Gran Sasso Science Institute

MATHEMATICS IN NATURAL, SOCIAL AND LIFE SCIENCES DOCTORAL PROGRAMME

Cohort XXXI - AY 2015/2018

Probabilistic methods for primitive matrix semigroups

PHD CANDIDATE
Costanza Catalano

PhD Thesis submitted on
January 20, 2019

ADVISOR
Prof. Raphaël Jungers
Université Catholique de Louvain

GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

SISSA
Scuola Internazionale Superiore di Studi Avanzati

Gran Sasso Science Institute

MATHEMATICS IN NATURAL, SOCIAL AND LIFE SCIENCES DOCTORAL PROGRAMME

Cohort XXXI - AY 2015/2018

Probabilistic methods for primitive matrix semigroups

PHD CANDIDATE
Costanza Catalano

PhD Thesis submitted on
January 20, 2019

ADVISOR
Prof. Raphaël Jungers
Université Catholique de Louvain

GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

ma per seguir virtute e canoscenza
SISSA
Scuola Internazionale Superiore di Studi Avanzati

Thesis Jury Members

Prof. Yurii Nesterov (Université Catholique de Louvain)
Prof. Michele Palladino (Gran Sasso Science Institute)
Prof. Jean-Eric Pin (IRIF, CNRS and University Paris-Diderot)
Prof. Igor Potapov (University of Liverpool)
Prof. Vladimir Protasov (Moscow State University, Università dell'Aquila)

Thesis Referees

Prof. Igor Potapov (University of Liverpool)
Prof. Vladimir Protasov (Moscow State University, Università dell'Aquila)

Abstract

We study the primitivity property of finitely generated matrix semigroups from a probabilistic point of view and via two approaches. A finite set of nonnegative matrices is called *primitive* if there exists a product of these matrices that is entrywise positive; the length of the shortest of these products is called the *exponent* of the set.

We firstly study the primitivity property of random matrix sets, by rephrasing it in terms of random labeled directed multigraphs. We extend classical models of random graph theory to labeled directed multigraphs and we show that these random models admit a sharp threshold with respect to the primitivity property. We also show that when primitive, these models have low exponent with high probability. We then prove that they exhibit the same threshold behavior with respect to the property of being column-primitive and we use these results for studying the 2-directability property and 3-directability property of random nondeterministic finite state automata (NDFAs). In particular, we show that an NDFA generated according to the uniform distribution admits a short 2-directing word and a short 3-directing word with high probability. Inspired by the probabilistic method, we then present a more involved randomized construction that generates primitive sets with large exponent with nonvanishing probability and we use our findings for exhibiting new families of synchronizing finite state automata with quadratic reset threshold.

Secondly, we embed the primitivity problem in a probabilistic game framework in order to study its properties. We develop a tool, that we call *the synchronizing probability function for primitive sets of matrices*, that captures the speed at which a primitive set reaches its first positive product thus representing the convergence of the primitivity process, and we show that this function must increase regularly in some sense. We then show that this function can be used for efficiently approximating the exponent of any given primitive set made of matrices having neither zero-rows nor zero-columns (NZ-matrices) and for (potentially) improving the upper bound on the maximal exponent among the primitive sets of NZ-matrices. Finally, we prove that in a primitive semigroup of matrix size $n \times n$, for all $k \leq \sqrt{n}$ the length of the shortest product having a row or a column with k positive entries is linear in n, question that is still open for synchronizing automata.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor prof. Raphaël Jungers for his support, priceless suggestions and ideas, endless patience, and for all the time he spent reading my various drafts. I thank him for all the opportunities he gave me, which have been crucial for my professional (and personal) development, and for funding me any time I needed.

Besides, I would like to thank prof. Protasov and prof. Potapov for their significant suggestions and for accepting the burden of reading my thesis, and prof. Guglielmi for tutoring me. My sincere thanks also goes to François Gonze for proof-reading and precious feedbacks, to Vladimir Gusev and Balasz Gerencsér for fruitful discussions and to Umer and Ludovic for their contribution. I want to express my gratitude to prof. Marcati and the Gran Sasso Science Institute for giving me this opportunity and for funding my conferences and travels.

I would like to thank all the beautiful people that I had the chance to meet in these three years. My gratitude goes to Eleonora for her friendship and for hosting me multiple times, to Aleksandar for our fulfilling conversations and de-stressing evenings, Lee for always reminding me the fun side of life and Joanne for being the amazing person she is. I thank Roberto for his support and, together with Lars and Gennaro, for hosting me when needed, Anna Paola for everything we shared, Filippo for our long phone calls when I was abroad and Giancarlo for all the cigarettes we shared. A special thanks also to Lucilla, who understood me as few people do.

I have endless gratitude for all the special people that I met in Belgium. I thank Carlos and Áaron for their support and for adopting me in their group when I did not know anybody else, Jeanne for her wise advices and endless help (I would have not survived without you), Gaia and Marco for being one of the most amazing persons I have ever met, Georgios for all the coffee breaks and for introducing me beautiful people, and Jovana for her support and for hosting me when I needed.

Lastly, I want to thank all the people who have been loving me and caring about me in these years (and still do), despite all my flaws.

A mio padre

Contents

Chapter 1

Introduction

In the first section of this this chapter we present the framework in which our work takes place, together with a brief introduction of the main concepts and ideas that will be used in this manuscript and the reasons that motivated it. We then summarize our contribution and its novelty in Section 1.2. A detailed state-of-the-art of our problem is later provided in Chapter 3.

1.1 Motivations

A finite set of nonnegative matrices is called *primitive* if there exists a product of these matrices, with repetitions allowed, that is entrywise positive. Equivalently, a finite set of nonnegative matrices is primitive if in the multiplicative semigroup generated by them there is a matrix that is entrywise positive (a *positive* matrix). Nonnegative matrix semigroups containing a positive matrix have been appearing in several fields as in stochastic switching systems [58, 90, 91, 93] or in time-inhomogeneous Markov chains [56, 57, 104], but the notion of primitive set was formalized just quite recently by Protasov and Voynov[1] [94] as an extension of the concept of *primitive matrix*, defined by Perron and Frobenius at the beginning of the 20th century in the famous theory that carries their names. Primitive sets have lately found applications in other fields as in consensus for discrete-time multi-agent systems [30], cryptography [41] and automata theory [17, 48]. All the applications of primitivity will be discussed more in detail in Chapter 3, Subsection 3.1.1. Primitivity of matrix sets can be interestingly rephrased in terms of labeled directed multigraphs: given a directed graph D with labelled edges and multiple edges allowed, we say that D is *primitive* if there exists a sequence of labels such that we can go from any vertex of the graph to any other vertex by following a path labeled by that sequence. The *exponent* of a primitive set is the length of its shortest positive product; in graph terms, we are looking for the shortest sequence of labels that connects each pair of vertices in a labeled directed multigraph.

In the last years, several papers have contributed in shedding light on primitivity. We mention here that Protasov and Voynov [94] proved that deciding whether a set of nonnegative matrices without zero-rows and zero-columns (called *NZ*-matrices or *allowable* matrices) is primitive can be done in polynomial time, and that a combinatorial characterization of primitive

[1] They actually called these sets *almost* primitive, but the adjective *almost* was dropped in the subsequent literature on primitive sets, see e.g. [17, 41, 48].

sets of this kind is possible (see Chapter 3, Theorem 3.7). Blondel et. al. [17] later proved that in the general case determining whether a set of a least three nonnegative matrices is primitive is an NP-hard problem and that the exponent can increase exponentially with respect to the matrix size; on the other hand, they proved that in case of NZ-matrices there exists a cubic upper bound on the exponent (see Chapter 3, Corollary 3.21). Better upper bounds have also been found for some classes of primitive sets [48, 57].

Primitive sets gained importance also in view of their strong connection to synchronizing (or directable) deterministic finite state automata (DFAs). Synchronizing DFAs are well-studied objects in theoretical computer science that are often used as models for error-resistant systems. Indeed, a synchronizing DFA is a finite state machine that can be reset to a fixed known state, independently on the current state in which the machine is. A sequence of inputs that resets a synchronizing DFA is called a *reset* or *synchronizing* word. The formal definition of synchronizing DFAs together with a survey of the most important results concerning them can be found in Chapter 3, Section 3.2. Other applications of synchronizing DFAs appear in symbolic dynamics [76], in robotics for part handling problems [78] and in resilience of data compression [97, 103]. Blondel et. al. [17] and Geréncser et. al. [48] showed that from any primitive set of NZ-matrices we can build a synchronizing DFA and, vice versa, from any synchronizing DFA we can build a primitive set. Furthermore, a primitive set with quadratic (cubic) exponent leads to a synchronizing DFA with quadratic (cubic) shortest reset word. This property is particularly of interest as one of the most long-standing open problem in automata theory regards indeed the length of the shortest reset word of a synchronizing DFA, called its *reset threshold*: in 1967 Černý [114] conjectured that any synchronizing DFA on n states has reset threshold of at most $(n-1)^2$. If this conjecture is true, the bound cannot be improved as there exists a family of DFAs having reset threshold of exactly $(n-1)^2$. Exhaustive search has confirmed the conjecture for small values of n [7, 33] while better upper bounds have been obtained for certain classes of DFAs [11, 50, 70, 99, 116]. Despite great efforts, it is still unclear whether this conjecture is true: on the one hand, the best known upper bound on the reset threshold of any n-state synchronizing DFA is cubic in n [43,87,108], on the other hand synchronizing DFAs with quadratic reset thresholds, called *extremal* or *slowly synchronizing* automata, are hard to detect and few families are known (see [7, 33, 37, 50, 73, 109] for examples and Table 3.1 in Chapter 3). Slowly synchronizing DFAs are hard to find in view of the fact that the *typical* length of the shortest reset word is proved to be much smaller: a DFA on n states sampled according to the uniform distribution has reset threshold of order $O(n \log n)$ with high probability (i.e. with probability that tends to 1 as n increases) [79].

The notion of synchronization can be extended to nondeterministic finite state automata (NDFAs); an NDFA can be seen as a generalization of a DFA in which the transitions from one state to another may be not defined or not uniquely defined. In this manuscript we focus on the 2-*directability* and the 3-*directability* properties of NDFAs, firstly introduced by Imreh and Steinby in [62]: loosely speaking, an NDFA is 2-directable if there exists a sequence of inputs w such that the set of states that can be reached by applying w is independent on the initial state, while an NDFA is 3-directable if there exists

a sequence of inputs w and a state that is reachable from any other state by applying w. Sequences of this kind are called, respectively, 2-*directing* and 3-*directing* words; the formal definitions of NDFAs and of the directability properties can be found in Chapter 3, Section 3.2. It is known that the length of the shortest 2-directing and 3-directing words of a directable NDFA on n states can be exponential in n [22, 46, 75]. Sets of binary $n \times n$ matrices can be equivalently seen as NDFAs on n states and, in this case, the primitivity property implies both the 2-directability and the 3-directability properties. An accurate description of the relation between primitive sets and directable DFAs and NDFAs is provided in Chapter 3, Section 3.3.

In this thesis, we approach primitivity from a probabilistic point of view and in two different ways: on one side we study the primitivity of random matrix sets, on the other side we embed the primitivity problem in a probabilistic game framework in order to study its properties.

The first approach is mainly inspired by *random graph theory* and *the probabilistc method*, and is presented in Chapter 4 as an extended version of our works [24, 25]. The theory of random graphs was initiated in the 60s by the outstanding work of two Hungarian mathematicians, Paul Erdös and Alfred Rényi, in their seminal papers [39, 40]. Since then it has become a rich and fast-growing field by attracting the interest of many great scientists, as the mathematician Béla Bollobás [18], that contributed to the development of this discipline. The theory of random graphs lies at the intersection between graph theory, combinatorics and probability theory: its objects of investigation are graphs where the edges appear at random, according to a given distribution. Random graphs can model, for example, *reliable networks* i.e. communication networks where each communication line has a certain probability to fail: we want to know what is the probability that it is still possible to send a message from any center to any other center despite the failures occurring. In graph theory terminology, this translates into checking the connectedness property of the underlying random graph. The random graph theory finds application also in other fields as in theoretical computer science, networks systems, natural and social sciences. The questions that are usually solved in random graph theory are typically of asymptotic nature, where we want to know the probability that a certain property (as the connectedness property for reliable networks) holds when the number of vertices of the graph n goes to infinity; if this probability tends to 1 we say that the property holds with *high probability*. In this way we can understand which are the *typical* characteristics of a graph: if a property holds with high probability it means that we expect to find it in the majority of graphs with not too small vertex set.

An interesting phenomenon happening in the theory of random graphs is the one of *thresholds*: roughly speaking, a random graph model has a threshold with respect to a given property if this property appears almost surely when the parameters of the model are above a certain threshold, while is does *not* appear almost surely when the parameters are below the threshold. We can then say that a threshold determines a phase transition between a moment at which a property is very unlikely to hold to a moment at which this property is very likely to hold. A short summary of random graph theory and its most important results can be found in Appendix A, as well as the formal definition of the threshold phenomenon.

Inspired by this rich literature, we investigate the primitivity property of random labeled directed multigraph by extending the two most famous models in random graph theory: the model where each edge appears with probability p (called the *binomial* model), and the model where exactly M edges are chosen uniformly at random (called the *uniform* model). For example, we extend the binomial model by considering a random model for labeled directed multigraphs on n vertices and m labels where, for any pair of vertices i and j and label l, an edge from i to j labeled by l appears with probability p; we are interested in the asymptotic probability that there exists a sequence of labels such that any two given vertices are connected by a path labeled by this sequence (the *primitivity* property). We find that this model admits a threshold with respect to the primitivity property and, surprisingly enough, this threshold is the same as the ones known for other properties of the binomial model, like the connectedness property. We also consider the *column-primitivity* property, that is the property of a nonnegative matrix set of admitting a product with a column that is entrywise positive; despite the column-primitivity property is weaker than primitivity, we prove that they share the same threshold behavior in the random models we consider. Moreover, the same threshold holds for the 2- and 3- directability properties of random NDFAs described by these models: this implies that an NDFA sampled according to the uniform distribution is both 2- and 3-directable with high probability, thus extending what was already known for random DFAs (see [12, 79]). We also investigate the *typical* length of the exponent of a primitive set; interestingly, it turns out to be less than quadratic, thus much smaller than the known upper bound. This fact extends again to column-primitive sets and to 2- and 3-directable NDFAs: the typical length of the shortest product having a positive column or of the shortest 2-directing and 3-directing words is less than quadratic, thus much smaller than what happens in the worst cases.

We exploit the randomized generation of primitive sets also in view of their relation with synchronizing DFAs. Since slowly synchronizing automata are difficult to find, it is natural to wonder whether a randomized procedure to generate DFAs that differs from the uniform distribution could obtain less structured synchronizing DFAs with possibly larger reset thresholds. Our idea is to design a suitable probability distribution over the set of synchronizing DFAs such that the measure of the set of slowly synchronizing DFAs with respect to this distribution does not vanish when the number of states n increases: in this way, for every n we would have a (large enough) positive probability to sample a slowly synchronizing DFA, and so a chance to actually generate (some of) them. This approach can be rooted back to *the probabilistic method*, developed in the 60s by the same Paul Erdős. The probabilistic method is a non-constructive method for ensuring the existence of a structure with certain desired properties by defining a suitable probabilistic space in which to embed the problem: if we manage to prove that a certain property happens with probability strictly greater than zero, than there must exist at least one object having this property, even if we might have no insight on how this object looks like. With this method, for example, it has been proved that for any positive integers g and k there exists a graph containing only cycles of length at least g and with chromatic number[2] at least k. For an account on

[2]We remind that the *chromatic number* of a graph G is the smallest number of colors

the probabilistic method we refer the reader to [3].

Our idea for generating slowly synchronizing automata is to generate *proper* synchronizing DFAs (i.e. synchronizing DFAs that need all their letters to synchronize) with more than two letters: this class of automata seemed to us the ideal class to look for extremal examples because proper synchronizing DFAs are are almost surely never generated by the uniform distribution [79] and they do not appear often in the literature (almost all the slowly synchronizing families that are known are 2-letters). To achieve this, what we will actually do is to design a randomized procedure that generates *proper primitive* sets (i.e. primitive sets that need every matrix to be primitive) having quadratic exponent with nonzero probability (and not too small); we will then show that proper primitive sets with quadratic exponent can be transformed into proper synchronizing DFAs with quadratic reset threshold. The reason why we are generating primitive sets instead of automata is that we can ensure them to be proper by making use of the mentioned characterization theorem for primitive sets, which provides a combinatorial property that a set must have in order *not* to be primitive: by ensuring it to each proper subset of a matrix set, we can make it proper. To the best of our knowledge, this is the first time where a constructive procedure for finding proper synchronizing DFAs is presented; we also show in this way that it is possible to develop suitable randomized procedures for generating slowly synchronizing DFAs. The primitive sets found by our algorithm are also one of the few examples known in the literature of primitive sets with quadratic exponent.

The second probabilistic approach to primitivity is inspired by game theory and smoothed analysis. The *smoothed analysis* appears in combinatorial optimization as a way to use probabilities in order to analyze the convergence of iterative algorithms on combinatorial structures (see for references [105]). In other words, the smoothed analysis is a way of measuring the complexity of an algorithm by giving a more realistic analysis of the practical performance. We have mentioned that computing the exponent of a primitive set is an NP-hard problem [48] and that the best upper bound known on the exponent of any primitive set of NZ-matrices is cubic with respect to the matrix dimension. On the other hand, there are no examples of primitive sets of NZ-matrices having cubic exponent (as otherwise the Černý conjecture would be disproved) and also few sets with quadratic exponent are known. We are interested in measuring how long it takes for a primitive set to reach its first positive product; in other words, we want to design a function that somehow expresses the speed at which this first positive product is reached, in order to possibly:

- have a tool for efficiently approximating of the exponent of a primitive set of NZ-matrices;
- improve the upper bound on the maximal exponent among the primitive sets of NZ-matrices and with same matrix size.

To do so, we embed the primitivity problem in a probabilistic game framework by defining a two-player zero-sum game on a suitable graph built from

that are needed to color all the vertices of G in such a way that any two adjacent vertices do not share the same color.

the primitive matrix set that we are considering: we call the *synchronizing probability function for primitive sets* (SPF) the function that describes the probability of winning of one of the two players if they both play optimally. By reformulating the game as a linear program and making use of convex optimization techniques, we show that the behavior of this function is closely related with properties of the primitive set. The SPF is an adaptation to primitive sets of the function developed by Jungers [68] for the study of synchronization of DFAs measuring the time at which a synchronizing DFA reaches its first reset word. Despite the similarities in the formulations of these two functions, they exhibit differences in their behavior as well as in the results that can be proven; the formulation for primitive sets seems at the moment more promising.

We conclude this section by declaring that in this manuscript we have tried to present all our results on primitivity both using the language of linear algebra (matrices, vectors...), the language of graph theory and, when it is the case, the language of automata theory. In this way the reader can choose the point of view that (s)he prefers or the one (s)he is the most accustomed to. We also did our best to provide examples of every object we were introducing as well as graphs and histograms to support our claims with numerical data, in the hope to make things clearer and more readable to the reader.

1.2 Contribution and outline

This manuscript is organized as follows.

In Chapter 2 is listed the notation that will be used through out the entire manuscript; the reader can thus refer to this chapter in case the use of some symbols or names is not clear or familiar.

Chapter 3 summarizes the state-of-the-art on primitive sets and directable DFAs and NDFAs. More precisely, Section 3.1 is devoted to primitive sets, their definitions and related examples and results; Subsection 3.1.1 provides details on the various applications of primitivity and introduces other similar concepts, as the *column-primitivity* property. Subsection 3.1.2 focuses on primitivity for NZ-sets and related results. Section 3.2 is devoted to deterministic and nondeterministic finite state automata, where they are formally defined. Subsection 3.2.1 focuses on synchronizing DFAs and the famous Černý conjecture by surveying the last achievements and providing examples, while Subsection 3.2.2 focuses on NDFAs and their 2- and 3-directability properties, again providing a summary of the last results and examples; we also mention in this section the notion of *partial* DFA. Section 3.3 concludes the chapter by reporting all the recent results that connect the world of binary matrix sets to automata theory; in particular, the relationship between primitive sets, column-primitive sets, synchronizing DFAs, directable NDFAs and partial DFAs is explained.

Chapter 4, as already mentioned, is about the randomized generation of primitive matrix sets and it is mainly based on our works [24,25]. In Subsection 4.1 we focus on *perturbed permutation sets*, which are sets of permutation matrices[3] with a 0-entry of one of the matrices changed into a 1. Sets of this

[3] A *permutation* matrix is a matrix having exactly one entry equal to 1 in every row and every column and all the other entries equal to zero.

kind have the least number of positive entries that a primitive set can have, which should intuitively lead to large exponents. We show that a perturbed permutation set generated according to the uniform distribution is primitive with high probability and that the rate of convergence to 1 as $n \to \infty$ of this probability is greater than $1 - n^{-1} - O\left(n^{-2}\right)$. We also show that a set of this kind has exponent of order $O(n \log n)$ with high probability. These results will constitute the building blocks of the main result of the subsequent Section 4.2, where we extend the binomial model for random graphs to labeled directed multigraphs: in other words, we consider the random set $\mathcal{B}_m(n,p)$ of $m \geq 2$ binary[4] matrices of size $n \times n$ where each entry of each matrix is independently set to 1 with probability p and to 0 with probability $1-p$. We show that these sets present a *sharp threshold* with respect to the primitivity property: more in detail, we show that when $p \geq (1+\alpha)(\log n + c)/n$ for some $c \in \mathbb{R}$ and $\alpha > 0$, a set of this kind is primitive and has exponent of order $O(n \log n)$ with high probability, while it is almost surely never primitive when $p \leq (1-\alpha)(\log n + c)/n$ for some $c \in \mathbb{R}$ and $\alpha > 0$. In the case $p \geq (1+\alpha)(\log n + c)/n$ we also show that the rate of convergence to 1 as n tends to infinity of the probability that $\mathcal{B}_m(n,p)$ is primitive is greater than $1 - n^{-1} - O\left(ne^{-np}\right) - O\left(n^{-2}\right)$. Finally, in the case $p = (\log n + c)/n$ for some $c \in \mathbb{R}$, the set exhibits an intermediate behavior and we show that its exponent is of order $O(n \log^3 n)$ with high probability under some conditions. This implies that a set of nonnegative matrices generated according to the uniform distribution is primitive but has low exponent (at most of order $O(n \log n)$) with high probability.

As corollaries, in Subsection 4.2.1 we show that any NDFA randomly generated according to the model $\mathcal{B}_m(n,p)$[5] is 2-directable and has a 2-directing word of length $O(n \log n)$ with high probability when $p \geq (1+\alpha)(\log n + c)/n$ for some $c \in \mathbb{R}$ and $\alpha > 0$, and that the 3-directability property of these sets has the same threshold described for primitivity. In particular, a random NDFA generated according to the uniform distribution has both a 2-directing word and a 3-directing word of length $O(n \log n)$ with high probability.

Finally, in Subsection 4.2.2 we show that the model $\mathcal{B}_m(n,p)$ presents the same sharp threshold with respect to the property of being column-primitive; we can thus say that, despite the column-primitivity property is weaker than primitivity, they almost coincide at the limit $n \to \infty$. We also show that for $p > (\log n + c)/n$, both the length of the shortest product with a positive column and the length of the shortest *scrambling*[6] product of the set $\mathcal{B}_m(n,p)$ are of order $O(n \log n)$ with high probability.

In Section 4.3 we consider another random model for matrix sets: the random set $\mathcal{B}_m(n,M)$ made of m matrices where each matrix is uniformly and independently chosen from the set of binary matrices having exactly M positive entries. In graph terms, $\mathcal{B}_m(n,M)$ is a random labeled directed multigraph on n vertices and m labels where, for each $l = 1, \ldots, m$, M directed edges labeled by l are chosen uniformly among all the sets of M egdes. This model

[4] A matrix is said to be *binary* if it has entries in $\{0,1\}$.

[5] In other words, we are considering random NDFAs on n states $\{q_1, \ldots, q_n\}$ and m letters $\{a_1, \ldots, a_m\}$ such that, for every $i, j \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, m\}$, the probability that the NDFA goes to state q_j from state q_i when applied the input a_k is equal to p.

[6] A nonnegative matrix A is *scrambling* if for any indices i, j there exists an index k such that $A[i,k] > 0$ and $A[j,k] > 0$.

can be seen as an extension of the uniform model in random graph theory (see Appendix A.1 or [18, 66]) to labeled directed multigraphs. We investigate the primitivity property of $\mathcal{B}_m(n, M)$ and we show that $M = n(\log n + c)$ represents a threshold. More precisely, we show that if $M > n(\log n + c)$ for some $c \in \mathbb{R}$, then $\mathcal{B}_m(n, M)$ is primitive and has exponent of order $O(n \log n)$ with high probability, while it is almost surely never primitive when $M < n(\log n + c)$ for some $c \in \mathbb{R}$. Notice that considering binary matrices is not restrictive as the primitivity property and the column-primitivity property are not influenced by the actual values of the positive entries of the matrices of the set; these results thus more generally hold for random sets of nonnegative matrices.

So far we have proved that our random models, when primitive, have low exponent with high probability. In Section 4.4 we present a more involved random model that manages to generate primitive sets with quadratic exponent with nonvanishing probability. This model is described by a randomized algorithm that constructs proper primitive perturbed permutation sets and is reported in Subsection 4.4.1, where we also show that we can easily transform proper primitive perturbed permutation sets into proper synchronizing DFAs. To the best of our knowledge, this is the first time where a constructive procedure for finding proper synchronizing DFAs is presented. In Subsection 4.4.2 we report the numerical results obtained by our randomized algorithm: we show that it manages to obtain sets with larger exponents than the uniform generation and other simple random models, and that with positive (and big enough) probability it generates primitive sets with quadratic exponent. In Subsection 4.4.3 we present the new families of slowly synchronizing DFAs that we built from the primitive sets found by our algorithm: they are 3-letter DFAs that do not resemble the Černý's family and with reset threshold of order $\Omega(n^2/4)$. This last result improves the state of the art in the direction initiated by Gonze et. al. in [49]: they prove that the largest diameter of the square graph (see Definition 15, Chapter 3) among the DFAs on n states and made of $m \geq 2$ permutation matrices is lower bounded by $n^2/4 + o(n^2)$ when n is odd. We prove that this lower bound also holds with respect to the largest diameter of the square graph among the *synchronizing* DFAs on n states containing $m \geq 2$ permutation matrices, for any $n \in \mathbb{N}$. Our families also generalize the slowly eulerian synchronizing automata presented by Szykuła and Vorel in [109].

Chapter 5 is about a new tool for studying the primitivity phenomenon that we call the *sychronizing probability function* for primitive sets (SPF), and it is mainly based on our works [9, 26, 27]. In Section 5.1 we embed the primitivity problem in a two-player game framework and we define the SPF as the probability that the first player wins if both players play optimally: the exponent of a primitive set is then described as the time at which this function reaches the value 1 thus representing the convergence of the primitivity process. In Subsection 5.1.1 we reformulate the game as a linear programming problem and we provide an analysis of some theoretical properties of the SPF by showing that this function captures the speed at which a primitive set reaches its first positive product and that it must increase regularly in some sense. We report some numerical experiments in Subsection 5.1.2, where we show that the SPF can be used to approximate the exponent of a primitive set of NZ-matrices and how to potentially obtain a better upper bound on the

maximal exponent among the primitive sets of $n \times n$ NZ-matrices.

In Section 5.2 we introduce the function $\bar{K}(t)$, which is an upper bound on the SPF: in Subsection 5.2.1 we show that stronger theoretical properties hold for this function and we show that an estimate on the first time at which $\bar{K}(t)$ reaches the value 1 would imply an estimate on the exponent of any primitive set of NZ-matrices. In particular, we state a conjecture on $\bar{K}(t)$ that, if true, would imply a quadratic upper bound on the maximal exponent among the primitive sets of $n \times n$ NZ-matrices and a quadratic upper bound on the maximal reset threshold within a class of synchronizing DFAs on n states.

In Subsection 5.2.2 we introduce the k-rendezvous time (k-RT), which is the length of the shortest product in a nonnegative matrix semigroup having a row or a column with k positive entries. The notion of k-RT extends to primitive sets the well-known problem in automata theory of bounding the length of the shortest word mapping a set of k states onto a single state in a synchronizing DFA. This problem in automata theory is still open for any $k \geq 4$, while for $k = 3$ is known a quadratic upper bound in n [49] (the case $k = 2$ is trivial). Surprisingly, we find that for primitive sets if k is small enough ($k \leq \sqrt{n}$), then the k-RT is linear in n. We then show that the k-RT can be used to improve the upper bound on the first time $\bar{K}$ reaches the value 1 and thus to improve the upper bound on the maximal exponent among the primitive sets of $n \times n$ NZ-matrices.

We conclude the manuscript with Chapter 6, were we highlight again the novelty of our work and we list some open problems related to what we presented, together with some ideas for further developments.

Few appendices have been added at the end to clarify the meaning of some concepts used. Appendix A is a short summary on the main results of random graph theory: Section A.1 introduces the models of random graphs we mentioned before and some general results on them, while Appendix A.2 focuses on the notion of *perfect matchings* of random bipartite graphs. Appendix B is a (rather informal) explanation of the complexity classes that we mention in this manuscript, while Appendix C reports few results on the block-permutation structures (see Definition 11, Chapter 3) that a binary matrix can have, results that we thought did not really fit in the regular chapters.

Chapter 2

Notation

In this small chapter we summarize the notation that will be used throughout the manuscript.

We indicate with $[n]$ the set $\{1, \dots, n\}$ and given $x \in \mathbb{R}$, $x > 0$, we denote with $\lfloor x \rfloor$ the maximal natural number that is smaller or equal than x and with $\lceil x \rceil$ the minimal natural number that is greater or equal than x. We denote with S_n the set of permutations over n elements.
Given two sequences a_n, b_n for $n \in \mathbb{N}$, we say that:

- $a_n = O(b_n)$ if there exist $C > 0$ and $N \in \mathbb{N}$ such that for every $n > N$, $a_n \leq Cb_n$;
- $a_n = \Omega(b_n)$ if there exist $C > 0$ and $N \in \mathbb{N}$ such that for every $n > N$, $a_n \geq Cb_n$;
- $a_n = \Theta(b_n)$ if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$;
- $a_n = o(b_n)$ if $\lim_{n\to\infty} a_n/b_n = 0$;
- $a_n \ll b_n$ or $b_n \gg a_n$ if $a_n \geq 0$ and $a_n = o(b_n)$.

The set of the vectors of the canonical basis of $\mathbb{R}^n$ is represented by $\mathscr{E}_n = \{e_1, \dots, e_n\}$. We denote with e the vector having all its entries equal to 1; the length of e, when not explicitly stated, will be clear from the context. We denote with $\mathbb{R}^n_{\geq 0}$ the set of the vectors of length n with nonnegative real entries, also called *nonnegative* vectors, and with $\mathbb{R}^n_{>0}$ the set of the vectors of length n with positive real entries, also called *positive* vectors. The *support* of a nonnegative vector v is the set $supp(v) = \{i : v_i > 0\}$, and the *weight* of a nonnegative vector v is the cardinality of its support. A nonnegative vector v is *stochastic* if $\sum_i v_i = 1$. We denote with $[v]$ the binary vector such that $[v]_i = 1$ if $v_i > 0$, $[v]_i = 0$ otherwise.

We denote with $\mathbb{R}^{n\times n}$ the set of all the $n \times n$ matrices with real entries, with $\mathbb{R}^{n\times n}_{\geq 0}$ the set of all the $n \times n$ matrices with nonnegative real entries and with $\mathbb{R}^{n\times n}_{>0}$ the set of all the $n \times n$ matrices with positive real entries. A matrix $M \in \mathbb{R}^{n\times n}_{\geq 0}$ is called a *nonnegative* matrix and it is also denoted by $M \geq 0$; a matrix $M \in \mathbb{R}^{n\times n}_{>0}$ is called a *positive* matrix and it is also denoted by $M > 0$. Given a matrix M, we indicate with $M[:, j]$ its j-th column, with $M[i, :]$ its i-th row and with M^T its transpose. Given a set of matrices $\mathcal{M} = \{M_1, \dots, M_m\}$, $\mathcal{M}^T$ denotes the transpose set $\{M_1^T, \dots, M_m^T\}$. Given R and C two sets of

indices, we denote with $M[R, C]$ the submatrix of M made by the rows of M indexed by R and by the columns of M indexed by C. We say that a matrix is *binary* if it has entries in $\{0, 1\}$. A *binary set* of matrices is a set of binary matrices. We call a matrix a *permutation* matrix if it is binary and it has exactly one 1 in every row and every column. With a slight abuse of notation, S_n will also denote the set of the $n \times n$ permutation matrices. A *row-stochastic* matrix is a nonnegative matrix where the entries of each row sum up to one, a *column-stochastic* matrix is a nonnegative matrix where the entries of each column sum up to one. The set of all the binary row-stochastic matrices of size $n \times n$ is indicated by $\mathcal{R}_n$, while $\mathcal{C}_n$ indicates the set of all the binary column-stochastic matrices. A matrix is NZ if it has at least one positive entry in every row and every column; we denote with $\mathcal{NZ}$ the set of all the binary NZ-matrices. A finite set of NZ-matrices is also called an NZ-set. We indicate with $\mathbb{I}_{i,j}$ the matrix such that $\mathbb{I}_{i,j}[i, j] = 1$ and all the other entries are equal to 0.

Definition 1. We say that a matrix A *dominates* a matrix B ($A \geq B$) if for every i, j, $A[i, j] \geq B[i, j]$.

In this manuscript we make use of the following boolean product between matrices, for the reasons that will be explained in the next chapter:

Definition 2. Let B_1, B_2 be two binary matrices of size $n \times n$. The *boolean product* $B_1 \odot B_2$ is defined as

$$B_1 \odot B_2(i, j) = \begin{cases} 1 & \text{if } \sum_{k=1}^{n} B_1[i, k]B_2[k, j] > 0 \\ 0 & \text{otherwise} \end{cases}.$$

Since this product is the only matrix-product used in this manuscript, we will simply write B_1B_2 for $B_1 \odot B_2$. Given a vector v, the product B_1B_2v is to be understood as $(B_1 \odot B_2) \cdot v$ with $\cdot$ the standard matrix-vector product.

Given a graph G with vertex set V and edge set E, for any $v, w \in V$ we denote with (v, w) the undirected edge between v and w, with $v \to w$ the directed edge leaving v and entering in w and with $v \xrightarrow{l} w$ the directed edge from v to w labeled by l. We use the notation $(v, w) \in E$, $v \to w \in E$, $v \xrightarrow{l} w \in E$, to indicate that the edge (v, w), $v \to w$, $v \xrightarrow{l} w$,belongs to the graph G. We say that a path of length s from vertex v to vertex w is labeled by $L = l_{k_1} \dots l_{k_s}$ if there exist $w_2, \dots, w_s \in V$ such that for every $j = 1, \dots, s$;

$$w_j \xrightarrow{l_{k_j}} w_{j+1} \in E,$$

where $w_1 = v$ and $w_{s+1} = w$. Given a sequence of labels $L = l_{k_1} \dots l_{k_s}$, we use the notation $v \xrightarrow{L} w \in E$ to indicate that there is a directed path in $G = (V, E)$ from vertex v to vertex w labeled by L. A graph is connected if for any two given vertices, there is a path connecting them. A directed graph is strongly connected if for any two given vertices, there is a directed path the first to the second. A *multigraph* is a graph where multiple edges are allowed.

Given a probability distribution $\mathbb{P}$ over a finite space Ω and $A, B \subset \Omega$ two events, we indicate with $\mathbb{E}(A)$ the expected value of the event A and with $\mathbb{P}(A|B)$ the conditional probability of A given B under the distribution $\mathbb{P}$. We say that an event E_n that describes a property of a random structure depending on a paramenter n holds *with high probality* (*whp*) if the probability that E_n happens converges to 1 as $n \to \infty$.

Chapter 3

Preliminaries

In this chapter we introduce the main concepts used in the manuscript: the notion of *primitive set of matrices* and of *synchronizing automaton*. We present a short state-of-the-art of the major results and open problems on these objects and we explore their connection with other concepts that appear in the literature.

The chapter is organized as follows: Section 3.1 focuses on primitive sets of matrices and Section 3.2 focuses on synchronizing automata. Section 3.3 shows how these two notions are linked together by embedding them in a unified framework. Throughout the entire manuscript we just consider nonnegative matrices; therefore every matrix we mention is to be intended as nonnegative, unless we explicitly say that it is not.

3.1 Primitive sets of matrices

The concept of *primitive matrix* was introduced by Perron and Frobenius at the beginning of the 20th century when developing the famous theory in linear algebra that still carries their names, the *Perron-Frobenius theory*:

Definition 3. A nonnegative matrix M is *primitive* if there exists $s \in \mathbb{N}$ such that $M^s > 0$[1]. The *exponent* of a primitive matrix M, denoted by $exp(M)$, is the smallest exponent s such that $M^s > 0$.

The notion of primitive matrix and the Perron Frobenius theory find applications, among others, in probability for the ergodicity of Markov chains, in nonnegative linear systems [15] and consensus theory [34] (for a brief description of consensus theory, see Subsection 3.1.1).

A necessary condition for a matrix to be primitive is to be *irreducible*. We remind that a matrix M is irreducible if and only if there does not exist a permutation matrix P such that PMP^T is upper block-triangular: indeed, if PMP^T would be upper block-triangular, then any of its powers would still be upper block-triangular, so primitivity could never be attained. Notice also that a primitive matrix cannot have neither zero-rows nor zero-columns, i.e. it has to be NZ; this also implies that if $M^s > 0$ for a given $s \in \mathbb{N}$, then $M^{s'} > 0$ for any $s' > s$. In 1950 Wielandt (see [102] for a transcription of his result)

[1] We remind that we use the notation $M > 0$ to denote the fact that the matrix M has all positive entries.

proved that for any $n \times n$ primitive matrix M, $exp(M)$ is at most quadratic in n: more precisely, he proved that for any $n \in \mathbb{N}$ and $M \in \mathbb{R}_{\geq 0}^{n \times n}$ primitive,

$$M^{n^2-2n+2} > 0\,. \tag{3.1}$$

He also showed that this bound cannot be improved as there exists a matrix N, reported here below, such that $exp(N) = n^2 - 2n + 2$:

$$N = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

We say that two matrices A and B are equal *up to a permutation* if there exists a permutation matrix P such that $A = PBP^T$. The following theorem establishes that, up to a permutation, the matrix N described above is the only matrix attaining the maximal exponent of n^2-2n+2; moreover, matrices with exponent close to this extremal value are very few and there also exist some values that the exponent of a primitive matrix can never attain.

Theorem 3.1 ([36],Theorem 6, 7, 8 and Corollary 4)**.** *1. For every integer $n > 2$, up to a permutation, there exists exactly one primitive $n \times n$ matrix N such that $exp(N) = n^2 - 2n + 2$ and exactly one $n \times n$ primitive matrix M such that $exp(M) = n^2 - 2n + 1$.*

2. For every even integer $n > 4$, there does not exist any $n \times n$ primitive matrix M such that $n^2-4n+6 \leq exp(M) \leq n^2-2n+1$. Furthermore, up to a permutation, there exist exactly three matrices with exponent equal to $n^2 - 4n + 6$ if n is a multiple of three and exactly four matrices with exponent equal to $n^2 - 4n + 6$ if n is a not multiple of three.

3. For every odd integer $n > 3$, there does not exist any $n \times n$ primitive matrix M such that $n^2 - 3n + 4 \leq exp(M) \leq n^2 - 2n + 1$. Furthermore, up to a permutation, there exists exactly one $n \times n$ primitive matrix with exponent equal to $n^2 - 3n + 4$, one $n \times n$ primitive matrix with exponent equal to $n^2 - 3n + 3$ and one $n \times n$ primitive matrix with exponent equal to $n^2 - 3n + 2$.

4. For every odd integer $n > 3$, there does not exist any $n \times n$ primitive matrix M such that $n^2-4n+6 \leq exp(M) \leq n^2-3n+2$. Furthermore, up to a permutation, there exist exactly three matrices with exponent equal to $n^2 - 4n + 6$ if n is a multiple of three and exactly four matrices with exponent equal to $n^2 - 4n + 6$ if n is not a multiple of three.

Primitivity of a matrix can be rephrased in terms of directed graphs:

Definition 4. The *directed graph associated to* an $n \times n$ nonnegative matrix M is the digraph $\mathcal{D}_M = (V, E)$ with $V = [n]$ and $i \to j \in E$ if and only if $M[i, j] > 0$.

Remark 1. Notice that there exists a path in $\mathcal{D}_M$ from vertex i to vertex j of length l if and only if $M^l[i,j] > 0$. In view of this, it is easy to prove ([21], Theorem 3.2.1) that a matrix M is irreducible if and only if $\mathcal{D}_M$ is strongly connected, i.e. if and only if there exists a directed path between any two given vertices of $\mathcal{D}_M$. Irreducibility of a matrix can hence be tested by controlling the strongly connectedness of the digraph $\mathcal{D}_M$ via breadth-first search. Similarly, a matrix M is primitive if and only if there exists $s \in \mathbb{N}$ such that for any vertices i, j of $\mathcal{D}_M$ there exists a path from i to j of length s. In this case we say that the digraph $\mathcal{D}_M$ is primitive and we set $exp(\mathcal{D}_M) = exp(M)$.

Primitive matrices can be characterized as follows:

Theorem 3.2 ([60], Chapter 8)**.** *Let M be an $n \times n$ irreducible matrix. Then M is* not *primitive if and only if there exists a partition $\Omega = \dot{\bigcup}_{k=1}^{r} \Omega_k$ of $[n]$ with $r \geq 2$ such that:*

$$\forall\, k \in [r],\ \forall\, i \in \Omega_k, \quad M[i,j] > 0 \Rightarrow\ j \in \Omega_{k+1}\ ,$$

where in case $k = r$ we set $\Omega_{k+1} = \Omega_1$.

In other words, Theorem 3.2 states that an irreducible matrix M is not primitive if and only if, after a suitable permutation of the basis, it has the following cycle-block-permutation structure:

$$M = \begin{pmatrix} 0 & B_1 & 0 & \cdots & 0 \\ 0 & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & \cdots & 0 & B_{r-1} \\ B_r & 0 & \cdots & 0 & 0 \end{pmatrix}, \tag{3.2}$$

where, for each $k \in [r]$, the block B_k has dimension $|\Omega_k| \times |\Omega_{k+1}|$ (where for $k = r$ we set $\Omega_{k+1} = \Omega_1$); in this case we say that M acts as a *cyclic-permutation* on the partition Ω.

The notion of primitivity can be extended to *sets of matrices* in different ways: here we will focus on the one introduced by Protasov and Voynov in 2012 and defined here below. For other ways of extending primitivity to matrix sets we refer the reader to Subsection 3.1.1.

Definition 5 ([94], Definition 1)**.** Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be a finite set of nonnegative matrices. We say that $\mathcal{M}$ is *primitive* if $M_{i_1} \cdots M_{i_l} > 0$ for some indices $i_1, \ldots, i_l \in [m]$. In this case, we call $M_{i_1} \cdots M_{i_l}$ a *positive* product.

Example 1. The following matrix set is primitive:

$$\mathcal{M} = \left\{ A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\}. \tag{3.3}$$

It can be verified that $A^2BA^4BA^2 > 0$.

Similarly to primitive matrices, irreducibility is a necessary but not sufficient condition for a matrix set to be primitive:

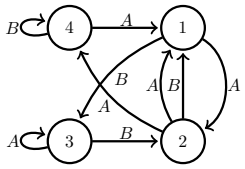


Figure 3.1: The digraph associated to the set $\mathcal{M}$ of Equation (3.3) with set of labels $\{A, B\}$.

Definition 6. A finite set of nonnegative matrices $\mathcal{M} = \{M_1, \ldots, M_m\}$ is said to be *irreducible* if the matrix $\sum_{i=1}^{m} M_i$ is irreducible.

Proposition 3.3. *If a matrix set $\mathcal{M} = \{M_1, \ldots, M_m\}$ is primitive, then it must be irreducible.*

Proof. If $\mathcal{M} = \{M_1, \ldots, M_m\}$ is reducible, then by Definition 6 there exists a permutation matrix P such that PM_iP^T is upper block-triangular for any $i \in [m]$. Consequently, any product $M_{i_1} \cdots M_{i_l}$ of matrices in $\mathcal{M}$ is such that $PM_{i_1} \cdots M_{i_l}P^T$ is upper block-triangular, so $\mathcal{M}$ cannot be primitive. □

A combinatorial characterization of primitive matrix sets similar to the one given by Theorem 3.2 is possible when all the matrices of the set are NZ, and it will be presented in Subsection 3.1.2 (Theorem 3.7). Notice that this time the NZ property is no more necessary for the primitivity of a set; indeed, the following set is clearly primitive but not NZ:

$$\left\{ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\} .$$

Primitivity of matrix sets can be rephrased in terms of *labeled* directed multigraphs:

Definition 7. Given a matrix set $\mathcal{M} = \{M_1, \ldots, M_m\}$, the *directed graph associated to* $\mathcal{M}$ is the labeled directed graph $\mathcal{D}_{\mathcal{M}} = (V, E)$ with multiple edges allowed and set of labels $\{l_1, \ldots, l_m\}$ where $V = [n]$ and $i \xrightarrow{l_k} j \in E$ if and only if $M_k[i, j] > 0$.

For simplicity, we will often use as labels of the labeled directed graph $\mathcal{D}_{\mathcal{M}}$ the matrix set $\{M_1, \ldots, M_m\}$ itself.

Example 2. Figure 3.1 represents the directed graph associated to the matrix set $\mathcal{M}$ in Equation (3.3).

Proposition 3.4. *Let $\mathcal{M}$ be a nonnegative matrix set and let $\mathcal{D}_{\mathcal{M}}$ be its associated directed graph. Then:*

(I) *$\mathcal{M}$ is irreducible if and only if $\mathcal{D}_{\mathcal{M}}$ is strongly connected;*

(II) *$\mathcal{M}$ is primitive if and only if there exists a sequence of labels $l = l_{k_1}, \ldots, l_{k_s}$ such that for any vertices i and j of $\mathcal{D}_{\mathcal{M}}$ there exists a path from i to j labeled by l (i.e. $i \xrightarrow{l} j \in E$).*

Proof. Item (I) is a direct consequence of Definition 6 and Remark 1.
Notice that there exists a path in $D_{\mathcal{M}}$ from vertex i to vertex j of length s labeled by $l_{k_1}, \ldots, l_{k_s}$ if and only if $M_{k_1} \cdots M_{k_s}[i,j] > 0$; hence (II) follows. □

It is then natural to wonder what is the minimal length of the labels' sequence that connect each vertex to any other vertex; in other words, we are asking what is the minimal length of a positive product in a primitive set.

Definition 8. The *exponent* of a primitive set $\mathcal{M}$ is the length of its shortest positive product and it is denoted by $exp(\mathcal{M})$. We indicate by $exp(n)$ the maximal exponent among all the primitive sets of $n \times n$ matrices.

Example 3. It can be verified that the exponent of the set $\mathcal{M}$ in Equation (3.3) is equal to 10.

Wieland's bound (3.1) implies that determining whether a matrix is primitive is decidable in polynomial time and so is computing its exponent; it is then natural to ask the same question for primitive sets of matrices. Unfortunately, in this case things become more complicated: Blondel et. al. [17] proved that the problem of deciding whether a matrix set is primitive is decidable, but NP-hard when the set has at least three matrices. The case of two matrices is, to the best of our knowlegde, still of unknown complexity. We will see in Subsection 3.1.1 that for sets of NZ-matrices the problem of deciding primitivity becomes somehow easier, as in this case it has polynomial-time complexity. Clearly, since determining whether a set is primitive is an NP-hard problem, so must be computing the exponent of a primitive set. Geréncser et. al. [48] showed that $exp(n)$ is at least exponential; in particular they proved that $exp(n)$ is $\Omega(3^{\frac{n}{3}})$ and $O(n^2 4^{\frac{n}{3}})$. Moreover, they showed the following asymptotic behavior of $exp(n)$:

$$\lim_{n\to\infty} \frac{\log exp(n)}{n} = \frac{\log 3}{3}. \tag{3.4}$$

It follows that the shortest positive product of a primitive set can be exponentially long. Again, for NZ-sets things become easier and we will see in Subsection 3.1.2 that in this case a cubic upper bound on $exp(n)$ is established. We underline the fact that Equation (3.4) provides an estimate on the magnitude of the exponent of a primitive set in the *worst* case, but it does not provide any information about the *average* behavior, i.e. what is the typical magnitude of the exponent of a primitive set. We will provide results in this direction in Chapter 4.

Finally, notice that the property of being primitive is affected just by the position of the positive entries within the matrices of the set and not by their actual values. Without loss of generality we can then assume that our matrix sets are made of *binary* matrices. Furthermore, we can assume to multiply the matrices within the semigroup of binary matrices, that is to use the so-called *boolean product* (see Chapter 2, Definition 2). In this manuscript, every matrix product can be equivalently read as a boolean product; the boolean product will also play a crucial role in the definition of the *Synchronizing Probability Function for primitive sets* in Chapter 5, even if it will add some difficulties in the study of its behavior due to its non-linear nature.

3.1.1 Applications of primitivity and related concepts

The concept of primitive set finds application in many different fields as in stochastic switching systems [91, 93], consensus for discrete-time multi-agent systems [30], time-inhomogeneous Markov chains [57, 104], criptography [41] and, finally, automata theory [17, 48]. We report in this subsection some of these applications together with some concepts closely related to primitivity. The connection between primitive sets and automata theory will be exhaustively explored in Section 3.3.

Synchronization of routers

Suppose to have four routers connected in a network as in Figure 3.2 a), router i initially having packets of type i, for $i = 1, 2, 3, 4$. The routers can communicate and transfer the packets between each others along the edges of the network; an edge can be used to transfer packets in just one direction or both ways. How they can exchange the packets is determined by the *routing protocol*, which is represented by the directed graph in Figure 3.2 b): the routers can choose to transfer the packets either by following the direction of the solid edges, or by following the direction of the dotted edges. For example, if the routers exchange theirs packets according to the dotted edges starting from the initial configuration of Figure 3.2 b), we will find in router 1 the packets of type **3** and **2**, in router 2 the packets of type **1**, in router 3 the packets of type **4** and in router 4 the packets of type **3**. If then they exchange their packets according to the solid edges, we will find in router 1 the packets of type **1**, in router 2 the packets of type **4**, in router 3 the packets of type **1** and **3** and in router 4 the packets of type **2** and **3**. We now want to know if it is possible that all the routers receive all the packets; since routers have no memory, we can reformulate the question as:

Q) Does there exists a $k \in \mathbb{N}$ such that, by choosing at each time $t = 1, \dots, k$ the edges (solid/dotted) according to which the routers transfer their packets, at time k all the routers receive *simultaneously* all the packets?

We can answer this question by describing the routing protocol with the following matrix set:

$$\mathcal{M} = \{M_1, M_2\} = \left\{ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \right\} ,$$

where M_1 represents the solid edges of the network (see Figure 3.2 b)) and M_2 its dotted edges. The routers can now transfer the packets by applying, at each time, the matrix M_1 or the matrix M_2. A product $M = M_{i_1} \cdots M_{i_l}$ for $i_1, \dots, i_l \in \{1, 2\}$, represents where the packets of the network are by applying matrix M_{i_t} at time t, for $t = 1, \dots, l$: in particular, $supp(M[:, i])$ indicates the packets that are in router i at time $t = l$. It is now clear that the answer to question Q) is positive if and only if the set $\mathcal{M}$ is primitive. It is easy to check that the product $M_1M_2M_2M_1M_2M_2M_1$ is positive, so by applying it to our network, at time $t = 7$ all the routers will receive simultaneously all the packets. Furthermore, since $exp(\mathcal{M}) = 7$, the routers will not receive

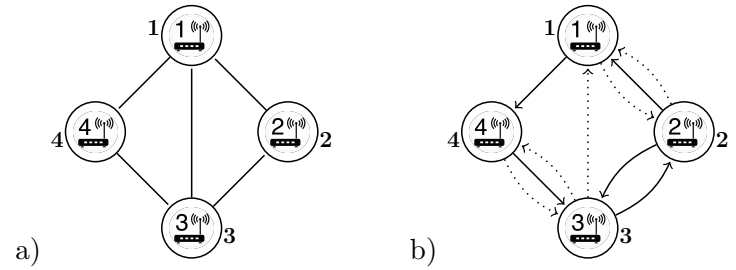


Figure 3.2: a) Network of routers. b) Example of a routing protocol on a network of routers. The number in bold next to each vertex indicates the type of packets each router initially has.

simultaneously all the packets any time sooner than $t = 7$. This example can be obviously generalized to n routers connected in a network, each of them having a different kind of packets and with a routing protocol described by a set of binary matrices. In this case there is a time at which all the routers receive all the packets simultaneously if and only if the routing protocol is a primitive set of matrices and the first time at which this can happen is equal to the exponent of the set.

Switched systems

Primitivity can be seen as one of the simplest reachability problems for switched systems. Let $\mathcal{M} = \{M_1, \dots, M_m\}$ be a finite set of nonnegative matrices. We consider the following discrete-time switched system:

$$\begin{cases} x_0 \in \mathbb{R}^n_{\geq 0} \\ x_{k+1} = M_{i_k} x_k, \quad M_{i_k} \in \mathcal{M}, \quad k \geq 1 \end{cases} \tag{3.5}$$

where at each time $k \geq 1$ a matrix $M_{i_k} \in \mathcal{M}$ is chosen and applied to the vector x_k. Since the matrices are nonnegative and $x_0 \in \mathbb{R}^n_{\geq 0}$, it follows that $x_k \in \mathbb{R}^n_{\geq 0}$ for every $k \geq 1$. A natural question to be asked is: can the switched system reach the interior of the nonnegative orthant? More formally, we are interested in the following questions:

Q1) Is it possible to choose at each time k a matrix $M_{i_k} \in \mathcal{M}$ such that at a certain time $\bar{k} \in \mathbb{N}$, $x_{\bar{k}} \in \mathbb{R}^n_{>0}$ independently on the initial condition x_0?

Q2) And, if it is possible, what is the first time $\bar{k}$ at which this can happen?

