Irreversible k-Threshold Processes: Graph-theoretical Threshold Models of the Spread of Disease and of Opinion

Paul A. Dreyer, Jr.
The RAND Corporation
Santa Monica, CA
dreyer@rand.org

Fred S. Roberts
DIMACS
Rutgers University
Piscataway, NJ
froberts@dimacs.rutgers.edu

Acknowledgements
The authors would like to thank the National Science Foundation for supporting this work through grants CCR 91-19999, BIR 94-12594, DBI 99-82983, SBR 97-09134, SES-92-11492 and EIA-02-05116 to Rutgers University.

Abstract

We will consider models of the spread of disease or opinion through social networks, represented as graphs. In our models, vertices will be in one of two states, 1 ("infected") or 0 ("uninfected") and change of state will take place at discrete times. After describing several such models, we will concentrate on the model, called an irreversible k-threshold process, where a vertex enters state 1 if at least k of its neighbors are in state 1, and where a vertex never leaves state 1 once it is in it. We will seek sets of vertices with the property that, if they are in state 1 at the beginning, then eventually all vertices end up in state 1. Such vertex sets correspond to vertices that can be infected with a disease or opinion so as to guarantee saturation of the population with the disease or opinion. We will also discuss ways to "defend" against such saturating sets, for example by "vaccination" or designing network topologies.

1 Introduction

Mathematical modeling of the spread of infectious disease has a long history, going back to the work of Bernoulli on smallpox in 1760. Much of the modern disease modeling literature studies the spread of disease using methods of dynamical systems. However, as diseases are spread through social networks, we can gain insight by using simple graph-theoretical models. The models we will describe have a variety of other applications, and in particular fall into an extensive literature on the spread of opinions through networks. Sample early work in this literature can be found in the work of Poljak and Sûra [24], DeGroot [3] and French [8].

We will consider models in which a network is represented by a graph $G = (V, E)$, where V is a set of vertices representing people and E means that the people are connected in the network. We will assume that each vertex v_i in a network assumes a *value* or is in a *state* $x_i(t)$ at time t. In our models, we will consider these values as changing only at discrete times, $t = 0, 1, \ldots$ and we will assume that each value is in $\{0, 1\}$. We will consider different models for how these values change over time, depending on values held by neighboring vertices in the graph G. In the disease application, we think of 1 as corresponding to a vertex having a given disease and 0 as corresponding to it not having the disease. In the case of opinion, we consider either agreement (1) or disagreement (0) with a given position.

Our major emphasis will be on a model we call an irreversible k-threshold process, where a vertex never leaves state 1 (is never cured once infected) and enters state 1 from state 0 if at least k of its neighbors are in state 0. A variation on the model will allow "vaccination" of a vertex, preventing it from ever entering state 1. We will be especially concerned with identifying sets of vertices we will call irreversible k-conversion sets, sets of vertices with the property that if they are all in state 1 at the beginning, then eventually all vertices end up in state 1. Such a set of vertices represents a set of individuals a bioterrorist could infect so as to be sure to infect everyone eventually, or a set of individuals we could recruit to have a given opinion so as to eventually "infect" everyone with this opinion. Defense against infection would then amount to finding good "vaccination" or "quarantine" strategies or to designing the network topology so as to prevent the network from having small irreversible k-conversion sets.

1.1 Threshold Processes

In a *k-threshold process*, we fix a number k and assume that a vertex changes its state at time $t+1$ if at least k of its neighbors are in the opposite state at time t. The number k is a threshold for state change. In an *irreversible k-threshold process*, a vertex never changes from state 1 to state 0, but changes from state 0 to 1 if at least k of its neighbors are in state 0 at the previous time. In terms of diseases, the idea is that an infection spreads to a vertex if sufficiently many of its neighbors are infected. In some models, $k = 1$ will suffice. It does not seem likely that ordinary k-threshold processes make sense for disease spread, since it is hard to understand the motivation for switching from 1 to 0. In this paper, we will concentrate on irreversible k-threshold processes. Extensive analogous results on k-threshold processes are contained in [4]. So are results about the *monotonic k-threshold processes* where the set of vertices in state 1 at time t is a subset of the set at time $t+1$.

1.2 Majority Processes

Mention should be made of *majority processes,* where at each time step, a vertex v changes its state if and only if at least half of its neighbors have the opposite state. There are variations of the majority process which include the state of v in the determination of majority, along with various rules for breaking ties (always choose 1 or always 0, always keep the current state, always switch states). These processes do not seem to be relevant to the spread of disease, though they do seem interesting for opinion change. We mention them here because they motivate our interest in a key notion of "dynamo" or "conversion" set that we shall define below.

Majority processes (sometimes also called repetitive polling processes) have had applications in problems in distributed computing (see [5], [18]). A concrete example of an application in distributed computing, in particular, maintaining data consistency, was suggested by Peleg in [23]. Assume that all of the processors in a distributed network store an identical bit of data and some of these bits get changed for whatever reason (rounding error, for example). If a processor discovers that the values of the bits of some of its neighbors are different, a majority process could be used to update the values of all of the processors until every processor has the same value for its bit. In a general context, though, there is no guarantee that every processor will return to the same value, much less the correct one. Peleg also referred to other applications of majority processes to distributed computing in system-level diagnosis and resource allocation.

The problem of maintaining data consistency raises an interesting question: for what graphs does the majority process always lead to the situation of all of the vertices having the same state as the initial global majority state, regardless of the initial state configuration? In [20], Mustafa and Pekeč analyzed *democratic consensus computers* (d.c.c.'s), graphs that exhibit precisely this behavior. They produced a variety of results, including that a graph that is a d.c.c. has diameter at most four, has a trivial min-cut and a non-unique max-cut, and that for a graph G on n vertices with maximum degree at least $n-3$, if G is a d.c.c., then G must have a vertex (called a *master*) that is adjacent to every other vertex. They conjectured that every d.c.c. contains a master, and the question of a full characterization of d.c.c.'s remains open.

For a given graph, one might wonder what subsets of the vertex set can "force" a majority process to the situation with every vertex in the same state. (More formally, a set of vertices M is a *dynamic monopoly*, or *dynamo* for short, if when a majority process is started with all of the vertices in M in state 1, then the system reaches the fixed point with all vertices being in state 1. Dynamos can be incredibly small; Berger [1] proved that for every n, there exists a graph with at least n vertices containing a dynamo of size 18. In addition, in [22], Peleg showed that for all graphs, the size of a dynamo that converts all vertices to state 1 in a single step (Peleg called such a dynamo a *(static) monopoly*) must be $\Omega(\sqrt{n})$, and that there are graphs with static monopolies of size $O(\sqrt{n})$. We shall consider a similar concept for irreversible threshold processes.

One can also consider special classes of dynamos. For example, if M is a dynamo and P_t denotes the set of vertices of value 1 at time t ($P_0 = M$), then as in monotone k-threshold processes, M is a *monotone dynamo* if $P_t \subseteq P_{t+1}$ for all $t \geq 0$. In addition, one can consider a different update rule where the state of a vertex changes from 0 to 1 if and only if more than half of its neighbors are in state 1, but not vice versa. These *irreversible* majority processes (and the associated *irreversible dynamos*) appear in the context of fault-tolerant computing, where a vertex entering state 1 corresponds to permanent faults occuring in the associated processor. In [5], [6], and [18], bounds are shown for both monotone and irreversible dynamos in toroidal graphs of various types and in butterflies, two graph structures which appear in parallel computation. In addition, in [23], Peleg showed that for most variants of majority processes, every monotone dynamo is of size $\Omega(\sqrt{n})$. There are, however, some majority process variants with constant size monotone dynamos; in the majority process where a vertex's state is counted in its own majority determination and in case of ties, the vertex enters state 1, any pair of adjacent vertices in the cycle C_n is a monotone dynamo.

Majority processes are an interesting model for opinion change in social networks. However, in the case of disease, they do not seem to make a lot of sense.

1.3 Vaccination

Motivated by the disease application, we can think of modifying an irreversible threshold process by allowing us to "vaccinate" certain vertices that are in state 0. Once a vertex is vaccinated, it can never change into the infected state 1. There are several variants of the vaccination problem that one can consider. One is to only allow vaccination at the beginning (time $t = 0$), say of f vertices (where f reflects a limitation on the amount of vaccine available). An alternative is to allow us to vaccinate $f(t)$ vertices at time t. The vaccination problem has been studied in the context of firefighting. We can think of the vertices as trees in a forest, and the state 1 corresponding to a tree being on fire. A fire spreads to neighboring trees under different rules, e.g., in an irreversible 1-threshold process. In such a process, if a tree is burning at time t, all its neighbors burn at time $t+1$ except those that are protected by "vaccination" or, what is equivalent, by placement of firefighters on them at a time no later than time t. There is a rather robust literature about this problem. We give some sample results here.

