum_{i,j} |X_{ij}|^2 \right]^{1/2}$. We always use this matrix norm if another norm is not specified.

2. The MAS problem vs the closest stable matrix problem

The following simple observation presents the MAS problem in the linear algebraic terms.

**Proposition 1** The MAS problem is equivalent to the following problem: given a graph $G$ with an adjacency matrix $A$, find the closest in Frobenius norm non-negative matrix $X$ with zero spectral radius:

$$\begin{cases}
\|X - A\| \to \min \\
X \geq 0, \quad \rho(X) = 0
\end{cases} \tag{1}$$

Every solution $X$ of the problem (1) is a Boolean matrix whose graph solves the MAS problem for $G$.

**Proof.** The matrix $A$ is Boolean. Hence, if we replace all strictly positive entries of $X$ by ones, then the spectral radius remains zero but the distance $\|X - A\|$ becomes smaller. If for each pair $(i, j)$, we set $X_{ij} = 0$ whenever $A_{ij} = 0$, we obtain the same effect. Consequently, problem (1) can be reduced to the set of Boolean matrices $X$ such that $X \leq A$ i.e., to matrices of subgraphs of $G$. In this case, $\|X - A\|$ is equal to the square root of the number of edges cut to obtain $X$ from $A$. On the other hand, $\rho(X) = 0$ precisely when the graph of $X$ is acyclic. Thus, problem (1) is equivalent to finding the acyclic subgraph obtained from $A$ by cutting the smallest number of edges, i.e., to the MAS problem.

Problem (1) looks similar to the closest stable matrix problem. We formulate only the non-negative version of this problem. Given a matrix $A \geq 0$, find the closest matrix such that $\rho(X) = 1$:

$$\begin{cases}
\|X - A\| \to \min \\
X \geq 0, \quad \rho(X) = 1
\end{cases} \tag{2}$$

This is the same problem (1) but with $\rho = 0$ replaced by $\rho = 1$. The closest stable matrix problem have been studied in a bulk of recent works due to its application in dynamical and controlled systems, mathematical economics, population dynamics, etc. (see [7, 12, 13, 21, 22] for the problem with general matrices and [3, 20, 14] for non-negative matrices). A question arises if the methods of solutions of the closest stable matrix problem can be useful for MAS? The main difficulty is that all known methods find only locally stable matrices. Moreover, problem (1) may have an exponential number of local minima [14]. It turns out, however, that in another matrix norm, the closest stable matrix can be efficiently found. For example,
in the $L_\infty$-norm: $\|X\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^n |X_{ij}|$. An algorithm presented in [20] finds the global minimum in the problem (2) with $L_\infty$-norm quite fast even in high dimensions (several thousands). Reformulating the problem in a new norm in term of graphs we come to the following variant of MAS:

**The max-MAS Problem.** Find the minimal integer $r$ such that a given graph $G$ can be made acyclic by cutting at most $r$ incoming edges from each vertex.

If we denote by $c_i$ the sum of cut incoming edges from $i$th vertex, then the MAS problem minimises $\sum_{i=1}^n c_i$ while the max-MAS problem minimises $\max_{i=1,\ldots,n} c_i$. This justifies our terminology. In the next section we will see that this problem can be efficiently solved with an idea of finding the closest stable matrix [20]. Note that in spite of similarity of problems (1) and (2) they are different, since the set of matrices with $\rho(X) = 1$ and with $\rho(X) = 0$ have different structures. Nevertheless, the max-MAS problem can be solved by the method based on the algorithms for solving the problem (2) in $L_\infty$-norm. Moreover, the same method can solve several generalisations of max-MAS. For example, when it is allowed to cut at most $r_i$ incoming edges from the vertex $i$, where $r_1, \ldots, r_n$ are given numbers. Some of those numbers may be zeros, in which case the corresponding vertex is “untouchable”. This and other generalizations are solved in Section 4. Then, in Section 5 we suggest an approximate solution of the classical MAS problem based on the presented algorithms. Numerical results and the complexity issue are presented in Section 6.

## 3. Algorithmic solution for the max-MAS problem

### 3.1. Spectral formulation

The max-MAS problem is formulated in linear algebraic terms as follows:

$$\begin{cases} 
\|X - A\|_\infty \rightarrow \min \\
X \geq 0, \ \rho(X) = 0 
\end{cases} \quad (3)$$

where $A$ is a given Boolean $n \times n$ matrix. The equivalence of this problem to max-MAS is proved in the same way as in Proposition [4]. This problem, it turn, is reduced to

$$\begin{cases} 
\rho(X) \rightarrow \min \\
X \geq 0, \ \|X - A\|_\infty \leq r 
\end{cases} \quad (4)$$

Indeed, if we are able to solve (4) for every integer $r$, then the minimal possible $r$ for which there is a non-negative matrix $X$ such that $\|X - A\|_\infty \leq r$ and $\rho(X) = 0$ is found merely by integer bisection in $r$.

