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 multi-agent consensus system to consensus. We transform this problem into the problem of the existence of a product of the (stochastic) transition matrices that has a positive column. This allows us to make use of results from automata theory to sets of stochastic matrices. Our main result is 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 the weighted average of values of other agents:

$$x(t+1) = A_t x(t),$$

with A_t *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], [19].

We consider a different kind of controllability: the system is not controlled by an exogenous input, but by choosing the matrix of interaction A_t 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. The problem consists here in finding a scheduling such that the agents converge to consensus.

Formally, we study the system

$$\begin{aligned} x(t+1) &= A_{\sigma(t)}x(t) \\ x(0) &= x_0, \end{aligned} \tag{1}$$

and we want to solve the following decision problem.

Problem 1. *Given a set of stochastic matrices $M = \{A_1, \ldots, A_m\}$, does there exist a switching sequence $\sigma : \mathbb{N} \mapsto \{1, \ldots, m\} : t \mapsto \sigma(t)$ such that, for any x_0, System (1) converges to consensus, i.e., to some vector $a\mathbf{1} = a \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^\top$ for some a?*[1]

This problem can be seen as an *open loop* control problem. Indeed, it deals with the existence of a sequence σ that steers System (1) to consensus *from any initial condition*. One could have σ 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], [20] and [13, Section 2.2.4]) and is known to be very hard. For instance, deciding, for a matrix set M, whether there is a product of matrices from M that converges to zero is an undecidable problem [17].

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. First, we reduce the problem to that of determining the existence of a *positive-column product*, i.e., a product of transition matrices that has a positive column. We call a *positive-column word* the sequence of indices of a positive-column product. It turns out that the existence of positive-column 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 positive-column word are called *synchronizing semiautomata*. 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. A semi-automaton is a triple (S, Σ, M) where S is a finite set of *states*, Σ is a finite set of *letters* and is called the *alphabet*, and M is a finite set of mappings from S to S. An automaton is a semi-automaton with a set of ACCEPT states. These ACCEPT states determine which input words are accepted by the automaton. Semi-automata can be represented as sets of digraphs where each node has outdegree one. An example is given in Figure 1. They can also be represented by a set of matrices, containing the

---

[1] This problem is Problem 2 of our article [7], that we restrict here to the case of stochastic matrices. We obtain here much stronger results with a different approach.
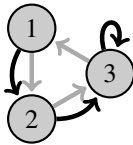
Fig. 1. A simple semi-automaton with three states and two actions (black and grey edges).

adjacency matrices of these digraphs. Since the transition function is deterministic, each node of each digraph has outdegree one, and the adjacency matrices of the digraphs have a single one on each row and zero everywhere else. Hence, they are binary and stochastic. The set corresponding to the semi-automaton of Figure 1 is

$$M = \left\{ A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \right\}.$$

Automata and semi-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. A semi-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*. Automata and synchronizing automata have generated intense research efforts in theoretical computer science [1], [8], [16], [18], [21], [22], 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[2]. Therefore, our investigation of Problem 1 yields 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,

---

[2]In fact, if the choices of matrices are not independent, the equivalence still holds provided that each matrix is chosen with nonzero *conditional* probability.

our results apply to any process with column-stochastic transition matrices, i.e., matrices whose transposes are stochastic[3]. 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 $x(t) = s(t)/w(t)$. 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 M, i.e., the existence of a positive-column 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). We obtain

- a polynomial bound ($O(n^3)$) on the length of positive-column words (Theorem 4),
- a polynomial-time ($O(n^4)$) algorithm to decide the existence of a positive-column word (Theorem 5),
- a proof of NP-hardness of finding the shortest positive-column word, which also holds for sets of matrices with positive diagonals (Theorem 6).

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 positive-column product.