Consider first infinite d-dimensional grids, where the neighbors of a vertex are those adjacent in one of the dimensions, and consider epidemics starting at a single vertex. If $d = 1$, then every epidemic can be stopped with one dose of vaccine per time period. Here, "stopped" means that it dies out after a finite number of time steps after which no other vertices are infected – enter state 1. If $d = 2$, then clearly one dose of vaccine per time period is not sufficient to control an epidemic that spreads in this planar grid. Two firefighters per time period suffice. Wang and Moeller [25] observe that for every *r-regular graph*, i.e., graph in which every vertex has exactly r neighbors, every epidemic starting at a single vertex can be controlled by using $r - 1$ doses of vaccine per time step. This is because we first vaccinate $r - 1$ neighbors of the vertex where the epidemic starts, the epidemic then spreads to the remaining neighbor, and we then vaccinate the $r - 1$ neighbors of this vertex that are not yet in the diseased state. This observation shows that if $d \geq 3$, then an epidemic starting at one vertex in the d-dimensional infinite grid can be controlled with $2d-1$ doses of vaccine per time step. However, Hartke [10] showed that $2d-2$ doses per time step do not suffice. If we allow outbreaks to start at a finite number of vertices, then for $d = 2$, Fogarty [7] showed that two doses per time step suffice to control any epidemic. However, for $d \geq 3$, Hartke [10] showed that, for every value of f, there is always an outbreak that cannot be controlled with f doses per time step. This led Ng and Raff [21] to consider a variable number $f(t)$ of doses per time step. They showed that if $f(t)$ is periodic with period $p_f \geq 1$, and if $R_f = [f(1) + f(2) + \ldots + f(p_f)]/p_f$, then if $d = 2$ and $R_f > 1.5$, any finite outbreak on the infinite planar grid can be controlled.

For some results on finite planar grids, see [10]. More generally, consider epidemics on finite graphs that break out at a single vertex. In general, the problem of determining if an outbreak at a single vertex in a given graph can be controlled in T time steps by using one dose of vaccine per time step is NP-complete (MacGillivray and Wang [19]). This same problem is of interest for special graphs such as trees. In an epidemic that starts at a root vertex of a tree, we can ask for an optimal strategy for controlling the epidemic by using one dose of vaccine per time step. A greedy algorithm would start at all vertices one step from the root and consider for each its weight, i.e., one plus the number of "descendants." The algorithm would choose to vaccinate a vertex of highest weight, choosing at random in case of ties. The epidemic would then spread to all unvaccinated neighbors of the root. We would next consider all neighbors of an infected vertex and vaccinate the neighbor with the largest weight, again choosing randomly in case of ties. This is a simple algorithm. It is easy to show it does not lead to an optimal number of uninfected vertices by the time an epidemic runs its course. However, Hartnell and Li [11] showed that the greedy algorithm always "saves" more than half the number of vertices saved by the optimal strategy.

In the rest of this paper, we will not allow vaccinations. However, we will allow epidemics to break out at more than one vertex.

1.4 $k-$Conversion Sets

For the rest of this paper, we consider a k-irreversible threshold process on a graph G. Let $V_1(t)$ be the set of all vertices in state 1 at time t and similarly $V_0(t)$. (Motivated by the notion of dynamo discussed in Section 1.2, we define a set of vertices S to be an *irreversible $k-$conversion set* if when an irreversible $k-$threshold process is started with $S \subseteq V_1(0)$, then the process reaches the situation with all of the vertices in state 1. We can define k-conversion sets for ordinary k-threshold processes in an analogous manner, and in such processes, it is also of interest to study k-conversion sets in monotone k-threshold processes. These two concepts are studied in detail in [4]. Irreversible $1-$conversion is trivial, since in a connected graph, if any vertex is in state 1, then at time $t = 1$, all of its neighbors will be in state 1, and at time $t = 2$, all of the vertices in its $2-$neighborhood will be in state 1, and so on. However, irreversible 1-conversion is of interest if we are allowed to vaccinate. Clearly, the superset of an irreversible k-conversion set is an irreversible k-conversion set. (However, this is not the case for ordinary k-conversion sets. The simplest counterexample occurs with a $2-$threshold process on two copies of K_4, the complete graph on four vertices, with four additional edges joining distinct pairs of vertices, one from each copy of K_4. Let S be one of the copies of K_4 plus any single vertex w from the other copy of K_4. This is a $2-$conversion set for the graph: after one time step, $V_1(1) = V - \{w\}$, $V_0(1) = \{w\}$, and at the next time step, $V_1(2) = V$. However, if another vertex from the second copy of K_4 is added to $V_1(0)$, the $2-$threshold process enters a $2-$cycle, with the state 0 and 1 vertices in the second copy of K_4 alternating at each time step.) Irreversible conversion sets have clear and interesting interpretations in various applications. In the case of opinion, they correspond to sets of individuals with the property that if they agree with an opinion at the beginning, then the entire group is guaranteed to agree with that opinion eventually. In the case of disease, we are looking for a set of vertices that an intentional attacker (bioterrorist) can infect to guarantee to infect all vertices eventually. In terms of bioterrorism defense, the goal is to design social networks so that irreversible conversion sets are either impossible to find or always large enough to offset the ability of an attacker to infect that large a set. Irreversible conversion sets also have an economic interpretation. We think of achieving saturation of a new product in a market by placing it with vertices of an irreversible conversion set in a process where one buys a product if sufficiently many of our neighbors own it.

Although much work has been done on dynamos in majority networks, conversion sets are a fairly new concept. In [14], Impagliazzo, Paturi, and Saks analyze circuits using *threshold gates*, which take n $\{0,1\}$ inputs $x_1, \ldots, x_n$ and output $\mathbf{1}(\sum_{i=1}^{n}(w_i x_i) - b)$ for a given set of weights $w_1, \ldots, w_n$ on edges and a threshold b, where $\mathbf{1}(u)$ is 1 if $u \geq 1$ and 0 otherwise. Some of their results deal with fixing the values of a subset of the inputs to the circuit to force a constant output to the circuit, regardless of the other inputs to the circuit. This is clearly a generalization of the concept of conversion sets, as every threshold network can be represented by a circuit using threshold gates.

For a graph G, let $C_k(G)$ denote the size of the smallest irreversible $k-$conversion set for G. We will be interested in determining $C_k(G)$ and in finding irreversible k-conversion sets of minimum size.

The rest of this paper is organized as follows: In the next section, we show that it is NP-complete to determine if a graph has an irreversible k-conversion set of size at most d, at least for $k \geq 3$. Section 3 considers special graphs: complete multipartite graphs, trees, and various kinds of grids. It gives exact values and bounds on the size of the smallest irreversible k-conversion set for such graphs. Section 4 gives a general lower bound for the size of such a set in a general graph. We close with a discussion of open problems and future research.

2 NP-Completeness of the Problem of Finding the Smallest Irreversible k-Conversion Set

The Irreversible $k-$CONVERSION SET problem is as follows:

Irreversible $k-$CONVERSION SET (IR$k-$CS)
Instance: A graph G and a positive integer d.
Question: Does G have an irreversible $k-$conversion set D, $|D| \leq d$?

We will show that this problem is NP-complete for a fixed $k \geq 3$, but we must first prove a result about irreversible $r-$conversion sets in $r-$regular graphs that will also be useful later.

Lemma 1. *Let $G = (V, E)$ be a connected $r-$regular graph, and let $D \subseteq V$. Then D is an irreversible $r-$conversion set if and only if $V - D$ is independent.*

Proof. In an irreversible $r-$conversion process, a vertex will switch from state 0 to state 1 if and only if all of its neighbors are in state 1. If $V - D$ is independent, then every vertex in $V - D$ will switch to state 1 at time $t = 1$. If $V - D$ is not independent, then two vertices starting in state 0 share an edge, and those adjacent vertices will never enter state 1. Hence, D cannot be an irreversible conversion set. $\square$

Corollary 1. *Let $G = (V, E)$ be a connected graph with no vertex of degree greater than r, let $S \subseteq V$ be the set of vertices of degree less than r, and let $D \subseteq V$. Then D is an irreversible $r-$conversion set if and only if $S \subseteq D$ and $V - D$ is independent.*

Although we do not use this result immediately, we will now note a similar result about irreversible $r-$conversion sets in $(r+1)-$regular graphs. In a graph G, a *feedback vertex set* is a subset S of the vertices such that the subgraph $G[V - S]$ induced by the vertices in $V - S$ is cycle-free (equivalently, $G[V - S]$ is a forest).

Proposition 1. *Let $G = (V, E)$ be an $(r + 1)-$regular graph, and let $S \subseteq V$. Then S is an irreversible $r-$conversion set if and only if S is a feedback vertex set.*

Proof. If S is not a feedback vertex set, then $V - S$ contains a cycle. Suppose $S = V_0(0)$. At every time step, every vertex v in the cycle will have at least two neighbors in state 0. If S is a feedback vertex set, then $G[V - S]$ is a forest, so at each time step, all of the isolated vertices and leaves in the forest induced by the vertices in state 0 will enter state 1, leaving a smaller forest, and this process will eventually reach the fixed point with the entire vertex set in state 1. $\square$

In a graph G, a *detour* is a simple path (no repeated vertices) of maximum length, and the *detour number*, denoted $dn(G)$, is the length of such a path. The detour number was first studied in [15]. The *transient length* of an irreversible threshold process is that value T of time beyond which no vertex enters state 1.

