A Structural Complexity Analysis of Synchronous Dynamical Systems

Eduard Eiben[1], Robert Ganian[2], Thekla Hamm[3], Viktoriia Korchemna[2]

[1] Department of Computer Science, Royal Holloway, University of London, UK
[2] Algorithms and Complexity Group, TU Wien, Austria
[3] Utrecht University, Netherlands
Eduard.Eiben@rhul.ac.uk, {rganian, vkorchemna}@ac.wien.ac.at, t.l.s.hamm@uu.nl

Abstract

Synchronous dynamic systems are well-established models that have been used to capture a range of phenomena in networks, including opinion diffusion, spread of disease and product adoption. We study the three most notable problems in synchronous dynamic systems: whether the system will transition to a target configuration from a starting configuration, whether the system will reach convergence from a starting configuration, and whether the system is guaranteed to converge from every possible starting configuration. While all three problems were known to be intractable in the classical sense, we initiate the study of their exact boundaries of tractability from the perspective of structural parameters of the network by making use of the more fine-grained parameterized complexity paradigm.

As our first result, we consider treewidth—as the most prominent and ubiquitous structural parameter—and show that all three problems remain intractable even on instances of constant treewidth. We complement this negative finding with fixed-parameter algorithms for the former two problems parameterized by treedepth, a well-studied restriction of treewidth. While it is possible to rule out a similar algorithm for convergence guarantee under treedepth, we conclude with a fixed-parameter algorithm for this last problem when parameterized by treedepth and the maximum in-degree.

Introduction

Synchronous dynamic systems are a well-studied model used to capture a range of diffusion phenomena in networks (Rosenkrantz et al. 2021; Chistikov et al. 2020). Such systems have been used, e.g., in the context of social contagions (e.g., the spread of information, opinions, fads, epidemics) as well as product adoption (Adiga et al. 2019; Gupta et al. 2018a; Ogihara and Uchizawa 2017).

Informally, a synchronous dynamic system (SyDS) consists of a directed graph G (representing an underlying network) with each node v having a local function f_v and containing a state value from a domain $\mathbb{B}$, which may evolve over discrete time steps[1]. While each node v begins with an initial state value at time 0, at each subsequent time step it receives an updated value by invoking the node's local function f_v on the state value of v and of all nodes with arcs to v (i.e., the closed in-neighborhood of v). In line with recent works (Rosenkrantz et al. 2021; Chistikov et al. 2020), here we focus our attention to the Boolean-domain case with deterministic functions, which is already sufficiently rich to model a variety of situations. SyDS with Boolean domains are sometimes also called synchronous Boolean networks, especially in the context of systems biology (Ogihara and Uchizawa 2020; Akutsu et al. 2007; Kauffman et al. 2003).

A central notion in the context of SyDS is that of a *configuration*, which is a tuple specifying the state of each node at a certain time step. In several use cases of SyDS, there are clearly identifiable configurations that are either highly desirable (e.g., when dealing with information propagation), or highly undesirable (when modeling the spread of a disease or computer virus). Indeed, the REACHABILITY problem (Rosenkrantz et al. 2021)—deciding whether a given target configuration will be reached from a given starting configuration—is a classical computational problem on SyDS (Ogihara and Uchizawa 2017; Akutsu et al. 2007).

In other settings such as in opinion diffusion (Auletta, Ferraioli, and Greco 2018; Auletta et al. 2017), we do not ask for the reachability of a specific configuration, but rather whether the system eventually converges into a *fixed point*, i.e., a configuration that transitions into itself. This idea has led to the study of two different problems on SyDS (Chistikov et al. 2020): in CONVERGENCE we ask whether the system converges (to an arbitrary fixed point) from a given starting configuration, while in CONVERGENCE GUARANTEE we ask for a much stronger property—notably, whether the system converges from all possible configurations.

In view of the fundamental nature of these three problems, it is somewhat surprising that so little is known about their complexity. The PSPACE-completeness of CONVERGENCE and CONVERGENCE GUARANTEE has been established two years ago (Chistikov et al. 2020), while the PSPACE-completeness of REACHABILITY on directed acyclic networks was established even more recently (Rosenkrantz et al. 2021). Earlier, Barrett et al. (Barrett et al. 2006) established the PSPACE-completeness of REACHABILITY on general directed networks of bounded treewidth and degree, albeit the bounds obtained in that work are very large. In spite of these advances, we still lack a detailed understanding of the complexity of fundamental problems on SyDS.

Contribution. Since REACHABILITY, CONVERGENCE

[1]Formal definitions are provided in the Preliminaries.

and CONVERGENCE GUARANTEE are all computationally intractable on general SyDS, it is natural to ask whether this barrier can be overcome by exploiting the structural properties of the input network. In this paper, we investigate these three problems through the lens of *parameterized complexity* (Downey and Fellows 2013; Cygan et al. 2015), a computational paradigm which offers a refined view into the complexity-theoretic behavior of problems. In this setting, we associate each input $\mathcal{I}$ with a numerical parameter k that captures a certain property of $\mathcal{I}$, and ask whether there is an algorithm that can solve such inputs in time $f(k) \cdot |\mathcal{I}|^{\mathcal{O}(1)}$ (for some computable function f)—if yes, the problem is called *fixed-parameter tractable*, and the class FPT of all such problems is the central notion of tractability in the parameterized setting.

We begin our investigation by considering the most widely studied and prominent graph parameter, notably *treewidth* (Robertson and Seymour 1984). Treewidth intuitively captures how "tree-like" a network is, and it is worth noting that in addition to its fundamental nature, the structure of real-world networks has been demonstrated to attain low treewidth in several settings (Maniu, Senellart, and Jog 2019). While CONVERGENCE GUARANTEE was already known to be intractable even on networks of constant treewidth (Rosenkrantz et al. 2021, Theorem 5.1)[2], previous reductions for REACHABILITY and CONVERGENCE only apply to networks of high treewidth (Chistikov et al. 2020). Here, we show:

Theorem 1. REACHABILITY *and* CONVERGENCE *are* PSPACE*-complete, even on SyDSs of treewidth* 2 *and maximum in-degree* 3*.*

The main technical contribution within the proof is the construction of a non-trivial counter which can loop over all exponentially many configurations of a set of nodes and whose structure is nothing more than a directed path. We believe the existence of such a counter is surprising and may be of independent interest; it contrasts previous counter constructions which relied on much denser connections between the nodes, but interestingly its simple structure comes at the cost of the configuration loop generated by the counter being highly opaque.