It is more convenient to consider problem (4) in slightly different terms. For the $i$th row of matrix $A$ we denote $\mathcal{B}(A_i, r) = \{x \in \mathbb{R}^n_+ \mid \|A_i - x\|_1 \leq r\}$, where $\|y\|_1 = \sum_{j=1}^n |y_j|$ is the $L_1$-norm. So, $\mathcal{B}(A_i, r)$ is the $L_1$-ball of radius $r$ centered at $A_i$ and intersected with the positive
orthant. Then $\mathcal{B}(A, r)$ denotes the set of matrices with $i$th row from $\mathcal{B}(A_i, r), i = 1, \ldots, n$. Clearly, $\mathcal{B}(A, r)$ is a convex polyhedron in the set of matrices.

Now we focus on the following problem equivalent to (4): for a given Boolean matrix $A$ and for $r \in \mathbb{N}$ solve
\[
\begin{cases}
\rho(X) \rightarrow \min \\
X \in \mathcal{B}(A, r)
\end{cases}
\] (5)
The solution $X$ is always a Boolean matrix. If we find the smallest non-negative integer $r$ for which this problem has a solution, then the graph of this solution $X$ solves the max-MAS problem.

3.2. Minimisation of spectral radius over product families

The problem (5) can be efficiently solved for each $r$. This is because $\mathcal{B}(A, r)$ is a product set of matrices. Optimising the spectral radius over product sets has been investigated in various contexts [19, 24, 20, 1]. Let us have arbitrary compact sets $F_i \in \mathbb{R}^n_+$. The set of matrices $\mathcal{F} = \{X \mid X_i \in F_i, i = 1, \ldots, n\}$ is called a product set. Every matrix from a product set is composed with rows chosen from the sets $F_i$ arbitrarily and independently. The sets $F_i$ are usually referred to as uncertainty sets. All methods of minimising the spectral radius over product families are actually based on the well-known formula
\[
\rho(X) = \sup \{ \lambda \mid \exists u \geq 0, u \neq 0 : X u \geq \lambda u \},
\] (6)
which holds for every non-negative matrix [3]. If we minimise the spectral radius of a matrix over a product set, this formula allows us to treat all rows separately:
\[
\min_{X \in \mathcal{F}} \rho(X) = \sup \{ \lambda \mid \exists u \geq 0, u \neq 0 : \langle X_i, u \rangle \geq \lambda u_i, X_i \in F_i, i = 1, \ldots, n \}. \tag{7}
\]
If all the uncertainty sets $F_i$ are finite, then there are several methods of applying linear programming to this problem. Their complexity, however, depends on the cardinalities of the uncertainty sets [23, 27]. In our case the sets $F_i = \mathcal{B}(A_i, r)$ are polytopes with exponentially many vertices. If the row $A_i$ contains $m_i$ ones and $m_i > r$, then the number of vertices is $\binom{m_i}{r}$. Therefore, the linear problem has an exponential number of inequalities, which makes this approach inefficient for the problem (5).

That is why we make use of another approach based on a recursive relaxation scheme suggested in [11, 20] and called the greedy algorithm. A similar scheme was also applied in Markov decision processes [2, 16, 25]. A crucial point of this approach is the following fact:

**Theorem A.** [19] A matrix $X \in \mathcal{F}$ has the minimal spectral radius over a product set $\mathcal{F}$ if and only if $X$ possesses a leading eigenvector $v$ such that $\langle X_i, v \rangle = \min_{x \in F_i} \langle x, v \rangle$ for each $i = 1, \ldots, n$.

The relaxation scheme works as follows. If in $k$th iteration we have a matrix $X^{(k)} \in \mathcal{F}$, then we compute its leading eigenvector $v^{(k)}$ and then, for every $i = 1, \ldots, n$, we replace the $i$th row of $X^{(k)}$ by the element $X_i \in F_i$ which minimises the scalar product $\langle X_i, v^{(k)} \rangle$.
We obtain $X^{(k+1)}$, etc. Thus, in each iteration we replace every row of the matrix by the optimal row in the corresponding uncertainty set, i.e., by the row making the shortest projection with the leading eigenvector. Formula (6) implies that this scheme is a relaxation: $\rho(X^{(k)})$ is decreasing in $k$, maybe non-strictly. Under some “positivity-like” assumptions on the uncertainty sets $\mathcal{F}^{(k)}$, the spectral radius strictly decreases, and hence the solution is found in finite time.

This scheme, however, has a serious disadvantage: without those “positivity-like” assumptions the algorithm may cycle. Moreover, for sparse matrices the cyclicity happens quite often. A modified greedy algorithm which never cycles was presented in [20]. According to numerical experiments, that algorithm has a very fast convergence and finds the minimum within a few iterations even in high dimensions [9]. Of course, if some of the sets $\mathcal{F}_i$ are of large cardinality, one iteration may take long, since it requires computing of $|\mathcal{F}_i|$ scalar products. This occurs, in particular, in our problem (5), where each set $\mathcal{F}_i = B(A_i, r)$ has an exponential number of extreme points. In this section we overcome this difficulty and modify the greedy algorithm specially for these sets. To this end we begin with introducing some further notation.

