Complex Adaptive Systems: Energy, Information and Intelligence: Papers from the 2011 AAAI Fall Symposium (FS-11-03)

Inhibiting the Diffusion of Contagions in Bi-Threshold Systems: Analytical and Experimental Results

Chris J. Kuhlman, V. S. Anil Kumar, Madhav Marathe, Samarth Swarup, Gaurav Tuli
Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24060.
{*ckuhlman, akumar, mmarathe, swarup, gtuli*}@*vbi.vt.edu*

S. S. Ravi, Daniel J. Rosenkrantz
Department of Computer Science,
University at Albany, State University of New York, Albany, NY 12222.
{*ravi, djr*}@*cs.albany.edu*

Abstract

We present a bi-threshold model of complex contagion in networks. In this model a node in a network can be in one of two states at any time step, and changes state if enough of its neighbors are in the opposite state, as determined by "up-threshold" and "down-threshold" parameters. This dynamical process models several types of social contagion processes, such as public health concerns and the spread of games on online networks. Motivated by recent literature calling for the investigation of peer pressure to reduce obesity, which can be viewed as a control problem of population dynamics, we focus on the computational complexity of finding critical sets of nodes, which are nodes that we choose to freeze in state 0 (a desirable state) in order to inhibit the spread of an undesirable state 1 in the network. We define a minimum-cost critical set problem and show that it is NP-complete for bi-threshold systems. We show that several versions of the problem can be approximated to within a factor of $O(\log n)$, where n is the number of nodes in the network. Using the ideas behind these approximations, we devise a heuristic, called the Maximum Contributor Heuristic (MCH), which can be used even when the diffusion model is probabilistic. We perform simulations with well-known networks from the literature and show that MCH outperforms the High Degree Heuristic by several orders of magnitude.

Keywords: Contagion, Diffusion, Bi-threshold systems.

1 Introduction

Problems related to contagions on social networks are being increasingly studied. We define a **contagion** as any entity, such as trust, a fad, or news that can spread through a population. Contagion processes studied in the literature include viral marketing and effects of referrals, formation of online communities, spread of online games, and trust propagation, among others (Domingos and Richardson 2001; Richardson and Domingos 2002; Guha et al. 2004; Leskovec, Adamic, and Huberman 2007; Habiba et al. 2008). All of these contagions, at their core, involve the spread of information, be it verbal as in word-of-mouth marketing or nonverbal as in adopting a fad because one observes many people doing so.

One mechanism used to describe the propagation of social contagions is based on a **threshold**. A **simple contagion** is a contagion in which an agent requires only one neighbor to possess it in order for the agent itself to acquire it. In this case, the threshold t is 1. For example, the spread of news through blogspace has been modeled as a simple contagion (Gruhl et al. 2004). A **complex contagion** requires two or more neighbors to possess it before an agent will acquire it; this corresponds to a threshold value $t \geq 2$. Examples include the adoption of risky new technologies and participation in social movements (Granovetter 1978; Centola and Macy 2007). See (Watts 2002) for a discussion of how threshold models are often utilized, even when more involved decision-making processes are available. Also, most of these studies focus on **ratcheted** (or **progressive**) 2-state systems, where an agent can be in state 0 or 1, but the only permissible state transition is from 0 to 1. An agent may not transition from state 1 to 0.

A threshold, in its most basic form, is a criterion describing the amount of local influence a node requires to change state: if a minimum number t of one's friends or acquaintances behave in a particular way, then one will adopt the same convention. Interestingly, whether one smokes, is obese, or is a heavy drinker—the three behaviors contributing most to chronic health conditions (Sturm 2002)—are all heavily influenced by peer behavior (Borsari and Carey 2001; Hoffman et al. 2006; O'Loughlin et al. 2009). These behaviors are also costly; e.g., the annual health care cost of obesity is $147 billion (Finkelstein et al. 2009). Finally, and important for this work, these behaviors can be repeatedly started and stopped. In the case of obesity, for example, it is well-known that dieters engage in cycles of dieting and non-dieting over time, a phenomenon referred to as "yo-yo dieting" (Atkinson et al. 1994). Thus, ratcheted models of behavior are inadequate to investigate these phenomena. A model that describes back-and-forth transitions between two states is required. Further, investigations into the feasibility of using peer influence to dissuade undesirable behaviors is called for in (Lopez 2008) in connection with obesity.

Reducing levels of chronic illnesses represent special problems because they require modifying people's ingrained habits such as over-eating. Hence, the difficulty in halting undesirable behavior may be greater than that of starting the behavior; i.e., a start threshold may be less than a threshold to stop. Particularly relevant for addictive behavior is the notion of **self-efficacy**, stemming from social cognitive science, which describes an addict's perceived ability to change her behavior. The greater a person's self-efficacy, the more empowered she feels to conquer an addiction. Furthermore two of the four factors that produce self-efficacy are peer-driven: observations of others and verbal persuasion. These ideas are summarized in (Rosenstock, Strecher, and Becker 1988). The points are: (i) self-efficacy can be modeled as an internal threshold phenomenon and (ii) that it is driven significantly by peer influences.

Back-and-forth models are useful beyond health care issues. The spread of games like FarmVille on the Facebook social network may be considered a complex contagion. Many other social phenomena that can be modeled as threshold-based back-and-forth systems are given in (Schelling 1978; Bischi and Merlone 2009).

Our objectives are to understand bi-threshold dynamics and how to inhibit an undesirable behavior for a range of network structures. The bi-threshold model is defined by two threshold values at each node, where the up-threshold is the number of neighbors in state 1 required for an agent to change from state 0 to state 1, and the down-threshold is the number of neighbors in state 0 required for an agent to change from state 1 to state 0. The model is formally discussed in Section 2, where we also discuss the use of peer influence as a mechanism to reduce or halt undesirable behavior; investigations of this type are explicitly called for by Lopez (2008). Specifically, we seek a set of peers (called a "critical set," defined formally later) that will drive a population away from an undesirable behavior. Following related work in Section 3, we introduce in Section 4 a theoretical framework for computing a critical set, which we call "the minimum cost critical set problem." We show that the problem is NP-hard in general, and this necessitates the use of heuristic approaches. We offer one in Section 5, abbreviated MCH, that is suited to deterministic and stochastic contagion dynamics. To evaluate the behavior of bi-threshold systems, we ran extensive simulations on three social networks from the literature and used an aggressive approach to tax MCH, as described in Section 6. Deterministic and probabilistic simulation results for bi-threshold models and for using peer influence to inhibit undesirable behaviors are presented in Section 7. Conclusions are given in Section 8.

Summary of results. Our results are summarized as follows.

1. We show that the minimum cost critical set problem is NP-hard for both simple and complex contagions. For simple contagions, we show that the problem can be approximated to within a factor of $O(\log n)$, where n is the number of nodes in the network and that no asymptotic improvement in performance guarantee is possible unless P = NP. For complex contagions, we show that a restricted version of the problem (see Section 4.3 for details) can be approximated to within a factor of $O(\log n)$. Even for this restricted version, we show that no asymptotic improvement in performance guarantee is possible unless P = NP.
2. Based on the approach used in the above approximation algorithms, we present a heuristic, called Maximum Contributor Heuristic (MCH), for finding critical nodes to prevent nodes from switching to state 1. We show that MCH outperforms the High Degree Heuristic (i.e., setting highest degree nodes as critical) by multiple orders of magnitude.
3. Over a large parameter input space, we show that MCH will halt diffusion and that the required number of critical nodes is dependent on network structure, thresholds, and seeding in complicated ways.