Intractability w.r.t. treewidth draws a parallel to the complexity behavior of the classical QBF problem—an archetypical PSPACE-complete problem which also remains PSPACE-complete on instances with bounded treewidth (Atserias and Oliva 2014). In fact, while being based on entirely different ideas, our reduction and Atserias' and Oliva's construction for QBF also show intractability for a slight restriction of treewidth called *pathwidth*, but do not exclude tractability under a related parameter *treedepth* (Nesetril and de Mendez 2012). Investigating the complexity of our problems under the parameter treedepth is the natural next choice, not only because it lies at the very boundary of intractability, but also because of its successful applications for a variety of other problems (Ganian et al. 2020; Ganian and Ordyniak 2018; Gutin, Jones, and Wahlström 2016) and its close connection to the maximum path length in the network[3]. While the complexity of QBF parameterized by treedepth remains a prominent open problem, as our second main technical contribution we show:

Theorem 2. REACHABILITY *and* CONVERGENCE *are fixed-parameter tractable when parameterized by the treedepth of the network.*

The main idea behind the proof of Theorem 2 is to show that the total periodicity of the configurations is upper-bounded by a function of the treedepth, and this fact then enables us to argue the correctness of an iterative pruning step that allows us to gradually reduce the instance to an equivalent one of bounded size. As for the third problem (CONVERGENCE GUARANTEE), fixed-parameter tractability w.r.t. treedepth is excluded by the intractability of the problem on stars.

While these results already provide a fairly tight understanding of the complexity landscape for two out of the three studied problems, they do raise the question of which structural properties of the network can guarantee the tractability of CONVERGENCE GUARANTEE. Intuitively, one of the main difficulties when dealing with CONVERGENCE GUARANTEE is that it is not even possible to enumerate all possible starting configurations of the network. Yet, in spite of this seemingly critical problem, we conclude by establishing fixed-parameter tractability of CONVERGENCE GUARANTEE when parameterized by treedepth plus the maximum in-degree:

Theorem 3. CONVERGENCE GUARANTEE *is fixed-parameter tractable when parameterized by the treedepth plus the maximum in-degree of the network.*

Our results are summarized in Table 1. For completeness, we remark that the results are robust in terms of the type of functions that can be used—in particular, all our algorithmic results hold even if we assume that the functions are black-box oracles. Moreover, in some settings it may be useful to ask whether a target configuration and/or convergence is reached up to an input-specified time point; incorporating this as an additional constraint leads to a strict generalization of REACHABILITY, CONVERGENCE and CONVERGENCE GUARANTEE, and all of the algorithms provided here can also directly solve these more general problems.

Preliminaries

We use standard terminology and notation for directed simple graphs (Diestel 2012), which will serve as models for the networks considered in this paper. We use $\delta^-(v)$ to denote the *in-neighbourhood* of a node v in a directed graph, i.e., the set of all nodes w such that the graph contains an edge wv which starts in w and ends in v.

It will be useful for us to consider tuples as implicitly indexed. This means, for two sets A and B we use B^A to denote the set of tuples with $|A|$-many entries, each of which is an element of B *and* at the same time we associate each of the entries with a fixed element of A. For a tuple $x \in B^A$

[2]In fact, the problem can be shown to be intractable even on stars via a simple argument.

[3]A class of networks has bounded treedepth if and only if there is a bound on the length of any undirected path.

	Unrestricted	Treewidth	Treedepth	Treedepth + in-degree
REACHABILITY	PSPACE-c (Rosenkrantz et al. 2021)	**PSPACE-c**†	**FPT**	FPT
CONVERGENCE	PSPACE-c (Chistikov et al. 2020)	**PSPACE-c**	**FPT**	FPT
CONVERGENCE GUARANTEE	PSPACE-c (Chistikov et al. 2020)	coNP-h	coNP-h	**FPT**

Table 1: Summary of our main results (marked in bold). These include (1) the PSPACE-completeness of the former two problems on networks of constant treewidth, (2) their fixed-parameter tractability with respect to the parameter treedepth, and (3) a fixed-parameter algorithm for CONVERGENCE GUARANTEE when parameterized by treedepth plus the in-degree of the network. The coNP-hardness of the latter problem on networks of constant treedepth and treewidth follows from previous work (Rosenkrantz et al. 2021, Theorem 5.1)).
† The PSPACE-completeness of REACHABILITY on inputs of bounded treewidth was already shown by Barrett et al. (Barrett et al. 2006), albeit the constants used in that reduction were very large while here we establish intractability for treewidth 2.

and an element $a \in A$, we denote by x_a the entry of x that is associated with a.

Synchronous Dynamic Systems. A *synchronous dynamic system* (*SyDS*) $\mathcal{S} = (G, \mathbb{B}, \{f_v \mid v \in V(G)\})$ consists of an *underlying directed graph* (the *network*) G, a *node state domain* $\mathbb{B}$, and for each node $v \in V(G)$ its *local function* $f_v \colon \mathbb{B}^{\delta^-(v)\cup\{v\}} \to \mathbb{B}$. In line with previous literature, in this paper we will always consider $\mathbb{B} = \{0,1\}$ to be binary; however, all results presented herein can be straightforwardly generalized to any fixed (i.e., constant-size) domain. A *configuration* of a SyDS is a tuple in $\mathbb{B}^{V(G)}$. The *successor* of a configuration x is the configuration y which is given by $y_v = f_v(x_v)$ for every $v \in V(G)$. A configuration x is a *fixed point* of a SyDS if it is its own successor, i.e. for all $v \in V(G)$, $f_v(x_v) = x_v$.

Given two configurations x and y of a SyDS $\mathcal{S}$, we say y is *reachable* from x if there is a sequence of configurations of $\mathcal{S}$ starting in x and ending in y such that every configuration in the sequence is followed by its successor.

The three problems on synchronous dynamic systems considered in this paper are formalized as follows:

REACHABILITY
Input: A SyDS $\mathcal{S} = (G, \mathbb{B}, \{f_v \mid v \in V(G)\})$, a configuration x of $\mathcal{S}$ called *starting configuration* and a configuration y of $\mathcal{S}$ called *target configuration*.
Task: Determine whether y is reachable from x.

