Computational Aspects of Analyzing Social Network Dynamics

Chris Barrett [1] **Harry B. Hunt III** [2] **Madhav V. Marathe** [1] **S. S. Ravi** [2]
Daniel J. Rosenkrantz [2] **Richard E. Stearns** [2] **Mayur Thakur** [3]

Abstract

Motivated by applications such as the spread of epidemics and the propagation of influence in social networks, we propose a formal model for analyzing the dynamics of such networks. Our model is a stochastic version of discrete dynamical systems. Using this model, we formulate and study the computational complexity of two fundamental problems (called reachability and predecessor existence problems) which arise in the context of social networks. We also point out the implications of our results on other computational models such as Hopfield networks, communicating finite state machines and systolic arrays.

1 Introduction and Motivation

With the growing importance of social networks, analysis of the dynamics of these networks is attracting the attention of researchers. In this paper, we propose a multiagent-based formal model of social network dynamics and study two fundamental analysis problems arising in that context. We take an abstract view of social network dynamics. In our model, each individual (agent) in the social network is represented by a node in the underlying undirected graph. The edges of the graph model dependencies among the agents. Depending on the context, an edge in the underlying graph may denote a "knows" relationship, a "can be infected by" relationship, a "lives close to" relationship, a "has common interests" relationship, etc. Further, each node has a state and a stochastic transition function that it evaluates at each time step to compute the value of its state at the next time step. The inputs to the transition function at a node are the state of the node itself and those of its neighbors in the underlying graph.

[1]Network Dynamics and Simulation Science Laboratory, Virginia Bioinformatics Institute and Virginia Tech, Blacksburg, VA 24061. Email: {cbarrett, mmarathe}@vbi.vt.edu.

[2]Department of Computer Science, University at Albany–SUNY, Albany, NY 12222. Email: {hunt, ravi, djr, res}@cs.albany.edu.

[3]Department of Computer Science, University of Missouri–Rolla, Rolla, MO 65409. Email: thakurk@umr.edu.

The following two examples present the motivation for the problems considered in this paper.

Example 1.1 *Consider the contact network for the population of a city, where the nodes represent individuals and edges represent contact between individuals; that is, there is an edge between two nodes if the corresponding individuals come into contact with each other during a certain period of time. We want to study the spread of epidemics in this population and how to best control it. A popular method of modeling the spread of epidemics is the SIR model [Eubank* et al.*, 2004], where the acronym stands for Susceptible-Infected-Recovered, the three possible states for each individual. Assume for simplicity that the total population is fixed; that is, there are no births or deaths. At any time, a susceptible node may become infected with a certain probability, depending on the number of infected neighbors. At any given time step t, let $s(t)$ denote the fraction of susceptible individuals, $i(t)$ denote the fraction of infected individuals and $r(t)$ denote the fraction of recovered individuals. In this context, it is of interest to study questions of the following form: Given positive numbers $0 < \alpha < 1$, $0 < \beta < 1$, $0 < p < 1$, and a positive integer τ, is the probability of the event "$i(t_0 + \tau) > \beta$" conditioned on the event "$i(t_0) < \alpha$" at least p? This question is an example of the* reachability *problem for the formal model considered in this paper.*

Example 1.2 *As a second example, consider the social influence network, where an edge between individuals denotes that they can exert a certain degree of influence on each other. Suppose a company wants to market a new product using word of mouth advertising [Kempe* et al.*, 2005]. The company's goal is to identify a suitable initial subset of individuals to whom free samples should be sent so that with probability at least p the number of individuals to whom the idea will propagate in t steps is at least N. (It is reasonable to model this propagation as a stochastic process since individuals may use different criteria to decide whether or not to participate in the propagation.) This example represents a more general form of the* predecessor existence *problem studied in this paper.*

Formally, we model dynamics of social networks using discrete dynamical systems. We refer to our model as a **Stochastic Synchronous Dynamical System** (SSyDS). Each SSyDS $\mathcal{S}$ over a domain $\mathbb{D}$ is specified as a pair $\mathcal{S} = (G, \mathcal{F})$. Here,

$G(V, E)$ is an undirected graph with n nodes, with each node having a state value from the domain $\mathbb{D}$. This graph represents the topological structure of the social network. The set $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ is a collection of stochastic interaction functions in the system. Here, f_i denotes the stochastic local transition function associated with node v_i, $1 \leq i \leq n$. A **configuration** of an SSyDS is an n-vector $(b_1, b_2, \ldots, b_n)$, where $b_i \in \mathbb{D}$ is the value of the state of node v_i ($1 \leq i \leq n$).