Primitivity provides an answer to these questions: the answer to question Q1) is positive if and only if the set $\mathcal{M}$ is primitive and the answer to question Q2) is $\bar{k} = exp(\mathcal{M})$. Indeed, if $M = M_{i_1} \cdots M_{i_r}$ is a positive product, then $Mx_0 > 0$ for all $x_0 \in \mathbb{R}^n_{\geq 0}$. On the other hand, if $Mx_0 > 0$ for all $x_0 \in \mathbb{R}^n_{\geq 0}$, then $Me_i = M[:, i] > 0$ for all $i \in [n]$ (where e_i is the i-th element of the canonical basis) and so M has to be a positive product.

Notice that a positive answer to question Q1) implies the existence of a switching sequence that contracts the state space of the system (3.5) into a one-dimensional subspace. Indeed, let $M = M_{i_1} \cdots M_{i_r}$ be a positive product of elements from $\mathcal{M}$ and suppose to apply to the system the periodic sequence $\dots M_{i_1} \dots M_{i_r} M_{i_1} \dots M_{i_r} = \dots MMM$. Since M is positive, it has a simple

dominant eigenvector v; this implies that as $k \to \infty$, for any initial condition x_0 the angle between $M^k x_0$ and v converges to 0. Consequently, for any initial conditions $x_0, y_0 \in \mathbb{R}^n_{\geq 0}$, as $k \to \infty$ the angle between $M^k x_0$ and $M^k y_0$ converges to 0, which means that the state space contracts to the one-dimensional subspace gnerated by the vector v.

Consider now some parameters $p_1, \ldots, p_m \in [0, 1]$ such that $\sum_{j=1}^m p_j = 1$, and let $d_1, d_2, \cdots \in [m]$ be a sequence of independent and identically distributed random variables such that, for all $k > 1$ and $j \in [m]$, $\mathbb{P}(d_k = j) = p_j$. We say that the switched system (3.5) is *stochastic* if at each time $k \geq 1$ the matrix of the switched system is chosen according to M_{d_k}. These systems are used, for example, to model random components failure in manufactoring systems (see [19], Chapter 1). The *largest Lyapunov exponent* of a stochastic discrete-time switched system is defined as

$$\lambda = \lim_{k \to \infty} \frac{1}{k} \mathbb{E} \left(\log \| M_{d_1} \cdots M_{d_k} \| \right) \tag{3.6}$$

where $\| \cdot \|$ is any matrix norm and $\mathbb{E}$ is the expected value. The largest Lyapunov exponent characterizes the rate of growth of the system with high probability: in particular, $\lambda < 1$ assures that all the trajectories of the system converge to 0 with high probability, hence the systems stabilizes with high probability. In general the problem of deciding whether $\lambda < 1$ or not is NP-complete [112] but it turns out that in case of primitive NZ-sets efficient algorithms are available for estimating the magnitude of the largest Lyapunov exponent of a stochastic discrete-time switched system [90, 91, 93].

Consensus

Switched systems as in Equation (3.5) where the matrices of the set are row-stochastic are particular cases of *discrete-time multi-agent systems*, and are widely used to represent systems of n agents who want to find an agreement on some value. In this case the j-th entry of x_k indicates the opinion of the agent j at time k and the matrices of $\mathcal{M}$ describe how each agent updates his opinion with respect to the others: $x_k = M_{i_{k-1}} x_{k-1}$ means that at time k, for all $j \in [n]$ the agent j is changing his opinion according to the weighted average on the other agents' opinion with weights $M_{i_{k-1}}[j, 1], \ldots, M_{i_{k-1}}[j, n]$. If

$$\lim_{k \to \infty} x_k = a(1, \ldots, 1)^T \tag{3.7}$$

for some $a > 0$, we say that the system reached *consensus*, because all the agents will eventually agreed on the value a. It turns out that the primitivity of $\mathcal{M}$ is a *sufficient* condition that assures the multi-agent system (3.5) to reach consensus independently on the initial condition x_0 [30]. A necessary and sufficient condition for consensus of these systems is given by the property of a matrix set to be *column-primitive*, concept that we introduce here below.

Column-primitive sets and the scrambling index

A nonnegative matrix with an entrywise positive column is called a *positive-column* matrix. Positive-column matrices are also called *Markov* matrices (see [104], Definition 4.7). Similarly to primitivity, this notion can be extended to *sets* of nonnegative matrices:

Definition 9. A set of nonnegative matrices $\mathcal{M}$ is *column-primitive* if it admits a product of these matrices, with repetitions allowed, that is positive-column. The length of its shortest positive-column product is called the *positive-column index* of $\mathcal{M}$ and is denoted by $pc(\mathcal{M})$.

If a set $\mathcal{M}$ is primitive, it is also column-primitive, and so $pc(\mathcal{M}) \leq exp(\mathcal{M})$. In Subsection 3.1.2 we will see that for irreducible sets of NZ-matrices, the concept of primitivity and column-primitivity are equivalent.

Consider again the switched system (3.5): we could ask a *weaker* question regarding the reachability of the nonnegative orthant, namely if *there exists* an initial condition $x_0 \in \mathscr{E}_n$ (where we remind that $\mathscr{E}_n$ indicates the the canonical basis of $\mathbb{R}^n$) such that the system reaches $\mathbb{R}^n_{>0}$ in finite time. The answer to this question is positive if and only if the set $\mathcal{M}$ is column-primitive and the first time t at which this can happen is $t = pc(\mathcal{M})$. Indeed, let $M = M_{i_1} \cdots M_{i_s}$ be a positive-column product such that $M[:, j] > 0$: then by taking $x_0 = e_j$, we have that $Mx_0 > 0$. On the other hand, let $x_0 = e_j \in \mathscr{E}_n$ such that $Mx_0 > 0$ for a product M: it must hold that $M[:, j] > 0$.

A nonnegative matrix A is said to be *scrambling* if for any row indices i and j there is a column index k such that $A[i, k] > 0$ and $A[j, k] > 0$. A positive-column matrix is a scrambling matrix; the opposite is not necessarily true. The following lemma holds:

Lemma 3.5 ([57], Corollary 4.3)**.** *If A is a scrambling matrix, then there exists $s \in \mathbb{N}$ such that A^s is positive-column.*

Scrambling matrices have been extensively investigated due to their application to Markov chains [85, 104]. Indeed, a scrambling row-stochastic matrix P has the property that P^t converges to a rank-one matrix as $t \to \infty$ (i.e. to a matrix whose rows are all equal); this fact it is known to guarantee the convergence of the Markov chain with transition matrix P to a unique limiting distribution [84].

Corollary 3.6. *A matrix set of nonnegative matrices admits a product that is scrambling if and only if it is column-primitive.*

Proof. Trivial by Lemma 3.5. □

Definition 10. The *scrambling index* of a column-primitive set $\mathcal{M}$ is the length of its shortest scrambling product, and it is denoted by $scr(\mathcal{M})$.

Akelbek and Kirkland [2] have been studied the scrambling index in the case of a single matrix A, providing an upper bound on $scr(A)$ in terms of the order and the girth[2] of the digraph associated to A (see Definition 4).

For any column-primitive set $\mathcal{M}$, it clearly holds that $scr(\mathcal{M}) \leq pc(\mathcal{M})$. If $\mathcal{M}$ is a primitive set, then it also holds that $scr(\mathcal{M}) \leq pc(\mathcal{M}) \leq exp(\mathcal{M})$.

Column-primitive sets of row-stochastic matrices are also called *almost contractive* families (see e.g. [94]). The reason for this name comes from the fact that a finite set of row-stochastic matrices $\mathcal{S}$ admits an infinite product converging to a rank-one matrix if and only if $\mathcal{S}$ is column-primitive ([29], Proposition 2). Therefore, if we construct an infinite product by sampling each matrix independently and with nonzero probability from $\mathcal{S}$, this product

[2] We remind that the *girth* of a (directed) graph is the length of its shortest cycle.

will converge to a rank-one matrix with high probability, as any finite product of matrices from $\mathcal{S}$ will appear infinitely many times with high probability. We can thus say that almost all the infinite products of a column-primitive set of row-stochastic matrices converge to a rank-one matrix. This fact finds application in time-inhomogeneous Markov chains [104], where the transition matrix $P(t)$ of the Markov chain changes at each time t and it is typically chosen from a finite set of row-stochastic matrices: if at every time t the matrix $P(t)$ is chosen independently and with nonzero probability from a finite set $\mathcal{S}$, then the system converges with high probability if and only if $\mathcal{S}$ is column-primitive [32].

Column-primitive sets find applications also in consensus theory: in case of row-stochastic matrices, the system (3.5) reaches consensus independently on the initial condition x_0 if and only if the set $\mathcal{M}$ is column-primitive ([30], Proposition 1). Moreover, if the matrices of the system (3.5) are chosen at each time independently and with nonzero probability from $\mathcal{M}$, the system converges to consensus with high probability independently on the initial condition. Indeed, in this case each infinite product converges with high probability to a one-rank matrix, thus fulfilling Equation (3.7). The scrambling index influences the rate of convergence of the system to consensus: if $P \in \mathcal{M}$ is a scrambling product of length l, then P^t converges to a rank-one matrix at an average rate of $\lambda_2^{1/l}$, where λ_2 is the second largest eigenvalue of P.

Strongly and weakly primitive sets, and the membership problem

The notion of primitive matrix can be extended to sets of matrices in other ways. Cohen and Sellers [31] introduce the concept of *strongly primitive* set (also called *eventually primitive* set in [41]): a set of nonnegative matrices $\mathcal{M}$ is said to be strongly primitive if there exists $c \in \mathbb{N}$ such that every product of length c of elements in $\mathcal{M}$ is positive. The minimal c for which this holds is denoted by $sexp(\mathcal{M})$. Cohen shows that $sexp(\mathcal{M}) \leq 2^n - 2$ for any strongly primitive set $\mathcal{M}$ of $n \times n$ matrices and that this upper bound is sharp, due to the existence of a family of strongly primitive sets reaching $sexp(\mathcal{M}) = 2^n - 2$; this extremal family is made of NZ-matrices, so there is no hope to get a lower upper bound on $sexp(\mathcal{M})$ in case of NZ-matrices, as it happens for primitivity. Hartfiel ([57], Theorem 4.3) shows that if all the matrices of a set $\mathcal{M}$ dominate a primitive matrix B, then the set is strongly primitive and $sexp(\mathcal{M}) \leq exp(B)$.

Fornasini introduces in [42] the notion of *weakly primitive* sets (also called *k-primitive* sets in [92]), based on the Hurwitz product: a set of matrices $\mathcal{M}$ is said to be weakly primitive if there exists a tuple $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{N}^m$ such that the Hurwitz product[3] of $\mathcal{M}$ with respect to α is (entrywise) positive; the minimal $|\alpha| = \sum_i \alpha_i$ for which this happens is denoted by $wexp(\mathcal{M})$. In graph terms, a set $\mathcal{M}$ is weakly primitive if there exists $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{N}^m$ such that for any vertices i, j of $D_{\mathcal{M}}$, there exists a path from i to j of length $|\alpha|$ containing exactly α_i edges labeled by M_i, for all $i = 1, \ldots, m$. A characterization of weakly primitive sets in terms of cycles of $D_{\mathcal{M}}$ is given by Olesky et. al. in [80], where they also show that $wexp(n, m) = \Theta(n^{m+1})$, where $wexp(n, m)$

[3]The Hurwitz product of $\mathcal{M} = \{M_1, \ldots, M_m\}$ associated to a tuple $\alpha = (\alpha_1, \ldots, \alpha_m)$ is the sum of all the possible products of matrices from $\mathcal{M}$ in which M_i appears exactly α_i times.

denotes the maximal weak exponent over all the weakly primitive sets of m matrices of size $n \times n$. Protasov presents a combinatorial classification of irreducible weakly primitive sets of NZ-matrices ([92], Theorem 1), which is similar to the one for primitive NZ-sets that we will see in Section 3.3 (Theorem 3.7): an irreducible NZ-set of $n \times n$ matrices is *not* weakly primitive if and only if there exists a nontrivial partition of $[n]$ on which all the matrices of the set act as *commuting* permutations, where two matrices A and B having a block-permutation structure on a given partition with respect to the permutations σ_1 and σ_2 (see Section 3.2, Definition 11) respectively act as *commuting permutations* if $\sigma_1\sigma_2 = \sigma_2\sigma_1$. Finally, Gusev et. al. [54] exploit the connection between weakly primitive sets and digraphs weighted by a semigroup and prove the existence of a polynomial-time algorithm for recognizing weak primitivity of NZ-sets for any fixed k. They also show that if $f(n)$ is an upper bound on $rt(n)$ (see Section 3.2, Definition 16), then $wexp(\mathcal{M}) \leq m(f(n) + n - 1)$ for any weakly primitive set $\mathcal{M}$ of m NZ-matrices of size $n \times n$.

We conclude this section by presenting primitivity as a particular case of the *membership problem*:

Problem 1 (The membership problem)**.** Given a set of matrices (not necessarily nonnegative) $\mathcal{M} = \{M_1, \ldots, M_m\}$ and a matrix M, determine if M belongs to the semigroup generated by $\mathcal{M}$, i.e. if there exist some indices $i_1, \ldots, i_l \in [m]$ such that $M = M_{i_1} \cdots M_{i_l}$.

Primitivity can hence be rephrased as a membership problem with $\mathcal{M}$ a binary matrix set and M the all-ones matrix, with respect to the semigroup generated by $\mathcal{M}$ according to the boolean matrix product (see Definition 2). Despite its simple formulation, the membership problem is not easy to solve. Paterson [83] proved that the membership problem when M is the all-zeros matrix (also called the *mortality problem*) is algorithmically undecidable for 3×3 integer matrices, thus the membership problem is undecidable when the matrix dimension n is greater than 2. The membership problem becomes decidable in polynomial time when dealing with sets of commuting matrices[4] [10]. Finally, for sets of 2×2 integer matrices, the membership problem is decidable in case of nonsingular matrices [88] and in case of matrices whose determinants take values in $\{0, \pm 1\}$ [89]. We will see in Section 3.3 that the mortality problem of binary matrix sets is strongly connected to the existence of *killing words* in nondeterministic automata.

3.1.2 NZ-sets and related results

We remind that a matrix is NZ if it has a positive entry in every row and in every column. Other authors have been called a matrix of this kind an *allowable* matrix (see e.g. [56, 58]). Notice that products of NZ-matrices are NZ. Primitive sets of NZ-matrices, also called NZ-sets, can be characterized in a similar combinatorial way as the one presented in Theorem 3.2 for primitive matrices and this result is due to Protasov and Voynov [94]; we present it here below after few definitions:

Definition 11. Let $\Omega = \dot{\bigcup}_{l=1}^{k} \Omega_l$ be a partition of $[n]$ with $k \geq 2$. We say that an $n \times n$ matrix M has a *block-permutation structure on the partition* Ω

[4] A set of matrices $\mathcal{M}$ is *commuting* if for any $A, B \in \mathcal{M}$, $AB = BA$.

if there exists a permutation $\sigma \in S_k$ such that for all $l = 1, \ldots, k$ and for all $i \in \Omega_l$, if $M[i,j] > 0$ then $j \in \Omega_{\sigma(l)}$. In this case, we also say that the matrix M has a block-permutation structure on the partition Ω *with respect to the permutation* σ. Finally, we say that a *set* of matrices *has a block-permutation structure* if there exists a partition on which *all* the matrices of the set have a block-permutation structure.

Equivalently, a matrix M has a block-permutation structure on a partition $\Omega = \dot{\bigcup}_{l=1}^{k} \Omega_l$ with respect to the permutation $\sigma \in S_k$ if $M[\Omega_i, \Omega_j]$ is a zero-matrix for all $j \neq \sigma(i)$.

Example 4. The following matrix has a block-permutation structure on the partition $\Omega = \{\{1,2\},\{3,4\},\{5,6\}\}$ with respect to the permutation $\sigma = (1)(23)$.

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Theorem 3.7 ([94], Theorem 1)**.** *An irreducible set of NZ-matrices is* not *primitive if and only if the set has a block-permutation structure. Furthermore, in case an irreducible set of NZ-matrices is not primitive, the partition with the maximal number of blocks on which it has a block-permutation structure is unique.*

The hypothesis of irreducibility and NZ-matrices cannot be omitted in Theorem 3.7: indeed, a reducible set cannot be primitive (see Proposition 3.3); furthermore it is possible to build a set that is not NZ, nonprimitive and such that it does not admit a block-permutation structure on any nontrivial partition, as shown in the next example borrowed from [94]:

Example 5. The following matrix set is not primitive but it does not admit a block-permutation structure on any nontrivial partition. It is due by the fail in fulfilling the request of NZ-matrices:

$$\left\{ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \right\}$$

The proof of Theorem 3.7 presented in [94] is based on geometrical arguments; lately, both Blondel et. al. [17] and Alpin and Alpina [4] have been independently provided a combinatorial proof of it.

Primitivity of NZ-sets can be checked in polynomial-time: Protasov and Voynov presented in [94] an algorithm based on Theorem 3.7 that, once assured that the set is irreducible[5], checks primitivity spending less than $2mn^3$ operations, bound that can be lowered to $(2mp+1)n^2$ in case each matrix of the set has less than p positive elements in every column. Unfortunately,

[5]This can be done in polynomial-time by checking the irreducibility of the matrix $\sum_{i=1}^{m} M_i$ via breadth-first search, see Definition 6.

computing the exponent of a primitive NZ-set is still hard: more precisely, Geréncser et. al. [48] proved that, if $\mathcal{M}$ is a primitive NZ-set, the following exact computational complexities are established for the following problems (for precise definitions of the complexity classes see Appendix B):

1. The problem of deciding whether $exp(\mathcal{M}) \leq k$ is NP-complete;
2. The problem of deciding whether $exp(\mathcal{M}) = k$ is DP-complete;
3. The problem of computing $exp(\mathcal{M})$ is $FP^{NP[\log]}$-complete.

Definition 12. We define $exp_{NZ}(n)$ to be the maximal exponent attainable by a primitive NZ-set of matrix dimension $n \times n$.

Contrary to what happens in the general case, $exp_{NZ}(n)$ does not grows exponentially:

Proposition 3.8 ([17], Corollary 18)**.**

$$exp_{NZ}(n) \leq (n^3 + 2n - 3)/3 \ .$$

Better upper bounds on the exponent can be obtained for some classes of primitive NZ-sets; we here mention the class of sets made of *fully indecomposable*[6] matrices, whose exponent is $\leq n - 1$ [57], and the class of sets of matrices with *total support*[7], whose exponent is $\leq 2n^2 - 5n + 5$ [48]. In [48] Geréncser et. al. also prove that for all positive integer c there is a polynomial time algorithm that approximates the exponent of an NZ-set of $n \times n$ matrices within a factor n/c.

A primitive set of NZ-matrices $\mathcal{M} = \{M_1, \ldots, M_m\}$ is also characterized by the fact that *almost all* its products are positive, in the sense that, given $\{d_k\}_{k\in\mathbb{N}}$ a sequence of independent and identically distributed random variables with uniform distribution on $[m]$, it holds that

$$\lim_{k\to\infty} \mathbb{P}(M_{d_1} \cdots M_{d_k} > 0) = 1 \ .$$

Indeed, since the matrices are NZ, if M is a positive product then any product that contains M as a subproduct is positive. Furthermore it is known that, as $k \to \infty$, the probability that $M_{d_1} \cdots M_{d_k}$ contains M tends to one.

When a set is irreducible and NZ, the primitivity property and the column-primitivity property are equivalent:

Proposition 3.9. *Let $\mathcal{M}$ be an irreducible NZ-set. Then $\mathcal{M}$ is primitive if and only if $\mathcal{M}$ is column-primitivity.*

Proof. Suppose $\mathcal{M}$ is column-primitive. It then admits a product with a positive column, say in position k; we denote this product with P_k. The set is irreducible, so $D_{\mathcal{M}}$ is strongly connected by Proposition 3.4. This implies that for every $j \in [n]$, there exists a product B_j of elements of $\mathcal{M}$ such that

[6] A matrix A is *fully indecomposable* if for any row indices $R = \{r_1, \ldots, r_s\}$ and column indices $C = \{c_1, \ldots, c_{n-s}\}$, $A[R, C]$ is not a zero-matrix.

[7] A matrix M has *total support* if there exists a nonnegative matrix B such that for all i, j, $B[i, j] = 0$ if and only if $M[i, j] = 0$, and B is doubly stochastic, where a matrix is *doubly-stochastic* if it is both row-stochastic and column-stochastic.

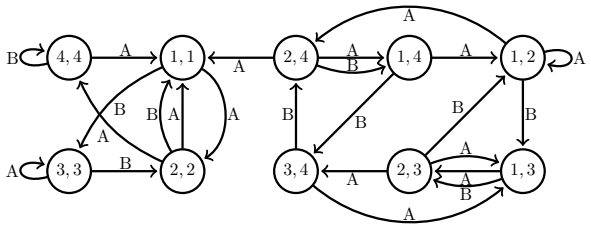


Figure 3.3: Square graph associated to the set $\mathcal{M} = \{A, B\}$ in Equation (3.3).

the matrix $P_j = P_k B_j$ has a positive column in position j. Let a_2 be an index such that $P_1[a_2, 2] > 0$ (it exists as the matrices are NZ): then the first two columns of $P_{a_2} P_1$ are positive. For $k = 3, \ldots, n$, let a_k be an index such that $P_{a_{k-1}} \cdots P_{a_2} P_1[a_k, k] > 0$: then the first k columns of $P_{a_k} P_{a_{k-1}} \cdots P_{a_2} P_1$ are positive. It follows that $P_{a_n} \cdots P_1$ is a positive product and so the set is primitive. The converse implication is trivial. □

Primitivity of irreducible NZ-sets can also be decided by the so-called *square graph criterion*, which basically checks the column-primitivity property of the set; we first define what is a *square graph* for an irreducible NZ-set and we then present the criterion in Proposition 3.10.

Definition 13. Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be an irreducible NZ-set of $n \times n$ matrices. The *square graph* associated to $\mathcal{M}$ is the labeled directed multigraph $\mathcal{SG}(\mathcal{M}) = (V, E)$ with labels $\{l_1, \ldots, l_m\}$, such that:

- $V = \{(i, j) : 1 \leq i \leq j \leq n\}$;
- $(i, j) \xrightarrow{l_k} (i', j') \in E$ if and only if $M_k[i, i'] > 0$ and $M_k[j, j'] > 0$, or $M_k[i, j'] > 0$ and $M_k[i, j'] > 0$.

A vertex of type (k, k) is called a *singleton*.

With a slight abuse of notation we will often take as label set of $\mathcal{SG}(\mathcal{M})$ the set $\mathcal{M} = \{M_1, \ldots, M_m\}$ itself, so the matrices $M_1, \ldots, M_m$ will label their corresponding edges.

Example 6. Figure 3.3 shows the square graph associated to the set $\mathcal{M} = \{A, B\}$ in Equation (3.3).

Remark 2. Notice that there exists a path in the square graph $\mathcal{SG}(\mathcal{M})$ from vertex (i, j) to vertex (i', j') sequentially labeled by $l_{k_1} \ldots l_{k_s}$ if and only if $M_{k_1} \cdots M_{k_s}[i, i'] > 0$ and $M_{k_1} \cdots M_{k_s}[j, j'] > 0$, or $M_{k_1} \cdots M_{k_s}[i, j'] > 0$ and $M_{k_1} \cdots M_{k_s}[j, i'] > 0$. In particular, there exists a path from (i, j) with $i \neq j$ to a singleton (v, v) labeled by $l_{k_1} \ldots l_{k_s}$ if and only if $M_{k_1} \cdots M_{k_s}[i, v] > 0$ and $M_{k_1} \cdots M_{k_s}[j, v] > 0$. This implies that the square graph of a primitive NZ-set is strongly connected, due to the existence of a positive product.

Proposition 3.10 (The square graph criterion for primitive sets)**.** *An irreducible NZ-matrix set $\mathcal{M}$ is primitive if and only if for all $(i, j) \in V$ there exists a path in $\mathcal{SG}(\mathcal{M})$ from (i, j) to a singleton.*

Proof. If $\mathcal{M}$ is primitive, then the thesis trivially follows by Remark 2.
Suppose now that for every $(i, j) \in V$ there exists a path from (i, j) to a

singleton. We show that we can construct a product with a positive column and so we conclude by using Proposition 3.9. Consider $M \in \mathcal{M}$ and c_1, d_1 be two indices such that $M[1, c_1] > 0$ and $M[2, d_1] > 0$ (they exist because the matrices are NZ). By hypothesis and Remark 2, there exists a product N_1 of elements of $\mathcal{M}$ such that $N_1(c_1, a_1) > 0$ and $N_1(d_1, a_1) > 0$, for some $a_1 \in [n]$. Let $A_1 = M$: the product $A_2 = A_1 N_1$ has two positive entries in column a_1 and rows $1, 2$. For $k = 2, \ldots, n$, let c_k be the index such that $A_k[k, c_k] > 0$, (a_k, a_k) be the singleton to which the vertex (a_{k-1}, c_k) is connected and N_k be a product of elements in $\mathcal{M}$ such that $N_k[a_{k-1}, a_k] > 0$ and $N_k[c_k, a_k] > 0$ and let $A_{k+1} = A_k N_k$. The matrix A_{k+1} has the first $k+1$ entries of column a_k positive, therefore A_n is a product with a positive column. □

Proposition 3.10 clearly holds also for the column-primitivity property, as in its proof we are building a product with a positive column. In this case, it would read as follows:

Proposition 3.11. *An NZ-set $\mathcal{M}$ is column-primitive if and only if for all $(i, j) \in V$ there exists a path in $\mathcal{SG}(\mathcal{M})$ from (i, j) to a singleton.*

Propositions 3.10 and 3.11 can be seen as an extension of the well-known square graph criterion for synchronizing automata that will be discussed in the next section, where the considered matrices are binary and row-stochastic. The square graph criterion for primitive sets says that we can decide whether an irreducible NZ-set is primitive by checking if all the nonsingleton vertices of its associated square graph are connected to a singleton. Unfortunately, although polynomial in time, this procedure is more expensive in terms of number of operations than the Protasov-Voynov algorithm. Indeed, we remind that the complexity of a breadth-first search algorithm on a graph $G = (V, E)$ is $O(|V| + |E|)$ in the worst case; in our case, the number of vertices of the square graph is $n(n+1)/2$ and, if p is the maximal number of positive entries that each column in the set has, then $|E| \leq mnp(np+1)/2$. This means that the breadth-first search has complexity

$$O\left(\frac{n(n+1)}{2} + \frac{mnp(np+1)}{2}\right) = O\left(mn^2p^2\right) ,$$

which is bigger than $O(mn^2p)$, the complexity of the Protasov-Voynov algorithm .

3.2 Automata

A *complete deterministic finite state automaton* (DFA) is a 5-tuple $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ where Q is a finite set of states, Σ is a finite set of input symbols called the *alphabet*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial state* and $F \subset Q$ is the set of the *accepted states*. The elements of Σ are called the *letters* of the automaton and a finite sequence of letters is called a *word*. We denote with Σ^* the set of all the finite words on the alphabet Σ. The transition function is naturally extended to the function $\delta^* : Q \times \Sigma^* \to Q$ by setting $\delta^*(q, w) = \delta(\ldots\delta(\delta(q, w_1), w_2)\ldots, w_s)$ for any word $w = w_1, \ldots, w_s \in \Sigma^*$ and $q \in Q$. We also use the notation $q.w = t$ for $\delta(q, w) = t$. A DFA reads a word w starting from the initial state q_0: if

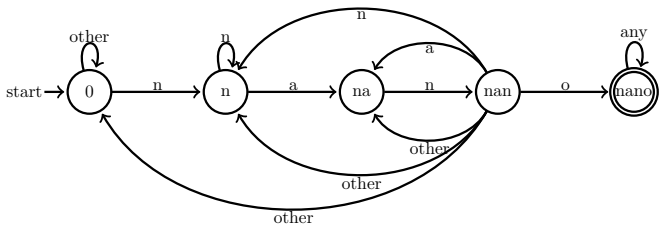


Figure 3.4: A DFA accepting the strings containing the word «nano». The set of states is $Q = \{0, n, na, nan, nano\}$, $q_0 = 0$, $F = \{nano\}$, Σ is the whole Latin alphabet. To improve the readability of the DFA we have labeled with *other* the action of any other letter but the ones already defined and with *any* the action of any letter of Σ.

$q.w \in F$, we say that the automaton *accepts* the word w, otherwise we say that is does not accept it. The set of words accepted by an automaton is called the *language recognized* by the automaton: the class of languages recognizable by DFAs is the class of *regular languages*. A DFA can be represented by a labeled directed graph $D = (Q, E)$ with set of labels Σ, where $q \xrightarrow{a} t \in E$ if and only if $\delta(q, a) = t$; the vertices of D that belongs to F are hightlighted in the graph, as well as for the initial state q_0. Automata are widely studied in theoretical computer science and discrete mathematics and they find applications in many diverse areas; we remind, among others, that they are used in pattern matching (e.g. searching for a patten in a string), in string matching (e.g. searching for a word in a document), in bioinformatics for motif search (genes sequence, proteins, ...) and in cyber security for deep packet inspection (finding virus, protocol anomalies, ...). Figure 3.4 represents the DFA that recognizes the pattern «nano» in a text string. A DFA in which some of the transitions are not defined is called *partial*.

Complete deterministic finite state automata are a particular case of non-deterministic finite automata (NDFA), where the transitions from one state to another by a letter may not be defined or not be uniquely defined. More formally, an NDFA is a 5-tuple $\mathcal{N} = \langle Q, \Sigma, \delta, q_0, F \rangle$ where Q is a finite set of states, Σ is a finite set of input symbols, $q_0 \in Q$ is the initial state, $F \subset Q$ is the set of the accepted states and $\delta \subseteq Q \times \Sigma \times Q$ is the transition function. The transition function δ is then extended to $\delta^* \subseteq Q \times \Sigma^* \times Q$ as before. An NDFA $\mathcal{N} = \langle Q, \Sigma, \delta, q_0, F \rangle$ is said to be *complete* if $\delta(q, a) \neq \emptyset$ for all $q \in Q$ and $a \in \Sigma$; in other words, in a complete NDFA every transition is defined.

In the rest of the section we will focus on *directable* DFAs and NDFAs and their properties. Finally, in Section 3.3 we will see how directable automata are linked to primitive sets of matrices.

3.2.1 Synchronizing DFAs and the Černý conjecture

There is a vast literature on synchronizing DFAs and the related Černý conjecture (see Conjecture 3.14). We here present the major results with no expectation to be fully exhaustive; for a more historical framework on synchronizing DFAs we refer the reader to [117].

Informally speaking, a DFA is *synchronizing* if it admits a word, called a

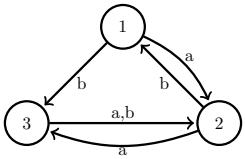


Figure 3.5: A synchronizing automaton with states $Q = \{1, 2, 3\}$ and alphabet $\Sigma = \{a, b\}$. The word $abba$ is synchronizing, as $q.abba = 2$ for every $q \in \{1, 2, 3\}$.

synchronizing or a *reset* word, that brings the automaton from every state to the same fixed state. In the theory of synchronizing DFA, the initial state q_0 and the set of accepting state F do not really play a role, as the focus is on the existence of a synchronizing word that resets the automaton independently on the initial state. In view of this, from now on we will identify a DFA just with a triple $\langle Q, \Sigma, \delta \rangle$.

Definition 14. A DFA $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ is called *directable* or *synchronizing* if there exists a word $w \in \Sigma^*$ and a state $q \in Q$ such that $\delta(q', w) = q$ for every $q' \in Q$. A word of this kind is called a *synchronizing*, a *reset* or a *directable* word.

Figure 3.5 presents an example of synchronizing DFA. The idea of synchronization is quite simple: we want to restore control over a device whose current state is unknown. For this reason, synchronizing DFAs are often used as models of error-resistant systems [28, 38]. They also find application in other research fields as in symbolic dynamics [76], in robotics for part handling problems (e.g. sorting objects in a specific direction not knowing the initial position) [78] or in resilience of data compression (Huffman codes) [97, 103].

The *square graph criterion* for synchronizing automata (see Proposition 3.12 below) determines whether a DFA is synchronizing or not in polynomial time. It is not a case that we call it in the same way as for primitive sets (see Proposition 3.10) because the two criteria are in fact the same; this will become clear in Section 3.3 where we will exploit the matrix representation of DFAs and NDFAs. For now, we present the square graph criterion for synchronizing automata in the traditional way:

Definition 15. Let $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA. The *square graph* associated to the automaton $\mathcal{A}$ is the labeled directed graph $\mathcal{SG}(\mathcal{A}) = (V, E)$ with set of labels Σ, where $V = \{(q, q') : q, q' \in Q\}$ and $(q_1, q_1') \xrightarrow{a} (q_2, q_2') \in E$ if and only if $(q_1.a, q_1'.a) = (q_2, q_2')$. A vertex of kind (q, q) is called a *singleton*.

Proposition 3.12 (Square graph criterion for synchronizing automata)**.** *Let $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA and $\mathcal{SG}(\mathcal{A}) = (V, E)$ be its associated square graph. Then $\mathcal{A}$ is synchronizing if and only if for every nonsingleton vertex $v \in V$ there is a path in $\mathcal{SG}(\mathcal{A})$ from it to a singleton.*

Proof. Suppose $\mathcal{A}$ is synchronizing and w is a synchronizing word. Then there exists $t \in Q$ such that $q.w = t$ for all $q \in Q$, which implies that $(q.w, q'.w) = (t, t)$ for all $q, q' \in Q$. Suppose now that every nonsingleton vertex is connected to a singleton. Given $Q' \subset Q$ and a word w, we define $Q'.w = \{q.w : q \in Q'\}$; clearly a word w is synchronizing if and only if $|Q.w| = 1$. We now

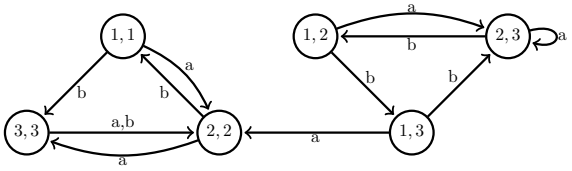


Figure 3.6: Square graph of the synchronizing automaton of Figure 3.5.

iteratively construct a word that has this property. Let $q_1, q_1' \in Q$, w_1 be the word such that $(q_1.w_1, q_1'.w_1) = (q_2, q_2)$ for some $q_2 \in Q$ and $K_1 = Q.w_1$; this implies that $|Q| > |K_1|$. Let now $q_2' \in K_1$, w_2 be the word such that $(q_2.w_2, q_2'.w_2) = (q_3, q_3)$ for some $q_3 \in Q$ and $K_2 = K_1.w_2$; this implies that $|K_1| > |K_2|$. For $i = 3, \ldots, |Q| - 1$, let $q_i' \in K_{i-1}$, w_i be the word such that $(q_i.w_i, q_i'.w_i) = (q_{i+1}, q_{i+1})$ for some $q_{i+1} \in Q$ and $K_{i+1} = K_i.w_{i+1}$; this implies that $|K_i| > |K_{i+1}|$. The set $K_{|Q|-1}$ has cardinality 1 and so the word $w_1 w_2 \ldots w_{|Q|-1}$ is a synchronizing word. □

Figure 3.6 shows the square graph of the automaton described in Figure 3.5: note that every nonsingleton vertex is connected to a singleton, in fact the automaton is synchronizing.

Proposition 3.12 shows that we can determine whether a DFA on n states is synchronizing in polynomial time with respect to n: it just suffices to build its associated square graph, which has $n(n+1)/2$ vertices and $|\Sigma|n(n+1)/2$ edges, and then perform a breadth-first-search on it. The breadth-first-search has complexity of $O(|V| + |E|)$ so we can solve the problem of determining whether an automaton is synchronizing in $O(|\Sigma|n^2)$ time.

When we want to restore control over a device by resetting it, we need to know not only if it is synchronizing but also which words are synchronizing and, possibly, which are the shortest ones.

Definition 16. The *reset threshold* of a synchronizing DFA $\mathcal{A}$, denoted by $rt(\mathcal{A})$, is the length of the its shortest synchronizing word. We also define $rt(n) = \max\{rt(\mathcal{A}) : \mathcal{A} \text{ is a synchronizing automaton on } n \text{ states}\}$.

Finding a reset word of a synchronizing DFA with no restriction on its length can still be done in polynomial time, as there exist some greedy algorithms that operate in time $O(n^3)$ (see for example [101]). When it comes to find a *short* synchronizing word or the *shortest* one, the problem becomes much harder; indeed, given $l \in \mathbb{N}$:

1. the problem of determining if $rt(\mathcal{A}) \leq l$ is NP-complete [38],
2. the problem of determining if $rt(\mathcal{A}) = l$ is CO-NP-hard [100],
3. the problem of computing $rt(\mathcal{A})$ is $FP^{NP[\log]}$-complete [81].

Also finding a relatively short reset word can be difficult:

Definition 17. Let $\mathscr{A}$ be an algorithm that, given any synchronizing automaton $\mathcal{A}$, finds an approximation $\mathscr{A}(\mathcal{A})$ of its reset threshold such that $\mathscr{A}(\mathcal{A}) \geq rt(\mathcal{A})$. We define the *approximation ratio* of $\mathscr{A}$ as the following function of $n \in \mathbb{N}$:

$$R_{\mathscr{A}}(n) = \sup\left\{\frac{\mathscr{A}(\mathcal{A})}{rt(\mathcal{A})} : \mathcal{A} \text{ is a synchronizing automaton on } n \text{ states.}\right\}.$$

Given a function $f : \mathbb{N} \to \mathbb{N}$, we say that $\mathscr{A}$ is an $f(n)$-*approximation* algorithm of the reset threshold if $f(n) \geq R_{\mathscr{A}}(n)$ for every n.

Berlinkov [12] proved that, unless P=NP, for any $c \in \mathbb{R}$ there is no c-approximation polynomial-time algorithm. Moreover, Gerbush and Heeriga [47] showed that there exists a constant C such that there is no $C\log(n)$-approximation polynomial-time algorithm, unless P=NP. On the other hand, they also showed that for every $k \geq 2$ there exists a polynomial-time algorithm (with a polynomial of degree k) with approximation ratio of $\lceil (n-1)/(k-1) \rceil$; lately, Ananichev and Gusev showed that this approximation ratio cannot be improved [6].

An easy way to compute upper and lower bounds on the reset threshold of a synchronizing DFA in time $O(n^2)$ is by using the square graph, as we are going to see in the following Corollary 3.13. To do so we need to define the *diameter* of the square graph, which has a slightly different formulation than the typical definition of the diameter of a graph[8]:

Definition 18. Let $\mathcal{A}$ be a synchronizing DFA and $\mathcal{SG}(\mathcal{A})$ be its square graph. For all $u \in V$, let $d_u = \min\{d(u, v) : v = (q, q) \in V\}$, where $d(u, v)$ is the length of the shortest path in $\mathcal{SG}(\mathcal{A})$ from u to v. We define the *diameter of the square graph* $\mathcal{SG}(\mathcal{A})$ as the following quantity:

$$diam(\mathcal{SG}(\mathcal{A})) = \max_{u=(r,t)\in V:\, r\neq t} d_u \ .$$

Corollary 3.13. *Let $\mathcal{A}$ be a synchronizing DFA on n states. It holds that*

$$diam(\mathcal{SG}(\mathcal{A})) \leq rt(\mathcal{A}) \leq (n-1)diam(\mathcal{SG}(\mathcal{A}))\,.$$

Proof. In the proof of Proposition 3.12 we have seen that the word $w = w_1 w_2 \dots w_{n-1}$ is synchronizing. For every $i = 1, \dots, n-1$, the length of w_i is smaller or equal than $diam(\mathcal{SG}(\mathcal{A}))$ and so $rt(\mathcal{A}) \leq |w| \leq (n-1)diam(\mathcal{SG}(\mathcal{A}))$. Let now w be a synchronizing word such that $|w| = rt(\mathcal{A})$. By definition, for all $r, t \in V$, $(r.w, t.w) = (q, q)$ for some $q \in V$ and so $diam(\mathcal{SG}(\mathcal{A})) \leq |w|$. □

One of the most longstanding open problems in automata theory concerns the reset threshold of a synchronizing DFA; it has been stated by Černý in 1964 in his pioneering paper on synchronizing DFAs and it reads as follows:

Conjecture 3.14 (The Černý conjecture [114])**.** *Any synchronizing DFA $\mathcal{A}$ on n states has a synchronizing word of length at most $(n-1)^2$.*

Černý also presented in [114] a family of automata $\{\mathscr{C}_n\}_{n\in\mathbb{N}}$ such that $\mathscr{C}_n$ has n states and $rt(\mathscr{C}_n) = (n-1)^2$, thus demonstrating that if his conjecture is true, then the bound is sharp. The automaton $\mathscr{C}_n$ has alphabet $\Sigma = \{a, b\}$ where

$$i.a = \begin{cases} i & \text{if } 1 \leq i \leq n-1 \\ 1 & \text{if } i = n \end{cases}, \quad i.b = \begin{cases} i+1 & \text{if } 1 \leq i \leq n-1 \\ 1 & \text{if } i = n \end{cases}. \tag{3.8}$$

Figure 3.7 reports the Černý's automaton $\mathscr{C}_n$ in its graph representation. In

[8]The *diameter* of a (directed) graph $G = (V, E)$ is usually defined as $\max_{u,v\in V} d(u, v)$, where $d(u, v)$ is the length of the shortest (directed) path from u to v.

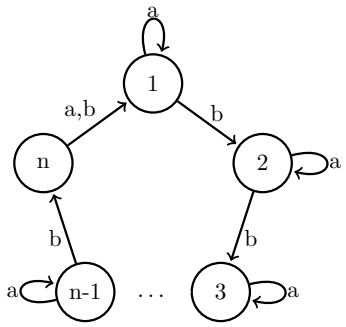


Figure 3.7: The Černý's automaton $\mathscr{C}_n$ on n states.

the last decades a great effort has been made to prove or disprove the Černý conjecture. Volkov ([116], Proposition 2.1) showed that, if for every strongly connected synchronizing DFA the Černý conjecture holds true, then so is for every synchronizing DFA; consequently from that point on the efforts have been focused mainly on strongly connected DFAs. Exhaustive search has confirmed the conjecture for $n \leq 6$ and any alphabet cardinality m [33], and for $7 \leq n \leq 10$ and $m = 2$ [110], but it is clear that even the most powerful computer cannot do much more as the search space increases exponentially in n and m and so some theoretical approaches are needed.

The validity of the Černý conjecture still remains unclear. Indeed, on the one hand for long time the best upper bound known for $rt(n)$ has been of $(n^3 - n)/6$, originated by the joint effort of Pin and its descending method [87] and a combinatorial result of Frankl [43]; it has recently been slightly improved to $(15617n^3 + 7500n^2 + 9375n - 31250)/93750$ by Szykuła [108], by (partially) solving the open question left by Gonze et. al. in [52]. On the other hand, DFAs having quadratic reset threshold in n, called *extremal* or *slowly synchronizing* automata, are very difficult to find and few of them are known; we will talk more in detail about these extremal DFAs a bit later in this paragraph. Synchronizing DFAs having reset threshold of exactly $(n-1)^2$, called *critical* automata, are even more rare: if we consider only *proper* critical automata, i.e. automata that need every letter to be synchronizing, just eight of them are known, and they are reported in Figure 3.8. If we count also the critical automata that can be obtained by the proper ones by adding one or more letters without modifying the reset threshold, we know 27 of them [33].

Better upper bounds on the reset threshold have been obtained for some classes of automata, and within certain classes it has been possible to prove the validity of the Černý conjecture. These results are partially reported in the first column of Table 3.1: we can observe that the Černý conjecture has been confirmed for the classes of Eulerian automata, pseudo-Eulerian automata and strongly connected weakly monotone automata, while for the classes of automata with full transition monoid, with simple idempotens, of one cluster automata and of 0−automata it has been found a quadratic upper bound in n on their reset thresholds, despite bigger than $(n-1)^2$. We provide the definitions of the classes of automata mentioned in the table here below:

Definition 19. A synchronizing automaton $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ with $|Q| = n$ is:

- *Eulerian* if it is strongly connected and the in-degree of any vertex is equal to its out-degree (and hence equal to the alphabet size);

- *pseudo-Eulerian* if we can find a probability $\mathbb{P}$ with support Σ such that the matrix $\pi(\mathbb{P})$ is doubly stochastic, where $\pi(\mathbb{P})$ is a $|Q| \times |Q|$ matrix such that $\pi(\mathbb{P})[q, r] = \sum_{a \in \Sigma : q.a = r} \mathbb{P}(a)$.
- with *full transition monoid* if the semigroup generated by the alphabet Σ is the full set of functions $\{f \mid f : [n] \to [n]\}$.
- *one cluster* if there exists a letter $a \in \Sigma$ such that for every $q, p \in Q$ there exist $r, s \in \mathbb{N}$ such that $p.a^r = q.a^s$.
- *strongly connected weakly monotone* if it is strongly connected and if there exists $l \in \mathbb{N}$ and a strictly increasing chain of stable binary relations[9] $\rho_0 \subset \rho_1 \subset \cdots \subset \rho_l$ on $Q \times Q$ such that:
  1. $\rho_0 = \{(q, q) : q \in Q\}$ is the identity relation;
  2. for each $i \in [l]$, $\pi_{i-1} := Eq(\rho_{i-1}) \subseteq \rho_i$, where $Eq(\rho_{i-1})$ denotes the smallest equivalence relation containing ρ_{i-1};
  3. for each $i \in [l]$, ρ_i / π_{i-1} is a partial order on Q/π_{i-1};
  4. $\pi_l = Q \times Q$.
- with *simple idempotents* if every letter $a \in \Sigma$ is either a permutation or idempotent (i.e. $a^2 = a$).
- a $0-$automaton with n nonzero states, if it has $n + 1$ states and exactly one sink state[10].

Because of the interest in the Černý conjecture and the fact that extremal automata are hard to detect, the search for synchronizing DFAs attaining quadratic reset threshold has been the subject of several contributions in recent years: Table 3.1 reports in the second column the reset threshold of some slowly synchronizing automata belonging to the corresponding families. Note that if we denote with $\mathcal{C}$ a generic class of automata and we set $rt_{\mathcal{C}}(n) = \max\{rt(\mathcal{A}) : \mathcal{A} \in \mathcal{C}, \mathcal{A} \text{ has } n \text{ states}\}$, then the reset threshold of a slowly synchronizing automaton belonging to $\mathcal{C}$ is a lower bound for $rt_{\mathcal{C}}(n)$. Other examples of slowly synchronizing automata can be found in [7, 33, 37, 50, 73]. Dzyga et. al. [37] also show that for any $m \in \mathbb{N}$, $m \geq 2$, there exists a slowly synchronizing automaton on n states and with m letters with reset threshold of order $n^2 - O(n)$.

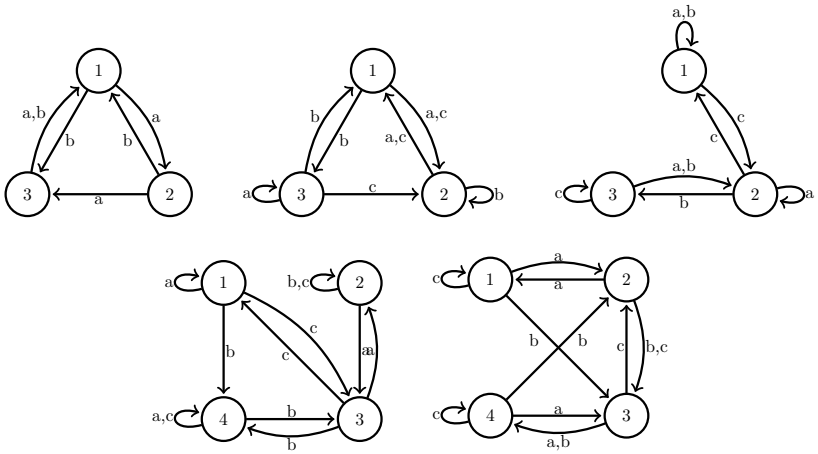
The extremal families found by Ananichev et. al. in [7] are particularly interesting because of their connection with the exponent of a primitive matrix and the road coloring problem (see Problem 2 at the and of this section). They exploit a result of Adler et. al. [1] stating that the digraph $D_{\mathcal{A}}$ of any strongly connected synchronizing automaton $\mathcal{A}$ is primitive and they show that $exp(D_{\mathcal{A}}) \leq rt(\mathcal{A}) + n - 1$ ([7], Proposition 2). By considering all the primitive digraphs on n vertices that realize the largest values of the exponent (see Theorem 3.1) and by labelling the edges of these digraph with a set of two labels, they produce families of two-letter synchronizing DFAs with quadratic reset threshold. Some of these families are presented in Figure 3.9.

[9] A relation ρ on $Q \times Q$ is *stable* if for any $p, q \in Q$ and $a \in \Sigma$, $(p, q) \in \rho$ implies $(p.a, q.a) \in \rho$.

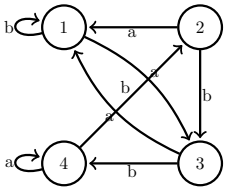
[10] A state q is a *sink* if for all $a \in \Sigma$, $q.a = q$.

Figure 3.8: The proper critical synchronizing automata known in the literature, excluding the Černý family.

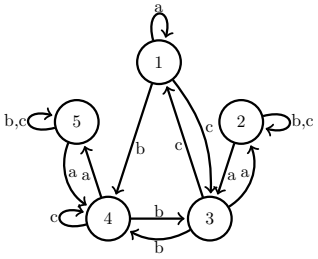
Critical automata found by Trakhtman in [110]:



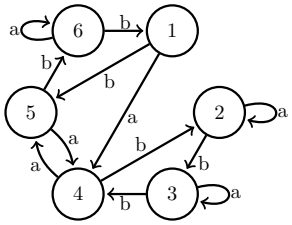
Černý-Piricka-Rozenaurova automaton [115]:



Roman automaton [95]:



Kari automaton [67]:



Class	**Upper b. on rt**	**Families with quadratic rt**
Eulerian automata	n^2-3n+3 Kari [70]	$(n^2-3)/2$ Szykuła and Vorel [109] (4 letters)
pseudo-Eulerian automata	n^2-3n+3 Steinberg [106]	$(n^2-3)/2$ Szykuła and Vorel [109] (4 letters)
Automata with full transition monoid	$2n^2-6n+5$ Gonze et. al. [50]	$n(n-1)/2$ Gonze et. al. [50] (n letters)
One cluster automata	$2n^2-7n+7$ Béal et al. [11]	$(n-1)^2$ Černý [114] (2 letters)
Strongly connected weakly monotone automata	$\lfloor n(n+1)/6 \rfloor$ Volkov [116]	Not yet found
Automata with simple idempotents	$2(n-1)^2$ Rystov [99]	$(n-1)^2$ Černý [114] (2 letters) $\geq (n^2+3n-6)/4$ for $n=4k+3$ [Conjectured $(n^2-1)/2$] $\geq (n^2+3n-8)/4$ for $n=4k+1$ [Conjectured $(n^2-1)/2$] $\geq (n^2+2n-4)/4$ for $n=4k$ [Conjectured $(n^2-2)/2$] $\geq (n^2+2n-12)/4$ for $n=4k+2$ [Conjectured $(n^2-10)/2$] **Our contribution** (3 letters)
0-automata with n nonzero states	$n(n+1)/2$ Rystov [98]	$n(n+1)/2$ Rystov [98]

Table 3.1: Table reporting upper bounds on the reset threshold for some classes of automata and examples of automata with large reset threshold belonging to these classes.

The great majority of the known extremal DFAs is two-letter and has a quite regular structure. In particular, the action of their letters is very much similar to the ones of Černý's, to the extent that almost all these automata present a cycle letter over the n vertices (like the letter a in $\mathscr{C}_n$, see Equation (3.8)) and all their other letters have an action similar to the one of letter b in $\mathscr{C}_n$.

Berlinkov [13] and Nicaud [79] formalized the fact that extremal automata are few and so very hard to find: the former proved that for any $m \geq 2$, if we generate a DFA of m letters according to the uniform distribution, then it is synchronizing with high probability. The latter proves that these DFAs have reset threshold of order $O(n \log^3 n)$ still with high probability, thus subquadratic. An experimental confirmation that two letters are usually enough to form a (fast) synchronizing automaton is provided in [61], where Igor et. al. show that the number of letters of a synchronizing DFA is negatively correlated to the magnitude of its reset threshold, out of a sampling of thousands DFAs uniformly generated. Interestingly, they also show that the diameter of the square graph of a synchronizing DFA is positively correlated with respect to the magnitude of its reset threshold. The following definition will help us to draw some conclusions from what we have just said:

Definition 20. A sychronizing automaton that needs every letter to synchronize is called a *proper* synchronizing automaton.

It follows that:

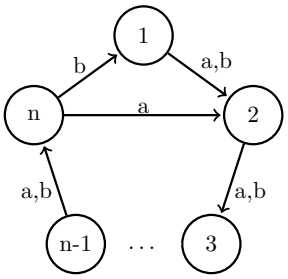
1. extremal automata belong to a set of measure almost 0 with respect to the uniform distribution, therefore there is very little hope to find them by a uniform sampling;

2. proper synchronizing automata with more than two letters are hard to find, as with high probability two letters are enough to make a DFA be synchronizing [12, 79].

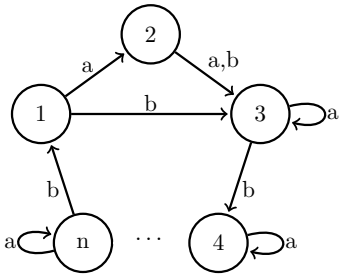
Item 2. suggests that proper synchronizing automata with more than two letters could form a suitable class in which to look for extremal examples, in view of the fact that they do not appear often in the literature and that the behavior of their reset threshold is still unclear. Item 1. implies that if we want to use a randomized procedure to generate possibly less structured automata with larger reset thresholds, we need something more involved than the uniform generation. Other randomized constructions have been considered in the literature: Gonze et. al. shows in [50] that for any $m \geq 2$, a DFA made of m permutation matrices chosen independently from S_n according to the uniform distribution and at least one matrix of rank[11] $< n$ is synchronizing and admits a synchronizing word of length $O(n \log n)$ with high probability (notice that any synchronizing DFA must have a matrix of rank $< n$). Recently Berlinkov and Nicaud [14] considered random *almost-group* automata, i.e. uniformly sampled automata made of $m \geq 1$ permutation letters and a letter of rank $< n$ which is a permutation over $n-1$ states; they show that for any $m \geq 1$, these kind of automata are synchronizing with high probability. The result of Gonze et. al. cited above straightforwardly implies that for any $m \geq 2$ a

[11] The *rank* of a binary row-stochastic matrix is the number of its nonzero columns.