CONVERGENCE
Input: A SyDS $\mathcal{S} = (G, \mathbb{B}, \{f_v \mid v \in V(G)\})$ and a configuration x of $\mathcal{S}$ called *starting configuration*.
Task: Determine whether there is a fixed point of $\mathcal{S}$ that is reachable from x.

CONVERGENCE GUARANTEE
Input: A SyDS $\mathcal{S} = (G, \mathbb{B}, \{f_v \mid v \in V(G)\})$.
Task: Determine whether for every configuration x of $\mathcal{S}$ there is a fixed point of $\mathcal{S}$ that is reachable from x.

Treewidth and Treedepth. Similarly to other applications of treewidth on directed networks (Ganian, Hamm, and Ordyniak 2021; Gupta et al. 2018b), in this submission we consider the treewidth and treedepth of the *underlying undirected graph*, which is the simple graph obtained by ignoring the orientations of all arcs in the graph. While directed analogues for treewidth have been considered in the literature, these have constant values on DAGs and hence cannot yield efficient algorithms for any of the considered problems (Rosenkrantz et al. 2021).

While treewidth has a rather technical definition involving bags and decompositions (Robertson and Seymour 1984; Downey and Fellows 2013), for the purposes of this article it will suffice to remark that graphs where removing a single node results in a tree have treewidth at most 2.

Treedepth is a parameter closely related to treewidth—in particular, the treedepth of a graph is lower-bounded by its treewidth. A useful way of thinking about graphs of bounded treedepth is that they are (sparse) graphs with no long paths. We formalize the parameter below.

A *rooted forest* $\mathcal{F}$ is a disjoint union of rooted trees. For a node x in a tree T of $\mathcal{F}$, the *height* (or *depth*) of x in $\mathcal{F}$ is the number of nodes in the path from the root of T to x. The *height of a rooted forest* is the maximum height of a node of the forest. Let $V(T)$ be the node set of any tree $T \in \mathcal{F}$.

Definition 4 (Treedepth). *Let the* closure *of a rooted forest $\mathcal{F}$ be the graph $\lambda(\mathcal{F}) = (V_c, E_c)$ with the node set $V_c = \bigcup_{T\in\mathcal{F}} V(T)$ and the edge set $E_c = \{xy \mid x$ is an ancestor of y in some $T \in \mathcal{F}\}$. A* treedepth decomposition *of a graph G is a rooted forest $\mathcal{F}$ such that $G \subseteq \lambda(\mathcal{F})$. The* treedepth $\mathrm{td}(G)$ *of a graph G is the minimum height of any treedepth decomposition of G.*

It is known that an optimal-width treedepth decomposition can be computed by a fixed-parameter algorithm (Nesetril and de Mendez 2012; Reidl et al. 2014; Nadara, Pilipczuk, and Smulewicz 2022) and also, e.g., via a SAT encoding (Ganian et al. 2019); hence, in our algorithms we assume that such a decomposition is provided on the input.

The Path-Gadget and Hardness for Treewidth

We provide a construction showing that even SyDSs which are directed paths can reach an exponential number of configurations. This functions as a crucial gadget for showing PSPACE-hardness of REACHABILITY and CONVERGENCE, but is also an interesting result in its own right as it significantly simplifies known constructions of SyDSs with exponential periods (Rosenkrantz et al. 2021).

Theorem 5. *For every $n \in \mathbb{N}$, there is a SyDS $\mathcal{X}^{(n)} = (G, \mathbb{B} = \{0,1\}, \{f_v \mid v \in V(G)\})$ and an initial configuration x^0, such that*

- *G is a directed path $v_1v_2 \dots v_{2n}$,*
- *x^0 is the all-zero configuration,*
- *the successor of x^0 is the configuration x^1 such that $x^1_{v_j} = 1$ if and only if $j = 1$ or $(j \mod 2) = 0$, and*
- *for every $i \in [n+1]$ and every tuple $t \in \mathbb{B}^i$, there exists $q \in \{0, \dots, 2^i - 1\}$ such that, for every $p \in \mathbb{N}$, the configuration of $\mathcal{X}^{(n)}$ after $2^i \cdot p + q$ successor steps restricted to $v_1, v_2, v_4, \dots, v_{2i-4}, v_{2i-2}$ is equal to t.*

In particular, for every $p \in \mathbb{N}$, after $2^i \cdot p$ steps, the node state of each of the nodes $v_1, v_2, v_4, \dots, v_{2i-4}, v_{2i-2}$ is 0.

Proof Sketch. Let us fix some $n \in \mathbb{N}$. For simplicity of notation, we will denote the local function f_{v_j} for the node v_j by f_j. The local function for the node v_1 is $f_1 \colon \mathbb{B} \to \mathbb{B}$ given by $f_1(b) = 1 - b$, that is the configuration alternates between 0 and 1. For every $i \in [n]$, the local function for the node v_{2i} is $f_{2i} \colon \mathbb{B}^2 \to \mathbb{B}$ given by $f_{2i}(b_1, b_2) = (b_1 - b_2 + 1) \cdot (b_2 - b_1 + 1)$. Equivalently, $f_{2i}(b_1, b_2) = 1$ if and only if $b_1 = b_2$ and $f_{2i}(b_1, b_2) = 0$ otherwise, and so the function is simply an evaluation of the equivalence relation. Next, for every $i \in [n-1]$, the local function for the node v_{2i+1} is $f_{2i+1} \colon \mathbb{B}^2 \to \mathbb{B}$ given by $f_{2i+1}(b_1, b_2) = b_1 \cdot (1 - b_2)$. Equivalently, the configuration on v_{2i+1} is 1 if and only if in the previous step, the configuration on v_{2i+1} was 0 and the configuration on v_{2i} was 1.

This finishes the description of the SyDS. We will now prove that the structure of configurations over the time steps has the desired properties. To this end, we denote the initial configuration x^0 and for configuration x^i, we denote its successor as x^{i+1}. Moreover, for simplicity of the notation, we denote by x^i_j the state of the node v_j in i-th step, that is $x^i_{v_j}$.

Clearly, $x^i_1 = 0$ if $i \mod 2 = 0$ and $x^i_1 = 1$ if $i \mod 2 = 1$. Moreover, it is also easy to verify now that $x^1_j = 1$ if and only if $j = 1$ or $(j \mod 2) = 0$ since for $j \geq 2$, we have by the definition of the local functions that $f_j(0,0) = 1$ if and only if j is even. In order to complete the proof, we first establish the following claim.