A single SSyDS transition from one configuration to another is obtained by updating the state of each node *synchronously* using the corresponding local transition function. For $1 \leq i \leq n$, the inputs to the function f_i are the state values of node v_i and those of the neighbors of v_i. For each combination of inputs to f_i and each element θ of $\mathbb{D}$, the function f_i specifies the probability that the next state value of v_i is θ. (For each combination of inputs, the sum of the probabilities assigned by f_i over the values $\theta \in \mathbb{D}$ must be 1.)

To further clarify the notion of stochastic local transition functions used here, consider a node v_i and let v_{i_1}, v_{i_2}, $\ldots$, v_{i_r} represent the neighbors of v_i in G. For any j and t, let s_j^t denote the state of node v_j at time t. The local transition function f_i at node v_i satisfies the following equation:

$$\Pr\{s_i^t = \theta \mid s_i^{t-1} = \theta', s_{i_1}^{t-1} = \theta_{i_1}^1, \ldots, s_{i_r}^{t-1} = \theta_{i_r}^1\} = f_i(\theta', \theta_{i_1}^1, \ldots, \theta_{i_r}^1, \theta). \quad (1)$$

The **generalized phase space** $\mathcal{P}_{\mathcal{S}}$ of an SSyDS $\mathcal{S}$ is a directed graph defined as follows. There is a node in $\mathcal{P}_{\mathcal{S}}$ for each configuration of $\mathcal{S}$. There is a directed edge from the node representing configuration $\mathcal{C}'$ to that representing configuration $\mathcal{C}$ if the probability that the system will reach $\mathcal{C}$ from $\mathcal{C}'$ in one step is (strictly) greater than zero. More generally, the directed edge from $\mathcal{C}'$ to $\mathcal{C}$ may be associated with a positive number p, $0 < p \leq 1$, representing the probability of the one-step transition. With the probability value for directed each edge, the generalized phase space represents the Markov chain for the dynamical system. Note that for a dynamical system over the Boolean domain with n nodes, the number of possible configurations is 2^n. Thus, the size of the Markov Chain is *exponential* in the size of the underlying dynamical system. For this reason, known results for Markov chains cannot be directly used to obtain the results obtained in this paper.

In the generalized phase space, when there is a directed edge from $\mathcal{C}'$ to $\mathcal{C}$, we say that $\mathcal{C}'$ is a **predecessor** of configuration $\mathcal{C}$. In general, a configuration in generalized phase space may have multiple predecessors.

The goal of an **analysis** problem is to determine whether a given SSyDS has a specified property. We consider two analysis problems in the context of such dynamical systems, namely **reachability** and **predecessor existence**. As mentioned earlier, a study of such analysis problems is useful in obtaining an understanding of the dynamics of social networks. Definitions of the problems considered in this paper are given below.

In the **reachability problem**, we are given a dynamical system $\mathcal{S}$, two configurations $\mathcal{I}$ and $\mathcal{B}$ and a probability value p; the question is whether $\mathcal{S}$ starting from $\mathcal{I}$ can reach $\mathcal{B}$ with a probability of at least p. This is an abstract and general version of the problem mentioned in Example 1.1. We use REACHABILITY to denote this problem. We also study a variant of this problem, called the t-REACHABILITY problem, where an integer t is also specified as part of the problem instance and the goal is to determine whether $\mathcal{S}$ starting from $\mathcal{I}$ can reach $\mathcal{B}$ in at most t steps with a probability of at least p.

In the **predecessor existence problem** (denoted by PRE), we are given a dynamical system $\mathcal{S}$, a configuration C and a probability value p; the question is whether there is a configuration C' such that the system starting from C' can reach C in one step with a probability of at least p. This is an abstract version of the problem discussed in Example 1.2 above. While Example 1.2 addresses t-step predecessors for some $t \geq 1$, the PRE problem is concerned with finding immediate (i.e., 1-step) predecessors.

Sequential Dynamical Systems (SDSs) where nodes update their states in a given *sequential* order (rather than synchronously) have also been considered in the literature (see for example [Barret *et al.*, 2003a]). A stochastic version of SDSs (denoted by SSDSs) can be defined in a manner similar that of SSyDSs. *The results presented in this paper are applicable to SSDSs as well.*

2 Summary of Results

Reachability in SSyDSs and SSDSs

Two versions of the reachability problem (denoted by REACHABILITY and t-REACHABILITY) for SSyDSs were defined in the previous section. We show that both the versions are hard for the complexity class[4] **RSPACE**(n). The result is proven by showing how a stochastic dynamical system can efficiently simulate a linear space bounded probabilistic Turing machine. Moreover, the hardness results hold even when the underlying graph is a simple path. By a minor modification to this proof, the hardness result can also be shown to hold for SSDSs (where nodes update their states sequentially). These results point out that, unless the complexity classes **P** and **RSPACE**(n) coincide, there is no efficient way of predicting the behavior of a stochastic dynamical system.