As this is the first work of its kind (see Section 3, Related Work), we study fundamental behaviors to understand bi-threshold systems and control of system dynamics under various conditions. A detailed practical application of the basic bi-threshold model to adolescent smoking is discussed in a companion work (Kuhlman et al. 2011).

2 Model Description and Problem Statement

2.1 Definition of the Bi-threshold Model

We model the propagation of contagions over a social network using discrete dynamical systems (DDSs) (e.g. (Barrett et al. 2006)). Let $\mathbb{B}$ denote the Boolean domain $\{0,1\}$. A **Synchronous Dynamical System** (SyDS) $\mathcal{S}$ over $\mathbb{B}$ is specified as a pair $\mathcal{S} = (G, \mathcal{F})$, where

(a) $G(V, E)$, an undirected graph with n nodes, represents the underlying social network over which the contagion propagates, and

(b) $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ is a collection of functions in the system, with f_i denoting the **local transition function** associated with node v_i, $1 \le i \le n$.

Each node of G has a state value from $\mathbb{B}$. Each function f_i specifies the local interaction between node v_i and its neighbors in G. We use the convention that node is *not* a neighbor of itself. In this paper, function f_i at node v_i ($1 \le i \le n$) is a **bi-threshold function**, characterized by two non-negative integer values denoted by $t_{\mathrm{up}}(v_i)$ and $t_{\mathrm{down}}(v_i)$. A precise definition of the function f_i is as follows.

(a) If the state of v_i is 0, then f_i is 1 if at least $t_{\mathrm{up}}(v_i)$ of the neighbors of v_i are in state 1; otherwise, the value of f_i is 0;

(b) If the state of v_i is 1, then f_i is 0 if at least $t_{\mathrm{down}}(v_i)$ of the neighbors of v_i are in state 0; otherwise, the value of f_i is 1.

Thus, $t_{\mathrm{up}}(v_i)(t_{\mathrm{down}}(v_i))$ called the **up** (**down**) **threshold** of v_i, represents the minimum number of neighbors of v_i that must be in state 1 (0) for v_i to change from 0 to 1 (1 to 0). A SyDS in which each node has a bi-threshold transition function is called a **bi-threshold** SyDS, denoted by BT-SyDS.

A **configuration** $\mathcal{C}$ of a SyDS at any time is an n-vector $(s_1, s_2, \ldots, s_n)$, where $s_i \in \mathbb{B}$ is the state of v_i. A single SyDS transition from one configuration to another can be expressed by the following pseudocode.

for each node v_i **do in parallel**
  (i) Compute the value of f_i. Let s_i' denote this value.
  (ii) Update the state of v_i to s_i'.
end for

Thus, in a SyDS, nodes update their state *synchronously*. Other update disciplines (e.g. sequential updates) for DDSs have also been considered (Barrett et al. 2006).

Example: Consider the BT-SyDS whose underlying graph is shown in Figure 1. Suppose for each node v_i, the function f_i is the bi-threshold function with $t_{\rm up}(v_i) = t_{\rm down}(v_i) = 1$, $1 \leq i \leq 6$. In the initial configuration, v_1 is in state 1 and all other nodes are in state 0. During the first time step, the state of node v_1 changes to 0 (since v_1 has a neighbor whose state is 0) and that of v_2 changes to 1 (since v_2 has a neighbor whose state is 1); the other nodes remain in state 0. In the second time step, it can be verified that v_1 changes to 1, v_2 changes to 0, v_3 changes to 1, v_5 changes to 1 and the other two nodes remain in state 0. Continuing this process, it can be seen that from time step 2 onwards, the BT-SyDS cycles between the two configurations $(1, 0, 1, 0, 1, 0)$ and $(0, 1, 0, 1, 0, 1)$.

If a SyDS has a transition from configuration $\mathcal{C}_1$ to configuration $\mathcal{C}_2$, we say that $\mathcal{C}_2$ is the **successor** of $\mathcal{C}_1$ and that $\mathcal{C}_1$ is a **predecessor** of $\mathcal{C}_2$. A configuration $\mathcal{C}$ is called a **fixed point** if the successor of $\mathcal{C}$ is $\mathcal{C}$ itself.

Model extensions. As defined above, the SyDS model is *deterministic*. One can also define a *probabilistic* version by specifying two non-negative numbers $p_{\rm up}(v_i)$ and $p_{\rm down}(v_i)$ with each node v_i. In such a case, $p_{\rm up}(v_i)$ ($p_{\rm down}(v_i)$) denotes the probability that v_i changes from 0 to 1 (1 to 0) when the number of its neighbors in state 1 (0) is at least $t_{\rm up}(v_i)$ ($t_{\rm down}(v_i)$). Also, homophily (Axelrod 1997) and factors from social cognitive theory (Rosenstock, Strecher, and Becker 1988) can be incorporated.

2.2 Additional Definitions Related to the Model

For any SyDS $\mathcal{S}$, the **phase space** of $\mathcal{S}$ is a directed graph with one node for each possible configuration; there is a directed edge from the node representing configuration $\mathcal{C}$ to that representing configuration $\mathcal{C}'$ if and only if $\mathcal{C}'$ is the successor of $\mathcal{C}$. Since the domain is $\mathbb{B} = \{0,1\}$ and the underlying graph has n nodes, the number of nodes in the phase space is 2^n; thus, the size of the phase space is *exponential* in the size of the SyDS.

For any deterministic SyDS, each configuration has a unique successor. Thus, the outdegree of each node in the phase space is 1. Each fixed point of a SyDS $\mathcal{S}$ is a self loop in the phase space of $\mathcal{S}$.

A BT-SyDS in which $t_{\rm up}(v) = t_{\rm down}(v) = 1$ for each node v is called a **simple** BT-SyDS. If at least one of the threshold values in $\mathcal{S}$ is *greater than* 1, then $\mathcal{S}$ is referred to as a **complex** BT-SyDS.

In this paper, we also use a slightly relaxed notion of fixed points, namely *pseudo* fixed points, as defined below.

Definition 2.1 *Given a SyDS $\mathcal{S}$, a configuration $\mathcal{C}$ of $\mathcal{S}$ is a* ***pseudo fixed point*** *if in the sequence of configurations obtained from $\mathcal{C}$, no node changes from 0 to 1.*

Thus, starting from a pseudo fixed point, nodes may change from 1 to 0; however, no node may change from 0 to 1. From the above definition, it can be seen that each fixed point is also a pseudo fixed point.

One can also define a pseudo fixed point from which the only allowed change is from 0 to 1. In the applications that motivated the BT-SyDS model, it is more natural to forbid 0 to 1 transitions rather than 1 to 0 transitions.

2.3 Critical Set Problem for BT-SyDSs

Many different computational problems have been considered for DDSs (e.g. (Barrett et al. 2006)). Here, our focus is on a particular problem that deals with the conversion of a given configuration into a pseudo fixed point by modifying some nodes whose current state is 1. Here, "modifying" a node involves two things: (i) the state of the node is changed from 1 to 0 and (ii) the up threshold of the node is changed to a high value (e.g. degree of the node + 1) so that the node can never change to 1. We say that such a node is **frozen at 0**. The modified node is *not* removed from the network; it remains in the network to allow more nodes to change from 1 to 0 in subsequent time steps. The set of nodes chosen for modification is called the **critical set**. In social settings, frozen nodes are a form of peer influence: they encourage neighboring nodes to change to or remain in state 0. Only those nodes whose current state is 1 can be part of the critical set. In the social network context, modifying a node involves a cost. So, it is important to find a critical set of minimum cost. A precise formulation of the corresponding problem, called the **Minimum Cost Critical Set** (MCCS) problem, is as follows.

