How even Tiny Influence can have a Big Impact!

Barbara Keller[1], David Peleg[2], and Roger Wattenhofer[1]

[1] ETH Zürich
[2] The Weizmann Institute, Rehovot

Abstract. An influence network is a graph where each node changes its state according to a function of the states of its neighbors. We present bounds for the stabilization time of such networks. We derive a general bound for the classic "Democrats and Republicans" problem and study different model modifications and their influence on the way of stabilizing and their stabilization time. Our main contribution is an exponential lower and upper bound on weighted influence networks. We also investigate influence networks with asymmetric weights and show an influence network with an exponential cycle length in the stable situation.

Keywords: Social Networks, Stabilization, Influence Networks, Majority Function, Equilibrium, Weighted Graphs, Asymmetric Graphs

1 Introduction

"My kid is . . . a brat, a bully, ugly, fat, a loser, out of control, smoking weed."
"My parents are . . . stupid, overprotective, annoying, idiots, too strict, . . . , *cousins*!"
Googles autocomplete feature (quotes above) may teach us a thing or two about what is going on in parent-kid relationships these days. Indeed, when their children become teenagers, parents around the world are frightened of losing their influence. Instead, the kids are rather influenced by (and will influence) their peers.
In this paper we want to understand the complexity of influence networks from a computer science perspective. How erratic can the behavior of such a weighted influence network be? Can even a bit of (bad) influence by the parents have an impact on all the (good) influence of the peers? Can a social network become unstable in the sense that the nodes of the network change their opinions often?
As it turns out, having weighted influence changes the behavior of networks quite dramatically. Whereas previous work showed that unweighted influence networks always stabilize in polynomial time, weighted networks may need exponential time to stabilize. Influence may also be highly asymmetric: If Justin Bieber and Katy Perry declare that facebook all of a sudden is cool again (yes, facebook currently is uncool, according to Google autocomplete), then quite a few of their combined 100M followers will agree. On the other hand, if the authors of this paper proclaim that this paper is a super submission, the impact on the reviewers may be limited. However, our story does not stop at asymmetric weighted influence: Teenagers might consider their parents a perfect counter example, i.e., many parents may be shocked to learn that they probably even have negative influence, that is, their children will try to do exactly the opposite of what they are told.

Background. Influence networks (INs) are networks whose entities are continuously influenced by the state of their neighbors. Such networks are widespread in nature and are of interest in many research areas. Despite their conceptual simplicity and easy to describe mode of operation, their dynamic behavior is sometimes surprisingly complex, and is known to be a source of many open and often hard to analyze problems.
In this paper we study a synchronous and generic version of these systems. We assume a graph $G = (V, E)$ where the nodes are influenced by the states of their neighbors. We assume binary states and focus on the synchronous setting where all nodes simultaneously update their state on each time step according to the majority of their neighbors. (Nonetheless, let us mention that all of the findings presented in this paper also apply to the *iterative* setting, where exactly one node is activated on each time step, and the schedule is determined by an adversary.) These systems "stabilize" after a certain amount of time and we are interested in the states to which they converge, and in their stabilization time within different models. We consider positive and negative influence as well as *weighted* influence, where the influence of different neighbors weighs differently. In addition, we investigate asymmetrically weighted INs (where the influence of v on its neighbor u may be different from that of u on v), which turns out to have different "stable" states than the INs with symmetric weights.

Related Work. The mechanisms by which people influence each other have been objects of considerable interest in psychology and sociology for a long time, and the roles of peer pressure, conformity and socialization, as well as persuasion in sales and marketing, were extensively studied [MJ34,Kel58,Car59]. With the appearance of online social networks, the computer science community too began investigating ties and influences in social networks, cf. [MMG$^+$07,LHK10].
Influence networks, based on the concept of nodes being continuously influenced by their neighbors, are studied in diverse areas, such as mechanical engineering [Koh89], brain science [RT98], ant colonies [AG92] and the spreading of diseases [KMLA13]. Even more heterogeneous are asymmetrically weighted influence networks, which play an important role in fields such as metabolic pathways [JTA$^+$00], power distribution networks [WS98] and citations between academic papers [GM79]. A famous application example concerns the classification of the importance of web pages [BP98,Kle99]. A lot of work has been done on analyzing rumor spreading, either structurally or algorithmically, using the random phone call model [FPRU90,KSSV00,SS11,DFF11]. One may try to predict the most influential nodes which can spread the rumor (in this case - a product) as widely as possible cf. [KKT05,CYZ10]. In [KOW08] the authors study a model somewhat closer to ours, which involves competitors on graphs, leading to nodes with different states. The main difference between these models and ours is that in the rumor spreading process nodes may change their state at most once during the execution, once they get infected they stay infected. In contrast, in our model a node can change its state several times until it reaches a "stable state", and even after entering this stable state, it may continue changing its opinion in a cyclic pattern.
In this sense cellular automata [Neu66,Wol02] are closer to our model. One can interpret our synchronous model as a cellular automata on a general graph instead of a grid, where the rule used to update the state of the nodes is to adopt the majority of the states of the neighboring nodes.