In contrast to stochastic dynamical systems, where reachability problems are **RSPACE**(n)-hard, the corresponding reachability problems for deterministic discrete dynamical systems are complete for **DSPACE**(n) [Barret *et al.*, 2003a]. It is known that **RSPACE**$(n) \subseteq$ **DSPACE**$(n^{3/2})$ [Saks and Zhou, 1995]; however, it is not known whether the exponent of n can be reduced to 1. Thus, under reasonable complexity theoretic assumptions, reachability problems for stochastic systems are harder than the corresponding problems for deterministic systems.

Predecessor Existence in SSyDSs

We show that the predecessor existence (abbreviated as PRE) problem for SSyDSs over any finite domain can be solved in polynomial time when the following conditions hold: (i) the treewidth[5] and maximum node degree of the underlying

[4]For definitions of the complexity classes used in this paper, we refer the reader to [Papadimitriou, 1994].

[5]For definitions of treewidth and related terms, we refer the reader to [Bodlaender, 1997].

graph are both bounded and (ii) the number of distinct probability values used in specifying the probability distributions for the local transition functions is bounded. The algorithm is based on dynamic programming. This result can also be extended to SSDSs.

It should be noted that the PRE problem remains **NP**-complete for deterministic dynamical systems, when either the treewidth or the maximum node degree is unbounded [Barrett *et al.*, 2003b]. As a consequence, the problem remains computationally intractable for such stochastic dynamical systems as well.

Our results for reachability and predecessor existence problems also initiate the study of stochastic discrete dynamical systems on graphs of bounded treewidth. It is known that 1D-CA with radius R can be equivalently viewed as synchronous dynamical systems on graphs with treewidth $O(R)$ [Barrett *et al.*, 2003b]. Thus, SSyDSs on treewidth bounded graphs can be viewed as *generalized* stochastic CA.

Applications to Other Computational Models

SSyDSs can be viewed as stochastic analogs of systolic networks [Kung, 1982]. Since SSyDSs are closely related to stochastic CA [Wolfram, 1986], they also serve as a formal model for studying problems in the context of multiagent systems [Wooldridge, 2002]. In addition, SSyDSs are also related to other well known computational models including discrete *recurrent Hopfield networks* [Orponen, 2000] which are used in machine learning and pattern recognition and *concurrent communicating finite state machines* [Harel *et al.*, 2002] which are used to model and verify distributed systems. The hardness results for the reachability problems for SSyDSs immediately imply analogous results for each of the above models.

3 Complexity of Reachability for SSyDSs and SSDSs

In this section, we show that the reachability problems for SSyDSs and SSDSs are hard for the complexity class **RSPACE**(n). Our proof also shows that the problem remains computationally intractable even for *simple* SSyDSs and SSDSs (e.g. systems in which the underlying graph is a simple path).

As mentioned earlier, we establish this complexity result by showing that a given linear space bounded probabilistic Turing Machine (TM) can be simulated by an appropriate SSyDS. We recall a few definitions related to probabilistic TMs.

A probabilistic Turing machine [Papadimitriou, 1994] M is a TM consisting of a finite control and a read-write tape which initially contains the input string. The finite control has one accepting state, one rejecting state and a collection of coin tossing states. A configuration of the Turing machine specifies the state of the finite control, description of the contents of the tape and the position of the head on the tape. A configuration is accepting, rejecting or coin-tossing if the state of the finite control is accepting, rejecting or coin tossing respectively. The transition relation of the Turing machine is such that from any coin tossing configuration, there are exactly two possible next moves, each with probability $1/2$. Once the machine reaches an accepting or rejecting configuration, it halts.

Given a probabilistic $O(n)$-space bounded Turing machine M which always halts after $2^{O(n)}$ moves and input string x, we say that M accepts x if the probability of the event that M reaches an accepting configuration starting from the initial configuration on input x is at least $1/2$. One way to view the computation of M on x is via a proof tree: the leaf nodes are labeled as either accepting or rejecting configurations, the root is the initial configuration and each internal node has exactly two children with the labels on the edges being $1/2$, denoting the probability of transition to that configuration. Thus the probability of reaching a leaf node (or a terminating configuration) l, denoted by $Pr(l)$, is the product of weights of the edges on the unique path from the root to the leaf. The probability of accepting an input string is $\sum_{l \in A} Pr(l)$, where A is the set of leaves marked as accepting.