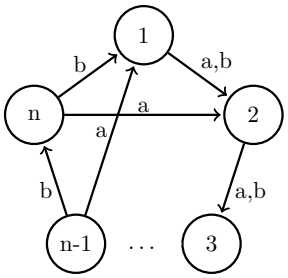
Figure 3.9: Some of the extremal automata found by Ananichev et. al. [7] by labeling a primitive digraph with large exponent.



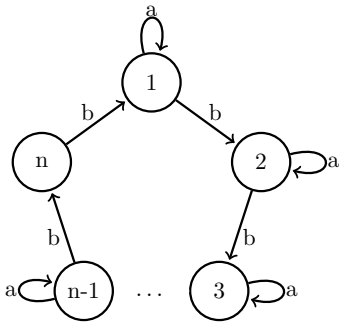
Family $\mathscr{W}_n$: $rt(\mathscr{W}_n) = n^2 - 3n + 3$.



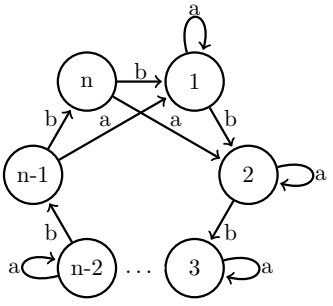
Family $\mathscr{E}_n$ (odd n): $rt(\mathscr{E}_n) = n^2 - 3n + 2$.



Family $\mathscr{D}_n$: $rt(\mathscr{D}_n) = n^2 - 3n + 4$.



Family $\mathscr{F}_n$: $rt(\mathscr{F}_n) = n^2 - 3n + 3$.



Family $\mathscr{B}_n$ (odd n): $rt(\mathscr{B}_n) = n^2 - 3n + 2$.

random almost-group automaton has reset threshold of order $O(n \log n)$ with high probability; it is still unclear what happens in the case $m = 1$. These results show that also for these other randomized constructions there is no hope to generate extremal automata: in Chapter 4 we will present a new randomized construction that manages to find synchronizing DFA with quadratic reset threshold.

We conclude this section by mentioning another problem connected to synchronizing DFAs, the *road coloring problem*, formulated in the 70s by Adler, Godwyn and Weiss [1] and recently positively solved by Trahtman in 2009 [111]. It is based on the fact that any directed graph with constant outdegree δ can be labeled (or colored) by δ labels (colors) such that every edge leaving the same vertex has a different label. Obviously, the obtained labeled digraph is a DFA: we say that the labeling (or coloring) is *synchronizing* if the resulting DFA is synchronizing. The road coloring problem reads as follows:

Problem 2 (The Road Coloring Problem)**.** Does every strongly connected directed graph with constant outdegree and with greatest common divisor (gcd) of the lenght of its cycles equal to 1 have a synchronizing coloring?

The request of having the gcd of the cycles length equal to 1 is a necessary condition to obtain a synchronizing coloring. Trahtman provided a positive answer to Problem 2 in [111] and Béal and Perrin presented in [23] an algorithm that produces a synchronizing coloring in polynomial time $O(n^2)$. Connected to the road coloring problem there is the problem of finding the so-called *optimal coloring value*, i.e. finding the coloring that minimizes the reset threshold among all the possible synchronizing colorings. Roman [96] proved that there does not exist any polynomial time algorithm that finds the exact optimal coloring value.

3.2.2 Directable NDFAs and partial automata

The notion of synchronization can be generalized to nondeterministic finite automata in several ways; in this manuscript we will focus on the 2-*directability* and the 3-*directability* properties, firstly introduced by Imreh and Steinby in [62]. As for synchronizing DFAs, the initial state q_0 and the set of accepting states $F \subseteq Q$ will not play a role in directable NDFAs, so from now on an NDFA will just be represented by a triple $\langle Q, \Sigma, \delta \rangle$ with Q, Σ, δ as usually defined. We remind that for every $q \in Q$ and $w \in \Sigma^*$, $\delta(q, w)$ is a *subset* of Q.

Definition 21. Let $\mathcal{N} = \langle Q, \Sigma, \delta \rangle$ be an NDFA. We say that $\mathcal{N}$ is

- 2-*directable* if there exists a word w (called a 2-*directing* word) such that for every $p, q \in Q$, $\delta(q, w) = \delta(p, w)$;
- 3-*directable* if there exists $v \in Q$ and a word w (called a 3-*directing word*) such that for every $q \in Q$, $v \in \delta(q, w)$.

Informally speaking, an NDFA is 2-directable if there exists a word w such that the set of states that can be reached by applying the word w is independent on the initial state, while it is 3-directable if there exist a word w and a state that is reachable from any other state by applying the word w.

The 2- and 3-directability properties are decidable ([62], Corollary 3.5). In case of *complete* NDFAs, the 3-directability property is decidable in polynomial time ([63], Proposition 8.3.13) as in this case the square graph criterion (Proposition 3.12) still holds. Indeed, we can define the square graph of a 3-directable NDFA $\mathcal{N}$ as in Definition 15; it then holds true that $\mathcal{N}$ is 3-directable if and only if in its square graph there exists a path from any nonsingleton vertex to a singleton one. We are not aware of any polynomial-time algorithm for checking the 2- and 3-directability properties of a general NDFA, nor of the existence of a polynomial-time algorithm for checking the 2-directability property of a complete NDFA. Directability has also been defined for other kind of automata as probabilistic automata [71] and weighted automata [64]. The notions of 2- and 3-directability and the relatives results have also been extended to fuzzy automata [107].

Similarly to synchronizing DFAs, we are interested in the length of the shortest 2-directing word of a 2-directable NDFA and in the length of the shortest 3-directing word of a 3-directable NDFA.

Definition 22. We define $d_2(\mathcal{N})$ to be the length of the shortest 2-directing word of a 2-directable NDFA $\mathcal{N}$ and $d_3(\mathcal{N})$ to be the length of the shortest 3-directable word of a 3-directable NDFA $\mathcal{N}$. We set, for $i = 2, 3$,

$$d_i(n) = \max\{d_i(\mathcal{N}) : \mathcal{N} \text{ is an } i\text{-directable NDFA on } n \text{ states}\} .$$

Remark 3. Notice that if w is a 2-directing word of an NDFA $\mathcal{N}$ and $\delta(q, w) \neq \emptyset$ for all $q \in Q$, then w is also a 3-directing word for $\mathcal{N}$. We call a word w such that $\delta(q, w) = \emptyset$ for all $q \in Q$ a *killing* word. It follows that, for any NDFA $\mathcal{N}$ that does *not* admit killing words, the 2-directability property implies the 3-directability property and so $d_3(\mathcal{N}) \leq d_2(\mathcal{N})$; this especially holds for any complete NDFA. The concept of killing words is connected with the mortality problem of nonnegative sets of matrices; we will see how in the next section.

The behavior of $d_2(n)$ and $d_3(n)$ is exponential in n; in particular, it holds that:

- $d_2(n) = \Theta(2^n)$ [22, 46];
- $d_3(n) = O(4^{n/3}n^2)$ [46];
- $d_3(n) = \Omega(3^{n/3})$ [75].

These results show how $d_2(\mathcal{N})$ and $d_3(\mathcal{N})$ behave in the *worst* case but they do not give any information about the *average* behavior of an NDFA, i.e. on the typical length of the shortest 2-directing or 3-directing word of a directable NDFA. We will provide results in this direction in Chapter 4.

The concept of synchronization can also be extended to *partial* DFAs, i.e. when some transitions $\delta(q, a)$ of a DFA are not defined:

Definition 23. A partial DFA $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ is *carefully synchronizing* if there exists a word w, called a *carefully synchronizing* word, and a state $p \in Q$ such that for all $q \in Q$, $q.w$ is defined and it is equal to p. We denote with $car(\mathcal{A})$ the length of the shortest carefully synchronizing word of $\mathcal{A}$ and we set $car(n) = \max\{car(\mathcal{A}) : \mathcal{A} \text{ is a carefully synchr. automaton on } n \text{ states}\}$.

Carefully synchronizing words can be much longer than synchronizing words and indeed $car(n)$ has an exponential behavior (see [33, 75, 118]. In particular De Bondt et. al. showed that for any $n \in \mathbb{N}$ there exists a 3-letter carefully synchronizing automaton with shortest carefully synchronizing word of length $\Omega(\phi^{n/3})$ and a 2-letter carefully synchronizing automaton with shortest carefully synchronizing word of length $\Omega(\phi^{n/5})$, where $\phi = (1+\sqrt{5})/2$.

A partial DFA can admit a killing word (see Remark 3 in this section). It is easy to prove that any partial DFA $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ can be equivalently represented by a complete DFA with a sink state $\mathcal{A}_0 = \langle Q \cup \{q_0\}, \Sigma, \delta' \rangle$ by setting for any $q \neq q_0$ and $a \in \Sigma$, $\delta'(q,a) = \delta(q,a)$ if $\delta(q,a)$ is defined, $\delta'(q,a) = q_0$ otherwise, and $\delta(q_0,a) = q_0$. It holds that w is a killing word for the partial DFA $\mathcal{A}$ if and only if w is a reset word for the corresponding DFA $\mathcal{A}_0$ with the sink state. It follows that determining whether a partial DFA $\mathcal{A}$ admits a killing word is decidable in polynomial-time. The result of Rystsov ([98], Theorem 6.1) on 0-automata (or automata with a sink state) also implies that the length of the shortest killing word of a partial DFA on n-states is at most $n(n+1)/2$. Ananichev [5] later showed that if a partial DFA on n state is partially monotone[12] then it admits a killing word of length at most $n + \lfloor (n-1)/2 \rfloor$, and this bound is tight for $n \geq 6$. On the other hand, by the results on synchronization that we have seen in the previous section, there does not exist any polynomial time algorithm that can approximate the length of the shortest killing word of a partial DFA within a constant factor (see Definition 17).

We will see in the next section that partial automata are also linked to primitive sets and that $car(n)$ and $exp(n)$ share the same the growth rate.

3.3 Connecting primitive sets and directable automata

An NDFA whose initial state and accepting states are not specified can be uniquely represented by a set of binary[13] matrices; this is formalized in the following Proposition:

Proposition 3.15. *An NDFA $\mathcal{N} = \langle Q, \Sigma, \delta \rangle$ with $Q = \{q_1, \ldots, q_n\}$ and $\Sigma = \{a_1, \ldots, a_m\}$ can be uniquely represented by the set of binary matrices $\{A_1, \ldots, A_m\}$ where for all $i \in [m]$, $A_i[l,k] = 1$ iff $q_k \in \delta(q_l, a_i)$. The action of a letter a_i on a state q_j is represented by the product $e_j^T A_i$ and the action of a word $a_{i_1} \ldots a_{i_l}$ on a state q_j is represented by the product $e_j^T A_{i_1} \cdots A_{i_l}$. If $\mathcal{N} = \langle Q, \Sigma, \delta \rangle$ is a DFA, then the matrices $\{A_1, \ldots, A_m\}$ are also row-stochastic, i.e. each of them has exactly one 1 in every row. If $\mathcal{N} = \langle Q, \Sigma, \delta \rangle$ is a partial DFA, then each matrix in $\{A_1, \ldots, A_m\}$ has at most one 1 in every row.*

Proof. The matrices $\{A_1, \ldots, A_m\}$ are the adjacency matrices of the letters $\{a_1, \ldots, a_m\}$ in the graph representation of $\mathcal{N}$. □

[12] A deterministic partial automaton $\langle Q, \Sigma, \delta \rangle$ is *monotone* if there is a linear order $\leq$ on Q such that if $q \leq q'$ and $\delta(q,a)$ and $\delta(q',a)$ are defined, then $\delta(q,a) \leq \delta(q',a)$.

[13] We remind that a binary matrix is a matrix having entries in $\{0,1\}$.

From now on we will mostly consider NDFAs and DFAs in their matrix representation, and so they will be mostrly presented as sets of binary matrices. The directability properties of NDFAs and DFAs can be rephrased in terms of properties of the semigroup generated by the matrix set:

Corollary 3.16. *Let $\mathcal{N} = \{A_1, \ldots, A_m\}$ be an NDFA in its matrix representation. Then $\mathcal{N}$ is:*

- *2-directable iff there exists a product $A = A_{j_1} \cdots A_{j_s}$ for some $j_1, \ldots, j_s \in [m]$ such that every column of A is either positive, or entrywise equal to 0;*

- *3-directable iff there exists a product $A_{j_1} \cdots A_{j_s}$ for some $j_1, \ldots, j_s \in [m]$ with a positive column.*

A (partial) DFA $\mathcal{A} = \{A_1, \ldots, A_m\}$ in its matrix representation is (carefully) synchronizing iff it admits a product with a column whose entries are all equal to 1, also called an all-ones column.

It is clear that a primitive set $\mathcal{N}$ of binary matrices is both a 2-directable and a 3-directable NDFA, so it holds that:

$$\max\{d_2(\mathcal{N}), d_3(\mathcal{N})\} \leq exp(\mathcal{N}) \quad \text{and} \quad \max\{d_2(n), d_3(n)\} \leq exp(n)\ . \quad (3.9)$$

Furthermore, it trivially follows by Corollary 3.16 that:

Corollary 3.17. *The 3-directability property and the column-primitivity property for binary matrix sets coincide.*

A less obvious connection between (carefully) synchronizing DFAs and primitive sets is due to the following Definition 24 and Theorem 3.19.

Definition 24. Let $\mathcal{M}$ be a set of binary matrices of size $n \times n$. The *DFA associated to* the set $\mathcal{M}$ is the automaton $Aut(\mathcal{M})$ made of $n \times n$ binary matrices where $A \in Aut(\mathcal{M})$ if and only if there exists $M \in \mathcal{M}$ such that, for all $i \in [n]$[:

a) if $M[i,:] = (0, \ldots, 0)$, then $A[i,:] = (0, \ldots, 0)$.

b) if $M[i,:] \neq (0, \ldots, 0)$, then $A[i,:]$ is a binary stochastic vector[14] such that $A[i,:] \leq M[i,:]$ entrywise.

Notice that if the matrix set $\mathcal{M}$ is NZ, then the associated DFA $Aut(\mathcal{M})$ is complete, i.e. all the transitions are defined and hence all the letters of $Aut(\mathcal{M})$ are binary and row-stochastic. For an example of a set $\mathcal{M}$ and its associated DFA $Aut(\mathcal{M})$ see the following example.

Example 7. Here we present a matrix set $\mathcal{M}$ and its associated partial DFA $Aut(\mathcal{M})$; it is partial because in $\mathcal{M}$ there is a matrix with a zero-row. Their

[14] A binary stochastic vector is simply a vector e_i of the canonical basis.

graph representation can be found in Figure 3.10.

$$\mathcal{M} = \left\{ A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \right\},$$

$$Aut(\mathcal{M}) = \left\{ \underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}}_{a_1}, \underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{a_2}, \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{b_1}, \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}}_{b_2} \right\}.$$

Notice that $a_1, a_2 \leq A$ and $b_1, b_2 \leq B$.

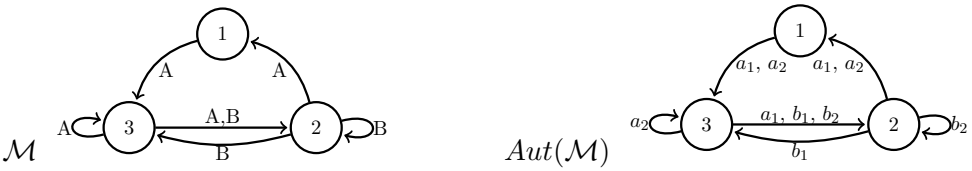


Figure 3.10: The set $\mathcal{M}$ and its associated DFA $Aut(\mathcal{M})$ of Example 7.

A set of binary matrices $\mathcal{M}$ and its associated DFA $Aut(\mathcal{M})$ share the same underlying directed graph but with different labeling, as it is clear by Figure 3.10: indeed, the cardinality of $Aut(\mathcal{M})$ is greater or equal than the cardinality of $\mathcal{M}$, and so is their number of labels. More precisely, if $\mathcal{M} = \{M_1, \ldots, M_m\}$ and we set for all $i \in [m]$ and $j \in [n]$:

$$\begin{cases} s_i^j = |supp(M_i[j,:])| & \text{if } |supp(M_i[j,:])| \geq 1 \\ s_i^j = 1 & \text{otherwise} \end{cases},$$

then the number of matrices (or labels) in $Aut(\mathcal{M})$ is equal to $\sum_{i=1}^{m} \prod_{j=1}^{n} s_i^j$.

The next theorem shows that we can easily build carefully synchronizing automata from primitive sets and that $exp(n)$ and $car(n)$ have the same growth rate.

Theorem 3.18 ([48], Theorem 8)**.**
Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be a primitive set of $n \times n$ binary matrices and let $\mathcal{M}^T = \{M_1^T, \ldots, M_m^T\}$ be the transpose set. Then $Aut(\mathcal{M})$ and $\mathcal{A}(\mathcal{M}^T)$ are carefully synchronizing and it holds that

$$exp(\mathcal{M}) \leq car\big(Aut(\mathcal{M})\big) + car\big(\mathcal{A}(\mathcal{M}^T)\big) + n - 1.$$

Moreover,

$$exp(n) = \Theta\big(car(n)\big) \ .$$

We have already observed that if a set $\mathcal{M}$ is NZ, then $Aut(\mathcal{M})$ is a complete deterministic automaton. The following theorem establishes a stronger connection between the exponent of a primitive NZ-set $\mathcal{M}$ and the reset threshold of $Aut(\mathcal{M})$:

Theorem 3.19 ([17], Theorems 16-17)**.**
Let $\mathcal{M}=\{M_1,\ldots,M_m\}$ be a set of $n\times n$ binary NZ-matrices and let $\mathcal{M}^T=\{M_1^T,\ldots,M_m^T\}$. Then the set $\mathcal{M}$ is primitive if and only if $Aut(\mathcal{M})$ (equivalently $Aut(\mathcal{M}^T)$) is synchronizing. If $\mathcal{M}$ is primitive, it also holds that:

$$\begin{cases}\max\Big\{rt\big(Aut(\mathcal{M})\big),rt\big(Aut(\mathcal{M}^T)\big)\Big\}\leq exp(\mathcal{M})\\ exp(\mathcal{M})\leq rt\big(Aut(\mathcal{M})\big)+rt\big(Aut(\mathcal{M}^T)\big)+n-1\end{cases}. \tag{3.10}$$

The following example reports a primitive NZ-set $\mathcal{M}$ and the synchronizing DFAs $Aut(\mathcal{M})$ and $Aut(\mathcal{M}^T)$ in both their matrix and graph representation.

Example 8. Consider the following primitive NZ-set:

$$\mathcal{M}=\left\{A=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix},B=\begin{pmatrix}1&0&1\\0&0&1\\0&1&0\end{pmatrix}\right\}.$$

The synchronizing DFAs $Aut(\mathcal{M})$ and $Aut(\mathcal{M}^T)$ are the following:

$$Aut(\mathcal{M})=\left\{a=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix},\ b_1=\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix},\ b_2=\begin{pmatrix}0&0&1\\0&0&1\\0&1&0\end{pmatrix}\right\},$$

$$\mathcal{A}(\mathcal{M}^T)=\left\{a=\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix},\ b_1=\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix},\ b_2'=\begin{pmatrix}1&0&0\\0&0&1\\1&0&0\end{pmatrix}\right\}.$$

One can verify that $exp(\mathcal{M})=8$, $rt\big(Aut(\mathcal{M})\big)=4$ and $rt\big(Aut(\mathcal{M}^T)\big)=2$.

Figure 3.11: The DFAs $Aut(\mathcal{M})$ and $Aut(\mathcal{M}^T)$ of Example 8.

Theorems 3.18 and 3.19 more generally hold for any set of matrices with nonnegative entries, due to the fact that the property of being primitive is not influenced by the magnitude of the positive entries of the matrices of the set. In this case we should change point b) of Definition 24 to:

b') if $M[i,:]\neq(0,\ldots,0)$, then $A[i,:]$ is a binary stochastic vector such that for all $i,j\in[n]$, $M[i,j]=0$ implies $A[i,j]=0$.

Equation (3.10) shows that a primitive NZ-set can be used for generating synchronizing DFAs; a primitive set $\mathcal{M}$ with quadratic exponent would imply that one of the DFAs $Aut(\mathcal{M})$ or $Aut(\mathcal{M}^T)$ has quadratic reset threshold. In particular, a primitive NZ-set with exponent greater than $2(n-1)^2+n-1$ would disprove the Černý conjecture.

Remark 4. Theorem 3.19 implies that for checking primitivity of an NZ-set $\mathcal{M}$ we can either use the square graph criterion for NZ-sets on $\mathcal{SG}(\mathcal{M})$ (see

Proposition 3.10) or use the square graph criterion for DFAs on $\mathcal{SG}(Aut(\mathcal{M}))$ (see Proposition 3.12). What is the relation between these two square graphs? It is easy to see that $\mathcal{SG}(Aut(\mathcal{M}))$ is a *subgraph* of $\mathcal{SG}(\mathcal{M})$; the edges of $\mathcal{SG}(\mathcal{M})$ that are missing in $\mathcal{SG}(Aut(\mathcal{M}))$ are exactly all the edges of the kind $(i,i) \xrightarrow{l} (j,k)$ for $j \neq k$ connecting a singleton vertex to a non-singleton one.

We now define the diameter of the square graph of a primitive NZ-set in the same way as done for DFAs (see Definition 18); our goal is to use the diameter of the square graph to find upper and lower bounds on the exponent of a primitive NZ-set, as already established for synchronizing DFAs by Corollary 3.13.

Definition 25. Let $\mathcal{M}$ be a primitive NZ-set and $\mathcal{SG}(\mathcal{M}) = (V, E)$ be its square graph, where $V = \{(i,j) : 1 \leq i \leq j \leq n\}$. We remind that for any $u, v \in V$, $d(u,v)$ denotes the shortest path in $\mathcal{SG}(\mathcal{M})$ from vertex u to vertex v and for any $u \in V$, $d_u = \min\{d(u,v) : v = (q,q) \in V\}$. The *diameter* of the square graph $\mathcal{SG}(\mathcal{M})$ is defined as follows:

$$diam(\mathcal{SG}(\mathcal{M})) = \max_{u=(r,t)\in V:\, r\neq t} d_u \ .$$

We now show that a primitive NZ-set and its associated automaton share the same square graph diameter; we can then make use of Theorem 3.19 to find upper and lower bounds for $exp(\mathcal{M})$.

Proposition 3.20. *Let $\mathcal{M}$ be a primitive NZ-set. It holds that*

$$diam(\mathcal{SG}(\mathcal{M})) = diam(\mathcal{SG}(Aut(\mathcal{M}))). \tag{3.11}$$

Therefore, it also holds that

$$diam(\mathcal{SG}(\mathcal{M})) \leq exp(\mathcal{M}) \leq n(diam(\mathcal{SG}(\mathcal{M})) + diam(\mathcal{SG}(\mathcal{M}^T)) + n - 1. \tag{3.12}$$

Proof. By Remark 4, the square graph $\mathcal{SG}(Aut(\mathcal{M}))$ is a subgraph of $\mathcal{SG}(\mathcal{M})$ so $diam(\mathcal{SG}(Aut(\mathcal{M}))) \leq diam(\mathcal{SG}(\mathcal{M}))$. By the same remark, the edges of $\mathcal{SG}(\mathcal{M})$ that are missing in $\mathcal{SG}(Aut(\mathcal{M}))$ are exactly the edges $(i,i) \xrightarrow{l} (j,k)$ for $j \neq k$ connecting a singleton vertex to a non-singleton one. These kind of edges never belong to the path that realizes the diameter of $\mathcal{SG}(\mathcal{M})$, as it is clear by Definition 25. Therefore, the path that realizes the diameter of $\mathcal{SG}(\mathcal{M})$ also belongs to $\mathcal{SG}(Aut(\mathcal{M}))$ and so $diam(\mathcal{SG}(Aut(\mathcal{M}))) \geq diam(\mathcal{SG}(\mathcal{M}))$.

The upper bound of Equation (3.12) is a straightforward consequence of Equation (3.11), Theorem 3.19 and Corollary 3.13. The lower bound of Equation (3.12) comes from the fact that, if A is a positive product of elements from $\mathcal{M}$ of length $exp(\mathcal{M})$, then in $\mathcal{SG}(\mathcal{M})$ there exists a path labeled by A connecting any non-singleton vertex to any singleton (see Remark 2). Therefore $d_u \leq exp(\mathcal{M})$ for any non-singleton u and so $diam(\mathcal{SG}(\mathcal{M})) \leq exp(\mathcal{M})$. □

We underline the fact that not all the synchronizing DFAs are associated to some primitive NZ-set, i.e. there exist examples of synchronizing DFAs $\mathcal{A}$ such that for every primitive NZ-set $\mathcal{M}$, $\mathcal{A} \neq Aut(\mathcal{M})$, as shown in Example 9. The synchronizing DFAs associated to some primitive sets thus form a special

class of synchronizing automata. On the other hand, every synchronizing DFA $\bar{\mathcal{A}}$ can be turned into a primitive NZ-set $\bar{\mathcal{M}}$ by adding a one in each of its zero-columns: it is not clear yet what would be the relationship between $rt(\bar{\mathcal{A}})$ and $exp(\bar{\mathcal{M}})$ in this case; indeed most of the times it holds that $\bar{\mathcal{A}} \subsetneq Aut(\bar{\mathcal{M}})$.

Example 9. Consider the synchronizing DFA of Figure 3.5; it can be represented by the following matrix set:

$$\mathcal{A} = \left\{ a = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, b = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right\}.$$

We claim that for any primitive NZ-set $\mathcal{M}$, $\mathcal{A} \neq Aut(\mathcal{M})$. Indeed, suppose by contrary that this would be the case and let $\mathcal{M}$ a primitive NZ-set such that $\mathcal{A} = Aut(\mathcal{M})$. There must be in $\mathcal{M}$ an NZ-matrix that dominates a, so there must exist $d_1, d_2, d_3, d_4, d_5, d_6 \in \{0, 1\}$ such that

$$M = \begin{pmatrix} d_1 & 1 & d_4 \\ d_2 & d_5 & 1 \\ d_3 & 1 & d_6 \end{pmatrix} \in \mathcal{M},$$

where $(d_1, d_2, d_3) \neq (0, 0, 0)$ since M is NZ. Suppose that $d_1 = 1$: in view of the definition of the associated DFA of an NZ-set (Definition 24), it must hold that

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \in Aut(\mathcal{M}),$$

which contradicts the hypothesis because this matrix does not belong to $\mathcal{A}$. We get to the same conclusion if we suppose that $d_2 = 1$ or $d_3 = 1$, so the claim is proven.

The following corollary establishes the best upper bound known so far for $exp_{NZ}(n)$.

Corollary 3.21. *Every primitive NZ-set of $n \times n$ matrices has a positive product of length at most* $(15617n^3 + 7500n^2 + 56250n - 78125)/46875$.

Proof. It suffices to apply the inequality $rt(n) \leq (15617n^3 + 7500n^2 + 9375n - 31250)/93750$ of Szykuła [108] to Equation (3.10). □

The Černý conjecture for synchronizing DFAs, if true, would imply a quadratic upper bound for $exp_{NZ}(n)$.

Conjecture 3.22 (Černý conjecture for primitive NZ-sets)**.** *Every primitive NZ-set of $n \times n$ matrices has a positive product of length at most $2n^2 - 3n + 1$.*

We have seen in Theorem 3.18 that $exp(n)$ has the same growth rate of $car(n)$. In case of NZ-sets, Genréncser et. al. [48] proved that $exp_{NZ}(n)$ has the same growth rate of a class of synchronizing DFAs. This class is defined as the set of synchronizing DFAs $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ such that $Q = [n]$ and there exists a partition $\dot{\bigcup}_{i=1}^{k} \Sigma_i$ of Σ having the following properties:

P1 for all $i \in [k]$ and $q \in Q$, there exists a letter $l \in \Sigma_i$ and a state $p \in Q$ such that $p.l = q$;

P2 for all $i \in [k]$ and $q_1, \ldots, q_n \in Q$ such that for all $j \in [n]$, $j.l_j = q_j$ for some $l_j \in \Sigma_i$, there exists a letter $l \in \Sigma_i$ such that for all $j \in [n]$, $j.l = q_j$.

Properties P1 and P2 can appear rather involved; informally speaking, a synchronizing DFA has properties P1 and P2 if for every $i \in [k]$, every state of the automaton is reachable from somewhere by a letter of Σ_i and if, given a list of n transformations of states by letters in Σ_i, it can be found a letter in Σ_i that performs all the transformations at once. Let $rt_{P_1P_2}(n) = \max\{rt(\mathcal{A}) : \mathcal{A}$ is a synchr. DFA on n states having properties P1, P2$\}$. Theorem 8 in [48] states that

$$exp_{NZ}(n) = \Theta(rt_{P_1P_2}(n)) .$$

So far we have seen the connection of directable NDFAs with the primitivity property; we now explore with the following corollary the relationship between the column-primitive property of a binary matrix set and the (carefully) synchronization of its associated automaton. The corollary can be easily proved by using the same techniques developed by Blondel et. al. in the proof of Theorem 3.19, which is not reported in this manuscript.

Corollary 3.23. *Let $\mathcal{M}$ be a finite set of binary matrices. Then the set $\mathcal{M}$ is column-primitive if and only if $Aut(\mathcal{M})$ is carefully synchronizing and it holds that $pc(\mathcal{M}) = car(Aut(\mathcal{M}))$. If $\mathcal{M}$ is also NZ, then $\mathcal{M}$ is column-primitive if and only if $Aut(\mathcal{M})$ is synchronizing and $pc(\mathcal{M}) = rt(Aut(\mathcal{M}))$.*

Proof. Suppose $Aut(\mathcal{M})$ is carefully synchronizing and let $A_{i_1} \cdots A_{i_s}$ one of its carefully synchronizing word of length $car(Aut(\mathcal{M}))$; $A_{i_1} \cdots A_{i_s}$ has a positive column. By Definition 24, for each $A \in Aut(\mathcal{M})$ there exists $M \in \mathcal{M}$ such that $A \leq M$: let $M_{i_1}, \ldots, M_{i_s} \in \mathcal{M}$ such that for all $k \in [s]$, $A_{i_k} \leq M_{i_k}$. This implies that $M_{i_1} \cdots M_{i_s}$ has a positive column so $\mathcal{M}$ is column-primitive and $pc(\mathcal{M}) \leq car(Aut(\mathcal{M}))$.

Suppose now that $\mathcal{M}$ is column-primitive and let $M_{i_1} \cdots M_{i_u}$ be one of the products of length $pc(\mathcal{M})$ with a positive column, say in position l. We claim that for every $r \in [u]$ we can safely set to zero some entries of M_{i_r} in order to make its nonzero rows be row-stochastic while making sure that the final product still has the l-th column positive. In other words, we claim that for every $r \in [u]$ we can select a binary matrix $A_r \leq M_{i_r}$ where the nonzero rows of A_r are row-stochastic and such that $A_1 \cdots A_u[:, l] = (1, \ldots, 1)^T$. If this is true, then by hypothesis A_r belongs to $Aut(\mathcal{M})$ for every r, so $A_1 \cdots A_u$ is a carefully synchronizing word for the automaton $Aut(\mathcal{M})$ and $rt(Aut(\mathcal{M})) \leq u = pc(\mathcal{M})$, which would conclude the proof. We now prove the claim: let D_r be the digraph associated to the matrix M_{i_r} and E_r be its edge set. The fact that $M_{i_1} \cdots M_{i_u}[:, l] = (1, \ldots, 1)^T$ means that for every $j \in [n]$ there exists a sequence of vertices $v_1^j, \ldots, v_{u+1}^j \in [n]$ such that:

$$v_1^j = j \ , \tag{3.13}$$

$$v_{u+1}^j = l \ , \tag{3.14}$$

$$(v_r^j, v_{r+1}^j) \in E_r \quad \forall r \in [u]. \tag{3.15}$$

We can impose an additional property on these sequences: if at step s two sequences share the same vertex, then they have to coincide for all the steps $s' > s$. More formally, if for some $s \in [u]$ we have that $v_s^j = v_s^{j'}$ for $j \neq j'$, then

we set $v^{j'}_{s'} = v^j_{s'}$ for all $s' > s$ as the new sequence $v^{j'}_1, \ldots, v^{j'}_s, v^j_{s+1}, \ldots, v^j_{u+1}$ for vertex j' fulfills all the requirements (3.13), (3.14) and (3.15). We now remove, for every $r \in [u]$, all the edges from E_r that are not of type (3.15); we call this new edge set $\tilde{E}_r$ and let $\tilde{D}_r$ be the subgraph of D_r with edge set $\tilde{E}_r$. Then, for every $r \in [u]$, we set A_r to be the adjacency matrix of $\tilde{D}_r$; by construction A_r has at most one positive entry in each row and $A_r \leq M_{i_r}$, so for all $r \in [u]$, $A_r \in Aut(\mathcal{M})$. Finally, $A_1 \cdots A_u[:, l] = (1, \ldots, 1)^T$ by construction.

Suppose now that the set $\mathcal{M}$ is NZ; then $Aut(\mathcal{M})$ is a complete DFA. We can apply the same argument above to prove that $\mathcal{M}$ is column-primitive if and only if $Aut(\mathcal{M})$ is synchronizing and that $exp(\mathcal{M}) = pc(Aut(\mathcal{M}))$: the only thing that changes is that, when we set to zero the entries of the matrix M_{i_r} to build $A_r \leq M_{i_r}$, some of the rows of A_r could be made of zeros. In this case, since M_{i_r} is NZ, we can safely set to 1 one entry of each of these zero-rows while keeping the property that $A_r \leq M_{i_r}$: A_r is then a binary row-stochastic matrix. □

We conclude this section by remarking that, in view of Proposition 3.15, the problem whether an NDFA admits a killing word is equivalent to the mortality problem for integer nonnegative matrices, so every result that applies to one problem, applies also to the other one. It has been showed that the length of the shortest killing word of an NDFA on n states can be exponential in n ([69], Theorem 16). In the case in which the semigroup generated by the matrices of the NDFA is *finite*, Kiefer and Mascle [72] proved that the mortality problem is decidable in polynomial time and that the shortest killing word has length of order $O(n^5)$.

Chapter 4

Primitivity of random sets

In the previous chapter we have seen that the problem of determining whether a set of matrices is primitive is NP-hard [17] and that their exponent can be exponentially large with respect to the matrix size n [48]. In case of NZ-matrices the problem becomes simpler; primitivity is decidable in polynomial time and a cubic upper bound in n is known for the exponent of any primitive NZ-set of $n \times n$ matrices [17]. Nonetheless, it is still unclear what is the *typical* behavior of a set of matrices: what is the probability to sample a primitive set? What is the expected magnitude of its exponent?

In this chapter we provide answers to these questions for some probability distributions: the contents presented are mainly based on our works [24, 25]. In Section 4.1 we focus on what we call *perturbed permutation sets*, i.e. sets of permutation matrices where a 0-entry of one of the matrices is changed into a 1. We show that a perturbed permutation set generated according to the uniform distribution is primitive and has exponent of order $O(n \log n)$ with high probability; therefore the expected exponent of a perturbed permutation set is much smaller than the known upper bound. This result will set the stage for another randomized generation, presented in Section 4.2: a set $\mathcal{B}_m(n, p)$ of $m \geq 2$ binary $n \times n$ matrices is generated by setting each entry of each matrix to 1 with probability p and to 0 with probability $1 - p$, independently of each others. This construction was inspired by the Erdős-Rényi binomial model on random graphs and its prolific literature (see for example [18, 39, 66] and Appendix A.1), where they investigate the properties of a random graph on n vertices, where an edge between any two vertices appears with probability p. In graph terms, our construction generalizes the Erdös-Rényi model to the extent that we consider *labeled directed* multigraphs where for each vertices $i, j \in [n]$ and label $l \in [m]$, an edge from vertex i to vertex j labeled by l appears with probability p. We show that for any given $c \in \mathbb{R}$, $p = (\log n + c)/n$ is a *sharp threshold* for the property of the random set $\mathcal{B}_m(n, p)$ to be primitive: this informally means that when $p > (\log n + c)/n$ the set $\mathcal{B}_m(n, p)$ is primitive with high probability, when $p < (\log n + c)/n$ the set $\mathcal{B}_m(n, p)$ is *not* primitive with high probability, while when $p = (\log n + c)/n$ it shows an intermediate behavior. We also show that when $p > (\log n + c)/n$, the exponent of $\mathcal{B}_m(n, p)$ is $O(n \log n)$ with high probability, while when $p = (\log n + c)/n$ it is of order $O(n \log^3 n)$ with high probability, under certain conditions. Therefore, also this randomized construction provides primitive sets with small exponent most of the times. In particular, this result implies that with high prob-

ability a binary matrix set generated according to the uniform distribution ($p(n) = 1/2,\ \forall n$) is primitive and has exponent of order $O(n \log n)$. We can thus say that primitive sets with large exponent are few and difficult to find by using a mere uniform random generation, at least for large n.

This threshold result for primitivity finds many applications: it let us prove that $p = (\log n + c)/n$ is as well a sharp threshold for $\mathcal{B}_m(n, p)$ with respect to the property of being *column-primitive* and that for $p > (\log n + c)/n$, $pc(\mathcal{B}_m(n, p))$ and $scr(\mathcal{B}_m(n, p))$ are of order $O(n \log n)$ with high probability. Moreover, by looking at $\mathcal{B}_m(n, p)$ as a random NDFA, we show that $p = (\log n + c)/n$ is a sharp threshold with respect to the property of being 3-*directable* and that for $p > (\log n + c)/n$, $\mathcal{B}_m(n, p)$ is also 2-*directable* with high probability. In particular, we show that an NDFA generated according to the uniform distribution has both a 2-directing word and a 3-directing word of length $O(n \log n)$ with high probability. This last result extends the state of the art on random directable automata initiated by Berlinkov [13] and Nicaud [79]: they showed that with high probability a DFA generated according to the uniform distribution is synchronizing and has a synchronizing word of length $O(n \log^3 n)$. We here show that this also holds for the 2-directability and 3-directability property of random NDFAs.

Inspired again by the random graph theory, in Section 4.3 we explore another randomized construction: we generate sets $\mathcal{B}_m(n, M)$ of m binary matrices where each matrix has exactly $M > 1$ positive entries, according to the uniform distribution. With a similar technique to the one developed for the set $\mathcal{B}_m(n, p)$, we show that for any $c \in \mathbb{R}$, $M = n(\log n + c)$ is a threshold for the property of $\mathcal{B}_m(n, M)$ to be primitive, i.e. that when $M > n(\log n + c)$ the set $\mathcal{B}_m(n, M)$ is primitive and has exponent of order $O(n \log n)$ with high probability, while when $M < n(\log n + c)$ it is not primitive almost surely.

Finally, in Section 4.4 we show that we can bias the generation of perturbed permutation sets in order to find primitive NZ-sets with quadratic exponent. Indeed, in Subsection 4.4.1 we present a randomized algorithm based on the combinatorial characterization theorem of primitive NZ-sets (see Theorem 3.7, Section 3.1.2) that manages to generate *proper*[1] perturbed permutation sets with quadratic exponent. The numerical results of our simulations are reported in Subsection 4.4.2. We then leverage our findings to synchronizing DFAs, by showing that the proper primitive sets found by our algorithm can be easily turned into slowly synchronizing proper DFAs; to the best of our knowledge, this is the first time where a constructive procedure for finding proper synchronizing DFAs is presented. The new families of extremal DFAs found by our procedure are presented in Subsection 4.4.3: they have reset threshold of order $\Omega(n^2/4)$ and they are one of the few examples of slowly synchronizing automata that do not resemble the Černý's family. We also improve the state of the art in the direction initiated by Gonze et. al. in [49] where they proved that the maximal diameter of the square graph (see Definition 18, Section 3.2) among the DFAs on n states and made of $m \geq 2$ permutation matrices is lower bounded by $n^2/4 + o(n^2)$ when n is odd. We prove that this lower bound holds for the maximal diameter among the *synchronizing* DFAs on n states and containing $m \geq 2$ permutation matrices, for all $n \in \mathbb{N}$.

[1] A primitive set is *proper* it needs all its matrices to be primitive.

Finally, our families can be seen as a generalization of the extremal Eulerian DFAs (see Definition 19) presented by Szykuła and Vorel in [109]: we prove that their automata are not minimal and that their construction, which they present just when $n = 4k + 1$ for $k \geq 2$, can be leveraged to any number of states $n \geq 4$ while keeping the automata *pseudo*-Eulerian (see Definition 19).

4.1 Random perturbed permutation sets

In this section we focus on *perturbed permutation* sets, that will be defined in Definition 27, and we show that in the case of uniform distribution these sets are primitive but have small exponent most of the times. This also implies that their associated DFAs (see Definition 24) have almost surely small reset thresholds. We remind that we indicate with $\mathbb{I}_{i,j}$ the matrix having the $[i, j]$-th entry equal to 1 and all the other entries equal to 0.

Definition 26. A *perturbed permutation matrix* is a permutation matrix where one of its 0-entries is changed into a 1. Equivalently, a perturbed permutation matrix is a matrix of the form $\bar{P} = P + \mathbb{I}_{i,j}$, where P is a permutation matrix such that $P[i', j] = 1$ for $i' \neq i$. We denote with $\bar{S}_n = \{\bar{P} : \bar{P} = P + \mathbb{I}_{i,j},\ P \in S_n,\ \exists\, i' \neq i : P[i', j] = 1\}$ the set of the perturbed permutation matrices of size $n \times n$.

Notice that any perturbed permutation matrix can be *uniquely* decomposed in the sum of a permutation matrix plus a matrix of type $\mathbb{I}_{i,j}$.

Definition 27. A *perturbed permutation set* is a finite set made of permutation matrices and one perturbed permutation matrix.

Example 10. The following set is a perturbed permutation set. The matrix $\bar{P}_2$ is a permutation matrix whose $(1, 3)$-th entry has been changed into a 1.

$$\left\{ P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \bar{P}_2 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\}.$$

Perturbed permutation sets are characterized by the fact that they have the least number of positive entries that a primitive NZ-set can have, which intuitively should lead to sets with large exponent. In view of their easy construction, they also appeared to us to be the good starting point to investigate the primitivity property. We will see in details in Section 4.4 that they have other interesting properties that are useful for the search of slowly synchronizing automata: namely, their associated DFAs (see Definition 24) are easily computable and *proper* primitive perturbed permutation sets generate *proper* synchronizing automata (Proposition 4.20).

Definition 28. Let $m, n \geq 2$ integers. A *random perturbed permutation set* $\bar{\mathcal{P}}_m(n)$ is a perturbed permutation set generated by choosing independently $m - 1$ permutation matrices according to the uniform distribution on S_n, and one *perturbed* permutation matrix according on the uniform distribution on $\bar{S}_n$.

In practice, a random perturbed permutation set $\bar{\mathcal{P}}_m(n)$ is equivalently generated by the following randomized procedure:

Procedure 1. (I) m permutation matrices $\{P_1, \ldots, P_m\}$ are sampled independently and uniformly from the set S_n;

(II) A matrix P_i is uniformly chosen from the set $\{P_1, \ldots, P_m\}$ and one of its 0-entries is uniformly selected and changed into a 1. It becomes then a perturbed permutation matrix $\bar{P}_i$;

(III) The final set is the set $\bar{\mathcal{P}}_m(n) = \{P_1, \ldots, P_{i-1}, \bar{P}_i, P_{i+1}, \ldots, P_m\}$.

We remind that we say that a property X holds for a random matrix set with *high probability* if the probability that property X holds tends to 1 as the matrix dimension n tends to infinity. We are now ready to state the main result of this section:

Theorem 4.1. *With high probability a random perturbed permutation set is primitive and has exponent of order $O(n \log n)$.*

The proof of Theorem 4.1 makes use of the following Corollary 4.2, which is a direct consequence of a result of Friedman et al. ([44], Theorem 2.1 and Theorem 2.2). We remind that the *diameter* of a strongly connected directed graph $D = (V, E)$ is equal to $\max_{u,v \in V} d(u, v)$ where $d(u, v)$ is the length of the shortest path connecting u to v.

Corollary 4.2 ([44], Theorem 2.1 and Theorem 2.2)**.** *Let $m \geq 2$ and $r \geq 1$ be two integers and let $\{P_1, \ldots, P_m\}$ be a set of m permutation matrices sampled uniformly and independently at random from S_n. Let D_r be the directed graph with vertex set the set of the r-tuples of distinct elements of $[n]$, having an edge from $(u_1, u_2, \ldots, u_r)$ to $(v_1, v_2, \ldots, v_r)$ if there exists an $i \in [m]$ such that for all $k \in [r]$, $P_i[u_k, v_k] = 1$. Then D_r has diameter of order $O(\log n)$ with high probability.*

Notice that Corollary 4.2 also holds in case some of the matrices P_i are sampled (uniformly) from the set $\bar{S}_n$, as in this case we would just add some more edges to D_r, thus not increasing its diameter.

Proof of Theorem 4.1.
It suffices to prove the theorem for $m = 2$. Let $\bar{\mathcal{P}}_2(n) = \{P_1, \bar{P}_2\}$ be a random perturbed permutation set with $\bar{P}_2 = P_2 + \mathbb{I}_{i,j}$ and let $i' \neq i$ be the integer such that $P_2[i', j] = 1$. Corollary 4.2 with $r = 2$ and $m = 2$ implies that, with high probability, for *any* indices $v_1, v_2, w_1, w_2 \in [n]$ there exists a product Q of elements of $\bar{\mathcal{P}}_2(n)$ of length $O(\log n)$ such that $Q[v_1, w_1] > 0$ and $Q[v_2, w_2] > 0$; we call this property F_2. We now construct a product of elements of $\bar{\mathcal{P}}_2(n)$ whose j-th column is positive; to do so we proceed recursively by constructing at each step a product that has one more positive entry in the j-th column than in the previous step. We will then construct a positive product from it. The matrix $\bar{P}_2$ has two ones in its j-th column; let a_1 and b_1 be two indices such that $\bar{P}_2[a_1, j] = 0$ and $\bar{P}_2[a_1, b_1] = 1$ (they do exist as the matrices are NZ). By property F_2 there exists a product Q_1 of elements in $\bar{\mathcal{P}}_2(n)$ such that $Q_1[j, i] > 0$ and $Q_1[b_1, i'] > 0$; then the product $\bar{P}_2 Q_1 \bar{P}_2 := K_1$ has at least three positive entries in its j-th column. For $s = 2, \ldots, n-2$, let now a_s and

b_s be two indices such that $K_{s-1}[a_s, j] = 0$ and $K_{s-1}[a_s, b_s] > 0$; by property F_2 there exists a product Q_s such that $Q_s[j, i] > 0$ and $Q_s[b_s, i'] > 0$ and so the product $K_{s-1}Q_s\bar{P}_2 := K_s$ has at least $s + 2$ positive entries in its j-th column. It is clear then that K_{n-2} has a positive column in position j; furthermore, as each product Q_i has length $O(\log n)$, K_{n-2} has length $O(n \log n)$. This reasoning can also be applied to the set $\bar{\mathcal{P}}_2^T(n) = \{P_1^T, \bar{P}_2^T\}$ since it is still a perturbed permutation set with $\bar{P}_2^T = P_2^T + \mathbb{I}_{j,i}$: there exist products $T_1, T_2, \ldots, T_{n-2}$ of elements in $\bar{\mathcal{P}}_2^T(n)$ of length $O(\log n)$ such that, by setting $W_1 = \bar{P}_2^T T_1 \bar{P}_2^T$ and $W_s = W_{s-1} T_s \bar{P}_2^T$ for $s = 2, \ldots, n-2$, the final product W_{n-2} has length $O(n \log n)$ and its i-th column is positive. Finally, property F_2 implies that there exists a product S of elements in $\bar{\mathcal{P}}_2(n)$ of length $O(\log n)$ such that $S[j, i] > 0$. Then $K_{n-2} S W_{n-2}^T$ is a positive product of elements in $\bar{\mathcal{P}}_2(n)$ of length $O(n \log n)$. □

Theorem 4.1 shows that a perturbed permutation set generated according the uniform distribution is primitive and has small exponent most of the time. By Theorem 3.19, this also implies that the associated DFA $Aut(\bar{\mathcal{P}}_m(n))$ is synchronizing and has small reset threshold most of the times.

We now present another proof of the fact that a random perturbed permutation set is primitive with high probability: this alternative proof provides a lower bound on the rate of convergence to 1 of the probability that $\bar{\mathcal{P}}_m(n)$ is primitive.

Theorem 4.3. *Let $\bar{\mathbb{P}}$ be the distribution of $\bar{\mathcal{P}}_m(n)$. It holds that:*

$$\bar{\mathbb{P}}(\bar{\mathcal{P}}_m(n) \text{ is primitive }) \geq 1 - \frac{1}{n} - O\left(\frac{1}{n^2}\right). \tag{4.1}$$

To prove Theorem 4.3 we make use of few intermediate results, namely Lemma 4.4, Lemma 4.5, Lemma 4.6, and Theorem 4.7. In the following, we denote with $\mathcal{P}_2(n)$ a random set of two permutation matrices sampled independently and uniformly from S_n and with $\mathbb{P}$ its distribution. Given a random matrix set $\mathcal{M}$, we denote with $\mathcal{M} \in \mathcal{PR}$ the event that $\mathcal{M}$ is primitive, with $\mathcal{M} \in \mathcal{I}$ the event that $\mathcal{M}$ is irreducible and with $\mathcal{M} \in \mathcal{B}ps$ the event that $\mathcal{M}$ has a block-permutation structure (see Definition 11).

Lemma 4.4. *Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be an irreducible set of matrices in which every matrix dominates a permutation matrix. Then, if $\mathcal{M}$ has a block-permutation structure on a partition Ω, all the blocks of Ω must have the same size.*

Proof. Let Q_i be a permutation matrix dominated by M_i; if M_i has a block-permutation structure on a given partition, so does Q_i on the same partition. Theorem 2 in [49] states that if a set of permutation matrices has a block-permutation structure then all the blocks of the partition must have the same size, so we conclude. □

Lemma 4.5. *The probability that $\mathcal{P}_2(n)$ has a block-permutation structure is $O(1/n^2)$.*

Proof. Due to Lemma 4.4, $\mathcal{P}_2(n)$ has a block-permutation structure if and only if it has a block-permutation structure on a partition with blocks of the

same size. If n is a prime number, $[n]$ does not have any nontrivial partition with blocks of the same size, thus the probability that $\mathcal{P}_2(n)$ has a block-permutation structure is 0. Suppose now that n is not prime: let $a \neq 1, n$ be a divisor of n ($a|n$) and $V^{a,n} = \dot{\bigcup}_{i=1}^{a} V_i^{a,n}$ be a partition of $[n]$ of a blocks of size n/a. The probability that a permutation matrix generated according to the uniform distribution in S_n has a block permutation structure on $V^{a,n}$ is equal to

$$\frac{(\frac{n}{a}!)^a a!}{n!}. \tag{4.2}$$

Indeed, the blocks of the partition $V^{a,n}$ can be arranged in $a!$ ways and the positive entries within each of these blocks can be arranged in $\frac{n}{a}!$ ways. It is known that the number of partitions of $[n]$ made of a blocks of size n/a is equal to $n!/(\frac{n}{a}!)^a a!$ (see for example [20]). Let now $\mathcal{V}^{a,n}$ denote the set of all the partitions of $[n]$ made of a blocks of size n/a. In view of Equation (4.2), it holds that:

$$\begin{aligned}\mathbb{P}(\mathcal{P}_2(n) \in \mathcal{B}ps) &\leq \mathbb{P}\left(\bigcup_{a|n} \bigcup_{V \in \mathcal{V}^{a,n}} \{\mathcal{P}_2(n) \text{ has a block-perm. struct. on } V\}\right) \\ &\leq \frac{1}{n!} \sum_{a|n} \left(\frac{n}{a}!\right)^a a! \ , \end{aligned} \tag{4.3}$$

where it is intended that $a \neq 1, n$. We now just need to prove that (4.3) is $O(1/n^2)$. Let $f_n(a) = (\frac{n}{a}!)^a \, a!$ for any divisor a of n; we can extend f_n to the entire real interval $I = [2, n/2]$ in the following way by using the Γ function:

$$f_n^e(x) = \Gamma(nx^{-1} + 1)^x \Gamma(x + 1), \quad x \in [2, n/2] \ , \tag{4.4}$$

where we remind that $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$, $\Gamma \in C^\infty$ and for all $n \in \mathbb{N}$, $\Gamma(n) = (n-1)!$ (see for example [8]). Clearly, $f_n^e(a) = f_n(a)$ for any $a|n$. We claim that f_n^e is convex on I and that there exists $N \in \mathbb{N}$ such that $f_n^e(2) \geq f_n^e(n/2)$ for all $n \geq N$: the proofs of these claims are in Lemma 4.6. By Weierstrass theorem, it follows that $f_n^e(x) \leq f_n^e(2)$ for all $x \in I$ and $n > N$, so it holds that

$$\mathbb{P}(\mathcal{P}_2(n) \in \mathcal{B}ps) \leq \frac{f_n^e(2)}{n!} n. \tag{4.5}$$

Using the Stirling approximation of the Γ function $\Gamma(x+1) \sim \sqrt{2\pi}\, x^{x+\frac{1}{2}} e^{-x}$, we obtain that $\dfrac{f_n^e(2)}{n!} n \sim \sqrt{2\pi}\, n^{\frac{3}{2}} 2^{-n} = O(1/n^2)$. □

We now prove Lemma 4.6 that we used in the above proof; we remind that for any $n \in \mathbb{N}$, we have introduced the function $f_n^e(x)$ defined as follows:

$$f_n^e(x) = \Gamma(nx^{-1} + 1)^x \Gamma(x + 1), \quad x \in [2, n/2] \ , \tag{4.6}$$

where Γ is the Gamma function (see for example [8]).

Lemma 4.6. *Let f_n^e be the function defined in Equation (4.6). Then, for any fixed integer $n \geq 2$, the function f_n^e is convex on the interval $I = [2, n/2]$. Moreover, there exists $N \in \mathbb{N}$ such that for all $n \geq N$, $f_n^e(2) \geq f_n^e(n/2)$.*

Proof. To prove the first statement, we show that the second derivative of $f_n^e(x)$ w.r.t. x is positive for all $x \in I$. We rewrite $f_n^e(x) = e^{G(x)}$ with $G(x) = x\log\left(\Gamma(\frac{n}{x}+1)\right) + \log(\Gamma(x+1))$, so $(f_n^e)''(x) = f_n^e(x)\left((G'(x))^2 + G''(x)\right)$. Since Γ is a positive function on the positive real axis, it suffices to prove that $G''(x) > 0$ on I. The derivative of the Γ function is the so called *digamma* function $\psi(x)$, so it holds that $G''(x) = \frac{n^2}{x^3}\psi'(\frac{n}{x}+1) + \psi'(x+1)$. Since ψ is known to be positive and strictly increasing for $x \geq 2$ (see [8]), we conclude.