The same model we study is used to study *a dynamic monopoly* [Pel96,FLL$^+$04,Man12], abbreviated dynamo. A dynamo is an initial set of vertices in an influence network, all with the same opinion, such that after a finite number of steps all nodes in the network share their opinion. The minimum size of a dynamo was studied in [Pel98], where it is shown that $\Omega(\sqrt{n})$ nodes are needed for a monotone dynamo (assuming no node ever changes back its state) and for 2-round dynamos (where the network stabilizes after exactly 2 rounds). Berger [Ber08] extends these results by proving that a constant number of nodes may suffice to convert a network of size at least n for arbitrary n. In the current work we ignore the final opinions of the network, and focus on stabilization time.
Updating rules taking into account the states of ones neighbors and a threshold is wide spread in biological applications and neural networks, and was studied already during the 1980's. Goles and Olivos [GO80] have shown that a synchronous binary influence network with symmetric weights and a generalized threshold function always leads to a cycle of length at most 2. This implies that after an influence network has balanced itself, each node has either a fixed opinion or changes its mind every round. This result was extended by Poljak and Sura [PS83] to a finite number of opinions. Moreover, Goles and Tchuente [GT83] show that an iterative behavior of a threshold function with symmetric weights always leads to a fixed point. Sauerwald and Sudholt [SS10] studied the evolution of cuts in a binary influence network model where nodes may flip sides probabilistically. In comparison, our work may be interpreted as looking at the deterministic and weighted case of that problem instead. In [FKW13] the authors proved a lower bound for the stabilization time of unweighted influence networks. They constructed a class of synchronous influence networks with stabilization time $\Omega(n^2/\log^2 n)$ and proved a worst case stabilization time of $\Theta(n^2)$ for iterative influence networks.

2 Preliminaries

Model. We model an *influence network* (IN) as a graph $G = (V, E, \omega, \mu_0)$. The set of nodes V is connected by an arbitrary set of edges E. Each edge is assigned a weight $\omega(e) \in \mathbb{N}$. We refer to an edge between nodes u and v with weight ω as (u, v, ω). Usually we talk about undirected edges, except in Section 4 (about asymmetric graphs), where we consider directed edges. In this case u, v, ω stands for an edge from node u towards node v with weight ω The weight of a graph G is defined as $\omega(G) = \sum_{e \in E} \omega(e)$.
Each node has an initial opinion (or state) $\mu_0(v) \in \{R, B\}$ (graphically interpreted as the Red and Blue colors respectively). The opinions of the nodes at every time t are also represented in the same way by a function $\mu_t : V \mapsto \{R, B\}$.
We define the "red neighborhood" of node v by $\Gamma_{R,t}(v) = \{u \in \Gamma(v) \mid \mu_t(u) = R\}$ at time t and similarly for the "blue neighborhood" $\Gamma_{B,t}(v)$. A node changes its opinion on time step t if a weighted majority of its neighbors has a different opinion. One can consider different actions in case of a tie. We chose the nodes own opinion as a tie breaker because of two reasons. First it seems to be a natural choice and secondly one can build an equivalent graph with the same behavior in asymptotic running time by cloning the graph and connecting each node with its clone and the neighbors of its

clones. More formally the state of a node at time $t+1$ is defined as

$$\mu_{t+1}(v) = \begin{cases} R, & \text{if } \sum\limits_{u \in \Gamma_{R,t}(v)} \omega(u,v) > \sum\limits_{u \in \Gamma_{B,t}(v)} \omega(u,v) \\ B, & \text{if } \sum\limits_{u \in \Gamma_{B,t}(v)} \omega(u,v) > \sum\limits_{u \in \Gamma_{R,t}(v)} \omega(u,v) \\ \mu_t(v), & \text{otherwise .} \end{cases}$$