Theorem 3.1 *There exists a constant μ such that the t-*REACHABILITY *and* REACHABILITY *problems for SSyDSs and SSDSs are **RSPACE**(n)-hard, even when the following restrictions hold: (i) The underlying graph is a simple path (and thus has pathwidth and treewidth of 1); in particular, the degree of each node is at most two. (ii) The size of the domain of state values for each node is at most μ. (iii) The number of distinct local transition functions is at most three.*

Proof sketch: We sketch the proof for the t-REACHABILITY problem SSyDSs. The proof for REACHABILITY is similar.

Let $M = (Q, \Sigma, \Sigma', q_0, q_f, F)$ denote a linear space bounded probabilistic Turing machine where Q is the (finite) set of states, Σ is the tape alphabet, $\Sigma' \subset \Sigma$ is the input alphabet, $q_0 \in Q$ is the initial state, $q_f \in Q$ is the final accepting state and F is the transition relation: given the current state and the current symbol scanned by the (read-write) head, F specifies the next state, the symbol to be written on the cell scanned by the head and the direction of head movement (left or right by one tape cell or stay on the same cell). The relation F specifies two moves, each with probability 1/2. Let $x = a_1 a_2 \ldots a_n$ be the input string given to M with $a_1 = \$$ and $a_n = ©$ being the end markers. A configuration or instantaneous description (ID) of M consists of the current state, the contents of the tape cells and the position of the head. The machine starts at q_0 with its head on the tape cell containing $a_1 = \$$. The ID at time zero is $\mathcal{I} = \langle (q_0, a_1), a_2, \ldots, a_n \rangle$. We may assume without loss of generality that if M accepts x, it replaces all the symbols on the tape cells between the end markers with the symbol $\not b$, moves the head to the cell containing \$, and halts in state q_f. Thus, the final ID is $\mathcal{B} = \langle (q_f, \$), \not b, \ldots, \not b, © \rangle$. The ID of M at time τ will be denoted by ID(τ).

Given M and input string x, we create an SSyDS $\mathcal{S}_{M_x} = (G, \mathcal{F})$ whose set of configurations include those of M. $\mathcal{S}_{M_x}$ is constructed so that for any $t \geq 0$, if M starting from $\mathcal{I}$ reaches a configuration $\mathcal{C}$ in t steps with probability p, then $\mathcal{S}_{M_x}$ starting from $\mathcal{I}$ reaches configuration $\mathcal{C}$ in $2t$ steps with probability p.

The underlying graph $G(V, E)$ of $\mathcal{S}_{M_x}$ is a simple path on n nodes, where $n = |x|$. Node v_i corresponds to the i^{th} tape cell, $1 \leq i \leq n$. Node v_i is adjacent only to nodes v_{i-1}

and v_{i+1}, with the exceptions that node v_1 is adjacent only to node v_2 and node v_n is adjacent only to node v_{n-1}. The state of each node v_i takes a value from the domain $\Sigma \cup (Q \times \Sigma) \cup (Q \times \Sigma \times \{0, 1\})$.

For any $\tau \geq 0$, a step of M that transforms ID(τ) into ID($\tau + 1$) probabilistically, can be captured by the collection of local probabilistic transition functions $\mathcal{F} = \langle f_1, f_2, \ldots, f_n \rangle$ where f_i is the function at node v_i, $1 \leq i \leq n$, as follows. The SSyDS simulates each step of M in two steps. Let $ID(\tau) = \langle c_1, \ldots c_{j-1}, (q, c_j), c_{j+1}, \ldots c_n \rangle$. Here, (q, c_j) denotes that the current state is q, the content of tape cell j is c_j and the head is currently scanning cell j. The node corresponding to cell j is v_j. In the first step, all nodes except v_j make a deterministic move; they copy their contents and do not change their state. Node v_j does a coin toss and modifies (q, c_j) to (q, c_j, σ) where σ denotes the outcome of coin toss (which is either 0 or 1). Note that $ID(\tau + 1)$ is identical to $ID(\tau)$, except possibly for c_{j-1}, (q, c_j) and c_{j+1}. Let these three values be d_{j-1}, d_j and d_{j+1} with probability $1/2$ and a_{j-1}, a_j, a_{j+1} with probability $1/2$. Note that the a_j, d_j values also encode the head location. Correspondingly, in the next step, all nodes of the SSyDS except v_{j-1}, v_j and v_{j+1} again copy their state; v_j goes to d_j or a_j, v_{j-1} goes to d_{j-1} or a_{j-1} and v_{j+1} goes to d_{j+1} or a_{j+1}, depending on the coin toss value σ. These moves again are deterministic. Thus, the probability of reaching either of the configuration is $1/2$ over the two steps.