Minimum Cost Critical Set (MCCS)

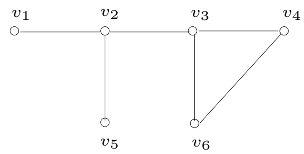
Instance: A BT-SyDS $\mathcal{S}$, a configuration $\mathcal{C}$ of $\mathcal{S}$, for each node $v \in V$, the cost $c(v) \geq 0$ for modifying v and a budget β on the modification cost.

Question: Is there a critical set B such that the cost of B is at most β and B modifies $\mathcal{C}$ into a pseudo fixed point?

We defined the MCCS problem using pseudo fixed points for two reasons. First, the cost of converting a given configuration to a pseudo fixed point is no more than that of converting it to a fixed point (since every fixed point is also a pseudo fixed point); in fact, one can construct examples where the conversion to a fixed point is much more expensive than the conversion to a pseudo fixed point. Second, once a bi-threshold system reaches a pseudo fixed point, it takes at most n additional transitions for the system to reach a fixed point. In other words, pseudo fixed points offer all the benefits of fixed points while incurring no additional cost.

3 Related Work

A number of references for simple and complex contagions, ratcheted threshold systems, and application domains were discussed in Section 1. Here we focus on three issues central to this paper: computational complexity results, bi-threshold systems, and mechanisms for inhibiting state transitions.



Initial Configuration:	(1, 0, 0, 0, 0, 0)
Configuration at time 1:	(0, 1, 0, 0, 0, 0)
Configuration at time 2:	(1, 0, 1, 0, 1, 0)
Configuration at time 3:	(0, 1, 0, 1, 0, 1)
Configuration at time 4:	(1, 0, 1, 0, 1, 0)

Note: For each node, the up and down threshold values are 1. Each configuration has the form $(s_1, s_2, s_3, s_4, s_5, s_6)$, where s_i is the state of node v_i, $1 \leq i \leq 6$. From time step 2 onwards, the system cycles between the two configurations $(1, 0, 1, 0, 1, 0)$ and $(0, 1, 0, 1, 0, 1)$.

Figure 1: An example of a bi-threshold synchronous dynamical system.

Most complexity results are for the spread maximization problem. The earliest results for complex contagions are provided in (Kempe, Kleinberg, and Tardos 2003), and enhancements have been made up through (Mossel and Roch 2007). See the latter for intermediate references and a description of the incremental improvements made by each. Maximizing influence in the voter model has been examined (Even-Dar and Shapira 2007). The only work addressing the complexity of thwarting diffusion is that of (Kuhlman et al. 2010a), where a critical set problem in the context of ratchet-up systems was investigated. The critical set problem is shown to be solvable in polynomial time for $t = 1$, whereas for $t \geq 2$, the problem is shown to be NP-hard. We know of no comparable work for bi-threshold systems.

There are a few models that permit transitions back and forth between two states. In the voter model each agent changes state (to 0 or 1) by assuming the state of one randomly selected neighbor; see (Even-Dar and Shapira 2007) and references therein. Thus, all neighbors of an agent are not utilized as in the bi-threshold model. In majority models (Dreyer and Roberts 2009), an agent switches to the state of the majority of its neighbors. In contrast, state change in the bi-threshold model may be induced by more or less than one-half of its neighbors, so it is a generalization of a majority model. Perhaps the model closest to the bi-threshold model is that of (Bischi and Merlone 2009) because it has up and down threshold behavior. However, the uniform mixing model assumes a completely connected population network, where every agent influences every other agent in the same fashion, and every agent has complete knowledge of the system. The model makes no distinction among agents. Our model accounts for local structure within the population, where in general, each agent is only influenced by a subset of the population. Among other differences, our agents may have heterogeneous, history-dependent up-thresholds and down-thresholds. We can also readily extend our work to generalized contagion models, such as that of (Dodds and Watts 2005), as we have done for ratchet-up systems.

The great majority of the work on thwarting diffusion is confined to simple contagions in ratchet-up systems. In (Chakrabarti et al. 2008), the maximum eigenvalue of the adjacency matrix is used to identify the node that causes the greatest decrease in the epidemic threshold. Reference (Eubank et al. 2006) provides an approximation for minimizing the number of agents that transition to state 1. Various methods for blocking in small networks have been studied (Borgatti 2006). Heuristics for stopping the diffusion of simple contagions on static and dynamic networks were investigated in (Habiba et al. 2008). Blocking of complex contagions in ratchet-up systems using heuristics was investigated in (Kuhlman et al. 2010a) and it was shown that a method that worked well for simple contagions (namely, setting the highest degree nodes critical (Habiba et al. 2008)) performs poorly for complex contagions. The complex contagion blocking scheme is only applicable to deterministic diffusion. The approach herein is applicable to both deterministic and stochastic diffusion.

4 Theoretical Results for the MCCS Problem

4.1 Preliminaries

In this section, we present complexity and approximability results for the minimum cost critical set (MCCS) problem for simple and complex contagions. For reasons of space, proofs of some results are omitted and only sketches are included for others. Detailed proofs appear in (Kuhlman et al. 2010b).

We assume throughout this section that the given BT-SyDS does not have any node whose up threshold is 0. The reason is that the 0 to 1 transitions of such nodes cannot be controlled by their neighbors. The following lemma is used throughout this section.

Lemma 4.1 *Let $\mathcal{S}$ be a BT-SyDS where the up threshold of each node is at least 1. Let $\mathcal{C}$ be a given configuration. The problem of determining whether $\mathcal{C}$ is a pseudo fixed point can be solved in polynomial time.*

Proof sketch: The idea is to run $\mathcal{S}$ starting from the given configuration $\mathcal{C}$ for n steps, where n is the number of nodes in the underlying graph of $\mathcal{S}$. Since the number of 1's in $\mathcal{C}$ that can change to 0 is at most n, if $\mathcal{C}$ is a pseudo fixed point, $\mathcal{S}$ will reach a fixed point in at most n transitions. Thus, $\mathcal{C}$ is not a pseudo fixed point if and only if some node changes from 0 to 1 during these n transitions. $\square$

4.2 Results for Simple BT-SyDSs

The results in this section rely on a structural property of pseudo fixed points created by freezing certain nodes of a simple BT-SyDS. Some terminology is needed to state this property. Consider a simple BT-SyDS $\mathcal{S}$ with underlying graph $G(V, E)$. Suppose $F \subseteq V$ is a nonempty subset frozen nodes. Let $\mathcal{C}$ be a configuration of $\mathcal{S}$ such that for each node $v \in F$, the state of v in $\mathcal{C}$ is 0. Each node $w \in V$ which is in state 1 in $\mathcal{C}$ is called a **normal 1-node**. Each node $w \in V - F$ which is in state 0 in $\mathcal{C}$ is called a **normal 0-node**. This following lemma provides a characterization of when $\mathcal{C}$ is a pseudo fixed point. A proof of this lemma appears in (Kuhlman et al. 2010b).

Lemma 4.2 *Suppose $\mathcal{S}$ is a simple BT-SyDS with underlying graph $G(V, E)$ and $F \subseteq V$ is nonempty subset of frozen nodes, Let $\mathcal{C}$ be a configuration of $\mathcal{S}$ such that for each node $v \in F$, the state of v in $\mathcal{C}$ is 0. Configuration $\mathcal{C}$ is a pseudo fixed point iff both of the following conditions hold: (i) No normal 0-node is adjacent to a normal 1-node. (ii) Each normal 1-node is adjacent to at least one node in F.* □

The following result shows that the MCCS problem is NP-complete for simple BT-SyDSs and that it cannot be approximated to within a factor $o(\log n)$.

