Reachability of Consensus and Synchronizing Automata

Pierre-Yves Chevalier, Julien M. Hendrickx, Raphaël M. Jungers

Abstract—We consider the problem of determining the existence of a sequence of matrices driving a discrete-time consensus system to consensus. We transform this problem into one of the existence of a product of the transition (stochastic) matrices that has a positive column. We then generalize some results from automata theory to sets of stochastic matrices. We obtain as a main result a polynomial-time algorithm to decide the existence of a sequence of matrices achieving consensus.

I. INTRODUCTION

Consensus systems represent groups of agents trying to reach agreement on some value. They are commonly used in many distributed computation systems and have attracted much research attention in recent years. Indeed, many decentralized systems are a combination of local computing and global synchronization, and consensus systems are an appropriate tool to perform the synchronization step. The simplest consensus system consists of agents computing weighted average of values of other agents:

\[ x(t + 1) = Ax(t), \]

with \( A \), stochastic matrices, i.e., their entries are nonnegative and the entries on each row sum up to one.

Recent works have considered the problem of controlling consensus systems. This research deals for instance with finding conditions on \( A \) and \( B \) under which system

\[ x(t + 1) = Ax(t) + Bu \]

can be steered into any desired configuration [6], [18].

We consider a different kind of controllability: the system is not controlled by an exogenous input, but by choosing the matrix of interaction \( A_i \) at each time. Consider for instance a wireless network of agents trying to converge to consensus. One solution to avoid interference is to partition the agents into groups which emit at different times. Thus, the problem arises of optimally scheduling the communication protocol such that the agents converge to consensus. Our problem consists here in finding a scheduling such that the agents converge to consensus.

Formally, we study the system

\[ x(t + 1) = A_{\sigma(t)}x(t), \]
\[ x(0) = x_0, \]

and we want to solve the following decision problem.

**Problem 1.** Given a set of stochastic matrices \( S = \{A_1, \ldots, A_m\} \), does there exist a sequence \( \sigma : \mathbb{N} \to \{1, \ldots, m\} : t \mapsto \sigma(t) \) such that, for any \( x_0 \), System (1) converges to consensus, i.e., to a state multiple of \( I = (1 \ldots 1)^T \).

This problem can be seen as an open loop control problem. Indeed, it deals with the existence of a sequence \( \sigma \) that steers System (1) to consensus from any initial condition. One could have \( \sigma \) depend on the initial condition \( x_0 \) or, more generally, on the state \( x(t) \) (closed loop control), but we proved in [7, Proposition 1.b] that the two are equivalent.

Our problem can be seen as deciding stabilizability of a switched system with control on the switching signal. This problem has motivated much research effort (see, e.g. [12], [19] and [13, Section 2.2.4]) and is known to be very hard. For instance, deciding, for a matrix set \( S \), whether there is a product of matrices from \( S \) that converges to zero is an undecidable problem [16].

We will show that for consensus systems, not only is the problem decidable, but it is decidable in polynomial time. Our proof technique proceeds in two steps. Firstly, we reduce the problem to that of determining whether there exists a column-positive product, i.e., a product of transition matrices that has a positive column. We call a column positive word the sequence of indices of a column-positive product. It turns out that the existence of column-positive words has been extensively studied for sets of binary stochastic matrices (i.e., stochastic matrices with the additional constraint that the entries are in \( \{0, 1\} \)). Sets of binary stochastic matrices that have a column-positive word are called synchronizing automata. Secondly, we leverage results on synchronizing automata and extend them to sets of stochastic matrices.