Lemma 2. *For an irreversible k-threshold process on a connected graph $G = (V, E)$ of order n, the transient length is at most $dn(G)$ ($dn(G) - 1$ if $k > 1$).*

Proof. Assume we have an irreversible $k-$threshold process on a graph G, and assume that there exists a vertex v_i which enters state 1 at a time $T > dn(G)$. Let $E(t)$ denote the set of vertices that enter state 1 at time $t \geq 0$ ($E(0) = V_1(0)$), and assume that $E(T') = \emptyset$ for all $T' > T$. The $E(t)$ are disjoint and every vertex in $E(t)$ is adjacent to some vertex in $E(t-1)$. Thus, every vertex in $E(t)$ is the endpoint of a simple path of length t. If $E(T) \neq \emptyset$, then there is a simple path in G of length $T > dn(G)$, a contradiction.

This bound can be improved if $k > 1$. If $k > 1$, every vertex in $E(t)$ has at least two neighbors in $E(0) \cup \cdots \cup E(t-1)$. We will show that every vertex in $E(T)$ is in a simple path of length $T+1$ that includes a vertex in each of $E(0), E(1), \ldots,$ and $E(T)$. Let $v \in E(T)$. It is the endpoint of a simple path P of length T, $P = v_{i_0}v_{i_1} \ldots v_{i_T}$, $v_{i_T} = v$, with $v_{i_j} \in E(j)$. If any of v's neighbors are not on P, then clearly G has a simple path of length $T + 1$ containing v. If all of v's neighbors are on P, we construct a simple path of length $T + 1$ as follows. We will show by induction that every edge $v_{i_{j-1}}v_{i_j}$ in P is contained in a simple path of length $j + 1$ consisting solely of vertices in $E(0) \cup \cdots \cup E(j)$. The claim is clearly true for $j = 1$; v_{i_1} has another neighbor in $E(0)$ besides v_{i_0}, so there is a path of length two containing the edge $v_{i_0}v_{i_1}$. Assume the claim is true for all $j \leq p$. Consider the edge $v_{i_p}v_{i_{p+1}}$ in P, with $p < T$. If $v_{i_{p+1}}$ has a neighbor $w \in E(0) \cup \cdots \cup E(p+1)$ not on

P, then $v_{i_0}v_{i_1}\dots v_{i_p}v_{i_{p+1}}w$ is a simple path of length $p+2$ containing the edge $v_{i_p}v_{i_{p+1}}$. If all of the neighbors of $v_{i_{p+1}}$ in $E(0)\cup\cdots\cup E(p+1)$ are on P, then let v_{i_j} be a neighbor of $v_{i_{p+1}}$ on P, $j \neq p$. By our inductive assumption, we know there is a simple path P' of length $j+2$ containing the edge $v_{i_j}v_{i_{j+1}}$ consisting of vertices in $E(0)\cup\cdots E(j+1)$. Let $P' = P_1'v_{i_j}v_{i_{j+1}}P_2'$, where P_1' and P_2' are disjoint simple paths in $G[E(0)\cup\cdots\cup E(j+1)]$. Consider the path $P'' = P_1'v_{i_j}v_{i_{p+1}}v_{i_p}\dots v_{i_{j+1}}P_2'$. The vertices that were in P' are distinct, and the vertices v_{i_m} are in $E(m)$ for $j+1 \leq m \leq p+1$, so they are distinct from the vertices in P' and from each other, so the path is simple, of length $p+2$, and consists solely of vertices in $E(0)\cup\cdots\cup E(p+1)$. Thus, if $k>1$, $T+1 \leq dn(G)$, or $T \leq dn(G)-1$. $\square$

Remark: This result gives a sharp bound on transient length. Consider an irreversible $1-$conversion process in a path P_n of n vertices. If $V_1(0)$ consists of one of the endpoints of the path, then the process completes entering all vertices in state 1 at time $n-1 = dn(P_n)$. This can be generalized for any irreversible $k-$conversion process by starting with a path $P_{k+1} = v_1v_2\dots v_{k+1}$, adding k leaves to v_1 and $k-1$ leaves to each of $v_2,\dots,v_{k+1}$. If $V_1(0)$ consists of all of the leaves added to the path, then vertex v_i enters state 1 at time $t=i$. Therefore, the process reaches the situation of all vertices in state 1 at time $t=k+1$, and the longest path in this graph is of length $k+2$, involving all of the vertices on the path and at most two leaves.

Theorem 1. *For a fixed integer $k \geq 3$, IRREVERSIBLE $k-$CONVERSION SET is NP-complete.*

Proof. By Lemma 2, we know that the transient length of an irreversible threshold process is polynomial in the number of vertices, so IR$k-$CS is in NP. Our reduction will be from the INDEPENDENT SET problem, given below.

INDEPENDENT SET
Instance: A graph G and an integer m.
Question: Is there an independent set in G of size $\geq m$?

Fricke, Hedetniemi, and Jacobs [9], using a reduction from NOT-ALL-EQUAL-3SAT, show that for a fixed $k \geq 3$, INDEPENDENT SET is NP-complete for the class of $k-$regular, non-bipartite graphs (the problem is in P for the class of bipartite graphs).

Given an instance of INDEPENDENT SET for a $k-$regular, non-bipartite graph G: "Does G have an independent set of size $\geq m$?", we construct an instance of IR$k-$CS: "Does G have an irreversible $k-$conversion set of size $\leq |V(G)|-m$?" By Lemma 1, a vertex set is an irreversible $k-$conversion set in a $k-$regular graph if and only if its complement is independent. Therefore, there is an irreversible $k-$conversion set C of size $\leq |V(G)|-m$ if and only if there is an independent set S of size $\geq m$. So, there is a 'yes' answer to the instance of IR$k-$CS if and only if there is a 'yes' answer to the instance of INDEPENDENT SET. $\square$

The complexity of IR2$-$CS still remains open. We cannot use the above arguments, since it is trivial to find independent sets in $2-$regular graphs (i.e., vertex-disjoint unions of cycles). In addition, Li and Liu [16] have developed a polynomial algorithm for finding a minimum feedback vertex set in $3-$regular graphs, so we cannot use Proposition 1.

3 Irreversible k-Conversion Sets for Special Graphs

In this section, we give constructions for minimum $k-$conversion sets for some classes of graphs. For other classes, we present constructive upper bounds for the value of $C_k(G)$ and give proofs of lower bounds.

Proposition 2. *For the path and cycle on n vertices, $C_2(C_n) = \lceil \frac{n}{2} \rceil$.*

Proof. For simplicity, we will denote the vertices of P_n (C_n) in order as $v_1, \dots, v_n$. By Lemma 1, if S is a an irreversible $2-$conversion set in C_n, $V - S$ must be independent. Thus, $|V - S| \leq \lfloor \frac{n}{2} \rfloor$ and $|S| \geq \lceil \frac{n}{2} \rceil$. Picking the complement of an independent set of size $\lfloor \frac{n}{2} \rfloor$ in C_n will be an irreversible $2-$conversion set of size $\lceil \frac{n}{2} \rceil$.

For the path P_n, by Corollary 1, both v_1 and v_n must be in any irreversible $2-$conversion set S. The largest independent set in P_n not containing v_1 or v_n has size $\lfloor \frac{n-1}{2} \rfloor$, hence $|S| \geq n - \lfloor \frac{n-1}{2} \rfloor = \lceil \frac{n+1}{2} \rceil$. Constructing irreversible 2-conversion sets of size $\lceil \frac{n+1}{2} \rceil$ is straightforward. □

3.1 Complete Bipartite and Multipartite Graphs

In this section, we consider the complete multipartite graph $K_{p_1,p_2,\dots,p_m}$ consisting of m classes of vertices, with p_i vertices in the i^{th} class V_i, and with edges between all vertices in different classes. We will assume that $p_1 \geq p_2 \geq \dots \geq p_m$.

Proposition 3. *For the complete bipartite graph $K_{m,n}$, $m \geq n$:*

$$C_k(K_{m,n}) = \begin{cases} k & \text{if } m, n \geq k \\ m & \text{if } m \geq k, n < k \\ m+n & \text{if } m, n < k \end{cases}$$

Proof. Let V_1 and V_2 be the two partite sets of V, with $|V_1| = m \geq n = |V_2|$. Any subset of V_1 of size k will be an irreversible $k-$conversion set if $m, n \geq k$. Clearly k is the lower bound for the size of any irreversible $k-$conversion set. If $m \geq k$ and $n < k$, V_1 is an irreversible $k-$conversion set. There are not enough vertices in V_2 to change the state of any vertex in V_1, so all of V_1 must be in state 1 at time $t = 0$. If both partite sets are smaller than k, then V is the only $k-$conversion set. □

Consider the complete $m-$partite graph $K_{p_1,p_2,\dots,p_m}$ with $p_1 \geq p_2 \geq \dots \geq p_m$. For any subset $J \subseteq [m] = \{1, \dots, m\}$, let $V_J = \bigcup_{j \in J} V_j$. A union of partite sets V_J has size *just above* k if $|V_J| \geq k$ and if $V_{J'}$ is created by removing the partite set of highest index from V_J ($J' = J - \{\max\{j : j \in J\}\}$), then $|V_{J'}| < k$.

Proposition 4. *Let G be a complete $m-$partite graph $K_{p_1,p_2,\dots,p_m}$ with $p_1 \geq p_2 \geq \dots \geq p_m$. Suppose that there is a union $V_J \subseteq V$ of partite sets such that:*

(1) V_J has size just above k, and

(2) $n - p_s \geq k$, where s is the largest element of J,

Then $C_k(G) = k$. Otherwise, $C_k(G) = \min\{|V_J| : V_J$ is a union of partite sets of size just above $k\}$.