An arbitrary non-negative vector $v \in \mathbb{R}_+^n$ defines an ordering on the set $V = \{1, \ldots, n\}$ by the values of components of $v$: $i \geq_j j$ if $v_i \geq v_j$. For every $r \leq n$, we can consider the set of $r$ largest elements of $V$ in this ordering. This set may not be unique if $v$ have some equal components. In this case we take any set of $r$ largest elements. Similarly, for any subset $S \subset V$, we consider the set $S'$ of $r$ largest elements of $S$. We say that the set $S'$ occupies $r$ largest components of $v$. If $|S| < r$, then we say that the whole set $S' = S$ occupies $r$ largest components of $v$ (although it contains less number of elements).

**Definition 1** Let $A$ be a Boolean $n \times n$ matrix and $r \in \mathbb{N}$. Let also $X$ be a Boolean $n \times n$ matrix and $v \in \mathbb{R}_+^n$ be a vector. A row $X_i$ is minimal in the ball $B(A_i, r)$ with respect to $v$ if the set of zeros of $X_i$ on the set $\text{supp} A_i$ occupies $r$ largest components of $v$ on that set.

Thus, if $\|A_i\|_1 > r$, then the minimality of $X_i$ means that in the index set $\text{supp} A_i$, there are $r$ indices corresponding to maximal (in this set) components of the vector $v$ on which all components of $X_i$ are zeros. If $\|A_i\|_1 < r$, then the minimality means simply that $X_i = 0$ on the whole set $\text{supp} A_i$.

It is easy to find the minimal row in the ball $B(A_i, r)$ with respect to a given vector $v$. One needs to order the index set $\text{supp} A_i$ by the values of the components of $v$ and take $r$ biggest (in this order) indices. Denote this set by $J$. If $|\text{supp} A_i| \leq r$, then put $J = \text{supp} A_i$. Then the minimal row is $X_{ij} = 0$ if $j \in J$ and $X_{ij} = A_{ij}$ otherwise.

The minimal row possesses the shortest possible projection to the vector $v$ among all elements of $B(A_i, r)$ as the following lemma asserts.

**Lemma 1** A row $X_i$ is minimal with respect to a vector $v$ in the ball $B(A_i, r)$ precisely when

$$\langle X_i, v \rangle = \min_{x \in B(A_i, r)} \langle x, v \rangle$$

(8)
Proof. Assume $|\text{supp } A_i| > r$. If the set of zeros of $X_i$ does not occupy $r$ largest components of $v$ on $\text{supp } A_i$, then there are numbers $j, k \in \text{supp } A_i$ such that $v_j > v_k$ and $v_j = 1, v_k = 0$. Interchanging those components we reduce the scalar product $\langle X_i, v \rangle$, which is a contradiction. Conversely, assume the set of zeros of $X_i$ occupies $r$ largest components of $v$ on $\text{supp } A_i$, but the minimal scalar product is smaller than for $X_i$ and is attained at some $x \in B(A_i, r)$. It can be assumed that $x$ is an extreme point of the ball $B(A_i, r)$, i.e., a Boolean vector. Hence $\langle A_i - x, v \rangle$ does not exceed the sum of $r$ largest on the set $\text{supp } A_i$ components of $v$, i.e., does not exceed $\langle A_i - X_i, v \rangle$. Hence $\langle x, v \rangle \geq \langle X_i, v \rangle$. If $|\text{supp } A_i| \leq r$, then the minimal (coordinatewise) element of the ball $B(A_i, r)$ is the origin, for which the minimal scalar product is attained.

Remark 1 If $v$ possesses some zero components, then the definition of the minimal row can be reduced to the support of $v$. We reduce all vectors to the set $S = \text{supp } v$ and do not pay attention to other components. Then minimal vectors stay minimal after this reduction.

Theorem 1 Let $A$ be a Boolean matrix and $r \in \mathbb{N}$. The problem (5) reaches its global minimum at some Boolean matrix $X$. This matrix is characterised by the property: there exists a leading eigenvector $v$ of $X$ such that each row $X_i$ is minimal in the ball $B(A_i, r)$ with respect to $v$.

Proof follows by combining Lemma 1 and Theorem A for the sets $F_i = B(A_i, r)$ and for the leading eigenvector $v$ of $X$.

If the eigenvector $v$ is sparse, then it makes sense to reduce all vectors to the set $S = \text{supp } v$ in the spirit of remark 2. Indeed, the scalar product $\langle X_i, v \rangle$ depends only on entries of the vector $X_i$ on the set $S$. Hence, if $v$ is not strictly positive, the criterion of Theorem 1 can be reduced to the set $S$:

Corollary 1 The criterion for solution of the problem (5) in Theorem 1 can be written in the following form. Let $S = \text{supp } v$. Denote by $v', X'_i$, and $A'_i$ the restrictions of those vectors to the set $S$. Then the matrix $X$ is a solution of the problem (3) if and only if there exists a leading eigenvector $v$ of $X$ such that for every $i \in S$ each row $X'_i$ is minimal in the ball $B(A'_i, r)$ with respect to $v'$.

If $v > 0$, then Corollary 1 coincides with Theorem 1. If $v$ has some zero components, then the criterion of Corollary 1 is simpler in practice since it involves only the submatrix $X' = X|_S$.