A. Synchronizing Automata

Synchronizing automata appeared in theoretical computer science in the sixties and have attracted lots of research attention. An automaton is a set of states together with a set of actions and a transition function. The transition function determines the new state of the automaton depending on the action and the old state. Automata can be represented as sets of graphs where each node has outdegree one. An example is given on Figure 1. They can also be represented as sets of matrices, each matrix being the adjacency matrix of one of the graphs. Since in each graph, each node has outdegree one, their adjacency matrices have one one on each row and zero

\[^1\text{This problem is Problem 2 of our submitted journal article [7], that we restrict here to the case of stochastic matrices. We obtain here much stronger results with a different approach.}\]
everywhere else. They are, therefore, binary and stochastic. The set corresponding to the automaton of Figure 1 is

\[ S = \left\{ A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \right\}. \]

Automata were initially introduced as simple models of computing devices: the actions representing different control commands, and the states representing different possible states of the memory registers of the device. An automaton is said to be synchronizing if there is a sequence of actions, independent of the state, that reinitializes the device, i.e., that drives the memory onto a particular state. This sequence of actions is called a synchronizing word. Synchronizing automata have generated intense research efforts in theoretical computer science [1], [8], [17], [20], [21], and provided several deep results.

Recently, several works made connections between synchronizing automata and topics in systems and control. For instance, [14] applied convex optimization tools to the study of synchronizing automata and [10] linked synchronization with game theoretical concepts.

B. Equivalent Problems and Applications

The question of convergence can also be asked in the case where the matrices come each with a certain probability at every time. We will see (Proposition 2) that a positive answer to Problem 1 is equivalent to convergence to consensus with probability 1 when the switching is random, each matrix is chosen with nonzero probability and choices of matrices at different times are independent. Therefore, our investigation of Problem 1 will yield results that are also applicable to consensus systems with random switching.

Problem 1 is also equivalent to that of determining whether an inhomogeneous Markov chain may be mixing, i.e., may forget its initial condition. Indeed an inhomogeneous Markov chain can be represented as the transpose of System 1. Moreover, thanks to Proposition 2, Problem 1 is also equivalent to the question of whether an inhomogeneous Markov chain is mixing with probability 1. More generally, our results apply to any process with column-stochastic transition matrices, i.e., matrices whose transposes are stochastic. In fact, any positive linear process that preserves the sum of the elements of the state vector can be represented with column-stochastic transition matrices. A particular case is the push-sum algorithm, a decentralized method to compute an average [2], [15]. In this algorithm, agents have initial values \( x_i(0) \) and they want to compute their average. Each agent \( i \) has two values \( s_i(t) \) and \( w_i(t) \). \( s_i(t) \) is a fraction of the sum of \( x_i(0) \) s and \( w_i(t) \) is a weight value. Each agent’s estimate of the average is \( x(t) = s_i(t)/w_i(t) \). At times, an agent sends a fraction of its values \( s_i(t), w_i(t) \) to another agent. Under suitable assumptions, the ratio \( s_i(t)/w_i(t) \) converges to the average of the \( x_i(0) \) s. This process preserves the sum of the values of agents and can therefore be represented with column-stochastic transition matrices. Hence, the results presented in this article automatically apply to the convergence analysis of the push-sum algorithm.

C. Outline

In Section II we present some results on automata that we generalize later in the article. In Section III, we prove the equivalence between Problem 1 and column-primitivity of the set \( S \), i.e., the existence of a column-positive product. In Sections IV and V we extend different results known for automata to finite sets of nonnegative matrices with no zero row (of which stochastic matrices are a subset). In particular, we obtain

- a polynomial bound \( (O(n^3)) \) on the length of column-positive words (Theorem 4),
- a polynomial-time \( (O(n^3)) \) algorithm to decide the existence of a column-positive word (Theorem 5),
- a proof of NP-hardness of finding the shortest column-positive word, which also holds for sets of matrices with positive diagonals (Theorem 3).

II. CLASSICAL RESULTS ON AUTOMATA

We state in this section a couple of classical results on automata. We will extend them to finite sets of nonnegative matrices with no zero row in order to obtain results on consensus systems. We recall that an automaton is said to be synchronizing if it has a column-positive product.

Conjecture 1 (Černý Conjecture [5]). If an automaton is synchronizing then it has a synchronizing word of length at most \( (n - 1)^2 \).