Proof. Let V_J be a union of partite sets of V satisfying conditions (1) and (2). If V_J has size exactly k, then clearly V_J is an irreversible $k-$conversion set for G. If V_J does not have size exactly k, let $J' = J - \{s\}$, where s is the largest element of J, and let $V_1(0)$ consist of $V_{J'}$ and $k - |V_{J'}|$ vertices in V_s. At time $t = 1$, all vertices will be in state 1 except for the state 0 vertices in V_s, which will enter state 1 at time $t = 2$ by condition (2). Thus, $C_k(G) \leq k$. Since $C_k(G)$ must be $\geq k$, equality is proven.

Assume that no union V_J of partite sets satisfies conditions (1) and (2). Let S be a minimum irreversible $k-$conversion set for G, and let V_J be the smallest union of partite sets that contains S. At time $t = 1$, every vertex in $V - V_J$ will be in state 1. If no $V_j, j \in J$, contains elements of $V-S$, then V_J has size exactly k and the result is straightforward. Thus, let us assume that some V_j, $j \in J$, contains elements of $V-S$. We will reach a contradiction. We may assume that at most one partite set $V_i \subseteq V_J$ has vertices in $V-S$. If two partite sets $V_i, V_j \subseteq V_J$ contain vertices in $V-S$, let $S_i = S \cap V_i$ and $S_j = S \cap V_j$. Removing $\min\{p_i - |S_i|, |S_j|\}$ vertices in S from V_j and adding the same number of vertices from V_i to S gives a new irreversible $k-$conversion set S' for G. This is because nothing changes in the other partite sets during the threshold process with $V_1(0) = S'$ as compared to the one with $V_1(0) = S$, because V_j will be in state 1 at time $t = 1$, and because a vertex in V_i will have at least as many state 1 neighbors in $V - V_i$ at time $t = 1$ as it did in the original process.

We can also assume that the partite set in V_J containing members of $V-S$ is the partite set V_s in V_J of highest index. To see this, let V_j be the partite set in V_J containing vertices in $V-S$. If $p_s > |(V-S) \cap V_j|$, consider the new set S' which removes $|(V-S) \cap V_j|$ of the vertices in V_s from S and includes all of V_j. Clearly, S' is an irreversible $k-$conversion set. If $p_s \leq |(V-S) \cap V_j|$, then consider the new set S' that removes all of the vertices in V_s from S and includes p_s more vertices from V_j. We let $J' = J - \{s\}$ and repeat the process of removing vertices in S from the partite set of highest index in J' and adding vertices from V_j to S until either no partite set contains vertices in S and $V-S$, or the only partite set that contains vertices in S and $V-S$ is the partite set of highest index in $V_{J'}$.

Suppose that V_s is the partite set of highest index in V_J and is the only partite set in V_J that contains vertices in $V-S$. If $n - p_s < k$, the 0 vertices in $V - s$ will never switch to 1 and this contradicts s being an irreversible k-conversion set. Suppose $n - p_s \geq k$. Then V_J does not have size just above k. Clearly, $|V_J| \geq k$ since S is an irreversible k-conversion set. This implies that $|V_{J'}| \geq k$ where $J' = J - \{s\}$. But now $V_{J'}$ is an irreversible k-conversion set. This is because if $V_1(0) = V_{J'}$, at time $t = 1$ all vertices in V_s become 1 and now $S \subset V_1(1)$. This contradicts S being a minimum irreversible k-conversion set since $V_{J'}$ is a proper subset of S. $\square$

Proposition 5. *Let G be the complete $m-$partite graph $K_{p_1,\ldots,p_m}$ with $p_2 + \cdots + p_m \geq k$. Then $C_k(G) = k$.*

Proof. Let S be any subset of $V_2 \cup \cdots \cup V_m$ of size k, and let $s_i = |S \cap V_i|$ for $2 \leq i \leq m$. We claim that S is an irreversible $k-$conversion set. Since every vertex in V_1 has k neighbors in state 1 in $V_2 \cup \cdots \cup V_m$, $V_1 \subset V_1(1)$. If $v \in V_0(1) \cap V_i$, then at time $t = 2$, v has at least $k - s_i + p_1$ neighbors in state 1 in $V - V_i$. Since $p_1 \geq p_i \geq s_i$, v has at least k neighbors in $V_1(1)$, so $v \in V_1(2)$. Hence, S is an irreversible $k-$conversion set. $\square$

Remark: Let G be the complete $m-$partite graph $K_{p_1,\ldots,p_m}$ with $p_1 \geq p_2 \geq \cdots \geq p_m$. If $n \geq 2k$ and $p_2 + \cdots + p_m < k$, then $C_k(G) = p_1$.

3.2 Trees

For a tree T, let L_T denote the set of leaves of T, i.e., vertices of degree 1. Clearly, L_T must be contained in any irreversible $k-$conversion set for $k > 1$, and in some circumstances, L_T is a minimum irreversible k-conversion set. We will use the term *internal vertex* in a tree to be any vertex that is not a leaf and use $T[W]$ to denote the subtree induced by vertices in set W.

Proposition 6. *If T is a tree with every internal vertex of degree $> k$ for $k > 1$, then $C_k(T) = |L_T|$.*

Proof. Let v be an internal vertex of T, and let $D = ecc(v)$, where the *eccentricity* $ecc(v)$ is the greatest distance between v and any other vertex. The result follows by proving the following statement by induction on t: "At time $t \geq 0$, in a $k-$threshold process with $V_1(0) = L_T$, $\{w : d(v,w) \geq D - t\} \subseteq V_1(t)$." $\square$

For a graph $G = (V, E)$, a set $S \subseteq V$ is a *vertex cover* for G if for every edge $uv \in E$, $u \in S$ or $v \in S$. It is easily shown that S is a vertex cover if and only if $V - S$ is independent.

Proposition 7. *Let T be a tree with every internal vertex of degree $k > 1$. Then S is an irreversible $k-$conversion set if and only if $S = C \cup L_T$, where C is a vertex cover of $T[V - L_T]$.*

Proof. If S is an irreversible $k-$conversion set for T, it must contain L_T, and $V - S$ must be independent, since if two 0 vertices shared an edge, they would never become 1. Hence, $S = C \cup L_T$ for some vertex cover C of $T[V - L_T]$. It remains to show that for any vertex cover C of $T[V - L_T]$, $S = C \cup L_T$ is an irreversible $k-$conversion set. Since $V - S$ is independent, all of the vertices in $V - S$ will become 1 at time $t = 1$. □

To extend these results to all trees, we make a few simplifying assumptions. We will restrict our analysis to trees with all internal vertices of degree $\geq k$. If there is an internal vertex v with $\deg(v) < k$, it will have to be in any irreversible $k-$conversion set. Thus, to every neighbor of v, v acts exactly like a leaf, so we can break T into $\deg(v)$ subtrees with v as a leaf in each subtree.

We define a *$k-$region* of T to be a maximal connected subset of vertices of degree k. We can partition the vertex set of a tree into leaves, $k-$regions, and non-$k-$regions (maximal connected subsets of vertices of degree $> k$). For a $k-$region S, define the *outer boundary* of S (denoted $B(S)$) to be the set of vertices not in S that have a neighbor in S (or, $N[S] - S$) and the *inner boundary* of S (denoted $b(S)$) to be the set of vertices in S adjacent to vertices not in S (or, $N[B(S)] \cap S$).

Proposition 8. *Let T be a tree with every internal vertex of degree $\geq k$. If C is a vertex cover for every subtree $T[S \cup B(S)]$, where S is a $k-$region of T, then $C \cup L_T$ is an irreversible $k-$conversion set for T.*

Proof. If C is a vertex cover for every subtree $T[B(S) \cup S]$, where S is a $k-$region, then after one time step, every vertex in every $k-$region will be 1, since the set of 0 vertices in a $k-$region S will be independent. At time $t = 1$, every non-$k-$region will be surrounded by vertices in state 1, and will proceed to turn to 1 as in the proof of Proposition 6. □

If C is an irreversible $k-$conversion set for a tree T where every internal vertex has degree $\geq k$, then $S - C$ must be independent for every $k-$region S. This is guaranteed by Proposition 8, but there is no guarantee that the irreversible $k-$conversion set described in Proposition 8 is minimum. If v is an inner boundary vertex of some $k-$region S, and w is its neighbor on the outer boundary of S, then there is no need for C to cover the edge vw, as long as k other neighbors of w eventually enter state 1.

We now present an algorithm to compute $C_k(T)$ that runs in $O(n)$ time, where n is the number of vertices in the tree. For simplicity, we will present the algorithm for $k = 2$ and then discuss how to implement it for larger k.

For a tree T rooted at a vertex r, a set S of vertices is an *almost irreversible $2-$conversion set* (a-I2CS) if when an irreversible $2-$conversion process is started with the vertices in S in state 1, then all of the vertices in $T - \{r\}$ enter state 1. The root vertex r may be in S to convert other vertices in T, but it is not necessary for r to be in state 1 when the process reaches its fixed point. However, if the root vertex r of T has degree 2 or greater, then any a-I2CS S for T is also an irreversible $2-$conversion set. If r is not in S, then r will enter state 1 after two of its neighbors enter state 1.