Claim 1 ($\star$). *For all $j \geq 1$ and $i \in \mathbb{N}$, it holds that $x^i_j = x^{i'}_j$, where $i' = i \mod 2^{\lfloor \frac{j}{2} \rfloor + 1}$. Moreover, if j is even, then $x^i_j = 1 - x^{i''}_j$, where $i'' = i + 2^{\lfloor \frac{j}{2} \rfloor}$.*

Proof of the Claim (Sketch). Let us first introduce some notation that will help with exposition. For a node v_j, we will let the *period vector* for the node v_j be the vector $\mathrm{pv}_j = (x^0_j x^1_j \dots x^{2^{\lfloor \frac{j}{2} \rfloor + 1} - 1}_j)$ of length $2^{\lfloor \frac{j}{2} \rfloor + 1}$. For simplicity, we mostly split the vector into smaller pieces consisting of at most three entries. Moreover, we use the exponent for a piece to specify how many times the same piece repeats in a row in the vector. For example, we could write the vector (01001011) as $(010)^2(11)$. The reason we call these the "period vectors" is that we will show that for all $i \in \mathbb{N}$ we have $x^i_j = x^{i'}_j$, where $i' = i \mod 2^{\lfloor \frac{j}{2} \rfloor + 1}$; in other words, the period vector repeats cyclically as the state x^i_j of the node v_j starting from x^0_j.

Let us proceed by computing the few first period vectors. This is straightforward, as we are always computing pv_{j+1} from pv_j using the fact that $x^0_{j+1} = 0$. In each of the cases, we also make sure that the period vector pv_j indeed repeats as the state changes of node v_j. Since the length of the period vector pv_{j+1} is either the same as the period vector pv_j or double the length of period vector pv_j, it suffices to verify that $x^{2^{\lfloor \frac{j}{2} \rfloor + 1}} = 0$ (assuming that we did the check for pv_j already). Moreover, we also check that for even j we have $x^i_j = 1 - x^{i''}_j$, where $i'' = i + 2^{\lfloor \frac{j}{2} \rfloor}$.

- $\mathrm{pv}_1 = (01)$;
- $\mathrm{pv}_2 = (01)(10) = (011)(0)$;
- $\mathrm{pv}_3 = (001)(0)$;
- $\mathrm{pv}_4 = (010)(0)(101)(1) = (010)^2(11)$;
- $\mathrm{pv}_5 = (001)^2(01)$;
- $\mathrm{pv}_6 = (010)^2(01)(101)^2(10) = (010)^2(011)^3(0)$;
- $\mathrm{pv}_7 = (001)^5(0)$;
- $\mathrm{pv}_8 = (010)^5(0)(101)^5(1) = (010)^6(110)^4(11)$;
- $\mathrm{pv}_9 = (001)^6(010)^4(01)$;
- $\mathrm{pv}_{10} = (010)^6(011)^4(01)(101)^6(100)^4(10)$ $= (010)^6(011)^{11}(001)^4(0)$;
- $\mathrm{pv}_{11} = (001)^{17}(000)(100)^3(1)$;
- $\mathrm{pv}_{12} = (010)^{17}(010)(110)^3(1)(101)^{17}(101)(001)^3(0)$ $= (010)^{18}(110)^{21}(100)^3(10)$;

The reason we computed the first 12 entries is that we will show that from here onward, the structure of the period vectors for v_j will begin to follow a cyclic pattern. Namely, we will distinguish the remaining nodes by $(j \mod 4)$ and show that the structure of their period vectors is the same as the structure of period vectors for v_9, v_{10}, v_{11}, and v_{12} respectively. More precisely, we show that for $j \geq 9$:

- If $j = 4k$, $\mathrm{pv}_j = (010)^\ell(110)^p(100)^q(10)$ for some $\ell, p, q \in \mathbb{N}$ such that $\ell + p + q = \frac{2^{2k+1}-2}{3}$.
- If $j = 4k+1$, $\mathrm{pv}_j = (001)^p(010)^q(01)$ for some $p, q \in \mathbb{N}$ such that $p + q = \frac{2^{2k+1}-2}{3}$.
- If $j = 4k+2$, $\mathrm{pv}_j = (010)^\ell(011)^p(001)^q(0)$ for some $\ell, p, q \in \mathbb{N}$ such that $\ell + p + q = \frac{2^{2k+2}-1}{3}$.
- If $j = 4k+3$, $\mathrm{pv}_j = (001)^p(000)(100)^q(1)$ for some $p, q \in \mathbb{N}$ such that $p + q = \frac{2^{2k+2}-4}{3}$.

To complete the proof of the claim, it suffices to verify the period vectors have the stated structure and that each of the cases satisfies the stated conditions. ■

Observe that the above claim already implies that for all $i \in [n+1]$ and $p \in \mathbb{N}$, the state of the node v_{2i-2} after $2^i \cdot p$ steps is 0 (i.e., $x^{2^i \cdot p}_{2i-2} = 0$).

Now, given the above claim on the structure of the configuration, it is also rather straightforward to show that for all $i \in [n+1]$ and all tuples $t \in \mathbb{B}^i$, there exists $q \in \{0, \dots, 2^i - 1\}$ such that, for every $p \in \mathbb{N}$, the configuration of $\mathcal{X}^{(n)}$ after $2^i \cdot p + q$ successor steps restricted to the nodes $v_1, v_2, v_4, \dots, v_{2i-4}, v_{2i-2}$ is equal to t. First note that for $i = 1$, the node v_1 flips always between 0 and 1 and the statement holds. Moreover, for $i = 2$, the nodes v_1 and v_2 have the following transitions through configurations $00 \rightsquigarrow 11 \rightsquigarrow 01 \rightsquigarrow 10 \rightsquigarrow 00$. Let us assume that for some $i \in [n]$ for every tuple $t \in \mathbb{B}^i$, there exists $q \in \{0, \dots, 2^i - 1\}$ such that for every $p \in \mathbb{N}$, the configuration of $\mathcal{X}^{(n)}$ after $2^i \cdot p + q$ successor steps restricted to the nodes $v_1, v_2, v_4, \dots, v_{2i-4}, v_{2i-2}$ is equal to t.