The best proven bound, however, is given by the next theorem.

Theorem 1 (Pin, Frankl [11], [17]). If an automaton is synchronizing then it has a synchronizing word of length at most \( \frac{n^3 - n}{6} \).

For the next theorem, we need to define a particular graph called the graph of pairs. For an automaton, the graph of pairs represents the image of each pair of states by the transition function. An example is given on Figure 2. The formal definition, which remains valid for graphs of pairs of sets of nonnegative matrices with no zero row, is the following.

Definition 1 (Graph of pairs). We call the graph of pairs \( F(S) \) the graph defined as follows.
The system will get closer to each other. When this happens sufficiently, a positive column corresponds to the primitivity of a set that has a positive column which means that there is a path to a node representing a single state.

**Theorem 2** (Eppstein [9, Theorem 4]). $S$ is synchronizing if and only if, in its graph of pairs (as defined above), from each node representing a pair, there is a path to a node representing a single state.

**Theorem 3** (Eppstein [9, Theorem 8]). The problem of deciding whether a given automaton has a synchronizing word of length at most $l$ is NP-hard.

**III. From Consensus to Column-Primitivity**

We prove the equivalence between the existence of a sequence of transition matrices that drives the system to consensus and column-primitivity. Recall that column-primitivity of a set $S$ is the existence of a column-positive product, that is, a product of matrices from $S$ that has a positive column. Intuitively, a positive column corresponds to an agent influencing all agents. In particular agents with the greatest and smallest values are influenced by this agent and will get closer to each other. When this happens sufficiently often, the system converges to consensus.

We note $N = \{1, \ldots, n\}$, $M = \{1, \ldots, m\}$ and $A_w$ as a shortcut for $A_{w_1} \dot{\ldots} A_{w_l}$.

**Proposition 1.** For a set $S = \{A_1, \ldots, A_m\}$ of stochastic matrices, the answer to Problem 1 is positive if and only if $S$ is column-primitive.

**Proof.** If $S$ is column-primitive, then there are $j^* \in N$ and a word $w = w_1 \ldots w_l$ such that the $j^*$th column of $A_w$ is positive. Defining $a = \min_i \{(A_w)_{ij}\}$, we have

$$\forall x \in \mathbb{R}^n, \max_i \{(A_w)_{ix}\} \leq (1-a) \max_i \{x_i\} + ax_j,$$

$$\forall x \in \mathbb{R}^n, \min_i \{(A_w)_{ix}\} \geq (1-a) \min_i \{x_i\} + ax_j,$$

from which follows:

$$\forall x \in \mathbb{R}^n, \max_i \{(A_w)_{ix}\} - \min_i \{(A_w)_{ix}\} \leq (1-a)(\max_i \{x_i\} - \min_i \{x_i\}).$$

This means that, for the system

$$x(t+1) = A_{w(t) \mod l} x(t),$$

we can conclude convergence of System (2).

**Only if:** In [7], Proposition 1.b, we proved that if the answer to Problem 1 is true, then there is a product $A_w = A_{w_1} \dot{\ldots} A_{w_l}$ such that

$$x(t+1) = A_w x(t)$$

converges to consensus for any initial condition $x(0) = x_0$. This means that $\lim_{t \to \infty} A_w^t = y^T$ for some $y \neq 0$. Therefore $\lim_{t \to \infty} A_w^t$ has a positive column and there is $t^*$ such that $A_w^{t^*}$ has a positive column which means that $w$ concatenated $t^*$ times is a column-positive word.

**Example 1.** The set

$$S = \{A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.8 & 0.2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \}$$

is column-primitive: the product

$$A_w = A_1 A_2 A_2 A_1 = \begin{pmatrix} 0 & 0.2 & 0 & 0.8 \\ 0 & 0.36 & 0 & 0.64 \\ 0 & 1 & 0 & 0 \\ 0 & 0.8 & 0 & 2 \end{pmatrix}$$

is column-positive. This implies the convergence of System (2) for sequence $\sigma = \ldots 112211221$ and any initial condition. Indeed, one can check that

$$\lim_{t \to \infty} x(t) = \lim_{s \to \infty} A_w^s x_0 = Iv^T x_0$$

where $v^T = (0, 0.565 \ldots, 0.072 \ldots, 0.361 \ldots)$.