We define the composition of a tree T_1 with a tree T_2 and the *composition of tree-subset pairs* (T_1, S_1) and (T_2, S_2) with S_i being an a-I2CS. We use the rule of composition that joins a tree T_1 rooted at r_1 and a tree T_2 rooted at r_2 (with $V(T_1) \cap V(T_2) = \emptyset$) by adding the edge r_1r_2 and specifying r_1 as the root of the resulting larger tree T. Let T_1 and T_2 be two trees rooted at vertices r_1 and r_2, respectively. Also, let S_1 and S_2 be a-I2CS's for T_1 and T_2, respectively. If T is the composition of T_1 with T_2, then consider the set $S' = S_1 \cup S_2$. S' might not be an a-I2CS for T, since if r_2 is not converted to state 1 by S_2 in T_2, then r_2 has at most one neighbor in T_2. If r_2 is not an isolated vertex in T_2 and r_1 is converted to state 1 in T_1 by S_1, then r_2 will eventually have two neighbors in T in state 1, so r_2 will enter state 1 as well. If not, then either r_1 or r_2 must be added to S' for it to be an a-I2CS. Let S be the resulting a-I2CS for T. We will use the notation $(T, S) = (T_1, S_1) \circ (T_2, S_2)$ to denote this composition of two trees and their associated a-I2CS's.

To construct an algorithm to compute $C_2(T)$ for any tree T, we characterize the class of possible tree-subset pairs (T, S) that can occur in a tree T with root vertex r, where S is an a-I2CS of T. For this problem, there are four classes of tree-subset pairs:

$$\begin{aligned}
[0] &= \{(T,S) | r \in S\} \\
[1] &= \{(T,S) | r \notin S, \deg(r) \geq 2\} \\
[2] &= \{(T,S) | r \notin S, \deg(r) = 1\} \\
[3] &= \{(T,S) | r \notin S, \deg(r) = 0\}
\end{aligned}$$

If $(T, S) = (T_1, S_1) \circ (T_2, S_2)$, we can determine the class of (T, S) if (T_1, S_1) is of class $[i]$ and (T_2, S_2) is of class $[j]$ for each pair (i, j), $0 \leq i, j \leq 3$. The classes of (T, S) resulting from each composition are summarized in Figure 1. The verification of the class of (T, S) for each pair (i, j) is straightforward. Note that in some cases it is necessary to add copies of r_1 or r_2 to S' in order to convert r_2.

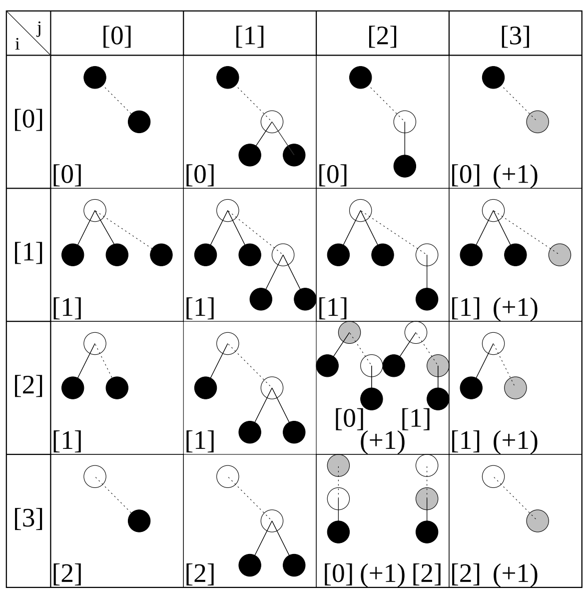

Figure 1: The number in brackets in the $([i],[j])^{th}$ entry in this table gives the class of (T, S) that results from composing (T_1, S_1) of class $[i]$ with (T_2, S_2) of class $[j]$. Examples are given for each pair $([i],[j])$. The black vertices denote the vertices that appear in S. A dotted edge represents the edge joining the roots of T_1 and T_2. If it was necessary to add a copy of r_1 or r_2 to S to convert r_2, the added vertex is shown in grey, and the number of vertices that are added to S is noted in parentheses. In some cases, the class of the resulting composition is different depending on whether r_1 or r_2 is added to S.

Using Figure 1, we can write out a system of recurrence relations that capture how we can obtain a tree-subset pair of class $[j]$ after composition. It is impossible to compose two tree-subset pairs to get a tree-subset pair (T, S) of class [3], because when two trees are composed, the root vertex has degree greater than zero. The superscripts ^{1} and ^{2} on some of the compositions denote that r_1 or r_2 has been added to S to make an a-I2CS for T.

$$\begin{aligned}
[0] &= [0]\circ[0] \cup [0]\circ[1] \cup [0]\circ[2] \cup [0]\circ[3]^2 \cup [2]\circ[2]^1 \cup [3]\circ[2]^1 &(1)\\
[1] &= [1]\circ[0] \cup [1]\circ[1] \cup [1]\circ[2] \cup [1]\circ[3]^2 &(2)\\
&\quad \cup[2]\circ[0] \cup [2]\circ[1] \cup [2]\circ[2]^2 \cup [2]\circ[3]^2 &\\
[2] &= [3]\circ[0] \cup [3]\circ[1] \cup [3]\circ[2]^2 \cup [3]\circ[3]^2 &(3)
\end{aligned}$$

For example, to see recurrence (2), we note that if $(T, S) = (T_1, S_1) \circ (T_2, S_2)$ is of class [1], then (T_1, S_1) must be of class [1] or [2] and (T_2, S_2) can be of any class. However, if (T_2, S_2) is of class [3] or if both (T_1, S_1) and (T_2, S_2) are of class [2], then r_2 must be added to S to produce an a-I2CS.

Given a tree T of order n, we root it at an arbitrary vertex that we label v_1. We then label the remaining vertices $v_2, \ldots, v_n$ using the well-known depth first search procedure so that if v_j is v_i's parent, then $i > j$. For each vertex v_i, let $Parent[i]$ equal the subscript of the label of the parent of v_i (assume that $Parent[1] = 0$).

The algorithm will build the tree T one edge at a time. At step i, the algorithm builds a graph G_i with vertex set V and designates a root for each maximal connected subtree of G_i. G_i is obtained from G_{i-1} by composing the subtree rooted at $v_{Parent[n-i]}$ with the subtree rooted at v_{n-i}.

In graph G_i there will be a $4-$place class vector associated to each root vertex. If v_k is the root of a subtree T_k in G_i, then the j^{th} entry of its class vector at step i equals the size of the smallest a-I2CS S_k for T_k such that (T_k, S_k) is of class $[j]$. The size can be ∞ if there is no such a-I2CS.

G_0 consists of n isolated vertices, each the root of a one vertex tree. The class vector associated with each vertex v_k in G_0 is $[1, \infty, \infty, 0]$, where '∞' is the entry for class [1] and [2], as the degree of an isolated vertex is zero, so it is impossible for a tree-subset pair to be of class [1] or [2] at the start of the algorithm. For classes [0] and [3], we associate the a-I2CS's $\{v_k\}$ and $\emptyset$ of sizes 1 and 0, respectively.

At step i of the algorithm, the only class vector that will change will be the one associated with the vertex $v_{Parent[n-i]}$. This is updated using the table in Figure 1 and the recurrences (1) - (3).

G_{n-1} is the tree T rooted at v_1. Therefore, the j^{th} entry of the class vector associated to v_1 is the size of the smallest a-I2CS S of T such that (T, S) is of class $[j]$. If (T, S) is of class [0] or [1], then S is also an irreversible $2-$conversion set, but not if (T, S) is of class [2] or [3], since v_1 is not converted by S. Thus, $C_2(T)$ equals the minimum of the first two entries of the class vector associated to v_1.

We now have all the ingredients for an algorithm to compute $C_2(T)$ for a tree T. The input to the algorithm is an $n-$place vector $Parent[1 \ldots n]$ for T, whose i^{th} entry is $Parent[i]$, as defined above. The output of the algorithm is the value of $C_2(T)$. Every vertex v_i has an associated class vector $Class[i]$ whose j^{th} entry is denoted $Class[i, j]$. Every class vector is initialized with the vector $Class[i] = [1, \infty, \infty, 0]$.

At step i, where i runs from 0 to $n-2$, the tree rooted at $v_{Parent[n-i]}$ is composed with the tree rooted at v_{n-i} producing a tree T'. The vector $Class[Parent[n-i]]$ is updated by a procedure called `combine` that is derived directly from the recurrences (1) - (3) summarized in Figure 1. $Class[Parent[n-i], j]$ equals the size of the smallest a-I2CS S' for T' such that the pair (T', S') is of class $[j]$. For example, let T' be the composition of a tree T_1 rooted at v_a with a tree T_2 rooted at v_b. Let S' be a minimum a-I2CS for T' such that (T', S') is of class [2]. By relation (3), if (T_1, S_1) is of class [2], then S' is the union of the smallest a-I2CS S_1 for T_1 such that (T_1, S_1) is of class [3] and the smallest a-I2CS S_2 for T_2 such that (T_2, S_2) is of class [0] or [1], or the smallest a-I2CS S_2 for T_2 such that (T_2, S_2) is of class [2] or [3], plus the vertex v_b. Since $Class[a, j]$ equals the size of the smallest a-I2CS S_1 for T_1 such that (T_1, S_1) is of class $[j]$, and $Class[b, j]$ is similarly defined for T_2, then $Class[a, 2]$ will be updated to equal the minimum of $Class[a, 3] + Class[b, 0]$, $Class[a, 3] + Class[b, 1]$, $Class[a, 3] + Class[b, 2] + 1$, and $Class[a, 3] + Class[b, 3] + 1$. The other values of $Class[a, j]$ are updated similarly.

