SIAM J. MATRIX ANAL. APPL.
Vol. 41, No. 3, pp. 1167–1182


# MAXIMAL ACYCLIC SUBGRAPHS AND CLOSEST STABLE MATRICES*

ALEKSANDAR CVETKOVIĆ† AND VLADIMIR YU. PROTASOV‡

**Abstract.** We develop a matrix approach to the maximal acyclic subgraph (MAS) problem by reducing it to find the closest nilpotent matrix to the matrix of the graph. Using recent results on the closest Schur stable systems and on minimizing the spectral radius over special sets of nonnegative matrices, we obtain an algorithm for finding an approximate solution of the MAS problem. Numerical results for graphs from 50 to 1500 vertices demonstrate its fast convergence and give a rate of approximation larger than 0.6 in most cases. This method also gives the precise solution for the following weakened version of MAS: find the minimal $r$ such that the graph can be made acyclic by cutting at most $r$ incoming edges from each vertex. We also consider several modifications in the case when each vertex is assigned its own maximal number $r_i$ of cut edges, and some of the edges are "untouchable." Some applications are discussed.

**Key words.** acyclic graph, nonnegative matrix, spectral radius, relaxation algorithm, closest stable matrix, spectrum of a graph

**AMS subject classifications.** 05C50, 15B48, 90C26

**DOI.** 10.1137/19M1305422

## 1. Introduction.

Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with the set of vertices $\mathcal{V} = \{g_1, \dots, g_n\}$ and with the set of edges $\mathcal{E}$. The vertices will be identified with the corresponding numbers $\{1, \dots, n\}$. The *maximal acyclic subgraph* (MAS) of $G$ is a graph $\widehat{G} = (\mathcal{V}, \widehat{\mathcal{E}})$ such that $\widehat{\mathcal{E}} \subset \mathcal{E}$, $\widehat{G}$ has no cycles, and the number of edges $|\widehat{\mathcal{E}}|$ is maximal. The MAS problem (in short, MAS) is to find a maximal acyclic subgraph of a given graph $G$. This problem was included by Karp in 1972 in his list of 21 NP-complete problems [20].

Most of methods in the literature usually find approximate solutions to the MAS problem. We say that a subgraph $G'$ of $G$ gives an approximate solution with an *approximation factor* $\delta \in (0, 1)$ if $G$ is acyclic and has at least $\delta\,|\widehat{\mathcal{E}}|$ edges. It is not known if efficient algorithms exist for obtaining an approximate solution with a factor $\delta > \frac{1}{2}$, and most likely the answer is negative [17].

It is interesting that the simplest algorithm for an approximate solution of MAS gives the best known result of $\delta = \frac{1}{2}$. Indeed, take an arbitrary renumbering of vertices of $G$ (a permutation $\sigma$ of the set $\{1, \dots, n\}$); then both the set of forward edges in the new ordering and the set of backward edges are acyclic, and one of them contains at least $\frac{1}{2}\,|\mathcal{E}| \geq \frac{1}{2}\,|\widehat{\mathcal{E}}|$ 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 largest number of forward edges. See [6, 9, 17, 18, 27] for algorithms of approximate solutions and for discussions of the complexity.

*Received by the editors December 9, 2019; accepted for publication (in revised form) by N. Gillis May 27, 2020; published electronically August 11, 2020.

https://doi.org/10.1137/19M1305422

**Funding:** The second author was supported by Russian Foundation for Basic Research projects 19-04-01227 and 20-01-00469. The second author's research was performed within the Basic Research Program of the National Research University Higher School of Economics and was supported within the 5-100 Program for State Support of Leading Universities of the Russian Federation.

†GSSI, L'Aquila, AQ, 67100, Italy (alegzander88@gmail.com).

‡DISIM, University of L'Aquila, L'Aquila, AQ, 67100, Italy, and National Research University Higher School of Economics, Moscow 101000, Russia (v-protassov@yandex.ru).

We develop a linear algebraic approach to MAS. First, we reformulate this problem as finding the closest (in the Frobenius norm) nonnegative matrix $X$ to a given matrix $A$ with the condition that $\rho(X) = 0$ (the spectral radius is equal to zero). Note that this problem is related to finding the closest Schur stable matrix to a given matrix $A$. That stabilization 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 contribute much to the MAS problem. Nevertheless, by replacing the Frobenius norm with the $L_\infty$-norm, we obtain the Schur stabilization problem, which can be effectively solved [26]. In graph terms, this replacement leads to the following problem: *find the minimal number $r$ 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 that solves this problem. For graphs with 100 vertices, finding the solution usually takes a few seconds on a standard laptop (detailed characteristics are given in section 6), and for 1000 vertices, it takes about 10 minutes. See section 6 for a discussion of numerical aspects. The algorithm is based on minimizing the Perron eigenvalue on special sets of matrices. In section 4 we formulate several generalizations of our problem which can be solved by similar methods. In section 5 we come back to the (classical) MAS problem. We show that a slight modification of our method can be used to find an approximate solution of MAS. The rate of approximation is estimated from below by the value $\gamma = \frac{|\mathcal{E}'|}{|\mathcal{E}|}$, where $\mathcal{E}'$ is the set of edges of the obtained acyclic subgraph. Since $\delta = \frac{|\mathcal{E}'|}{|\widehat{\mathcal{E}}|} \geq \frac{|\mathcal{E}'|}{|\mathcal{E}|}$, we see that $\delta \geq \gamma$. In the numerical experiments, our method gives a rate of approximation better than $\gamma \approx 0.6$. In section 6 we present numerical results and discuss the complexity issue, and in section 7 we consider some applications.