Let $t \in \mathbb{B}^{i+1}$, we will show that there exists $q \in \{0, \dots, 2^{i+1} - 1\}$ such that for every $p \in \mathbb{N}$, the configuration of $\mathcal{X}^{(n)}$ after $2^{i+1} \cdot p + q$ successor steps restricted to the nodes $v_1, v_2, v_4, \dots, v_{2i-2}, v_{2i}$ is equal to t. Denote by t' the restriction of t to the first i bits. By induction hypothesis, there exists $q' \in \{0, \dots, 2^i - 1\}$ such that for every $p' \in \mathbb{N}$, the configuration of $\mathcal{X}^{(n)}$ after $2^i \cdot p' + q'$ successor steps restricted to the nodes $v_1, v_2, v_4, \dots, v_{2i-4}, v_{2i-2}$ is equal to t'. By the above claim,

$$x_{2i}^{2^i \cdot p' + q'} = 1 - x_{2i}^{2^i \cdot p' + q' + 2^i}.$$

It follows that for every $p \in \mathbb{N}$, t appears as the restriction of the configuration to $v_1, v_2, v_4, \dots, v_{2i-2}, v_{2i}$ after $2^{i+1} \cdot p + q$ steps, where q is either equal to q' or to $q' + 2^i$. Repeating the same argument for all $t \in \mathbb{B}^{i+1}$ completes the proof. $\square$

With the path-gadget ready, we can proceed to establishing our first hardness result. The idea of the reduction used in the proof loosely follows that of previous work (Rosenkrantz et al. 2021), but uses multiple copies of the path gadget from Theorem 5 (instead of a single counting gadget that has a tournament as the network) to avoid dense substructures. Note that the usage of the structurally simpler, but behaviourally more complicated, path-gadget requires some additional changes to the construction. Furthermore, compared to the earlier result of Barrett et al. (Barrett et al. 2006), we obtain much smaller bounds on the treewidth and maximum degree (on the other hand, our reduction does not provide a bound on the bandwidth of the graph).

Theorem 1. Reachability *and* Convergence *are* PSPACE*-complete, even on SyDSs of treewidth* 2 *and maximum in-degree* 3*.*

Proof Sketch. Both problems are known and easily seen to be in PSPACE (Chistikov et al. 2020). For showing hardness, we give a reduction starting from QBF which is known to be PSPACE-hard (Garey and Johnson 1979) and takes as input a formula $Q_n x_n \dots Q_1 x_1 \varphi$ where all Q_i are either $\forall$ or $\exists$ and φ is a 3SAT-formula.

From this we construct a SyDS $\mathcal{S}$ which converges to one specific configuration (the all-one configuration) if and only if the given formula is true.

The overall idea is for (multiple copies[4] of) the SyDS $\mathcal{X}^{(n)}$ (see Theorem 5) to iterate over all possible truth assignments of the variables x_1 to x_n. While it would be possible to simply associate every significant node $\mathcal{X}^{(n)}$ with a variable, we will add auxiliary *variable nodes* as to go through the variable assignments in a well-structured way. Then *clause nodes* aggregate the implied truth values of the clauses of φ. These clause nodes can then be used as incoming neighbours for *subformula nodes* to derive the truth value of formulas $Q_i x_i \dots Q_1 x_1 \varphi$ respectively. Once the truth value of $Q_n x_n \dots Q_1 x_1 \varphi$ is determined to be true we use a *control node* to set the whole SyDS to a configuration with all entries being 1. Otherwise the copies of $\mathcal{X}^{(n)}$ continue to loop through their period. We call the nodes v_1 and v_{2i} with $i \in [n]$ of a copy of $\mathcal{X}^{(n)}$ *significant.*

To not obfuscate the core of the construction we initially allow an arbitrary node degree and give a description on how to reduce it to a constant at the end of the proof. The overall structure of the underlying graph which we construct is depicted in Figure 1. The starting configuration will have an entry 0 for all nodes except the ones in the copies of $\mathcal{X}^{(n)}$ which will instead start at configuration x^1 from Theorem 5.

[4] all copies will be independent, have no additional in-neighbours and start at the same configuration

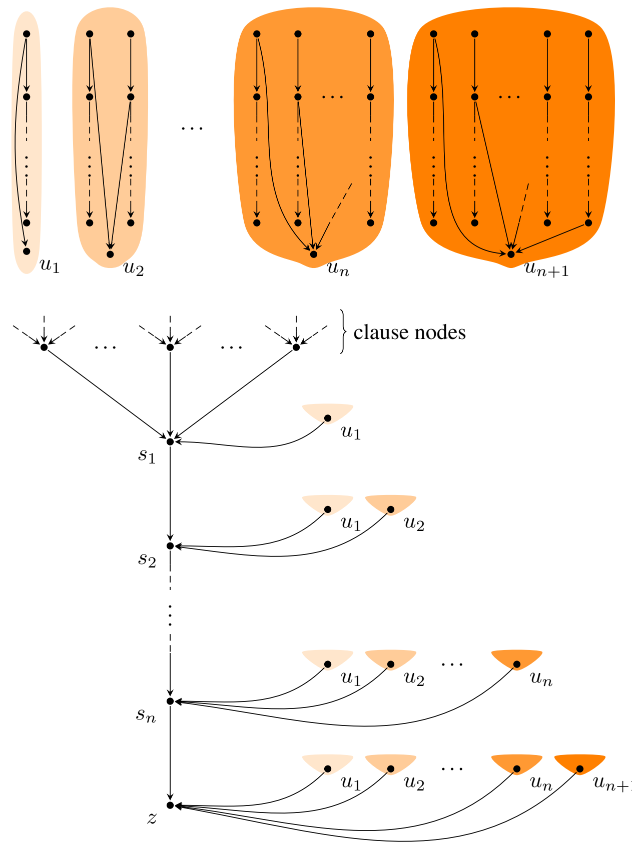

Figure 1: Overview of the network constructed in the proof of Theorem 1. To not clutter the figure, the arcs from z to all other nodes omitted, and subgraphs introduced for variable nodes are illustrated in detail at the top and only indicated by orange (darker shade $\simeq$ higher index) in all other places.

We describe our construction in a hierarchical manner, going from the most fundamental to the most complex (in the sense that they rely on the earlier ones) types of nodes.