Theorem 4.1 *(a) The* MCCS *problem is NP-complete for simple BT-SyDSs. (b) There is a constant $c' > 0$ such that the* MCCS *problem for simple BT-SyDSs with n nodes cannot be approximated to within a factor less than $c' \ln n$, unless P = NP.*

Proof sketch for Part (a): Using Lemma 4.1, it can be seen that the MCCS problem is in NP. To prove NP-hardness, we use a reduction from the Minimum Dominating Set (MDS) problem which is known to be NP-complete (Garey and Johnson 1979). Let an instance of the MDS problem be given by the undirected graph $G(V, E)$ and integer K. We construct an instance of the MCCS problem for a simple BT-SyDS $\mathcal{S}$ as follows. The underlying graph $G'(V', E')$ of $\mathcal{S}$ consists of a copy of G plus three new nodes denoted by x, y and z. There is an edge joining node x to one node, say v_1, of V. Two other edges $\{x, y\}$ and $\{y, z\}$ are also added. So, E' consists of all the edges of E plus the three new edges added. The up and down threshold values for all the nodes in V' are 1. In the chosen configuration $\mathcal{C}$ (which needs to be converted into a pseudo fixed point), node z is in state 0 and all the other nodes are in state 1. The modification cost for each node of G' is chosen as 1. This completes the construction of the MCCS instance, and it is easy to see that the construction can be done in polynomial time. It can be shown using Lemma 4.2 that the graph G has a dominating set of size at most K if and only if the resulting MCCS instance has a solution of cost at most $K + 1$.

Proof sketch for Part (b): We rely on the following result from (Raz and Safra 1997): There is a constant $c > 0$ such that the MDS problem cannot be approximated to within a factor less than $c \ln n$ for n node graphs, unless P = NP. Using that result in conjunction with the reduction presented in the proof of Part (a), we show that if there is an approximation algorithm for the MCCS problem with a performance guarantee less than $(c/4) \ln n$, then there is an approximation algorithm for the MDS problem with a performance guarantee less than $c \ln n$. □

Our next result shows that the MCCS problem can be approximated to within a factor $O(\log n)$ for simple BT-SyDSs. This result also relies on Lemma 4.2.

Theorem 4.2 *There is an approximation algorithm with a performance guarantee of $O(\log n)$ for simple BT-SyDSs with n nodes.*

Proof sketch: Our approximation algorithm chooses frozen nodes in two stages. Frozen nodes chosen in the first stage prevent nodes in state 0 from changing to 1. In the second stage, the chosen nodes ensure that nodes which are in state 1 cannot change to 0 during one transition and then back to 1 in a later transition. This second stage relies on a known $O(\log n)$ approximation algorithm for the Minimum Set Cover problem (Vazirani 2001). The performance guarantee of the algorithm is determined by the second stage. □

4.3 Hardness and Approximation Results for Complex BT-SyDSs

We show that the MCCS problem is computationally intractable for complex BT-SyDSs.

Theorem 4.3 *(i) The* MCCS *problem is NP-complete for complex BT-SyDSs in which the maximum threshold value is 2 and all nodes have the same modification cost. (ii) Unless P = NP, the* MCCS *problem for complex BT-SyDSs cannot be approximated to within $o(\log n)$, where n is the number of nodes in the given BT-SyDS.*

Proof sketch – Part (i): Proof of membership in NP again relies on Lemma 4.1. To prove NP-hardness, we use a reduction from the Minimum Vertex Cover (MVC) problem which is known to be NP-complete (Garey and Johnson 1979). Let the given instance I of the MVC problem consist of graph $G_1(V_1, E_1)$ and integer K. Let $V_1 = \{v_1, v_2, \ldots, v_n\}$ and $E_1 = \{e_1, e_2, \ldots, e_m\}$. We construct a BT-SyDS $\mathcal{S}$ as follows. The underlying graph $G(V, E)$ of $\mathcal{S}$ has one node p_i for each node v_i of G_1 ($1 \leq i \leq n$) and one node q_j for each edge e_j of G_1 ($1 \leq j \leq m$). If edge e_j joins nodes v_x and v_y in G_1, then node q_j is adjacent to nodes p_x and p_y in G. The up and down thresholds for each node p_i are 1. The up threshold for each node q_j is 2 and the down threshold is 1. The cost of modifying any node is chosen as 1. In the chosen configuration $\mathcal{C}$, each node p_i is set to 1 and each node q_j is set to 0. The upper bound on the cost of modification is set to K. This completes the construction of the MCCS instance. We note that in the resulting MCCS instance, each threshold value is at most 2 and all nodes have the same modification cost. Obviously, the construction can be carried out in polynomial time. It can be shown that G_1 has a vertex cover of size at most K if and only if the MCCS problem for $\mathcal{S}$ has a solution of cost at most K.

Proof sketch – Part (ii): We presented the above reduction from MVC to MCCS to show that MCCS remains NP-complete even for BT-SyDSs with a maximum threshold of 2. We can carry out a similar reduction from the Minimum Set Cover (MSC) problem. The idea is to treat the base set of the MSC instance similar to the edge set of the MVC instance and each subset of the MSC instance similar to a node of the MVC instance. Such a reduction shows that the MSC problem has a solution of size at most K if and only if the resulting MCCS instance has a solution of cost at most K. This reduction in conjunction with the fact that the MSC problem cannot be approximated to within a factor $o(\log n)$ unless P = NP (Vazirani 2001), implies the result of Part (ii). □

An examination of the reductions used in the proofs of Parts (i) and (ii) of Theorem 4.3 shows that the initial configuration $\mathcal{C}$ produced in the reduction has the following property: The nodes which are in state 1 in $\mathcal{C}$ form an *independent set* in the underlying graph; that is, for any pair of nodes u and v which are in state 1 in $\mathcal{C}$, the edge $\{u, v\}$ is not in the underlying graph. Thus, Part (ii) of Theorem 4.3 points out that even such a restricted version of the MCCS problem cannot be approximated to within a factor $o(\log n)$ for complex BT-SyDSs. Our next result shows that this restricted version can indeed be approximated to within $O(\log n)$. This is done by reducing the problem in an approximation preserving manner to the Minimum Set Multi-Cover problem (MSMC) defined below.

Minimum Set Multi-Cover (MSMC)

Instance: A base set $U = \{u_1, u_2, \ldots, u_n\}$, a positive integer coverage requirement r_i for each element u_i, $1 \leq i \leq n$, a collection $S = \{S_1, S_2, \ldots, S_m\}$ of nonempty subsets of U, a nonnegative cost c_j for each set S_j, $1 \leq j \leq m$, a cost bound β.

Question: Is there a set multi-cover of cost at most β (i.e., is there a subcollection $S' \subseteq S$ such that the cost of S' is at most β and every element u_i of U is covered by at least r_i sets in S'?

It is known that MSMC is NP-hard and that it can be efficiently approximated to within a factor of $O(\log n)$ (Vazirani 2001). We obtain an approximation algorithm for the restricted version of the MCCS problem by reducing it in an approximation preserving manner to the MSMC problem. Some details of this reduction are given in the proof sketch for the following theorem.