The second statement follows from the Stirling approximation:

$$\frac{f_n^e(\frac{n}{2})}{f_n^e(2)} \sim \frac{e^{\frac{n}{2}}2^{n-\frac{1}{2}}}{n^{n+1}} \longrightarrow 0 \text{ as } n \to \infty.$$

□

The last result that we need before proving Theorem 4.3 is the following:

Theorem 4.7 ([35], Theorem 1)**.** *The probability that a random pair of elements sampled independently and uniformly at random from S_n generates a transitive group is equal to:*

$$1 - \frac{1}{n} - \frac{1}{n^2} - \frac{14}{n^3} - \frac{23}{n^4} - \frac{171}{n^5} + O\left(\frac{1}{n^6}\right) . \tag{4.7}$$

We are finally ready to prove Theorem 4.3.

Proof of Theorem 4.3.
We prove it for $m = 2$ without loss of generality. Note that if a perturbed permutation set $\{\bar{P}_1, P_2\}$ with $\bar{P}_1 = P_1 + \mathbb{I}_{i,j}$ has a block-permutation structure on a given partition, so it has the set $\{P_1, P_2\}$ on the the same pertition. On the other hand, if a set of permutation matrices $\{P_1, P_2\}$ is irreducible, so is $\{\bar{P}_1, P_2\}$ for any perturbation $\bar{P}_1$ of P_1. It follows that,

$$\bar{\mathbb{P}}(\bar{\mathcal{P}}_2(n) \in \mathcal{I} \text{ and } \bar{\mathcal{P}}_2(n) \notin \mathcal{B}ps) \geq \mathbb{P}(\mathcal{P}_2(n) \in \mathcal{I} \text{ and } \mathcal{P}_2(n) \notin \mathcal{B}ps), \tag{4.8}$$

and so

$$\begin{aligned} \bar{\mathbb{P}}(\bar{\mathcal{P}}_2(n) \in \mathcal{PR}) &= \bar{\mathbb{P}}(\bar{\mathcal{P}}_2(n) \in \mathcal{I} \text{ and } \bar{\mathcal{P}}_2(n) \notin \mathcal{B}ps) && (4.9)\\ &\geq \mathbb{P}(\mathcal{P}_2(n) \in \mathcal{I} \text{ and } \mathcal{P}_2(n) \notin \mathcal{B}ps) \\ &\geq \mathbb{P}(\mathcal{P}_2(n) \in \mathcal{I}) - \mathbb{P}(\mathcal{P}_2(n) \in \mathcal{B}ps), && (4.10)\end{aligned}$$

where (4.9) holds by Theorem 3.7. Theorem 4.7 implies that the probability that $\mathcal{P}_2(n)$ is irreducible is asymptotically equal to Equation (4.7), so $\mathbb{P}(\mathcal{P}_2(n) \in \mathcal{I}) = 1 - 1/n - O(1/n^2)$. By applying Lemma 4.5 to Equation (4.10), we conclude. □

Remark 5. Theorem 4.3 implies that $\bar{\mathbb{P}}(\bar{\mathcal{P}}_2(n) \text{ is not primitive}) \lesssim \frac{1}{n} + \frac{1}{n^2} + \frac{4}{n^3} + \frac{23}{n^4} + \frac{171}{n^5} + \sqrt{2\pi}\frac{n^{3/2}}{2^n} = g(n)$. It holds that $g(n) \leq 0.05$ for $n \geq 21$, $g(n) \leq 0.01$ for $n \geq 100$ and $g(n) \leq 0.005$ for $n \geq 201$.

4.2 Random sets with independent entries

In the previous section we have studied the random model $\bar{\mathcal{P}}_m(n)$ and we have showed that it is primitive with high probability, with convergence rate of at least $1 - 1/n + O(1/n^2)$. We have also showed that it has exponent of order $O(n \log n)$ with high probability.

In this section we study another random model for binary sets, denoted by $\mathcal{B}_m(n, p)$, which comes as a generalization to matrix sets of the standard *binomial* model in random graph theory. The binomial model was introduced by Erdős and Rényi in their seminal paper [39] in 1946, where they study the properties of a random graph on n vertices where each edge appears with probability $p \in [0, 1]$. The paramenter p is typically considered as a *function* of the number of vertices n. For further information about this random graph model we refer the reader to Appendix A.1 and to [18, 66]. The results of the previous section will play a key role in the discovery of a *threhsold* behavior for the model $\mathcal{B}_m(n, p)$ with respect to the primitivity property and we will prove that this model has small exponent most of the times. In Subsections 4.2.1 and 4.2.2 we then show that $\mathcal{B}_m(n, p)$ admits the same threshold with respect to the 3-directability property and the column-primitivity property (Definitions 9 and 21) and that both the positive-column index and the scrambling index (Definitions 9 and 10) are small most of the times, while a weaker result holds for the 2-directability property. We now introduce the model $\mathcal{B}_m(n, p)$.

Given a property $\mathscr{P}$ and a set $\mathcal{B} = \{B_1, \ldots, B_m\}$ of binary matrices, we write $\mathcal{B} \in \mathscr{P}$ to indicate that the set $\mathcal{B}$ has the property $\mathscr{P}$. Given two binary matrix sets of equal cardinality $\mathcal{B} = \{B_1, \ldots, B_m\}$ and $\mathcal{B}' = \{B'_1, \ldots, B'_m\}$, we say that $\mathcal{B}'$ *dominates* $\mathcal{B}$ ($\mathcal{B}' \geq \mathcal{B}$) if for all $i \in [m]$, it holds that $B'_i \geq B_i$. A property $\mathscr{P}$ is said to be *increasing* if for any matrix sets $\mathcal{B}' \geq \mathcal{B}$, if $\mathcal{B} \in \mathscr{P}$ then $\mathcal{B}' \in \mathscr{P}$. We denote with $B(n, p)$ an $n \times n$ random binary matrix where each entry is independently set to 1 with probability p and to 0 with probability $1 - p$; we denote with $\mathcal{B}_m(n, p) = \{B_1(n, p), \ldots, B_m(n, p)\}$ a set of $m \geq 2$ matrices obtained independently in this way. The parameter p may depend on the matrix size n, so it is to be intended as a sequence of real numbers $p(n) \in [0, 1]$, $n \in \mathbb{N}$; to ease the notation, we will sometimes avoid to explicit the dependency of p on n, so we will write $B(n, p)$ instead of $B(n, p(n))$ and $\mathcal{B}_m(n, p)$ instead of $\mathcal{B}_m(n, p(n))$.

Remark 6. In graph terms, the random matrix $B(n, p)$ can be equivalently seen as a random directed graph on n vertices where, for every $i, j \in [n]$, an edge from vertex i to vertex j appears with probability p. Furthermore, it can also be equivalently seen as a random bipartite graph of bipartitions $V_1 \cup V_2$, $|V_1| = n = |V_2|$, where for every $i \in V_1$ and vertex $j \in V_2$, an edge between i and j appears with probability p. In view of this, we can say that $B(n, p)$ behaves as the well-known random graph models $D(n, p)$ and $G(n, n, p)$ (see Appendix A.1 for further details). The random set $\mathcal{B}_m(n, p)$ can thus be seen as a random *labeled* directed multigraph on n vertices and m labels: for any $i, j \in [n]$ and $l \in [m]$, an edge from vertex i to vertex j labeled by l appears with probability p.

Definition 29. Given an increasing property $\mathscr{P}$, a sequence $\hat{p}(n) \in [0, 1]$, is called a *threshold* for the random binary set $\mathcal{B}_m(n, p)$ *with respect to* $\mathscr{P}$ if, for

any sequence $p(n) \in [0,1]$:

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathscr{P}\Big) = \begin{cases} 1 & \text{if } p \gg \hat{p} \\ 0 & \text{if } p \ll \hat{p} \end{cases},$$

Where we remind that $p \ll \hat{p}$ if and only if $\lim_{n\to\infty} p(n)/\hat{p}(n) = 0$. Furthermore, a sequence $\hat{p}(n) \in [0,1]$ is said to be a *sharp* threshold for the random binary set $\mathcal{B}_m(n,p)$ *with respect to* $\mathscr{P}$ if for any sequence $p(n) \in [0,1]$:

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathscr{P}\Big) = \begin{cases} 1 & \text{if } \exists N, \alpha > 0 : \forall n > N,\ p(n) \geq (1+\alpha)\hat{p}(n) \\ 0 & \text{if } \exists N, \alpha > 0 : \forall n > N,\ p(n) \leq (1-\alpha)\hat{p}(n) \end{cases}.$$

A (sharp) threshold thus represents a *phase transition* for $\mathcal{B}_m(n,p)$ from not having property $\mathscr{P}$ with high probability to having property $\mathscr{P}$ with high probability.

Remark 7. Note that thresholds are in general defined up to the asymptotic relation $\hat{q} = \Theta(\hat{p})$; in other words, if $\hat{p}$ is a threshold, then so is every sequence $\hat{q}(n) \in [0,1]$ for which there exist $C, c > 0$ and $N \in \mathbb{N}$ such that $\forall\, n \geq N$, $c\hat{p}(n) \leq \hat{q}(n) \leq C\hat{p}(n)$. This implies that a threshold is never uniquely defined, despite it is customary to call it *the* threshold (see for example [18, 66]). The same can be said about a *sharp* threshold $\hat{p}$: in this case, any sequence $\hat{q}(n) \in [0,1]$ such that $\lim_{n\to\infty} \hat{q}(n)/\hat{p}(n) = 1$ is as well a sharp threshold.

The primitivity property $\mathcal{PR}$ for a binary matrix set is an increasing property. The following theorem establishes a sharp threshold for $\mathcal{B}_m(n,p)$ with respect to the primitivity property and provides an asymptotic estimate of the expected exponent of $\mathcal{B}_m(n,p)$ when it is an NZ-set.

Theorem 4.8. *Let $m \geq 2$ be an integer, $c \in \mathbb{R}$ and $\hat{p}(n) = (\log n + c)/n$. Then for any sequence $p(n) \in [0,1]$ it holds that:*

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathcal{PR}\Big) = \begin{cases} 1 & \textit{if } \exists N, \alpha > 0 : \forall n > N,\ p(n) \geq (1+\alpha)\hat{p}(n) \\ 0 & \textit{if } \exists N, \alpha > 0 : \forall n > N,\ p(n) \leq (1-\alpha)\hat{p}(n) \end{cases}.$$

In other words, $\hat{p}$ is a sharp threshold *for $\mathcal{B}_m(n,p)$ with respect to the primitivity property. Furthermore, it holds that*

$$a(m,c) \leq \lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,\hat{p}(n)) \in \mathcal{PR}\Big) \leq 1 - \big(1 - e^{-e^{-c}}\big)^m, \tag{4.11}$$

where $a(m,c) = 1 - \big(1 - e^{-2e^{-c}}\big)^m - me^{-2e^{-c}}\big(1 - e^{-2e^{-c}}\big)^{m-1}$. Moreover:

(I) *if $\exists\, N, \alpha > 0 : \forall\, n > N$, $p(n) \geq (1+\alpha)\hat{p}(n)$, then with high probability it holds that*

$$exp\Big(\mathcal{B}_m(n,p(n))\Big) = O(n \log n);$$

(II) *under the condition that $\mathcal{B}_m(n,\hat{p})$ is a primitive NZ-set, with high probability it holds that*

$$exp\Big(\mathcal{B}_m(n,\hat{p}(n))\Big) = O(n \log^3 n).$$

Remark 8. We remind that the primitivity property translates in graph terms as the existence of a sequence of labels such that between any two vertices of the graph there exists a directed path connecting them labeled by that sequence. Theorem 4.8 above thus states that $\hat{p}(n) = (\log n + c)/n$ is a sharp threshold for the random labeled directed multigraph $\mathcal{B}_m(n,p)$ with respect to this property.

Before proving Theorem 4.8 we need four preliminary results, the following Lemma 4.9, Lemma 4.10, Corollary 4.11 and Theorem 4.12, the latter presented by Nicaud in ([79], Theorem 3).

Lemma 4.9. *Let $\mathscr{B}$ be a finite set of $n \times n$ binary matrices such that for all $P, Q \in S_n$, $\mathscr{B} = \{PCQ : C \in \mathscr{B}\} := P\mathscr{B}Q$ and for all $C, D \in \mathscr{B}$, there exist $T_1, T_2 \in S_n$ such that $C = T_1 D T_2$. Let $X_{\mathscr{B}}$ be a random variable with values in $\mathscr{B} \cup \{0\}$, defined in the following way: a random binary matrix $B(n,p)$ is generated, then $X = 0$ if $B(n,p)$ does not dominate any matrix in $\mathscr{B}$, otherwise $X = C$ with C sampled uniformly among the elements of $\mathscr{B}$ dominated by $B(n,p)$. Let $\mathbb{P}_{X_{\mathscr{B}}}$ be the distribution of $X_{\mathscr{B}}$. Then, for any $C, D \in \mathscr{B}$, it holds that:*

$$\mathbb{P}_{X_{\mathscr{B}}}(C) = \mathbb{P}_{X_{\mathscr{B}}}(D). \tag{4.12}$$

Proof. Let $\mathbb{P}$ be the distribution of $B(n,p)$; we write $\mathbb{P}(M)$ for $\mathbb{P}\big(B(n,p) = M\big)$. By definition, for any $C \in \mathscr{B}$, $\mathbb{P}_{X_{\mathscr{B}}}(C) = \sum_{M \geq C} \mathbb{P}(M) |\{C' \in \mathscr{B} : M \geq C'\}|^{-1}$, where M is taken in the set of the binary matrices. Let $C, D \in \mathscr{B}$ and $T_1, T_2 \in S_n$ such that $C = T_1 D T_2$. Observe that $\mathbb{P}(M)$ depends only on the number of positive entries of M so $\mathbb{P}(M) = \mathbb{P}(T_1^{-1} M T_2^{-1})$ as T_1 and T_2 are permutations. It follows that

$$\begin{aligned}
\mathbb{P}_{X_{\mathscr{B}}}(C) &= \sum_{M \geq T_1 D T_2} \mathbb{P}(T_1^{-1} M T_2^{-1}) \, |\{C' \in \mathscr{B} : M \geq C'\}|^{-1} = \\
&= \sum_{T_1^{-1} M T_2^{-1} \geq D} \mathbb{P}(T_1^{-1} M T_2^{-1}) \, |\{C' \in T_1^{-1} \mathscr{B} T_2^{-1} : T_1^{-1} M T_2^{-1} \geq C'\}|^{-1} = \\
&= \mathbb{P}_{X_{\mathscr{B}}}(D).
\end{aligned}$$

□

Remark 9. Both the sets S_n and $\bar{S}_n$ satisfy the hypothesis of Lemma 4.9. The only nontrivial fact to be proven is that for every $C, D \in \bar{S}_n$ there exist $T_1, T_2 \in S_n$ such that $C = T_1 D T_2$. Let $D = P_1 + \mathbb{I}_{i_1, j_1}$ where $P_1 \in S_n$, $P_1[i_1, j_1'] = 1$ for $j_1' \neq j_1$ and $P_1[i_1', j_1] = 1$ for $i_1' \neq i_1$, and let $C = P_2 + \mathbb{I}_{i_2, j_2}$ where $P_2 \in S_n$, $P_2[i_2, j_2'] = 1$ for $j_2' \neq j_2$ and $P_2[i_2', j_2] = 1$ for $i_2' \neq i_2$. Let $\sigma_1, \sigma_2 \in [n]^n$ such that for all $k \in [n]$, $P_1[k, \sigma_1(k)] = 1$ and $P_2[k, \sigma_2(k)] = 1$. Clearly, $\sigma_1(i_1) = j_1'$, $\sigma_1(i_1') = j_1$, $\sigma_2(i_2) = j_2'$ and $\sigma_1(i_2') = j_2$. Finally, let $a = (a_1, \ldots, a_{n-2}) = [n] \setminus \{i_2, i_2'\}$ and $b = (b_1, \ldots, b_{n-2}) = [n] \setminus \{i_1, i_1'\}$. We define the matrices T_1 and T_2 in the following way:

- $T_1[i_2, i_1] = 1$, $T_1[i_2', i_1'] = 1$, for all $k \in [n-2]$, $T_1[a_k, b_k] = 1$;
- $T_2[j_1, j_2] = 1$, $T_2[j_1', j_2'] = 1$, for all $k \in [n-2]$, $T_2[\sigma_1(b_k), \sigma_2(a_k)] = 1$;

It is easy to see that $T_1, T_2 \in S_n$ and that $T_1 D T_2 = C$.

Lemma 4.10. *Let $\mathcal{R}_n$ be the set of all the binary row-stochastic matrices of size $n \times n$. Let $X_{\mathcal{R}_n}$ be a random variable with values in $\mathcal{R}_n$, defined in the following way: a random binary matrix $B(n,p)$ is generated conditioned to the fact that $B(n,p)$ is NZ; then $X = R$ with R sampled uniformly among the elements of $\mathcal{R}_n$ dominated by $B(n,p)$. Let $\mathbb{P}_{X_{\mathcal{R}_n}}$ be the distribution of $X_{\mathcal{R}_n}$. Then $\mathbb{P}_{X_{\mathcal{R}_n}}$ is the uniform distribution on $\mathcal{R}_n$.*

Proof. We show that $\mathbb{P}_{X_{\mathcal{R}_n}}(R)$ does not depend on $R \in \mathcal{R}_n$ and so it must be the uniform distribution. To ease the notation, we write $\mathbb{P}(M)$ for $\mathbb{P}\big(B(n,p) = M \,|\, B(n,p) \text{ is } NZ\big)$. Let $a = (a_1, \dots, a_n)$ be a vector in $[n]^n$; we write that $M = a$ if, for every $i \in [n]$, the i-th row of M has exactly a_i positive entries. Then it holds that,

$$\begin{aligned}\mathbb{P}_{X_{\mathcal{R}_n}}(R) &= \sum_{a_1=1}^{n} \cdots \sum_{a_n=1}^{n} \sum_{\substack{M=a,\\ M \geq R}} \frac{\mathbb{P}(M)}{|\{R' \in \mathcal{R}_n : M \geq R'\}|} \\ &= \sum_{a_1=1}^{n} \cdots \sum_{a_n=1}^{n} \prod_{i=1}^{n} a_i^{-1} \binom{n-1}{a_i - 1} p^{a_i} (1-p)^{n-a_i}.\end{aligned}$$

□

Remark 10. Notice that the above Lemma 4.10 holds as well if we replace $\mathcal{R}_n$ with $\mathcal{C}_n$, the set of all the binary column-stochastic matrices of size $n \times n$, as $\mathcal{C}_n = \{R^T : R \in \mathcal{R}_n\}$.

The following corollary establishes the existence of a sharp threshold for $B(n,p)$ with respect to the property of dominating a permutation matrix; this result will play a significant role in the proof of Theorem 4.1.

Corollary 4.11. *Let $c \in \mathbb{R}$ and $\hat{p}(n) = (\log n + c)/n$. Then $\hat{p}$ is a* sharp threshold *for $B(n,p)$ with respect to the property of dominating a permutation matrix. Furthermore, it holds that*

$$\lim_{n \to \infty} \mathbb{P}\Big(\exists P \in S_n : B(n, \hat{p}(n)) \geq P\Big) = e^{-2e^{-c}} .$$

Proof. It is a straightforward consequence of Proposition A.7 and Theorem A.8 in Appendix A.1, where the connection between random bipartite graphs and $B(n,p)$ is explained and exploited. □

The last result that we need for the proof of Theorem 4.1 is the following theorem, proved by Nicaud in [79]. We will use it to estimate the magnitude of the exponent of $\mathcal{B}_m(n,p)$ when $p = (\log n + c)/n$.

Theorem 4.12 ([79], Theorem 3)**.** *Let $\mathcal{A}$ be a random n-state DFA of $m \geq 2$ letters where each letter is chosen independently and uniformly at random from $\mathcal{R}_n$. Then $\mathcal{A}$ admits a synchronizing word of length $O(n \log^3 n)$ with high probability.*

Remark 11. Theorem 4.12 says that if we uniformly and independently sample $m \geq 2$ binary row-stochastic matrices from $\mathcal{R}_n$, then with high probability there exists a product of length $O(n \log^3 n)$ of these matrices with a positive column. Equivalently, if we uniformly and independently sample $m \geq 2$ binary column-stochastic matrices from $\mathcal{C}_n$, with high probability there exists a product of length $O(n \log^3 n)$ of these matrices with a positive row.

We are finally ready to prove Theorem 4.8.

Proof of Theorem 4.8.
With a slight abuse of notation we denote with $\mathbb{P}$ the distribution of $\mathcal{B}_m(n,p)$. Suppose first that there exists $\alpha>0$ and $N\in\mathbb{N}$ such that $\forall\, n > N$, $p(n) \geq (1+\alpha)\hat{p}(n)$. We need to prove that

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathcal{PR} \text{ and } exp(\mathcal{B}_m(n,p(n))) = O(n\log n)\Big) = 1.$$

Without loss of generality we can just consider the case $m=2$. By Corollary 4.11, we know that $B(n,p)$ dominates a permutation matrix with high probability. We claim that $B(n,p)$ also dominates a perturbed permutation matrix with high probability: indeed this probability is equal to the probability that $B(n,p)$ dominates a permutation matrix minus the probability that $B(n,p)$ *is* a permutation matrix and this latter term is smaller than $\big(np(1-p)^{n-1}\big)^n$, which tends to 0 as n goes to infinity. In view of this, the set $\mathcal{B}_2(n,p)$ dominates a perturbed permutation set with high probability. We now use Theorem 4.1 on the perturbed permutation set dominated by $\mathcal{B}_2(n,p)$. By Remark 9, both the sets S_n and $\bar{S}_n$ satisfy the hypothesis of Lemma 4.9: let $X = X_{S_n}$ and $\bar{X} = X_{\bar{S}_n}$ be the random variables defined in the same lemma. We assume X and $\bar{X}$ to be independent. Lemma 4.9 implies that for every $P\in S_n$ and for every $\bar{P}\in\bar{S}_n$,

$$\mathbb{P}_X(P) = \frac{1-\mathbb{P}_X(0)}{n!} \quad \text{and} \quad \mathbb{P}_{\bar{X}}(\bar{P}) = \frac{1-\mathbb{P}_{\bar{X}}(0)}{n!\,n(n-1)}\ ;$$

indeed one can verify that $|\bar{S}_n| = n!\,n(n-1)$. The fact that $B(n,p)$ dominates a permutation matrix with high probability implies that $\mathbb{P}_X(0)\to 0$ as $n\to+\infty$ and the fact that $B(n,p)$ dominates a perturbed permutation matrix with high probability implies that $\mathbb{P}_{\bar{X}}(0)\to 0$ as $n\to+\infty$. Let $\mathbb{P}_{X\times\bar{X}} = \mathbb{P}_X\cdot\mathbb{P}_{\bar{X}}$ be the joint distribution of X and $\bar{X}$ on $S_n\times\bar{S}_n$ and let $\Omega\subset S_n\times\bar{S}_n$ be the event that a perturbed permutation set of cardinality 2 is primitive and with exponent of order $O(n\log n)$. Since $\mathcal{PR}$ is an increasing property, it holds that:

$$\mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathcal{PR} \text{ and } exp(\mathcal{B}_m(n,p(n))) = O(n\log n)\Big) \geq \mathbb{P}_{X\times\bar{X}}(\Omega) \tag{4.13}$$

and

$$\mathbb{P}_{X\times\bar{X}}(\Omega) = \big(1-\mathbb{P}_X(0)\big)\big(1-\mathbb{P}_{\bar{X}}(0)\big) \sum_{\{P_1,\bar{P}_2\}\in\Omega} (n!)^{-1}\big(n!\,n(n-1)\big)^{-1}. \tag{4.14}$$

The summation in the right-hand side of Equation (4.14) is the probability that the random perturbed permutation set of cardinality two $\bar{\mathcal{P}}_2(n)$ is primitive and with exponent of order $O(n\log n)$, which goes asymptotically to 1 by Theorem 4.1; it follows that Equation (4.14) goes asymptotically to 1. In view of the inequality (4.13), we conclude.

Suppose now that there exist $\alpha>0$ and $N\in\mathbb{N}$ such that $\forall\, n > N$, $p(n) \leq (1-\alpha)\hat{p}(n)$. We need to prove that $\lim_{n\to\infty}\mathbb{P}(\mathcal{B}_m(n,p(n))\in\mathcal{PR}) = 0$. If every matrix of a matrix set has a zero-row, the set cannot be primitive: we show that $\mathcal{B}_m(n,p)$ has this property with high probability. Indeed, the probability

that each matrix of $\mathcal{B}_m(n,p)$ has a zero-row is equal to $(1-(1-(1-p)^n)^n)^m$; by hypothesis $(1-p)^n \to 0$ as $n \to \infty$, so $(1-(1-p)^n)^n \sim e^{-n/e^{pn}}$ that tends asymptotically to 0.

It remains to prove Equation (4.11) and item (II); we start by proving Equation (4.11). The term $1 - \mathbb{P}(\mathcal{B}_m(n,\hat{p}) \in \mathcal{PR}) = \mathbb{P}(\mathcal{B}_m(n,\hat{p}) \notin \mathcal{PR})$ is lower bounded by the probability that each matrix in $\mathcal{B}_m(n,\hat{p})$ has a zero-row, which is equal to $\big(1-\mathbb{P}(B(n,\hat{p}) \text{ has no zero rows})\big)^m$. The probability that $B(n,\hat{p})$ has exactly k zero-rows is a binomial distribution of parameters n and $q(n) = (1-\hat{p}(n))^n$, which converges as $n \to \infty$ to a Poisson distribution of mean $\mu = e^{-c} = \lim_{n\to\infty} nq$. This implies that $\mathbb{P}(B(n,\hat{p}) \text{ has no zero rows})$ converges asymptotically to $e^{-e^{-c}}$, and so

$$1 - \lim_{n\to\infty} \mathbb{P}(\mathcal{B}_m(n,\hat{p}(n)) \in \mathcal{PR}) \geq \big(1 - e^{-e^{-c}}\big)^m,$$

which proves the upper bound in Equation (4.11). For the lower bound, let E be the event that there exist at least two matrices in $\mathcal{B}_m(n,\hat{p})$ such that each of them dominates a permutation matrix; it holds that $\mathbb{P}(\mathcal{B}_m(n,\hat{p}) \in \mathcal{PR}) \geq \mathbb{P}(\mathcal{B}_m(n,\hat{p}) \in \mathcal{PR} \,|\, E)\mathbb{P}(E)$. The term $\mathbb{P}(\mathcal{B}_m(n,\hat{p}) \in \mathcal{PR} \,|\, E)$ tends asymptotically to 1: this can be proved similarly as in the case where $p(n) \geq (1+\alpha)\hat{p}(n)$, by introducing the random variables $X = X_{S_n}$ and $\bar{X} = X_{\bar{S}_n}$ as in Lemma 4.9. The difference is that now $B(n,\hat{p})$ is generated conditioned to the fact that it dominates a permutation matrix so X takes value in S_n; Equation (4.12) still holds and so we can apply Theorem 4.1. It then remains to show that $\mathbb{P}(E)$ tends to $a(m,c)$ as $n \to \infty$; this is straightforward as the probability that $B(n,\hat{p})$ dominates a permutation matrix tends asymptotically to $e^{-2e^{-c}}$ by Corollary 4.11.

Finally, we prove item (II). We can suppose $m = 2$ without loss of generality. We want to prove that

$$\lim_{n\to\infty} \mathbb{P}\Big(exp(\mathcal{B}_2(n,\hat{p}(n))) = O(n^3 \log n) \,\Big|\, \mathcal{B}_2(n,\hat{p}(n)) \in \mathcal{NZ} \cap \mathcal{PR}\Big) = 1.$$

We claim that with high probability $\mathcal{B}_2(n,\hat{p})$, conditioned to the fact that $\mathcal{B}_2(n,\hat{p})$ is NZ, admits a product of length $O(n\log^3 n)$ with a positive column (i.e. $pc(\mathcal{B}_2(n,\hat{p})) = O(n^3 \log n)$) and a product of length $O(n\log^3 n)$ with a positive row. Indeed, let X_r^1 and X_r^2 be two independent random variables such that $X_r^1 = X_{\mathcal{R}_n} = X_r^2$, where $X_{\mathcal{R}_n}$ is the random variable defined in Lemma 4.10, and let $\mathbb{P}_{X_r^1 \times X_r^2} = \mathbb{P}_{X_r^1} \cdot \mathbb{P}_{X_r^2}$ be their joint distribution. Let $E_r \subset \mathcal{R}_n \times \mathcal{R}_n$ be the event that a set of two binary row-stochastic matrices admit a product of length $O(n\log^3 n)$ with a positive column. It holds that

$$\mathbb{P}\Big(pc(\mathcal{B}_2(n,\hat{p})) = O(n^3 \log n) \,\Big|\, \mathcal{B}_2(n,\hat{p}) \in \mathcal{NZ} \cap \mathcal{PR}\Big) \geq \mathbb{P}_{X_r^1 \times X_r^2}(E_r)$$

where $\mathbb{P}_{X_r^1 \times X_r^2}(E_r) \to 1$ as $n \to \infty$ by Theorem 4.12, since by Lemma 4.10 $\mathbb{P}_{X_r^1}$ and $\mathbb{P}_{X_r^2}$ are the uniform distribution on $\mathcal{R}_n$. By a similar reasoning and in view of Remark 11, $\mathcal{B}_2(n,\hat{p})$ admits a product of length $O(n\log^3 n)$ with a positive row with high probability, conditioned to the fact that $\mathcal{B}_2(n,\hat{p}) \in \mathcal{NZ} \cap \mathcal{PR}$. Finally, since $\mathcal{B}_2(n,\hat{p})$ is primitive by hypothesis, it is also irreducible, which means that for all $i,j \in [n]$, it admits a product L_{ij} of length at most $n-1$ with $L_{ij}[i,j] > 0$. Let now C be a product of $\mathcal{B}_2(n,\hat{p})$ of length $O(n\log^3 n)$ with the i-th column positive and R be a product of $\mathcal{B}_2(n,\hat{p})$ of

length $O(n\log^3 n)$ with the j-th row positive: the product $CL_{ij}R$ is a positive product of length $O(n\log^3 n)$ so (II) follows. □

Notice that, since the primitivity property is not influenced by the magnitude of the positive entries of the matrices of the set, Theorem 4.8 is naturally extended to random sets of nonnegative matrices. Furthermore, in the case $p = \hat{p}$, both the left-hand term and the right-hand term of Equation (4.11) approach 1 as the number of matrices m increases, which is reasonable to expect. We underline that the difference in the upper bounds on $exp(\mathcal{B}_m(n,p))$ that we get when $p = \hat{p}$ or when $p \geq (1+\alpha)\hat{p}$ is due to the fact that it is not possible to use the same reasoning in the proof. Indeed, when $p = \hat{p}$ the probability that $B(n,\hat{p})$ dominates a permutation matrix is asymptotically equal to a constant strictly smaller than 1 (Corollary 4.11) and so we cannot make use of Theorem 4.1 anymore. Notice also that the condition that all the matrices of the set are NZ is weaker than requiring that all the matrices of the set dominate a permutation matrix; for an example of NZ-matrices that do not domitate a permutation matrix see Example 14 in Chapter 5.

Corollary 4.13. *Let $c \in \mathbb{R}$, $\hat{p}(n) = (\log n + c)/n$ and $p(n) \in [0,1]$ a sequence such that for some $\alpha > 0$ and for every $n \in \mathbb{N}$, $p(n) \geq (1+\alpha)\hat{p}(n)$. Then, for every $m \geq 2$ integer, it holds that:*

$$\mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathcal{PR}\Big) \geq 1 - \frac{1}{n} - O\left(\frac{n}{e^{np}}\right) - O\left(\frac{1}{n^2}\right) .$$

Proof. It is known that the probability the $B(n,p)$ does not dominate a permutation matrix is of order $O(ne^{-np})$ ([66], Remark 4.3). By Equations (4.13) and (4.14) in the proof of Theorem 4.8 and by Theorem 4.3, it holds that

$$\begin{aligned}\mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathcal{PR}\Big) &\geq \left(1 - O\left(\frac{n}{e^{np}}\right)\right)^2 \bar{\mathbb{P}}(\bar{\mathcal{P}}_m(n) \in \mathcal{PR}) \\ &\geq 1 - \frac{1}{n} - O\left(\frac{n}{e^{np}}\right) - O\left(\frac{1}{n^2}\right) .\end{aligned}$$

□

In Chapter 3 we have seen that $exp(n)$, i.e. the maximal exponent over all the primitive sets of $n \times n$ matrices, is exponential in n: our result shows that the sets whose exponent reaches $exp(n)$ must be very few and that they are almost impossible to be attained by the random model $\mathcal{B}_m(n,p)$ and in particular from a uniform distribution; indeed the *average* exponent is much smaller. In view of the connection between primitive sets and synchronizing DFAs established by Theorem 3.19, Theorem 4.8 also suggests that there is very little hope of generating slowly synchronizing automata from $\mathcal{B}_m(n,p)$, no matter how the function $p(n)$ behaves.

4.2.1 Random NDFAs: 2- and 3-directability

The random binary set $\mathcal{B}_m(n,p)$ can be seen as a random nondeterministic finite state automaton. Indeed, in view of Proposition 3.15, the set $\mathcal{B}_m(n,p)$ represents a random n-states NDFA of m letters where, for every state i and j and every letter a, it holds that $j \in \delta(i,a)$ with probability p. We here apply

Theorem 4.8 to the 2-directability and 3-directability properties of $\mathcal{B}_m(n,p)$. We remind that we use the notation $d_2(\mathcal{N})$ and $d_3(\mathcal{N})$ to indicate, respectively, the shortest 2-directing word and the shortest 3-directing word of a directable NDFA $\mathcal{N}$.

Corollary 4.14. *Let $m \geq 2$ be an integer and $\hat{p}(n) = (\log n + c)/n$ for some $c \in \mathbb{R}$. Let $p(n) \in [0,1]$ be a sequence such that there exist $N, \alpha > 0$ such that for all $n > N$ it holds that $p(n) \geq (1+\alpha)\hat{p}(n)$. Then with high probability the NDFA $\mathcal{B}_m(n,p)$ is 2-directable and $d_2\big(\mathcal{B}_m(n,p)\big) = O(n \log n)$. In particular, for any fixed integer $m \geq 2$, with high probability an m-letter NDFA on n states generated according to the uniform distribution is 2-directable and has a 2-directing word of length $O(n \log n)$.*

Proof. It is a straightforward consequence of the matrix characterization of the 2-directability property (Corollary 3.16), Equation (3.9) and Theorem 4.8. The uniform distribution is obtained by choosing $p(n) = 1/2$, $\forall n \in \mathbb{N}$. □

The following corollary shows that $\hat{p}(n) = (\log n + c)/n$ is as well a sharp threshold for the NDFA $\mathcal{B}_m(n,p)$ with respect to the 3-directability property. We denote the 3-directability property with $\mathscr{D}_3$.

Corollary 4.15. *Let $m \geq 2$ be an integer, $c \in \mathbb{R}$ and $\hat{p}(n) = (\log n + c)/n$. Then for any sequence $p(n) \in [0,1]$ it holds that:*

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathscr{D}_3\Big) = \begin{cases} 1 & \text{if } \exists N, \alpha > 0\colon \forall n > N,\ p(n) \geq (1+\alpha)\hat{p}(n) \\ 0 & \text{if } \exists N, \alpha > 0\colon \forall n > N,\ p(n) \leq (1-\alpha)\hat{p}(n) \end{cases}.$$

In other words, $\hat{p}$ is a sharp threshold *for the NDFA $\mathcal{B}_m(n,p)$ with respect to the 3-directability property. Furthermore, it holds that*

$$a(m,c) \leq \lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,\hat{p}(n)) \in \mathscr{D}_3\Big) \leq 1 - \big(1 - e^{-e^{-c}}\big)^m. \tag{4.15}$$

where $a(m,c) = 1 - \big(1 - e^{-2e^{-c}}\big)^m - me^{-2e^{-c}}\big(1 - e^{-2e^{-c}}\big)^{m-1}$. Moreover:

1. *if $p(n) \in [0,1]$ is such that $\exists\, \alpha, N > 0 : \forall\, n > N,\ p(n) \geq (1+\alpha)\hat{p}(n)$, then with high probability it holds that*

$$d_3\Big(\mathcal{B}_m(n,p(n))\Big) = O(n \log n);$$

2. *under the condition that $\mathcal{B}_m(n,\hat{p})$ is a primitive NZ-set, with high probability it holds that*

$$d_3\Big(\mathcal{B}_m(n,\hat{p}(n))\Big) = O(n \log^3 n).$$

In particular, for any fixed integer $m \geq 2$, with high probability an m-letter NDFA on n states generated according to the uniform distribution is 3-directable and has a 3-directing word of length $O(n \log n)$.

Proof. Suppose that the sequence $p(n)$ is such that there exist $N, \alpha > 0$ such that $\forall\, n > N$, $p(n) \geq (1+\alpha)\hat{p}(n)$. In view of the matrix characterization of the

3-directability property (Corollary 3.16), Equation (3.9) and Theorem 4.8, it holds that

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathscr{D}_3 \text{ and } d_3(\mathcal{B}_m(n,p(n))) = O(n\log n)\Big) = 1.$$

If there exist $N, \alpha > 0$ such that for all $n > N$, $p(n) \leq (1-\alpha)\hat{p}(n)$, then $\lim_{n\to\infty} \mathbb{P}(\mathcal{B}_m(n,p(n)) \in \mathscr{D}_3) = 0$ due to the same argument used in the proof of Theorem 4.8: with high probability all the matrices of $\mathcal{B}_m(n,p)$ have a zero-row, hence they cannot attain a product with a positive column.

Corollary 3.16, Equation (3.9) and Theorem 4.8 also trivially imply the lower bound in Equation (4.15) and item *2.*

It remains to prove the upper bound in Equation (4.15). In the proof of Theorem 4.8 we have seen that the asymptotic probability for $\mathcal{B}_m(n,\hat{p})$ to have each matrix with a zero-row is equal to $(1-e^{-e^{-c}})^m$, in which case $\mathcal{B}_m(n,\hat{p})$ is not 3-directable. Therefore, $\lim_{n\to\infty} \mathbb{P}(\mathcal{B}_m(n,\hat{p}(n)) \in \mathscr{D}_3) \leq 1-(1-e^{-e^{-c}})^m$.

The uniform distribution is obtained by choosing $p(n)=1/2$, $\forall\, n\in\mathbb{N}$. □

Notice again that, for any fixed $c \in \mathbb{R}$, the right-hand term and the left-hand term of Equation (4.16) both tend to 1 as the number of matrices m (the cardinality of the alphabet of the NDFA) increases.

4.2.2 Column-primitivity

We have seen that for a binary matrix set $\mathcal{M}$ the properties of column-primitivity and 3-directability coincide (Corollary 3.17) and so $d_3(\mathcal{M}) = pc(\mathcal{M})$. By Corollary 4.15, it follows that for any $c \in \mathbb{R}$, $\hat{p}(n) = (\log n + c)/n$ is a sharp threshold for $\mathcal{B}_m(n,p)$ with respect to the property of being column-primitive. We reformulate here below Corollary 4.15 in terms of column-primitivity for the sake of completeness, property that we denote with $\mathcal{CP}$. We remind that $scr(\mathcal{M}) \leq pc(\mathcal{M})$ for any column-primitive set $\mathcal{M}$.

Corollary 4.16. *Let $m \geq 2$ be an integer, $c \in \mathbb{R}$ and $\hat{p}(n) = (\log n + c)/n$. Then for any sequence $p(n) \in [0,1]$ it holds that:*

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,p(n)) \in \mathcal{CP}\Big) = \begin{cases} 1 & \text{if } \exists N,\alpha>0\colon \forall n>N,\ p(n) \geq (1+\alpha)\hat{p}(n) \\ 0 & \text{if } \exists N,\alpha>0\colon \forall n>N,\ p(n) \leq (1-\alpha)\hat{p}(n) \end{cases}.$$

In other words, $\hat{p}$ is a sharp threshold *for the random set $\mathcal{B}_m(n,p)$ with respect to the column-primitivity property. Furthermore, it holds that*

$$a(m,c) \leq \lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n,\hat{p}(n)) \in \mathcal{CP}\Big) \leq 1 - \big(1-e^{-e^{-c}}\big)^m.$$

where $a(m,c) = 1-\big(1-e^{-2e^{-c}}\big)^m - me^{-2e^{-c}}\big(1-e^{-2e^{-c}}\big)^{m-1}$. Moreover:

(i) *if $\exists\, N,\alpha>0 : \forall\, n>N$, $p(n) \geq (1+\alpha)\hat{p}(n)$, then with high probability it holds that*

$$pc\Big(\mathcal{B}_m(n,p(n))\Big) = O(n\log n) \quad \textit{and} \quad scr\Big(\mathcal{B}_m(n,p(n))\Big) = O(n\log n).$$

(i) *Under the condition that $\mathcal{B}_m(n,\hat{p})$ is a primitive NZ-set, with high probability it holds that*

$$pc\Big(\mathcal{B}_m(n,\hat{p}(n))\Big) = O(n\log^3 n) \quad \textit{and} \quad scr\Big(\mathcal{B}_m(n,\hat{p}(n))\Big) = O(n\log^3 n).$$

In particular, for any fixed integer $m \geq 2$, a set of m binary $n \times n$ matrices generated according to the uniform distribution has both a scrambling product and a positive-column product of length $O(n \log n)$ with high probability.

Proof. Straightforward by Corollary 4.15 and Corollary 3.17. □

Notice that, since the column-primitivity property is not influenced by the magnitudes of the positive entries of the matrices of the set, Corollary 4.16 is naturally extended to random sets of nonnegative matrices.

4.3 Random sets with fixed number of positive entries

We here consider another random model for matrix sets, where we fix the number of positive entries that each matrix of the set can have. This is again inspired by a standard model in random graph theory, the *uniform* model, introduced by Erdős and Rényi in their seminal paper [39] in 1946: they study the properties of a random graph on n vertices sampled uniformly among the ones having exactly $M > 1$ edges. The paramenter M is typically considered as a *function* of the number of vertices n. For further information about this random graph model we refer the reader to Appendix A.1 and to [18, 66].

We denote with $B(n, M)$ a random binary matrix sampled uniformly among the binary matrices having exactly M positive entries and we indicate with $\mathcal{B}_m(n, M) = \{B_1(n, M), \ldots, B_m(n, M)\}$ a set of $m \geq 2$ binary matrices obtained independently in this way. The parameter $M \in [n^2]$ may depend on the matrix size n, so it is to be intended as a sequence of integers $M(n) \in [n^2]$, $n \in \mathbb{N}$. To ease the notation, we will usually avoid to explicit the dependency of M on n, so we will write $B(n, M)$ for $B(n, M(n))$ and $\mathcal{B}_m(n, M)$ for $\mathcal{B}_m(n, M(n))$. We are interested in the probability of $\mathcal{B}_m(n, M)$ to be primitive as $n \to \infty$ and if it occurs a threshold phenomenon.

Definition 30. Given an increasing property $\mathscr{P}$, a sequence $\hat{M}(n) \in [n^2]$ is called a *threshold* for the random binary set $\mathcal{B}_m(n, M)$ *with respect to* $\mathscr{P}$ if, for any sequence $M(n) \in [n^2]$:

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n, M(n)) \in \mathscr{P}\Big) = \begin{cases} 1 & \text{if } M \gg \hat{M} \\ 0 & \text{if } M \ll \hat{M} \end{cases},$$

where we remind that $M \gg \hat{M}$ if and only if $\lim_{n\to\infty} \hat{M}(n)/M(n) = 0$ (see also Chapter 2).

The following theorem establishes that $\hat{M}(n) = n(\log n + c)$ is a threshold for the random set $\mathcal{B}_m(n, M)$ with respect to the primitivity property.

Theorem 4.17. *Let $m \geq 2$ be an integer, $c \in \mathbb{R}$ and $\hat{M}(n) = n(\log n + c)$. Then for any sequence $M(n) \in [n^2]$ it holds that:*

$$\lim_{n\to\infty} \mathbb{P}\Big(\mathcal{B}_m(n, M(n)) \in \mathcal{PR}\Big) = \begin{cases} 1 & \textit{if } M \gg \hat{M} \\ 0 & \textit{if } M \ll \hat{M} \end{cases}.$$

In other words, $\hat{M}$ is a threshold for the random set $\mathcal{B}_m(n, M)$ with respect to the primitivity property. Furthermore, if $M \gg \hat{M}$, with high probability it holds that:

$$exp\Big(\mathcal{B}_m(n, M(n))\Big) = O(n \log n).$$

Remark 12. In graph terms, the random set $\mathcal{B}_m(n, M)$ represents a random labeled directed multigraph on n vertices and with m labels such that, for each $l \in [m]$, M edges labeled by l are chosen uniformly at random from the set of all the possible sets of M edges. Theorem 4.17 thus establishes that $\hat{M}(n) = n(\log n + c)$ is a threshold for this random labeled directed multigraph with respect to the property of admitting a sequence of labels connecting any two given vertices.

Before proving Theorem 4.17, we need the following Lemma 4.18 and Corollary 4.19.

Lemma 4.18. *Let $\mathscr{B} = S_n$ or $\mathscr{B} = \bar{S}_n$ and let $X_{\mathscr{B}}$ be a random variable with values in $\mathscr{B} \cup \{0\}$, defined in the following way: a random binary matrix $B(n, M)$ with $M > n$ is generated, then $X = 0$ if $B(n, M)$ does not dominate any matrix in $\mathscr{B}$, otherwise $X = C$ with C sampled uniformly among the elements of $\mathscr{B}$ dominated by $B(n, M)$. Let $\mathbb{P}_{X_{\mathscr{B}}}$ be the distribution of $X_{\mathscr{B}}$. For any $C, D \in \mathscr{B}$ it holds that:*

$$\mathbb{P}_{X_{\mathscr{B}}}(C) = \mathbb{P}_{X_{\mathscr{B}}}(D). \tag{4.16}$$

Proof. Let $\mathbb{P}$ be the distribution of $B(n, M)$, that is the uniform distribution over the set of the binary matrices with exactly M positive entries. Given a binary matrix A with M positive entries, we write $\mathbb{P}(A)$ for $\mathbb{P}(B(n, M) = A)$. By Remark 9, it holds that for any $C, D \in \mathscr{B}$, there exists $Q, T \in S_n$ such that $C = QDT$; furthermore, for any $Q, T \in S_n$, $\mathscr{B} = \{QCT : C \in \mathscr{B}\} := Q\mathscr{B}T$. We denote with $|A|$ the number of positive entries of a binary matrix A. By definition,

$$\mathbb{P}_{X_{\mathscr{B}}}(C) = \sum_{\substack{A: |A| = M \\ A \geq C}} \mathbb{P}(A) |\{C' \in \mathscr{B} : A \geq C'\}|^{-1},$$

so it follows that:

$$\begin{aligned}
\mathbb{P}_{X_{\mathscr{B}}}(C) &= \sum_{\substack{A: |A| = M, \\ A \geq QDT}} \mathbb{P}(Q^{-1}AT^{-1}) \, |\{C' \in \mathscr{B} : A \geq C'\}|^{-1} = \\
&= \sum_{\substack{A: |Q^{-1}AT^{-1}| = M, \\ Q^{-1}AT^{-1} \geq D}} \mathbb{P}(Q^{-1}AT^{-1}) \, |\{C' \in Q^{-1}\mathscr{B}T^{-1} : Q^{-1}AT^{-1} \geq C'\}|^{-1} = \\
&= \mathbb{P}_{X_{\mathscr{B}}}(D).
\end{aligned}$$

□

The following corollary establishes the existence of a threshold for $B(n, M)$ with respect to the property of dominating a permutation matrix; this result will play a significant role in the proof of Theorem 4.17.

Corollary 4.19. *Let $c \in \mathbb{R}$ and $\hat{M}(n) = n(\log n + c)$. Then $\hat{M}$ is a* threshold *for $B(n, M)$ with respect to the property of dominating a permutation matrix and with respect to the property of being irreducible.*

Proof. It is a straightforward application of Proposition A.7 and Theorem A.9 in Appendix A.2 when regarding the property of dominating a permutation matrix, and of Theorem A.5 in Appendix A.1 when regarding the irreducibility property, in view of the fact that a binary matrix is irreducible if and only if its associated digraph is strongly connected. □

We are now ready to prove Theorem 4.17.

Proof of Theorem 4.17.
Suppose $M \gg \hat{M}$. We can consider $m = 2$ without loss of generality. We will proceed mimicking the proof of Theorem 4.8: by Corollary 4.19, $B(n, M)$ dominates a permutation matrix with high probability, which implies that it also dominates a perturbed permutation matrix with high probability as $M(n) > n$. This implies that, if we define the random variables $X = X_{S_n}$ and $\bar{X} = X_{\bar{S}_n}$ as in Lemma 4.18, $\mathbb{P}_X(0) \to 0$ and $\mathbb{P}_{\bar{X}}(0) \to 0$ as $n \to +\infty$. Furthermore, by Lemma 4.10, we have that for every $P \in S_n$ and $\bar{P} \in \bar{S}_n$,

$$\mathbb{P}_X(P) = \frac{1 - \mathbb{P}_X(0)}{n!} \quad \text{and} \quad \mathbb{P}_{\bar{X}}(\bar{P}) = \frac{1 - \mathbb{P}_{\bar{X}}(0)}{n!\, n(n-1)}.$$

We consider X and $\bar{X}$ to be independent; let $\mathbb{P}_{X \times \bar{X}} = \mathbb{P}_X \cdot \mathbb{P}_{\bar{X}}$ be their joint distribution on $S_n \times \bar{S}_n$ and let $\Omega \subset S_n \times \bar{S}_n$ be the event that a perturbed permutation set of cardinality 2 is primitive and with exponent of order $O(n \log n)$. Since $\mathcal{PR}$ is an increasing property, by denoting with $\mathbb{P}$ the distribution of $\mathcal{B}_m(n, M)$, it holds that:

$$\mathbb{P}\Big(\mathcal{B}_m(n, M(n)) \in \mathcal{PR} \text{ and } exp(\mathcal{B}_m(n, M(n))) = O(n \log n)\Big) \geq \mathbb{P}_{X \times \bar{X}}(\Omega)\ . \tag{4.17}$$

The right-hand side of Equation (4.17) goes asymptotically to 1 in view of Equation (4.14) and Theorem 4.8.

Suppose now $M \ll \hat{M}$. Let $\mathcal{B}_m(n, M) = \{B_1(n, M), \ldots, B_m(n, M)\}$: since for every $i \in [m]$, $B_i(n, M)$ has exactly M positive entries, the matrix $\sum_i B_i(n, M)$ has at most mM positive entries. Therefore,

$$\mathbb{P}\Big(\mathcal{B}_m(n, \hat{M}(n)) \in \mathcal{PR}\Big) \leq \mathbb{P}\Big(B(n, mM(n)) \text{ is irreducible}\Big). \tag{4.18}$$

The right-hand side of Equation (4.18) goes to 0 as $n \to \infty$ by Corollary 4.19 so we conclude. □

Remark 13. Notice that here we proved that $\hat{M}(n) = n(\log n + c)$ is a threshold for $\mathcal{B}_m(n, M)$ with respect to the property of being primitive, but we did not say anything about whether it is sharp or not. We do believe that this is the case, i.e. that $\mathbb{P}(\mathcal{B}_m(n, \hat{M}) \in \mathcal{PR}) \longrightarrow a \in (0, 1)$ as $n \to \infty$. To achieve this, it could be enough to extend item *2.* of Proposition A.3 (see Appendix A.2) to $\mathcal{B}_m(n, \hat{M})$ and $\mathcal{B}_m(n, p)$, thus showing that the models $\mathcal{B}_m(n, \hat{M})$ and $\mathcal{B}_m(n, p)$ present the same asymptotic behavior when M is close to $n^2 p$. We leave this for future work.

4.4 A more involved randomized generation

Our goal is to find a randomized procedure for generating primitive sets with large exponent, possibly of quadratic order or higher. Perturbed permutation

sets seems to be the ideal candidates, as they have the minimal number of positive entries that a primitive NZ-set can have, that should intuitively lead to larger exponents. Theorem 4.1 showed that if we generate these sets uniformly at random, we expect small exponents; furthermore, it shows that a permutation matrix and a perturbed permutation matrix are usually enough to form a primitive set, so any perturbed permutation set with more than two matrices has redundant elements most of the times. In view of this, we decided to focus on the generation of *proper* perturbed permutation sets:

Definition 31. A primitive set is said to be *proper* if it needs all its matrices to be primitive. Equivalently, a primitive set is proper if and only if by deleting any of its matrices it is no more primitive.

In this section we describe a randomized procedure to build proper primitive sets of any cardinality and we show that it manages to generate primitive sets with quadratic exponent. Moreover, the following proposition shows that we can transform proper primitive perturbed permutation sets in proper synchronizing DFAs:

Proposition 4.20. *Let $\mathcal{M} = \{P_1, \dots, P_{m-1}, P_m + \mathbb{I}_{i,j}\}$ be a proper primitive perturbed permutation set and let $j' \neq j$ be the integer such that $P_m[i, j'] = 1$. The synchronizing automaton $Aut(\mathcal{M})$ (see Definition 24) can be written as $Aut(\mathcal{M}) = \{P_1, \dots, P_{m-1}, P_m, A\}$ with $A = P_m + \mathbb{I}_{i,j} - \mathbb{I}_{i,j'}$. If $Aut(\mathcal{M})$ is not proper, then $\bar{\mathcal{A}} = \{P_1, \dots, P_{m-1}, A\}$ is.*

Proof. Suppose $Aut(\mathcal{M})$ is not proper; the only matrix we can delete from the set without losing synchronization is P_m. Indeed, we cannot delete A as all the other matrices are permutation matrices. For $i = 1, \dots, m-1$, let $\mathcal{M}_i$ be the set obtained from $\mathcal{M}$ by erasing P_i; by hypothesis, $\mathcal{M}_i$ is not primitive so the automaton $Aut(\mathcal{M}_i)$ is not synchronizing. But $Aut(\mathcal{M}_i)$ is indeed the automaton obtained by erasing P_i from $Aut(\mathcal{M})$, so $\bar{\mathcal{A}}$ has to be synchronizing and proper. □

In order to build proper primitive sets we make use of the Protasov-Voynov characterization theorem (Theorem 3.7, Chapter 3), which describes a combinatorial property that an NZ-set must have in order *not* to be primitive: by constructing a primitive set such that each of its proper subsets has this property, we can make it *proper*.