**Proposition 2** (Equivalence between existence of a convergent trajectory and convergence with random switching). **Problem 1 is equivalent to the problem:** "does System (2) converge with probability one when at each step, each transition matrix $A_k$ from set $S$ is chosen with i.i.d. nonzero probability?".

**Proof.** Only if is evident.

If: If the answer to Problem 1 is positive, there is a column-positive product (Proposition 1). Under independent random switching this product appears infinitely often. $\max_i x_i(t) - \min_i x_i(t)$ decreases by a factor $(1 - \min_i \{(A_w)_{ij}\})$ each time $A_w$ appears and does not increase in between. From there, we can prove convergence as in the proof of Proposition 1. \qed
IV. AUTOMATA BASED CRITERIA

In this section, we extend Theorems 1 and 2 to finite sets of nonnegative matrices with no zero row. Moreover, we prove that a proof of the Černý Conjecture would also immediately extend to these sets. We define \( R_n \), the set of \( n \times n \) matrices with nonnegative entries and no zero row. We proceed by associating an automaton \( S' \) to any finite set \( S \subset R_n \) and by proving that any synchronizing word of \( S' \) can be transformed into a column-positive word of \( S \) and vice-versa.

**Definition 2** (Graph associated with a word). Given a set \( S = \{A_1, \ldots, A_m\} \subset R_n \) and a word \( w = w_1 \ldots w_l \), we call graph associated with the word \( w \) the graph whose adjacency matrix is

\[
\begin{pmatrix}
0 & A_{w_1} & \cdots & \cdots & \cdots & A_{w_l} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\vdots & \cdots & \ddots & \ddots & \ddots & \ddots \\
\vdots & \cdots & \cdots & \ddots & \ddots & \ddots \\
\vdots & \cdots & \cdots & \cdots & \ddots & \ddots \\
0 & \cdots & \cdots & \cdots & \cdots & 0
\end{pmatrix}
\]

**Example 1 (continued).** For set \( S \) of Example 2, the graph associated with the word 11221 is depicted on Figure 3.

![Graph associated with the word 11221](image)

The edges corresponding to matrix \( A_1 \) are in black, those corresponding to matrix \( A_2 \) are in grey.

**Definition 3** (Sign pattern domination). We will write \( A \succeq B \) and say that matrix \( A \) dominates matrix \( B \) if

\[
A_{ij} = 0 \Rightarrow B_{ij} = 0.
\]

The next definition and the proof technique of Lemma 1 are inspired by a similar construction in [4, Theorem 17].

**Definition 4** (Automaton associated with a set). Let \( S = \{A_1, \ldots, A_m\} \subset R_n \). We call the automaton associated with the set \( S \) the automaton \( S' \) containing all binary stochastic matrices that are dominated by some matrix of \( S \), that is

\[
S' = \{ A' | A' \in B_n \text{ and } \exists A \in S \text{ s.t. } A \succeq A' \},
\]

with \( B_n = \{ A | A \in \{0,1\}^{n \times n}, AI = I \} \) the set of binary stochastic matrices.

Note that the associated automaton can contain a large number of matrices but we will only use it in proofs and we will not construct it explicitly in any algorithm.

**Example 1 (continued).** The automaton associated with the set \( S \) of Example 2 is

\[
S' = \{ A'_1, A'_2, A'_3 \}
\]

with

\[
A'_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},
A'_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},
A'_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.
\]

**Definition 5** (In-tree and spanning in-tree). We call an in-tree a connected graph in which, for a vertex \( r \) called the root and any other vertex \( v \), there is exactly one directed path from \( r \) to \( v \). For a graph \( G \), we call a spanning in-tree an in-tree that has the same set of nodes as \( G \) and whose set of edges is a subset of that of \( G \).