It can be seen that $\mathcal{S}_{M_x}$ reaches configuration $\mathcal{B}$ with probability at least $1/2$ iff M accepts x with probability at least $1/2$. The size of the domain of $\mathcal{S}_{M_x}$ is a constant that depends only on Σ and Q. ■

It is also possible to formulate reachability problems where a family of final configurations is specified and the question is whether the system can reach any of these configurations with a given probability. An example of such a family consists of configurations in which at least k of the nodes have the state value 1. In the context of studying the spread of epidemics in social networks, such a family of configurations may represent situations where a large section of the population is infected. By slightly modifying the proof of the above theorem, it is possible to show that the corresponding reachability problem remains **RSPACE**(n)-hard.

4 Results on Predecessor Existence

This section develops a polynomial time algorithm for the PRE problem for restricted forms of stochastic dynamical systems. The main result of this section is the following.

Theorem 4.1 *Consider the class of SSyDSs over the Boolean domain satisfying the following conditions: (a) The degree of each node in the underlying graph is bounded. (b) The treewidth of the underlying graph is bounded. (c) The number of distinct probability values used in specifying all the stochastic node functions is bounded. The* PRE *problem for such SSyDSs can be solved in polynomial time.*

Our proof of the above theorem is based on a known result for a restricted class of *deterministic* SyDSs. In a deterministic SyDS, each local transition function is deterministic; that is, for each combination of inputs to the function, the function produces only one output which becomes the next state of the corresponding node. We use SyDS to denote a deterministic synchronous dynamical system. Reference [Barrett *et al.*, 2003b] presents a polynomial time algorithm for the PRE problem for SyDSs over the Boolean domain, where the underlying graph is treewidth bounded and each node computes a symmetric[6] function. Since our algorithm relies on the approach used in [Barrett *et al.*, 2003b], we begin with a sketch of the algorithm from that reference.

Theorem 4.2 *[Barrett* et al.*, 2003b] The* PRE *problem for SyDSs over the Boolean domain where the underlying graph is treewidth bounded and each node computes a symmetric function can be solved in polynomial time.*

Proof sketch: Let $\mathcal{S}$ be the given SyDS whose underlying graph $G(V, E)$ has a treewidth of k. It is well known that a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ of G can be constructed in time that is a polynomial in the size of G. Moreover, this can be done so that T is a binary tree; that is, each node of T has at most two children [Bodlaender, 1997].

For a given node i of the tree decomposition, the SyDS nodes in X_i are referred to as **explicit nodes** of i. If a given explicit node of i is also an explicit node of the parent of i, that node is referred to as an **inherited node** of i; and if it does not occur in the parent of i, it is referred to as an **originating node** of i. The set of all explicit nodes occurring in the subtree of T rooted at i that are not explicit nodes of i are referred to as **hidden nodes** of i. Without loss of generality, it can be assumed that the number of nodes of T with fewer than two children is at most n, the number of nodes in G.

Let $\mathcal{C}$ be the configuration specified in the given instance of the PRE problem for $\mathcal{S}$. Consider a given node i of the tree decomposition. Suppose α is a given assignment of state values to the explicit nodes of i and β is a given assignment of state values to the hidden nodes of i. The combined assignment $\alpha \cup \beta$ is said to be **viable** for i if for every hidden node w of i, the evaluation of the local transition function f_w gives the value $\mathcal{C}(w)$, using the value $\beta(w)$ for w and the value $(\alpha \cup \beta)(u)$ for every neighbor u of w. The combined assignment $\alpha \cup \beta$ is said to be **strongly viable** for i if the above condition holds for every node w that is either a hidden node or an originating node of i.

For a given node i of the tree decomposition, and a given assignment β to the states of the hidden nodes of i, define a function $h_\beta : X_i \longrightarrow \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers, as follows. For $y \in X_i$,

> $h_\beta(y)$ is the number of hidden nodes w of i such that $\{w, y\} \in E$ and $\beta(w) = 1$.

The value $h_\beta(y)$ is the number of hidden nodes w of i whose old value is an input parameter to the computation of the new value of y, and $\beta(w) = 1$.