In particular, Theorem 3.7 implies that a primitive set of m matrices is proper if and only if each of its subsets of cardinality $m-1$ has a block-permutation structure on a certain partition; as we are dealing with perturbed permutation sets, these partitions must have blocks of the same size by Proposition 4.4. Given q a divisor of n, we say that a set of $n \times n$ matrices has a *q-permutation structure* if it has a block-permutation structure on a partition made of q blocks of size n/q; we call a partition of this kind a *q-partition*. We will impose to each subset of cardinality $m-1$ of the matrix set to have a q-permutation structure, for some divisor q of n.

The algorithm first generates a set of permutation matrices satisfying the requested block-permutation structures and then a 0-entry of one of the obtained matrices is changed into a 1; while doing this last step, we will make sure to preserve all the block-permutation structures of the matrix. The procedure is described in detail in the following section; we here underline that

it finds perturbed permutation sets that, if are primitive, are also proper; the construction itself does not guarantee primitivity and this property has to be verified at the end.

4.4.1 The algorithm

We remind that, given $R, C \subset [n]$ and a matrix M, we indicate with $M[R, C]$ the submatrix of M with rows indexed by R and columns indexed by C.

For generating a primitive perturbed permutation set $\mathcal{M} = \{M_1, \dots, M_m\}$ of m matrices of size $n \times n$, we choose m prime numbers $q_1 \geq \cdots \geq q_m \geq 2$ and we set $n = \prod_{i=1}^{m} q_i$. For every $j \in [m]$, we require that the set obtained from $\mathcal{M}$ by erasing matrix M_j, that is the set $\{M_1, \dots, M_{j-1}, M_{j+1}, \dots, M_m\}$, to have a q_j-permutation structure; this construction will ensure the set to be proper. More in detail, for all $j \in [m]$ we enforce the existence of a q_j-partition $\Omega^{q_j} = \dot{\bigcup}_{i=1}^{q_j} \Omega_i^{q_j}$ of $[n]$ on which, for all $k \neq j$, the matrix M_k has to have a block-permutation structure. By Definition 11, this implies that for every $k \in [m]$ and for every $j \neq k$ there must exist a permutation $\sigma_j^k \in S_{q_j}$ such that for all $i \in [q_j]$ and $l \neq \sigma_j^k(i)$, $M_k[\Omega_i^{q_j}, \Omega_l^{q_j}]$ is a zero-matrix. The algorithm initializes every entry of each matrix to 1 and then, step by step, it sets to 0 the entries that are not compatible with the conditions that we are requiring. As our final goal is to have a set of permutation matrices with the desired properties, at every step we need to make sure that each matrix dominates at least one permutation matrix, despite the increasing number of zeros among their entries.

Definition 32. Given a matrix M and a q-partition $\Omega^q = \dot{\bigcup}_{i=1}^{q} \Omega_i^q$, we say that a permutation $\sigma \in S_q$ is *compatible* with M and Ω^q if for all $i \in [q]$, there exists a permutation matrix Q_i such that

$$M\left[\Omega_i^q, \Omega_{\sigma(i)}^q\right] \geq Q_i. \tag{4.19}$$

The algorithm itself is formally presented in Algorithm 1; we here describe in words how it operates. Each entry of each matrix is initialized to 1. The algorithm has two for-loops: the outer one on $j = 1, \dots, m$, where a q_j-partition $\Omega^{q_j} = \dot{\bigcup}_{i=1}^{q_j} \Omega_i^{q_j}$ of $[n]$ is uniformly randomly sampled, and the inner one on $k = 1, \dots, m$ with $k \neq j$ where we verify whether there exists a permutation $\sigma_j^k \in S_{q_j}$ that is compatible with M_k and Ω^{q_j}. If it does exist, we choose one among all the compatible permutations and the algorithm moves to the next step $k + 1$. If such permutation does not exist, then the algorithm exits the inner for-loop and it selects another q_j-partition of $[n]$; it then repeats the inner for-loop for $k = 1, \dots, m$ with $k \neq j$ with this new partition. If after $T1$ steps it is choosing a different q_j-partition $\bar{\Omega}^{q_j}$, the existence for each $k \neq j$ of a permutation $\bar{\sigma}_j^k \in S_{q_j}$ that is compatible with M_k and $\bar{\Omega}^{q_j}$ is not established, we stop the algorithm and we say that *it did not converge*. If the inner for-loop is completed, then for each $k \neq j$ the algorithm modifies the matrix M_k by keeping unchanged for each $i = 1, \dots, q_j$ the block $M_k\left[\Omega_i^{q_j}, \Omega_{\sigma_j^k(i)}^{q_j}\right]$ and by setting to zero all the other entries of M_k, where σ_j^k is its selected compatible permutation; the matrix M_k has now a block-permutation structure over the partition Ω^{q_j}. The algorithm then moves to the next step $j + 1$. If it manages to finish the outer for-loop, we have a set of binary matrices with the desired

block-permutation structures. We then just need to select for every $i \in [m]$ a permutation matrix $P_i \leq M_i$ and then to randomly change a 0-entry of one of the matrices $\{P_1, \ldots, P_m\}$ into a 1 without modifying its block-permutation structures: this is always possible as the blocks of the partitions are nontrivial and a permutation matrix has just n positive entries. We finally check whether the obtained perturbed permutation set is primitive. Here below we list the procedures that the algorithm uses:

1. $[p, P] = Extractperm(M, met)$
   This function returns $p = 0$ and $P = M$ if the matrix M does not dominate a permutation matrix, while it returns $p = 1$ and $P \leq M$ if M dominates a permutation matrix P. In this latter case, the matrix P is chosen according to the value of the input met: if $met = 3$ (method 3), we simply use the MatLab function *dmperm(M)*, which finds the Dulmage-Mendelsohn decomposition of a bipartite graph (see Appendix A.2). In particular, $dmperm(M)$ determines whether the binary matrix M dominates a permutation matrix or not, and in the first case it also returns a permutation matrix dominated by M. The function *dmperm* is a deterministic procedure that tends to return the permutation matrix dominated by M that is the closest to the identity matrix: in particular, if $\mathbf{1}$ is the all-ones matrix, then $dmperm(\mathbf{1})$ is indeed the identity matrix. If $met = 2$ (method 2), we still use the MatLab routine *dmperm* but on the matrix $Q_1 M Q_2$, where Q_1 and Q_2 are two permutation matrices sampled uniformly at random from S_n; if $S \leq Q_1 M Q_2$ is the permutation matrix selected by *dmperm*, we set $P = Q_1^T S Q_2^T$. This method has be conceived for making the choice of the permutation matrix $P \leq M$ randomized and possibly close to the uniform distribution. We will see that method 3 will play an important role in our numerical experiments in Subsection 4.4.2 and in the discovery of new families of synchronizing DFAs with quadratic reset threshold in Subsection 4.4.3.

2. $[a, A] = DomPerm(M, \Omega, met)$
   It returns $a = 1$ if there exists a permutation compatible with the matrix M and the partition $\Omega = \dot{\bigcup}_{i=1}^{q} \Omega_i^q$, it returns $a = 0$ and $A = M$ otherwise. In the former case it chooses one of the compatible permutations σ according to met and returns the $n \times n$ matrix A such that $A[\Omega_i^q, \Omega_{\sigma(i)}^q] = M[\Omega_i^q, \Omega_{\sigma(i)}^q]$ for all $i \in [q]$, and all the other entries of A are equal to zero. The matrix A has then a block-permutation structure on Ω. More precisely, $DomPerm$ acts in two steps: it first defines a $q \times q$ matrix B such that, for all $i, k \in [q]$,
   $$B[i,k] = \begin{cases} 1 & \text{if } M[\Omega_i^q, \Omega_k^q] \text{ dominates a permutation matrix} \\ 0 & \text{otherwise} \end{cases};$$
   this is done by calling $ExtractPerm$ with input $M[\Omega_i^q, \Omega_k^q]$ and met for all $i, k \in [q]$. Notice that there exists a permutation compatible with M and Ω if and only if B dominates a permutation matrix. The second step of the procedure is then to call $[p, P] = ExtractPerm(B, met)$ and setting $a = 0$ and $A = M$ if $p = 0$, while if $p = 1$ we set $a = 1$ and A as described before with $\sigma = P$ (i.e. $\sigma(i) = j$ iff $P[i, j] = 1$).

3. $Mset = Addone(P_1, ..., P_m)$

It changes a 0-entry of one of the matrices $P_1, ..., P_m$ into a 1 preserving all its block-permutation structures. The matrix and the entry are chosen uniformly at random and the procedure iterates the choice till it finds a compatible entry (which always exists); it then returns the final perturbed permutation set *Mset*.

4. $pr = Primitive(Mset)$
   It returns $pr = 1$ if the matrix set $Mset = \{M_1, \ldots, M_m\}$ is primitive, $pr = 0$ otherwise. It first verifies if the set is irreducible by checking the strong connectivity of the digraph D_N where $N = \sum_{i=1}^{k} M_i$ via breadth-first search, then primitivity is checked by the Protasov-Voynov algorithm ([94], Section 4).

All the above routines have polynomial time complexity in n, apart from routine *Primitive* that has time complexity of $O(mn^2)$.

Algorithm 4.1: Algorithm for generating proper primitive perturbed permutation sets.

```
Input: q_1,...,q_m, T1, met
Initialize M_1,...,M_m as all-ones matrices
for j:=1 to m do
        t1=0
        while t1 < T1 do
                t1=t1+1
                choose a q_j-partition Omega_j
                for k=1 to m and k!=j do
                    [a,A_k]=DomPerm(M_k,Omega_j,met)
                    if a==0, exit inner for-loop end
                end
                if a==1, exit while-loop end
        end
        if t1==T1
            display 'does not converge', exit procedure
        else
            set M_k=A_k for all k=1,...,m and k!=j
        end
end
for i:=1 to m do
    [p_i,P_i]=Extractperm(M_i,met)
end
Mset=Addone(P_1,...,P_m)
pr=Primitive(Mset)
return Mset, pr
```

Remark 14. 1. In all our numerical experiments the algorithm always converged, i.e. it always ended before reaching the stopping value $T1$, for $T1$ large enough. This is probably due to the fact that the matrix dimension n grows exponentially as the number of matrices m increases, which produces enough degrees of freedom. We leave the proof of this fact for future work.

2. A recent work of Alpin and Alpina ([4], Theorem 3) generalizes Theorem 3.7 for the characterization of primitive sets to sets that are allowed to be reducible and the matrices to have zero columns (but not zero rows). Clearly, DFAs fall within this category. Our algorithm could leverage this result in order to directly construct proper synchronizing DFAs. We also leave this for future work.

Example 11. We here provide an example on how our algorithm works. The entries of the matrices equal to 0 are indicated by a dot.
We set $m = 3$ and $p_1 = p_2 = p_3 = 2$, so $n = 2^3 = 8$; the algorithm will construct a proper primitive perturbed permutation set of 3 matrices M_1, M_2 and M_3 of size 8×8. The matrices are initialized as all-ones matrices:

$$M_1 = \begin{pmatrix} 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1 \end{pmatrix},\ M_2 = \begin{pmatrix} 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1 \end{pmatrix},\ M_3 = \begin{pmatrix} 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1 \end{pmatrix}.$$

j = 1: we choose the partition $\Omega_{q_1} = \{1,2,3,4\} \cup \{5,6,7,8\}$.

- $k = 2$**:** we choose the permutation $\sigma_1^2 = (1,2)$ for M_2 and we check that it is compatible with M_2 and Ω_{q_1}.
- $k = 3$**:** we choose the permutation $\sigma_1^3 = (1)(2)$ for M_3 and we check that it is compatible with M_3 and Ω_{q_1}.

We set to 0 the entries of the blocks in M_2 and M_3 that are not compatible with the block-permutation structures defined by Ω_{q_1} and σ_1^2 and by Ω_{q_1} and σ_1^3, respectively:

$$M_2 = \begin{pmatrix} \cdot&\cdot&\cdot&\cdot&1&1&1&1\\ \cdot&\cdot&\cdot&\cdot&1&1&1&1\\ \cdot&\cdot&\cdot&\cdot&1&1&1&1\\ \cdot&\cdot&\cdot&\cdot&1&1&1&1\\ 1&1&1&1&\cdot&\cdot&\cdot&\cdot\\ 1&1&1&1&\cdot&\cdot&\cdot&\cdot\\ 1&1&1&1&\cdot&\cdot&\cdot&\cdot\\ 1&1&1&1&\cdot&\cdot&\cdot&\cdot \end{pmatrix},\ M_3 = \begin{pmatrix} 1&1&1&1&\cdot&\cdot&\cdot&\cdot\\ 1&1&1&1&\cdot&\cdot&\cdot&\cdot\\ 1&1&1&1&\cdot&\cdot&\cdot&\cdot\\ 1&1&1&1&\cdot&\cdot&\cdot&\cdot\\ \cdot&\cdot&\cdot&\cdot&1&1&1&1\\ \cdot&\cdot&\cdot&\cdot&1&1&1&1\\ \cdot&\cdot&\cdot&\cdot&1&1&1&1\\ \cdot&\cdot&\cdot&\cdot&1&1&1&1 \end{pmatrix}.$$

j = 2: we choose the partition $\Omega_{q_2} = \{1,3,5,7\} \cup \{2,4,6,8\}$.

- $k = 1$**:** we choose the permutation $\sigma_2^1 = (1)(2)$ for M_1 and we check that it is compatible with M_1 and Ω_{q_2}.
- $k = 3$**:** we choose the permutation $\sigma_2^3 = (12)$ for M_3 and we check that it is compatible with M_3 and Ω_{q_2}.

We set to 0 the entries of the blocks in M_1 and M_3 that are not compatible with the block-permutation structures defined by Ω_{q_2} and σ_2^1 and by Ω_{q_2} and σ_2^3, respectively:

$$M_1 = \begin{pmatrix} 1&\cdot&1&\cdot&1&\cdot&1&\cdot\\ \cdot&1&\cdot&1&\cdot&1&\cdot&1\\ 1&\cdot&1&\cdot&1&\cdot&1&\cdot\\ \cdot&1&\cdot&1&\cdot&1&\cdot&1\\ 1&\cdot&1&\cdot&1&\cdot&1&\cdot\\ \cdot&1&\cdot&1&\cdot&1&\cdot&1\\ 1&\cdot&1&\cdot&1&\cdot&1&\cdot\\ \cdot&1&\cdot&1&\cdot&1&\cdot&1 \end{pmatrix},\ M_3 = \begin{pmatrix} \cdot&1&\cdot&1&\cdot&\cdot&\cdot&\cdot\\ 1&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot\\ \cdot&1&\cdot&1&\cdot&\cdot&\cdot&\cdot\\ 1&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot\\ \cdot&\cdot&\cdot&\cdot&1&\cdot&1&\cdot\\ \cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&1\\ \cdot&\cdot&\cdot&\cdot&1&\cdot&1&\cdot\\ \cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&1 \end{pmatrix}.$$

j = 3: we choose the partition $\Omega_{q_3} = \{1,4,5,8\} \cup \{2,3,6,7\}$.

- $k = 1$**:** we choose the permutation $\sigma_3^1 = (12)$ for M_1 and we check that it is compatible with M_1 and Ω_{q_3}.

$k = 2$**:** we choose the permutation $\sigma_3^2 = (1)(2)$ for M_2 and we check that it is compatible with M_2 and Ω_{q_3}.

We set to 0 the entries of the blocks in M_1 and M_2 that are not compatible with the block-permutation structures defined by Ω_{q_3} and σ_3^1 and by Ω_{q_3} and σ_3^2, respectively:

$$M_1 = \begin{pmatrix} \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \end{pmatrix}, \; M_2 = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\ \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

For i = 1, 2, 3: we choose a permutation matrix $P_i \leq M_i$.

$$P_1 = \begin{pmatrix} \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \; P_2 = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \; P_3 = \begin{pmatrix} \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix}.$$

We perturb one of the matrices: we change $P_2[1, 7]$ from 0 to 1.

$$P_1 = \begin{pmatrix} \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \; \bar{P}_2 = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \; P_3 = \begin{pmatrix} \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix}.$$

The set $\{P_1, \bar{P}_2, P_3\}$ is a primitive perturbed permutation set.

4.4.2 Numerical results

We here compare four methods of generating random primitive sets with respect to the magnitude of their exponents. Unfortunately, computing the exponent of thousands of generated instances is prohibitive (computing the exponent is an NP-hard problem [17]), so we use as proxy the diameter of the square graph, which is known to be a lower bound for the exponent (see Proposition 3.20, Equation (3.12)) and to be computable in polynomial time. In view of Proposition 3.20, which establishes the equality between the diameter of the square graph of a primitive set and the diameter of the square graph of its associated DFA, the values that we compute can also be seen as proxies of the reset thresholds of the DFAs associated to the generated primitive sets.

We call **method 1** the sets generated by Procedure 1 (see Section 4.1) with $m = 2$ (i.e. random perturbed permutation sets of two matrices); **method 2** and **method 3**, already introduced in the previous paragraph, refer to our randomized construction where, respectively, a permutation matrix is extracted from a binary one randomly or deterministically. Finally, we call **method 4** a set generated by the following procedure:

Procedure 2.
1. Two permutation matrices P_1 and P_2 are sampled uniformly and independently at random from S_n;
2. A 1-entry of P_1 is selected uniformly at random. Suppose this entry is in row i and column j; we select uniformly an index $\bar{j} \in [n] \setminus \{j\}$ and we set $P_1[i, j] = 0$ and $P_1[i, \bar{j}] = 1$;

3. Let $i' \neq i$ be the other index such that $P_1[i', \bar{j}] = 1$. We select uniformly an index $\bar{i} \in [n] \setminus \{i, i'\}$ and we set $P_1[\bar{i}, j] = 1$.

The matrix P_1 generated by Procedure 2 does *not* dominate a permutation matrix by construction and it has the least number of positive entries that an NZ-matrix that does not dominate a permutation matrix can have. Procedure 2 has been developed because Theorem 4.1 and Theorem 4.8 show that, when all the matrices of the set dominate a permutation matrix and the underlying distribution is «simple», we expect small exponents with high probability.

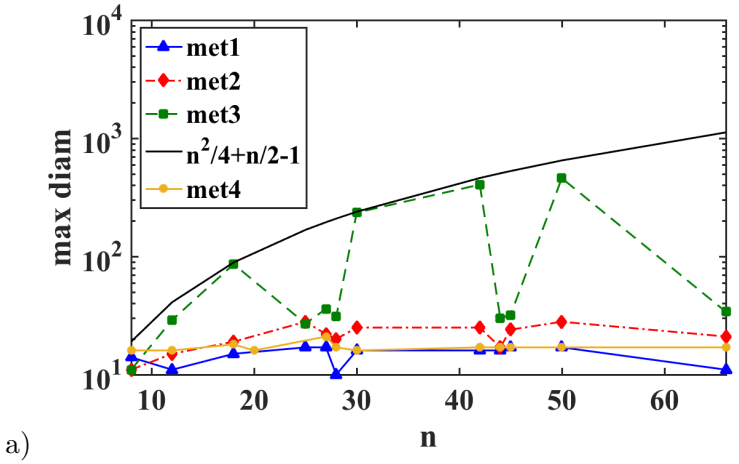
For each method and each choice of n we run the algorithm $it(n) = 50n^2$ times, thus producing each time $50n^2$ sets. This choice for $it(n)$ has been made by taking into account two facts: on the one hand, it is desirable to keep constant the rate $it(n)/k_m(n)$ between the number of sampled sets $it(n)$ and the cardinality $k_m(n)$ of the state space. Since $k_m(n+1)/k_m(n)$ grows approximately as n^m, $k_m(n)$ explodes very fast, so we also have to deal with the limited computational speed of our computers. The choice of $it(n) = 50n^2$ comes as a compromise between these two issues. Among the $it(n)$ generated sets, we select the primitive ones and we compute their square graph diameters. We set $T1 = 1000$ for method 2 and 3.

Figure 4.1 a) reports on the y axis the maximal square graph diameter found among the primitive sets generated by methods 1, 2, 3 and 4 for each matrix dimension n, when n is the product of three prime numbers. Figure 4.1 b) reports the same when n is the product of four prime numbers. We can see that our randomized construction manages to reach higher values of the square graph diameter than methods 1 and 4; in particular, method 3 reaches quadratic diameters in case of three matrices.

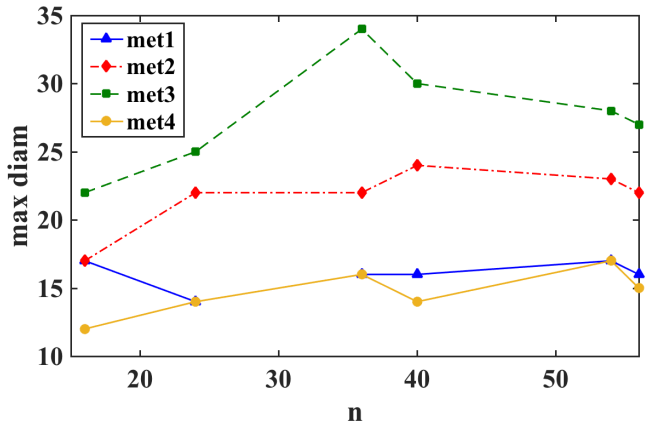
We also report in Figure 4.1 c) the behavior of the *average* diameter of the proper primitive sets generated on $50n^2$ iterations when n is the product of three prime numbers: we can see that in this case method 2 does not perform better than method 1 and 4, while method 3 performs just slightly better. This behavior reflects the fact that primitive sets with quadratic exponent are hard to find (see Theorems 4.1 and 4.8) and so, even when we manage to randomly generate them, we expect to generate just *few* of them.

Figure 4.2 reports the percentage of the generated sets that are *not* primitive, where we divide nonprimitive sets into two categories: reducible sets and *imprimitive* sets, i.e. irreducible sets that are not primitive. We can see that the percentage of nonprimitive sets generated by method 1 and 4 approaches 0 as n increases, behavior that we partially expected (see Theorem 4.1, Section 4.1), while method 2 seems to always produce a non-negligible percentage of nonprimitive sets, although quite small. The behavior is reversed for method 3: most of the generated sets are not primitive, which can be interpreted as a good sign. Indeed, nonprimitive sets can be seen as sets with *infinite* exponent; as we are generating a lot of them with method 3, we intuitively should expect that, when a primitive set is generated, it has high chances to have large diameter.

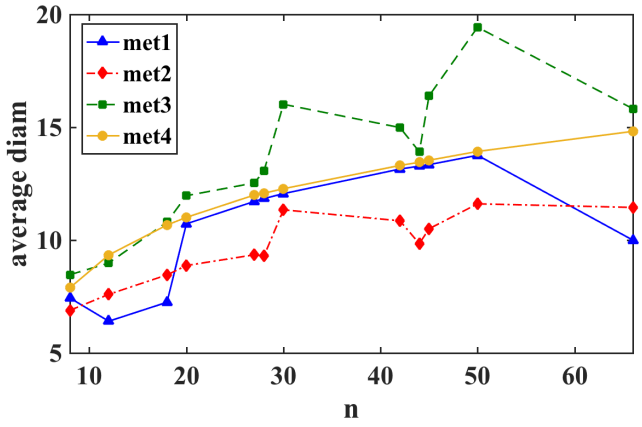
In the following section we present the slowly synchronizing automata found by our randomized construction.



a)



b)



c)

Figure 4.1: a) Comparison between methods 1, 2, 3 and 4 with respect to the maximal square graph diameter found on $50n^2$ iterations when n is the product of three prime numbers; the y axis is in logarithmic scale.
b) Comparison between methods 1, 2, 3 and 4 with respect to the maximal square graph diameter found on $50n^2$ iterations when n is the product of four prime numbers.
c) Average square graph diameter obtained by methods 1, 2, 3 and 4 when n is the product of three prime numbers.

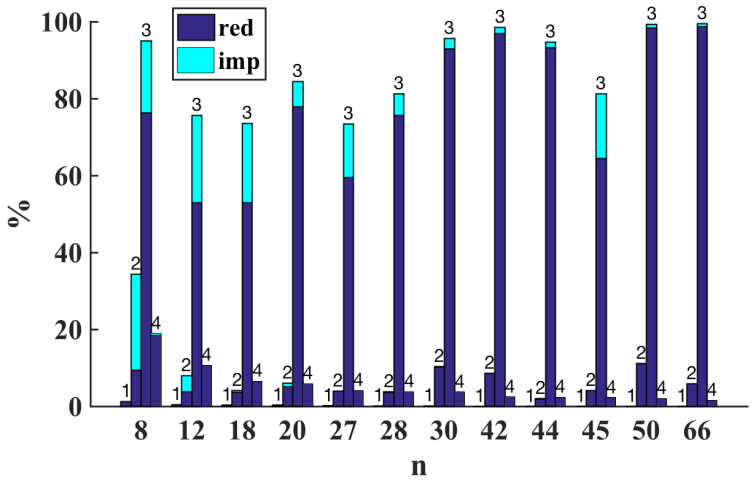


Figure 4.2: Percentage of nonprimitive sets (divided into reducible and imprimitive sets) generated by methods 1, 2, 3 and 4 (indicated above each bar) when n is the product of three prime numbers. For instance, on sets of dimension $n = 20$, method 1 generates 0.35% of nonprimitive sets (0.35% reducible, 0% imprimitive), method 2 generates 6.15% of nonprimitive sets (5.18% reducible, 0.97% imprimitive), method 3 generates 84.5% of nonprimitive sets (77.9% reducible, 6.6% imprimitive) and method 4 generates 5.88% of nonprimitive sets (5.88% reducible, 0% imprimitive).

4.4.3 New families of slowly synchronizing automata

In the previous section we have seen that we managed to generate proper primitive sets with quadratic square graph diameter, which implies that their associated synchronizing DFAs have quadratic square graph diameter as well, and hence quadratic reset threshold.

We present in this section these new families of slowly synchronizing automata and we show that their reset threshold is of order $\Omega(n^2/4)$. Our automata are proper and are made of three letters: two symmetric permutation matrices and a matrix that fixes all the states but one, which make them belong to the class of automata *with simple idempotents* (see Definition 19). This characteristic makes also our families differ from the Černý family and the other known families of extremal automata (e.g. [7, 33, 37, 55, 73]) to the extent that, if we set $r(\mathcal{A}) = \min\{k \in \mathbb{N} : a^k = a,\ \forall\, a \in \Sigma\}$ where Σ is the alphabet of the DFA $\mathcal{A}$, then $r(\mathcal{A}) = 3$ for any automaton $\mathcal{A}$ that belongs to our families while $r(\mathscr{C}_n) = n + 1$ for the Černý automaton $\mathscr{C}_n$ on n states[2].

Our automata are the associated DFAs of proper primitive sets made of a perturbed identity matrix and two symmetric permutations. The following proposition shows that primitive sets of this kind must have a very specific shape; we then present our families, prove closed formulas for their square graph diameter and finally state a conjecture on their reset thresholds. With a slight abuse of notation we identify a permutation matrix Q with its underlying permutation, that is we say that $Q(i) = j$ if and only if $Q[i, j] = 1$; the identity matrix is denoted by I. Note that a permutation matrix is symmetric if and

[2] And similarly $r(\mathcal{A})$ is linear in n for most of the known extremal automata.

only if its cycle decomposition is made of fixed points and cycles of length 2.

Proposition 4.21. *Let $\mathcal{M}_{i,j} = \{\bar{I}_{i,j}, Q_1, Q_2\}$ be a matrix set of $n \times n$ matrices where $\bar{I}_{i,j} = I + \mathbb{I}_{i,j}$, $j \neq i$, is a perturbed identity and Q_1 and Q_2 are two symmetric permutations. If $\mathcal{M}$ is irreducible then, up to a relabeling of the vertices, Q_1 and Q_2 have the following form:*
- if n is even

$$Q_1(i) = \begin{cases} 1 & \text{if } i = 1 \\ i+1 & \text{if } i \text{ even}, 2 \leq i \leq n-2 \\ i-1 & \text{if } i \text{ odd}, 3 \leq i \leq n-1 \\ n & \text{if } i = n \end{cases}, \quad Q_2(i) = \begin{cases} i-1 & \text{if } i \text{ even} \\ i+1 & \text{if } i \text{ odd} \end{cases} \tag{4.20}$$

or

$$Q_1(i) = \begin{cases} n & \text{if } i = 1 \\ i+1 & \text{if } i \text{ even}, 2 \leq i \leq n-2 \\ i-1 & \text{if } i \text{ odd}, 3 \leq i \leq n-1 \\ 1 & \text{if } i = n \end{cases}, \quad Q_2(i) = \begin{cases} i-1 & \text{if } i \text{ even} \\ i+1 & \text{if } i \text{ odd} \end{cases} \tag{4.21}$$

- if n is odd

$$Q_1(i) = \begin{cases} 1 & \text{if } i = 1 \\ i+1 & \text{if } i \text{ even} \\ i-1 & \text{if } i \text{ odd}, 3 \leq i \leq n \end{cases}, \quad Q_2(i) = \begin{cases} i-1 & \text{if } i \text{ even} \\ i+1 & \text{if } i \text{ odd}, 1 \leq i \leq n-2 \\ n & \text{if } i = n \end{cases}. \tag{4.22}$$

Proof. The set $\mathcal{M}$ is irreducible if and only if the digraph D induced by the matrix $\bar{I}_{i,j} + Q_1 + Q_2$ is strongly connected (see Proposition 3.3). If D is strongly connected, then the digraph induced by the matrix $Q_1 + Q_2$ must be strongly connected as Q_1 and Q_2 are symmetric and the matrix $\bar{I}_{i,j}$ adds just a single edge that is not a selfloop in D. Consider vertex 1: there must exist a matrix in the set $\{Q_1, Q_2\}$ that links it to another vertex; let this matrix be Q_2 (without loss of generality) and label this vertex with 2. As Q_2 is symmetric, we have $Q_2(1) = 2$ and $Q_2(2) = 1$. This implies that Q_1 needs to link vertex 2 to some vertex other than 1 as otherwise the digraph would not be strongly connected; we label this vertex with 3 and so we have $Q_1(2) = 3$ and $Q_1(3) = 2$. By iterating this reasoning, it follows that Q_1 and Q_2 must be as in (4.20) or (4.21) if n is even or as in (4.22) if n is odd. □

Proposition 4.22. *A matrix set $\mathcal{M}_{i,j} = \{\bar{I}_{i,j}, Q_1, Q_2\}$ of type (4.21) is never primitive.*

Proof. Due to the symmetry of the digraph $D_{Q_1+Q_2}$, up to a relabeling of the vertices we can assume without loss of generality that $i = 1$. If j is odd, all the three matrices have a block-permutation structure over the partition $\{\{1, 3, \ldots, n-1\}, \{2, 4, \ldots, n\}\}$, while if j is even they have a block-permutation structure over the partition $\{\{1, j\}, \{2, j-1\}, \ldots, \{\frac{j}{2}, \frac{j}{2}+1\}, \{j+1, n\}, \{j+2, n-1\}, \ldots \ \ldots, \{\frac{n+j}{2}, \frac{n+j}{2}+1\}\}$. By Theorem 3.7, the set cannot be primitive. □

Definition 33. We define $\mathcal{A}_{i,j} = Aut(\mathcal{M}_{i,j})$, the associated DFA of the set $\mathcal{M}_{i,j}$ (see Definition 24). The DFA $\mathcal{A}_{i,j}$ can be written as $\mathcal{A}_{i,j} = \{\underline{I}_{i,j}, Q_1, Q_2\}$, where $\underline{I}_{i,j} = I + \mathbb{I}_{i,j} - \mathbb{I}_{i,i}$.

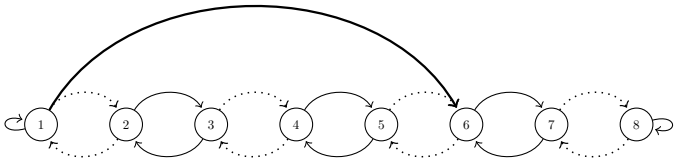


Figure 4.3: The automata $\mathcal{A}_{1,6}$ for $n = 8$; $rt(\mathcal{A}_{1,6})$=31. Dashed arrows refer to matrix Q_2, normal arrows to matrix Q_1 and bold arrows to matrix $\underline{I}_{1,6}$, where its selfloops have been omitted.

Figure 4.3 represents the automaton $\mathcal{A}_{1,6}$ for $n = 8$.
We set:

- $\mathcal{E}_n = \mathcal{A}_{1,n-2}$, for $n = 4k$ and $k \geq 2$;
- $\mathcal{E}'_n = \mathcal{A}_{1,n-4}$, for $n{=}4k + 2$ and $k \geq 2$;
- $\mathcal{O}_n = \mathcal{A}_{\frac{n-1}{2},\frac{n+1}{2}}$, for $n = 4k + 1$ and $k \geq 1$;
- $\mathcal{O}'_n = \mathcal{A}_{\frac{n-1}{2},\frac{n+1}{2}}$, for $n = 4k + 3$ and $k \geq 1$.

Theorem 4.23. *The DFAs $\mathcal{E}_n$, $\mathcal{E}_n$, $\mathcal{O}_n$ and $\mathcal{O}'_n$ are synchronizing and it holds that:*

1. *$\mathcal{E}_n$ has square graph diameter of $(n^2 + 2n - 4)/4$;*
2. *$\mathcal{E}'_n$ has square graph diameter of $(n^2 + 2n - 12)/4$;*
3. *$\mathcal{O}_n$ has square graph diameter of $(n^2 + 3n - 8)/4$;*
4. *$\mathcal{O}'_n$ has square graph diameter of $(n^2 + 3n - 6)/4$.*

Therefore $\mathcal{E}_n$, $\mathcal{E}'_n$, $\mathcal{O}_n$ and $\mathcal{O}'_n$ have reset threshold of order $\Omega(n^2/4)$.

Proof. We prove the theorem just for the family $\mathcal{E}_n$; the results for the other families can be obtained by similar reasonings. In the following we describe the shape of $\mathcal{SG}(\mathcal{A}_{1,n-2})$ for $n = 4k$ in order to compute its diameter, which in this case is the maximal distance between a non-singleton vertex and the singleton $(n - 2, n - 2)$. The synchronization of $\mathcal{E}_n$ will be assured by the fact that in $\mathcal{SG}(\mathcal{A}_{1,n-2})$ any non-singleton vertex is connected to the singleton $(n-2, n-2)$, by making use of Proposition 3.12. We invite the reader to refer to Figure 4.4, situated at the end of this section, during the proof.

We set $\underline{I} = \underline{I}_{1,n-2}$ to ease the notation. The digraph $S(\mathcal{A}_{1,n-2} \setminus \{\underline{I}\})$, without considering the singletons, is disconnected and has $n/2$ strongly connected components that we denote with $C_0, C_1, \ldots, C_{n/2-1}$. The component C_0 is made of the vertices $\{(s, n - s + 1) : s \in [n/2]\}$, while for $i \in [n/2 - 1]$,

$$C_i = \{(s, 2i{-}s{+}1) : s \in [i]\} \cup \{(s, 2i{+}s) : s \in [n{-}2i]\} \cup \{(n{-}2i{+}s, n{-}s{+}1) : s \in [i]\}.$$

Notice that C_0 has size $n/2$, while for every $i \in [n/2 - 1]$, C_i has size n. The components $C_0, C_1, \ldots, C_{n/2-1}$ look like "chains" due to the symmetry of Q_1 and Q_2 (see Figure 4.4). Therefore, the vertices $(1, n)$ and $(3, n - 2)$ belong to C_0, the vertices $(1, 2i)$ and $(1, 2i + 1)$ belong to C_i for $1 \leq i \leq n/2{-}1$, the

vertices $(n-4, n-2)$ and $(n-2, n)$ belong to C_1, the vertices $(1, n-2)$ and $(5, n-2)$ belong to $C_{n/2-1}$ and the vertices $(n-2i-2, n-2)$ and $(n-2i+3, n-2)$ belong to C_i for $2 \leq i \leq n/2-2$. The matrix $\underline{I}$ connects the components $\{C_i\}_i$ by linking, for every $a = 2, \ldots, n$, vertex $(1, a)$ to vertex $(a, n-2)$ in such a way that the $\{C_i\}_i$ can be ordered from the farthest to the closest to the singleton $(n-2, n-2)$ (see Figure 4.4). Indeed, the diagram in Figure 4.5 (situated at the end of this section) shows how the components $\{C_i\}_i$ are linked together for $2 \leq i \leq n/2-1$: an arrow between two vertices means that there exists a word mapping the first vertex to the second one, a number next to the arrow represents the length of such word if the two vertices belong to the same component while arrows connecting vertices from different components are labeled by $\underline{I}$; bold vertices represent the ones that are linked by $\underline{I}$ to other chains. How C_0 is connected to C_1 is directly shown in Figure 4.4. It follows that the digraph $\mathcal{SG}(\mathcal{A}_{1,n-2})$ is formed by "layers" represented by the components $\{C_i\}_i$ where

$$C_0,\, C_1,\, C_{\frac{n-4}{2}},\, C_3,\, C_{\frac{n-8}{2}},\, C_5,\, C_{\frac{n-12}{2}}, \ldots \tag{4.23}$$

is the sequence of components from the farthest to the closest to the singleton $(n-2, n-2)$. In order to compute the diameter we need to find the length of the shortest path from vertex $(n/2, n/2+1)$ to vertex $(n-2, n-2)$, path that is colored in red in Figure 4.4. This means that for $0 \leq i \leq n/2-1$ we have to compute the distance d_i in C_i between vertices $(2i, n-2)$ and $(1, n-2i-1)$ if i is odd or between vertices $(2i+1, n-2)$ and $(1, n-2i+2)$ if i is even. In view of (4.23), we have the following sequence for the d_is:

$$d_0 = \frac{n}{2} - 1,\; d_1 = n-2,\; d_{\frac{n-4}{2}} = 1,\; d_3 = n-3,\; d_{\frac{n-8}{2}} = 5,\; d_5 = n-7,\; d_{\frac{n-12}{2}} = 9, \ldots$$

Since the number of edges labeled by $\underline{I}$ that appear in the path corresponding to the diameter is $n/2$, the diameter is equal to

$$diam(\mathcal{SG}(\mathcal{A}_{1,n-2})) = \frac{n}{2} + \sum_{k=0}^{\frac{n}{2}-1} d_k = \frac{n^2}{4} + \frac{n}{2} - 1.$$

□

Figure 4.6 at the end of this section represents the square graph of the automaton $\mathcal{E}_8$, where its diameter is colored in red. All the singletons but the one that belongs to the diameter have been omitted.

Theorem 4.23 implies that the matrix sets belonging to the family $\{\mathcal{M}_{i,j}\}$ whose associated DFAs belong to the families $\mathcal{E}_n$, $\mathcal{E}'_n$, $\mathcal{O}_n$ or $\mathcal{O}'_n$, are primitive and have at least quadratic exponent. This is formalized in the next Corollary.

Corollary 4.24. *The NZ-sets $\mathcal{M}_{1,n-1}$ for $n = 4k$, $\mathcal{M}_{1,n-4}$ for $n = 4k+2$ and $\mathcal{M}_{\frac{n-1}{2},\frac{n+1}{2}}$ for $n = 2k+1$ are primitive and have exponent of order $\Omega(n^2/4)$.*

Proof. Consider $\mathcal{M}_{1,n-1}$ for $n = 4k$. Since $Aut(\mathcal{M}_{1,n-1}) = \mathcal{E}_n$ by definition and $\mathcal{E}_n$ is synchronizing, by Theorem 3.19 the set $\mathcal{M}_{1,n-1}$ is primitive and $exp(\mathcal{M}_{1,n-1}) \geq rt(\mathcal{E}_n)$. By Theorem 4.23, $\mathcal{M}_{1,n-1}$ has exponent of order $\Omega(n^2/4)$. The other cases can be proved analogously. □

Our numerical experiments suggest that the reset threshold of the families $\mathcal{E}_n$, $\mathcal{E}'_n$, $\mathcal{O}_n$, $\mathcal{O}'_n$ is of order $\Omega(n^2/2)$.

Conjecture 4.25.

1. *The automaton $\mathcal{E}_n$ has reset threshold of $(n^2-2)/2$;*
2. *The automaton $\mathcal{E}'_n$ has reset threshold of $(n^2-10)/2$;*
3. *The automata $\mathcal{O}_n$ and $\mathcal{O}'_n$ have reset threshold of $(n^2-1)/2$.*

Furthermore, they represent the synchronizing automata with the largest possible reset threshold among the family $\mathcal{A}_{i,j}$ for respectively $n = 4k$, $n = 4k+2$, $n = 4k+1$ and $n = 4k+3$.

Notice that, although the randomized construction for proper primitive sets presented in Section 4.4.1 is defined when the matrix size n is the product of at least three prime numbers, we here exhibited an extremal DFA on n states for *any* value of n. Szykuła and Vorel presented in [109] a family $\mathcal{A}_n$ of eulerian slowly synchronizing DFAs on n states for $n = 4k+1$ and with reset threshold of $(n^2-3)/2$. Our automaton $\mathcal{O}_n$ is a sub-automaton of $\mathcal{A}_n$, i.e. $\mathcal{A}_n = \mathcal{O}_n \cup \{\underline{I}_{\frac{n+1}{2},\frac{n-1}{2}}\}$, where we remind that $\underline{I}_{ij} = I + \mathbb{I}_{ij} - \mathbb{I}_{ii}$. We have hence proved that their family $\mathcal{A}_n$ is not proper and that their construction can be generalized to any n. Furthermore, we have conjectured that by deleting the letter $\underline{I}_{\frac{n+1}{2},\frac{n-1}{2}}$ in $\mathcal{A}_n$, its reset threshold gets higher.

Theorem 4.23 can also be seen as an improvement in the direction initiated by Gonze et. al. in [49], where they show that the largest square graph diameter among the DFAs on n states and made of $m \geq 2$ permutation matrices is lower bounded by $n^2/4 + o(n^2)$ when n is odd. We have proved that this lower bound also holds for the largest square graph diameter among the *synchronizing* DFAs on n states containing $m \geq 2$ permutation matrices, for any $n \in \mathbb{N}$.

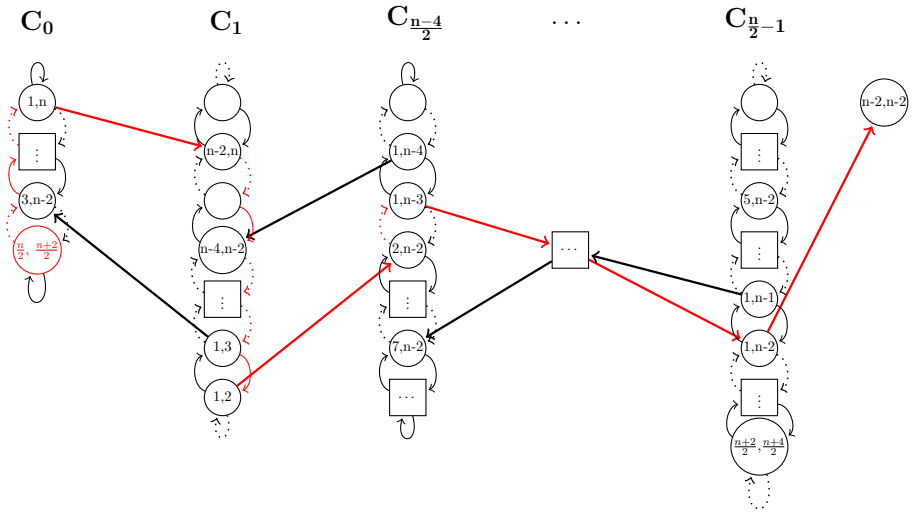


Figure 4.4: Square graph of the family $\mathcal{E}_n$, where all the singletons but $(n-2, n-2)$ have been omitted. There are $n/2$ chains, the first one ($\mathbf{C_0}$) has $n/2$ vertices, the others have n vertices; the missing chains and vertices are represented by squared boxes with dots. Normal lines refer to matrix Q_1, dotted lines to matrix Q_2 and bold lines to matrix $\underline{I}$, where its selfloops have been omitted. The red path is the diameter.

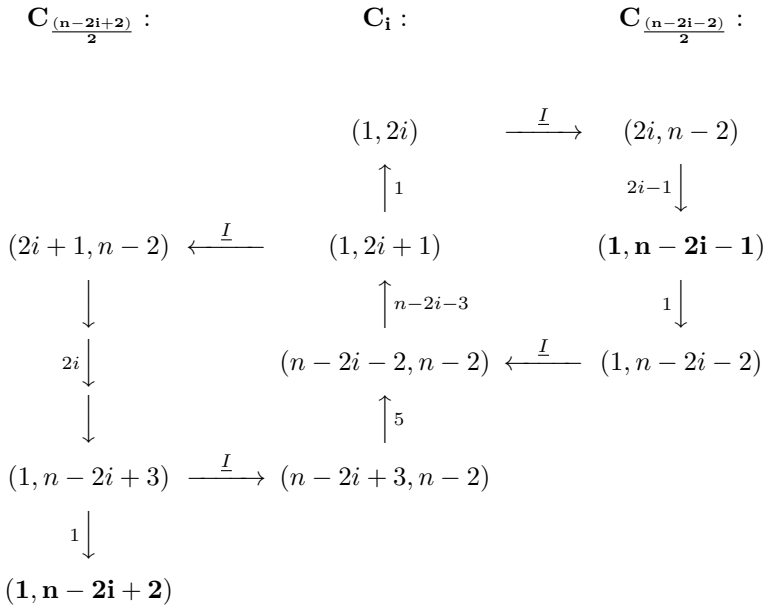


Figure 4.5: Diagram on how the components $\{C_i\}_i$ in the proof of Theorem 4.23 are linked together. Vertices in the same column belong to the same component (indicated above). An arrow between two vertices means that there exists a word mapping the first vertex to the second one, the number next to the arrow represents the length of such word while arrows connecting vertices from different components are labeled by $\underline{I}$. Bold vertices represent the ones that are linked by $\underline{I}$ to other chains.

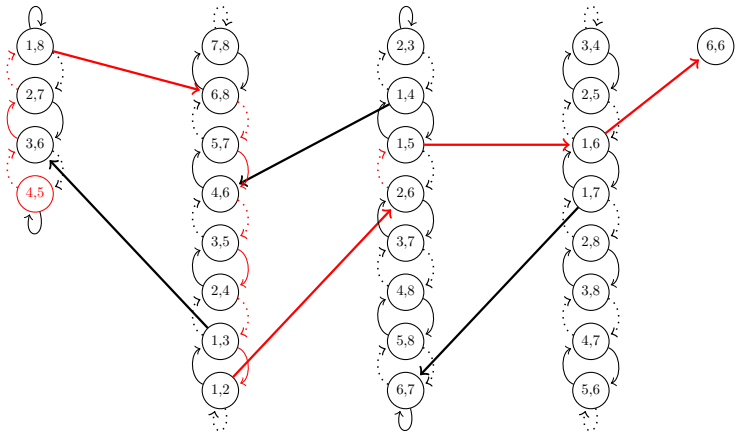


Figure 4.6: Square graph of automaton $\mathcal{E}_8$, $diam(\mathcal{SG}(\mathcal{E}_8)) = 19$. Normal lines refer to matrix Q_1, dotted lines to matrix Q_2 and bold lines to matrix $\underline{I}_{1,6}$, where its selfloops have been omitted. The red path is the diameter.

Chapter 5

The synchronizing probability function for primitive sets

This chapter is devoted to the description of a new tool for studying primitivity, the *Synchronizing Probability Function for primitive sets* (SPF) and it is mainly based on our works [9, 26, 27].

In Chapter 3 we have seen that computing the exponent of a primitive NZ-set is NP-hard [17], even if deciding whether an NZ-set is primitive is solvable in polynomial time. The best upper bound for $exp_{NZ}(n)$ is of order $O(n^3)$ (Corollary 3.21) but it has not yet been found a primitive NZ-set having exponent that is cubic in n. A set of this kind, if it exists, would imply the existence of a synchronizing DFA with cubic reset threshold (Theorem 3.19), thus disproving the Černý conjecture. In view of this, any improvement on the upper bound of $exp_{NZ}(n)$ or any method for approximating the exponent of a primitive NZ-set is particularly of interest.

In 2012 Jungers [68] introduced a probabilistic tool to study synchronizing DFAs: by looking at synchronization as a two-player probabilistic game, he developed the concept of *Sychronizing Probability Function for Automata* (SPFA), a function that describes the speed at which an automaton synchronizes. By reformulating the game as a linear programming problem and making use of convex optimization techniques, he shows that the behavior of the SPFA is closely related with properties of the synchronizing automaton. This tool has lately been used by Gonze and Jungers [51] to prove a quadratic upper bound on the length of the shortest word of a synchronizing DFA mapping three states onto one, also called the *triple-rendezvous time*.

In this chapter, we study the primitivity phenomenon by extending the concept of SPFA to primitive sets, developing what we call the *Synchronizing Probability Function for primitive sets* (SPF). Our goal is to design a function that increases smoothly, representing the convergence of the primitivity process. The SPF is also partly inspired from the *smoothed analysis* in combinatorial optimization, where probabilities are used in order to analyze the convergence of iterative algorithms on combinatorial structures (see e.g. [105]). The chapter is structured as follows: in Section 5.1 we present primitivity as a two-player game and we define the SPF as the probability that the first player wins if both players play optimally. The boolean product between matrices that we have defined in Chapter 3 (Definition 2) will play a central role in its definition. We then reformulate the game as a linear programming

problem and we provide an analysis of some theoretical properties of the SPF in Subsection 5.1.1: we show that this function captures the speed at which a primitive set reaches its first positive product and that it must increase regularly in some sense. Some numerical experiments are reported. In Subsection 5.1.2 we then present some results on the computability of the SPF and we show how it can be used to efficiently approximate the exponent of a primitive NZ-set. Finally, in Section 5.2 we introduce the *approximated* synchronizing probability function, that is an upper bound on the SPF. We show in Subsection 5.2.1 that stronger theoretical properties hold for this new function and that they can be used to obtain an upper bound on $exp_{NZ}(n)$. In Subsection 5.2.2 we introduce the concept of k-rendezvous time (k-RT) for NZ-sets by extending to primitive set the notion already introduced for synchronizing automata, which represents the length of the shortest product having a row or a column with k positive entries. We show that for any fixed k small enough, the k-RT is linear with respect to the matrix size n, result that has not yet being proved for synchronizing automata. We also show how better estimates on the k-RT, combined with the results on the approximated SPF, would lead to an improvement on the upper bound on $exp_{NZ}(n)$.

5.1 Primitivity as a two-player game

We here introduce primitivity as a two-player probabilistic game on a labeled directed multigraph. We remind that all the matrix products have to be read as *boolean* matrix products (see Definition 2, Chapter 3) and that, given $v \in \mathbb{R}^n_{\geq 0}$, we denote with $[v]$ the binary vector such that $[v]_i = 1$ if $v_i > 0$, $[v]_i = 0$ otherwise.

Definition 34. Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be a binary NZ-set of $n \times n$ matrices. We define $\mathscr{D}_{\mathcal{M}} = (\mathcal{V}_{\mathcal{M}}, \mathcal{E}_{\mathcal{M}})$ to be the labeled directed multigraph with set of labels $\mathcal{L}$ such that:

- $\mathcal{V}_{\mathcal{M}} = \{v \in \{0,1\}^n : v \neq (0, \ldots, 0)\}$;
- $\mathcal{L} = \{M_1, \ldots, M_m\}$;
- $v \xrightarrow{M_i} w \in \mathcal{E}_{\mathcal{M}}$ if and only if $[vM_i] = w$.

Remark 15. Notice that for any $v \in \mathcal{V}_{\mathcal{M}}$ and $i \in [m]$, there exists exactly one edge in $\mathscr{D}_{\mathcal{M}}$ leaving v and labeled by M_i. This implies that, given a sequence of labels $l = M_{i_1}, \ldots, M_{i_r}$ and a vertex $v \in \mathcal{V}_{\mathcal{M}}$, there is exactly one path in $\mathscr{D}_{\mathcal{M}}$ of length r leaving v and labeled by l. This implies in turn that for any sequence of labels $l = M_{i_1}, \ldots, M_{i_r}$ and vertex $v \in \mathcal{V}_{\mathcal{M}}$, there exists a unique vertex $w \in \mathcal{V}_{\mathcal{M}}$ such that $v \xrightarrow{l} w \in \mathcal{E}_{\mathcal{M}}$ (for further explanation on the notation, see Chapter 3).

The following example reports a binary NZ-set $\mathcal{M}$ and its associated labeled directed multigraph $\mathscr{D}_{\mathcal{M}}$.

Example 12. Consider the following matrix set:

$$\mathcal{M} = \left\{ M_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \right\}.$$

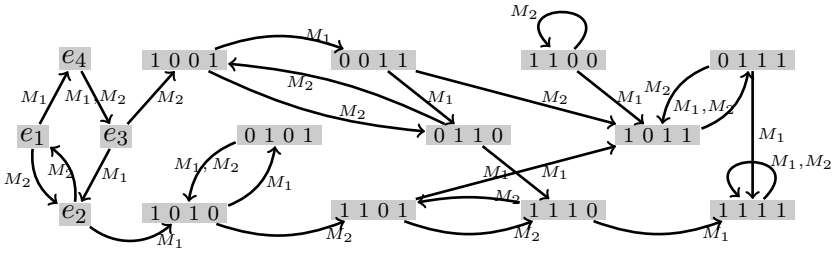


Figure 5.1: The labeled directed multigraph $\mathscr{D}_{\mathcal{M}}$ of the matrix set $\mathcal{M}$ in Example 12. We remind that in this case $e_1 = (1,0,0,0)$, $e_2 = (0,1,0,0)$, $e_3 = (0,0,1,0)$ and $e_4 = (0,0,0,1)$.

The graph $\mathscr{D}_{\mathcal{M}}$ is shown in Figure 5.1.

We now fix a binary NZ-set $\mathcal{M} = \{M_1, \ldots, M_m\}$ of $n \times n$ matrices and an integer $t \geq 1$. We are going to describe a game between two players on the graph $\mathscr{D}_{\mathcal{M}} = (\mathcal{V}_{\mathcal{M}}, \mathcal{E}_{\mathcal{M}})$. We remind that we indicate with $\mathscr{E}_n$ the canonical basis of $\mathbb{R}^n$ and that we denote with e the all-ones vector $(1, 1, \ldots, 1)$; the length of e will be clear from the context.