**Lemma 1.** Let \( S = \{A_1, \ldots, A_m\} \subset R_n \) be a column-primitive set, \( S' = \{A'_1, \ldots, A'_m\} \) its associated automaton and \( w = w_1 \ldots w_l \) be a word on alphabet \( M \). The word \( w \) is column-positive for \( S \) if and only if there is a word \( w' = w'_1 \ldots w'_l \) on alphabet \( \{1, \ldots, m'\} \) that is synchronizing for \( S' \) and such that

\[
\forall i \in \{1, \ldots, l\}, A_{w_l} \succeq A'_{w'_l}.
\]

**Proof.** If: The product \( A_{w} = A_{w_l} \ldots A_{w_1} \) dominates \( A'_{w'_l} \ldots A'_{w'_1} \) because each \( A_{w_l} \) dominates \( A'_{w'_l} \) and domination is preserved under multiplication. In particular, if \( A'_{w'_l} \) has a positive column, the same column is positive in \( A_{w_l} \).

Only if: Suppose that \( w \) is column-positive and let us call \( j^* \) the index of the positive column. This means that, in the graph \( G(w) \) associated with the word \( w \), from each node \((i,0)\) there exists a path to node \((j^*, m)\). The graph of these paths has a spanning in-tree rooted in \( j^* \). In \( G(w) \), for each node, there is at most one outgoing edge that belongs to the spanning in-tree. Therefore, some edges of \( G(w) \) can be removed such that the graph still has the same spanning in-tree and each node has exactly one outgoing edge. We perform the corresponding operations the matrices forming product \( A_{w_l} \ldots A_{w_1} \), that is, we set to zero positive elements that do not correspond to edges of the spanning in-tree and such that on each row of each matrix, exactly one element remains positive. Then we set to one all remaining positive elements. We obtain a new product \( A'_{w'_l} \ldots A'_{w'_1} \) for which

- the \( j^* \) column is positive
- \( \forall i, A_{w_l} \succeq A'_{w'_l} \)
- \( \forall i, A'_{w'_l} \) is binary and stochastic,

from which we conclude that each \( A'_{w'_l} \) belongs to \( S' \) the automaton associated with the set \( S \) and that \( w' \) is synchronizing for \( S' \). \( \square \)

**Example 1 (continued).** The graph associated with the word 11221 is represented on Figure 4. The in-tree is in black. We see that removing the dashed edges allows keeping the in-tree and having exactly one outgoing edge from each node. Without these dashed edges, the graph...
becomes that associated with the word 11223 of automaton $S' = \{A_1', A_2', A_3'\}$.

Lemma 1 allows extending Theorems 1 and 2 to finite subsets of $\mathcal{R}_n$.

**Theorem 4.** Let $S \in \mathcal{R}_n$ be a finite set. If $S$ is column-primitive then it has a column-positive word of length at most $n^3/n$.

**Proof.** Suppose that $S$ is column-primitive. By Lemma 1 its associated automaton $S'$ is synchronizing. Then by Theorem 1 $S'$ has a synchronizing word of length at most $n^3/n$. Finally, we reapply Lemma 1 to conclude that $S$ has a column-positive word of length at most $n^3/n$.

**Theorem 5.** If Conjecture 7 holds, then any column-primitive set has a column-positive word of length at most $(n-1)^2$.

**Proof.** The proof is similar to that of Theorem 4.

**Theorem 6.** A finite set $S \in \mathcal{R}_n$ is column-primitive if and only if, in its graph of pairs, from each node representing a pair there is a path to a node representing a single state.

**Proof.** The graph of pairs of set $S$ and that of its associated automaton $S'$ are the same. Lemma 1 then allows concluding.