For a given node i of the tree decomposition, and a given assignment α to the states of the explicit nodes of i, define the set H_α to be the set of functions h from X_i to $\mathbb{N}$ such that there exists an assignment β to the states of the hidden nodes

[6]A Boolean function is **symmetric** if its value depends only on the number of inputs which are 1.

of i satisfying the following two conditions: $\alpha \cup \beta$ is viable for i and h is h_β.

Let d be the maximum node degree of G. For any given β and $y \in X_i$, the maximum possible value of $h_\beta(y)$ is d. The maximum possible value of $|X_i|$ is $k+1$ (where k is the treewidth). So, function h_β can be represented as a table with at most $k+1$ entries, where each entry is an integer value in the range 0 through d. Hence, an upper bound on $|H_\alpha|$ is $(d+1)^{k+1}$.

The PRE problem for $\mathcal{S}$ can be solved by using bottom-up dynamic programming on the decomposition tree. For each node i of T, a table with an entry for each assignment α to the states of the explicit nodes of i is computed. The value of the entry for each such assignment α is the set H_α. Let J_i refer to the entire table for node i. Since the treewidth k is a constant, the size of the table for each node of the decomposition tree is a polynomial in n, the number of nodes of the underlying graph $G(V, E)$. Using these facts, it can be shown that the bottom-up computation of the tables for all the nodes can be carried out in polynomial time. ■

To obtain the polynomial time algorithm alluded to in the statement Theorem 4.1, we need to extend the above dynamic programming algorithm to a more general class of symmetric functions, namely r-symmetric functions. A Boolean function is r-**symmetric** if its inputs can be partitioned into at most r classes such that the value of the function depends only on the number of 1's in each class. Note that every symmetric function is 1-symmetric. Also, if the maximum node degree of the underlying graph of a SyDS is Δ, then each node function is $(\Delta+1)$-symmetric. The following lemma shows that the result of Theorem 4.2 can be extended to the case where each node computes an r-symmetric function. The corresponding algorithm is obtained by modifying the dynamic programming outline given in the proof of Theorem 4.2. (The details are omitted because of space reasons.)

Lemma 4.3 *The* PRE *problem for SyDSs over the Boolean domain where the underlying graph is treewidth bounded and each node computes an r-symmetric function for some $r \geq 1$ can be solved in polynomial time.* ■

We can now state the main idea behind the proof of Theorem 4.1: the PRE problem for SSyDSs satisfying the conditions of Theorem 4.1 can be efficiently reduced to the PRE problem for *deterministic* SyDSs over the Boolean domain where each local transition function is r-symmetric (for some $r \geq 1$) and the underlying graph has bounded treewidth. Since the latter problem can be solved in polynomial time (Lemma 4.3), Theorem 4.1 would follow. We now sketch this reduction.

Let $\mathcal{S}$ be an SSyDS satisfying the conditions of Theorem 4.1. Let $\mathcal{C}$ denote the given configuration for which we need to determine whether there is a predecessor with probability at least p. Let k and Δ (fixed values) denote the treewidth and the maximum node degree of the underlying graph $G(V, E)$, with n denoting $|V|$. Consider any node v_i, and let Γ_i (also a fixed value) denote the total number of probability values used to specify the stochastic local transition function f_i at v_i. (Note that if the distinct probability values used to specify f_i are $\rho_1, \rho_2, \ldots, \rho_t$, then Γ_i represents the number of distinct values in the collection $\{\rho_1, 1-\rho_1, \rho_2, 1-\rho_2, \ldots, \rho_t, 1-\rho_t\}$.)

We now show how to construct a *deterministic* SyDS $\mathcal{S}_1$ over the Boolean domain from the stochastic SyDS $\mathcal{S}$. The underlying graph $G_1(V_1, E_1)$ of $\mathcal{S}_1$ is constructed as follows. To begin with, G_1 contains a copy of all the nodes and edges of G. Next, for each node v_i of G, $1 \leq i \leq n$, we create $q = \Gamma_i$ additional nodes. Let $v_i^1, v_i^2, \ldots, v_i^q$, denote these additional nodes, called the **auxiliary nodes** associated with v_i. (Thus, each auxiliary node of v_i is associated with one probability value from the specification of the stochastic local transition function f_i in $\mathcal{S}$.) Each of the nodes v_i^j, $1 \leq j \leq q$, is adjacent to v_i; further, if v_i is adjacent to $v_{i_1}, v_{i_2}, \ldots, v_{i_\ell}$ in G, then each of the nodes v_i^j, $1 \leq j \leq q$, is adjacent to v_{i_1}, $v_{i_2}, \ldots, v_{i_\ell}$ in G_1. Finally, G_1 has one more node, denoted by X. There is an edge between X and *each* auxiliary node created above.