Throughout the paper, by the graph of a nonnegative $n \times n$ matrix $A$ we mean the graph $G$ with $n$ vertices $\{1, \dots, n\}$ such that there is an edge from a vertex $i$ to a vertex $j$ if and only if $A_{ji} > 0$. We use a standard componentwise ordering of real matrices $A \geq (>) B$ if $A_{ij} \geq (>) B_{ij}$ for all $i, j$, and we use 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 [13, Chapter 8], if $A \geq 0$, then there is an eigenvalue of $A$ denoted $\lambda_{\max}$ equal to the spectral radius of $A$, and there is a nonnegative eigenvector of $A$ corresponding to $\lambda_{\max}$. This eigenvalue and the corresponding eigenvector are referred to as *leading*. The vectors will be denoted by bold letters and their components by standard letters, e.g., $\boldsymbol{x} = (x_1, \dots, x_n)$. Matrices (vectors) with entries 0 and 1 are Boolean. In particular, an adjacency matrix of a graph ($A_{ji} = 1 \Leftrightarrow$ there is an edge from $i$ to $j$; otherwise, $A_{ji} = 0$) is Boolean.

We use the notation $\mathcal{V} = \{1, \dots, n\}$. 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 $\mathcal{S} \subset \mathcal{V}$, we denote by $X|_{\mathcal{S}}$ the $|\mathcal{S}| \times |\mathcal{S}|$ principal submatrix of $X$, which is a restriction of $X$ to the index set $\mathcal{S}$. By $\boldsymbol{x}|_{\mathcal{S}}$ we denote the restriction of the vector $\boldsymbol{x} \in \mathbb{R}^n$ to the $|\mathcal{S}|$-dimensional space of vectors with all nonzero components in $\mathcal{S}$.

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

**2. The MAS problem versus the closest stable matrix problem.** The MAS problem can be formulated with spectra of graphs by applying the following well-known fact: *a directed graph is acyclic if and only if the spectral radius of its*

*adjacency matrix is zero.* Indeed, if $\rho(A) = 0$, then $A^n = 0$ (this follows easily if we write $A$ in its Jordan normal form). Hence, the graph $G$ has no walks of length $n$. Therefore, it cannot have cycles; otherwise, a cycle generates walks of all lengths. Conversely, if $G$ is acyclic, then it cannot have long walks, and hence some high power of $A$ is equal to zero, which implies $\rho(A) = 0$.

The following simple observation presents the MAS problem in 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 the Frobenius norm) nonnegative matrix $X$ with zero spectral radius:*

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

*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., equivalent to the MAS problem. ∎

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

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

This is the same as problem (1) except with $\rho = 0$ replaced by $\rho = 1$. The closest stable matrix problem has been studied in many recent works due to its application in dynamical and controlled systems, mathematical economics, population dynamics, etc. (see [8, 14, 15, 24, 28] for the problem with general matrices and [3, 26, 16] for nonnegative matrices). The question arises of whether the methods for 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 [16]. 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 [26] finds the global minimum in the problem (2), with the $L_\infty$-norm being quite fast even in high dimensions (several thousands). By 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.*

Suppose a graph $G$ is made acyclic by cutting some edges. If $c_i$ denotes the number of cut incoming edges from the $i$th vertex, then the MAS problem minimizes

the sum $\sum_{i=1}^{n} c_i$, while the max-MAS problem minimizes $\max_{i=1,\dots,n} c_i$. This justifies our terminology. In the next section we will see that this problem can be efficiently solved by the method of finding the closest stable matrix [26]. Note that in spite of their similarity, problems (1) and (2) are different, since the sets of matrices with $\rho(X) = 1$ and $\rho(X) = 0$, respectively, 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 generalizations of max-MAS, for example, when it is allowed to cut at most $r_i$ incoming edges from the vertex $i$, where $r_1, \dots, r_n$ 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 discussed in section 6. In section 7 we discuss possible applications and consider the application to small-world networks in detail.

**3. Algorithmic solution for the max-MAS problem.** We first reformulate the MAS and max-MAS problems in terms of optimizing the spectral radius of a nonnegative matrix. Then we establish a relation between those problems and the problem of stabilization of a positive linear system. Then we apply the technique of stabilization to the construction of acyclic graphs. We will see that the max-MAS problem can be solved completely by this approach. Approximate solutions for MAS will be considered in section 5.

**3.1. Spectral formulation of the max-MAS problem.** The max-MAS problem is formulated in linear algebraic terms as follows:

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

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 1. This problem, in turn, is reduced to

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

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

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

Now we focus on the following problem, which is equivalent to (4): for a given Boolean matrix $A$ and for $r \in \mathbb{N}$, solve

$$\begin{cases} \rho(X) \ \to \ \min, \\ X \ \in \ \mathcal{B}(A, r)\,. \end{cases} \tag{5}$$

The solution $X$ is always a Boolean matrix. If we find the smallest nonnegative integer $r$ for which this problem has a solution with the objective function value equal to zero, then the graph of this solution $X$ solves the max-MAS problem.