Game 1. 1. Player B secretly chooses an initial vertex $e_i \in \mathscr{E}_n \subset \mathcal{V}_{\mathcal{M}}$;

2. Player A chooses a sequence $l = M_{i_1} \ldots M_{i_r}$ of at most t matrices in $\mathcal{M}$;

3. Let $w \in \mathcal{V}_{\mathcal{M}}$ such that $e_i \xrightarrow{l} w \in \mathcal{E}_{\mathcal{M}}$ (w exists and is unique by Remark 15). An entry w_j of $w = (w_1, \ldots, w_n)$ is chosen uniformly at random: if $w_j = 1$ then Player A wins, otherwise Player B wins.

Notice that the vertex $w \in \mathcal{V}_{\mathcal{M}}$ in point 3. of Game 1 is the vector $w = [e_i M_{i_1} \cdots M_{i_r}]$.

We consider that both players can choose probabilistic strategies. The *policy* of Player B is a probability distribution over the canonical basis $\mathscr{E}_n$, that is any stochastic vector $p \in \mathbb{R}^n_{\geq 0}$; he chooses the vertex e_i with probability p_i.

Definition 35. Given a set $\mathcal{M}$ of binary NZ-matrices, we denote with $\mathcal{M}^{\leq t}$ the set of all the (boolean) products of elements from $\mathcal{M}$ of length at most t and we indicate with h_t the cardinality of $\mathcal{M}^{\leq t}$. Finally, we denote with $\mathcal{M}^t$ the set of all the (boolean) products of elements from $\mathcal{M}$ of length exactly t.

The policy of Player A is a probability distribution over the set $\mathcal{M}^{\leq t}$, that is a stochastic vector q of length h_t: Player A chooses to play the j-th element of $\mathcal{M}^{\leq t}$ with probability q_j.

We are interested in an optimal strategy for Player A. Notice that if Player A can play a sequence $l = M_{i_1} \ldots M_{i_r}$ such that for all $e_i \in \mathscr{E}_n$, $e_i \xrightarrow{l} e \in \mathcal{E}_{\mathcal{M}}$, then Player A is sure to win. To meet this conditions, the product $M = M_{i_1} \cdots M_{i_r}$ has to have all positive entries, i.e. it has to be a *positive* product; therefore, if the matrix set $\mathcal{M}$ is primitive and $t \geq exp(\mathcal{M})$, then Player A has an optimal strategy for winning surely by playing a positive product. For $t < exp(\mathcal{M})$ Player A wants to maximize her probablity of winning. The term

$$p^T M_{i_1} \cdots M_{i_r} \frac{e}{n} \tag{5.1}$$

represents the probability that Player A wins by playing the product $M_{i_1} \cdots M_{i_r}$ given the policy p of Player B; indeed e/n is the uniform distribution over the set $[n]$. Player A wants to maximize the term (5.1) over all her choices of the product $M_{i_1} \cdots M_{i_r} \in \mathcal{M}^{\leq t}$, while Player B wants to minimize it over all his choices of the distribution p, if he wants to play optimally. The *Synchronizing Probability Function for primitive sets*, presented in the following definition, formalizes this idea: it represents the probability that Player A wins if both players play optimally.

Definition 36. Let $\mathcal{M}$ be a binary NZ-set of $n \times n$ matrices. The *Synchronizing Probability Function (SPF)* for the set $\mathcal{M}$ is the function $K_{\mathcal{M}} : \mathbb{N} \to \mathbb{R}_{\geq 0}$:

$$K_{\mathcal{M}}(t) = \min_{p \in \mathbb{R}^n_{\geq 0},\, p^T e = 1} \left\{ \max_{M \in \mathcal{M}^{\leq t}} p^T M(e/n) \right\} . \tag{5.2}$$

By convention we assume that the product of length zero $\mathcal{M}^0$ is the identity matrix. Sometimes we will indicate the SPF just with $K(t)$ when the matrix set will be clear from the context.

We have seen that if the set $\mathcal{M}$ is primitive, then Player A has a strategy for winning surely when $t \geq exp(\mathcal{M})$. The opposite is also true: if Player A is sure to win at time t, then $\mathcal{M}$ must have a positive product of length at most t. The following proposition formalizes this fact:

Proposition 5.1. *The function $K_{\mathcal{M}}(t)$ takes values in $[0, 1]$ and is nondecreasing in t. Moreover, there exists $t \in \mathbb{N}$ such that $K_{\mathcal{M}}(t) = 1$ if and only if $\mathcal{M}$ is primitive. In this case, $exp(\mathcal{M}) = \min\{t : K_{\mathcal{M}}(t) = 1\}$.*

Proof. Since we are using the boolean matrix product, for every $M \in \mathcal{M}^{\leq t}$ and any stochastic vector $p \in \mathbb{R}^n_{\geq 0}$, it holds that $0 \leq p^T M(e/n) \leq p^T e \leq 1$, so $K_{\mathcal{M}}(t)$ takes values in $[0, 1]$. $K_{\mathcal{M}}(t)$ is nondecreasing since $\mathcal{M}^{\leq t} \subseteq \mathcal{M}^{\leq t+1}$ for every t. Finally, Equation (5.2) implies that $K_{\mathcal{M}}(t)$ is equal to 1 if and only if for any stochastic vector p there exist $M \in \mathcal{M}^{\leq t}$ such that $p^T M(e/n) = 1$. By taking $p = e/n$, it follows that the matrix M is the all-ones matrix and so $exp(\mathcal{M}) = \min\{t : K_{\mathcal{M}}(t) = 1\}$ by the definition of the exponent of a primitive set. □

The next example shows the graph plot of the SPF of three different primitive matrix sets. Proposition 5.1 says that we can read the magnitude of their exponents directly from the graphs of their SPFs, as it is equal to the abscissa of the point at which $K(t)$ reaches the value 1.

Example 13. Figure 5.2 reports the SPF of the following primitive sets:

$$\mathcal{M}_0 = \left\{ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \right\}, \tag{5.3}$$

$$\mathcal{M}_1 = \left\{ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \right\}, \tag{5.4}$$

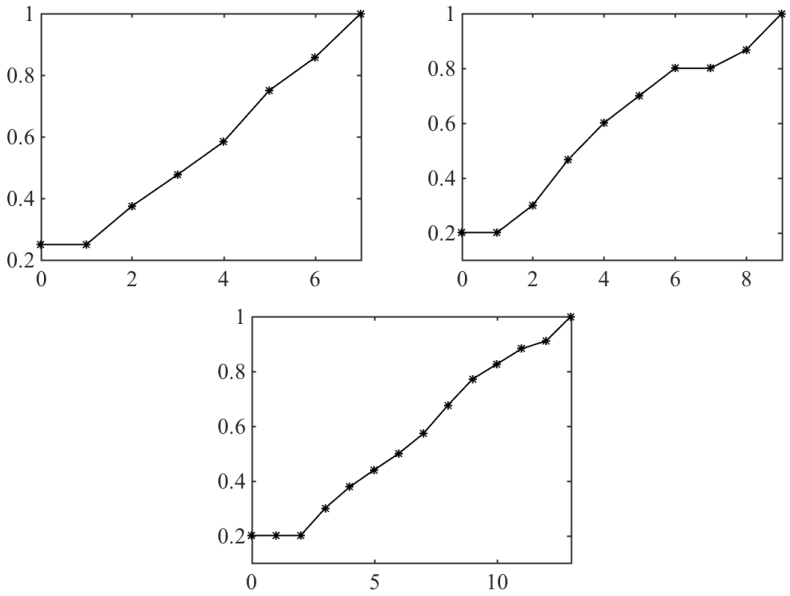


Figure 5.2: The function $K(t)$ of the set $\mathcal{M}_0$ in Equation (5.3) (upper-left picture), of the set $\mathcal{M}_1$ in Equation (5.4) (upper-right picture) and of the set $\mathcal{M}_2$ in Equation (5.5) (bottom picture).

$$\mathcal{M}_2 = \left\{ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \right\}. \quad (5.5)$$

It holds that $exp(\mathcal{M}_0) = 7$, $exp(\mathcal{M}_1) = 9$ and $exp(\mathcal{M}_2) = 13$.

The SPF measures how fast a set reaches a positive product by taking into account the evolution of the matrix semigroup generated by the set; the SPF seems to increase quite regularly after an initial stagnation.

In the next section we will see how to reformulate the SPF as the solution of a linear programming problem, which will enable us to prove some interesting properties on its behavior.

5.1.1 The linear programming formulation

The SPF can be reformulated as a linear programming problem, as it is showed in the following Theorem 5.2. Before stating it, we need the following definition; we remind that $\mathcal{M}^{\leq t}$ and h_t are defined in Definition 35 and that e represents the all-ones vector.

Definition 37. Given a set $\mathcal{M}$ of $n \times n$ binary NZ-matrices, we define the matrix H_t to be the $n \times h_t$ matrix whose i-th column is equal to $A_i e$, with A_i the i-th element of $\mathcal{M}^{\leq t}$.

The matrix H_t has entries in $[n]$ due to the boolean product and $H_0 = e$; in particular, if c^i is the i-th column of H_t and A_i is the i-th element of $\mathcal{M}^{\leq t}$, c^i_l is the number of positive entries in the l-th row of A_i. Note that if the vector ne is a column of H_t, then there must be a positive product in $\mathcal{M}^{\leq t}$ and so $K_{\mathcal{M}}(t) = 1$.

Theorem 5.2. *The synchronizing probability function $K_{\mathcal{M}}(t)$ is given by:*

$$\min_{p,k} \quad \frac{k}{n} \qquad s.t. \qquad \begin{cases} p^T H_t \leq k e^T \\ p^T e = 1 \\ p \geq 0 \end{cases}, \tag{5.6}$$

where p is vector of length n. The function $K_{\mathcal{M}}(t)$ is also given by:

$$\max_{q,k} \quad \frac{k}{n} \qquad s.t. \qquad \begin{cases} H_t q \geq k e \\ e^T q = 1 \\ q \geq 0 \end{cases}, \tag{5.7}$$

where q is a vector of length h_t.

Proof. Programs (5.6) and (5.7) are the dual of each other. Since they both admit feasible solutions, their optima must be equal by the duality theorem of linear programming (see [16], Theorem 4.2). □

The linear program (5.6) represents the point of view of Player B: he wants to minimize the outcome of Player A over his possible choices of p, thus maximizing his own outcome (as they are playing a zero-sum game, see for references [77], Chapter 3). The dual formulation (5.7) represents instead the point of view of Player A, where she wants to maximize her outcome over all her possible choices of q. Theorem 5.2 shows that Player B can make his policy p public without changing the outcome of the game if both players play optimally, as Player A can as well play before Player B.

We now exploit Theorem 5.2 to analyze the game. The first result characterizes the behavior of $K(t)$ for small and big t: it shows that the SPF presents an initial stagnation at the value $1/n$ of length at most $n-1$ and it has to leave it with high discrete derivative; with high discrete derivative it also leaves the last step before hitting the value 1. This is formalized in the following proposition, where we remind the definition of H_t in Definition 37.

Proposition 5.3. *Let $\mathcal{M}$ be a primitive binary NZ-set and let $K(t) = K_{\mathcal{M}}(t)$. It holds that:*

1. $K(0) = 1/n$,
2. $K(n) > 1/n$,
3. $K(t) > 1/n \quad \Rightarrow \quad K(t) \geq (n+1)/n^2$,
4. $K(t) < 1 \quad \Rightarrow \quad K(t) \leq (n^2-1)/n^2$.

Proof.

1. Since $H_0 = e$, then $k = 1$ and $p = e/n$ is a feasible solution for the linear program (5.6), so $K(0) \leq 1/n$. On the other hand, $q = 1$ and $k = 1$ is a feasible solution for the linear program (5.7), so $K(0) \geq 1/n$.

2. We claim that $K(t) = 1/n$ if and only if H_t has an all-ones row. In fact, if H_t has the i-th row entrywise equal to 1, then $p = e_i$ and $k = 1$ is an optimal solution for the linear program (5.6), so $K(t) = 1/n$. On the other hand, suppose that every row of H_t has at least one entry greater than 1: let p be a stochastic vector and j an index such that $p_j > 0$.

Then it holds that $\max_i\{(p^T H_t)_i\} \geq 2p_j + (1 - p_j) = 1 + p_j > 1$, which implies that $k > 1$ and so $K(t) > 1/n$; we hence proved the claim. Since the set $\mathcal{M}$ is primitive and NZ, there must be a matrix $M \in \mathcal{M}$ with at least two positive entries in the same row; therefore, H_1 must have a column with an entry ≥ 2. Suppose this entry is in row s. As $\mathcal{M}$ is irreducible, its associated digraph is strongly connected (see Chapter 3), so for any $l \in [n]$ there exists a product P_l of at most $n-1$ matrices in $\mathcal{M}$ such that $P_l[l, s] > 0$. This implies that $(P_l M e)_l \geq (Me)_s \geq 2$ and $P_l M \in \mathcal{M}^{\leq n}$. Therefore, for every $l \in [n]$, H_n has a column whose l-th entry is greater than 1, which implies that $K(n) > 1/n$.

3. Let $p \in \mathbb{R}^n_{\geq 0}$ be a stochastic vector: then there must exist an index j such that $p_j \geq 1/n$. By what proved in item 2., if $K(t) > 1/n$ then every column of H_t has an entry greater than one, so $\max_i\{(p^T H_t)_i\} \geq 1+1/n$. It follows that $K(t) \geq (n+1)/n^2$.

4. If $K(t) < 1$, then every column of H_t has at least one entry smaller than n. It follows that

$$k = \frac{1}{n} e^T k e \leq \frac{1}{n} e^T H_t q = \left(\frac{1}{n} e^T H_t\right) q \leq \left(\frac{n^2-1}{n}\right) e^T q = \frac{n^2-1}{n}$$

and so $K(t) \leq (n^2-1)/n^2$.

□

The following proposition are the *complementary slackness* conditions of linear programming (see [16], Section 4.3).

Proposition 5.4. *For any primitive NZ-set $\mathcal{M}$, any integer $t \geq 1$ and any optimal solutions (p^*, q^*) of programs (5.6) and (5.7) respectively, it holds that:*

- $q^*_j(k - (p^{*T} H_t)_j) = 0$ *for all* $1 \leq j \leq h_t$,
- $p^*_i((H_t q^*)_i - k) = 0$ *for all* $1 \leq i \leq n$.

Proof. Since (p^*, q^*) are optimal solutions of (5.6) and (5.7) respectively, we have that $p^* \geq 0$, $q^* \geq 0$, $H_t q^* - ke \geq 0$ and $p^{*T} H_t - ke \leq 0$. Therefore,

$$0 \leq p^{*T}(H_t q^* - ke) = p^{*T} H_t q^* - k = (p^{*T} H_t - ke) q^* \leq 0.$$

This implies that each component of $p^{*T}(H_t q^* - ke)$ and $(p^{*T} H_t - ke)q^*$ has to be equal to zero. □

Computing the SPF can be hard due to the possible exponential growth of the matrix H_t (see Definition 37). The next results show that we can implement some strategies in order to reduce the size of the linear program. In particular, Proposition 5.5 shows that, in order to compute the SPF, there is no need to use the full matrix H_t but we can always potentially find a much smaller submatrix that reaches the same optimal value. Furthermore, Proposition 5.6 states that we can replace the set $\mathcal{M}^{\leq t}$ by the set $\mathcal{M}^t$ (see Definition 35) in some cases.

Proposition 5.5. *For any binary NZ-set $\mathcal{M}$ of $n \times n$ matrices and any integer $t \geq 1$, there always exists a submatrix H'_t of H_t of size $n \times s$, $s \leq n$, such that we can replace H_t with H'_t in the program (5.7) without changing the optimal value.*

Proof. We indicate with $(5.7)'$ the program (5.7) where H_t is replaced by a submatrix H'_t of size $n \times s$ and with $K'(t)$ its optimum. Let q'^* be one of the optimal solutions of $(5.7)'$; q'^* is a feasible solution also for program (5.7) so $K'(t) \geq K(t)$. We now show that for an appropriate submatrix H', we have $K'(t) \leq K(t)$. Let q^* be an optimal solution of (5.7) with all positive entries: if this is not the case, we can just remove its zero entries and the corresponding columns of H_t without changing the optimum. If H_t has more than n columns, the system $H_t x = 0$ has a nonzero solution. We can suppose without loss of generality that

$$e^T x \leq 0. \tag{5.8}$$

Then by setting

$$\lambda = \min_{x_i<0}\{q_i^* / (-x_i)\} \tag{5.9}$$

we obtain that $q^* + \lambda x$ is a feasible solution for program (5.7): indeed $q^* + \lambda x \geq 0$ by Equation (5.9) and $H_t(q^* + \lambda x) = H_t q^* \geq ke$. Furthermore Equation (5.8) implies that $e^T(q^* + \lambda x) \leq 1$ since $\lambda > 0$. In the case $e^T(q^* + \lambda x) < 1$ we can increase a nonzero entry of $q^* + \lambda x$ until the sum is equal to one without losing optimality. By construction, $q^* + \lambda x$ has a zero entry so we can remove the corresponding column in H_t without changing the optimum. We conclude by iteratively applying the above argument until there are no more than n columns in H_t. □

Proposition 5.6. *For any integer $t \geq 1$ and for any binary NZ-set $\mathcal{M}$ in which there exists at least a matrix that dominates a permutation matrix, the set $\mathcal{M}^{\leq t}$ can be replaced by the set $\mathcal{M}^t$ in program (5.6) without changing the optimal value.*

Proof. Since $\mathcal{M}^t \subset \mathcal{M}^{\leq t}$, it is clear that the optimal value decreases; we show that it actually remains the same. Let $A_i \in \mathcal{M}^{\leq t_i}$ for $t_i < t$: we claim that there exists a product $L \in \mathcal{M}^t$ such that $A_i e \leq Le$. In this case we can erase the column $A_i e$ from H_t as for any optimal solution p of program (5.6), $p^T A_i e \leq p^T L e \leq k$. Let $M \in \mathcal{M}^{t-t_i}$ be a product that dominates a permutation matrix (it always exists by hypothesis) and $L = A_i M$; then for every column a of A_i there exist a column l of L such that $a \leq l$, which implies $A_i e \leq Le$. Since L is a product of length t, the claim is proven. □

Proposition 5.6 may fail for sets in which all the matrices do *not* dominate a permutation matrix, as showed in Example 14. In this case, if we denote by $K^=(t)$ the optimal solution of program (5.6) with $\mathcal{M}^{\leq t}$ replaced by $\mathcal{M}^t$, $K^=(t)$ can still provide an approximation of $K(t)$. Indeed, let s be the first time such that $\mathcal{M}^{\leq s}$ contains a matrix that dominates a permutation matrix (s must exist if the set is primitive). Then, for every $t > s$, it holds that

$$K(t) \geq K^=(t) \geq K(t-s)\ . \tag{5.10}$$

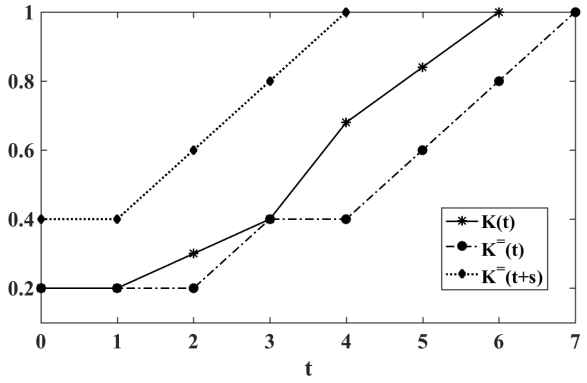


Figure 5.3: The functions $K(t)$, $K^{=}(t)$ and $K^{=}(t+s)$ of the matrix set $\mathcal{M}$ in Equation (5.11), Example 14. In this case it holds that $s = 3$.

This means that, if s is small enough, $K^{=}(t)$ is an accurate approximation of $K(t)$. Furthermore, Equation (5.10) implies that

$$\min\{t : K^{=}(t+s) = 1\} \leq exp(\mathcal{M}) \leq \min\{t : K^{=}(t) = 1\},$$

so $K^{=}$ also provides upper and lower bounds for the exponent of the primitive set $\mathcal{M}$. An example of the functions $K(t)$, $K^{=}(t)$ and $K^{=}(t+s)$ is reported in Figure 5.3.

Example 14. We consider the following primitive set in which every matrix does *not* dominate a permutation matrix:

$$\mathcal{M} = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \right\}. \tag{5.11}$$

It can be checked that

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \\ 2 & 2 \end{pmatrix} \quad \text{and} \quad H_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 & 1 & 1 \\ 2 & 2 & 4 & 4 & 3 & 3 \end{pmatrix} = H_1 \cup H_2^{=},$$

where $H_2^{=}$ is the matrix whose columns are the set $\{Me : M \in \mathcal{M}^2\}$. The solution of program (5.6) for $t = 2$ is $K_{\mathcal{M}}(t) = 0.3$ for $p = (0, 0, 1/2, 1/2, 0)^T$, while if we substitute H_2 with $H_2^{=}$ we have that $K_{\mathcal{M}}^{=}(t) = 0.2$ for $p = (0, 0, 0, 1, 0)^T$. In Figure 5.3 is reported the behavior of the functions $K(t)$, $K^{=}(t)$ and $K^{=}(t+s)$ for the set $\mathcal{M}$; in this case it holds that $s = 3$ (i.e. $\mathcal{M}^3$ is the first set in which it appears a matrix that dominates a permutation matrix), $exp(\mathcal{M}) = 6$, $\min\{t : K^{=}(t) = 1\} = 7$ and $\min\{t : K^{=}(t+s) = 1\} = 4$.

Notice that the recursive construction of H_{t+1} from H_t as $H_{t+1} = \{Mc : c \text{ column of } H_t \text{ and } M \in \mathcal{M}\}$ is not valid, due to the boolean matrix product that we are using; an example of why this relation is false is presented in Example 15. Consequently, in order to compute H_{t+1} we first need to compute

$\mathcal{M}^{\leq t+1} = \{NM : N \in \mathcal{M}^{\leq t}, M \in \mathcal{M}\}$ recursively from $\mathcal{M}^{\leq t}$, and then set $H_{t+1} = \{Me : M \in \mathcal{M}^{\leq t+1}\}$. The following strategies can be implemented in order to reduce the size of H_t and so decrease the complexity of the problem:

- **If $\mathbf{A_1}, \mathbf{A_2} \in \mathcal{M}^{\leq \mathbf{t}}$ and $\mathbf{A_2} \leq \mathbf{A_1}$, then $\mathbf{A_2}$ can be erased from $\mathcal{M}^{\leq \mathbf{t}}$.** First notice that $A_2 \leq A_1$ implies $A_2 e \leq A_1 e$, and so for any stochastic vector p such that $p^T A_1 e \leq k$, it also holds $p^T A_2 e \leq k$. We can therefore erase $A_2 e$ from H_t without changing the optimal value. Secondly, for any binary NZ-matrix B, $A_2 \leq A_1$ implies $BA_2 \leq BA_1$, which again implies $BA_2 e \leq BA_1 e$. Consequently, the product A_2 can be permanently erased from $\mathcal{M}^{\leq t}$ as every product of type BA_2 with $B \in \mathcal{M}^{\leq s}$ will *not* play a role in the solution of program (5.6) at time $t+s$, for any $s \geq 1$.

- **If $\mathbf{c_1}$ and $\mathbf{c_2}$ are two columns of $\mathbf{H_t}$ and $\mathbf{c_1} \leq \mathbf{c_2}$, the column $\mathbf{c_1}$ can be erased from $\mathbf{H_t}$.** Indeed, for any stochastic vector p, the constraint $p^T c_1 \leq k$ in program (5.6) is automatically fulfilled by the constrain $p^T c_2 \leq k$.

- **If $\mathbf{r_1}$ and $\mathbf{r_2}$ are two rows of $\mathbf{H_t}$ and $\mathbf{r_1} \geq \mathbf{r_2}$, then $\mathbf{r_1}$ can be erased from $\mathbf{H_t}$.** Indeed, for any stochastic vector q, the constraint $r_1 q \geq k$ in program (5.7) is automatically fulfilled by the constraint $r_2 q \geq k$.

Example 15. Consider the following matrices:

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

It holds that $(A \odot B)e = Be = (2,1,1,2)^T$, while $A(Be) = A(2,1,1,2)^T = (4,1,1,2)$; therefore $(A \odot B)e \neq A(Be)$, where $\odot$ is the boolean product.

5.1.2 Approximation of the exponent

Computing the exponent of a primitive set $\mathcal{M}$ is an NP-hard problem, and so must be computing the SPF until $t = exp(\mathcal{M})$. In this section we describe how to use the synchronizing probability function to approximate the exponent of a primitive NZ-set.

We say that the function $K(t)$ has a *stagnation* at time $\bar{t}$ if there exists an integer $l > 0$ such that $K(\bar{t}) = K(\bar{t}+1) = \cdots = K(\bar{t}+l)$. If $K(t)$ has a stagnation at time $\bar{t}$, we denote with $l_{\bar{t}}$ the maximal integer such that $K(\bar{t}) = K(\bar{t}+1) = \cdots = K(\bar{t}+l_{\bar{t}})$. Proposition 5.3 showed that $K(t)$ has always an initial stagnation at time $\bar{t} = 0$ for $l_0 \leq n-1$; this upper bound on l_0 is sharp as there exist sets whose SPF is constantly equal to $1/n$ untill $t = n-1$ (see Example 16). This fact suggests that we could start solving the linear program (5.6) directly from $t = l_0 + 1$, as the behavior for $t \leq l_0$ is known. The problem whether we can do this without computing the sets $\mathcal{M}^{\leq t}$ for all $t \leq l_0$ is still open.

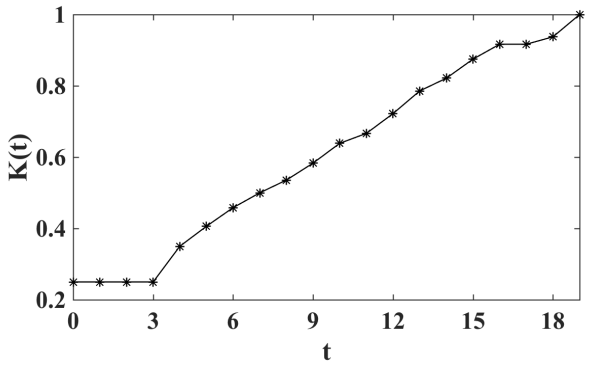


Figure 5.4: The SPF of the matrix set $\mathcal{M}$ in Example 16.

Example 16. Consider the matrix set

$$\mathcal{M} = \left\{ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\}. \tag{5.12}$$

Its SPF is reported in Figure 5.4. We can see that $n = 4$ and its initial stagnation lasts till $t = 3 = n - 1$.

After the initial stagnation, the SPF seems to have a quite linear behavior: this could be leveraged to guess the magnitude of the exponent of a primitive NZ-set without explicitly computing it. This idea is developed in the next paragraph, where we report numerical experiments that show the goodness of the approximation of the exponent via the SPF. We then approach the problem of approximating the exponent from a theoretical point of view by showing that results on the behavior of $K(t)$ could be used to obtain an upper bound on $exp_{NZ}(n)$.

Linear approximation of the SPF

We want to approximate the behavior of the SPF via a linear function. One simple way to do it is to choose a time $t' > l_0$ and consider the straight line r_1 passing through the points $(l_0, K(l_0))$ and $(t', K(t'))$: the abscissa of the point at which r_1 reaches the value 1 is our approximation of the exponent; we call this the r_1-*method*. We can also consider as straight line, the line r_2 that is computed as linear regression on all the points $(s, K(s))$ for $s = l_0, l_0+1, \ldots, t'$ via least square method; we call this the r_2-*method*. It is reasonable to think about t' as an increasing function of n; intuitively, the greater t' is, the better the approximation should be. Figure 5.5 represents the lines r_1 and r_2 of the primitive set $\mathcal{M}$ in Example 16, where in this case $l_0 = 3$ and we have chosen $t' = 8$. Both the methods return slightly more than 16 as approximation of the exponent of $\mathcal{M}$, while the real value is $exp(\mathcal{M}) = 19$.

We would like to know how good is the approximation of the exponent via the linearization of the SPF. To establish that, we would need to know the exponents of a large number of primitive NZ-sets of different matrix size in order to compare them with the corresponding approximations. Many issues arise:

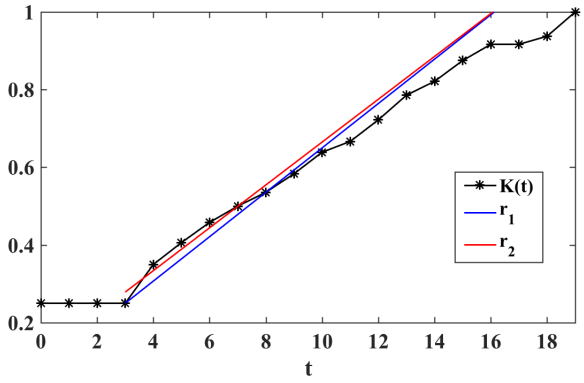


Figure 5.5: The SPF of the matrix set $\mathcal{M}$ in Example 16 together with the approximation lines r_1 and r_2 for $l_0 = 3$ and $t' = 8$; the exponent of $\mathcal{M}$, which is equal to 19, is approximated with 16 by both methods.

1. primitivity is a rather new concept so, to the best of our knowledge, there does not exist any database collecting the exponents of several primitive sets that we can use to test our approximation;

2. if we generate a binary NZ-set according to the uniform distribution, it has very low exponent most of the times, usually of magnitude around 5 regardless of the matrix size. Consequently, in this case the real exponent is computable but it is too low to meaningfully test our approximation;

3. a random perturbed permutation set (see Definition 27) has generally a larger exponent compared to the uniform generation of a binary set, but then computing the exact exponents of thousands of such instances becomes hard;

4. very few primitive sets with quadratic exponent are known (see for example the sets presented in Chapter 4, Corollary 4.24) and usually are provided just quadratic *lower* bounds on their exponents, not the exact values.

In view of this, we decided to compare our method with another approximation method. The *Eppstein's heuristic* [38] is a greedy algorithm for approximating the reset threshold of a synchronizing DFA: it basically computes the word $w = w_1 w_2 \cdots w_n$ that appears in the proof of the square graph criterion for synchronizing DFAs (Proposition 3.12). In other words, the Eppstein heuristic makes use of the square graph in order to compute at each step i the shortest word w_i that merges two given states together; in this way, in $n-1$ steps we have merged all the n states into a single one, thus finding a reset word. Clearly, the Eppstein's heuristic provides an *upper* bound on the reset threshold of a synchronizing DFA.

Let now $\mathcal{M}$ be a primitive NZ-set of $n \times n$ matrices: $\mathcal{M}$ and $\mathcal{M}^T$ are also column-primitive sets, and since the square graph criterion also holds for column-primitivity (Proposition 3.11), we can use the Eppstein heuristic to find an upper bound on $pc(\mathcal{M})$ and $pc(\mathcal{M}^T)$, that is on the shortest positive-column products of $\mathcal{M}$ and $\mathcal{M}^T$ respectively. In view of Corollary 3.23, Theorem 3.19 and Proposition 3.10, if $Epp(pc(\mathcal{M}))$ represents the Eppstein upper

bound on $pc(\mathcal{M})$, then it holds that

$$diam(\mathcal{SG}(\mathcal{M})) \leq exp(\mathcal{M}) \leq Epp(pc(\mathcal{M})) + Epp(pc(\mathcal{M}^T)) + n - 1 \ . \quad (5.13)$$

We will compare the approximation of the exponent via SPF with the upper and lower bounds in Equation (5.13).

For our first experiment we proceed as follows: we choose three different functions for t', namely $t'(n) = \log n$, $t'(n) = (3 \log n)/2$ and $t'(n) = 2 \log n$. For each of these functions and each matrix size $n = 10, 15, 20, 25$, we generate $5n$ random perturbed permutations sets (see Definition 28). For each primitive generated set, we compute the approximation of the exponent via SPF using the r_1-method and the r_2-method, that we respectively denote with $r_1(exp(\mathcal{M}))$ and $r_2(exp(\mathcal{M}))$; we then check if

$$r_1(exp(\mathcal{M})),\ r_2(exp(\mathcal{M})) \geq diam(\mathcal{SG}(\mathcal{M})) \quad (5.14)$$

and if

$$r_1(exp(\mathcal{M})),\ r_2(exp(\mathcal{M})) \leq Epp(pc(\mathcal{M})) + Epp(pc(\mathcal{M}^T)) + n - 1. \quad (5.15)$$

The data we obtained showed that in *all* the cases Equation (5.14) was fulfilled, i.e. that the r_1-method and the r_2-method were providing approximations that were greater than $diam(\mathcal{SG}(\mathcal{M}))$ every time. In Figure 5.6 we report the percentage of sets whose approximations of the exponent via the r_1-method and the r_2-method resulted to fulfil Equation (5.15) with respect to the matrix size n. We can notice that the SPF approximation usually behaves better than the Eppstein heuristic for smaller values of n, while the behavior is reversed for larger values of n. We also underline that the SPF approximation seems to behave better when $t'(n)$ becomes larger (as we were expecting) and that the r_1-method seems to provide slightly better approximations than the r_2-method.

We then tested the SPF approximation on primitive sets with quadratic exponent. The first families we consider are the families $\mathcal{M}_{1,n-2}$ for $n = 4k$, $\mathcal{M}_{1,n-4}$ for $n = 4k + 2$ and $\mathcal{M}_{\frac{n-1}{2},\frac{n+1}{2}}$ for $n = 2k + 1$ presented in Chapter 4, Corollary 4.24. We assume that Conjecture 4.25 on the reset threhsold of the automata $\mathcal{E}_n = Aut(\mathcal{M}_{1,n-2})$, $\mathcal{E}'_n = Aut(\mathcal{M}_{1,n-4})$ and $\mathcal{O}_n = Aut(\mathcal{M}_{\frac{n-1}{2},\frac{n+1}{2}})$ holds true. Therefore, in view of Theorem 3.19 and Equation (5.13) it holds that:

$$\begin{cases} \frac{n^2-2}{2} \leq exp(\mathcal{M}_{1,n-2}) \leq \frac{n^2-2}{2} + Epp\big(pc(\mathcal{M}^T_{1,n-2})\big) + n - 1 & \text{if } n = 4k, \\ \frac{n^2-10}{2} \leq exp(\mathcal{M}_{1,n-4}) \leq \frac{n^2-10}{2} + Epp\big(pc(\mathcal{M}^T_{1,n-4})\big) + n - 1 & \text{if } n = 4k+2, \\ \frac{n^2-1}{2} \leq exp(\mathcal{M}_{\frac{n-1}{2},\frac{n+1}{2}}) \leq \frac{n^2-1}{2} + Epp\big(pc(\mathcal{M}^T_{\frac{n-1}{2},\frac{n+1}{2}})\big) + n - 1 & \text{if } n = 2k+1. \end{cases} \quad (5.16)$$

Figure 5.7 reports the SPF approximation of the exponent of these extremal primitive sets via the r_1-method and r_2-method for $t'(n) = \log n$ and for n from 5 to 15. We call *upper b.* and *lower b.* respectively the right-hand terms and

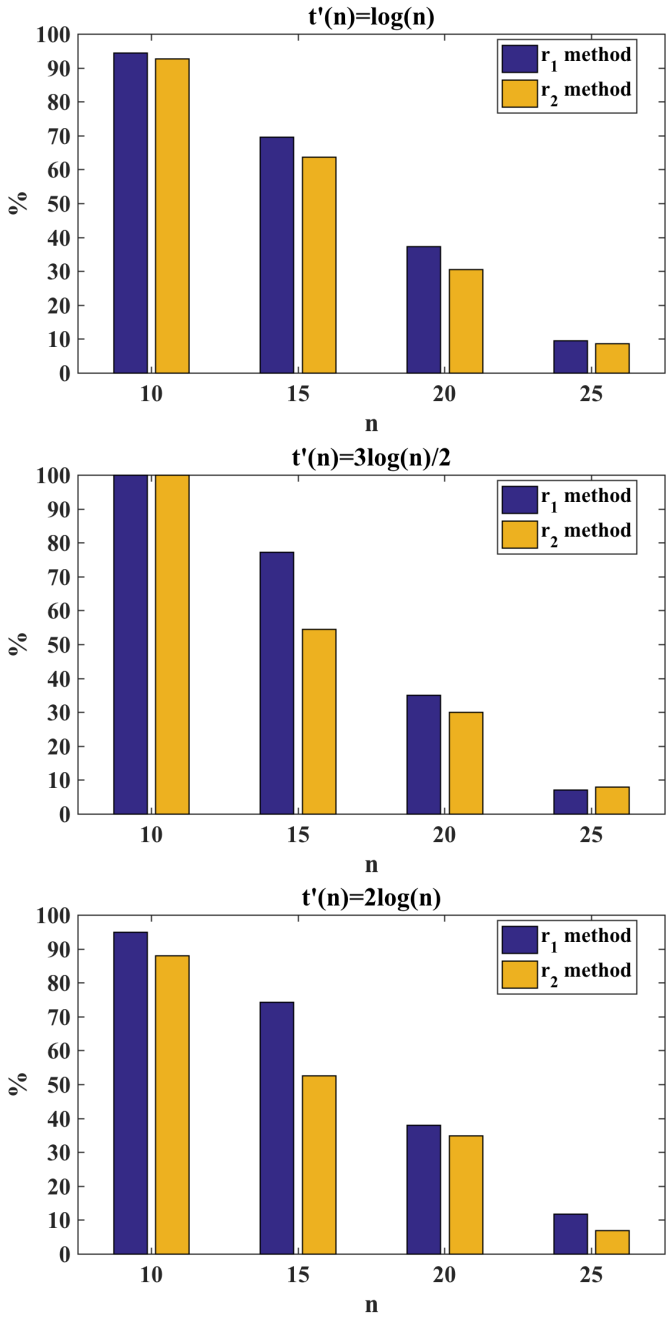


Figure 5.6: Percentage of the generated perturbed permutation sets for which it resulted that the SPF approximation was smaller than the Eppstein heuristic in Equation (5.13), for each matrix size $n = 10, 15, 20, 25$ and method r_1 and r_2. The top figure refers to the choice $t'(n) = \log n$, the center figure to $t'(n) = (3 \log n)/2$ and the bottom figure to $t'(n) = 2 \log n$.

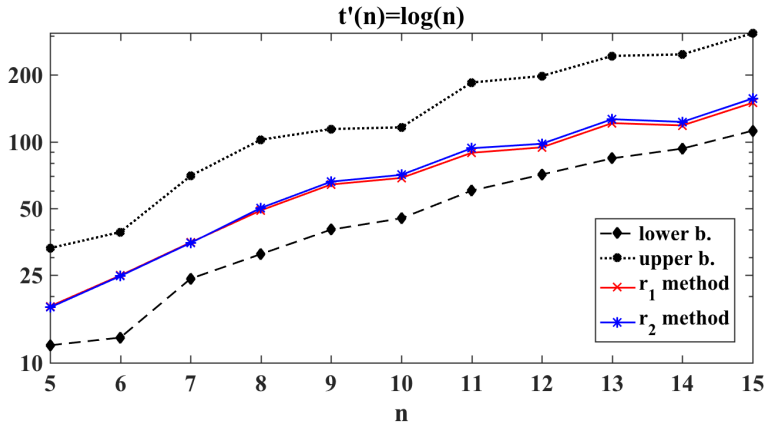


Figure 5.7: Behavior of the SPF approximation of $exp(\mathcal{M}_n)$ via the r_1-method and r_2-method with respect to the upper and lower bounds of Equation (5.16) for $t'(n) = log(n)$.

left-hand terms of Equation (5.16). We can notice that the r_1-method and r_2-method behave quite similary and that they always successfully provide a better approximation of $exp(\mathcal{M}_{i,j})$ than the upper and lower bounds of Equation (5.16).

Secondly, we tested the SPF approximation on the family of primitive sets whose associated DFAs are the Černý familiy: for every $n \in \mathbb{N}$, we set $\mathscr{C}_n^{NZ} = \{a + \mathbb{I}_{n,n}, b\}$ where $\mathscr{C}_n = \{a, b\}$ is the Černý automaton on n states defined in Equation (3.8). Clearly, it holds that $\mathscr{C}_n = Aut(\mathscr{C}_n^{NZ})$; we remind that $rt(\mathscr{C}_n) = (n-1)^2$. Notice that $(\mathscr{C}_n^{NZ})^T = \{a^T + \mathbb{I}_{n,n}, b^T\}$ and that $Aut((\mathscr{C}_n^{NZ})^T) = \{a^T, b^T\} = \mathscr{C}_n^T$. The automaton $\mathscr{C}_n^T$ is equal to $\mathscr{C}_n$ up to a relabeling of the vertices so it holds that $rt(\mathscr{C}_n^T) = (n-1)^2$. By Theorem 3.19 it follows that:

$$(n-1)^2 \leq exp(\mathscr{C}_n^{NZ}) \leq 2(n-1)^2 + n - 1\,. \tag{5.17}$$

Figure 5.8 reports the SPF approximation of $exp(\mathscr{C}_n^{NZ})$ via the r_1-method and r_2-method for $t'(n) = \log n$, $t'(n) = 3\log n/2$ and $t'(n) = 2\log n$ and for n from 5 to 15. We call *upper b.* and *lower b.* respectively the right-hand term and left-hand term of Equation (5.17). We can notice that the r_1-method and r_2-method behave quite similary but for $t'(n) = 2\log n$ sometimes the r_2-method manages to get a better approximation of $exp(\mathscr{C}_n^{NZ})$ than the lower bound $(n-1)^2$ while the r_1-method does not. We can observe again that, as the function $t'(n)$ increases from $\log n$ to $2\log n$, the SPF approximation gets better.

Upper bounding $\exp_{\mathbf{NZ}}(\mathbf{n})$ via the SPF

Suppose that one could prove the existence of a function $a = a(n)$ such that for any primitive NZ-set of $n \times n$ matrices and stagnation point $\bar{t}$ of $K(t)$ with $K(\bar{t}) < 1$, it holds that $l_{\bar{t}} \leq a$ (i.e. any stagnation has length at most a). Suppose furthermore that one could prove the existence of a function $b = b(n)$ such that, for any $t_1 > t_2$ integers and for any primitive NZ-set of $n \times n$ matrices, $K(t_1) > K(t_2)$ implies that $K(t_1) - K(t_2) \geq 1/b$. In view of the fact

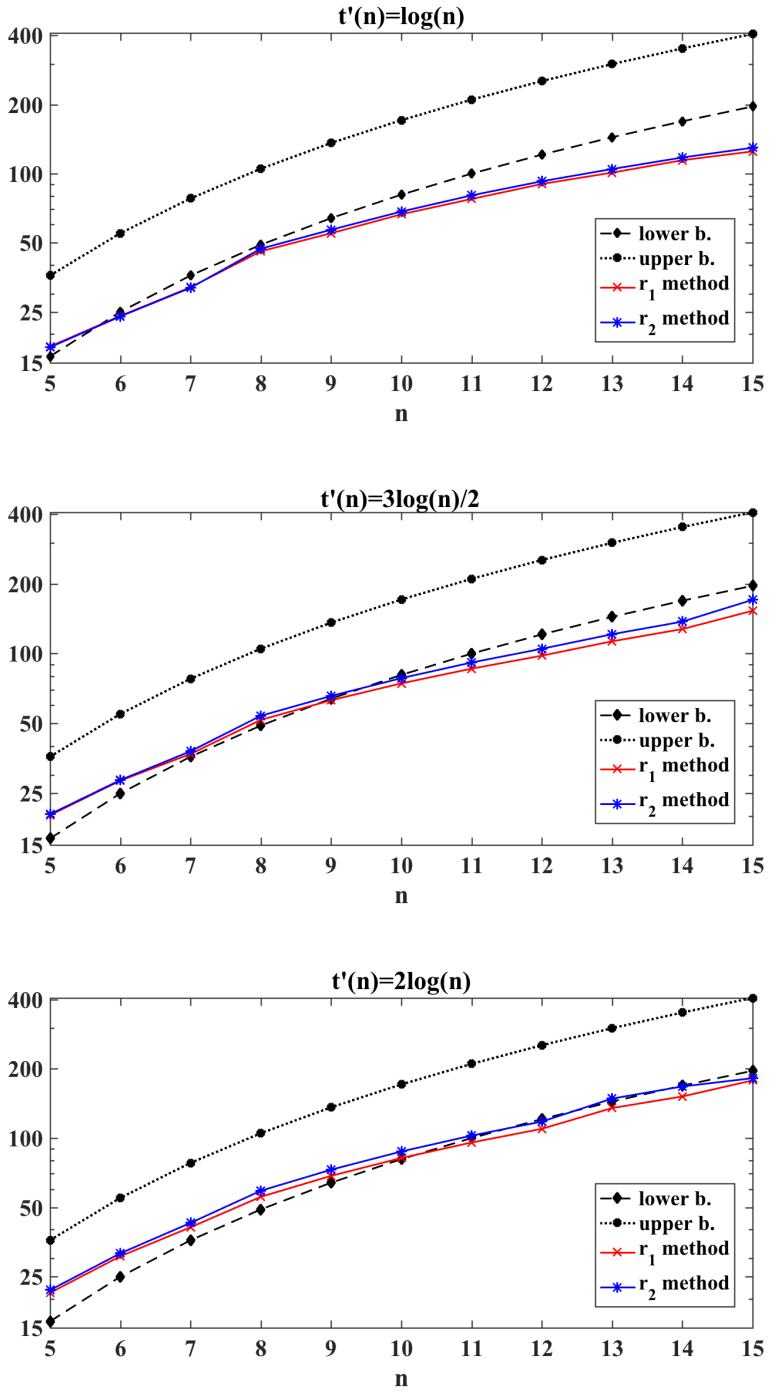


Figure 5.8: Behavior of the SPF approximation of $exp(\mathscr{C}_n^{NZ})$ via the r_1-method and r_2-method with respect to the upper and lower bounds of Equation (5.17) for $t'(n) = log(n)$, $t'(n) = 3log(n)/2$ and $t'(n) = 2log(n)$.

that $exp(\mathcal{M}) = \min\{t : K_{\mathcal{M}}(t) = 1\}$ and $K(0) = 1/n$, it would hold that for any primitive NZ-set $\mathcal{M}$ of $n \times n$ matrices

$$exp(\mathcal{M}) \leq ab(n-1)/n \ . \tag{5.18}$$

In particular, if both $a(n)$ and $b(n)$ were linear in n, we would have a quadratic upper bound on $exp_{NZ}(n)$. Unfortunately our numerical simulations suggest that the difference $K(t_1) - K(t_2)$ for $t_1 > t_2$ can be arbitrarily small, thus letting open the question whether there exists a function $b(n)$ such that $K(t_1) - K(t_2) \geq 1/b$ for any $t_1 > t_2$. What we can say about the stagnations of $K(t)$ is summarized in Proposition 5.7, but before stating it we need the following definition:

Definition 38. Given a binary NZ-set $\mathcal{M}$ and an integer t, we denote with P_t the set of optimal solutions of program (5.6).

Since H_t (see Definition 37) has always rank ≥ 1, then $dim(P_t) \leq n-1$. In the following, given a set of vectors V and a matrix M, we set $M^T V = \{M^T v : v \in V\}$.

Proposition 5.7. *If $K_{\mathcal{M}}(t) = K_{\mathcal{M}}(t+1)$, then $P_{t+1} \subseteq P_t$ and for any binary row-stochastic matrix R such that $R \leq M$ for some $M \in \mathcal{M}$, it holds that $R^T P_{t+1} \subseteq P_t$.*

Proof. The fact that $P_{t+1} \subseteq P_t$ is trivial. Let now $p \in P_{t+1}$, R be a binary row-stochastic matrix such that $R \leq M$ for some $M \in \mathcal{M}$, and let $A \in \mathcal{M}^{\leq t}$. By hypothesis, $nK(t) = k \geq p^T(MA)e \geq p^T(RA)e = p^T R(Ae)$, where the last two passages hold because R is binary and row-stochastic. Since $(p^T R)^T = R^T p$ is a stochastic vector, it follows that $R^T p \in P_t$. □

We remark that, if we prove that P_{t+1} is *strictly* contained in P_t at any time t such that $K(t) = K(t+1)$, then it would hold that $K(t+n) > K(t)$ for any t such that $K(t) < 1$ in view of the fact that $dim(P_{t+1}) < dim(P_t) \leq n-1$; so in this case we would have $a(n) = n$. This can be proved if we define the SPF in the same way as in Definition 36 but using the standard matrix-product instead of the boolean product: the drawback is that now $K(t) = 1$ does not guarantee anymore the presence of a positive product of length t as in this case the matrix H_t does not count anymore the number of positive entries in the rows of each element of $\mathcal{M}^{\leq t}$. The use of the boolean product between matrices (but not between a matrix and a vector) results in the disappearance of some associative properties: for example, it is *no longer true* that for any binary $n \times n$ matrices M, B_1, B_2, $B_1 e \leq B_2 e$ implies that $MB_1 e \leq MB_2 e$, as shown in Example 17. Notice that the implication would still hold true in the case that M is a binary *row-stochastic* matrix.

Example 17. Consider the following matrices:

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

It holds that $A_1 e \leq A_2 e$ but $(MA_1)e \geq (MA_2)e$, where the product used between matrices is the boolean product. Notice that, if it was the case that

$A_1 \leq A_2$, then it would hold that $MA_1 \leq MA_2$ for any NZ-matrix M, which in turn would imply that $A_1 e \leq A_2 e$ and $MA_1 e \leq MA_2 e$. Indeed the above example works because $A_1 \not\leq A_2$.

In the next section we show that we can define a function $\bar{K}(t) \geq K(t)$ where we can bound the length of its stagnations by a function $a(n) = O(n^2)$ and the magnitude of its jumps $\bar{K}(t_1) - \bar{K}(t_2) \geq 1/b$ by a linear function $b(n)$.

5.2 Approximating the synchronizing probability function

5.2.1 The function $\bar{\mathbf{K}}$

We can simplify Game 1 described at the beginning of this chapter by requiring Player B to consider just *deterministic* strategies, i.e. to choose his policy p among the vectors of the canonical basis $\mathscr{E}_n = \{e_1, e_2, \ldots, e_n\}$.

Definition 39. Given a binary primitive NZ-set $\mathcal{M}$, we define the *approximated synchronizing probability function* as the function

$$\bar{K}_{\mathcal{M}}(t) = \min_{e_i \in \mathscr{E}_n} \left\{ \max_{M \in \mathcal{M}^{\leq t}} e_i^T M \frac{e}{n} \right\} \ . \tag{5.19}$$

Cleary this new function is, at each time t, an upper bound on $K(t)$. We remind that the matrix H_t is defined in Definition 37.

Proposition 5.8. *The approximated SPF $\bar{K}(t)$ is given by the optimal value of the following linear program:*

$$\min_{e_i \in \mathcal{E}_n,\, k} \ \frac{k}{n} \quad s.t. \quad e_i^T H_t \leq k e^T \ . \tag{5.20}$$

It is also given by

$$\bar{K}_{\mathcal{M}}(t) = \frac{1}{n} \min_i \Big\{ \max\big\{ H_t[i,:] \big\} \Big\} \ . \tag{5.21}$$

Furthermore, $\bar{K}_{\mathcal{M}}(t) \geq K_{\mathcal{M}}(t)$ for every $t \geq 0$ and $\min\{t : \bar{K}_{\mathcal{M}}(t) = 1\} \leq exp(\mathcal{M})$.

Proof. Trivial. □

In this case a dual formulation of the linear program (5.20) as in Theorem 5.2 is no more possible, so in this simplified game Player B needs to keep his choice secret. Figure 5.9 shows, for each matrix set $\mathcal{M}_0$, $\mathcal{M}_1$, $\mathcal{M}_2$ respectively in Equations (5.3), (5.4) and (5.5), both the functions $K(t)$ and $\bar{K}(t)$. In view of Equation (5.21), the function $\bar{K}(t)$ takes values in the set $\{j/n : j \in [n]\}$. It then holds that

$$\bar{K}(t_1) > \bar{K}(t_2) \quad \Rightarrow \quad \bar{K}(t_1) - \bar{K}(t_2) \geq 1/n \ . \tag{5.22}$$

The following theorem shows that we can upper bound the length of the stagnations of $\bar{K}$ by a linear function in *almost* all the cases. We denote with $\bar{P}_t \subseteq \mathscr{E}_n$ the set of the optimal solutions of the linear program (5.20); it clearly holds that $1 \leq |\bar{P}_t| \leq n$.

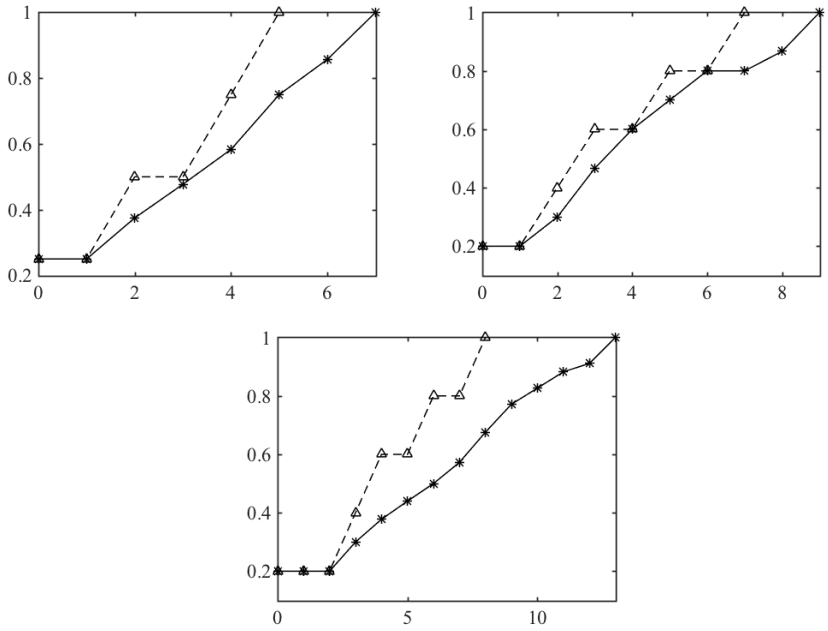


Figure 5.9: The functions $K(t)$ (solid line) and $\bar{K}(t)$ (dashed line) of the sets $\mathcal{M}_0$ in Equation (5.3) (left picture), $\mathcal{M}_1$ in Equation (5.4) (central picture) and $\mathcal{M}_2$ in Equation (5.5) (right picture).