The local transition functions for the nodes of G_1 are chosen as follows. For each node v_i, $1 \leq i \leq n$, the local transition function g_i is the constant function which outputs 1 for every input. For each node v_i, the local transition functions for the auxiliary nodes of v_i are chosen as follows. Let $\mathcal{C}(v_i)$ denote the state value of node v_i in the specified (final) configuration for $\mathcal{S}$. For each j, $1 \leq j \leq q$, consider the auxiliary node v_i^j, and let ρ_j denote the probability associated with v_i^j. Let N_i^j denote the set of all neighbors of v_i^j, *except* X. The (deterministic) local transition function g_i^j at v_i^j outputs 1 if and only if one of the following conditions holds.

(a) The initial value of v_i^j is 1, and for the input combination formed by the values assigned to the nodes in N_i^j and the transition of v_i to $\mathcal{C}(v_i)$, the probability assigned by f_i equals ρ_j.

(b) The initial value of v_i^j is 0, and for the input combination formed by the values assigned to the nodes in N_i^j and the transition of v_i to $\mathcal{C}(v_i)$, the probability assigned by f_i *does not equal* ρ_j.

The local transition function g_X at node X is defined as follows. Let N_X denote the set of neighbors of X. (By our construction, N_X is the set of all auxiliary nodes.) For any $w \in N_x$, define the function $\eta(w)$ as follows: If the state value of w is 0, then $\eta(w) = 1$; otherwise, $\eta(w)$ is the probability value associated with the auxiliary node w. The value of the function g_X is equal to 1 if and only if $\Pi_{w \in N_x} \eta(w) \geq p$, where p is the probability threshold specified as part of the PRE problem instance for the SSyDS $\mathcal{S}$.

The final configuration $\mathcal{C}_1$ for $\mathcal{S}_1$ has the value 1 for all the nodes of G_1. This completes the reduction. It can be seen that the reduction can be carried out in polynomial time. The following lemma points out some properties of the above construction.

Lemma 4.4 *(a) The graph G_1 is treewidth bounded. (b) Each of the local transition functions of $\mathcal{S}_1$ is r-symmetric for some r. (c) For the stochastic SyDS $\mathcal{S}$ and configuration $\mathcal{C}$, there is a predecessor with probability at least p if and only if there is a predecessor for the configuration $\mathcal{C}_1$ for the deterministic SyDS $\mathcal{S}_1$.* ■

Theorem 4.1 follows from Lemmas 4.3 and 4.4.

It can also be shown that all the polynomial time results mentioned above hold when the underlying domain is finite (i.e., of fixed size) instead of being Boolean.

5 Related Work

Many social scientists have studied stochastic transition functions similar to the ones considered in this paper; see for example [Axelrod, 1994; Epstein and Axtell, 1996; Macy and Willer, 2002]. Computational aspects of CA have been studied by a number of researchers; see for example [Wolfram, 1986]. Much of this work addresses decidability of properties for infinite CA. A stochastic version of 1-dimensional CA has been studied from a formal language perspective in [Mahajan, 1992]. A comprehensive survey by Sarkar [Sarkar, 2000] provides additional information regarding other known results for CA. Questions concerning the existence of fixed points and garden of Eden configurations (i.e., configurations which don't have predecessors) in deterministic SDSs are addressed in [Barrett *et al.*, 2003b; Tosic and Agha, 2005].

The predecessor existence problem has been studied in the context of CA [Sutner, 1995; Green 1987] and other computational models [Orponen, 2000]. The problem also arises naturally in other applications such as reverse engineering finite discrete dynamical systems from time series data, testing liveness properties of certain network protocols and detecting unreachable states in distributed systems; see [Barrett *et al.*, 2006] and the references cited therein.

6 Future Work

There are several directions for further work. First, it will be useful to study the complexity of reachability problems when there are additional restrictions on the stochastic functions computed by the nodes. Second, it is of interest to extend the results for the immediate predecessor problem to a more general version, namely finding t-step predecessors, for fixed $t \geq 1$. A different research direction is to formulate and study other problems that arise in the context of social network dynamics as problems on stochastic dynamical systems.

Acknowledgments: We thank the referees for providing helpful suggestions.

References

[Axelrod, 1994] R. Axelrod, *The Complexity of Cooperation*, Princeton University Press. Princeton, NJ, 1994.