Conjecture 1 (Černý Conjecture [5]). *If a semi-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], [18]). *If a semi-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 the digraph of pairs. For a semi-automaton, the digraph of pairs represents the image of each pair of states by the transition function. An example is given in Figure 2. The formal definition, which

---

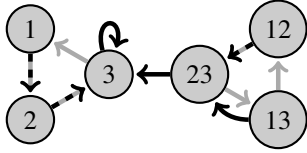[3]Stochastic matrices are sometimes referred to as *row-stochastic*.

Fig. 2. Digraph of pairs for semi-automaton of Figure 1.

remains valid for digraphs of pairs of sets of nonnegative matrices with no zero row, is the following.

Definition 1 (Digraph of pairs). *We call the digraph of pairs $\mathcal{F}(M)$ the digraph defined as follows.*
- *One node for each unordered pair of states with repetition ($\frac{n(n+1)}{2}$ nodes),*
- *an edge from pair i_1, i_2 to pair j_1, j_2 if there is a matrix $A_k \in M$ such that $(A_k)_{i_1 j_1}, (A_k)_{i_2 j_2} > 0$ (or $(A_k)_{i_1 j_2}, (A_k)_{i_2 j_1} > 0$), where i_1, i_2, j_1, j_2 are not necessarily different.*

Theorem 2 (Eppstein [9, Theorem 4]). *M is synchronizing if and only if, in its digraph 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]). *Deciding whether a given semi-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 M is the existence of a positive-column product, that is, a product of matrices from M 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.

Without loss of generality, we assume that the set S of states is equal to $\{1, \ldots, n\}$ and the alphabet Σ equal to $\{1, \ldots, m\}$. We note A_w as an abbreviation for $A_{w_l} \ldots A_{w_1}$.

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

Proof. *If:* Let A_w have a positive j^{th} column. Defining $a = \min_i\{(A_w)_{ij}\}$, and using the stochasticity of A_w, we have

$$\forall x \in \mathbb{R}^n, i \in S, \ (A_w x)_i = \sum_{k \in N \setminus \{j\}} (A_w)_{ik} x_k + (A_w)_{ij} x_j$$
$$\leq \max_h \{x_h\}(1-a) + a x_j,$$

and

$$\forall x \in \mathbb{R}^n, \ \max_i\{(A_w x)_i\} \leq (1-a)\max_h\{x_h\} + a x_j$$
$$\forall x \in \mathbb{R}^n, \ \min_i\{(A_w x)_i\} \geq (1-a)\min_h\{x_h\} + a x_j,$$

from which follows:

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

Hence, for the system

$$x(t+1) = A_{w_{t \bmod l+1}} x(t), \quad (2)$$

$\lim_{s \to \infty} \max_i\{x_i(sl)\} - \min_i\{x_i(sl)\} = 0$. Finally, since

$$\forall x \in \mathbb{R}^n, \forall k \in \Sigma, \ \max_i\{(A_k x)_i\} \leq \max_i\{x_i\}$$
$$\forall x \in \mathbb{R}^n, \forall k \in \Sigma, \ \min_i\{(A_k x)_i\} \geq \min_i\{x_i\},$$

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 $A_w = A_{w_l} \ldots A_{w_1}$ such that

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

converges to consensus for any initial condition $x(0) = x_0$. This means that $\lim_{t \to \infty} A_w^t = \mathbf{1} y^\top$ for some $y^\top \mathbf{1} = 1$. 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 positive-column word. $\square$

Example 1. *The set*

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

is column-primitive: the product

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

is positive-column. This implies the convergence of System (1) for the sequence $\sigma = \ldots 1122111221$ and any initial condition. Indeed, one can check that

$$\lim_{t \to \infty} x(t) = \lim_{s \to \infty} A_w^s x_0 = \mathbf{1} v^\top x_0$$

where $v^\top = \begin{pmatrix} 0 & 0.565\ldots & 0.072\ldots & 0.361\ldots \end{pmatrix}$.

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

Proof. *"Only if"* is evident.