For each variable x_i, a variable node u_i has an incoming arc from the j-th significant node of the j-th of i private copies of $\mathcal{X}^{(n)}$ for all $j \in [i]$. In addition we add an auxiliary node u_{n+1} which will not actually represent a variable of φ itself but otherwise be treated and hence also be referred to as a variable node for convenience. u_{n+1} has incoming arcs from the j-th significant node of the j-th of $n+1$ private copies of $\mathcal{X}^{(n)}$ for all $j \in [n+1]$. It will become apparent later that it is useful to introduce u_{n+1} and that $X^{(n)}$ cycles only after 2^{n+1} time steps rather than just 2^n for technical reasons. We define f_{u_i} to change the configuration at u_i if and only if the configuration of all of its in-neighbours is 0 and to leave it unchanged otherwise. To give some intuition for the behaviour of the variable nodes with this construction it is useful to refer to the *position* of a tuple $t \in \mathbb{B}^{\{v_1\}\cup\{v_{2i}|i\in[n]\}}$ as the number of time steps required to reach a configuration of $\mathcal{X}^{(n)}$ that is equal to t on the significant nodes starting from the configuration x^1. Theorem 5 upper-bounds the position of a tuple by 2^{n+1}.

Claim 2. *At any time step, taken together the variable nodes $u_{n+1} \dots u_1$ encode the position of the configuration of the significant nodes of the path gadgets in binary (that are the same on different copies) in the previous time step.*

This means that the variable nodes just like the significant nodes of $\mathcal{X}^{(n)}$ have the property that on them any shared configuration is reachable, and in addition these shared configurations are reached in the order of the binary numbers they represent. Clearly this construction for each u_i leads to an underlying tree and hence has treewidth 1.

For each clause $C = x_{j_1} \vee x_{j_2} \vee x_{j_3}$ of φ, a clause node c has incoming arcs from private variable nodes u_{j_1}, u_{j_2} and u_{j_3} and its local function is 1 if and only if the configuration of at least one of these in-neighbours is 1. This means we will have a copy of each variable node with its own copies of $\mathcal{X}^{(n)}$ for each occurrence of that variable in φ. The underlying graph formed by this construction is still a tree.

For each $i \in [n]$ we introduce a subformula node s_i. s_1 has incoming arcs from all clause nodes and a private variable node u_1. For $i \in [n] \setminus \{1\}$, s_i has incoming arcs from s_{i-1} and private variable nodes $u_1, \dots, u_i$. As before, this construction still results in a tree.

Now we turn to the definition of the local functions for subformula nodes. We define f_{s_1} as follows: If the entry from u_1 is 1, then f_{s_1} maps to 1 if and only if all entries from the clause nodes are 1. Otherwise if $Q_1 = \exists$, f_{s_1} maps to 1 if and only if the entry for s_1 *or* all entries from the clause nodes are 1. Otherwise the entry from u_1 is 0 and $Q_1 = \forall$, and we let f_{s_1} map to 1 if and only if the entry for s_1 *and* all entries from the clause nodes are 1. For $i \in [n] \setminus \{1\}$ we define f_{s_i} as follows: If the incoming variable nodes in the order $u_i \dots u_1$ are the binary encoding of $2^{i-1} + i - 1$ then f_{s_i} maps to the entry for s_{i-1}. Otherwise if the incoming variable nodes in the order $u_i \dots u_1$ are the binary encoding of $i-1$ and $Q_i = \exists$ then f_{s_i} maps to 1 if and only if the entry for s_i *or* the entry for s_{i-1} is 1. Otherwise if the incoming variable nodes in the order $u_i \dots u_1$ are the binary encoding of $i-1$ and $Q_i = \forall$ then f_{s_i} maps to 1 if and only if the entry for s_i *and* the entry for s_{i-1} is 1. Otherwise the incoming variable nodes in the order $u_i \dots u_1$ do not encode $2^{i-1} + i - 1$ or $i-1$ and we let f_{s_i} map to the entry for s_i.

It now remains to show (via a non-trivial inductive argument) that at time step $2^n + n$, the configuration at s_n contains 1 if and only if the QBF input is true. In this case and only in this case we want the SyDS to converge, and in particular interrupt the looping through different configurations of the copies of $\mathcal{X}^{(n)}$. For this purpose we introduce a control node z which has an incoming arc from every node of a private copy of every variable node $u_{n+1} \dots u_1$ and and an outgoing arc to all other nodes. We let f_z map to 1 if and only if the entries for the variable nodes encode $2^n + n$ in binary (this is where the auxiliary node u_{n+1} becomes important as this number cannot be encoded by the even-index nodes of only $\mathcal{X}^{(n)}$) and the entry for r_n is 1 or if the entry for f_z is already 1. We add an entry for f_z to all previously defined local functions and condition the previously defined behaviour on this entry being 0. As soon as this entry is 1 the local functions of all nodes should also map to 1. In this way the all-one configuration is reachable from the all-zero configuration if and only if the QBF input formula is true and this is the only way for the SyDS to converge.

Note that the network constructed in this manner consists of a directed tree plus the additional vertex z.

Modification for constant in-degree. To obtain a network with constant maximum in-degree, note that by *currying* (Curry 1980) one can write every local function that takes polynomially many arguments as a polynomial-length series of functions that each take 2 arguments. In this way we can essentially replace every node with an in-degree of more than two by a tree of polynomial depth in which each node has in-degree 2. $\square$

An Algorithm Using Treedepth

In this section, we exploit the treedepth decomposition of the network to transform the instances of REACHABILITY and CONVERGENCE into equivalent ones of bounded size. Our algorithms then proceed by simulating all possible configurations of the resulting smaller network.

Let $\mathcal{F}$ be the treedepth decomposition of G; without loss of generality, we may assume that $\mathcal{F}$ is a tree. We denote the subtree of $\mathcal{F}$ rooted in v by $\mathcal{F}_v$. Let G_v^* be the subgraph of G induced by the nodes associated to $\mathcal{F}_v$ and the neighbourhood of $\mathcal{F}_v$ in G. Our aim is to iteratively compress nodes of G from the leaves to the root to obtain a graph G' with number of nodes bounded by some function of $\mathrm{td}(G)$.