**3.2. Minimization of spectral radius over product families of nonnegative matrices.** The problem (5) can be efficiently solved for each $r$. This is because $\mathcal{B}(A,r)$ is a *product set* of matrices. Optimizing the spectral radius over product sets has been investigated in various contexts [25, 30, 26, 2]. Let us have arbitrary compact sets $\mathcal{F}_i \in \mathbb{R}^n_+$. The set of matrices $\mathcal{F} = \{X \mid X_i \in \mathcal{F}_i,\, i = 1,\dots,n\}$ is called a product set. Every matrix from a product set is composed of rows arbitrarily and independently chosen from the sets $\mathcal{F}_i$. The sets $\mathcal{F}_i$ are usually referred to as *uncertainty sets*. All methods of minimizing the spectral radius over product families are actually based on the well-known formula

$$\rho(X) \;=\; \sup\big\{\,\lambda \;\big|\; \exists\, \boldsymbol{u} \geq 0\,,\, \boldsymbol{u} \neq 0 \;:\; X\,\boldsymbol{u} \;\geq\; \lambda\,\boldsymbol{u}\,\big\}\,, \tag{6}$$

which holds for every nonnegative matrix [5]. If we minimize the spectral radius of a matrix over a product set, this formula allows us to treat all rows separately:

$$\min_{X\in\mathcal{F}} \rho(X) \;=\; \sup\big\{\,\lambda \;\big|\; \exists\, \boldsymbol{u} \geq 0\,,\; \boldsymbol{u} \neq 0 \;:\; \langle X_i\,,\, \boldsymbol{u}\rangle \;\geq\; \lambda\, u_i\,, X_i \in \mathcal{F}_i,\, i = 1,\dots,n\big\}\,. \tag{7}$$

If all of the uncertainty sets $\mathcal{F}_i$ are finite, then there are methods for applying linear programming to this problem. Their complexity, however, depends on the cardinalities of the uncertainty sets [29]. In our case, the sets $\mathcal{F}_i = \mathcal{B}(A_i,r)$ 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 [2, 26] and called *the greedy algorithm*. A crucial point of this approach is the following fact.

THEOREM A (see [25]). *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 $\boldsymbol{v}$ such that $\langle X_i, \boldsymbol{v}\rangle = \min_{\boldsymbol{x}\in\mathcal{F}_i}\langle \boldsymbol{x}, \boldsymbol{v}\rangle$ for each $i = 1,\dots,n$.*

The relaxation scheme works as follows. If in the $k$th iteration we have a matrix $X^{(k)} \in \mathcal{F}$, then we compute its leading eigenvector $\boldsymbol{v}^{(k)}$, and then for every $i = 1,\dots,n$, we replace the $i$th row of $X^{(k)}$ by the element $X_i \in \mathcal{F}_i$ which minimizes the scalar product $\langle X_i, \boldsymbol{v}^{(k)}\rangle$. We obtain $X^{(k+1)}$, make the next iteration, 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 $k$ (perhaps nonstrictly). 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 is true, for instance, when all vectors from all of the sets $\mathcal{F}^{(k)}$ are strictly positive, or when each matrix from the product family $\mathcal{F}$ is irreducible (does not have a nontrivial invariant coordinate subspace, i.e., a subspace spanned by several vectors of the canonical basis).

This scheme, however, has a serious disadvantage: without those "positivity-like" assumptions, the algorithm may cycle. Moreover, for sparse matrices, the cyclicity occurs quite often. A modified greedy algorithm which never cycles was presented in [26]. According to numerical experiments, that algorithm has a very fast convergence and finds the minimum within a few iterations even in high dimensions [11]. Of course, if some of the sets $\mathcal{F}_i$ are of large cardinality, one iteration may take long, since it requires computing $|\mathcal{F}_i|$ scalar products. This occurs, in particular, in our problem (5), where each set $\mathcal{F}_i = \mathcal{B}(A_i,r)$ has an exponential number of extreme

points. In this section we overcome this difficulty and modify the greedy algorithm specifically for these sets. To this end, we begin by introducing some further notation.

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

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

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

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

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

LEMMA 1. *A row $X_i$ is minimal with respect to a vector $\boldsymbol{v}$ in the ball $\mathcal{B}(A_i, r)$ precisely when*

$$\langle X_i, \boldsymbol{v} \rangle = \min_{\boldsymbol{x} \in \mathcal{B}(A_i, r)} \langle X_i, \boldsymbol{v} \rangle . \tag{8}$$

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

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

THEOREM 1. *Let $A$ be a Boolean matrix, and let $r \in \mathbb{N}$. Let the problem* (5) *reach its global minimum at some Boolean matrix $X$. Then this matrix $X$ is characterized by the following property: there exists a leading eigenvector $\boldsymbol{v}$ of $X$ such that each row $X_i$ is minimal in the ball $\mathcal{B}(A_i, r)$ with respect to $\boldsymbol{v}$.*

*Proof.* The proof follows by combining Lemma 1 and Theorem A for the sets $\mathcal{F}_i = \mathcal{B}(A_i, r)$ and for the leading eigenvector $\boldsymbol{v}$ of $X$. ∎

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

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

If $\boldsymbol{v} > 0$, then Corollary 1 coincides with Theorem 1. If $\boldsymbol{v}$ has some zero components, then the criterion of Corollary 1 is simpler in practice since it involves only the sumbatrix $X' = X|_{\mathcal{S}}$.

**3.3. Some auxiliary facts on nonnegative matrices.** A nonnegative 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 $\Lambda \subset \mathcal{V}$ such that for each $i \in \Lambda$, the support of the $i$th column of $A$ is contained in $\Lambda$. It is well known (see, 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.

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

$$P^{-1}AP = \begin{pmatrix} A_1 & * & \dots & * \\ 0 & A_2 & * & \vdots \\ \vdots & & \ddots & * \\ 0 & \dots & 0 & A_q \end{pmatrix}. \tag{9}$$

For each $j = 1, \dots, q$, the matrix $A_j$ in the $j$th diagonal block is irreducible. Any nonnegative matrix possesses a unique Frobenius factorization up to a permutation of blocks (see [13, Chapter 8]).