Theorem 4.4 *Consider the restricted version of the* MCCS *problem for complex BT-SyDSs where each node has an up threshold of at least 1 and where the nodes in state 1 in the given initial configuration form an independent set in the underlying graph. This version of* MCCS *can be approximated to within a factor $O(\log n)$, where n is the number of nodes in the underlying graph.*

Proof sketch: We show how the restricted version of the MCCS problem can be reduced to an instance of the MSMC problem such that any solution of cost α to the MCCS problem corresponds to a solution of cost α to the MSMC instance and vice versa.

Let $G(V, E)$ denote the underlying graph of $\mathcal{S}$. Let $V_1 \subseteq V$ denote the subset of nodes whose values in $\mathcal{C}$ are 1 and let $V_2 = V - V_1$. The set V_2 represents the set of elements to be covered in the MSMC instance. For each node $u \in V_1$, we construct a set S_u of the neighbors of u from V_2. The cost of S_u is the modification cost $c(u)$ of u. For each node $v \in V_2$, let $t_{\mathrm{up}}(v)$ denote its up threshold and let d_v denote the number of neighbors of v from V_1. The coverage requirement r_v for v is set to $\max\{d_v - t_{\mathrm{up}}(v) + 1,\ 0\}$. Elements with coverage requirement of 0 can be deleted. We thus have an instance of the MSMC problem. We now show that any solution of cost α for the MSMC problem corresponds to a solution of the same cost for the MCCS problem.

Suppose there is a solution of cost α for the MCCS problem. Let the set of frozen nodes in this solution be denoted by V'. Consider the solution to the MSMC problem by choosing the set S_u for each node $u \in V'$. It can be verified that this collection satisfies the multi-cover requirement and that its cost is α.

Now, suppose there is a solution of cost α for the MSMC problem. Construct the set of nodes frozen V' by adding each node u such that the set S_u is in the solution to the MSMC problem. Consider the configuration $\mathcal{C}_1$ obtained by freezing the nodes in V' to 0. Using the fact that nodes which are in state 1 in $\mathcal{C}$ form an independent set, it can be shown that the successor of $\mathcal{C}_1$ is the configuration in which all nodes are in state 0. Since each node has an up threshold of at least 1, the configuration of all 0's is a fixed point for $\mathcal{S}$. It follows that $\mathcal{C}_1$ is a pseudo fixed point.

Thus, the required approximation algorithm for MCCS first constructs an instance of the MSMC problem and uses the known $O(\log n)$ approximation for that problem. $\square$

Whether there is a good approximation algorithm for the general version of the MCCS problem for complex BT-SyDSs is open.

5 An Algorithm for Finding Critical Nodes

We reiterate that a critical set, for our purposes in this paper, is a set of agents that behave in a desired way (corresponding to state 0) and that do not deviate from this behavior. They thus influence their neighbors to refrain from undesirable behaviors (state 1) such as excessive drinking, binge eating, and smoking.

The approximation algorithm for the MCCS problem mentioned in the proof of Theorem 4.4 is based on a greedy approach which proceeds as follows (Vazirani 2001). In each iteration, it chooses a set for which the unit cost of covering an element (i.e., the ratio of the cost of a set to the number of elements covered by the set) is a minimum among the remaining sets. The algorithm terminates when the coverage requirement for all the elements has been satisfied. In this section, we use this approach to develop an efficient heuristic (called **Maximum Contributor Heuristic** or MCH) to compute a set B of critical nodes that attempts to eliminate all $0 \rightarrow 1$ state transitions. An important advantage of MCH is that it can be employed for both deterministic and probabilistic diffusion. Also, our method increases the candidate set of nodes from which critical sets are obtained, and thus will perform at least as well and often better than a method such as (Kuhlman et al. 2010a). For simplicity, it is assumed all the nodes have the same modification cost. With this assumption, the greedy approach has a simple interpretation: in each iteration, choose a set that covers the maximum number of elements. The algorithm can be readily changed to allow nodes to have different modification costs.

First we introduce a few definitions and notation, and describe the central aspect of the algorithm's operation, followed by the algorithm itself. We say a node v is **affectable** at time i if at this time it is in state 0 and and at time $i+1$ it transitions to state 1 (i.e., v has a number of neighbors in state 1 such that $nbr_1(v) \geq t_{up}(v)$). Let A_i be the set of all affectable nodes at time i. If a neighbor $y \in nbr_1(v)$ is chosen for the critical set B, so that y is frozen at 0, then v may cease to be affectable because $nbr_1(v)$ is reduced. If so, v is removed from A_i. A node u_i^{mc} is a **maximum contributor** if it is in state 1 at time i and has the greatest number of neighbors that are affectable at time i. By definition, $u_i^{mc} \in Q_i$, the set of all nodes that are in state 1 at time i. Our goal is to select a minimum set of critical nodes from Q_i that reduces A_i to the empty set. We compute a critical node set B for a diffusion instance using Algorithm 1. The loop at line marker 1 is executed at each time i, and selects a node u_i^{mc} and updates A_i and Q_i as just described.

6 Networks and Experimental Parameters

Because a goal is to evaluate the influence of network structure on blocking behavior, the three networks of Figure 2 were chosen for study due to their ranges in properties such as numbers of nodes and edges, average degree d_a, and average clustering coefficients cc_a. Variations from a factor of 2 (on numbers of edges and in d_a) to an order of magnitude (in cc_a) were evaluated. We also wanted to investigate networks much larger than those of high school students used in a youth smoking study (Kuhlman et al. 2011). All networks were taken as undirected so that diffusion can occur between pairs of nodes in either direction.

Algorithm 1: Compute Critical Set For One Iteration

input : sequence of sets $(Q_i)_{i=1}^{i_{max}}$ of all nodes in state 1 at time i; network G.
output : set of critical nodes B.

Determine from $(Q_i)_{i=1}^{i_{max}}$ the sequence of sets $(A_i)_{i=1}^{i_{max}}$.
Set r =MAX_INT.
for *(* $i = 1$ **to** i_{max} *)* **do**
    // If the number of nodes in state 1 at time i is less than or equal to β, then
    // this set is the critical set.
    if *(* $|Q_i| \leq \beta$ *)* **then** Set $B = Q_i$; Return B.
    // Execute greedy covering strategy.
    Set $B_i = \emptyset$.
    if *(* A_i *is empty)* **then**
        Set B as the set of β nodes randomly chosen from Q_i; Return B.
1   **while** *((* A_i *not empty)* **and** *(* $|B_i| \leq \beta$ *))* **do**
        Identify and remove u_i^{mc} from Q_i; add u_i^{mc} to B_i.
        Remove all nodes $v \in A_i$ where $nbr_1(v) < t_{up}(v)$.
    if *(* A_i *is empty)* **then** Return B_i.
    else if *(* $|A_i| < r$ *)* **then** Set $i_{low} = i$ and $r = |A_i|$.
Return $B_{i_{low}}$.

Since a focus is stopping diffusion, we use an aggressive seeding approach to strain the MCH critical node determination algorithm: we select seed nodes (i.e., nodes initially in state 1) such that they form a connected subgraph of the 20-core of a network. That is, each seeded node is very well connected, with at least 20 neighbors, to promote diffusion, which is a more stringent condition than those used in other works; e.g., (Habiba et al. 2008).