Theorem 5.9. *Let $\mathcal{M}$ be a binary primitive NZ-set and $t \in \mathbb{N}$ such that $\bar{K}(t) < 1$ and $\bar{K}(t) = \bar{K}(t+1) = k/n$ for some $k \in [n]$. Then it holds that:*

1. *if $|\bar{P}_t| < n$, $\bar{K}(t+n-1) > \bar{K}(t)$.*
2. *if $|\bar{P}_t| = n$, $\bar{K}_{\mathcal{M}}(t + \frac{n^2(k-1)}{2k} + n) > \bar{K}_{\mathcal{M}}(t)$.*

In particular, $\bar{K}(n) > \bar{K}(0) = 1/n$.

Proof. 1. If $\bar{K}(t) = \bar{K}(t+1)$, then $\bar{P}_{t+1} \subseteq \bar{P}_t$. By the same reasoning used in the proof of Proposition 5.7, it holds that for any binary row-stochastic matrix R s.t. $R \leq M$ for a matrix $M \in \mathcal{M}$, $R^T \bar{P}_{t+1} \subseteq \bar{P}_t$. We now claim that $\bar{P}_{t+1} \subsetneq \bar{P}_t$. Indeed, suppose by contrary that $\bar{P}_{t+1} = \bar{P}_t$. This means that $R^T \bar{P}_t \subseteq \bar{P}_t$ for any binary row-stochastic matrix R dominated by an element of $\mathcal{M}$, which in turn implies that for any product $R_1 \cdots R_l$ of binary row-stochastic matrices dominated by matrices in $\mathcal{M}$, it holds that $(R_1 \cdots R_l)^T \bar{P}_t \subseteq \bar{P}_t$. The set of all the binary row-stochastic matrices dominated by a least a matrix in $\mathcal{M}$ is the automaton $Aut(\mathcal{M})$ (see Definition 24): since $\mathcal{M}$ is primitive, the automaton $Aut(\mathcal{M})$ is synchronizing by Theorem 3.19, and so there exists a product $\bar{R} = R_{i_1} \cdots R_{i_s}$ of its letters that has an all-ones column, say in position j. Since $\bar{R}^T \bar{P}_t = \{e_j\}$, we have that $e_j \in \bar{P}_t$. The set $\mathcal{M}$ is irreducible and so the automaton $Aut(\mathcal{M})$ is strongly connected (see Proposition 3.3); this implies that for any $l \neq j$ there exists a product W_l of the matrices in $Aut(\mathcal{M})$ such that $W_l[j,l] = 1$ and so the product $\bar{R}W_l$ has an all-ones column in position l. In view of the fact that $(\bar{R}W_l)^T \bar{P}_t = \{e_l\}$ and $(\bar{R}W_l)^T \bar{P}_t \subseteq \bar{P}_t$, it holds that $e_l \in \bar{P}_t$ for every $l \in [n]$, which contradicts the hypothesis. This means that $\bar{P}_{t+1} \subsetneq \bar{P}_t$ and so $|\bar{P}_{t+1}| < |\bar{P}_t| < n$. Now, if $\bar{K}(t+2) > \bar{K}(t+1)$ we are done; otherwise we can iterate the same argument on $\bar{P}_{t+1}$ thus proving that

$|\bar{P}_{t+2}| < |\bar{P}_{t+1}|$. It follows that if $\bar{K}(t) = \bar{K}(t+1) = \cdots = \bar{K}(t+n-2)$, then $|\bar{P}_{t+n-2}| = 1$, and since the set of the optimal solutions cannot be empty, it must hold that $\bar{K}(t+n-1) > \bar{K}(t)$.
2. Let $\bar{s} > 0$ be the maximal integer such that $\bar{K}(t) = \bar{K}(t+1) = \cdots = \bar{K}(t+\bar{s})$ and $|\bar{P}_t| = |\bar{P}_{t+1}| = \cdots = |\bar{P}_{t+\bar{s}}| = n$. We can apply item *1.* at time $t+d+1$, so it holds that $\bar{K}(t+\bar{s}+n) > \bar{K}(t)$. We now show that $\bar{s} \leq n^2(k-1)/2k$ and so the thesis follows. By the definition of $\bar{s}$, each of the matrices $H_t, H_{t+1}, \ldots, H_{t+\bar{s}}$ (see Definition 37) has the following properties: all the entries are $\leq k$ and in each row there is an entry equal to k. This is equivalent to say that, for every $u = 0, \ldots, \bar{s}$, all the matrices in $\mathcal{M}^{\leq t+u}$ have at most k positive entries in each row and for all $i \in [n]$, there exists a matrix in $\mathcal{M}^{\leq t+u}$ that has exactly k positive entries in the i-th row. We now exhibit a product in $\mathcal{M}^{\leq t+n^2(k-1)/2k+1}$ that has a row with $k+1$ positive entries, which implies that $\bar{s} \leq n^2(k-1)/2k$. For every $i \in [n]$, let $W_i \in \mathcal{M}^{\leq t}$ be a product with k positive entries in the i-th row. We claim that there are at least $a_k = \lceil (n-k)/k \rceil$ rows of W_i whose support is not contained in $W_i[i,:]$; this comes from the fact that W_i is NZ and each row does not have more than k positive entries. Let $r^i_1, \ldots, r^i_{a_k}$ be the indices of these rows. Note that a product L such that $L[q,i] = 1 = L[q, r^i_j]$ for some $j \in [a_k]$ and $i, q \in [n]$ would imply that LW_i has a row with at least $k+1$ positive entries. We now want to estimate the minimal length of L over all $i, q \in [n]$ and $j \in [a_k]$[1]. To do so we make use of the square graph $\mathcal{SG}(\mathcal{M}^T) = (V, E)$ associated to $\mathcal{M}^T$ (see Definition 13); in view of Remark 2, it holds that path in $\mathcal{SG}(\mathcal{M}^T)$ from (i,j) to (q,q) labeled by $M^T_{l_1} \ldots M^T_{l_u}$ means that $M_{l_u} \cdots M_{l_1}[q,i] > 0$ and $M_{l_u} \cdots M_{l_1}[q,j] > 0$. Consequently, we need to estimate the minimal length on $i, q \in [n]$ and $j \in [a_k]$ of the shortest path in $\mathcal{SG}(\mathcal{M}^T)$ connecting (i, r^i_j) to (q,q). The vertex set V has cardinality $n(n+1)/2$ and it has exactly n vertices of type (q,q); furthermore, in the set of vertices $\{(i, r^i_j)\}^{i \in [n]}_{j \in [a_k]}$ there are at least $na_k/2$ different elements. Therefore, this minimal length is at most of $n(n+1)/2 - na_k/2 - n + 1 = (n^2(k-1)/2k) + 1$. This means that there exists a product $L \in \mathcal{M}^{\leq (n^2(k-1)/2k)+1}$ and $i \in [n]$ such that LW_i has a row with at least $k+1$ positive entries, and so $\bar{s} \leq (n^2(k-1)/2k)$. This in turn implies that $\bar{K}(t + (n^2(k-1))/2k + n) > \bar{K}(t)$.

Lastly, we have to prove that $\bar{K}(n) > \bar{K}(0)$. If $\bar{K}(1) > \bar{K}(0)$, we can conclude since $\bar{K}$ is nondecreasing. Suppose now that $\bar{K}(1) = \bar{K}(0) = 1/n$; we claim that $|\bar{P}_1| < n$ and so $\bar{K}(1+n-1) = \bar{K}(n) > \bar{K}(1)$ by item *1.* Since the set $\mathcal{M}$ is primitive and NZ, there must exist a matrix in $\mathcal{M}$ with at least two positive entries in the same row, as otherwise $\mathcal{M}$ would be a set of permutation matrices, which is never primitive. This means that the matrix H_1 (see Definition 37) must have an entry ≥ 2, say in row i and column j, so $(e_i^T H_1)_j \geq 2$. Since $\bar{K}(1) = 1/n$, by the representation of $\bar{K}$ via the linear program (5.20), it follows that $e_i \notin \bar{P}_1$ and so $|\bar{P}_1| < n$. □

Corollary 5.10. *Let $\mathcal{M}$ be a binary primitive NZ-set and $t, s \in \mathbb{N}$ such that $\bar{K}(t) < 1$, $\bar{K}(t) = \bar{K}(t+s)$, $|\bar{P}_t| = n$ and $|\bar{P}_{t+s}| < n$. Then it holds that $\bar{K}(t+s+n-1) > \bar{K}(t)$.*

Proof. It suffices to apply Theorem 5.9 at time $t+s$. □

[1]For every $i, q \in [n]$ and $j \in [a_k]$ the product L exists by Theorem 3.10 applied on $\mathcal{M}^T$.

Corollary 5.10 shows that in order to improve the upper bound on the length of the stagnations of $\bar{K}$ when $|\bar{P}_t| = n$, it suffices to estimate the value $\bar{s} = \max\{s > 0 : |\bar{P}_{t+s}| = n \text{ and } \bar{K}(t) = \bar{K}(t+s)\}$. In particular, if $\bar{s}$ is linear in n, then so is the length of the stagnation. In Subsection 5.2.2 we will see how to compute an upper bound on $\bar{s}$.

The function $\bar{K}$ is particularly of interest because if we manage to find an upper bound on $\min_t\{\bar{K}(t) = 1\}$, then we would have an upper bound on $exp(\mathcal{M})$, as formalized by the following proposition.

Proposition 5.11. *If there exists a function $U(n)$ such that, for any primitive NZ-set $\mathcal{M}$ of $n \times n$ matrices, it holds that $\min\{t : \bar{K}_{\mathcal{M}}(t) = 1\} \leq U(n)$, then $exp_{NZ}(n) \leq 2U(n)$.*

Proof. Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be a binary primitive NZ-set of $n \times n$ matrices and let $t_0 = \min\{t : \bar{K}_{\mathcal{M}}(t) = 1\}$. By Equation (5.21), we have that every row of H_{t_0} has an entry equal to n, which means that for every $i = 1, \ldots, n$, there exists a matrix $M_i \in \mathcal{M}^{\leq t_0} \subset \mathcal{M}^{\leq U(n)}$ that has the i-th row entrywise positive. Since the function $U(n)$ depends only on n, we can apply the same reasoning to the set $\mathcal{M}^T = \{M_1^T, \ldots, M_m^T\}$: for every $i = 1, \ldots, n$, there exists a matrix $N_i \in (\mathcal{M}^T)^{\leq U(n)}$ that has the i-th row entrywise positive. The matrix $N_i^T M_i$ is a positive product of elements of $\mathcal{M}$ of length at most $2U(n)$, therefore $exp_{NZ}(n) \leq 2U(n)$. □

Our numerical results suggest that the length of the stagnations of $\bar{K}$ when $|\bar{P}_t| = n$ should be much shorter:

Conjecture 5.12. *There exists a linear function $f(n)$ such that, for every binary primitive NZ-set $\mathcal{M}$ of $n \times n$ matrices and fir every $t \in \mathbb{N}$ s.t. $\bar{K}_{\mathcal{M}}(t) < 1$, $\bar{K}_{\mathcal{M}}(t) = \bar{K}_{\mathcal{M}}(t+1)$ and $|\bar{P}_t| = n$, it holds that*

$$\bar{s} = \max\{s > 0 : |\bar{P}_{t+s}| = n \text{ and } \bar{K}_{\mathcal{M}}(t) = \bar{K}_{\mathcal{M}}(t+s)\} \leq f(n).$$

If Conjecture 5.12 is true, it would lead to a quadratic upper bound on $exp_{NZ}(n)$ and on the reset threshold of a certain class of automata, as stated by the following propositions.

Proposition 5.13. *If Conjecture 5.12 is true, then it holds that $exp_{NZ}(n) = O(n^2)$.*

Proof. Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be a binary primitive NZ-set of $n \times n$ matrices. If Conjecture 5.12 is true, then by Corollary 5.10 it holds that $\bar{K}(t + f(n) + n - 1) > \bar{K}(t)$ for every $t \in \mathbb{N}$ such that $\bar{K}(t) < 1$. This, combined with Equation (5.22), implies that $\min\{t : \bar{K}_{\mathcal{M}}(t)\} = O(n^2)$. By applying Proposition 5.11, we conclude. □

Proposition 5.14. *If Conjecture 5.12 is true, then for any synchronizing DFA $\mathcal{A}$ on n states for which there exists a binary NZ-set $\mathcal{M}$ such that $\mathcal{A} = Aut(\mathcal{M})$ (see Definition 24) it holds that $rt(\mathcal{A}) = O(n^2)$.*

Proof. Straightforward by Proposition 5.13 and Theorem 3.19. □

In the next section we see how to improve the upper bound on the length of the stagnations of $\bar{K}$ when $|\bar{P}_t| = n$, by introducing the *k-rendezvous time*.

5.2.2 The k-rendezvous time

The synchronization problem for automata is about finding the length of the shortest word mapping the whole set of states onto one single state. We can weaken this request by asking what is the length of the shortest word mapping a set of $k \geq 2$ states onto one single state. In the matrix framework, we are asking what is the lenght of the shortest product having a column with k positive entries. The case $k = 2$ is trivial, as any synchronizing automaton has a letter mapping two states onto one; for $k = 3$ Gonze and Jungers [51] presented a quadratic upper bound in the number of the states of the automaton while, to the best of our knowledge, the cases $k \geq 4$ are still open. Clearly, the case $k = n$ is the problem of computing the reset threshold.

Here we extend the above described problem to primitive sets of NZ-matrices by introducing the *k-rendezvous time*, defined as follows:

Definition 40. Let $\mathcal{M}$ be a primitive NZ-set of $n \times n$ matrices and $2 \leq k \leq n$ an integer. The *k-rendezvous time* (k-RT) of the set $\mathcal{M}$, denoted by $rt_k(\mathcal{M})$, is the length of the shortest product of elements of $\mathcal{M}$ having a row or a column with k positive entries. We set

$$rt_k(n) = \max\{rt_k(\mathcal{M}) \colon \mathcal{M} \text{ is a primitive NZ-set of } n \times n \text{ matrices.}\}$$

Clearly, $rt_n(\mathcal{M}) = \min\{pc(\mathcal{M}), pc(\mathcal{M}^T)\}$ which implies, by Corollary 3.23, that $rt_n(\mathcal{M}) = \min\{rt_k(Aut(\mathcal{M})), rt_k(Aut(\mathcal{M}^T))\}$, where we denote with $rt_k(\mathcal{A})$ the length of the shortest word in the automaton $\mathcal{A}$ mapping a set of k states onto one.

We are interested in finding an upper bound on $rt_k(n)$ that depends just on n and k. We will show that for any fixed k small enough ($k \leq \sqrt{n}$), $rt_k(n)$ is *linear* in n, result that has not yet being proved for automata.

Definition 41. Let $2 \leq k \leq n-1$ be an integer and M be an $n \times n$ nonnegative matrix having all its rows and columns of weight[2] at most k. Suppose that there exists a column c of M with weight equal to k. We define:

- $a_k^n(M) = |\{v : v \text{ column of } M \text{ and } supp(v) \nsubseteq supp(c)\}|$;
- $a_k^n = \min\{a_k^n(M) : M \in \mathbb{R}^{n\times n},\ |supp(M[:,j])| \leq k\ \forall j,\ |supp(M[i,:])| \leq k\ \forall i,\ \exists \bar{j}\ s.t.\ |supp(M[:,\bar{j}])| = k\}$.

It holds that $a_k^n \geq 1$ since $k \leq n-1$ and the set is NZ.

Lemma 5.15. *Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be primitive NZ-set of $n \times n$ matrices and let $M \in \mathcal{M}^t$ a product with a column c of weight at least k. Then, there exists a matrix $D \in \mathcal{M}^{\leq t+\frac{n}{2}(n+1-a_k^n)}$ such that D has a column of weight at least $k+1$.*

Proof. If M has a column of weight $\geq k+1$, the lemma is proved. Suppose that c has weight equal to k and all the other columns have weight $\leq k$. The set $\mathcal{M}$ is primitive, so its underlying digraph is strongly connected. Since $M \in \mathcal{M}^t$ has the column c of weight k, for every $i = 1, \ldots, n$, there exists a matrix $W_i \in \mathcal{M}^{t+n-1}$ such that the i-th column of W_i, that we denote by w_i, has weight equal to k. For each $i \in [n]$, let $\{v_1^i, v_2^i, \ldots, v_{a_k^n}^i\}$ be the indices of a_k^n

[2] We remind that the *weight* of a nonnegative vector v is the number of its positive entries.

columns of W_i whose supports are not contained in $supp(w_i)$ and let $\mathcal{SG}(\mathcal{M})$ be the square graph of $\mathcal{M}$ (see Definition 13). We have seen in Remark 2 that a path from a vertex (i,j) to a vertex (q,q) in $\mathcal{SG}(\mathcal{M})$ labeled by $P = M_{s_1} \dots M_{s_l}$ means that, for any NZ-matrix L, $LP[:,q] \geq L[:,i] + L[:,j]$. Therefore, a path P from a vertex (i, w_j^i) to a vertex (q,q) for some $j \in [a_k^n]$ and $q \in [n]$, would result in a matrix W_iP with the q-th column of weight at least $k+1$. We want to estimate $\min\{dist((i,w_j^i),(q,q)) : q \in [n],\, i \in [n],\, j \in [a_k^n]\}$, where $dist(v_1,v_2)$ is the length of the shortest path from vertex v_1 to vertex v_2 in the square graph $\mathcal{SG}(\mathcal{M})$. In the set $\{(i,w_j^i)\}_{i,j}$ there are at least $na_k^n/2$ different elements and the number of non-sigleton vertices of $\mathcal{SG}(\mathcal{M})$ is $n(n-1)/2$. Therefore, there must be a path from a vertex in $\{(i,w_j^i)\}_{i,j}$ to a singleton (q,q) of length at most $n(n-1)/2 - na_k^n/2 + 1$. Since $W_i \in \mathcal{M}^{t+n-1}$ for every $i \in [n]$, this implies that there exists a matrix $D \in \mathcal{M}^{\leq t+n+\frac{n}{2}(n-1-a_k^n)}$ that has a column of weight at least $k+1$. □

We say that a product M of elements from the matrix set $\mathcal{M}$ *attains* the $rt_k(\mathcal{M})$ if its length is equal to $rt_k(\mathcal{M})$ and it has a column or a row with k positive entries.

Proposition 5.16. *Let $n \geq 2$ be an integer. Consider the sequence $B_n(k)$ for $k = 2, \dots, n-1$ defined by the following recursion:*

$$\begin{cases} B_n(2) = 1, \\ B_n(k+1) = B_n(k) + \dfrac{n(n+1-a_k^n)}{2} \quad \text{for } 2 \leq k \leq n-1. \end{cases} \tag{5.23}$$

Then $rt_k(n) \leq B_n(k)$.

Proof. We proceed by induction on k. If $k = 2$, then $rt_2(n) = 1$ because any primitive NZ-set has a matrix with a row or a column with two positive entries; indeed otherwise the set would be made of permutation matrices and so it would not be primitive. Suppose now that $rt_k(n) \leq B_n(k)$; we prove that $rt_{k+1}(n) \leq B_n(k+1)$. In particular, we will prove that $rt_{k+1}(\mathcal{M}) \leq B_n(k+1)$ for any primitive NZ-set $\mathcal{M}$ of $n \times n$ matrices. Let fix the set $\mathcal{M}$ and let M be a product that attains the $rt_k(\mathcal{M})$. Suppose first that M has a column of weight k. By Lemma 5.15, there exists a matrix $D \in \mathcal{M}^{\leq rt_k(\mathcal{M})+n+\frac{n}{2}(n-1-a_k^n)}$ such that D has a column of weight at least $k+1$, so $rt_{k+1}(\mathcal{M}) \leq rt_k(\mathcal{M}) + n + \frac{n}{2}(n-1-a_k^n) \leq B_n(k+1)$ by the inductive hypothesis.

Suppose now that M has a row of weight k. Then M^T is a product of matrices from $\mathcal{M}^T$ with a column of weight k and all the other columns and rows of weight at most k. By Lemma 5.15, there exists a matrix $C \in (\mathcal{M}^T)^{\leq rt_k(\mathcal{M}^T)+n+\frac{n}{2}(n-1-a_k^n)}$ such that C has a column of weight at least $k+1$. Since $rt_k(\mathcal{M}^T) = rt_k(\mathcal{M})$, then C^T is a product of elements from $\mathcal{M}$ of length at most $rt_k(\mathcal{M}) + \frac{n}{2}(n+1-a_k^n)$ and with a row of weight $\geq k+1$, so $rt_{k+1}(\mathcal{M}) \leq rt_k(\mathcal{M}) + \frac{n}{2}(n+1-a_k^n) \leq B_n(k+1)$ by the inductive hypothesis. □

Note that Proposition 5.16 also holds when a_k^n is replaced by a function $l(k,n)$ such that $l(k,n) \leq a_k^n$ for every $n \geq 2$ and $2 \leq k \leq n-1$. Our goal is to find an explicit expression for $l(k,n)$; we propose two methods here below. We fix an $n \times n$ NZ-matrix M having all its rows and columns of weight at most k and such that there exists $\bar{j} \in [n]$ such that $|supp(c)| = k$ for $c = M[:,\bar{j}]$. For simplicity, we can suppose without loss of generality that the column c

has the first k entries equal to 1 and the last $n-k$ entries equal to 0.

Method 1. We want to find an upper bound on the number of columns of M whose support is a subset of $supp(c)$. We remind that M is NZ, so each column must have at least a positive entry. Since each row has at most k positive entries, there must be at most $k(k-1)$ columns different from c whose support is a subset of $supp(c)$. Therefore there are at least $n-k(k-1)-1$ columns whose support is not a subset of $supp(c)$, so $n-k(k-1)-1 \leq a_k^n$.

Method 2. Since the matrix M is NZ, every row must have at least one positive entry, and by hypothesis every column can have at most k positive entries. Therefore, there must be at least $\lceil (n-k)/k \rceil$ columns in M whose support is not a subset of $supp(c)$. This implies that $\lceil (n-k)/k \rceil \leq a_k^n$.

Suppose $n \geq 4$. We define $f_n(k) : [2, \ldots, n-1] \to \mathbb{N}$ as the following function:

$$\begin{aligned} f_n(k) &= \max_{2 \leq k \leq n-1} \{n-k(k-1)-1, \lceil (n-k)/k \rceil, 1\} \\ &= \begin{cases} n-k(k-1)-1 & \text{if } 2 \leq k \leq \lfloor \sqrt{n} \rfloor, \\ \lceil (n-k)/k \rceil & \text{if } \lfloor \sqrt{n} \rfloor + 1 \leq k \leq \lfloor n/2 \rfloor, \\ 1 & \text{if } \lfloor n/2 \rfloor + 1 \leq k < n. \end{cases} \end{aligned} \tag{5.24}$$

Clearly, $f_n(k) \leq a_k^n$ for all $n \geq 4$ and $2 \leq k \leq n-1$. The recursion (5.23) with a_k^n replaced by $f_n(k)$ now reads as:

$$\begin{cases} B_n(2) = 1, & \\ B_n(k+1) = B_n(k) + n + \dfrac{nk(k-1)}{2} & \text{if } 2 < k \leq \lfloor \sqrt{n} \rfloor, \\ B_n(k+1) = B_n(k) + n + \dfrac{n^2(k-1)}{2k} & \text{if } \lfloor \sqrt{n} \rfloor + 1 \leq k \leq \lfloor n/2 \rfloor, \\ B_n(k+1) = B_n(k) + \dfrac{n^2}{2} & \text{if } \lfloor n/2 \rfloor + 1 \leq k < n, \end{cases} \tag{5.25}$$

where we had considered $f_n(k) = (n-k)/k$ for $\lfloor \sqrt{n} \rfloor + 1 \leq k \leq \lfloor n/2 \rfloor$ to ease the computation.
Let $k^* = \lfloor \sqrt{n} \rfloor$ and $k^{**} = \lfloor n/2 \rfloor$. It is easy to verify that the function below solves the recursion (5.25):

$$\begin{cases} B_n(2) = 1, & \\ B_n(k) = n(k^3 - 3k^2 + 8k - 12)/6 + 1 & \text{if } 2 \leq k \leq k^*, \\ B_n(k) = B_n(k^*) + \dfrac{n(n+2)(k-k^*)}{2} - \dfrac{n^2}{2} \displaystyle\sum_{j=k^*}^{k-1} \frac{1}{j} & \text{if } k^* < k \leq k^{**}, \\ B_n(k) = B_n(k^{**}) + (k-k^{**})n^2/2 & \text{if } k^{**} < k \leq n. \end{cases} \tag{5.26}$$

The above Equation (5.26) thus represents an upperbound for $rt_k(n)$ for any $n \geq 4$ and $2 \leq k \leq n$. We can therefore state the following proposition:

Proposition 5.17. *For any fixed integer $k \leq \sqrt{n}$, $rt_k(n)$ is linear in n. For any fixed integer $k \leq n/2$, $rt_k(n)$ is at most quadratic in n.*

Proof. Trivial by Equation (5.26). □

We now show how to use the term a_k^n to improve the upper bound on the length of the stagnations of the approximated SPF $\bar{K}$ (see Subsection 5.2.1).

Proposition 5.18. *Let $\mathcal{M}$ be a primitive NZ-set of $n \times n$ matrices and let $t \geq 2$ be an integer such that $\bar{K}(t) < 1$, $\bar{K}(t) = \bar{K}(t+1) = k/n$ for some $k \in \{2, \ldots, n-1\}$, and $|\bar{P}_t| = n$. Then it holds that*

$$\bar{K}_{\mathcal{M}}(t + n(n+1-a_k^n)/2) > \bar{K}_{\mathcal{M}}(t). \tag{5.27}$$

Proof. By Corollary 5.10, it suffices to show that

$$\bar{s} = \min\{s > 0 : |\bar{P}_{t+s}| = n \text{ and } \bar{K}(t) = \bar{K}(t+s)\} \leq n(n-1-a_k^n)/2 + 1.$$

By hypothesis, the matrix H_t (see Definition 37) has in every row at least one entry equal to k, while all the other entries have magnitude $\leq k$. This means that for all $i \in [n]$, there exists a matrix in $\mathcal{M}^{\leq t}$ having the i-th row with k positive entries. In view of this, Lemma 5.15 implies that there exists a matrix in $\mathcal{M}^{\leq t+n(n-1-a_k^n)/2+1}$ with a row of weight at least $k+1$, which in turn implies that $H_{t+n(n-1-a_k^n)/2+1}$ has a row with an entry $\geq k+1$. Consequently, $|\bar{P}_{t+n(n-1-a_k^n)/2+1}| < n$ and so $\bar{s} \leq t + n(n-1-a_k^n)/2 + 1$. □

Notice that Proposition 5.18 still holds if we substitute the term a_k^n with any function $l(k,n)$ such that $l(k,n) \leq a_k^n$ for any $n \geq 2$ and $2 \leq k \leq n-1$. In particular, we can substitute a_k^n with the function $f_n(k)$ defined in Equation (5.24) in order to obtain an upper bound on $exp_{NZ}(n)$; despite this upper bound is not better than the ones already known, we report it here below for the sake of completeness.

Corollary 5.19. *It holds that:*

$$exp_{NZ}(n) \leq 2\left(\frac{1}{2}n^3 - \frac{1}{3}n^2\sqrt{n} - \frac{1}{4}n^2\log\left(\frac{n}{4}\right) + \frac{1}{2}n^2 + \frac{1}{3}n\sqrt{n} + \frac{n}{2} - 2\right) .$$

Proof. By Proposition 5.11, we just need to prove that

$$\min\{t : \bar{K}(t) = 1\} \leq \frac{1}{2}n^3 - \frac{1}{3}n^2\sqrt{n} - \frac{1}{4}n^2\log\left(\frac{n}{4}\right) + \frac{1}{2}n^2 + \frac{1}{3}n\sqrt{n} + \frac{n}{2} - 2 .$$

In view of Theorem 5.9, Proposition 5.18 and the definition of $f_n(k)$ in Equation (5.24), it holds that

$$\min\{t : \bar{K}(t) = 1\} \leq 2n - 2 + \sum_{k=2}^{n-1} \frac{n}{2}(n+1-a_k^n) \leq$$
$$\leq \frac{n^3}{2} - \frac{n^2}{2} + n - 2 - \frac{n}{2}\left(\sum_{k=2}^{\lfloor\sqrt{n}\rfloor}(n-k(k-1)+1) + \sum_{k=\lfloor\sqrt{n}\rfloor+1}^{\lfloor n/2\rfloor}\left(\frac{n}{k}-1\right) + \sum_{k=\lfloor n/2\rfloor+1}^{n-1} 1\right).$$

By using the following bounds, we get the desired upper bound on $\min\{t : \bar{K}(t) = 1\}$:

- for any $x \in \mathbb{R}$, $\lfloor x \rfloor \leq x \leq \lfloor x \rfloor + 1$;

- if $\mathcal{S}_N = \sum_{k=1}^{N} \frac{1}{k}$, then $\log(N+1) \leq \mathcal{S}_N \leq (\log N) + 1$;

- $\sum_{k=1}^{N} k = \frac{N(N+1)}{2}$ and $\sum_{k=1}^{N} k^2 = \frac{N(N+1)(2N+1)}{6}$.

□

One of the reasons why the upper bound on $exp_{NZ}(n)$ in Corollary 5.19 does not improve the ones already known is that $f_n(k) = 1$ for $k \geq n/2$, which means that we are estimating the number of columns that we can sum up to the column c of weight k in order to increase its weight with 1, i.e. with the smallest possible value. This is not optimal because, if we are in the worst case where we can sum up to c just one column c', this column has to have $n - k$ positive entries in the rows corresponding to the zero entries of c, so by summing the columns c and c' we would obtain a positive column. We would then directly jump to the n-RT, instead to the $(k+1)$-RT as we are considering now. We believe that taking into account this fact would lead to an improvement on the upper bound of $exp_{NZ}(n)$.

We conclude this section by showing that $a_k^n = f_n(k)$, as we can always find an $n \times n$ NZ-matrix M such that $a_k^n(M) = f_n(k)$. This implies that there is no hope to improve the bounds on $rt_k(n)$ and especially $exp_{NZ}(n)$ by refining the estimate on a_k^n and new strategies need to be implemented, as the one just mentioned above.

Proposition 5.20. *For every $n \geq 2$ and $2 \leq k \leq n-1$, it holds that*

$$a_k^n = f_n(k).$$

Proof. In view of Equation (5.24) it holds that $a_k^n \geq f_n(k)$, so it just suffices to prove that for every $n \geq 2$ and $2 \leq k \leq n-1$, $a_k^n \leq f_n(k)$. This means that for every $n \geq 2$ and $2 \leq k \leq n-1$ we need to exhibit an $n \times n$ NZ-matrix M such that $a_k^n(M) = f_n(k)$. The case $k \geq \lceil n/2 \rceil$ trivially holds as $f_n(k) = 1$, which is the minimal value possible for $a_k^n(M)$.

Suppose now that $k \leq \lfloor \sqrt{n} \rfloor$; we need to prove that there exists an $n \times n$ NZ-matrix M with every columns and rows of weight $\leq k$ and with a column c of weight k such that there are exactly $a_n^k = n - k(k-1) - 1$ columns in M whose supports are not subsets of $supp(c)$. Let $s \in \mathbb{N}$ and $0 \leq r < k$ such that $n = sk + r$. We first consider the case $r > 0$: for every $i \in [s[$, let B_i be the $k \times (s+1)$ matrix having the i-th column entrywise equal to 1 and all the other entries equal to zero and let R_{s+1} be the $r \times (s+1)$ matrix having the $(s+1)$-th column entrywise equal to 1 and all the other entries equal to zero. For every $j \in [k]$, let C_j be the $k \times (k-1)$ matrix having the j-th row entrywise equal to 1 and all the other entries equal to zero. Finally, let e_i be the i-th element of the canonical basis and $v = n - k(k-1) - s - 1$. The following $n \times n$ NZ-matrix satisfies the properties required (the empty spaces have to be considered zero entries):

$$M = \left[\begin{array}{c|cccc|cccc} B_1 & C_1 & C_2 & \cdots & C_k & & & & \\ \hline B_2 & & & & & e_1 & e_2 & \cdots & e_v \\ \vdots & & & & & & & & \\ B_s & & & & & & & & \\ R_{s+1} & & & & & & & & \end{array}\right]$$

Indeed, all the columns and rows of M have weight of at most k. Let c be the first column of M: it has exactly k positive entries in the first k rows. It is easy to see that the columns of M whose supports are not subsets of $supp(c)$ are the columns indexed by $j = 2, \ldots, s+1$ and the last v columns. Therefore, the number of columns whose supports are not subsets of $supp(c)$ is equal to $s + v = n - k(k-1) - 1$, which is the value of $f_n(k)$.

We now consider the case $r = 0$. We set $v = n - k(k-1) - s$; then the following $n \times n$ NZ-matrix satisfies the properties required:

$$M = \left[\begin{array}{c|cccc|cccc} B_1 & C_1 & C_2 & \cdots & C_k & & & & \\ \hline B_2 & & & & & e_1 & e_2 & \cdots & e_v \\ \vdots & & & & & & & & \\ B_s & & & & & & & & \end{array}\right]$$

Indeed, as before, all the columns and rows of M have weight of at most k. Let c be the first column of M: it is easy to see that the columns of M whose supports are not subsets of $supp(c)$ are the columns indexed by $j = 2, \ldots, s$ and the last v columns. Therefore, the number of columns whose supports are not subsets of $supp(c)$ is equal to $s + v = n - k(k-1) - 1$, which is the value of $f_n(k)$.

Suppose now that $\lfloor \sqrt{n} \rfloor + 1 \leq k \leq \lfloor n/2 \rfloor$: we need to prove that there exists an $n \times n$ NZ-matrix M with every columns and rows of weight $\leq k$ and with a column c of weight k such that there are exactly $a_n^k = \lceil n - k/k \rceil$ columns in M whose supports are not subsets of $supp(c)$. Firstly, we consider the case where $n = sk + r$ for $0 < r < k$: let B_i, R_{s+1} and C_i as defined in the case $k \leq \lfloor \sqrt{n} \rfloor$ and let D_k be the $k \times (n - k(k-1) - s - 1)$ matrix with the k-th row entrywise equal to 1 and all the other entries equal to zero. The following matrix satisfies the properties required:

$$M = \left[\begin{array}{c|cccc|c} B_1 & C_1 & C_2 & \cdots & C_{k-1} & D_k \\ \hline B_2 & & & & & \\ \vdots & & & & & \\ B_s & & & & & \\ R_{s+1} & & & & & \end{array}\right]$$

Indeed, all the columns and rows of M have weight of at most k. Let c be the first column of M: it has exactly k positive entries in the first k rows. The columns of M whose supports are not subsets of $supp(c)$ are the columns indexed by $j = 2, \ldots, s+1$. Therefore, the number columns whose supports are not subsets of $supp(c)$ is equal to $s = n/k - r/k = \lfloor n/k \rfloor = \lceil (n-k)/k \rceil$, which is the value of $f_n(k)$.

Finally, suppose that $r = 0$. Then the following matrix satisfies the properties required:

$$M = \left[\begin{array}{c|cccc|c} B_1 & C_1 & C_2 & \cdots & C_{k-1} & D_k \\ \hline B_2 & & & & & \\ \vdots & & & & & \\ B_s & & & & & \end{array}\right]$$

as all the columns and rows of M have weight of at most k and if we consider c as the first column of M, the columns of M whose supports are not subsets

of $supp(c)$ are the columns indexed by $j = 2, \ldots, s$. Therefore, the number of columns whose supports are not subsets of $supp(c)$ is equal to $s - 1 = (n/k) - 1 = (n - k)/k$, which is the value of $f_n(k)$. □

We have already mentioned how to possibly improve the upper bound on $exp_{NZ}(n)$ by taking into account not only the number of columns a_k^n that we can add to c but also the weight w of the new column that we form, thus directly jumping from the k-RT to the $(k + w)$-RT. Another idea could be to use the linear upper bound on the k-RT for $k \leq \sqrt{n}$ to get a better upper bound on the αk-RT, for $\alpha \in \mathbb{N}$.

Chapter 6

Conclusions and open problems

The concepts of primitive sets, column-primitive sets, synchronizing DFAs and directable NDFAs originated in different fields and have been developed by the respective communities almost independently. In this manuscript we have presented these notions in a unified framework, where we have highlighted and exploited the connection between each others. We have then tackled the primitivity phenomenon via two probabilistic approaches.

In the first part we have extended the binomial model and the uniform model of random graph theory (see Appendix A.1 and [18, 66]) to labeled directed multigraphs via the random models $\mathcal{B}_m(n,p)$ and $\mathcal{B}_m(n,M)$. We have showed that $p = (\log n + c)/n$ is a sharp threshold for $\mathcal{B}_m(n,p)$ with respect to the property of being primitive and that $M = n(\log n + c)$ is a threshold for $\mathcal{B}_m(n,M)$ with respect to the property of being primitive. We have also proved that these random models, when primitive, have small exponent with high probability. In particular, an uniformly sampled set of binary matrices is primitive and has exponent of order $O(n \log n)$ with high probability.

These results rely on our preliminary result stating that a random perturbed permutation set is primitive and with exponent of order $O(n \log n)$ with high probability. We have then showed that the same sharp threshold holds for $\mathcal{B}_m(n,p)$ with respect to the property of being column-primitive and that the length of its shortest scrambling product and of its shortest positive-column product is small with high probability. NDFAs can be modeled by $\mathcal{B}_m(n,p)$ and we have proved that $p = (\log n + c)/n$ is as well a sharp threshold with respect to the 3-directability property; regarding the 2-directability property, we have proved that it holds with high probability any time we are above the threshold. In particular, we have shown that with high probability an uniformly sampled NDFA of at least two letters has both a 2-directing word and a 3-directing word of length $O(n \log n)$, thus extending to directable NFDAs what was already known for synchronizing DFAs [79].

We have then presented a more involved randomized algorithm generating proper perturbed permutation sets, based on a recent characterization of primitive sets of NZ-matrices (Theorem 3.7): we have showed that with positive probability it generates primitive sets with quadratic exponent. These sets are one of the few examples of primitive sets with quadratic exponent that are known and we have showed that they can be transformed into new fami-

lies of slowly synchronizing DFAs, where we have proved that they have reset thresholds of order $\Omega(n^2/4)$ by providing closed formulas for their square graph diameter. To the best of our knowledge, this is one of the few cases where an extremal family of synchronizing DFAs does not resemble the Černý's one and that a constructive procedure for building proper synchronizing DFAs is presented.

In the second part we have embedded primitivity in a game-theoretical framework, where we have developed the *synchronizing probability function* for primitive sets.The SPF takes into account the speed at which a primitive set reaches its first positive product: numerical experiments have shown that its behavior seems smooth and regular (after a potential stagnation phase of length smaller than n), and it can thus be used to efficiently compute an approximation of the exponent. This approximation seems to improve the Eppstein heuristic in a fairly good amount of cases.

We have then introduced the function $\bar{K}(t)$, which is an upper bound on the SPF, and we have showed that it cannot remain constant for too long. We have also proved that an estimate of the time at which $\bar{K}(t)$ reaches the maximal value of 1 would imply an upper bound on $exp_{NZ}(n)$. Supported by numerical experiments, we have stated a conjecture that, if true, would lead to a quadratic upper bound on $exp_{NZ}(n)$ and to a quadratic upper bound on the reset threshold of any synchronizing DFA on n states associated to some primitive NZ-set. Finally, we have introduced the k-rendezvous time for primitive sets by generalizing concepts already used in automata theory: we have seen that it is linear in n for $k \leq \sqrt{n}$, a result that has not yet being proved for synchronizing DFAs. We have also showed that improvements on the estimate on the k-rendezvous time would lead to improvements on the estimate of the stagnation length of $\bar{K}(t)$ and hence possibly to a better upper bound on $exp_{NZ}(n)$.

Nonnegative matrix semigroups are rather simple objects, and yet we have seen that they find applications in many different fields. In this manuscript we have borrowed techniques from various disciplines (random graph theory, game theory, automata theory, optimization...) for proving results on primitivity and related concepts (column-primitivity property, synchronization, directability...). We believe that our probabilistic approaches to primitivity and the bridges that we have established between these different techniques could be combined and enhanced to bring a better insight on these phenomena and possibly to shed some light on questions that are open for the moment in the various fields, as the Černý conjecture.

We found particularly interesting the connection between random graph theory and the randomized generation of sets of binary matrices, and we believe that many results already proved in random graph theory could be used for, or extended to, random labeled directed multigraphs in order to study the asymptotic behavior of properties of matrix semigroups. For example, it might be interesting (and probably not too difficult) to extend Propositions A.2 and A.3 (see Appendix A) to the models $\mathcal{B}_m(n,p)$ and $\mathcal{B}_m(n,M)$, thus showing that they present the same asymptotic behavior when M is near n^2p. Random labeled directed graphs could be used for example for traffic models, where the labels could represent the amount of congestion in each road.

Testing primitivity for $\mathcal{B}_m(n,p)$ can also be read in terms of *reliable net-*

works. Reliable networks were born as a simple representation of communication networks where each communication line has a certain probability to fail; usually, we are interested in the probability that it is still possible to send a message from a center to any other center despite the failures occurring. The *binomial* model $G(n,p)$ in random graph theory, that is a random graph on n vertices where each edge appears with probability p, can be used to model reliable networks. Indeed, $G(n,p)$ has the same distribution of a complete graph on n vertices where each edge is independently destroyed with probability $1-p$. Guaranteeing the communication between any pair of vertices translates in this case into checking the connectedness property of $G(n,p)$. Suppose now to start from a labeled directed multigraph on n vertices and m labels such that for every $i,j \in [n]$ and $k \in [m]$ there exists an edge going from i to j labeled by k, and suppose that each of these edges are independently destroyed with probability $1-p$. In this model the communication between two adjacent centrers can fail just in one direction or both. For security reasons, we are interested not only in the probability that it is still possible to send a message from any center to any other center despite the failures occurring but we want these communications to follow the *same* transmission protocol, in order to identify the messages as *secure*. More precisely, we are interested in the probability that exists a sequence of labels such that every center can communicate with any other center by following a transmission line that is labeled by that sequence. This is clearly equal to investigating the primitivity property of $\mathcal{B}_m(n,p)$.

Time-inhomogeneous Markov chains and consensus systems find applications in many natural and social settings, as for example in floacking [65], which is a particular case of coordination of a multi-agent system. Suppose to generate the matrix set from which the transition matrices of the time-inhomogeneous Markov chain are chosen (or the matrix set from which the switching matrices of the consensus system are chosen) according to the models $\bar{\mathcal{P}}_m(n)$, $\mathcal{B}_m(n,p)$ or $\mathcal{B}_m(n,M)$, where we consider p above the threshold $\hat{p} = (\log n + c)/n$ and M above the threshold $\hat{M} = n(\log n + c)$. We can change the magnitude of the positive entries of the models $\bar{\mathcal{P}}_m(n)$, $\mathcal{B}_m(n,p)$ and $\mathcal{B}_m(n,M)$ in order to make them row-stochastic without modifying the primitivity property. Our results of Chapter 4 implies that in this case with high probability the Markov chain admits a sequence of transition matrices that converges to a one-rank matrix and with high probability the consensus system admits a switching sequence that makes it converge to consensus.
In Subsection 3.1.1 we have seen that the scrambling index influences the rate of convergence of the consensus system to consensus: it would be interesting to develop a function similar to the SPF for approximating the scrambling index of a column-primitive set.

Notice that the threshold $\hat{p} = (\log n + c)/n$ of $\mathcal{B}_m(n,p)$ with respect to the property of being primitive does *not* depend on the number of matrices m. It seems natural to wonder whether choosing different probabilities for each matrix of the set $\mathcal{B}_m(n,p)$, that is considering $\mathcal{B}_m(n,\mathbf{p}) = \{B_1(n,p_1),\dots,B_m(n,p_m)\}$ for $\mathbf{p} = (p_1(n),\dots,p_m(n)) \in [0,1]^m$ and $n \in \mathbb{N}$, would result in a random model having a threshold with respect to the primitivity property that depends on m. We can conjecture that this threshold exists and depends on $\sum_{i=1}^m p_i$; if this was true, we could choose the sequences

$p_1(n), \ldots, p_m(n)$ such that $\mathcal{B}_m(n, \mathbf{p})$ is primitive with high probability, while for every $i \in [m]$, $\mathcal{B}_{m-1}(n, \mathbf{p} \setminus p_i)$ is not primitive with high probability, where $\mathbf{p} \setminus p_i$ is the vector $\mathbf{p}$ without its i-th entry. In this way with high probability we would sample a *proper* primitive set of m matrices, which might lead to new families of primitive sets with quadratic exponent and/or to new slowly synchronizing DFAs.

We also believe that our randomized algorithm generating primitive sets with quadratic exponents could be further improved in order to find new families of slowly synchronizing automata: for example, we could think about modifying the way a permutation matrix is extracted from a binary one in the procedure *Extractperm*, about selecting the permutations on the partitions according to a probability distribution different from the uniform one, or about considering other kind of sets rather than perturbed permutation sets. As mentioned in Subsection 4.4.1, one can also think to apply our construction directly to DFAs by leveraging the recent result of Alpin and Alpina characterizing nonsynchronizing DFAs ([4], Theorem 3 and Section 2). Furthermore, Conjecture 4.25 on the exact magnitude of the reset threshold of our families of slowly synchronizing automata still has to be proven, even if our numerical results seem to confirm it.

Regarding the SPF, Conjecture 5.12 on the linear length of the stagnations of $\bar{K}$ is open. Results on the length of the stagnations of the SPF $K(t)$ and on the magnitude of its jumps would be also of interest as they could be used to possibly obtain a better upper bound on $exp_{NZ}(n)$ than the one presented in Corollary 5.19, by making use of Equation (5.18). Finally, we could obtain a better upper bound on $exp_{NZ}(n)$ also by improving the estimate of the k-RT for $k \geq \sqrt{n}$ (see Proposition 5.18). Besides, we would like to make the SPF more efficiently computable, at least for small t: for example we wonder whether we could avoid to compute its initial stagnation, as nothing interesting is happening there, thus immediately starting from the first time at which $K(t) > 1/n$. We are also interested in other methods for approximating the exponent of a primitive NZ-set, for example by adapting the Eppstein heuristic to make it directly finding a positive product in a primitive semigroup, instead of finding just a product with a positive column.

Bibliography

[1] R. L. Adler, L. W. Goodwyn, and B. Weiss. Equivalence of topological markov shifts. *Israel Journal of Mathematics*, 27(1):49–63, 1977.

[2] M. Akelbek and S. Kirkland. Primitive digraphs with the largest scrambling index. *Linear Algebra and its Applications*, 430(4):1099 – 1110, 2009.

[3] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley Publishing, 4th edition, 2016.

[4] Yu. A. Al'pin and V. S. Al'pina. Combinatorial properties of entire semigroups of nonnegative matrices. *Journal of Mathematical Sciences*, 207(5):674–685, 2015.

[5] D. S. Ananichev. The mortality threshold for partially monotonic automata. In *Developments in Language Theory*, pages 112–121, 2005.

[6] D. S. Ananichev and V. V. Gusev. Approximation of reset thresholds with greedy algorithms. *Fundamenta Informaticae*, 145(3):221–227, 2016.

[7] D. S. Ananichev, M. V. Volkov, and V. V. Gusev. Primitive digraphs with large exponents and slowly synchronizing automata. *Journal of Mathematical Sciences*, 192(3):263–278, 2013.

[8] G. E. Andrews, R. Askey, and R. Ranjan. *Special Functions*, volume 71 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 1999.

[9] U. Azfar, C. Catalano, L. Charlier, and R. M. Jungers. A linear bound on the k-rendezvous time for primitive sets of allowable matrices. *Developments in Language Theory*, 2019. Submitted.

[10] L. Babai, R. Beals, J.-Y. Cai, G. Ivanyos, and E. M. Luks. Multiplicative equations over commuting matrices. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 498–507, 1996.

[11] M.-P. Béal, M. V. Berlinkov, and D. Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata. *International Journal of Foundations of Computer Science*, 22:277–288, 2011.

[12] M. V. Berlinkov. Approximating the minimum length of synchronizing words is hard. *Theory of Computing Systems*, 54(2):211–223, 2014.

[13] M. V. Berlinkov. On the probability of being synchronizable. In *Algorithms and Discrete Applied Mathematics*, pages 73–84, 2016.

[14] M. V. Berlinkov and C. Nicaud. Synchronizing random almost-group automata. In *Implementation and Application of Automata*, pages 84–96, 2018.

[15] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Classics in Applied Mathematics. Society for Industrial Mathematics, 1987.

[16] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1st edition, 1997.

[17] V. D. Blondel, R. M. Jungers, and A. Olshevsky. On primitivity of sets of matrices. *Automatica*, 61:80–88, 2015.

[18] B. Bollobás. *Random Graphs*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2001.

[19] E. Boukas. *Stochastic switching systems: analysis and design*. Birkhauser, 2005.

[20] R. A. Brualdi. *Introductory Combinatorics*. Pearson Education, 2004.

[21] R. A. Brualdi and Ryser H. J. *Combinatorial matrix theory*. Cambridge University Press, 1991.

[22] H. V. Burkhard. Zum langenproblem homogener experimente an determinierten und nicht-deterministischen automaten. *Elektronische Informationsverarbeitung und Kybernetik*, 12:301 – 306, 1976.

[23] M.-P. Béal and D. Perrin. A quadratic algorithm for road coloring. *Discrete Applied Mathematics*, 169:15 – 29, 2014.

[24] C. Catalano and R. M. Jungers. On random primitive sets, directable ndfas and the generation of slowly synchronizing dfas. *Journal of Automata, Languages, and Combinatorics*, 2018. Accepted.

[25] C. Catalano and R. M. Jungers. On randomized generation of slowly synchronizing automata. In *Mathematical Foundations of Computer Science*, pages 48:1–48:21, 2018.

[26] C. Catalano and R. M. Jungers. The synchronizing probability function for primitive sets of matrices. In *Developments in Language Theory*, pages 194–205, 2018.

[27] C. Catalano and R. M. Jungers. The synchronizing probability function for primitive sets of matrices. *Developments in Language Theory Special Issue*, 2018. Submitted.

[28] Y.-B. Chen and D. J. Ierardi. The complexity of oblivious plans for orienting and distinguishing polygonal parts. *Algorithmica*, 14(5):367–397, 1995.

[29] P.-Y. Chevalier, V. V. Gusev, R. M. Jungers, and J. M. Hendrickx. Sets of stochastic matrices with converging products: Bounds and complexity. *CoRR*, abs/1712.02614, 2017. Preprint.

[30] P.-Y. Chevalier, J. M. Hendrickx, and R. M. Jungers. Reachability of consensus and synchronizing automata. In *IEEE Conference on Decision and Control*, pages 4139–4144, 2015.

[31] J. E. Cohen and P.-H. Sellers. Sets of nonnegative matrices with positive inhomogeneous products. *Linear Algebra and its Applications*, 47:185 – 192, 1982.

[32] O. L. V. Costa, M. D. Fragoso, and R. P. Marques. *Discrete-Time Markov Jump Linear Systems*. Probability and Its Applications. Springer-Verlag London, 1 edition, 2005.

[33] M. de Bondt, H. Don, and H. Zantema. DFAs and PFAs with long shortest synchronizing word length. In *Developments in Language Theory*, pages 122–133, 2017.

[34] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[35] J. D. Dixon. Asymptotics of generating the symmetric and alternating group. *The electronic journal of combinatorics*, 12, 2005.

[36] A. L. Dulmage and N. S. Mendelsohn. Gaps in the exponent set of primitive matrices. *Illinois Journal of Mathematics*, 8(4):642–656, 12 1964.

[37] M. Dzyga, R. Ferens, V. V. Gusev, and M. Szykuła. Attainable values of reset thresholds. In *Mathematical Foundations of Computer Science*, volume 83, pages 40:1–40:14, 2017.

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

[39] A. Erdős and A. Rényi. On random matrices. *Publ. Math. Inst. Hungar. Acad. Sci.*, 8:455–461, 1964.

[40] P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[41] V. M. Fomichev, Ya. E. Avezova, A. M. Koreneva, and S. N. Kyazhin. Primitivity and local primitivity of digraphs and nonnegative matrices. *Journal of Applied and Industrial Mathematics*, 12(3):453–469, 2018.

[42] E. Fornasini and M. E. Valcher. Directed graphs, 2d state models, and characteristic polynomials of irreducible matrix pairs. *Linear Algebra and its Applications*, 263:275 – 310, 1997.

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

[44] J. Friedman, A. Joux, Y. Roichman, J. Stern, and J.-P. Tillich. The action of a few random permutations on r-tuples and an application to cryptography. In *Symposium on Theoretical Aspects of Computer Science*, pages 375–386, 1996.

[45] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[46] Z. Gazdag, S. Iván, and J. Nagy-György. Improved upper bounds on synchronizing nondeterministic automata. *Information Processing Letters*, 109(17):986 – 990, 2009.

[47] M. Gerbush and B. Heeringa. Approximating minimum reset sequences. In *Implementation and Application of Automata*, pages 154–162, 2011.

[48] B. Gerencsér, V. V. Gusev, and R. M. Jungers. Primitive sets of nonnegative matrices and synchronizing automata. *Siam Journal on Matrix Analysis and Applications*, 39(1):83–98, 2018.

[49] F. Gonze, B. Gerencsér, and R. M. Jungers. Synchronization approached through the lenses of primitivity. In *35th Benelux Meeting on Systems and Control*, page 96, 2016.

[50] F. Gonze, V. V. Gusev, B. Gerencsér, R. M. Jungers, and M. V. Volkov. On the interplay between Babai and Černý's conjectures. In *Developments in Language Theory*, pages 185–197, 2017.

[51] F. Gonze and R. M. Jungers. On the synchronizing probability function and the triple rendezvous time. In *Language and Automata Theory and Appl.*, pages 212–223, 2015.

[52] F. Gonze, R. M. Jungers, and A. N. Trahtman. A note on a recent attempt to improve the pin-frankl bound. *Discrete Mathematics & Theoretical Computer Science*, 17(1), 2015.

[53] A. J. Graham and D. A. Pike. A note on thresholds and connectivity in random directed graphs. *Atlantic Electronic Journal of Mathematics*, 3(1):1–5, 2008.

[54] V. V. Gusev, R. M. Jungers, and E. V. Pribavkina. Generalized primitivity of labeled digraphs. *Electronic Notes in Discrete Mathematics*, 61:549 – 555, 2017.

[55] V. V. Gusev and E. V. Pribavkina. Reset thresholds of automata with two cycle lengths. In *International Conference on Implementation and Application of Automata*, pages 200–210, 2014.