After step $n - 1$, all of the edges of the tree T rooted at v_1 have been included. Therefore, $Class[1, j]$ equals the size of the smallest a-I2CS S for T such that (T, S) is of class $[j]$. We want S to be an irreversible 2−conversion set for T, so $C_2(T)$ will be the minimum of $Class[1, 0]$ and $Class[1, 1]$. An example of the algorithm is illustrated in Figure 2.

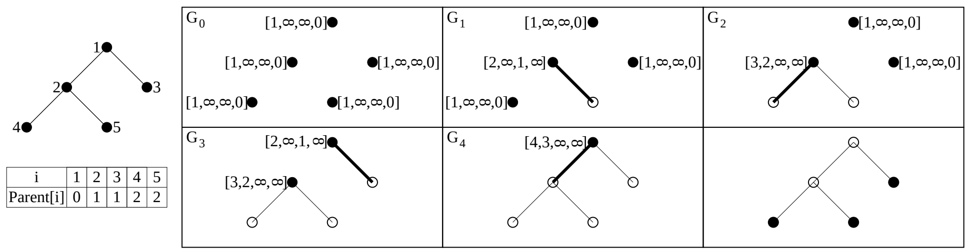

Figure 2: An illustration of the algorithm to compute $C_2(T)$. At step i, the root vertex of each maximal connected subtree in G_i is denoted with a filled in circle, and the class vector associated to each root vertex at step i is noted next to the vertex. The bold edge denotes the edge added by the composition of the subtree rooted at $v_{Parent[n-i]}$ with the subtree rooted at v_{n-i}. The filled in circles in the lower right tree give the minimum irreversible 2−conversion set S for T. Note that (T, S) is of class [1], as the root vertex is not in S. Also, the minimum of the first two entries of the class vector associated to the root vertex of $G_4 = T$ is the entry associated with class [1].

Pseudo-code for the algorithm is given below.

```
                    procedure C_2(Parent);

begin

for i:=1 to n do
    initialize Class[i,0...3] to [1,∞,∞,0];
for j:=0 to n-2 do
    p:=Parent[n-j];
    combine(Class,p,n-j);
C_2(T) := min {Class[1,0], Class[1,1]};
output C_2(T);
end(C_2);

                    procedure combine(Class,a,b);

begin

Class'[0]:= min {Class[a,0]+Class[b,0], Class[a,0]+Class[b,1],
                 Class[a,0]+Class[b,2], Class[a,0]+Class[b,3]+1,
                 Class[a,2]+Class[b,2]+1, Class[a,3]+Class[b,2]+1};
```

```
Class′[1]:= min {Class[a,1]+Class[b,0], Class[a,1]+Class[b,1],
                 Class[a,1]+Class[b,2], Class[a,1]+Class[b,3]+1,
                 Class[a,2]+Class[b,0], Class[a,2]+Class[b,1],
                 Class[a,2]+Class[b,2]+1, Class[a,2]+Class[b,3]+1};
Class′[2]:= min {Class[a,3]+Class[b,0], Class[a,3]+Class[b,1],
                 Class[a,3]+Class[b,2]+1, Class[a,3]+Class[b,3]+1};
Class′[3]:= ∞;
for j:=0 to 3 do
    Class[p,j] = Class′[j];
end(combine);
```

Since the procedure `combine` takes $O(1)$ time to compute, it is clear that the procedure C_2 runs in $O(n)$ time. To summarize, we have the following theorem.

Theorem 2. *For a tree T of order n, the procedure C_2*`(Parent)` *computes $C_2(T)$ in $O(n)$ time.*

To extend this algorithm to compute $C_k(T)$ for $k \geq 3$, one simply defines additional classes of trees: one where the root is in S, one where the root is not in S but its degree is at least k, and k classes where the root is not in S and the degree is j, $0 \leq j < k$. For a fixed k, running the `combine` procedure takes $O(1)$ time, and the procedure is executed $n-1$ times, producing an $O(n)$ algorithm.

3.3 Grids

Let $G_{m,n} = P_m \times P_n$ be the standard $m \times n$ grid. Here, the vertices will be denoted $v_{i,j}$, $1 \leq i \leq m$, $1 \leq j \leq n$, and there is an edge between $v_{i,j}$ and $v_{k,l}$ if and only if $|i-k| + |j-l| = 1$. We will frequently refer to the *parity* of a vertex $v_{i,j}$ which is identical to the parity (even or odd) of $i+j$. For simplicity, we will refer to even vertices or odd vertices, instead of vertices of even (odd) parity. We also introduce the *circulant grid* $C_{m,n} = P_m \times C_n$, which is the standard grid with the edge $v_{i,1}v_{i,n}$ added for every i, $1 \leq i \leq m$ and the *toroidal grid* $T_{m,n} = C_m \times C_n$ which is the circulant grid with the edge $v_{1,j}v_{m,j}$ added for every j, $1 \leq j \leq n$. We will refer to the rows and columns of a grid by $R^i = \{v_{i,1}, \ldots, v_{i,n}\}$ and $C^j = \{v_{1,j}, \ldots, v_{m,j}\}$.

The toroidal grid is $4-$regular, which allows us to take advantage of some of our earlier results on conversion sets in regular graphs.

Proposition 9. *For the toroidal grid graph $T_{m,n}$:*

$$C_4(T_{m,n}) = \begin{cases} \max\{n\lceil m/2\rceil, m\lceil n/2\rceil\} & \text{if } m \text{ or } n \text{ is odd} \\ \frac{mn}{2} & \text{if } m \text{ and } n \text{ are even} \end{cases}$$

Proof. By Lemma 1, S is an irreversible $4-$conversion set in a $4-$regular graph if and only if $V-S$ is independent. $S \cap R^i$ must be a vertex cover for each R^i, and $S \cap C^j$ must be a vertex cover for each C^j, so each row must contain at least $\lceil n/2\rceil$ state 1 vertices and each column must contain at least $\lceil m/2\rceil$ state 1 vertices. Hence,

$$C_4(T_{m,n}) \quad \geq \quad \max\{n\lceil m/2\rceil, m\lceil n/2\rceil\}. \tag{4}$$

We can construct a $4-$conversion set as follows. Given a row R^i of a grid and a set of vertices T in that row, T' is a *left (right) cyclic shift* of T in row $R^{i'}$ if $v_{i,j} \in T \Leftrightarrow v_{i',j-1}(v_{i',j+1}) \in T'$ for $2 \leq j \leq n$ $(1 \leq j \leq n-1)$ and $v_{i,n}(v_{i,1}) \in T \Leftrightarrow v_{i',1}(v_{i',n}) \in T'$. Assume that n is odd, and let C be a vertex cover for R^1 of size $\lceil n/2\rceil$. If m is even, then let C' be the right cyclic shift of C in

R^2, and then let C'' be the left cyclic shift of C' in R^3, and so on. If m is odd, then assume that $n \leq m$. For the first $n+2$ rows (or n rows if $n = m$), generate a vertex cover for R^j via a right cyclic shift of the cover for R^{j-1} (where C is the vertex cover for R^1). If $m = n$, then the cover for R^m is the left cyclic shift of C, and if $m = n+2$, then the cover for R^m is the right cyclic shift of C. If $m > n+2$, in R^{n+3}, let C' be the left cyclic shift of the vertex cover in R^{n+2}, and in R^{n+4}, let C'' be the right cyclic shift of C', and continue to alternate cyclic shifts to the left and right so that the cover for row R^m is the right cyclic shift of C.

Let S be the union of all of these vertex covers. Since we have assumed that n is odd and that if m is odd, then $n \leq m$, $m\lceil n/2 \rceil \geq n\lceil m/2 \rceil$. We claim that S forms an irreversible $4-$conversion set of size $m\lceil n/2 \rceil = \max\{n\lceil m/2 \rceil, m\lceil n/2 \rceil\}$, which together with the lower bound in Equation (4) gives us the desired result for the case when n is odd. No vertex in $V - S$ has a neighbor in $V - S$ in its row, since S is a vertex cover for each row. Also, no vertex in $V - S$ has a neighbor in $V - S$ in its column, since both of its neighbors in the row are in S, so when the vertex cover in its row is cyclically shifted, a vertex in $V - S$ will have a neighbor in S in the row above and below it. Finally, it is easy to see that either the first or the last vertex in each row or columns is in S. Hence, $V - S$ is independent.