An **iteration** is a diffusion instance and a **simulation** is a set of iterations. Each iteration of a simulation only varies in the composition of the seed node set. For deterministic simulations, 50 seed sets were used with each parameter set. For probabilistic diffusion, 20 instances were completed for each seed set to capture stochastic propagation, so 1000 iterations (20×50) comprise one simulation. We chose breadth (20 stochastic diffusion instances per experiment, for some 1600 parameter sets) over depth, which was justified by the

- **Networks**
  1. `epinions`: edges represent trust relationships between reviewers on Epinions.com; 75879 nodes, 405740 edges, $d_a = 10.7$, $cc_a = 0.138$.
  2. `slashdot`: edges represent friend and foe relationships of participants on Slashdot tech news; 77360 nodes, 469180 edges, $d_a = 12.1$, $cc_a = 0.0555$.
  3. `astroph`: edges join coauthors of papers published on arXiv in the Astro Physics category; 18771 nodes, 198050 edges, $d_a = 21.1$, $cc_a = 0.633$.
- **Study Parameters**
  1. **networks:** the 3 above.
  2. **threshold sets** (t_{up}, t_{down})**:** all combinations of values 1 to 3.
  3. **numbers of seed nodes** n_s**:** 2, 3, 5, 10, 20. (Seeding of 20-core.)
  4. **budgets on critical set sizes** β**:** 16 values from 5 to 2000.
  5. **state transitions probabilities** (p_{up}, p_{down})**:** (0.5, 0.5), (1,1).
- **Study Metrics**
  1. **number of parameter sets:** over 1600.
  2. **number of diffusion instances:** over 570000.
  3. **number of critical sets determined:** over 390000.

Figure 2: Networks (J. Leskovec website 2009), parameters, and study metrics.

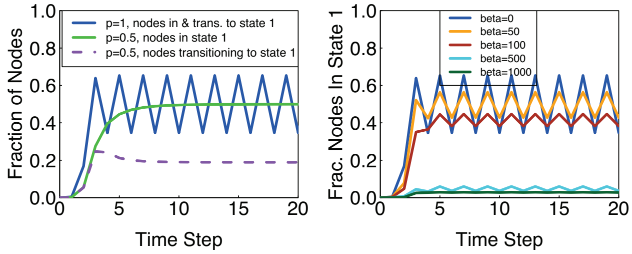
small average and maximum coefficient-of-variation (COV) values (0.099 and 0.22, respectively) of computed critical node sets over all experiments.

Simulations were first performed with no critical nodes (i.e., $\beta = 0$). Results were used as input to Algorithm 1. The critical node sets were then used in a second set of simulations to quantify the effect of critical nodes on diffusion dynamics. For deterministic diffusion, there is one critical node set for each seed node set; for probabilistic diffusion we use a different approach described below with the results.

7 Simulation Results

We present empirical results from the simulation study. We demonstrate fundamental differences between deterministic and stochastic diffusion dynamics, but show that both systems will not reach a pseudo fixed point without an external forcing mechanism. The mechanism we use is the introduction of critical nodes. With increasing numbers β of critical nodes, transitions from state 0 to state 1 can be eliminated; we implicitly assume that state 1 is undesirable and hence we seek to minimize the numbers of nodes in this state, and to ever reach state 1. Network structure plays a large role in the dynamics: for small β, the connectedness of a network as measured by the largest component of a k-core governs behavior, while for large β, the number of nodes with large degrees controls behavior. We also compare the efficacy of deterministic, probabilistic, and hybrid probabilistic critical node schemes and show that neither probabilistic scheme always outperforms the other. All data shown represent the average of 50 and 1000 instances for deterministic and probabilistic diffusion, respectively. Also, since $p_{up} = p_{down}$ for all results, we use p for both.

Baseline behaviors for deterministic and stochastic diffusion, and critical nodes. Network dynamics for deterministic and probabilistic diffusion within the `slashdot` network are depicted in Figure 3 for $n_s = 2$ seed nodes. Figure



(a) (b)

Figure 3: Dynamics for the `slashdot` network with $t_{up} = t_{down} = 1$ and $n_s = 2$. (a) Fraction of nodes currently in state 1 and transitioning to state 1 for $p = 1$ and $\beta = 0$ (blue curve); fraction of nodes currently in state 1 for $p = 0.5$ and $\beta = 0$ (green curve); and fraction of nodes transitioning to state 1 for $p = 0.5$ and $\beta = 0$ (purple curve); and (b) fraction of nodes currently in state 1 for $p = 1$ with critical nodes.

3(a) shows the fraction of nodes in state 1 for deterministic diffusion (blue oscillating curve), which is also the fraction of nodes *transitioning* to state 1 at each time step; i.e., once a node is in state 1, it changes state between 0 and 1 at every subsequent time step. Thus, a 2-cycle is formed in phase space for deterministic diffusion. Also shown for probabilistic diffusion are the fraction of nodes in state 1 (green curve), and fraction of nodes transitioning to state 1 (dashed purple curve). Although the extreme oscillations for the deterministic results (blue curve) are probably unrealistic for most practical applications, the point is that stochasticity can markedly dampen these oscillations and can also dramatically reduce the numbers of state transitions at each time step (blue curve vs. purple curve).

Figure 3(b) shows the fraction of nodes in state 1 for different numbers of critical nodes. As β increases from 0 to 1000, the number of nodes in state 1 decreases and the oscillation amplitude also decreases.

Figure 3 shows that once a non-zero set of nodes reaches state 1, a system will not drive itself to zero nodes in state 1. This holds for deterministic and probabilistic diffusion (for probabilistic diffusion, at least not in the short-term). The only way to achieve the objective of very few or zero nodes in state 1 is to prevent nodes from ever reaching state 1. Controlling diffusion with critical nodes can accomplish this goal. For networks of large maximum degree (herein, maximum degrees can be as high as 3000), we see that the needed number β of critical nodes can be larger than $500n_s$.

From a practical standpoint, this suggests that absent some intervention, in realistic populations where $t_{up}, t_{down} \approx 1 - 3$, undesirable behaviors will not be naturally driven from a system. Peer influence or some other factor such as an advertising campaign must be used to drive the unwanted behavior from the population. Here we use peer influence, through the MCH algorithm, in the form of individuals who exhibit a desired behavior, cannot be persuaded to adopt an undesirable behavior, and hence who influence others to adopt the wanted behavior.

Comparison of the MCH blocking algorithm with one for simple contagions. We compare the efficacy of MCH for selecting critical nodes with a strategy that works well in determining critical nodes for simple contagions where the only state transition is $0 \to 1$: setting high degree nodes as critical (Habiba et al. 2008). We refer to this approach as the high degree strategy (HDS). Table 1 summarizes the average fraction of nodes ever reaching state 1 for the three networks and 3 threshold sets. All results are for deterministic diffusion. Numbers of critical nodes are also given as a fraction of network nodes in column 3. The number of seed nodes is 2; a smaller number means the driving force for diffusion is smaller and hence easier to block, thereby enhancing the effectiveness of HDS. The table shows, e.g., that for HDS and threshold sets (1,1) and (1,2), 88% of `astroph` nodes on average reach state 1; for MCH, no nodes reach state 1.

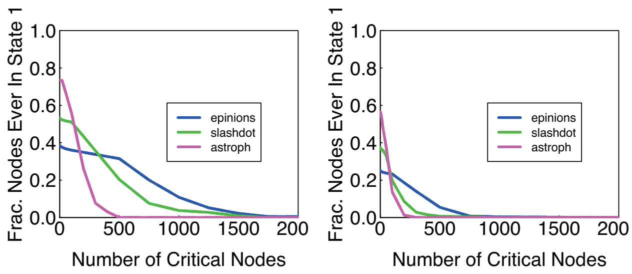
Table 1: Comparison of High Degree Strategy and Maximum Contributor Heuristic. The last two columns show the fraction of network nodes that reach state 1. MCH is far more effective in blocking diffusion.