Let $A$ be an $n \times n$ nonnegative matrix. Its leading eigenvector $\boldsymbol{v}$ 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 normalization 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 $\boldsymbol{u}$ of the principal submatrix of $A$ that consists of blocks $A_1, \dots, A_m$. Complement this vector with zeros to an $n$-dimensional vector and obtain a minimal eigenvector $\bar{\boldsymbol{u}}$ of the matrix $P^{-1}AP$. Respectively, the vector $P\bar{\boldsymbol{u}}$ 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 (see [26]). *If a matrix $A \geq 0$ has a strictly positive minimal leading eigenvector $\boldsymbol{v}$, then its leading eigenvalue is simple. In the Frobenius factorization* (9)*, the spectral radius of $A_q$ is equal to $\rho(A)$, and the spectral radii of all other blocks are smaller than $\rho(A)$.*

In the case of a 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 and then so is the leading eigenvalue 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 factorization is available.

PROPOSITION 3 (see [26]). *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 $P$, correspond to the last block $A_q$ in the Frobenius factorization* (9).

Thus, if $A$ has a strictly positive minimal leading eigenvector, then the basic set of $A$ can be found as follows. We consider the permutation $P$ of the basis vectors which takes $A$ to its Frobenius form (9). Then we take the last block $A_q$ in this factorization and denote its size by $d$. Then the basic set is $\{k \mid e_k = P^{-1}e_j,\ j = n-d+1, \dots, n\}$, where $\{e_j\}_{j=1}^n$ is the canonical basis in $\mathbb{R}^n$. In particular, if $A$ has a Frobenius form, i.e., if $P$ is the identity permutation, then the basic set is $\{n-d+1, \dots, n\}$.

*Proof of Proposition* 3. 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 factorization (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 for minimizing the spectral radius over an $L_1$-ball.** We describe the algorithm for solving problem (5) of minimizing the spectral radius over an $L_1$-ball $\mathcal{B}(A, r)$, by which we find the closest stable matrix on a product family. The radius $r$ is assumed to be a natural number.

**Notation for Algorithm 1.** Let $\mathcal{V} = \{1, \dots, n\}$. For a matrix $X$, vector $\boldsymbol{x}$, and a subset $K \subset \mathcal{V}$, we denote by $X|_K$ and $\boldsymbol{x}|_K$, respectively, the restrictions of the matrix and the vector to the index set $K$ (see the introduction). We use the notions of minimal leading eigenvector (Definition 1) and basic set (Definition 2). We use the obvious fact that if $\boldsymbol{v}$ is a minimal leading eigenvector of a matrix $X$ and $\mathcal{S} = \operatorname{supp} \boldsymbol{v}$, then the matrix $X|_{\mathcal{S}}$ has a strictly positive minimal leading eigenvector $\boldsymbol{v}|_{\mathcal{S}}$.

**3.5. Explanation and proof for Algorithm 1.**

**Explanation for Algorithm 1.** The algorithm is a relaxation scheme: the value of the spectral radius $\rho(X)$ is nonincreasing during the whole algorithm.

The algorithm consists of finitely many iterations (*), and each iteration consists of several steps (**). In one iteration, during all steps except the last, the value $\rho(X)$ is the same. After the last step, this value becomes strictly smaller; then the iteration is completed, and we start the next iteration.

**Algorithm 1:** Algorithm for minimizing the spectral radius over an $L_1$-ball.

**Data:** A Boolean $n \times n$ matrix $A$, a number $r \in \mathbb{N}$
**Result:** A matrix $\widehat{X}$ which is a solution of problem (5).
**begin**
  Set $X^{(1)} = A$, $k = 1$.
1 **(*)** $k$`th iteration.` We have a Boolean $n \times n$ matrix $X^{(k)}$.
2 **if** $\rho(X^{(k)}) < 1$ **then**
    **STOP**. Algorithm 1 terminates. Denote $\widehat{X} = X^{(k)}$ and **Return**;
  **else**
3 Denote $X = X^{(k)}$. Compute a minimal leading eigenvector $\boldsymbol{v}$ of $X$ (take any of them, if there are several), set $\mathcal{S} = \mathcal{S}^{(k)} = \operatorname{supp} \boldsymbol{v}$, $X' = X|_{\mathcal{S}}$, $\boldsymbol{v}' = \boldsymbol{v}|_{\mathcal{S}}$;
4 **(**)** `Main loop.` *Input*: a triple $(X, \boldsymbol{v}, \mathcal{S})$, where $X$ is a Boolean $n \times n$ matrix, $\boldsymbol{v}$ is a minimal leading eigenvector of $X$, and $\mathcal{S} = \operatorname{supp} \boldsymbol{v}$. *Output*: a new triple $(X, \boldsymbol{v}, \mathcal{S})$.
    Let $\mathcal{H} \subset \mathcal{S}$ be the basic set of $X'$, and let $\mathcal{I}$ be the set of indices $i \in \mathcal{S}$ for which the row $X'_i$ is minimal in the ball $\mathcal{B}(A'_i, \boldsymbol{v}')$ with respect to the vector $\boldsymbol{v}'$.
5 **if** $\mathcal{I} = \mathcal{S}$ **then**
      **STOP**. Algorithm 1 terminates. Define the $n \times n$ matrix $\widehat{X}$ as follows: $\widehat{X}_i = X_i$ for $i \in \mathcal{S}$ and $\widehat{X}_i = A_i$ for $i \notin \mathcal{S}$. **Return**;
    **else**
6 Define the next matrix $\tilde{X}$ as follows:
      If $i \in \mathcal{I}$ or $i \notin \mathcal{S}$, then $\tilde{X}_i = X_i$;
      Otherwise, if $i \in \mathcal{S} \setminus \mathcal{I}$, then $\tilde{X}_i|_{\mathcal{S}}$ is the minimal row in the ball $\mathcal{B}(A'_i, r)$ with respect to $\boldsymbol{v}'$ and $\tilde{X}_{ij} = A_{ij}$ for all $j \notin \mathcal{S}$.
      Set $\tilde{X}' = \tilde{X}|_{\mathcal{S}}$ ;
7 **if** $\mathcal{H} \subset \mathcal{I}$ **then**
        $\rho(\tilde{X}') = \rho(X')$, the leading eigenvalue of $\tilde{X}'$ is simple;
8 We compute the minimal leading eigenvector $\tilde{\boldsymbol{v}}'$ of $\tilde{X}'$. The set $\mathcal{S}$ is not changed. Set $X = \tilde{X}, \boldsymbol{v} = \tilde{\boldsymbol{v}}$;
9 **if** $\tilde{\boldsymbol{v}}' > 0$ **then**
          Go to (**);
        **else**
10 set $\mathcal{S} = \operatorname{supp} \tilde{\boldsymbol{v}}'$, $X' = \tilde{X}|_{\mathcal{S}}$, and $\boldsymbol{v}' = \tilde{\boldsymbol{v}}|_{\mathcal{S}}$. Go to (**).
      **else**
11 we have $\mathcal{H} \not\subset \mathcal{I}$ and $\rho(\tilde{X}) < \rho(X)$. Set $X^{(k+1)} = \tilde{X}$ and go to the next $(k+1)$th iteration (*);
12 **return** $\widehat{X}$ *is a solution;*