If both m and n are even, then simply start with a vertex cover of row R^1 by $n/2$ vertices and generate each subsequent row's covering by a right cyclic shift. The proof that $V - S$ is independent is straightforward. $\square$

Proposition 10. *For the standard grid graph $G_{m,n}$:*

$$C_4(G_{m,n}) = 2m + 2n - 4 + \lfloor (m-2)(n-2)/2 \rfloor.$$

Proof. Because all of the vertices on the border of the grid have degree less than 4, they must be in any $4-$conversion set S. The vertices in the interior of the grid have degree 4, and no two vertices in $V - S$ can be neighbors. Hence, $|V - S| \leq \lceil (m-2)(n-2)/2 \rceil$, corresponding to the set of even non-border vertices or the set of odd non-border vertices. Let $V - S$ be the larger of these two sets. It can be shown (as in Corollary 1) that every vertex within distance d of any border vertex will be in state 1 for all $t \geq d$, as long as $V_0(t)$ remains independent for all $t \geq 0$, which is guaranteed by the independence of $V_0(0)$. Thus, the smallest irreversible $4-$conversion set for $G_{m,n}$ has size $2m + 2n - 4 + \lfloor (m-2)(n-2)/2 \rfloor$. $\square$

The following results on irreversible $3-$conversion sets in grids are derived from work on "irreversible dynamos" by Flocchini, et al. ([5], [6]), and from work on minimum feedback sets in grids by Luccio [17].

Proposition 11. *([5], [6], [17]) For the toroidal grid graph $T_{m,n}$ and the standard grid graph $G_{m,n}$, assume that $m \leq n$. Then:*

1. $\frac{(m-1)(n-1)+1}{3} \leq C_3(G_{m,n}) \leq \frac{(m-1)(n-1)}{3} + \frac{3m+2n-3}{4} + 5$
2. $\frac{mn+2}{3} \leq C_3(T_{m,n}) \leq \frac{mn}{3} + \frac{23m+13n-5}{12}$

The results in this proposition come from the work of Flocchini, et al. and Luccio, where they provided constructions of strong dynamos and feedback vertex sets in toroidal grid graphs, which correspond to irreversible $3-$conversion sets. With some small modifications, their results can also be modified for standard grid graphs (which are no longer $4-$regular, hence requiring some additional border vertices in S). Some sample constructions are given in Figure 3.

We close this section with some results on $2 \times n$ and $3 \times n$ grids.

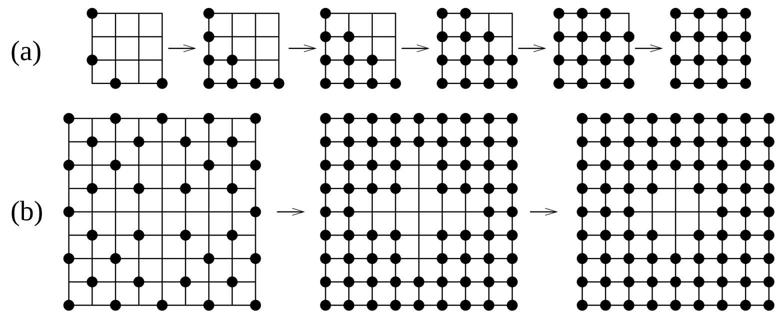

Figure 3: Some constructions of conversion sets based on work from Flocchini et al., and Luccio. (a) A minimum irreversible 2−conversion set for $G_{4,4}$. (b) A minimum irreversible 3−conversion set for $G_{9,9}$.

Theorem 3. *For the toroidal grid graph $T_{m,n}$, the standard grid graph $G_{m,n}$, and the circulant grid graph $C_{m,n}$:*

$$\begin{aligned} C_3(T_{3,n}) &= n+1 \\ C_3(G_{3,n}) &= \begin{cases} (3n+1)/2 & \textit{if } n \textit{ is odd} \\ (3n+2)/2 & \textit{if } n \textit{ is even} \end{cases} \\ C_3(C_{3,n}) &= \begin{cases} (3n+1)/2 & \textit{if } n \textit{ is odd} \\ 3n/2 & \textit{if } n \textit{ is even} \end{cases} \end{aligned}$$

Proof. For the toroidal grid graph $T_{3,n}$, by Proposition 1, S is an irreversible 3−conversion set if and only if $V-S$ is cycle free. Every column must contain a vertex in S, so $|S| \geq n$. However, if every column has exactly one vertex in S, then every column has two adjacent vertices in $V-S$. Therefore, in every column C^j, some vertex in $(V-S)\cap C^j$ has a neighbor in $(V-S)\cap C^{j-1}$ and some vertex in $(V-S)\cap C^j$ has a neighbor in $(V-S)\cap C^{j+1}$. Therefore, the subgraph induced by $V-S$ contains a cycle which contains at least one vertex from each column in the grid, which is a contradiction. Hence, $C_3(T_{3,n}) \geq n+1$. Let S be $v_{1,1}$ and all of the even vertices in rows $R^2 \cup R^3$. $V-S$ is cycle-free, so S is an irreversible 3−conversion set of size $n+1$.

For the standard grid graph $G_{3,n}$, if S is an irreversible 3−conversion set, S must contain the four corners, since they have degree two. In addition, S must be a vertex cover of R^1 and R^3, since all of the non-corner border vertices have degree three. Finally, there cannot be a path in $V-S$ connecting two border vertices in $V-S$, as none of the vertices along such a path would ever enter state 1, since all of the non-border vertices would always have two state 0 neighbors, and the two border vertices would always have a state 0 neighbor. So, every path between two border vertices in $V-S$ must contain a vertex in R^2 in S.

If a column C^j contains two border vertices in $V-S$, then $v_{2,j}$ must be in S. In addition, we know that $v_{1,j-1}$, $v_{1,j+1}$, $v_{3,j-1}$, and $v_{3,j+1}$ must be in S, since there cannot be two consecutive border vertices in $V-S$.

Let $C^i, C^{i+1}, \ldots, C^j$ be a maximal collection of consecutive columns where each column has two border vertices in S. We can assume that no vertex $v_{2,j'}$, $i \leq j' \leq j$, is in S. To see this, let S be a minimum irreversible 3−conversion set that contains $v_{2,j'}$ for some j', $i \leq j' \leq j$. $S - \{v_{2,j'}\} \cup \{v_{2,i-1}\}$ is also a minimum irreversible 3−conversion set, because any shortest path between two border vertices one of which is in a column C_u for $u < i$ and that contains $v_{2,j'}$ also contains $v_{2,i-1}$. Similarly, $S - \{v_{2,j'}\} \cup \{v_{2,j+1}\}$ is a minimum irreversible 3−conversion set. If either $v_{2,i-1}$ or $v_{2,j+1}$ was already in S, then S was not minimal, since $v_{2,j'}$ could be removed.

Let $C^i, C^{i+1}, \ldots, C^j$ be a maximal collection of consecutive columns where each column has one border vertex in S. Since $V - S$ must be independent in the border vertices, the border vertices in S will have to alternate rows in consecutive columns. In addition, columns C^{i-1} and C^{j+1} must have two border vertices in S, since if $v_{1,i}$ is in $V - S$, $v_{1,i-1}$ must be in S, and since the collection of columns is maximal, $v_{3,i-1}$ must also be in S. For every consecutive pair of columns C^a and C^{a+1} in this collection, at least one of $v_{2,a}$ and $v_{2,a+1}$ must be in S, or else there would be a path of length four between two border vertices in $(V - S) \cap (C^a \cup C^{a+1})$.

Therefore, every column that contains only one vertex in S has two neighboring columns that each contain two vertices in S. If S is minimal, we can assume that no column contains three vertices in S. Thus, if c_i is the number of columns with i vertices in S, then $c_0 = 0$, $c_3 = 0$, and $|S| = c_1+2c_2$. If n is even, then $c_2 \geq n/2+1, c_1 \leq n/2-1$, and $|S| = c_1+2c_2 \geq n/2-1+2(n/2+1) = (3n+2)/2$. If n is odd, then $c_2 \geq (n+1)/2, c_1 \leq (n-1)/2$, and $|S| \geq (n-1)/2 + 2[(n+1)/2] = (3n+1)/2$. If n is odd, letting S be the even vertices of $G_{3,n}$ gives an irreversible 3−conversion set of size $(3n+1)/2$. If n is even, letting S be the even vertices of $G_{3,n}$ except $v_{2,n}$, and including $v_{1,n}$ and $v_{3,n}$ gives an irreversible 3−conversion set of size $(3n+2)/2$.

For the circulant grid graph $C_{3,n}$, we apply the same argument as above, that every column with one vertex in S has two neighboring columns each containing two vertices of S. Again, $c_0 = 0$, $c_3 = 0$, and $|S| = c_1 + 2c_2$. If n is even, then $c_2 \geq n/2, c_1 \leq n/2$, so $|S| \geq n/2 + 2(n/2) = 3n/2$. If n is odd, then $c_2 \geq (n+1)/2, c_1 \leq (n-1)/2$, so $|S| \geq (n-1)/2 + 2[(n+1)/2] = (3n+1)/2$. In both cases, letting S consist of the even vertices in $C_{3,n}$ gives an irreversible 3−conversion set of the desired size. □

Some of the conversion sets constructed in the proofs above are shown in Figure 4.

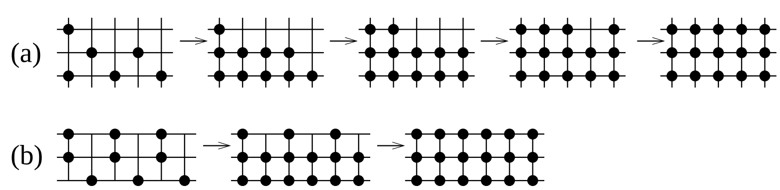

Figure 4: Minimum irreversible conversion sets (c.s.) for various $2 \times n$ and $3 \times n$ grids. (a) An irreversible 3−c.s. for $T_{3,5}$. (b) An irreversible 3−c.s. for $C_{3,6}$. All edges that extend beyond the border of the grid are assumed to wrap around to the opposite side.