The graph of pairs can be constructed in $O(mn^4)$ operations: for each pair of nodes $(i_1, i_2), (j_1, j_2)$, we add an edge if there is $k \in M$ such that $(A_k)_{i_1j_1}, (A_k)_{i_2j_2} > 0$. The reachability condition can be checked in $O(|V| + |E|) = O(n^4)$ operations with a search algorithm (for example a depth-first search algorithm).

**Example 1 (continued).** On the graph of pairs of the set $S$ as defined in Example 1 one can see that from each pair there is a path to a single state (Figure 5). Therefore, $S$ is column-primitive.

**V. SHORTEST COLUMN-POSITIVE WORD: NP-HARDNESS**

Deciding whether an automaton has a synchronizing word of length at most $l$ is an NP-hard problem (see Theorem 3). In fact, even approximating the length of the shortest synchronizing word with a small but fixed accuracy is NP-hard, as proven by Berlinkov [3].
(A_{w_k} A_{w_k})_{11} = \sum_j (A_{w_k})_{ij} (A_{w_k})_{j1} = (A_{w_k})_{11} = 0.

Iterating this reasoning proves the claim.

We now prove that if the F is satisfiable if and only if S has a column-positive product of length smaller or equal to v.

**Only if**: For a satisfiable assignment, let us define $P = A_{w_1} \ldots A_{w_t}$ where $A_{w_i} = A_{X_i}$ if $X_i$ is assigned to TRUE and $A_{w_i} = A_{X_i}$ if $X_i$ is assigned to FALSE. The length of this product is clearly v. We prove that its first column is positive. The element $(1, 1)$ is positive in all matrices $A_{w_k}$, therefore, it is positive in P. The element $(1 + i, 1)$ is positive in matrix $A_{w_i}$ because this matrix is equal to either $A_{X_i}$ or $A_{-X_i}$. By Claim 1, we have $P_{1+i,1} > 0$. Finally, the element $(1 + v + i, 1)$ is positive in one of the matrices $A_{w_k}$ because one of these matrices correspond to assigning the variable $X_k$ to satisfy clause $C_i$. By Claim 1, we have $P_{1+v+i,1} > 0$. and we conclude that the first column of P is positive.

**If**: The first row of any product of matrices from S is $(1 \ldots 1 \ldots 0)$. Therefore, if a product is column-positive its first column is positive.

We prove that if there is column-positive product $P = A_{w_1} \ldots A_{w_t}$ of length at most v, then for each variable $X_i$ then either $A_{X_i}$ or $A_{-X_i}$ appears in the product. Indeed, if it is not the case, Claim 1 says that $P_{1+i,1} = 0$ and P is not column-positive.

We have also that not both $A_{X_i}$ and $A_{-X_i}$ appear in the product because it would imply either that P is longer than v contradicting our assumption or that for some variable $X_j$ neither $A_{X_j}$ nor $A_{-X_j}$ appears in the product, contradicting what we just proved.

By Claim 1 $P_{1+v+i,1} > 0$ implies that for some k, $(A_{w_k})_{11} > 0$ meaning that assigning $X_k$ to TRUE if $w_k = X_k$ and to FALSE if $w_k = X_k$ satisfies clause $C_i$. Since $P_{1+v+i,1} > 0$ holds for all the clauses i, that means that this assignment is satisfiable.

\[ A_{X_1} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & I_{w \times v} & 0 \\ 0 & 0 & I_{v \times v} \\ 0 & 1 & I_{c \times c} \\ 0 & 0 & I_{w+c \times v+c} \end{pmatrix} \]

The second 1 represents the assignment of variable $X_1$; the third one represents satisfaction of the second clause. Similarly, the other matrices are defined as

with

$$v_{-X_1} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}^\top,$$

$$v_{X_2} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}^\top,$$

$$v_{-X_2} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \end{pmatrix}^\top,$$

$$v_{X_3} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}^\top,$$

$$v_{-X_3} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}^\top.$$

Assignment $\neg X_1, X_2, \neg X_3$ corresponds to product $A_{-X_1} A_{X_2} A_{-X_3} = A_w$

with

$$w = \begin{pmatrix} 1 & 1 & 1 & 2 & 1 & 3 \end{pmatrix}^\top.$$

The first column is positive because the assignment is satisfiable.