Network	**Threshold Sets,** (t_{up}, t_{down})	**Num. (and %) Critical Nodes**	**Frac. MCH**	**Frac. HDS**
`astroph`	(1,1), (1,2)	500 (3%)	0.00007	0.88
`astroph`	(2,2)	50 (0.3%)	0.0001	0.64
`slashdot`	(1,1), (1,2)	2000 (3%)	0.018	0.63
`slashdot`	(2,2)	50 (0.1%)	0.00003	0.15
`epinions`	(1,1), (1,2)	1000 (1%)	0.00002	0.76
`epinions`	(2,2)	50 (0.1%)	0.00002	0.25

Performance of the MCH blocking algorithm. Fractions of nodes ever reaching state 1 where $n_s = 20$ and $t_{down} = 1$ are displayed in Figure 4. The first and second plots are for $t_{up} = 2$ and 3, respectively. Both plots show "crossover" across the networks because of graph structure effects. For small β, `astroph` has the greatest fraction of nodes reaching state 1, and the largest t-core size for t-threshold diffusion, while for larger β, the larger network with the greatest degrees (i.e., `epinions`) enables diffusion to circumvent the critical nodes using these "hub" nodes. These results illustrate that as t_{up} increases, the number of critical nodes required to halt all diffusion decreases; e.g., the `slashdot` network requires 1500 critical nodes when $t_{up} = 2$ and 500 when $t_{up} = 3$. In contrast, decreasing t_{down} (to promote transitions to state 0) has little effect on critical set sizes (results not shown). These results are consistent with the intuitive notion that it may be easier for society to attack problems by increasing the threshold (or cost) for adoption; these results help to quantify this qualitative observation. We return to the effect of t_{down} at the end of this section.

Interactions among seed set sizes, critical nodes, and deterministic and probabilistic diffusion are displayed in Figure 5 for the `epinions` network. The first plot shows deterministic diffusion results with $t_{up} = t_{down} = 2$. For $n_s = 20$, roughly $\beta = 100n_s = 2000$ critical nodes are required to stop diffusion.

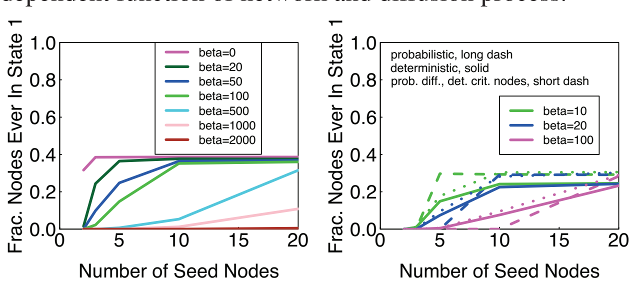
We compare deterministic and probabilistic diffusion and blocking implementations in Figure 5(b). In the absence of critical nodes, the discriminating factor in diffusion is deterministic versus probabilistic state transitions; the seed sets



(a) (b)

Figure 4: Fraction of nodes ever to reach state 1 as a function of β for the three networks where $p = 1$, $t_{down} = 1$, and $n_s = 20$; (a) $t_{up} = 2$; and (b) $t_{up} = 3$. Results are virtually identical for $t_{down} = 2, 3$ and are not shown.

are the same in both sets of simulations. For deterministic simulations, a critical node set is computed for each seed set. For stochastic simulations, a critical set is determined for each of the 20 instances that have the same seed node set. The β nodes with the greatest frequency of occurrence within the 20 sets are taken as the sole critical set and applied to all 20 iterations to assess their blocking capability. Figure 5(b) illustrates differences between probabilistic and deterministic diffusion (long dash vs. solid curves) for the epinions network, for 3 critical set sizes. For $\beta = 10$, probabilistic diffusion gives greater fractions of nodes reaching state 1, while for $\beta = 20$ and 100 the deterministic and probabilistic curves intersect. These behaviors are caused by two competing mechanisms. On one hand, stochastic diffusion has a driving force that is less than that for $p = 1$. On the other hand, critical nodes are determined over 20 instances and represent an averaging process. The critical node sets for deterministic diffusion, in contrast, are for a particular seed set. Which factors dominate are a problem-dependent function of network and diffusion process.



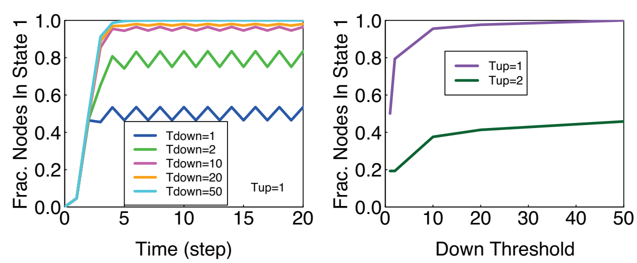
(a) (b)

Figure 5: For the epinions network, the fractions of nodes ever in state 1 for different numbers of seed and critical nodes: (a) for $p = 1$, $t_{up} = t_{down} = 2$; and (b) for $p = 0.5$ and 1, $t_{up} = t_{down} = 3$.

We also used the critical node sets from deterministic diffusion in simulations of stochastic diffusion (short dashes Figure 5(b)). The deterministic blocking nodes are less effective, relative to the probabilistic critical nodes, for $\beta = 100$, but are more effective for $\beta = 10$. As the number of nodes ever reaching state 1 is driven to zero, the two methods converge because all diffusion will be halted at time step 1. Less computational effort is required to produce the critical nodes for deterministic diffusion.

Effect of t_{down} in perpetuating an unwanted behavior. A natural question is how large must t_{down} be to prohibit any node from transitioning from state 1 to state 0, thereby reducing the bi-threshold system to a ratchet-up system, and "locking-in" the unwanted behavior. Clearly $t_{down} = d_{max} + 1$, where d_{max} is the maximum degree of any node in a network, will result in a system where the unwanted behavior is prevalent because no node has $d_{max} + 1$ neighbors. It turns out that much smaller values of t_{down} can suffice.

Figure 6(a) shows the fraction of nodes in state 1 for t_{down} values ranging from 1 to 50. As t_{down} increases, the difficulty in transitioning to state 0 increases. For $t_{down} = 50$, all nodes move to and remain in state 1. Since $t_{up} = 1$, a connected graph will produce a cascade where all nodes are in state 1, and hence the system will reach a fixed point. If we take the mean of the steady state value of each curve in Figure 6(a) and plot them against t_{down}, we obtain the purple curve in Figure 6(b). The other curve is generated from analogous data where $t_{up} = 2$. The theoretical maximum in spread size can be computed based on k-cores and is 49% of nodes for the epinions network and $t_{up} = 2$, which agrees with the experimental result. From these data, we make two observations. First, since $d_{max} = 3044$ for epinions, the system behaves as a ratchet-up system for $t_{down} \ll (d_{max} + 1)$. Second, by the time t_{down} increases to 10 (< 50), these systems are asymptotically approaching ratchet up systems. From a practical standpoint, these results tell us that a large t_{down} must be reduced in order for people to revert to state 0. This may be the case, for example, for drug use, where reducing addiction may reduce t_{down}. Thus, interventions may also be needed to reduce the impediments to giving up undesirable behavior.



(a) (b)

Figure 6: For the epinions network: (a) the fraction of nodes currently in state 1 for $t_{up} = 1$, $p = 1$ and $n_s = 20$; (b) mean fraction of nodes in state 1 at steady state as a function of t_{down} for $t_{up} = 1$ and $t_{up} = 2$.