[Barret *et al.*, 2003a] C. Barrett, H. Hunt III, M. Marathe, S. Ravi, D. Rosenkrantz and R. Stearns, "Reachability Problems for Sequential Dynamical Systems with Threshold Functions", *Theoretical Computer Science*, Vol. 295, Issues 1-3, Feb. 2003, pp. 41–64.

[Barrett *et al.*, 2003b] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz and R. E. Stearns, "Predecessor and Permutation Existence Problems for Sequential Dynamical Systems", *Proceedings of the International Conference on Discrete Models for Complex Systems* (DM-CS 2003), Lyon, France, June 2003.

[Barrett *et al.*, 2006] C. Barrett, S. Eubank and M. V. Marathe, "Modeling and Simulation of Large-Scale Biological, Information and Socio-Technical Systems: An Interaction Based Approach", in *Interactive Computation: The New Paradigm*, edited by D. Goldin, S. Smolka and P. Wegner, Springer 2006.

[Bodlaender, 1997] H. Bodlaender, "Treewidth: Algorithmic Techniques and Results", *Proceedings of the 22nd Symposium on Mathematical Foundations of Computer Science*, 1997, pp. 29–36.

[Epstein and Axtell, 1996] J. Epstein and R. Axtell, *Growing Artificial Societies: Social Science from the Bottom Up*, Brookings/MIT Press, 1996.

[Eubank *et al.*, 2004] S. Eubank, H. Guclu, V. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai and N. Wang, "Monitoring and Mitigating SmallPox Epidemics: Strategies Drawn from a Census Data Instantiated Virtual City", *Nature*, Vol. 429, May 2004, pp. 180–184.

[Green 1987] F. Green, "NP-Complete Problems in Cellular Automata", *Complex Systems*, Vol. 1, No. 3, 1987, pp. 453–474.

[Harel *et al.*, 2002] D. Harel, O. Kupferman, and M. Y. Vardi, "On the complexity of verifying concurrent transition systems", *Information and Computation*, Vol. 173, 2002, pp. 143–161.

[Kempe *et al.*, 2005] D. Kempe, J. Kleinberg and E. Tardos, "Influential Nodes in a Diffusion Model for Social Networks", *Proc. 32nd International Colloquium on Automata, Languages and Programming* (ICALP), 2005.

[Kung, 1982] H. Kung, "Why Systolic Architectures", *IEEE Computers*, 15(1), 1982, pp. 37–42.

[Macy and Willer, 2002] M. W. Macy and R. Willer, "From Factors to Actors: Computational Sociology and Agent-Based Modeling", *Annual Review of Sociology*, Vol. 28, 2002, pp. 143–166.

[Mahajan, 1992] M. Mahajan, "Studies in Language Classes Defined by Different Types of Time-Varying Cellular Automata", Ph.D. Thesis, Indian Institute of Technology, Madras, 1992.

[Nichitiu and Remila, 1998] C. Nichitiu and E. Remila, "Simulations of Graph Automata", *Proc. MFCS'98 Satellite Workshop on Cellular Automata*, Brno, Czech Republic, Aug. 1998.

[Orponen, 2000] P. Orponen, "An Overview of the Computational Power of Recurrent Neural Networks", *Proc. 9th Finnish AI Conference STeP 2000 - Millennium of AI*, H. Hyotyniemi, ed. Espoo, Finland, 2000, pp. 89–96.

[Papadimitriou, 1994] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.

[Saks and Zhou, 1995] M. Saks and S. Zhou, "RSPACE(n) is contained in DSPACE($n^{3/2}$)", *Proc. IEEE FOCS 1995*, pp. 344–353.

[Sarkar, 2000] P. Sarkar, "A Brief History of Cellular Automata", *ACM Computing Surveys*, Vol. 32, No. 1, 2000, pp. 80–107.

[Sutner, 1995] K. Sutner, "On the Computational Complexity of Finite Cellular Automata", *Journal of Computer and System Sciences*, Vol. 50, No. 1, Feb. 1995, pp. 87–97.

[Tosic and Agha, 2005] P. Tosic and G. Agha, "On Computational Complexity of Counting Fixed Points in Symmetric Boolean Graph Automata", *Proc. UC'05 - Fourth International Conference on Unconventional Computation*, Sevilla, Spain, October 2005, Springer-Verlag LNCS series, Vol. 3699, pp. 191-205.

[Wolfram, 1986] S. Wolfram (Editor), *Theory and Applications of Cellular Automata*, World Scientific, 1987.

[Wooldridge, 2002] M. Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons, Chichester, England, 2002.