3.3. Some auxiliary facts on non-negative matrices

A non-negative matrix $A$ is called irreducible if it does not have a nontrivial invariant coordinate subspace, i.e., a subspace spanned by some elements $e_i$ of the canonical basis. A matrix is irreducible if and only if its graph is strongly connected. Reducibility means that
there is a proper nonempty subset \( \Lambda \subset V \) such that for each \( i \in \Lambda \), the support of the \( i \)th column of \( A \) is contained in \( \Lambda \).

For every matrix \( A \geq 0 \), there exists a suitable permutation \( P \) of the basis of \( \mathbb{R}^n \), after which \( A \) gets a block upper triangular form with \( q \geq 1 \) diagonal blocks \( A_j \) called the Frobenius factorization:

\[
P^{-1}AP = \begin{pmatrix}
A_1 & * & \ldots & * \\
0 & A_2 & * & \\
\vdots & \ddots & \ddots & * \\
0 & \ldots & 0 & A_q
\end{pmatrix}.
\] (9)

For each \( j = 1, \ldots, q \), the matrix \( A_j \) in the \( j \)th diagonal block is irreducible. Any non-negative matrix possesses a unique Frobenius factorization up to a permutation of blocks (see [10, chapter 8]).

It is well-known (e.g. [5]) that an irreducible matrix has a simple leading eigenvalue. The converse is not true: a matrix with a simple leading eigenvalue can be reducible.

Let \( A \) be an \( n \times n \) non-negative matrix. Its leading eigenvector \( v \) is called minimal if there is no other leading eigenvector that possesses a strictly smaller (by inclusion) support. A minimal eigenvector is unique up to normalisation if and only if the geometrical multiplicity of the leading eigenvector is one. A minimal leading eigenvector can be found by Frobenius factorization [9]. It suffices to take the smallest \( m \) such that \( \rho(A_m) = \rho(A) \) i.e., the “highest” diagonal block with the maximal spectral radius. Then consider the leading eigenvector \( u \) of the principal submatrix of \( A \) that consists of blocks \( A_1, \ldots, A_m \). Complement this vector by zeros to an \( n \)-dimensional vector and obtain a minimal eigenvector \( \bar{u} \) of the matrix \( P^{-1}AP \). Respectively, the vector \( P\bar{u} \) is the minimal leading eigenvector of \( A \).

Can the minimal leading eigenvector be strictly positive, i.e., possess a full support? Of course. This case is characterized by the following property:

Proposition 2 [20] If a matrix \( A \geq 0 \) has a strictly positive minimal leading eigenvector \( v \), then its leading eigenvalue is simple. In the Frobenius factorisation [9] the spectral radius of \( A_q \) is equal to \( \rho(A) \) and the spectral radii of all the other blocks are smaller than \( \rho(A) \).

In case of positive minimal leading eigenvector we can define the notion of a basic set, which is needed in our algorithm.

Definition 2 Suppose a matrix \( A \) has a strictly positive minimal leading eigenvector. Then the basic set of \( A \) is the support of the leading eigenvector of the matrix \( A^T \).

If the minimal leading eigenvector is strictly positive, then by Proposition 2 the leading eigenvalue is simple. Then so is the leading eigenvalue of of the transposed matrix. This implies the correctness of Definition 2. Another consequence of Proposition 2 is that the basic set can be found without computation of the leading eigenvector of \( A^T \), provided the Frobenius factorisation is available.

Proposition 3 [20] Suppose a matrix \( A \) has a strictly positive minimal leading eigenvector; then the basic set of \( A \) is the set of indices which after the permutation \( P \) form the last block \( A_q \) in the Frobenius factorisation [9].
Proof. By Proposition 2 the matrix $A$ and hence $P^{-1}AP$ has a simple leading eigenvalue which is located in the last block $A_q$ of factorisation (9). Hence the leading eigenvector of $[P^{-1}AP]^T$ has its support in the last block. On the other hand, $[P^{-1}AP]^T = P^{-1}A^TP$, from which the proposition follows.

Now the preliminary work is done and we are ready to present the main result.

3.4. The algorithm of solution of problem (5)

Notation for Algorithm 1. Let $\mathcal{V} = \{1, \ldots, n\}$. For a matrix $X$, vector $x$, and a subset $K \subset \mathcal{V}$, we denote by $X|_K$ and $x|_K$ the restrictions of the matrix and of the vector to the index set $K$ (see Introduction). We use the notions of minimal leading eigenvector (Definition 1) and of the basic set (Definition 2). We use an obvious fact that if $v$ is a minimal leading eigenvector of a matrix $X$ and $S = \text{supp} \ v$, then the matrix $X|_S$ has a strictly positive minimal leading eigenvector $v|_S$. 
Algorithm 1: Algorithm 1 for solving problem (5)

Data: A Boolean $n \times n$ matrix $A$, a number $r \in \mathbb{N}$

Result: A matrix $\hat{X}$ which is a solution of problem (5).

begin

1. Set $X^{(1)} = A$.

(*) $k$th iteration. We have a Boolean $n \times n$ matrix $X^{(k)}$.