If: If the answer to Problem 1 is positive, then there is a positive-column product (Proposition 1). Under independent random switching this product appears infinitely often with probability one. $\max_i x_i(t) - \min_i x_i(t)$ 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. $\square$

IV. CRITERIA BASED ON AUTOMATA

In this section, we extend Theorems 1 and 2 to finite sets of nonnegative matrices with no zero row, corresponding to digraphs with outdegree at least one. Moreover, we prove that a proof of the Černý Conjecture would also immediately extend to these sets.

We define $\mathcal{R}_n$, the set of $n \times n$ matrices with nonnegative entries and no zero row. We proceed by associating a semi-automaton M' to any finite set $M \subset \mathcal{R}_n$ and by proving that any synchronizing word of M' can be transformed into a positive-column word of M and vice-versa.

Definition 2 (Digraph associated with a word). *Given a set $M = \{A_1, \ldots, A_m\} \subset \mathcal{R}_n$ and a word $w = w_l \ldots w_1$, we call* digraph associated with the word w *the digraph whose adjacency matrix is*

$$\begin{pmatrix} 0 & A_{w_l} & & & \\ & \ddots & \ddots & & \\ & & \ddots & A_{w_2} & \\ & & & \ddots & A_{w_1} \\ & & & & 0 \end{pmatrix}.$$

Example 1 (continued). *For set M of Example 1, the digraph associated with the word 11221 is depicted in Figure 3.*
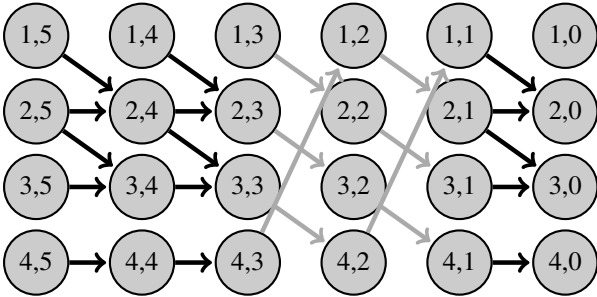

Fig. 3. Digraph associated with the word 11221 for M defined in Example 1. The edges corresponding to matrix A_1 are in black, those corresponding to matrix A_2 are in grey.

Definition 3 (Zero pattern domination). *We write $A \succeq B$ and say that matrix A dominates matrix B if the digraph corresponding to A has more edges than that corresponding to B, or formally*

$$A_{ij} = 0 \Rightarrow B_{ij} = 0. \tag{3}$$

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

Definition 4 (Semi-automaton associated with a set). *Let $S = \{A_1, \ldots, A_m\} \subset \mathcal{R}_n$. We call the* semi-automaton *associated with the set M, the semi-automaton M' containing all binary stochastic matrices that are dominated by some matrix of M, that is*

$$M' \triangleq \{A' \mid A' \in \mathcal{B}_n \text{ and } \exists A \in M \text{ s. t. } A \succeq A'\},$$

with $\mathcal{B}_n \triangleq \{A \mid A \in \{0,1\}^{n \times n}, A\mathbf{1} = \mathbf{1}\}$ *the set of binary stochastic matrices.*

Note that the associated semi-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 semi-automaton associated with the set M of Example 1 is $M' = \{A'_1, A'_2, A'_3\}$, with*

$$A'_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ A'_2 = A_2, \ A'_3 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Definition 5 (In-tree and spanning in-tree). *We call an in-tree a digraph 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 digraph $\mathcal{G}$, we call a* spanning in-tree *an in-tree that has the same set of nodes as $\mathcal{G}$ and whose set of edges is a subset of that of $\mathcal{G}$.*