(*) In the $k$th iteration, we have a matrix $X = X^{(k)}$. We find its minimal leading eigenvector $\boldsymbol{v}$ and denote $\mathcal{S} = \operatorname{supp} \boldsymbol{v}$. Then until the end of this iteration, we work on the set $\mathcal{S}$ without involving other indices. We consider the matrix $X'$, which is a restriction of $X$ to the set $\mathcal{S}$. Then the vector $\boldsymbol{v}' = \boldsymbol{v}|_{\mathcal{S}}$ is a positive minimal leading

eigenvector of $X'$. We find the basic set $\mathcal{H}$ of $X'$ (Definition 1); see (10):

$$X^{(k)} = \begin{array}{|c|c|} \multicolumn{2}{c}{\overbrace{\phantom{XX}}^{\mathcal{S}}} \\ \hline X & * \\ \hline 0 & * \\ \hline \multicolumn{2}{c}{\underbrace{\phantom{XX}}_{\Omega}} \end{array}; \qquad X = \begin{array}{|c|c|} \multicolumn{2}{c}{\overbrace{\phantom{XXX}}^{\mathcal{S}}} \\ \hline B & * \\ \hline 0 & C \\ \hline \multicolumn{2}{c}{\underbrace{\phantom{XX}}_{\mathcal{H}}} \end{array}. \tag{10}$$

(**) The input of this loop is a triple $(X, \boldsymbol{v}, \mathcal{S})$, where $X$ is Boolean $n \times n$ matrix, $\boldsymbol{v}$ is a minimal leading eigenvector of $X$, and $\mathcal{S}$ is the support of $\boldsymbol{v}$. The output is a new triple $(X, \boldsymbol{v}, \mathcal{S})$, where $X$ is a modified matrix of the same size with either the same set $\mathcal{S}$ or a new (smaller) set $\mathcal{S}$.

We denote $X' = X|_{\mathcal{S}}$, $\boldsymbol{v}' = \boldsymbol{v}|_{\mathcal{S}}$ and replace all rows of $X'$ by the minimal rows in the corresponding balls $\mathcal{B}(A'_i, r)$ with respect to $\boldsymbol{v}'$. 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 $\mathcal{H}$, where $\mathcal{H} \subset \mathcal{S}$ is the basic set of $X'$, are already minimal, then the leading eigenvalue of $\tilde{X}'$ is simple (see Theorem 2 below).

We compute the minimal leading eigenvector $\tilde{\boldsymbol{v}}'$ of $\tilde{X}'$. If $\tilde{\boldsymbol{v}}' > 0$, then denote $\boldsymbol{v}' = \tilde{\boldsymbol{v}}'$, $\tilde{X}' = X'$, keep the same $\mathcal{S}$, and go to the next step (**). Otherwise, we set $\mathcal{S} = \operatorname{supp} \tilde{\boldsymbol{v}}' = \mathcal{S}$, restrict everything to this set, and go to (**). Thus, we have a sequence of matrices $X'$ with simple leading eigenvalues (unless the support $\mathcal{S}$ 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 this until, in some step, not all rows with indices in $\mathcal{H}$ 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 the following three main components:

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

(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 $\mathcal{S}$ is nonincreasing. When, at some step, $\rho(X)$ strictly decreases, we recompute the set $\mathcal{S}$ and start the new iteration with this set.

Inside one step of the algorithm (in the inner loop), the algorithm does not cycle (see Theorem 2).

(3) *Stopping criterion.* The algorithm stops when the current matrix $X' = X|_{\mathcal{S}}$ is minimal in every row. In this case, $X$ is also minimal in every row, and hence (see Theorem 1) $X$ is a solution of problem (5).

Now we are going to show that Algorithm 1 is well-defined and always finds a solution within finite time. The well-definedness means that at each iteration, matrix $X'$ has a leading eigenvector, which is unique up to normalization. Additionally, we prove that $X'$ has a simple leading eigenvalue. The claim that the algorithm finds the global solution in a finite time means the following two things: (1) All statements

formulated in the description of the algorithm are correct, and (2) the algorithm does not cycle.