2. if $\rho(X^{(k)}) < 1$ then

   STOP. Algorithm 1 terminates. Denote $\hat{X} = X^{(k)}$ and go to return;

else

3. Denote $X = X^{(k)}$. Compute a minimal leading eigenvector $v$ of $X$ (take any of them, if there are several ones), set $S = S^{(k)} = \text{supp } v$, $X' = X|_S$, and go to (**);

(**) Main loop. We have an $n \times n$ matrix $X$, its minimal leading eigenvector $v$, $S \subset V$. We consider a submatrix $X' = X|_S$ and its strictly positive minimal leading eigenvector $v' = v|_S$.

   Let $\mathcal{H} \subset S$ be the basic set of $X'$ and $\mathcal{I}$ be the set of indices $i \in S$ for which the row $X'_i$ is minimal in the ball $B(A'_i, v')$ with respect to the vector $v'$.

5. if $\mathcal{I} = S$ then

   STOP. Algorithm 1 terminates. Define the $n \times n$ matrix $\hat{X}$ as follows:

   $\hat{X}_i = X_i$ for $i \in S$ and $\hat{X}_i = A_i$ for $i \notin S$. Go to return;

else

6. Define the next matrix $\tilde{X}$ as follows:

   If $i \in \mathcal{I}$ or $i \notin S$, then $\tilde{X}_i = X_i$;

   Otherwise, if $i \in S \setminus \mathcal{I}$, then $\tilde{X}_i|_S$ is the minimal row in the ball $B(A'_i, r)$ with respect to $v'$ and $\tilde{X}_{ij} = A_{ij}$ for all $j \notin S$.

   Set $X' = \tilde{X}|_S$;

7. if $\mathcal{H} \subset \mathcal{I}$ then

   $\rho(\tilde{X}') = \rho(X')$, the leading eigenvalue of $\tilde{X}'$ is simple;

   We compute the minimal leading eigenvector $\tilde{v}'$ of $\tilde{X}'$. The set $S$ is not changed;

   if $\tilde{v}' > 0$ then

      set $X = \tilde{X}$, $v = \tilde{v}$, and $S$ is as before. Go to (**);

   else

      set $S = \text{supp } \tilde{v}'$, $X' = \tilde{X}|_S$, and $v' = \tilde{v}|_S$. Go to (**).

else

   we have $\mathcal{H} \notin \mathcal{I}$ and $\rho(\tilde{X}) < \rho(X)$. Set $X^{(k+1)} = \tilde{X}$ and go to the next $(k+1)$st iteration (*);

return $\hat{X}$ is a solution;
3.5. Explanations and proofs

Explanations.

The algorithm is a relaxation scheme: the value of the spectral radius $\rho(X)$ is non-increasing during the whole algorithm.

The algorithm consists of iterations (*), each iteration consists of several steps (**). During all steps (in one iteration) the value $\rho(X)$ is the same. The iteration is completed when this value becomes strictly smaller.

(*): In $k$th iteration we have a matrix $X = X^{(k)}$. We find its minimal leading eigenvector $v$ and denote $S = \text{supp } v$. Then till the end of this iteration we work on the set $S$ without involving other indices. We consider the matrix $X'$, which is a restriction of $X$ to the set $S$. Then the vector $v' = v|_S$ is a positive minimal leading eigenvector of $X'$. We find the basic set $H$ of $X'$ (Definition 1), see (10).

(**): We replace all rows of $X'$ by the minimal rows in the corresponding balls $B(A'_i, r)$ with respect to $v'$. All other elements of $X$ are not changed. We obtain a matrix $\tilde{X}$.

If all rows of $X'$ are already minimal, then $X'$ is a solution, Algorithm 1 terminates.

If all rows with indices in $H$ are already minimal, then the leading eigenvalue of $\tilde{X}'$ is simple (Theorem 2 below). We compute the minimal leading eigenvector $\tilde{v}'$ of $\tilde{X}'$. If $\tilde{v}' > 0$, then denote $v' = \tilde{v}'$, $\tilde{X}' = X'$, keep the same $S$ and go to the next step (**). Otherwise we set $S = \text{supp } \tilde{v}' = S$, restrict everything to this set and go to (**). Thus, we have a sequence of matrices $X'$ with simple leading eigenvalues, unless the support $S$ gets smaller. Then we pass to the smaller support (i.e., to a submatrix), again obtain a sequence of matrices with simple leading eigenvalue, unless the support gets smaller, etc. We do it until in some step not all rows with indices in $H$ are minimal. In this case $\rho(X') < \rho(X)$. We set $X^{(k+1)} = \tilde{X}$ and go to the next iteration (*).

In Algorithm 1 we have three main components:

1) Invariants. In each iteration we have a Boolean matrix $X$, its minimal leading eigenvector $v$ and a set of indices $S = \text{supp } v$;

2) Progress measure. The spectral radius $\rho(X)$ strictly decreases in iterations. Inside each iteration the spectral radius is the same and the index set $S$ is non-increasing. When at
some step $\rho(X)$ strictly decreases, we recompute the set $S$ and start the new iteration with this set.