Lemma 1. *Let $M = \{A_1, \ldots, A_m\} \subset \mathcal{R}_n$, $M' = \{A'_1, \ldots, A'_{m'}\}$ its associated semi-automaton and $w = w_l \ldots w_1$ be a word over the alphabet Σ. The word w is positive-column for M if and only if there is a word $w' = w'_l \ldots w'_1$ over the alphabet $\{1, \ldots, m'\}$ that is synchronizing for M' and such that*

$$\forall i \in \{1, \ldots, l\}, \ A_{w_i} \succeq A'_{w'_i}.$$

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

Only if: We call the node (i, j) with $i \in \{1, \ldots, n\}$, $j \in \{0, \ldots, l\}$ the node corresponding to the ith row of the $(l - j)^{\text{th}}$ block-row of the matrix of Definition 2. This numbering is represented in Figure 3. Suppose that A_w has a positive k^{th} column. Therefore, in the digraph $\mathcal{G}(w)$ associated with the word w, from each node (i, l), there exists a path to node $(k, 0)$. The digraph of these paths has a spanning in-tree rooted in k. In $\mathcal{G}(w)$, for each node, there is at most one outgoing edge that belongs to the spanning in-tree. Therefore, some edges of $\mathcal{G}(w)$ can be removed such that the digraph still has the same spanning in-tree and each node has exactly one outgoing edge. We perform the corresponding operations on the matrices that form the 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 1 all remaining positive elements. We obtain a new product $A'_{w'_l} \ldots A'_{w'_1}$ for which

- the kth column is positive
- $\forall i, A_{w_i} \succeq A'_{w'_i}$
- $\forall i, A'_{w'_i}$ is binary and stochastic by construction,

from which we conclude that each $A'_{w'_i}$ belongs to M' the semi-automaton associated with the set M and that w' is synchronizing for M'. $\square$

Example 1 (continued). *The digraph associated with the word 11221 is represented in 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 digraph becomes that*
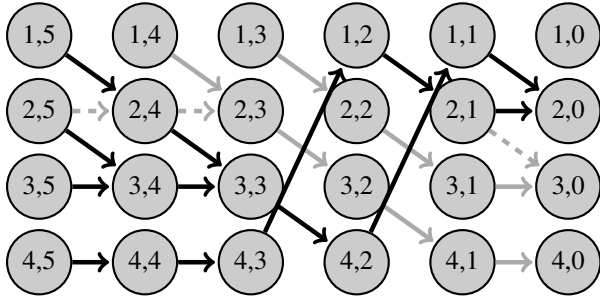

Fig. 4. Digraph associated with the word 11221. In black: the in-tree. Without the dashed edges, the digraph is that of the word 11223 of semi-automaton M'.

associated with the word 11223 of semi-automaton $M = \{A_1', A_2', A_3'\}$.

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

Theorem 4. *Let $M \subset \mathcal{R}_n$ be a finite set. If M is column-primitive then it has a positive-column word of length at most $\frac{n^3-n}{6}$ and at most $(n-1)^2$ if Conjecture 1 holds.*

Proof. Suppose that M is column-primitive. By Lemma 1, its associated semi-automaton M' is synchronizing. Then by Theorem 1, M' has a synchronizing word of length at most $\frac{n^3-n}{6}$. Finally, we reapply Lemma 1 to conclude that M has a positive-column word of length at most $\frac{n^3-n}{6}$. ☐

Theorem 5. *A finite set $M \subset \mathcal{R}_n$ is column-primitive if and only if, in its digraph of pairs, from each node representing a pair there is a path to a node representing a single state.*

Proof. The digraph of pairs of set M and that of its associated semi-automaton M' are the same. Lemma 1 concludes. ☐

The digraph 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 $k \in \Sigma$ such that $(A_k)_{i_1 j_1}, (A_k)_{i_2 j_2} > 0$. The reachability can be checked in $O(|V| + |E|) = O(n^4)$ operations (for example with a depth-first search algorithm).