THEOREM 2. *Algorithm* 1 *is well-defined. It finds the global solution of problem* (5) *in a finite number of steps.*

*Proof.* First, we need to prove the correctness of interim conclusions: in steps 2 and 5, the matrix $X$ is optimal; in step 7, the assertion $\mathcal{H} \subset \mathcal{J}$ implies that $\lambda_{\max}$ is simple and is not changed after this step; and in step 11, $\mathcal{H} \not\subset \mathcal{J}$ implies that $\lambda_{\max}$ becomes strictly smaller. Algorithm 1 is a modification of the algorithm from [26] derived specifically for the uncertainty sets $\mathcal{F}_i = \mathcal{B}(A_i, r)$. The proofs for steps 3, 5, 7, and 11 are the same as the proof of analogous Theorem 8 from [26]. We only replace solutions of the problem $\langle X_i, \boldsymbol{v}\rangle \to \min,\ X_i \in \mathcal{B}(A_i, r)$ by the minimal rows of the matrix $X$ with respect to $\boldsymbol{v}$ (see Lemma 1) and use the criterion of the solution of problem (5) from Corollary 1. The proof of noncyclicity is also the same as that for Theorem 8 from [26]. We also note that due to Theorem 1, Algorithm 1 runs over extreme points of the sets $\mathcal{B}(A_i, r)$, i.e., over Boolean vectors. Since the total number of Boolean vectors is finite, the noncyclicity 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 $\rho(X) = 0$, and hence $X$ has the minimal possible spectral radius. ∎

## 4. Algorithmic solution of the max-MAS problem and generalizations.

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

Before looking at numerical results and discussing the complexity, we note that several generalizations 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 nonnegative 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 $\mathcal{B}(A_i, r)$ with $\mathcal{B}(A_i, r_i)$. 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.*

This is solved by usual (noninteger) bisection and invoking Algorithm 1. One need only modify the definition of minimal row as follows: the ordering of the $i$th row by numbers $v_j$ is replaced with 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 the edges are "untouchable," i.e., it is prohibited to cut them.*

This is 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 the $i$th row the positions of untouchable edges are removed from the set $\operatorname{supp} A_i$.

**5. The max-MAS problem and an approximate solution for the classical MAS problem.** Having solved the max-MAS problem, we obtain a matrix $\widehat{X}$ and the corresponding acyclic graph denoted by $G_0$. This graph can be considered as an approximate solution for the MAS problem for the graph $G$. However, usually $G_0$ has fewer than $\frac{1}{2}|\mathcal{E}|$ edges and so gives a bad approximation for MAS. The reason is obvious: the algorithm for 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 for approximate solution of MAS.

**Data:** A graph $G$

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

**begin**

  Apply Algorithm 1 to the graph $G$. Obtain a solution $G_0$ to the max-MAS problem;

1 Take the matrix $\widehat{X}$ of $G_0$. Find its Frobenius factorization: $P^{-1}\widehat{X}P$, where $P$ is a permutation matrix;

2 Set $Y_{ij} = \left[P^{-1}AP\right]_{ij}$ if $j > i$ and $Y_{ij} = 0$ otherwise;

3 **return** *Set* $\bar{X} = PYP^{-1}$. *Then the graph* $\bar{G}$ *of the matrix* $\bar{X}$ *is an approximate solution for MAS;*

**Explanation for Algorithm 2.** The solution $\widehat{X}$ of the max-MAS problem has spectral radius equal to zero. Hence, its Frobenius factorization $Z = P^{-1}\widehat{X}P$ is upper triangular with zero diagonal. Replacing the overdiagonal part of $Z$ by the overdiagonal 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 ordering of vertices in $\mathcal{V}$ corresponding to the max-MAS solution $G_0$. In this ordering (given by the permutation matrix $P$) the matrix $\widehat{X}$ 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 be easily modified to find approximate solutions of several generalizations of the MAS problem that were inspired by Problems 1–3 in section 4. We have the following 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 raises another variant of MAS as follows:

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

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

**6. Numerical efficiency and the complexity issue for the algorithmic solution of the MAS problem.** In this section we demonstrate the practical efficiency of our methods. We show and discuss the results of numerical experiments with random graphs of various densities for both the MAX and max-MAS problems. Then we discuss the complexity issue.

**6.1. Numerical results for random graphs of various density.** The algorithm for the max-MAS problem demonstrates very good efficiency. It consists of 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 $\frac{|\mathcal{E}|}{n^2}$ between 49% and 91%, and Table 2 shows results for sparsity between 5% and 74%. For each dimension $n = |\mathcal{V}|$ from 20 to 1500, we performed 20 experiments and set the average number of steps (*# 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 numerical experiments were performed on a standard laptop with the following specifications: Dell XPS 13 with Intel Core i7-6500U CPU @ 2.50GHz and 8GB RAM. The algorithm was coded in Python, and the code for the algorithm can be found on https://github.com/ringechpil/thesis_codes.

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.

In all of our examples, Algorithm 1 finds the solution within 3–5 steps, and this number grows very slowly with the dimension. Then to solve the max-MAS problem we need to apply Algorithm 1 at most $\log_2 n + 1$ times. We see from Tables 1 and 2 that for graphs with 250 vertices the complete solution of the 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.