Inside one step of the algorithm (in the inner loop) the algorithm does not cycle (Theorem \[\text{2}\]).

3) Stopping criterion. The algorithm stops when the current matrix $X' = X|_S$ is minimal in every row. In this case $X$ is also minimal in every row and hence (Theorem \[\text{1}\]) $X$ is a solution of problem \[\text{5}\].

**Theorem 2** Algorithm 1 is well-defined. It finds the global solution of problem \[\text{5}\] in a finite number of steps.

The well-definedness means that at each iteration matrix $X'$ has a leading eigenvector, which is unique up to a normalization. We are proving more: $X'$ has a simple leading eigenvalue. The claim that the algorithm finds the global solution in a finite time means two things: 1) all the intermediate conclusions of the algorithm are correct; 2) the algorithm does not cycle.

**Proof.** First we need to prove the correctness of intermediate conclusions: in steps 2 and 5 the matrix $X$ is optimal; in step 7 the assertion $H \subset J$ implies that $\lambda_{\text{max}}$ is simple and is not changed after this step, and in step 11, $H \not\subset J$ implies that $\lambda_{\text{max}}$ becomes strictly smaller.

Algorithm 1 is a modification of the algorithm from \[20\] derived specially for the uncertainty sets $F_i = B(A_i, r)$. The proofs for the steps 3,5,7,11 are the same as the proof of analogous Theorem 8 from \[20\]. We only replace solutions of the problem $\langle X_i, v \rangle \to \min, X_i \in B(A_i, r)$ by the minimal rows of the matrix $X$ with respect to $v$ (Lemma \[1\]) and use the criterion of the solution of problem \[5\] from Corollary \[1\]. The proof of non-cyclicity is also the same as for Theorem 8 from \[20\]. We also note that due to Theorem \[1\] Algorithm 1 runs over extreme points of the sets $B(A_i, r)$, i.e., over Boolean vectors. Since the total number of Boolean vectors is finite, the non-cyclicity implies that Algorithm 1 terminates within finite time. In step 2 the algorithm terminates since for an integer matrix $X$, $\rho(X) < 1$ means that $\rho(X) = 0$, hence $X$ has the minimal possible spectral radius.

\[\blacksquare\]
4. Solution of the max-MAS problem and generalizations

Solution of the max-MAS problem. Take \( r_0 = \|A_i\|_{\infty} \). Since the ball \( B(A, r_0) \) contains zero matrix, it follows that \( \min_{X \in B(A, r_0)} \rho(X) = 0 \). Then applying Algorithm 1 and the integer bisection on the segment \([0, r_0]\) we find the smallest \( r \) such that \( \min_{X \in B(A, r)} \rho(X) = 0 \). For this \( r \), the matrix \( \hat{X} \) provided by Algorithm 1 and the graph \( \hat{G} \) of this matrix give the answer. Solution is completed.

Before looking at numerical results and discussing the complexity, we note that several generalisations of the max-MAS problem can be solved using slight modifications of our method. Below we formulate three of them.

Problem 1. To each vertex of a graph \( G \), a non-negative integer is assigned. Make the graph acyclic by cutting at most the assigned number of incoming edges from each vertex.

Let a number \( r_i \geq 0 \) be assigned to the \( i \)th vertex, \( i = 1, \ldots, n \). Problem 1 is solved with Algorithm 1 by replacing all balls \( B(A_i, r) \) with \( B(A_i, r_i) \). If the minimal spectral radius is zero, then the answer is affirmative and the matrix of the desired graph is available.

Problem 2. Solve the max-MAS problem for a weighted graphs, with given positive weights of edges.

Solved by usual (non-integer) bisection and invoking Algorithm 1. One needs only to modify the definition of minimal row as follows: the ordering of the \( i \)th row by numbers \( v_j \) is replaced by ordering by numbers \( \alpha_{ij}v_j \), where \( \alpha_{ij} \) is the weight of an edge from \( g_j \) to \( g_i \).

Problem 3. Solve the max-MAS problem with an extra assumption that some of edges are “untouchable”, i.e., it is prohibited to cut them.

Solved as the usual max-MAX with the following modification of the definition of minimal row (Definition 1): For each \( i = 1, \ldots, n \), in \( i \)th row the positions of untouchable edges are removed from the set \( \text{supp} A_i \).

5. The max-MAS problem and an approximate solution for MAS

Having solved the max-MAS problem we obtain an acyclic graph \( \hat{G} \). This graph can be considered as an approximate solution for the MAS problem for the graph \( G \). However, usually \( \hat{G} \) has much less than \( \frac{1}{2}|E| \) edges and so gives a bad approximation for MAS. The reason is obvious: the algorithm of solving the max-MAS problem tries to cut the maximal allowed number of incoming edges from each vertex and therefore cuts more edges than needed. Nevertheless, the following modified scheme gives satisfactory results:
Algorithm 2: Algorithm 2 for approximate solution of MAS

Data: A graph $G$

Result: An acyclic subgraph $\bar{G}$, which is an approximate solution to MAS problem.