**REFERENCES**

[1] R. L. Adler and B. Weiss. Similarity of automorphisms of the torus. *Memoirs of the American Mathematical Society*, 98, 1970.

[2] V. Rényi. On a certain graph-theoretical problem arising from automata theory. *Acta Mathematica Academiae Scientiarum Hungaricae*, 10(3--4):447–457, 1959.

[3] J. Berger. The complexity of the decision problem for automorphisms. *Theoretical Computer Science*, 5(1):73–82, 1977.

[4] D. Eppstein. Reset sequences for monotonic automata. *SIAM Journal on Computing*, 28(6):1931–1936, 1999.

[5] A. Chapman and M. Mesbahi. On symmetry and controllability of multi-agent systems. In *Proceedings of the 15th IEEE Conference on Decision and Control*, 2004.

[6] J. Černý. Poznámka k homogénnym eksperimentom s konečnými automatami. *Matematicko-fyzikálny Casopis SAV*, 14:208–216, 1964.

[7] M. V. Berlinkov. Approximating the minimum length of synchronizing words is hard. *Lecture Notes in Computer Science*, 6072:37–47, 2010.

[8] V. D. Blondel, R. M. Jungers, and A. Olshevsky. On primitivity of sets of matrices. *Submitted*, 2014.

[9] F. Benes, V. D. Blondel, P. Thiran, and M. Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *IEEE International Symposium on Information Theory Proceedings*, pages 1752–1757, 2010.

[10] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of networks. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, 2014.

[11] P.-Y. Chevalier, J. M. Hendrickx, and R. M. Jungers. Efficient algorithms for the consensus decision problem. *Submitted*, 2014.

[12] François Gonze and R. M. Jungers. On the synchronizing probability function and the triple rendezvous time for synchronizing automata. *Submitted*, 2015.

[13] D. Eppstein. Reset sequences for monotonic automata. *SIAM Journal on Computing*, 19(3):500–510, 1990.

[14] F. M. Fominykh, P. V. Martyugin, and M. V. Volkov. P1-ways for synchronization. *International Journal of Foundations of Computer Science*, 24(6):765–780, 2013.

[15] P. Frankl. An extremal problem for two families of sets. *European Journal of Combinatorics*, 3:125–127, 1982.

[16] J. C. Gower and P. Carley. Stability and stabilization of discrete time switched systems. *International Journal of Control*, 79(7):719–728, 2006.

[17] R. M. Jungers. The joint spectral radius, theory and applications. In *Lecture Notes in Control and Information Sciences*, volume 385. Springer-Verlag, Berlin, 2009.

[18] R. M. Jungers. The synchronizing probability function of an automaton. *SIAM Journal on Discrete Mathematics*, 26(1):177–192, 2012.

[19] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 482–491, 2003.

[20] M. S. Paterson. Unsolvability in 3 × 3 matrices. *Studies in Applied Mathematics*, 49:105–107, 1970.

[21] J.-É. Pin. On two combinatorial problems arising from automata theory. *Annals of Discrete Mathematics*, 17:535–548, 1983.

[22] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *International Journal of Control*, 82(16):2083–2090, 2009.

[23] D. Stanford and J. M. Urbano. Some convergence properties of matrix sets. *SIAM Journal on Matrix Analysis and Applications*, 15(4):1332–1340, 1994.

[24] A. N. Trahtman. The road coloring problem. *Israel Journal of Mathematics*, 172(2):51–60, 2009.
[21] M. V. Volkov. Synchronizing automata and the Černý conjecture. In *LATA’08: Proceedings of the 2nd International Conference on Language and Automata Theory and Applications*, pages 11–27. Springer-Verlag, 2008.