Lemma 6. *Let $\mathcal{I}$ be an instance of* CONVERGENCE *or* REACHABILITY *and let $\mathcal{F}$ be the treedepth decomposition of G of height L. Assume that for some $v \in V(G)$, all the subtrees rooted in children of v contain at most m nodes. Then $\mathcal{I}$ can be modified in polynomial time to an equivalent instance $\mathcal{I}'$ with network G' which has a treedepth decomposition $\mathcal{F}'$ of height at most L such that $\mathcal{F}' \setminus \mathcal{F}'_v \subseteq \mathcal{F} \setminus \mathcal{F}_v$ and $|V(\mathcal{F}'_v)| \le m^{m-1} \cdot (4^{L+m} \cdot |\mathbb{B}|^{L+m+1})^m + 1$.*

Proof Sketch. We say that two children u and w of v have the same type (denoted by $u \equiv w$) if there exists an isomorphism $\phi : G_u^* \to G_w^*$ that is the identity on the neighbourhood of $\mathcal{F}_u$ such that for every $z \in V(\mathcal{F}_u)$:

- the initial states of z and $\phi(z)$ are the same.
- $f_{\phi(z)}$ acts on $\mathbb{B}^{\delta^-(\phi(z))\cup\{\phi(z)\}}$ as f_z acts on $\mathbb{B}^{\delta^-(z)\cup\{z\}}$ (the orderings of the neighbours of z and their images agree).

In this case, the states of z and $\phi(z)$ coinside at each time step. Since the composition of two such isomorphisms results in an isomorphism with the same properties, $\equiv$ is an equivalence relation on the set of children of v with boundedly many equivalence classes.

Let u and w be the children of v of the same type. Intuitively, since the configurations of the rooted subtrees $\mathcal{F}_u$ and $\mathcal{F}_v$ coincide at each time step, adapting the functions of the ancestors allows us to construct a new network which only preserves one of them. □

Lemma 6 enables us to iteratively compress instances of CONVERGENCE or REACHABILITY.

Corollary 7. *There exists a computable function $h : \mathbb{N} \to \mathbb{N}$ such that every instance $\mathcal{I}$ of* CONVERGENCE *or* REACHABILITY *with $\mathrm{td}(G) = L$ can be transformed in polynomial time into an equivalent instance $\mathcal{I}'$ of the same problem with $|V(G')| \leq h(L)$.*

We are ready to prove the main theorem of the section:

Theorem 2. REACHABILITY *and* CONVERGENCE *are fixed-parameter tractable when parameterized by the treedepth of the network.*

Proof. Given an instance of CONVERGENCE or REACHABILITY with network G of treedepth L, we apply Corollary 7 to transform it into an equivalent instance where G' has at most $h(L)$ nodes. Then G' has at most $|\mathbb{B}|^{h(L)}$ possible configurations. Therefore it suffices to simulate the first $|\mathbb{B}|^{h(L)}$ time steps of the reduced SyDS to solve CONVERGENCE or REACHABILITY. □

Restricting the In-Degree

In this final section, we turn our attention to CONVERGENCE GUARANTEE. In particular, while one cannot hope to extend Theorem 2 to the CONVERGENCE GUARANTEE problem due to known lower bounds (Rosenkrantz et al. 2021), one can observe that the reduction used there requires nodes with high in-degree. Here, we show that when we restrict the inputs by including the in-degree as a parameter in addition to treedepth, the problem becomes fixed-parameter tractable.

Let us start by showing that CONVERGENCE GUARANTEE can be solved efficiently for networks without long directed paths and nodes of large in-degrees. In fact, the same argument also allows us to obtain a more efficient algorithm for CONVERGENCE in this setting.

Lemma 8. CONVERGENCE GUARANTEE *(or* CONVERGENCE*) can be solved in time $|\mathbb{B}|^{2(pd^p+1)} \cdot \mathcal{O}(n^3)$ (or $|\mathbb{B}|^{pd^p+1} \cdot \mathcal{O}(n^3)$, respectively), where:*

- *p is the maximum length of a directed path in the network,*
- *d is the maximum in-degree of the input network, and*
- *n is the number of nodes in the network.*

Proof. For a node $v \in V(G)$, we denote by X_v the set of all $u \in V(G)$ such that G contains a directed path from u to v. Observe that to solve the instance $\mathcal{I} = (S, x)$ of CONVERGENCE (or $\mathcal{I} = S$ of CONVERGENCE GUARANTEE), it is sufficient to solve its restriction to every set X_v (denoted $\mathcal{I}_v = (S_v, x_v)$ or $\mathcal{I}_v = S_v$ correspondingly). Let d and p be the maximum in-degree and length of a simple directed path in G respectively, then each X_v contains at most $pd^p + 1$ elements. Therefore S_v can have at most $|\mathbb{B}|^{pd^p+1}$ different configurations. For CONVERGENCE, we start from x_v, simulate $|\mathbb{B}|^{pd^p+1}$ time steps and check whether the resulting configuration is a fixed point. We return "Yes" if and only if every (S_v, x_v) reaches a fixed point. In case of CONVERGENCE GUARANTEE, we proceed similarly, but for every S_v at first branch over at most $|\mathbb{B}|^{pd^p+1}$ possible starting configurations. Since the number of sets X_v is $\mathcal{O}(n)$ and the simulation of one step requires time of at most $\mathcal{O}(n^2)$, we get the time bounds of $|\mathbb{B}|^{pd^p+1} \cdot \mathcal{O}(n^3)$ and $|\mathbb{B}|^{2(pd^p+1)} \cdot \mathcal{O}(n^3)$ for CONVERGENCE and CONVERGENCE GUARANTEE correspondingly. □

As an immediate corollary, we have:

Theorem 3. CONVERGENCE GUARANTEE *is fixed-parameter tractable when parameterized by the treedepth plus the maximum in-degree of the network.*

However, bounding only the in-degrees of nodes is not sufficient to achieve tractability of the problem. Indeed, by reducing from 3-UNSAT we obtain:

Theorem 9. CONVERGENCE GUARANTEE *is co-*NP*-hard even if G is a DAG with maximum in-degree of* 3.

Concluding Remarks

Our results shed new light on the complexity of the three most fundamental problems on synchronous dynamic systems. They also identify two of these—REACHABILITY and CONVERGENCE—as new members of a rather select club of problems with a significant complexity gap between parameterizing by treewidth and by treedepth. It is perhaps noteworthy that the few known examples of this behavior are predominantly (albeit not exclusively (Gutin, Jones, and Wahlström 2016)) tied to problems relevant to AI research (Ganian and Ordyniak 2018; Ganian et al. 2020; Ganian, Hamm, and Ordyniak 2021).