begin

1. Apply Algorithm 1 to the graph $G$. Obtain a solution $\hat{G}$ to the max-MAS problem;
2. Take the matrix $\hat{X}$ of $\hat{G}$. Find its Frobenius factorisation: $P^{-1}\hat{X}P$, where $P$ is a permutation matrix;
3. Set $Y_{ij} = [P^{-1}AP]_{ij}$ if $j > i$ and $Y_{ij} = 0$ otherwise;
4. return Set $\bar{X} = PYP^{-1}$. Then the graph $\bar{G}$ of the matrix $\bar{X}$ is an approximate solution for MAS;

end

Explanation. The solution $\hat{X}$ of max-MAS problem has spectral radius equal to zero. Hence its Frobenius factorisation $Z = P^{-1}\hat{X}P$ is upper triangular with zero diagonal. Replacing the over-diagonal part of $Z$ by the over-diagonal part of the matrix $P^{-1}AP$ we keep the spectral radius equal to zero and reduce the distance to the matrix $P^{-1}AP$. Denote the obtained matrix by $Y$. We have $\rho(Y) = 0$ and the inverse permutation $PYP^{-1}$ is an approximate solution for MAS.

Remark 2 In fact Algorithm 2 finds the enumeration of vertices in $V$ corresponding to the max-MAS solution $\hat{G}$. In this enumeration (given by the permutation matrix $P$) the matrix $\hat{X}$ has an upper triangular form with zero diagonal. Then we set $\bar{G}$ to be the acyclic graph corresponding to this enumeration.

Note that Algorithm 2 can easily be modified to find approximate solutions of several generalizations of the MAS problem that inspired by Problems 1-3 in Section 4. For example:

Find the maximal acyclic subgraph under the extra assumptions that at most $r_i$ incoming edges are cut from the $i$th vertex, $i = 1, \ldots, n$, where $\{r_i\}_{i=1}^n$ are given integers.

This corresponds to Problem 2. Problem 3 rises another variant of MAS:

Find the maximal acyclic subgraph under the extra assumptions that some edges are untouchable.

We are not aware of any known algorithms from the literature for approximate solutions of those problems.

6. Numerical efficiency and complexity issue

The algorithm of solution for the max-MAS problem demonstrates a very good efficiency. It consists in integer bisection in parameter $r$, where in each iteration of the bisection we solve problem (5) with Algorithm 1. The total number of iterations therefore does not exceed
$1 + \log_2 \|A\|_\infty$; in each step we apply Algorithm 1. Numerical results are shown in Tables 1 and 2. Table 1 demonstrates results for random graphs with sparsity $\frac{|E|}{n^2}$ between 49% and 91%, Table 2 shows results for sparsity between 5% and 74%. For each dimension $n = |\mathcal{V}|$ from 20 to 1500, we made 20 experiments and put the average number of steps ($\# \text{ steps}$) and the average running time. Let us recall that by one step we mean one computation of the leading eigenvector, because this is the most expensive operation. The experiments were done in a standard laptop. Every time we use the obtained solution for the max-MAS problem to find the approximate solution for the MAS problem (for the same graph). The rate of approximation $\gamma$ is written in the last row.

Every time Algorithm 1 finds the solution within 3-5 steps and this number grows very slow with the dimension. Then to solve the max-MAS problems we need to apply Algorithm 1 at most $\log_2 n + 1$ times. We see form the Tables that for graphs with 250 vertices the complete solution of max-MAS problem takes less than 35 steps, which is done for less than 9 seconds; for graphs with 1000 vertices the solution takes less than 11 minutes. The average rate of approximation $\gamma$ is quite stable and stays close to 0.6 for all dimensions.

| $d$ | 50  | 250 | 500  | 1000 | 1500 |
|-----|-----|-----|------|------|------|
| time | 0.36s | 8.1s | 66.42s | 622.43s | 2860.79s |
| $\# \text{ steps}$ | 17  | 34.6  | 38.5  | 44.7  | 50.3  |
| $\gamma$ | 0.644 | 0.621 | 0.615 | 0.616 | 0.616 |

Table 1. Solving the max-MAS and approximating MAS for graphs with sparsity 9 – 51%

| $d$ | 50  | 250 | 500  | 1000 | 1500 |
|-----|-----|-----|------|------|------|
| time | 0.35s | 6.56s | 61.06s | 605.73s | 2614.02s |
| $\# \text{ steps}$ | 18.9 | 32.1 | 41.8 | 43.1 | 43.7 |
| $\gamma$ | 0.6 | 0.592 | 0.592 | 0.593 | 0.592 |

Table 2. Solving the max-MAS and approximating MAS for graphs with sparsity 26 – 95%

The theoretical complexity of the max-MAS problem is not known for us. We can only conjecture that it is polynomial. The complexity of the problem of minimizing the spectral radius of product families have been studied in the literature, mostly on the Markov decision processes, see [2, 12, 27] and references therein. It is known that it is polynomial in the total cardinality of the uncertainty sets. For our problem, this does not mean the polynomial solvability, since the number of extreme points in the balls $\mathcal{B}(A_i, r)$ can be exponential. On the other hand, it was shown in [9] that for positive strictly convex smooth sets $\mathcal{F}_i$, the greedy algorithm has a quadratic convergence. This certainly explains its fast convergence for finite sets $\mathcal{F}_i$ but does not give good estimate for the theoretical complexity. At least, an idea to approximate the balls $\mathcal{B}(A_i, r)$ by convex smooth set does not work, since the
parameters of quadratic convergence depend on radii of curvature of the sets $\mathcal{F}_i$ which are too large for a tight approximation.