Example 1 (continued). *On the digraph of pairs of the set M as defined in Example 1, from each pair, there is a path to a single state (Figure 5). Hence, M is column-primitive.*

V. SHORTEST POSITIVE-COLUMN WORD: NP-HARDNESS

Deciding whether a semi-automaton has a synchronizing word of length at most l is NP-hard (see Theorem 3). In fact, even approximating the length of the shortest synchronizing word with a fixed accuracy is NP-hard [3].
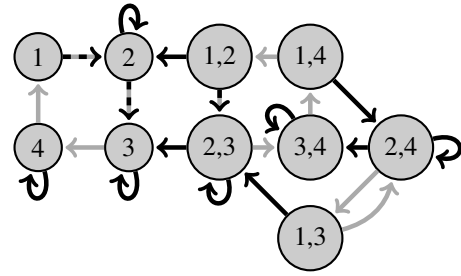

Fig. 5. Digraph of pairs of M (Example 1). The black edges represent matrix A_1, the grey ones matrix A_2.

These results extend directly to the column-primitivity problem for finite subsets of $\mathcal{R}_n$, because a semi-automaton is a particular subset of $\mathcal{R}_n$. However, we show here that the problem remains NP-hard *even if the matrices have positive diagonal elements*. Restricting a problem can make it easier and in particular, for sets with positive diagonals, checking column-primitivity is easier. M is column-primitive if and only if its digraph $\mathcal{G}(M)$, defined as the union of the digraphs $\mathcal{G}(A_k)$ of which A_k are the adjacency matrices, has a spanning in-tree. In that case, there is a positive-column word of length at most $n-1$ (given by the spanning in-tree).

Problem 2. *Given a set $M \subset \mathcal{R}_n$ of matrices with positive diagonals, does there exist a positive-column word of length smaller or equal to l?*

Theorem 6. *Problem 2 is NP-hard.*

Proof. We proceed by reduction from 3-SAT. A 3-SAT instance consists of variables $X_1, \ldots, X_v$, clauses $C_1, \ldots, C_c$ of the form $L_1 \vee L_2 \vee L_3$ where each L_i is a literal, that is, either a variable or the negation of a variable. The problem is to determine whether the formula $C_1 \wedge \cdots \wedge C_c$ is satisfiable.

Given a 3-SAT formula F with v variables, we construct a set of matrices that has a positive-column word of length smaller or equal to v (in fact never strictly smaller) if and only if the 3-SAT problem is satisfiable.

We define a set M of $2v$ matrices. The matrices have size $(1+c+v) \times (1+c+v)$. The matrix A_{X_i}, representing literal X_i, has ones on the diagonal, a 1 in position $(1+i, 1)$ and a 1 in position $(1+v+j, 1)$ for every j for which assigning X_i to TRUE satisfies clause C_j. The matrix $A_{\neg X_i}$ has ones on the diagonal, a 1 in position $(1+i, 1)$ and a 1 in position $(1+v+j, 1)$ if assigning X_i to FALSE satisfies clause C_j.

Let $P = A_{w_l} \ldots A_{w_1}$ be a product of matrices from M. From the structure of the matrices, one can verify :

Claim 1: P has a positive $(i, 1)$ element if and only if at least one of the matrices A_{w_k} has a positive $(i, 1)$ element.

We prove that the formula F is satisfiable if and only if M has a positive-column product of length at most v.

If: Let $P = A_{w_l} \ldots A_{w_1}$ be a positive-column product of length $l \leq v$. The first row of P is equal to $\begin{pmatrix} 1 & 0 & \ldots & 0 \end{pmatrix}$, and therefore, the column that is positive is the first one. For each variable X_i then either A_{X_i} or $A_{\neg X_i}$ appears in the product. Indeed, if it is not the case, Claim 1 says that

$P_{1+i,1} = 0$ and P is not positive-column. Also, at most one among A_{X_i} and $A_{\neg X_i}$ appear in the product. Indeed, the presence of both of them would imply that P is longer than v, contradicting our assumption.