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

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

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

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

**6.2. The complexity of Algorithm 1 and comparison with other algorithms.** The theoretical complexity of the max-MAS problem is not known to us. We can only conjecture that it is polynomial. It was shown in [11] that for positive strictly convex smooth sets $\mathcal{F}_i$, the greedy algorithm has quadratic convergence. This certainly explains the fast convergence of Algorithm 1 for finite sets $\mathcal{F}_i$ but does not

give a good estimate for the theoretical complexity. At the least, approximating the balls $\mathcal{B}(A_i, r)$ by a convex smooth set does not work, since the parameters of quadratic convergence depend on radii of the curvature of the sets $\mathcal{F}_i$, which are too large for a tight approximation.

There are many algorithms in the literature for the approximate solution of MAS; for short surveys see [9, 18] and the references therein. To the best of our knowledge, the most effective algorithms have the approximation factor $\delta = 0.5 + \Omega(\frac{\alpha}{\log n})$, where $0.5 + \alpha = \frac{|\widehat{\mathcal{E}}|}{|\mathcal{E}|}$ is the fraction of the original edges contained in the maximal acyclic subgraph [9]. Let us recall that for positive functions $f, g$, the symbol $f = \Omega(g)$ means that there is a constant $C > 0$ for which $\Omega(g(t)) \geq C\, g(t)$ for all $t$. The numerical results presented here show that our approach outputs graphs with $\delta \geq \gamma \approx 0.6$, even for large size $n$ or large density. Moreover, this estimate does not seem to decrease with the growth of $n$. Recall that $\gamma$ is a lower bound for the approximating factor $\delta$, where we replace $|\widehat{\mathcal{E}}|$ by $|\mathcal{E}|$. Therefore, the true values of $\delta$ in our examples are better (and can be much better) than those given in Tables 1 and 2. Therefore, we see that our Algorithm 1 gives, at least in the numerical examples, no worse an approximation than those known in the literature, and it performs very fast even for relatively large graphs.

We also remark that the estimates given in Tables 1 and 2 are for randomly generated graphs, with the only restriction being sparsity. As we we shall see in the next section, for graphs from applied problems, such as small-world networks, we get even better results.

**7. Applications.** Making the graph acyclic has a wide variety of applications. One is discovering hierarchies within a graph [32, 31], which is very useful for both real-world and virtual social networks. Others include testing electrical circuits [12] and "telling stories" [1], which is used in biology to represent metabolic networks that describe biochemical road maps [33]. Approximating the MAS can also be used in data flow/pipeline optimization [22], machine learning, and artificial intelligence [21, 23].

In [7] a connection between a positive linear switching system (LSS) and its asymptotic stability was established: a positive LSS is asymptotically stable if and only if its corresponding graph is acyclic. Using this fact and utilizing our max-MAS algorithm along with an LSS stabilization algorithm introduced in [10], we can cut some interdependencies and construct a stable LSS from an unstable one while closely maintaining to its original structure.

We now apply our algorithm on small-world networks. Newman–Watts–Strogatz small-world graphs can be used for modeling social networks [34, 4] (but also for modeling networks in biology, epidemiology, and neuroscience; see [19] and references therein). They can be defined using three parameters: $n$, the number of nodes forming a ring; $k$, the degree of each node, where each edge connects a node with its $k$ nearest neighbors; and $p$, the rewiring probability, i.e., the probability that an edge will be rewired from a neighboring node to some random distant node. It is instructive to have a degree $k \gg \ln n$ (but one that is not too large) in order to avoid making a graph overly dense and connected. Also, $p$ should not be too large, since rewiring too many edges causes the graph to lose its small-world structure and end up resembling a random network.

In the following series of numerical experiments, we keep the parameters $k = 25, p = 0.1$ fixed, while we vary the number of vertices $n$. The results are shown in Table 3.

TABLE 3
*Solving the max-MAS and approximating MAS for Newman–Watts–Strogatz small-world graphs with $k = 25,\ p = 0.1$.*

| $n$ | 50 | 250 | 500 | 1000 |
|---|---|---|---|---|
| time | 0.39s | 27.39s | 169.79s | 1607.81s |
| # steps | 13.6 | 43 | 53.8 | 73.4 |
| $\gamma$ | 0.61 | 0.617 | 0.618 | 0.627 |

TABLE 4
*Solving the max-MAS and approximating MAS for Newman–Watts–Strogatz small-world graphs with $n = 500,\ p = 0.1$.*

| $k$ | 5 | 10 | 25 | 100 | 250 |
|---|---|---|---|---|---|
| time | 76.07s | 76.98s | 126.63s | 138.15s | 119.65s |
| # steps | 69.8 | 39.2 | 55 | 37.6 | 30.6 |
| $\gamma$ | 0.79 | 0.672 | 0.621 | 0.544 | 0.523 |

TABLE 5
*Solving the max-MAS and approximating MAS for Newman–Watts–Strogatz small-world graphs with $n = 500,\ k = 25$.*

| $p$ | 0.02 | 0.3 | 0.6 |
|---|---|---|---|
| time | 167.51s | 80.54s | 73.94s |
| # steps | 71 | 25.6 | 22.6 |
| $\gamma$ | 0.613 | 0.62 | 0.623 |

We then keep the number of vertices and the rewiring $n = 500, p = 0.1$ fixed, while we vary the number of connected nearest neighbors $k$ (see Table 4).

We see that as the network gets denser, our algorithm tends to cut significantly more edges. In this manner, our algorithm works better for sparser graphs, which is convenient, since small-world networks are usually not dense. We also perform tests for the graphs with a fixed number of vertices and node degree $n = 500, k = 25$, and we vary the rewiring probability $p$ (see Table 5).

In regard to the number of preserved edges, we notice no significant changes as the graph's structure leans more toward the random network.

**Acknowledgments.** The authors are grateful to the anonymous referees for attentive reading and for many valuable remarks.