Acknowledgements. The work was prepared within the framework of the HSE University Basic Research Program and funded by the Russian Academic Excellence Project ’5-100’.

References

[1] M.Akian, S.Gaubert, J.Grand-Clment, and J.Guillaud, *The operator approach to entropy games*, Theory Comput. Syst. 63 (2019), no 5, 1089-1130

[2] E.Altman, *Constrained Markov decision process*, INRIA, 2004.

[3] J.Anderson, *Distance to the nearest stable Metzler matrix*, (2017), arXiv:1709.02461v1

[4] E.Asarin, J.Cervelle, A.Degorre, C.Dima, F.Horn, and V.Kozyakin, *Entropy games and matrix multiplication games*, In 33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016), February 2016, Orléans, France, 11:1 - 11:14.

[5] A.Berman and R.J.Plemmons, *Nonnegative matrices in the mathematical sciences*, Academic Press, Now York, 1979.

[6] B.Berger, P.W.Shor, *Tight bounds for the Maximum Acyclic Subgraph problem*, J. Algorithms, 25 (1997), no 1, 1–18.

[7] R. Byers, *A bisection method for measuring the distance of a stable to unstable matrices*, SIAM J. on Scientific and Statistical Computing, 9 (1988), 875–881.

[8] M.Charikar, K.Makarychev, and Yu.Makarychev, *On the advantage over random for maximum acyclic subgraph*, 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07) (2007)

[9] A.Cvetković and V.Yu.Protasov, *The greedy strategy for optimising the Perron eigenvalue*, arXiv:1807.05099 (2018).

[10] F.R.Gantmacher, *The theory of matrices*, Chelsea, New York, 2013.

[11] N. Gillis and P. Sharma, , *On computing the distance to stability for matrices using linear dissipative Hamiltonian systems*, Automatica 85 (2017), 113–121.

[12] V.Goyal, J.Grand-Clement, *A first-order approach to accelerated value iteration*, arXiv:1905.09963
[13] N. Guglielmi and M. Manetta. *Approximating real stability radii*, IMA Journal of Numerical Analysis, 35 (2015), no. 3, 1402–1425.

[14] N. Guglielmi and V. Protasov, *On the closest stable/unstable nonnegative matrix and related stability radii*, SIAM J. Matrix Anal. 39 (2018), no 4, 1642–1669.

[15] V. Guruswami, R. Manokaran, and P. Raghavendra, *Beating the random ordering is hard: inapproximability of maximum acyclic subgraph*, FOCS 2009, Conference proceedings.

[16] A.J. Hoffman and R.M. Karp, *On nonterminating stochastic games*, Management Science. Journal of the Institute of Management Science. Application and Theory Series, 12 (1966), 359-370.

[17] R. Hassin, S. Rubinstein, *Approximations for the maximum acyclic subgraph problem*, Information Processing Letters, 51 (1994) no 3, 133–140.

[18] R.M. Karp, *Reducibility among combinatorial problems. In Complexity of Computer Computations*, New York: Plenum (1972), 85-103.

[19] Y. Nesterov, V. Yu. Protasov, *Optimizing the spectral radius*, SIAM J. Matrix Anal. Appl. 34 (2013), no 3, 999–1013.

[20] Y. Nesterov, V. Yu. Protasov, *Computing closest stable non-negative matrix*, SIAM J. Matrix Anal. Appl., to appear. [http://www.optimization-online.org/DBTML/2017/08/6178.html](http://www.optimization-online.org/DBTML/2017/08/6178.html)

[21] C. Mehl, V. Mehrmann, and P. Sharma, *Stability radii for linear hamiltonian systems with dissipation under structure-preserving perturbations*, SIAM Journal on Matr. Anal. Appl., vol. 37 (2016), no. 4, 1625–1654.

[22] F.X. Orbandexivry, Y. Nesterov, and P. Van Dooren, *Nearest stable system using successive convex approximations*, Automatica, 49 (2013), 1195–1203.

[23] I. Post, Y. Ye, *The simplex method is strongly polynomial for deterministic markov decision processes*, Mathematics of Operations Research, published online (2015), https://doi.org/10.1287/moor.2014.0699

[24] V. Yu. Protasov, *The spectral simplex method*, Math. Prog., 156 (2016), 485–511

[25] U.G. Rothblum, *Multiplicative Markov decision chains*, Math. Oper. Research, 9(1984), 6-24.

[26] R. Tarjan, *Depth first search and linear graph algorithms*, SIAM J. Comput. 1 (2) (1972), 146-160.
[27] Y.Ye, *Policy-iteration methods are strongly polynomial for the MDP with a fixed discount rate*, Mathematics of Operations Research, published online (2011), https://doi.org/10.1287/moor.1110.0516