[56] J. Hajnal. On products of non-negative matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 79(3):521–530, 1976.

[57] D. J. Hartfiel. *Nonhomogeneous matrix products*. World Scientific Publishing, 2002.

[58] H. Hennion. Limit theorems for products of positive random matrices. *The Annals of Probability*, 25(4):1545–1587, 10 1997.

[59] J. E. Hopcroft and R. M. Karp. A $n^{5/2}$ algorithm for maximum matchings in bipartite. In *Switching and Automata Theory*, pages 122–125, 1971.

[60] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.

[61] T. P. Igor, A. Roman, M. Szykuła, and B. Zielinski. A machine learning approach to synchronization of automata. *Expert Syst. Appl.*, 97:357–371, 2018.

[62] B. Imreh and M. Steinby. Directable nondeterministic automata. *Acta Cybernetica*, 14(1):105–115, February 1999.

[63] M. Ito. *Algebraic Theory of Automata and Languages*. World Scientific, 2004.

[64] S. Iván. Synchronizing weighted automata. In *Automata and Formal Languages*, pages 301–313, 2014.

[65] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[66] S. Janson, T. Luczak, and A. Rucinski. *Random Graphs*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[67] K. Jarkko. A counter example to a conjecture concerning synchronizing words in finite automata. *Bulletin of the EATCS*, 73:146, 2001.

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

[69] J.-Y. Kao, N. Rampersad, and J. Shallit. On nfas where all states are final, initial, or both. *Theoretical Computer Science*, 410(47):5010 – 5021, 2009.

[70] J. Kari. Synchronizing finite automata on eulerian digraphs. *Theoretical Computer Science*, 295(1):223 – 232, 2003.

[71] D. J. Kfoury. Synchronizing sequences for probabilistic automata. *Studies in Applied Mathematics*, 49(1):101–103, 1970.

[72] S. Kiefer and M. Corto. On finite monoids over nonnegative integer matrices and short killing words. *CoRR*, abs/1808.00940, 2018. Submitted.

[73] A. Kisielewicz and M. Szykuła. Synchronizing automata with extremal properties. In *Mathematical Foundations of Computer Science*, pages 331–343, 2015.

[74] L. Lovász and M. D. Plummer. Bipartite graphs with perfect matchings. In *Matching Theory*, volume 121, pages 121 – 141. North-Holland, 1986.

[75] P. V. Martyugin. Lower bounds for length of carefully synchronizing words. In *Satellite Workshop on Words and Automata of the International Computer Science Symposium in Russia*, 2006.

[76] A. Mateescu and A. Salomaa. Many-valued truth functions, Černý's conjecture and road coloring. In *EATCS Bulletin*, page 134–150, 1999.

[77] R.B. Myerson. *Game theory*. Harvard University Press, 1997.

[78] B. K. Natarajan. An algorithmic approach to the automated design of parts orienters. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, SFCS '86, pages 132–142, 1986.

[79] C. Nicaud. Fast synchronization of random automata. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 60, pages 43:1–43:12, 2016.

[80] D. D. Olesky, B. Shader, and P. van den Driessche. Exponents of tuples of nonnegative matrices. *Linear Algebra and its Applications*, 356(1):123 – 134, 2002.

[81] J. Olschewski and M. Ummels. The complexity of finding reset words in finite automata. In *Mathematical Foundations of Computer Science*, pages 568–579, 2010.

[82] I. Palásti. On the strong connectedness of directed random graphs. *Studia Scientiarum Mathematicarum Hungarica 1*, 1:205–214, 1966.

[83] M. S. Paterson. Unsolvability in 3x3 matrices. *Studies in Applied Mathematics*, 49(1):105–107, 1996.

[84] A. Paz. Graph-theoretic and algebraic characterizations of some markov processes. *Israel Journal of Mathematics*, 1(3):169–180, 1963.

[85] A. Paz. *Introduction to Probabilistic Automata*. Computer Science and Applied Mathematics. Academic Press, Inc., 1971.

[86] J. K. Percus. *Combinatorial Methods*, volume 4 of *Applied Mathematical Sciences*. Springer-Verlag New York, 2 edition, 2000.

[87] J.-E. Pin. On two combinatorial problems arising from automata theory. In *International Colloquium on Graph Theory and Combinatorics*, volume 75, pages 535–548, 1983.

[88] I. Potapov and P. Semukhin. Decidability of the membership problem for 2x2 integer matrices. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 170–186, 2017.

[89] I. Potapov and P. Semukhin. Membership Problem in GL(2,Z) Extended by Singular Matrices. In *Mathematical Foundations of Computer Science*, volume 83, pages 44:1–44:13, 2017.

[90] V. Yu. Protasov. Invariant functions for random matrices. *Functional Analysis Its Applications*, 44(3):230–233, 2010.

[91] V. Yu. Protasov. Invariant functions for the lyapunov exponents of random matrices. *Sbornik: Mathematics*, 202(1):101, 2011.

[92] V. Yu. Protasov. Classification of k-primitive sets of matrices. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1174–1188, 2013.

[93] V. Yu. Protasov and R. M. Jungers. Lower and upper bounds for the largest lyapunov exponent of matrices. *Linear Algebra and its Applications*, 438(11):4448–4468, 2013.

[94] V. Yu. Protasov and A. S. Voynov. Sets of nonnegative matrices without positive products. *Linear Algebra and its Applications*, 437:749–765, 2012.

[95] A. Roman. A note on Černý conjecture for automata with 3-letter alphabet. *Journal of Automata, Languages and Combinatorics*, 13(2), 2008.

[96] A. Roman. The NP-completeness of the road coloring problem. *Inf. Process. Lett.*, 111:342–347, 2011.

[97] B. Rudner. Construction of minimum-redundance codes with an optimum synchronizing property. *IEEE Transactions on Information Theory*, 17(4):478–487, 1971.

[98] I. K. Rystsov. Reset words for commutative and solvable automata. *Theoretical Computer Science*, 172(1):273 – 279, 1997.

[99] I. K. Rystsov. Estimation of the length of reset words for automata with simple idempotents. *Cybernetics and Systems Analysis*, 36(3):339–344, 2000.

[100] W. Samotij. A note on the complexity of the problem of finding shortest synchronizing words. In *AUTOMATA*, 2007.

[101] S. Sandberg. *Homing and Synchronizing Sequences*, pages 5–33. Springer Berlin Heidelberg, 2005.

[102] H. Schneider. Wielandt's proof of the exponent inequality for primitive nonnegative matrices. *Linear Algebra and its Applications*, 353(1):5 – 10, 2002.

[103] M. P. Schützenberger. On the synchronizing properties of certain prefix codes. *Information and Control*, 7(1):23–36, 1964.

[104] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer Series in Statistics. Springer-Verlag New York, 2 edition, 1981.

[105] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[106] B. Steinberg. The averaging trick and the černý conjecture. In *Developments in Language Theory*, pages 423–431, 2010.

[107] M. Steinby. Directable fuzzy and nondeterministic automata. *CoRR*, abs/1709.07719, 2017.

[108] M. Szykuła. Improving the upper bound the length of the shortest reset words. In *Symposium on Theoretical Aspects of Computer Science*, volume 96, pages 56:1–56:16, 2018.

[109] M. Szykuła and V. Vorel. An extremal series of eulerian synchronizing automata. In *Developments in Language Theory*, pages 380–392, 2016.

[110] A. N. Trahtman. *An efficient algorithm finds noticeable trends and examples concerning the Černy conjecture*, pages 789–800. Springer Berlin Heidelberg, 2006.

[111] A. N. Trahtman. The road coloring problem. *Israel Journal of Mathematics*, 172(1):51–60, 2009.

[112] J. N. Tsitsiklis and V. D. Blondel. The lyapunov exponent and joint spectral radius of pairs of matrices are hard—when not impossible—to compute and to approximate. *Mathematics of Control, Signals and Systems*, 10(1):31–40, 1997.

[113] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979.

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

[115] J. Černý, A. Piricka, and B. Rosenaueriva. On directable automata. *Kybernetika*, page 289–298, 1971.

[116] M. V. Volkov. Synchronizing automata preserving a chain of partial orders. In *International Conference on Implementation and Application of Automata*, pages 27–37, 2007.

[117] M. V. Volkov. Synchronizing automata and the Černý conjecture. In *Language and Automata Theory and Applications*, pages 11–27, 2008.

[118] V. Vorel. Subset synchronization of transitive automata. In *AFL*, 2014.

[119] D. B. West. *Introduction to Graph Theory*. Pearson, 2 edition, 2000.

Appendices

Appendix A

Properties of random graphs

The theory of random graphs was originated and developed by the two Hungarian mathematicians Erdős and Rényi in a series of papers appeared between 1959 and 1968. Since then, it has attracted the interest of many great scientists and has become a rich and fast-growing field. In this appendix we survey the results of random graph theory mentioned in Chapter 4 and we describe their connection with properties of random matrices. Section A.1 introduces the most known models in random graph theory by providing rigorous definitions and preliminary results; Section A.2 focuses on the property of a random bipartite graph to have a perfect matching and links it with the property of a random matrix to dominate a permutation matrix. The contents of this appendix are mainly based on the monographs of Bollobás [18] and of Janson et. al. [66].

A.1 Models of random graphs

In this paragraph we introduce two of the most used models for random graphs together with the notion of (sharp) threshold, and we report some classical results: our intention here is not to be fully exhaustive, as for achieving this we would need much more than an appendix, but rather to provide a background to the models of random sets of binary matrices presented in Chapter 4.

Informally speaking, a random graph is a graph constructed by a randomized procedure. The most famous models for random graphs are the following:

- the *uniform* random graph $G(n, M)$ denotes a graph selected from the set of graphs with vertex set $[n]$ and with exactly M edges according to the uniform distribution. Therefore, given $\bar{G}$ a graph on n vertices and M edges,
$$\mathbb{P}(G(n, M) = \bar{G}) = \binom{\binom{n}{2}}{M}^{-1}.$$
Clearly, $0 \leq M \leq N = \binom{n}{2}$; M usually depends on the cardinality of the vertex set n.

- the *binomial* random graph $G(n, p)$ denotes a graph with vertex set $[n]$ where each edge (i, j) appears independently with probability p (and does not appear with probability $1 - p$). Clearly, $0 \leq p \leq 1$; p usually depends on the cardinality of the vertex set n.

The binomial random graph can be seen as a particular case of *reliable networks*. Reliable networks were born as a simple representation of communication networks where each communication line has a certain probability to fail; clearly, if we start from the complete graph K_n on n vertices and we independently destroy each edge of K_n with probability $1-p$, we get the same distribution of the random model $G(n,p)$. In reliable networks, one is often interested in the probability that it is still possible to send a message from a center to any other center despite the failures occurring; this translates into checking the connectedness property of the underlying random graph.

We also consider the *bipartite* version of the aforementioned random models, as well as their *directed* version:

- the *uniform bipartite* random graph $G(n,n,M)$ denotes a graph selected according to the uniform distribution from the set of bipartite graphs with bipartition $V_1 \cup V_2$, $V_1 = [n] = V_2$, and with exactly M edges. Therefore, given $\bar{G}$ a bipartite graph with bipartitions $V_1 \cup V_2$, $V_1 = [n] = V_2$, and M edges,

$$\mathbb{P}(G(n,n,M) = \bar{G}) = \binom{n^2}{M}^{-1}.$$

  Clearly, $0 \leq M \leq N = n^2$; M usually depends on n.

- the *binomial bipartite* random graph $G(n,n,p)$ denotes a bipartite graph with bipartition $V_1 \cup V_2$, $V_1 = [n] = V_2$ where each edge (i,j), $i \in V_1$, $j \in V_2$, appears independently with probability p (and does not appear with probability $1-p$). Clearly, $0 \leq p \leq 1$; p usually depends on n.

- the *uniform directed* random graph $D(n,M)$ denotes a directed graph selected according to the uniform distribution from the set of directed graphs on n vertices and with exactly M edges. Therefore, given $\bar{D}$ a directed graph on n vertices and with M edges,

$$\mathbb{P}(D(n,M) = \bar{D}) = \binom{n^2}{M}^{-1}.$$

  Clearly, $0 \leq M \leq N = n^2$; M usually depends on n.

- the *binomial directed* random graph $D(n,p)$ denotes a directed graph on n vertices where each edge $i \to j$ appears independently with probability p (and does not appear with probability $1-p$). Clearly, $0 \leq p \leq 1$; p usually depends on n.

Remark 16. Let A be a binary $n \times n$ matrix and let $G_A = (V_1 \cup V_2, E)$ be the bipartite graph such that $V_1 = [n] = V_2$ and $(i,j) \in E$ if and only if $i \in V_1$, $j \in V_2$ and $A[i,j] = 1$. This establishes a one-to-one correspondence between the set of binary $n \times n$ matrices and the set of bipartite graphs with bipartitions $V_1 \cup V_2$, $V_1 = [n] = V_2$. Similarly, given a binary $n \times n$ matrix A, let $D_A = (V,E)$ be the directed graph on n vertices such that $i \to j \in E$ if and only if $A[i,j] = 1$. This establishes a one-to-one correspondence between the set of binary $n \times n$ matrices and the set of directed graphs on n vertices. In view of this, the random binary matrix $B(n,p)$ (see Chapter 4) is comparable

with the graph models $G(n,n,p)$ and $D(n,p)$: in particular, any property of $G(n,n,p)$ or $D(n,p)$ can be translated into a property of $B(n,p)$ and vice versa. The same holds for $B(n,M)$ with respect to $G(n,n,M)$ and $D(n,M)$.

All the random graph models introduced before depend on a parameter, namely M or p, which is usually supposed to be a function of the size of the graph n: typically, the results on random graphs are of asymptotic nature, where we wonder what is the probability that a certain property holds when n goes to infinity. To better formalize this, we first need to introduce some concepts.

Definition 42. Given a graph property $\mathcal{Q}$, we denote with $G \in \mathcal{Q}$ the fact that the graph G has the property $\mathcal{Q}$.
A graph property $\mathcal{Q}$ is said to be *increasing* if for any graph G_1, G_2 such that G_1 is a subgraph of G_2, if $G_1 \in \mathcal{Q}$ then $G_2 \in \mathcal{Q}$.

For example, the property of being connected is an increasing property for undirected graphs, the property of being strongly connected is an increasing property for directed graphs and the property of having a perfect matching is an increasing property for bipartite graphs; this last property will be explained in detail in the next section.

Lemma A.1 ([18], Theorem 2.1)**.** *Let $\mathcal{Q}$ be an increasing property, $M_1 \leq M_2$ and $p_1 \leq p_2$. Then it holds that:*

$$\mathbb{P}(G(n,M_1)) \leq \mathbb{P}(G(n,M_2)), \qquad \mathbb{P}(G(n,p_1)) \leq \mathbb{P}(G(n,p_2)),$$
$$\mathbb{P}(G(n,n,M_1)) \leq \mathbb{P}(G(n,n,M_2)), \qquad \mathbb{P}(G(n,n,p_1)) \leq \mathbb{P}(G(n,n,p_2)),$$

where $\mathbb{P}$, case by case, denotes the probability distribution of the corresponding random graph model.

The following results show that in case of increasing properties, the random models $G(n,M)$ and $G(n,p)$ present the same asymptotic behavior when M is near Np, and the same holds for $G(n,n,M)$ and $G(n,n,p)$.

Proposition A.2 ([66], Proposition 1.12, and [53], Theorem 4)**.** *Let $\mathcal{Q}$ be an arbitrary graph property, $p = p(n) \in [0,1]$ for $n \in \mathbb{N}$, $0 \leq a \leq 1$. Then:*

1. *if for every sequence $M = M(n)$ such that $M = Np + O(\sqrt{Np(1-p)}$ with $N = \binom{n}{2}$, it holds that $\mathbb{P}(G(n,M)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$, then it also holds that $\mathbb{P}(G(n,p)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$.*

2. *if for every sequence $M = M(n)$ such that $M = Np + O(\sqrt{Np(1-p)}$ with $N = n^2$, it holds that $\mathbb{P}(G(n,n,M)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$, then it also holds that $\mathbb{P}(G(n,n,p)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$.*

3. *if for every sequence $M = M(n)$ such that $M = Np + O(\sqrt{Np(1-p)}$ with $N = n^2$, it holds that $\mathbb{P}(D(n,M)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$, then it also holds that $\mathbb{P}(D(n,p)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$.*

Proposition A.3 ([66], Proposition 1.13)**.** *Let $\mathcal{Q}$ be an increasing graph property, $M = M(n) \in [0,N]$ for $n \in \mathbb{N}$, $0 \leq a \leq 1$. Then:*

1. *if for any sequence $p = p(n)$ such that $p = M/N + O(\sqrt{M(N-M)/N^3})$ with $N = \binom{n}{2}$ it holds that $\mathbb{P}(G(n,p)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$, then it also holds that $\mathbb{P}(G(n,M)) \in \mathcal{Q}) \longrightarrow a$ as $n \to \infty$.*

2. *if for any sequence* $p = p(n)$ *such that* $p = M/N + O(\sqrt{M(N-M)/N^3})$ *with* $N = n^2$ *it holds that* $\mathbb{P}(G(n,n,p)) \in \mathcal{Q}) \longrightarrow a$ *as* $n \to \infty$*, then it also holds that* $\mathbb{P}(G(n,n,M)) \in \mathcal{Q}) \longrightarrow a$ *as* $n \to \infty$*.*

Notice that Proposition A.2 holds for every graph property $\mathcal{Q}$, while in Proposition A.3 the property $\mathcal{Q}$ has to be increasing. These results show that in a great number of cases, $G(n,M)$ and $G(n,p)$ are practically interchangeable provided that M is close to Np, and this also holds for the models $G(n,n,M)$ and $G(n,n,p)$,

An interesting thing happening in the theory of random graphs is the phenomenon of *thresholds*, that we present here below. Notice that the definition of threshold for the random binary sets $\mathcal{B}_m(n,p)$ and $\mathcal{B}_m(n,M)$ given in Chapter 4 comes naturally as extension of these notions.

Definition 43. Let $\mathcal{Q}$ be a graph property. A sequence $\hat{p} = \hat{p}(n)$ is called a *threshold* for the random graph model $G(n,p)$ with respect to the property $\mathcal{Q}$ if for any sequence $p(n) \in [0,1]$, it holds that:

$$\lim_{n\to\infty} \mathbb{P}\Big(G(n,p(n)) \in \mathcal{Q}\Big) = \begin{cases} 0 & \text{if } p \ll \hat{p} \\ 1 & \text{if } p \gg \hat{p} \end{cases}.$$

A sequence $\hat{M} = \hat{M}(n)$ is called a *threshold* for the random graph model $G(n,M)$ if for any sequence $M(n) \in [0,N]$ with $N = \binom{n}{2}$, it holds that :

$$\lim_{n\to\infty} \mathbb{P}\Big(G(n,M(n)) \in \mathcal{Q}\Big) = \begin{cases} 0 & \text{if } M \ll \hat{M} \\ 1 & \text{if } M \gg \hat{M} \end{cases}.$$

The same definition of threshold holds for $G(n,n,p)$ and $G(n,n,M)$, and for $D(n,p)$ and $D(n,M)$.

A threshold is therefore a critical time that determines a phase transition between a moment at which a property is very unlikely to hold to a moment at which this property is very likely to hold. As already pointed out in Chapter 4, despite we refer to $\hat{p}$ or $\hat{M}$ as *the* thresholds, it is clear that they are defined up to constant factors: indeed, if $\hat{p}$ is a threshold, so is any sequence $\hat{p}'$ such that $\hat{p}' = \Theta(\hat{p})$, and the same holds for $\hat{M}$. A very interesting result is the following:

Theorem A.4 ([66], Theorem 1.24)**.** *Every increasing graph property has a threshold.*

A threshold can be *sharp*, which means (loosely speaking) that we have a phase transition any time we pass from $\lim_{n\to\infty} p/\hat{p} < 1$ to $\lim_{n\to\infty} p/\hat{p} > 1$. Not every monotone property has a sharp threshold.

Definition 44. Let $\mathcal{Q}$ be an increasing graph property. A sequence $\hat{p} = \hat{p}(n)$ is called a *sharp threshold* for the random graph model $G(n,p)$ with respect to the property $\mathcal{Q}$ if for any sequence $p(n) \in [0,1]$ it holds that:

$$\lim_{n\to\infty} \mathbb{P}(G(n,p) \in \mathcal{Q}) = \begin{cases} 0 & \text{if } \exists \bar{n}, \alpha > 0 : \forall n > \bar{n},\ p(n) \leq (1-\alpha)\hat{p}(n) \\ 1 & \text{if } \exists \bar{n}, \alpha > 0 : \forall n > \bar{n},\ p(n) \geq (1+\alpha)\hat{p}(n) \end{cases}.$$

A sequence $\hat{M} = \hat{M}(n)$ is called a *sharp threshold* for the random graph model $G(n, M)$ if for any sequence $M(n) \in [0, N]$ with $N = \binom{n}{2}$ it holds that:

$$\lim_{n\to\infty} \mathbb{P}(G(n, M) \in \mathcal{Q}) = \begin{cases} 0 & \text{if } \exists \bar{n}, \alpha > 0 : \forall n > \bar{n},\ M(n) \leq (1-\alpha)\hat{M}(n) \\ 1 & \text{if } \exists \bar{n}, \alpha > 0 : \forall n > \bar{n},\ M(n) \geq (1+\alpha)\hat{M}(n) \end{cases}.$$

The same definition of sharp threshold holds for $G(n, n, p)$ and $G(n, n, M)$. A threshold that is not sharp is called *coarse*.

A sharp threshold is never uniquely defined: if $\hat{p}$ is a sharp threshold, so is any sequence $\hat{p}'$ such that $\lim_{n\to\infty} \hat{p}'/\hat{p} = 1$, and the same holds for $\hat{M}$. The next theorem shows that our random graph models have a sharp threshold with respect to the (strongly) connectedness property.

Theorem A.5 ([40] Theorem 1, [53] Corollary 1 and [82])**.** *For any $c \in \mathbb{R}$:*

1. *$\hat{M} = n(\log n + c)/2$ is a sharp threshold for $G(n, M)$ with respect to the property of being connected;*
2. *$\hat{M} = n(\log n + c)$ is a sharp threshold for $D(n, M)$ with respect to the property of being strongly connected;*
3. *$\hat{p} = (\log n + c)/n$ is a sharp threshold for $G(n, p)$ and $D(n, p)$ with respect to the property of being connected and strongly connected, respectively.*

There are also examples of coarse thresholds: for example $G(n, p)$ has a coarse threshold with respect to the property of containing a given graph as a subgraph ([66], Chapter 4). We will see in the next section that $G(n, n, p)$ and $G(n, n, M)$ have a sharp threshold with respect to the property of having a perfect matching.

Remark 17. When in Chapter 4 we introduce the notion of (sharp) threshold for the random matrix model $\mathcal{B}_m(n, p)$, we are extending the concepts of thresholds to random *labeled* directed multigraphs: indeed, $\mathcal{B}_m(n, p)$ is a random model for a directed multigraph on n vertices and m labels, where for any $i, j \in [m]$ and $k \in [m]$, the edge $i \xrightarrow{k} j$ appears with probability p. Notice that for the strongly connectedness property $\mathcal{SC}$ of $\mathcal{B}_m(n, p)$ (or equivalently, the irreducibility property of $\mathcal{B}_m(n, p)$) its labels do not play a role, so it holds that

$$\mathbb{P}(\mathcal{B}_m(n, p) \in \mathcal{SC}) = \mathbb{P}(D(n, 1 - (1-p)^m) \in \mathcal{SC}).$$

Things become more interesting for the primitivity or column-primitivity property of $\mathcal{B}_m(n, p)$, as in this case the labels of $\mathcal{B}_m(n, p)$ do play a role. It is somehow surprising to see that $\mathcal{B}_m(n, p)$ experiences the same sharp threshold of $\hat{p} = (\log n + c)/n$ with respect to the primitivity and column-primitivity property as $D(n, p)$ for strong connectivity, as well as $\mathcal{B}_m(n, M)$ experiences the same threshold of $\hat{M} = n(\log n + c)$ with respect to primitivity as $D(n, M)$ for strong connectivity.

As already anticipated in the conclusive section, we believe that it would be interesting to prove analogous results to Proposition A.2 and A.3 for the models $\mathcal{B}_m(n, p)$ and $\mathcal{B}_m(n, M)$ as, to the best of our knowledge, there still does not exist much literature on random labeled directed (multi)graphs.

A.2 Perfect matchings

The contents presented in this section are mostly based on the book of West [119]. We introduce the notion of *perfect matchings* and we report the main results on them; we then show the equivalence between the problem of finding a perfect matching in a bipartite graph and the problem of finding a permutation matrix dominated by a given binary one. Finally, we survey the results on the random graph models introduced in the previous section with respect to the property of admitting a perfect matching.

Definition 45. Let $G = (V, E)$ be a graph with vertex set V and edge set E. A *matching* for G is a subset $T \subseteq E$ such that at most one edge in T is incident to each vertex of V. A matching is *perfect* if exactly one edge in T is incident to each vertex of V.

We are interested in perfect matchings for bipartite graphs. Given a bipartite graph $G = (V, E)$ with bipartition $V = V_1 \cup V_2$ and a vertex $v \in V$, we denote with $\mathcal{N}(v)$ the set of vertices that are adjacent to v in G. Given $A \subset V_1$ or $A \subset V_2$, we set $\mathcal{N}(A) = \bigcup_{a \in A} \mathcal{N}(a)$. The following theorem, proved by Hall in 1935 and after that called the *Hall's theorem*, provides a necessary and sufficient condition for a bipartite graph to admit a perfect matching.

Theorem A.6 ([119], Theorem 3.1.11)**.** *Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. Then G has a perfect matching if and only if for all $A \subseteq V_1$, $|\mathcal{N}(A)| \geq |A|$.*

The problem of determining whether a bipartite graph $G = (V, E)$, $V = V_1 \cup V_2$, admits a perfect matching is decidable in polynomial time: Hopcroft and Karp provided in [59] an algorithm that finds a maximum matching in time $O(E\sqrt{V})$, where a matching T is *maximum* if the set of vertices to which the edges of T are adjacent is the maximal possible. Clearly, if this set of vertices is equal to V then the matching is perfect, otherwise it means that the bipartite graph G does not admit a perfect matching.

We now show the equivalence between the problem of finding a perfect matching of a bipartite graph (if any) and the problem of selecting a permutation matrix dominated by a give binary one (if any).

Proposition A.7. *Let B be a binary matrix of size $n \times n$ and let $G_B = (V_1 \cup V_2, E)$ be the bipartite graph where $V_1 = [n] = V_2$ and $(v_1, v_2) \in E$ if and only if $v_1 \in V_1$, $v_2 \in V_2$ and $B[v_1, v_2] = 1$. Then B dominates a permutation matrix if and only if G_B admits a perfect matching.*

Proof. Suppose that B dominates a permutation matrix P and let $G_P = (V_1 \cup V_2, E_P)$ be the subgraph of G_B induced by P, i.e. $(v_1, v_2) \in E_P$ if and only if $v_1 \in V_1$, $v_2 \in V_2$ and $P[v_1, v_2] = 1$. Since P is a permutation matrix, it has exactly one 1 in every row and every column, so G_P must be a perfect matching. On the other hand, if T is a perfect matching for G_B, then the matrix P such that $P[i, j] = 1$ if $(i, j) \in T$, $P[i, j] = 0$ otherwise, is a permutation matrix dominated by B. □

Proposition A.7 establishes a one-to-one correspondence between the set of permutation matrices dominated by a binary matrix B and the set of perfect

matchings that G_B has. This implies that the number of permutation matrices dominated by a binary matrix B is equal to the number of different perfect matchings of G_B, which is in turn equal to the magnitude of the permanent[1] of B (see for example [86], part C, 1§). Valiant shows in [113] that, unless P=NP, there is no deterministic polynomial-time algorithm that computes the permanent of binary matrix; this means that there does not exist any polynomial-time algorithm that finds the number of permutation matrices that are dominated by a given binary one.

Consider now the random bipartite graphs $G(n, n, p)$ and $G(n, n, M)$ introduced in the previous section. The following theorems, originally proved by Erdős and Rényi in their seminal paper [39], establishes a *sharp threshold* for $G(n, n, p)$ and $G(n, n, M)$ with respect to the property of admitting a perfect matching.

Theorem A.8 ([66], Theorem 4.1)**.** *For any $c \in \mathbb{R}$, $\hat{p}(n) = (\log n + c)/n$ is a* sharp threshold *for $G(n, n, p)$ with respect to the property of admitting a perfect matching ($\mathcal{PM}$). Furthermore,*

$$\lim_{n\to\infty} \mathbb{P}\Big(G(n, n, \hat{p}(n)) \in \mathcal{PM}\Big) = e^{-2e^{-c}} .$$

Theorem A.9 ([18], Corollary 7.13)**.** *For any $c \in \mathbb{R}$, $\hat{M}(n) = n(\log n + c)$ is a* sharp threshold *for $G(n, n, M)$ with respect to the property of admitting a perfect matching ($\mathcal{PM}$). Furthermore,*

$$\lim_{n\to\infty} \mathbb{P}\Big(G(n, n, \hat{M}(n)) \in \mathcal{PM}\Big) = e^{-2e^{-c}} .$$

The reason why the strongly connectedness property and the perfect matching property share the same sharp threshold is that both these properties appear with high probability as long as the last isolated vertex disappears. Theorem A.8 also holds for $G(n, p)$ when n is even; in this case we have that $\lim_{n\to\infty} \mathbb{P}(G(n, \hat{p}) \in \mathcal{PM}) = e^{-e^{-c}}$ ([66], Corollary 4.5).

We can apply Theorem A.8 to the vertex-disjoint cycle covers of a directed graph.

Definition 46. A *vertex-disjoint cycle cover* for a directed graph D is a set of disjoint cycles which are subgraphs of D and contains all its vertices.

Lemma A.10. *A binary matrix B dominates a permutation matrix if and only if is associated directed graph $\mathcal{D}_B$ (see Definition 4) admits a vertex-disjoint cycle cover.*

Proof. Trivial. □

Corollary A.11. *For any $c \in \mathbb{R}$, $\hat{p}(n) = (\log n + c)/n$ is a* sharp threshold *for the random directed graph model $D(n, p)$ with respect to the property of admitting a vertex-disjoint cycle cover ($\mathcal{VDC}$). Furthermore,*

$$\lim_{n\to\infty} \mathbb{P}\Big(D(n, \hat{p}(n)) \in \mathcal{VDC}\Big) = e^{-2e^{-c}} .$$

Proof. Straightforward by Lemma A.10 and Theorem A.8. □

[1] We remind that the permanent of a matrix A of size $n \times n$ is equal to $perm(A) = \sum_{\sigma\in S_n} \prod_{i=1}^{n} A[i, \sigma(i)]$.

We end this section by recalling what is the *Dulmage-Mendelsohn decomposition* of a bipartite graph and its connection with perfect matchings. The definition below is based on [74].

Definition 47. Let $G = (V, E)$, $V = V_1 \cup V_2$, be a bipartite graph. Let F be the set of vertices $v \in V$ for which there is at least one maximum matching of G not covering v. Let $A = \mathcal{N}(F) \setminus F$ and $C = V \setminus (F \cup A)$. The *coarse Dulmage-Mendelsohn decomposition* of G is made of the subgraphs G_1, G_2 and G_3 of G where:

- G_1 is the subgraph of G induced by C;
- G_2 is the subgraph of G induced by $(F \cap V_1) \cup (A \cap V_2)$;
- G_3 is the subgraph of G induced by $(F \cap V_2) \cup (A \cap V_1)$.

If the bipartite graph $G = (V, E)$ admits a perfect matching, the subgraphs G_2 and G_3 in its coarse Dulmage-Mendelsohn decomposition are empty. The subgraph G_1 always admits a perfect matching, which can be found via the Hopcroft-Karp algorithm [59]. In matrix theory framework, the coarse Dulmage-Mendelsohn decomposition of a binary matrix B determines whether it dominates a permutation matrix and the Hopcroft-Karp algorithm finds one of such permutation matrices.

Appendix B

Complexity classes

We here provide the definitions of the complexity classes that are mentioned in this manuscript. Our intention is to give an idea of how hard is solving the problems presented in the previous chapters rather than provide a rigorous and detailed account of complexity theory; therefore, sometimes we will be less formal, in favor of a more intuitive description of the objects that we are going to present. The contents of this section are mainly based on [45] and we invite the reader to refer to this reference for any further needed detail.
In the following, we use the word *algorithm* to indicate a deterministic Turing machine.

We call a *problem* a general question to be answered, depending to some parameters that are left unspecified; an *instance* of a problem is obtained by specifying particular values of the parameters of the problem. We say that an algorithm *solves* a problem if it can be applied to any instance of the problem and it always produces the correct solution. A *decision* problem is a problem that can be posed as a yes-or-no question; an instance of a decision problem Π either belong to the set Y_Π of the «yes» answers, or to the set N_Π of the «no» answers. A decision problem is *decidable* if there exists an algorithm that solves it, otherwise it is said to be *undecidable*. The *halting problem* is an example of undecidable problem: there does not exist an algorithm that correctly determines whether an arbitrary program will eventually halt when run.

The complexity classes have been conceived to classify how difficult is to solve a given problem, i.e. how *fast* we can solve the problem with respect to the size of the inputs. The *time-complexity function* of an algorithm solving a problem Π expresses, for each input size n, the largest amount of time needed by the algorithm to solve the problem on instances of size n. The problems that are traditionally considered *tractable* are the ones having polynomial-time complexity, i.e. problems that can be solved by an algorithm whose time-complexity function is upper bounded by a polynomial function of the input size; the class P represents these tractable problems. Problems that do not admit a polynomial-time solving algorithm are usually considered *intractable*; however, most of the known intractable problems are decidable and can be solved in polynomial-time by the aid of a nondeterministic computer. These last problems belong to the so-called NP class.

A *nondeterministic* algorithm (or nondeterministic Turing machine) is an algorithm composed by a preliminary *guessing* stage and by a second *checking*

stage. Given an instance of a decision problem, the guessing stage merely guesses some structure S which may solve the problem, and then the checking stage checks if S indeed solves the problem. A bit more formally, a nondeterministic algorithm *solves* a decision problem Π if, for every instance I of Π:

1. if $I \in \mathrm{Y}_\Pi$ then there exists some structure S such that, when guessed at the guessing stage, will lead the checking stage to return «yes»;
2. if $I \in \mathrm{N}_\Pi$ then there does not exist any structure S such that, when guessed at the guessing stage, will lead the checking stage to return «yes».

A nondeterministic algorithm is said to operate in *polynomial time* if the checking stage operates in polynomial time. We are now ready to define the classes P and NP:

Definition 48. Given a problem Π, it is said to belong to:

1. P if there exists a polynomial-time algorithm that correctly solves the problem on every instance;
2. NP if there exists a polynomial-time nondeterministic algorithm that correctly solves the problem on every instance.

Loosely speaking, we can say that the problems in P are easily solvable, while problems in NP are easily checkable. It is not difficult to see that $\mathrm{P} \subseteq \mathrm{NP}$, as for every problem in P solved by a polynomial-time algorithm $\mathscr{A}$, we can use $\mathscr{A}$ as the checking stage of the nondeterministic algorithm, ignoring the initial guess. The problem whether $\mathrm{P} = \mathrm{NP}$, firstly formulated by Cook in 1971, is one of the big unsolved problems in mathematics and computer science and it is one of the remaining Millennium Prize Problems selected by the Clay Mathematics Institute. It is generally believed that $\mathrm{P} \subsetneq \mathrm{NP}$, as the algorithms in NP seem to be more powerful than the ones in P, but no proofs are so far available and the best we can say is that every algorithm in NP can be solved by a deterministic algorithm having exponential time complexity.

The *complement* Π^c of a decision problem Π is the problem originating by interchanging the «yes» and the «no» answer, i.e. the problem defined on the same instances of Π such that $\mathrm{Y}_{\Pi^c} = \mathrm{N}_\Pi$ and $\mathrm{N}_{\Pi^c} = \mathrm{Y}_\Pi$. It is clear that if Π belongs to P, then also Π^c belongs to P, because an algorithm in P will halt for all inputs. The class NP does not have the same symmetry between the «yes» and «no» answers and indeed it is still unclear whether Π^c belongs to NP any time Π belongs to NP .

Definition 49. A decision problem is in CO-NP if its complement is in NP.

It is generally believed that $\mathrm{NP} \neq \text{CO-NP}$; in view of what said before, it holds that $\mathrm{NP} \cap \text{CO-NP} \supseteq \mathrm{P}$.

We now want to characterize the "hardest" problems in NP. A *polynomial transformation* of a decision problem Π_1 to a decision problem Π_2 is a function $f : D_{\Pi_1} \to D_{\Pi_2}$, where D_{Π_i} is the set of instances of problem Π_i, such that f is computable in polynomial-time and for all istances I of Π_1, $I \in \mathrm{Y}_{\Pi_1}$ if

and only if $f(I) \in Y_{\Pi_2}$. Therefore, if f is a polynomial transformation of a decision problem Π_1 to a decision problem Π_2, by solving Π_2 we can solve Π_1 in the same time (up to a polynomial factor); we can thus say that problem Π_2 is *at least as hard to solve as* problem Π_1. This definition is the base concept of the NP-complete class:

Definition 50. A decision problem Π is NP-complete if it belongs to NP and, given any other problem Π' in NP, there exists a polynomial transformation of Π' to Π.

We can thus regard to NP-complete problems as the *hardest* problems in NP in the sense that, if we can solve one NP-complete problem in polynomial time, then all the problems in NP would be solvable in polynomial time. This would also result in P = NP. On the other hand, if P$\neq$ NP, then NP-complete $\cap$ P $= \emptyset$. It has been proved that if there exists an NP-complete problem that belongs also to CO-NP, then NP = CO-NP. Some famous examples of NP-complete problems are the satisfiability problem (SAT), the problem of deciding whether a graph has an Hamiltonial circuit, the problem of deciding whether a graph admits a vertex cover of size smaller than a given constant and the problem of deciding whether a graph admits a clique of size greater than a given constant. This means that any problem in NP can be polynomially reduced to each of these problems.

We define in a similar way the class of CO-NP-complete problems, which is the class of the *hardest* problems in CO-NP:

Definition 51. A problem Π is CO-NP-complete if it belongs to CO-NP and, given any other problem Π' in CO-NP, there exists a polynomial transformation of Π' to Π.

It can be proved that if P $\neq$ NP, then there exist problems that are not in P nor NP-complete; a problem of this kind is said to belong to NPI, as it has an *intermediate* complexity between the *tractable* problems of P and the hardest problems in NP. Examples of problems that are potential members of NPI are the graph isomorphism (determining whether two given graphs are isomorphic), linear programming problems and the problem of finding two integers $m, n > 1$ such that $K = mn$, for a given positive integer K.

Decision problems can be represented as *languages*. A *language* is a set of words $L \subset \Sigma^*$ over a finite alphabet Σ. The *encoding scheme* of a problem Π describes each instance of the problem in terms of a string of symbols over a finite alphabet: the *language associated* to a decision problem Π with encoding scheme e is the set of strings encoding the instances of the problem belonging to Y_Π. The complexity classes can then be equivalently described in terms of languages. Despite it may appear that the encoding scheme does play a role in establishing to which complexity class a given problem belongs, this is not really the case, as long as the encoding is *reasonable*: informally speaking, an encoding is reasonable if it is concise, i.e. if it encodes the instances without useless "extra padding", and if it is decodable, i.e. one has to be able to decode the meaning of any string encoding a component of any particular instance in polynomial time. A reasonable encoding scheme is for example the one used by our personal computer to memorize data. If a problem is encoded by a reasonable encoding scheme, then its complexity class does not depend

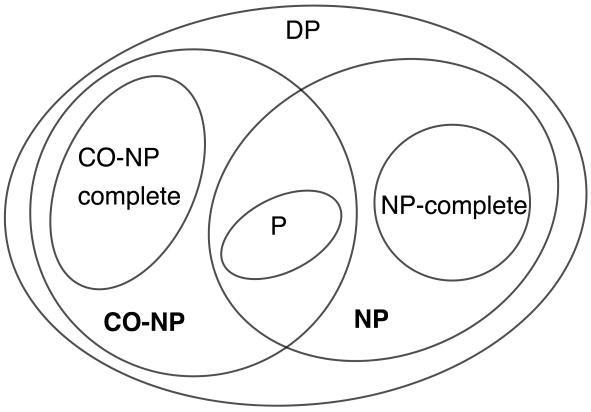


Figure B.1: Relation between the mentioned complexity classes of decision problems, assuming that P $\neq$ NP and NP $\neq$ CO-NP. It may or may not hold that P = NP $\cap$ CO-NP.

on the particular reasonable encoding scheme chosen. In this framework, a *polynomial transformation* of a language L_1 on the alphabet Σ_1 to a language L_2 on the alphabet Σ_2 is a function $f : \Sigma_1^* \to \Sigma_2^*$ such that f is computable in polynomial time and, for all $x \in \Sigma_1^*$, $x \in L_1$ if and only if $f(x) \in L_2$.
The next complexity class that we introduce belongs to the so-called *boolean hierarchy*, that is the complexity hierarchy of boolean combinations (intersection, union and complementation) of NP languages.

Definition 52. 1. A language L is in DP if there exists a language L_1 in NP and a language L_2 in CO-NP such that $L = L_1 \cap L_2$.

2. A language L is DP-complete if it belongs to DP and for every language L' in DP, there is a polynomial transformation of L' to L.

The DP class contains NP $\cup$ CO-NP; the DP-complete problems are the *hardest* problems in DP. See Figure B.1 to see how the complexity classes mentioned so far are related to each others.

We now introduce the class of NP-hard problems: loosely speaking, this is the class of all the problems that are at least as hard as any NP-complete problem. The difference is that this time we do not restrict ourselves to decision problems, nor we care if such problems are in NP. We instead consider the class of *search* problems: a search problem Π consists of a set D_Π of instances and of a finite set $S_\Pi[I]$ for all $I \in D_\Pi$. An algorithm is said to *solve a search problem* Π if for any input instance I, it returns «no» if $S_\Pi[I]$ is empty, while it returns a solution $s \in S_\Pi[I]$ otherwise. Clearly, a decision problem is a special case of search problem; search problems are also called *functional* problems.
Given two search problems Π and Π', a *polynomial-time Turing reduction* of Π to Π' is an algorithm $\mathscr{A}$ that solves Π by using an hypothetical subroutine S for solving Π' such that, if S were polynomial in time, then $\mathscr{A}$ would be polynomial in time. The notion of Turing reduction can be rigorously defined by the concept of *oracle Turing machine*: we do not report here its definition, for which we refer the reader to Chapter 5 of [45]. We just mention that an oracle Turing machine consists of a standard deterministic Turing machine

augmented with an *oracle tape*; the oracle tape is a «black box» that is able to solve certain decision problems in a single operation.

Definition 53. A search problem Π is said to be NP-hard if there exists an NP-complete problem Π' such that there is a polynomial-time Turing reduction of Π' to Π.

Therefore, a problem is NP-hard if it is at least as hard to solve as the hardest problem in NP; clearly, all the NP-complete problems are NP-hard. The complement of an NP-hard problem is NP-hard.

The classes P, NP and CO-NP can be easily extended to search problems and are denoted respectively by FP, FNP and CO-FNP: FP is then the class of search problems that can be solved in polynomial time by a deterministic Turing machine, FNP is the class of search problems that can be solved in polynomial time by a nondeterministic Turing machine and CO-FNP is the class of search problems whose complements belong to FNP. The last complexity classes that we are going to define belong to the so-called *polynomial hierarchy* for search problems, that is a generalization of the classes FP, FNP and CO-FNP to oracle machines.

Definition 54.

1. A search problem Π belongs to the class $\mathrm{FP}^{\mathrm{NP}}$ if there exists a search problem Π' in FNP such that Π is polynomial-time Turing reducible to Π'. In other words, a problem Π belongs to the class $\mathrm{FP}^{\mathrm{NP}}$ if it can be solved by a deterministic polynomial-time algorithm equipped with the ability to use an oracle for any FNP problem.

2. A search problem Π belongs to the class $\mathrm{FP}^{\mathrm{NP}[\log]}$ if it can be solved by a deterministic logarithmic-time algorithm equipped with the ability to use an oracle for any FNP problem.

Appendix C

Block-permutation structures of a matrix

We have seen that an irreducible NZ-set of matrices is not primitive if and only if each matrix of the set has a block-permutation structure on the same nontrivial partition (Theorem 3.7, Chapter 3); in this section we provide some little results regarding the block-permutation structures of a nonnegative matrix. We show that checking whether a nonnegative matrix has a block-permutation structure on a given partition is polynomial in time and that the problem of finding all the nontrivial block-permutation structures of a nonnegative matrix is at least exponential in time in the worst case; we then present a formula that counts the number of the block-permutation structures of a permutation matrix in terms of the length of its cycles in its cycle decomposition. We also present a formula that counts the block-permutation structures of a permutation matrix with blocks of the same size.

Proposition C.1. *Let M be a nonnegative NZ-matrix of size $n \times n$ and let $\Omega = \dot{\bigcup}_{i=1}^{r} \Omega_i$, $r \geq 2$, be a partition of $[n]$. The problem of checking if M has a block-permutation structure on Ω takes at most $O(n^3)$ operations, thus it is polynomial in time.*

Proof. To check whether M has a block-permutation structure on the partition Ω we can use the algorithm described in Algorithm C.1; it is a modification of the Protasov-Voynov algorithm for checking primitivity and we now describe in words how it operates. Observe that $r \leq n$ and $1 \leq |\Omega_i| \leq n - r + 1$ for all $i \in [r]$. Our goal is to check whether there exists a permutation $\sigma \in S_r$ such that for all $l \in [r]$ and $i \in \Omega_l$, if $M[i,k] > 0$ then $k \in \Omega_{\sigma(l)}$. We initialize a vector σ of length r as a zero-vector. At the end of the algorithm, if the matrix M turns out to have a block-permutation structure on the partition Ω, then σ will be a permutation compatible with M and Ω.

At each step $l \in [r]$, we set $K_l = \{k : \exists i \in \Omega_l \text{ s.t. } M[k,i] > 0\}$ and we check if there exists Ω_s such that $K_l \subseteq \Omega_s$: if not, we stop the algorithm and we conclude that M does not have a block-permutation structure on the partition Ω. In case of positive answer, we check if $\sigma(s) = 0$: if this is the case, we set $\sigma(s) = l$ and we proceed to the next step $l + 1$, if it is not, we stop the algorithm and we conclude that M does not have a block-permutation structure on the partition Ω. If the algorithm manages to finish the last iteration $l = r$, then M has a block-permutation structure on Ω with respect the permutation

σ, that is for all $l \in [r]$ and $i \in \Omega_l$, if $M[i,k] > 0$ then $k \in \Omega_{\sigma(l)}$.
By looking at Algorithm C.1 we can see that for each $l \in [r]$, line 5 of the procedure takes at most $n(n-r+1)$ operations, line 7 takes at most n operations and line 9 takes at most $(n-r+1)^2$ operations, so in total we need $O(r(n+n(n-r+1)+(n-r+1)^2) = O(n^3)$ operations. □

Algorithm C.1: Algorithm for checking if a NZ-matrix admits a block-permutation structure on a given partition.

```
1  Input: M, Omega=Omega_1,...,Omega_r.
2  sigma=zeros(1,r)
3  for l=1:r
4          c_1,..,c_t = columns of M indicized by Omega_l
5          S=indices of the positive entries of c_1,..,c_t
6          s1= first element of S.
7          find k such that s1 belongs to Omega_k
8          if sigma(k)=! 0: display "not feasible", RETURN, end
9          if S is not a subset of Omega_k:
10                 display "not feasible", RETURN,
11         end
12         sigma(k)=l;
13 end
```

Proposition C.2. *Finding all the block-permutation structures of a nonnegative matrix of size $n \times n$ is at least exponential in n in the worst case.*

Proof. Let I be the identity matrix of size $n \times n$; it admits a block-permutation structure on *any* nontrivial partition on $[n]$. The number of nontrivial partitions of $[n]$ is equal to $B_n - 2$, where B_n is the Bell number that is known to be at least exponential in n. □

We now exhibit closed formulas for computing the number of partitions on which a given permutation matrix has a block-permutation structure, also in the case we require all the blocks to have the same size.

We first need a bit of notation: we indicate with $a|b$ the fact that the natural number a is a divisor of the natural number b. Given a permutation matrix A of size $n \times n$, let $\sigma_A \in S_n$ be its associated permutation on n elements, i.e. $\sigma_A(i) = j$ if and only if $A[i,j] = 1$. Let $C_{\sigma_A} = \{C_1, \dots, C_m\}$ be the decomposition of σ_A in its disjoint cycles and let l_i be the length of C_i. We denote by $\mathscr{P}_{\sigma_A}$ the set of all the possible partitions of C_{σ_A}; given $\mathcal{P} \in \mathscr{P}_{\sigma_A}$, we indicate with $\{P_1, \dots, P_{r(\mathcal{P})}\}$ the elements of $\mathcal{P}$. For all $i = 1, \dots, r(\mathcal{P})$, we indicate with $\{C_{j_1^i}, \dots, C_{j_{|P_i|}^i}\}$ the cycles of P_i and we set $gcd(l_{P_i}) = gcd(\{l_{j_k^i} : k = 1, \dots, |P_i|\})$, the greatest common divisor of the length of the cycles in P_i. Finally, given a divisor $d_i | gcd(P_i)$, for every $k = 1, \dots, |P_i|$ we denote with $d'_{j_k^i}$ the positive integer such that $l_{j_k^i} = d_i d'_{j_k^i}$.

Proposition C.3. *Given a permutation matrix A of size $n \times n$, the number of partitions of $[n]$ on which A has a block-permutation structure is equal to*

$$\left(\sum_{\mathcal{P} \in \mathscr{P}_{\sigma_A}} \prod_{i=1}^{|\mathcal{P}|} \sum_{d | gcd(l_{P_i})} d^{|P_i|-1} \right) - 2 \quad . \tag{C.1}$$

Furthermore, the number of partitions of $[n]$ with blocks of the same size on which A has a block-permutation structure is equal to

$$-2+\sum_{\mathcal{P}\in\mathscr{P}_{\sigma_A}}\ \sum_{\substack{d_1\mid gcd(l_{P_1})\\ \vdots \\ d_r\mid gcd(l_{P_r}):\\ d'_{j_1^1}+\ldots+d'_{j^1_{|P_1|}}=d'_{j_1^2}+\ldots+d'_{j^2_{|P_2|}}=\ldots=d'_{j_1^r}+\ldots+d'_{j^r_{|P_r|}}}}\ \prod_{s=1}^{r} d_s^{|P_s|-1} \quad , \tag{C.2}$$

where to ease the notation we have set $r=r(\mathcal{P})$.

Proof. We first prove equation C.1; we prove it for $m=1,2$ (i.e. when σ_A has just one or two cycles in its cycle decomposition) and then we discuss how to generalize it to any m.

If $m=1$, then σ_A is a cycle on n elements. It is easy to see that in this case there exists a partition Ω on which A has a block-permutation structure if and only if there exists $d\neq 1,n$ s.t. $n=dd'$, such that $\Omega=\dot{\bigcup}_{j=1}^{d}\Omega_j$ where $\Omega_j=\{j,j+d',\ldots,j+d(d'-1)\}$ and $\sigma_A(\Omega_j)=\Omega_{j+1}$ for all $j=1,\ldots,d-1$ (and $\sigma_A(\Omega_d)=\Omega_1$). Therefore, the number of partitions of $[n]$ on which A has a block-permutation structure is equal to $|\{d|n : d\neq 1,n\}|=\sum_{d|n} d^0-2$, where this last term is exactly Equation (C.1) for $m=1$.

If $m=2$, $C_{\sigma_A}=\{C_1,C_2\}$. We can first consider the block-permutation structures of C_1 and C_2 individually, i.e. the block-permutation structures of $A[C_1,C_1]$ and $A[C_2,C_2]$; by joining the corresponding partitions, we get a partition of $[n]$ on which A has a block-permutation structure. These partitions are the ones covered by Equation (C.1) when $\mathcal{P}=\{\{C_1\},\{C_2\}\}$. We now need to consider C_1 and C_2 together: suppose that A has a block-permutation structure on a partition $\Omega=\dot{\bigcup}_{j=1}^{d}\Omega_j$ and that there exist $i\in C_1$ and $j\in C_2$ such that $i,j\in\Omega_1$. In view of what we have observed in the case $m=1$, it must hold that $d|l_1$ and $d|l_2$, so $d|gcd(l_1,l_2)$. On the other hand, if $d|gcd(l_1,l_2)$, we can partition C_1 in d subsets of size $d'_1=l_1d^{-1}$ and C_2 in d subsets of size $d'_2=l_2d^{-1}$. We can hence create from them d different partitions of $[n]$, each of them made of d blocks of size $d'_1+d'_2$. This justifies the terms of Equation (C.1) when $\mathcal{P}=\{\{C_1,C_2\}\}$; the term -2 appears in order to avoid considering the trivial partitions.

For $m\geq 3$, the idea is the following. Consider $\mathcal{P}\in\mathscr{P}_{\sigma_A}$, $\mathcal{P}=\{P_1,\ldots,P_{r(\mathcal{P})}\}$ and for every i, $P_i=\{C_{j_1^i},\ldots,C_{j^i_{|P_i|}}\}$: the term $\prod_{i=1}^{|\mathcal{P}|}\ \sum_{d|gcd(l_{P_i})} d^{|P_i|-1}$ indicates the number of partitions of $[n]$ on which A has a block-permutation structure with the property that, for every $C_{j_k^i}$ and $C_{j_k^{i'}}$ such that $i\neq i'$, elements of $C_{j_k^i}$ and elements of $C_{j_k^{i'}}$ are never in the same block. The term $\sum_{d|gcd(l_{P_i})} d^{|P_i|-1}$ counts in how many ways we can partition $\bigcup_k C_{j_k^i}$ in order to have in each block of the partition at least one element from every $C_{j_k^i}$ (this partition will have d blocks of size $\sum_{k=1}^{|P_i|} d'_{j_k^i}$).

For Equation (C.2), we apply the same reasoning of the previous case but the fact that we just have to consider the partitions with blocks of the same size, which is guaranteed by the condition:
$d'_{j_1^1}+\ldots+d'_{j^1_{|P_1|}}=d'_{j_1^2}+\ldots+d'_{j^2_{|P_2|}}=\ldots=d'_{j_1^r}+\ldots+d'_{j^r_{|P_r|}}$. □