One question left open for future work is the exact complexity classification of CONVERGENCE GUARANTEE on networks of bounded treedepth or treewidth. Indeed, while previous work (Rosenkrantz et al. 2021) shows that the problem is coNP-complete on DAGs, it is not clear why the problem should be included in coNP on general networks (in particular, while convergence from a fixed starting state is polynomial-time checkable on DAGs, it is PSPACE-complete on general networks). Another question that could be tackled in future work is whether Theorem 3 can be generalized to treewidth instead of treedepth.

Acknowledgements

Robert Ganian and Viktoriia Korchemna acknowledge support by the Austrian Science Fund (FWF, project Y1329). Thekla Hamm acknowledges support by the Austrian Science Fund (FWF, project J4651).

References

Adiga, A.; Kuhlman, C. J.; Marathe, M. V.; Mortveit, H. S.; Ravi, S. S.; and Vullikanti, A. 2019. Graphical dynamical systems and their applications to bio-social systems. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 11: 153–191.

Akutsu, T.; Hayashida, M.; Ching, W. K.; and Ng, K. P. 2007. Control of Boolean networks: Hardness results and algorithms for tree structured networks. *Journal of Theoretical Biology*, 244: 670–679.

Atserias, A.; and Oliva, S. 2014. Bounded-width QBF is PSPACE-complete. *J. Comput. Syst. Sci.*, 80(7): 1415–1429.

Auletta, V.; Caragiannis, I.; Ferraioli, D.; Galdi, C.; and Persiano, G. 2017. Robustness in Discrete Preference Games. In Larson, K.; Winikoff, M.; Das, S.; and Durfee, E. H., eds., *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, 1314–1322. ACM.

Auletta, V.; Ferraioli, D.; and Greco, G. 2018. Reasoning about Consensus when Opinions Diffuse through Majority Dynamics. In Lang, J., ed., *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 49–55. ijcai.org.

Barrett, C. L.; III, H. B. H.; Marathe, M. V.; Ravi, S. S.; Rosenkrantz, D. J.; and Stearns, R. E. 2006. Complexity of reachability problems for finite discrete dynamical systems. *J. Comput. Syst. Sci.*, 72(8): 1317–1345.

Chistikov, D.; Lisowski, G.; Paterson, M.; and Turrini, P. 2020. Convergence of Opinion Diffusion is PSPACE-Complete. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 7103–7110. AAAI Press.

Curry, H. B. 1980. Some Philosophical Aspects of Combinatory Logic. In Barwise, J.; Keisler, H. J.; and Kunen, K., eds., *The Kleene Symposium*, volume 101 of *Studies in Logic and the Foundations of Mathematics*, 85–101. Elsevier.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer. ISBN 978-3-319-21274-6.

Diestel, R. 2012. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer. ISBN 978-3-642-14278-9.

Downey, R. G.; and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer. ISBN 978-1-4471-5558-4.

Ganian, R.; Hamm, T.; and Ordyniak, S. 2021. The Complexity of Object Association in Multiple Object Tracking. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 1388–1396. AAAI Press.

Ganian, R.; Lodha, N.; Ordyniak, S.; and Szeider, S. 2019. SAT-Encodings for Treecut Width and Treedepth. In Kobourov, S. G.; and Meyerhenke, H., eds., *Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments, ALENEX 2019, San Diego, CA, USA, January 7-8, 2019*, 117–129. SIAM.

Ganian, R.; and Ordyniak, S. 2018. The complexity landscape of decompositional parameters for ILP. *Artif. Intell.*, 257: 61–71.

Ganian, R.; Peitl, T.; Slivovsky, F.; and Szeider, S. 2020. Fixed-Parameter Tractability of Dependency QBF with Structural Parameters. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, 392–402.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman. ISBN 0-7167-1044-7.

Gupta, A.; Swarup, S.; Marathe, A.; Vullikanti, A.; Lakkaraju, K.; and Letchford, J. 2018a. Designing Incentives to Maximize the Adoption of Rooftop Solar Technology. In André, E.; Koenig, S.; Dastani, M.; and Sukthankar, G., eds., *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, 1950–1952. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM.

Gupta, S.; Roy, S.; Saurabh, S.; and Zehavi, M. 2018b. Winning a Tournament by Any Means Necessary. In Lang, J., ed., *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 282–288. ijcai.org.

Gutin, G. Z.; Jones, M.; and Wahlström, M. 2016. The Mixed Chinese Postman Problem Parameterized by Pathwidth and Treedepth. *SIAM J. Discret. Math.*, 30(4): 2177–2205.

Kauffman, S.; Peterson, C.; Samuelsson, B.; and Troein, C. 2003. Random Boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences*, 100(25): 14796–14799.

Maniu, S.; Senellart, P.; and Jog, S. 2019. An Experimental Study of the Treewidth of Real-World Graph Data. In Barceló, P.; and Calautti, M., eds., *22nd International Conference on Database Theory, ICDT 2019, March 26-28, 2019, Lisbon, Portugal*, volume 127 of *LIPIcs*, 12:1–12:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Nadara, W.; Pilipczuk, M.; and Smulewicz, M. 2022. Computing treedepth in polynomial space and linear fpt time. *CoRR*, abs/2205.02656.

Nesetril, J.; and de Mendez, P. O. 2012. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer. ISBN 978-3-642-27874-7.

Ogihara, M.; and Uchizawa, K. 2017. Computational complexity studies of synchronous Boolean finite dynamical systems on directed graphs. *Inf. Comput.*, 256: 226–236.

Ogihara, M.; and Uchizawa, K. 2020. Synchronous Boolean Finite Dynamical Systems on Directed Graphs over XOR Functions. In Esparza, J.; and Král', D., eds., *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, 76:1–76:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Reidl, F.; Rossmanith, P.; Villaamil, F. S.; and Sikdar, S. 2014. A Faster Parameterized Algorithm for Treedepth. In Esparza, J.; Fraigniaud, P.; Husfeldt, T.; and Koutsoupias, E., eds., *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, 931–942. Springer.

Robertson, N.; and Seymour, P. D. 1984. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1): 49–64.

Rosenkrantz, D. J.; Marathe, M. V.; Ravi, S. S.; and Stearns, R. E. 2021. Synchronous Dynamical Systems on Directed Acyclic Graphs: Complexity and Algorithms. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 11334–11342. AAAI Press.