## REFERENCES

[1] V. Acuña, E. Birmelé, L. Cottret, P. Crescenzi, F. Jourdanf, V. Lacroixa, A. Marchetti-Spaccamela, A. Marinoe, P.V. Milreu, M.-F. Sagot, and L. Stougie, *Telling stories: Enumerating maximal directed acyclic graphs with a constrained set of sources and targets*, Theor. Comput. Sci., 457 (2012), pp. 1–9.

[2] M. Akian, S. Gaubert, J. Grand-Clément, and J. Guillaud, *The operator approach to entropy games*, Theory Comput. Syst., 63 (2019), pp. 1089–1130.

[3] J. Anderson, *Distance to the nearest stable Metzler matrix*, in Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), 2017, pp. 6567–6572.

[4] V. Arnaboldi, A. Passarella, M. Conti, and R.I.M. Dunbar, *Online Social Networks: Human Cognitive Constraints in Facebook and Twitter Personal Graphs*, Computer Science Reviews and Trends, Elsevier, 2015.

[5] A. Berman and R.J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, 1979.

[6] B. Berger and P.W. Shor, *Tight bounds for the maximum acyclic subgraph problem*, J. Algorithms, 25 (1997), pp. 1–18.

[7] C. Briat, *Sign properties of Metzler matrices with applications*, Linear Algebra Appl., 515 (2017), pp. 53–86.

[8] R. Byers, *A bisection method for measuring the distance of a stable matrix to the unstable*

*matrices*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 875–881, https://doi.org/10.1137/0909059.

[9] M. Charikar, K. Makarychev, and Yu. Makarychev, *On the advantage over random for maximum acyclic subgraph*, in Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07), Providence, RI, 2007, pp. 625–633.

[10] A. Cvetković, *Stabilizing the Metzler matrices with applications to dynamical systems*, Calcolo, 57 (2020), 1.

[11] A. Cvetković and V. Yu. Protasov, *The Greedy Strategy for Optimising the Perron Eigenvalue*, preprint, https://arxiv.org/abs/1807.05099, 2020, submitted.

[12] G. Even, J. Naor, B. Schieber, and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, J. Electronic Testing, 1 (1990), pp. 163–174.

[13] F.R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, 2013.

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

[15] N. Guglielmi and M. Manetta, *Approximating real stability radii*, IMA J. Numer. Anal., 35 (2015), pp. 1402–1425.

[16] N. Guglielmi and V. Yu. Protasov, *On the closest stable/unstable nonnegative matrix and related stability radii*, SIAM J. Matrix Anal., 39 (2018), pp. 1642–1669, https://doi.org/10.1137/18M1172454.

[17] V. Guruswami, R. Manokaran, and P. Raghavendra, *Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph*, in Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science, Philadelphia, PA, 2008, pp. 573–582.

[18] R. Hassin and S. Rubinstein, *Approximations for the maximum acyclic subgraph problem*, Inform. Process. Lett., 51 (1994), pp. 133–140.

[19] M.D. Humphries and K. Gurney, *Network "small-world-ness": A quantitative method for determining canonical network equivalence*, PLoS ONE, 3 (2008), e0002051.

[20] R.M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, Plenum, 1972, pp. 85–103.

[21] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, Adaptive Computation and Machine Learning, The MIT Press, 2009.

[22] R. Kramer, R. Gupta, and M.L. Soffa, *The combining DAG: A technique for parallel data flow analysis*, IEEE Trans. Parallel Distrib. Syst., 5 (1994), pp. 805–813.

[23] Z. Liu, K. Li, and X. He, *Cutting cycles of conditional preference networks with feedback set approach*, Comput. Intell. Neurosci., 2018 (2018), 2082875.

[24] C. Mehl, V. Mehrmann, and P. Sharma, *Stability radii for linear Hamiltonian systems with dissipation under structure-preserving perturbations*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1625–1654, https://doi.org/10.1137/16M1067330.

[25] Y. Nesterov and V. Yu. Protasov, *Optimizing the spectral radius*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 999–1013, https://doi.org/10.1137/110850967.

[26] Yu. Nesterov and V. Yu. Protasov, *Computing closest stable nonnegative matrix*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 1–28, https://doi.org/10.1137/17M1144568.

[27] A. Newman, *The maximum acyclic subgraph problem and degree-3 graphs*, in Approximation, Randomization, and Combinatorial Optimization (Berkeley, CA, 2001), Lecture Notes in Comput. Sci. 2129, Springer, Berlin, 2001, pp. 147–158.

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

[29] I. Post and Y. Ye, *The simplex method is strongly polynomial for deterministic Markov decision processes*, Math. Oper. Res., 40 (2015), pp. 797–1088, https://doi.org/10.1287/moor.2014.0699.

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

[31] J. Sun, D. Ajwani, P.K. Nicholson, A. Sala, and S. Parthasarathy, *Breaking cycles in noisy hierarchies*, in WebSci '17: Proceedings of the 2017 ACM on Web Science Conference, 2017, pp. 151–160.

[32] N. Tatti, *Faster way to agony: Discovering hierarchies in directed graphs*, in Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2014), Lecture Notes in Comput. Sci. 8726, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, eds., Springer, Berlin, Heidelberg, 2014, pp. 163–178.

[33] J.B. Toledo, M. Arnold, G. Kastenmüller, R. Chang, R.A. Baillie, X. Han, M. Thambisetty et al., *Metabolic network failures in Alzheimer's disease: A biochemical road map*, Alzheimers Dement., 13 (2017), pp. 965–984.

[34] D. Watts and S. Strogatz, *Collective dynamics of "small-world" networks*, Nature, 393 (1998), pp. 440–442.