By Claim 1, $P_{1+v+i,1} > 0$ implies that for some k, $(A_{w_k})_{i1} > 0$ meaning that assigning X_k to TRUE if $w_k = X_k$ and to FALSE if $w_k = \neg X_k$ satisfies clause C_i. Since $P_{1+v+i,1} > 0$ holds for any clause i, this assignment is satisfiable.

Only if: For a satisfiable assignment, let us define $P = A_{w_v} \ldots A_{w_1}$ where $A_{w_i} = A_{X_i}$ if X_i is assigned to TRUE and $A_{w_i} = A_{\neg X_i}$ 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 $A_{\neg X_i}$. 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 conclude that the first column of P is positive. $\qquad\square$

Example 2. *Consider the formula*

$$(\neg X_1 \vee \neg X_2 \vee \neg X_3) \wedge (X_1 \vee X_2 \vee X_3) \wedge (\neg X_1 \vee X_2 \vee \neg X_3).$$

In the reduction of Theorem 6, we construct the set

$$M = \{A_{X_1}, A_{\neg X_1}, A_{X_2}, A_{\neg X_2}, A_{X_3}, A_{\neg X_3}\}$$

with

$$A_w = \left(\begin{array}{c|c} 1 & 0 \\ \hline v_w & I_{v+c \times v+c} \end{array} \right)$$

$$v_{X_1} = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 \end{array} \right)^\top.$$

The first 1 represents the assignment of variable X_1; the second 1 represents satisfaction of the second clause. Similarly, the other matrices are defined with

$$v_{\neg X_1} = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 1 \end{array} \right)^\top$$
$$v_{X_2} = \left(\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 1 & 1 \end{array} \right)^\top$$
$$v_{\neg X_2} = \left(\begin{array}{ccc|ccc} 0 & 1 & 0 & 1 & 0 & 0 \end{array} \right)^\top$$
$$v_{X_3} = \left(\begin{array}{ccc|ccc} 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right)^\top$$
$$v_{\neg X_3} = \left(\begin{array}{ccc|ccc} 0 & 0 & 1 & 1 & 0 & 1 \end{array} \right)^\top.$$

Assignment $\neg X_1, X_2, \neg X_3$ corresponds to the product

$$A_{\neg X_1} A_{X_2} A_{\neg X_3} = A_w$$

with $v_w = \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 2 & 1 & 3 \end{array} \right)^\top$. The first column is positive because the assignment is satisfiable.

VI. Concluding Remarks

We have provided an algorithm to determine the existence of a switching sequence that makes a consensus system converge. Our result directly applies when the switching can be controlled, but is also relevant in case of random switching, as we have shown the equivalence between these two questions. Our algorithm has a better complexity than previous approaches [7] but we do not know if this complexity is optimal.

We have also studied the existence of switching sequences that make the system converge, and that are periodic with a small period. One the one hand, we have shown that finding the sequence that makes the system converge with the smallest period is NP-hard. On the other hand, we have shown the existence of a sequence that makes the system converge and that has a small period.

References

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

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

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

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

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

[6] A. Chapman and M. Mesbahi. On symmetry and controllability of multi-agent systems. In *Proceedings of the 53th IEEE Conference on Decision and Control*, 2014.

[7] P.-Y. Chevalier, J. M. Hendrickx, and R. M. Jungers. Efficient algorithms for the consensus decision problem. *To appear in SIAM Journal on Control and Optimization*, 2014.

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

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

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

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

[12] J.C. Geromel and P. Colaneri. Stability and stabilization of discrete time switched systems. *International Journal of Control*, 79(07):719–728, 2006.

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

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

[15] 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.

[16] E. F. Moore. Gedanken-experiments on sequential machines. *Annals of mathematics studies*, 34:129–153, 1956.

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

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

[19] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *International Journal of Control*, 48(1):162–186, 2009.

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

[21] A. N. Trahtman. The road coloring problem. *Israel Journal of Mathematics*, 172(2):51–60, 2009.

[22] 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.