4 Lower Bounds on $C_k(G)$

For the irreversible threshold process, we have the trivial lower bound $C_k(G) \geq k$, and there are several graphs that achieve this bound.

The following lower bound for irreversible conversion set sizes for r−regular graphs is obtained using a method suggested by Eli Berger [2].

Proposition 12. *If G is an r−regular graph with n vertices, $C_k(G) \geq (1 - \frac{r}{2k})n$ for $k \leq r < 2k$.*

Proof. We define a slightly different method for updating an irreversible k−threshold network. Instead of switching every 0 vertex with k or more 1 neighbors simultaneously, at each time step, we pick any 0 vertex with k or more 1 neighbors and switch it to 1, and repeat this process until no more vertices can be switched. If S is an irreversible $k-$conversion set under the normal update rule and S_t is the set of vertices that enter state 1 at time t, then under the new update rule, one can switch all of the vertices in S_1 one at a time, then the vertices in S_2, and so on. Therefore, if S is an irreversible $k-$conversion set under the normal update rule, S will also be an irreversible $k-$conversion set under this new update rule.

Define the *energy* of a network to be the number of edges between vertices of opposite value. Let $E(t)$ be the *energy* of the network at time t. For every vertex that we switch from 0 to 1, we lose an amount of energy of size at least k and create an amount of energy of size at most $r-k$. Hence, switching a vertex from 0 to 1 reduces the amount of energy in the network by at least $2k-r$, so $E(t+1)+(2k-r) \le E(t)$. If S is an irreversible $k-$conversion set, then $E(n-|S|)=0$. Therefore, $(n-|S|)(2k-r) \le E(0)$, or $n \le |S| + \frac{E(0)}{2k-r}$. However, $E(0)$ is at most $r|S|$, so $n \le |S| + \frac{r|S|}{2k-r} = \frac{2k|S|}{2k-r}$, and $|S| \ge (1-\frac{r}{2k})n$. □

If we consider irreversible 4−conversion on the toroidal grid $T_{m,n}$, this result provides a sharp bound ($C_4(T_{m,n}) \ge mn/2$) for irreversible 4−conversion sets when m and n are even.

5 Discussion and Open Problems

The results described here have been formulated mostly in the language of spread of disease. As models of disease spread, they are somewhat oversimplified. However, we feel that models are tools for reasoning about problems and, as such, these models provide ways to formalize concepts such as effect of the topology of social networks on the spread of disease; alternative vaccination strategies; and other concepts. As we have noted, the concepts and ideas developed here also have potential interest in other applications. They were originally developed for understanding how opinions might spread through social networks. Clearly the work also has interest in the context of the "firefighter problem." We hope that the results and ideas are also of interest as mathematical concepts and problems.

There are several questions that would provide avenues for further research.

1. We would like a characterization of irreversible minimum conversion sets for all complete multipartite graphs and trees, as well as other classes of graphs (chordal graphs, interval graphs, planar graphs, other kinds of grids, etc.), or possibly algorithms for finding such sets.

2. We would like to know the complexity of the $(IR)2-CS$ problem.

3. In our work, updates to the vertices occur in parallel at each time step. We could update individual vertices serially, which would certainly have an effect on the dynamics of the network. In addition, we have considered a deterministic update rule. In [12] and [13], Hassin and Peleg replace the deterministic update rule with a random one, where the probability that a vertex switches signs is equal to the fraction of its neighbors with the opposite sign. We could also replace this with any probability function, perhaps one based on some threshold k (for example, a vertex switches sign with probability i/k where i is the number of neighbors with the opposite sign: if $i \ge k$ the vertex switches sign automatically).

4. As we have noted, analogous results on the k-threshold processes where one can switch back from state 1 to state 0 would be of interest. There are extensive results about this process and also the process we have called monotone in [4]. However, exact results for minimum conversion sets are still missing for the most part. Other models of interest could allow switching back from state 1 to state 0 automatically after a given number of time steps (given number of days for a disease to run its course). We could also bring in "immunity" after a vertex has changed from state 1 to state 0. Probabilistic variants of all of these models would also be of interest.

5. We have described vaccination strategies in Section 1.3. A number of questions and research directions remain open here. For instance, the bulk of the work described in Section 1.3 involves rectangular grids and trees. The same problems are of interest for chordal graphs, interval graphs, planar graphs, and other kinds of grids such as annular grids. There is very little known about vaccination strategies where we have a varying number $f(t)$ of doses of vaccine each time period. Also, the models considered so far do not consider the possibility that vaccination will only be successful with a certain probability and they do not consider the possible side effects of vaccination, which could be brought into the models in various ways.

6. Other public health interventions could be made precise in the language of the irreversible k-threshold processes. For example, can we make precise the notion of "ring quarantine" – quarantining all individuals within a certain graph distance D from newly-infected individuals? What are "optimal" values of D? Could we have quarantines only lasting a certain number of time steps? Could we bring in noncompliance with quarantines?

7. We would like to find graph topologies that protect against having small irreversible k-conversion sets. For example, one could ask, given n and m, what are graphs on n vertices that have no irreversible k-conversion sets of m or fewer vertices?

In short, the results in this paper are just a beginning. They illustrate the types of analyses one can make using graph-theoretical models of the spread of disease or opinion. There is much more work to do.

References

[1] E. Berger. Dynamic monopolies of constant size. Technical report, Los Alamos National Laboratories, 1999.
Available at `http://xxx.lanl.gov/abs/math/9911125`.

[2] E. Berger. Personal communications via e-mail, May 1999.

[3] M.H. Degroot. Reaching a consensus. *Journal of the American Statistical Association*, 69:167–182, 1974.

[4] P.A. Dreyer. *Applications and Variations of Domination in Graphs.* PhD thesis, Rutgers University, 2000.

[5] P. Flocchini, E. Lodi, F. Luccio, L. Pagli, and N. Santoro. Irreversible dynamos in tori. In *Proceedings of EUROPAR*, pages 554–562, 1998.

[6] P. Flocchini, E. Lodi, F. Luccio, L. Pagli, and N. Santoro. Monotone dynamos in tori. In *Proceedings of the* 6^{th} *Colloquium on Structural Information and Communication Complexity*, 1999.

[7] P. Fogarty. Catching the fire on grids. Masters Thesis, University of Vermont, 2003.

[8] J.R.P. French. A formal theory of social power. *Psychology Review*, 63:181–194, 1956.

[9] G.H. Fricke, S.T. Hedetniemi, and D.P. Jacobs. Independence and irredundance in $k-$regular graphs. *Ars Combinatorica*, 49:271–279, 1998.

[10] S.G. Hartke. *Graph-theoretic Models of Spread and Competition.* PhD thesis, Rutgers University, 2004.

[11] B. Hartnell and Q. Li. Firefighting on trees: How bad is the greedy algorithm? *Congressus Numerantium*, 145:187–192, 2000.

[12] Y. Hassin and D. Peleg. Distributed probabilistic polling and applications to proportionate agreement. In *Proceedings of the* 26^{th} *International Colloquium on Automata, Languages, and Programming*, pages 402–411, 1999.

[13] Y. Hassin and D. Peleg. Extremal bounds for probabilistic polling in graphs. In *Proceedings of the 7th International Colloquium on Structural Information and Communication Complexity*, 2000.

[14] R. Impagliazzo, R. Paturi, and M. Saks. Size-depth tradeoffs for threshold circuits. *SIAM Journal of Computing*, 26(3):693–707, June 1997.

[15] S.F. Kapoor, H.V. Kronk, and D.R. Lick. On detours in graphs. *Canadian Mathematics Bulletin*, 11:195–201, 1968.

[16] D. Li and Y. Liu. A polynomial algorithm for finding the minimum feedback vertex set of a $3-$regular simple graph. *Acta Mathematica Scientia*, 19:375–381, 1999.

[17] F. Luccio. Almost exact minimum feedback vertex set in meshes and butterflies. *Information Processing Letters*, 66:59–64, 1998.

[18] F. Luccio, L. Pagli, and H. Sanossian. Irreversible dynamos in butterflies. In *Proceedings of the Sixth Colloquium on Structural Information and Communication Complexity*, 1999.

[19] G. MacGillivray and P. Wang. On the firefighter problem. *J. Combin. Math. Combin. Comput.*, 47:83–96, 2003.

[20] N. Mustafa and A. Pekeč. Democratic consensus and the majority rule. Technical report, BRICS, 2000. Available at `http://www.brics.dk/RS/00/8/`.

[21] K.L. Ng and P. Raff. Fractional firefighting in the two dimensional grid. National University of Singapore and Rutgers University, submitted 2006.

[22] D. Peleg. Local majorities, coalitions, and monopolies in graphs: A review. In *Proceedings of the Third Colloquium on Structural Information and Communication Complexity*, 1996.

[23] D. Peleg. Size bounds for dynamic monopolies. *Discrete Applied Mathematics*, 86:263–273, 1998.

[24] S. Poljak and M. Sûra. On periodical behavior in society with symmetric influences. *Combinatorica*, 3:119–121, 1983.

[25] P. Wang and S.A. Moeller. Fire control on graphs. *J. Combin. Math. Combin. Comput.*, 41:19–34, 2002.