8 Summary and Conclusions

We presented theoretical and experimental results for understanding bi-threshold systems and blocking diffusion in such systems. Investigation of the latter issue was specifically called for in (Lopez 2008) where it was speculated that peer influence may be used advantageously to stop the spread of obesity. We defined the minimum cost critical set (MCCS) problem for inhibiting the spread of undesirable contagions and presented theoretical results for both simple and complex contagions. While the question of devising an approximation algorithm with a provably good performance guarantee for the MCCS problem for complex BT-SyDSs is open, we presented a heuristic algorithm to find critical nodes for both deterministic and probabilistic diffusion. We showed that our heuristic outperforms the strategy of setting the highest degree nodes critical by multiple orders of magnitude. In a companion work, the bi-threshold model has been applied to study adolescent smoking behavior (Kuhlman et al. 2011).

Acknowledgments. We thank the reviewers, our external collaborators, and members of NDSSL for their suggestions and comments. This work has been partially supported by NIH MIDAS project 2U01GM070694-7, NSF PetaApps Grant OCI-0904844, DTRA R&D Grant HDTRA1-0901-0017, DTRA CNIMS Grant HDTRA1-07-C- 0113, NSF Netse CNS-1011769, NSF SDCI OCI-1032677, NSF CNS-0845700, and DOE DE-SC0003957.

References

Atkinson, R.; Dietz, W.; Foreyt, J.; Goodwin, N.; Hill, J.; Hirsch, J.; Pi-Sunyer, F.; Weinsier, R.; Wing, R.; Hoofnagle, J.; Everhart, J.; Hubbard, V.; and Yanovski, S. 1994. Weight Cycling. *Journal of the American Medical Association* 272(15):1196–1202.

Axelrod, R. 1997. The Dissemination of Culture. *Journal of Conflict Resolution* 41(2):203–226.

Barrett, C. L.; Hunt III, H. B.; 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.

Bischi, G., and Merlone, U. 2009. Global Dynamics in Binary Choice Models with Social Influence. *J. Math. Sociology* 33:277–302.

Borgatti, S. 2006. Identifying sets of key players in a social network. *Comput. Math. Organiz. Theory* 12:21–34.

Borsari, B., and Carey, K. 2001. Peer Influences of College Drinking: A Review of the Research. *Journal of Substance Abuse* 13:391–424.

Centola, D., and Macy, M. 2007. Complex Contagions and the Weakness of Long Ties. *American J. Sociology* 113(3):702–734.

Chakrabarti, D.; Wang, Y.; Wang, C.; Leskovec, J.; and Faloutsos, C. 2008. Epidemic Thresholds in Real Networks. *ACM Trans. Inf. Syst. Secur.* 10(4):13–1–13–26.

Dodds, P., and Watts, D. 2005. A Generalized Model of Social and Biological Contagion. *J. Theo. Biology* 232(4):587–604.

Domingos, P., and Richardson, M. 2001. Mining the Network Value of Customers. In *Proc. ACM KDD*, 57–61.

Dreyer, P., and Roberts, F. 2009. Irreversible k-Threshold Processes: Graph-Theoretical Threshold Models of the Spread of Disease and Opinion. *Discr. Appl. Math.* 157:1615–1627.

Eubank, S.; Kumar, V. S. A.; Marathe, M. V.; Srinivasan, A.; and Wang, N. 2006. Structure of Social Contact Networks and Their Impact on Epidemics. In *Discrete Methods in Epidemiology*, volume 70. American Math. Soc., Providence, RI. 179–200.

Even-Dar, E., and Shapira, A. 2007. A Note on Maximizing the Spread of Influence in Social Networks. In *WINE 2007, LNCS 4858*, 281–286.

Finkelstein, E.; Trogdon, J.; Cohen, J.; and Dietz, W. 2009. Annual Medical Spending Attributable to Obesity: Payer- and Service-Specific Estimates. *Health Affairs* 28(5):w822–w831.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. San Francisco, CA: W. H. Freeman and Co.

Granovetter, M. 1978. Threshold Models of Collective Behavior. *American J. Sociology* 83(6):1420–1443.

Gruhl, D.; Guha, R.; Liben-Nowell, D.; and Tomkins, A. 2004. Information Diffusion Through Blogspace. In *Proc. WWW*, 491–501.

Guha, R.; Kumar, R.; Raghavan, P.; and Tomkins, A. 2004. Propagation of Trust and Distrust. In *Proc. WWW*, 403–412.

Habiba; Yu, Y.; Berger-Wolf, T.; and Saia, J. 2008. Finding Spread Blockers in Dynamic Networks. In *Proc. SNA-KDD Workshop*.

Hoffman, B. R.; Sussman, S.; Unger, J. B.; and Valente, T. W. 2006. Peer influences on adolescent cigarette smoking: A theoretical review of the literature. *Substance Use and Misuse* 41:103–155.

J. Leskovec website. 2009. 8 April 2009, http://cs.stanford.edu/people/jure/.

Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the Spread of Influence Through a Social Network. In *Proc. ACM KDD*, 137–146.

Kuhlman, C.; Kumar, V.; Marathe, M.; Ravi, S.; and Rosenkrantz, D. 2010a. Finding Critical Nodes for Inhibiting Diffusion of Complex Contagions in Social Networks. In *Proc. ECML PKDD*.

Kuhlman, C. J.; Kumar, V. S. A.; Marathe, M. V.; Swarup, S.; Tuli, G.; Ravi, S. S.; and Rosenkrantz, D. J. 2010b. Theoretical Results on Bi-Threshold Dynamical Systems. NDSSL Technical Report No. 11-089.

Kuhlman, C.; Kumar, V.; Marathe, M.; Swarup, S.; Tuli, G.; Ravi, S.; and Rosenkrantz, D. 2011. A Bi-Threshold Model of Complex Contagion and its Application to the Spread of Smoking Behavior. In *Proc. SNA-KDD Workshop*.

Leskovec, J.; Adamic, L.; and Huberman, B. 2007. The Dynamics of Viral Marketing. *ACM Trans. on the Web* 1(1).

Lopez, R. 2008. Is Obesity Contagious? *Expert Rev. Endocrinol. Metab.* 3(1):21—22.

Mossel, E., and Roch, S. 2007. On the Submodularity of Influence in Social Networks. In *Proc. ACM STOC*, 128–134.

O'Loughlin, J.; Karp, I.; Koulis, T.; Paradis, G.; and DiFranza, J. 2009. Determinants of First Puff and Daily Cigarette Smoking in Adolescents. *American J. Epidemiology* 170(5):585–597.

Raz, R., and Safra, S. 1997. A Sub-constant Error-Probability Low-Degree Test, and Sub-constant Error-Probability PCP Characterization of NP. In *Proc. 29th Ann. ACM Symp. on Theory of Comp. (STOC 1997)*, 475–484.

Richardson, M., and Domingos, P. 2002. Mining Knowledge-Sharing Sites for Viral Marketing. In *Proc. ACM KDD*, 61–70.

Rosenstock, I.; Strecher, V.; and Becker, M. 1988. Social Learning Theory and the Health Belief Model. *Health Education Quarterly* 15(2):175–183.

Schelling, T. 1978. *Micromotives and Macrobehavior*. W. W. Norton and Company.

Sturm, R. 2002. The Effects Of Obesity, Smoking, And Drinking On Medical Problems And Costs. *Health Affairs* 21(2):245–253.

Vazirani, V. V. 2001. *Approximation Algorithms*. Springer.

Watts, D. 2002. A Simple Model of Global Cascades on Random Networks. *PNAS* 99(9):5766–5771.