A synchronous IN develops in rounds. In each round the nodes simultaneously update their opinion to the weighted majority of their neighbors according to the above rule. As INs are deterministic, they necessarily enter a cyclic pattern after a certain number of rounds. We call an IN *stable* with a cycle of length q if each node changes its opinion in a cyclic pattern with cycle length k for some $k \leq q$. This means that in a stable IN it is still possible that nodes change their opinions.

Definition 1. *An IN $G = (V, E, \mu_0)$ is* stable *at time t with cycle length q, if for all vertices $v \in V : \mu_{t+q}(v) = \mu_t(v)$. A* fixed state *of an IN G is a stable state with cycle length* 1*. The* stabilization time *c of an IN G is the smallest t for which G is stable.*

The Classical (Unweighted) Model. The classical model, sometimes also known as "Democrats and Republicans", is unweighted, i.e., all the edges of the IN have weight exactly 1. We hereafter refer to such an IN as an *unweighted influence network*, or UIN in short. A known basic fact concerning the dynamical behavior of UIN's is the following.

Theorem 1 ([GO80]). *The cycle length of the stable state of a synchronous unweighted influence network with symmetric weights is at most* 2*.*

Theorem 2 ([Win08]). *An n-node unweighted influence network stabilizes in $O(n^2)$ time steps.*

Using Theorem 1 and Theorem 2, near-tight bounds were recently established in [FKW13] on the stabilization time of UIN's.

Theorem 3 ([FKW13]). *There is a family of n-node unweighted influence networks that require $\Omega(n^2/\log^2 n)$ rounds to stabilize.*

The proof of the upper bound in Theorem 2 uses a bound argument on the edges. Each edge (v, u) is substituted by two directed edges $\langle v, u\rangle$ and $\langle u, v\rangle$, with the same weight, referred to as the outgoing and incoming edges of v, respectively. One can think of these edges as representing "advice" given between neighbors. The outgoing edge from node v to node u can be seen as the opinion that node v proposes to its neighbor u and the incoming edge can be seen as the opinion that node u proposes to v. In each time step t, each of these directed edges is declared to either "succeed" or "fail". The outgoing edge $\langle v, u\rangle$ succeeds on time step t if the neighbor u accepts the opinion proposed by v during the round leading from time step t to time step $t+1$, namely, $\mu_{t+1}(u) = \mu_t(v)$, and fails otherwise.
The analysis is based on the initial observation that a UIN starts with a certain number of failed edges $f(0)$ at time step 0, which is naturally bounded by $f(0) \leq 2|E|$. It is

shown that as long as the UIN has not stabilized, the number of failed edges $f(t)$ decreases in every round by at least one. Using the same arguments, the upper bound for a UIN can be restated as $2|E|$.

Theorem 4 ([Win08]). *An n-node unweighted influence network with edge set E stabilizes in at most $2|E|$ time steps.*

Friends and Fiends. Some online networks allow not only to be connected to one's friends but also to one's fiends (e.g. Epinions, Slashdot). We model such a network by allowing not only positive but also negative influence between members. Informally, one can think of a negative link between u and v as u's tendency to adopt the opinion opposite to that of v.
The proof given in [Win08] for the upper bound, as well as the lower bound construction used in [FKW13], can be applied to this model. The definition of a "failed edge" has to be updated to apply also for negative edges. This is done in a straightforward manner by using the same definition for successful and failed edges in the case of positive ties and by using the opposite definition in case of negative ties. Namely, a negative outgoing edge $\langle v, u\rangle$ fails on time step t if u adopts v's opinion on time step $t+1$, and succeeds otherwise. We get the following results.

Lemma 1. *There exists a family of n-node unweighted synchronous influence networks with stabilization time $\Omega(n^2/\log^2 n)$.*

Lemma 2. *An n-node unweighted influence network with positive and negative friendship ties stabilizes in $O(n^2)$ time steps.*

3 Weighted Graphs

In a social network it seldom happens that all ties have the exact same interpretation. Considering, for instance, different acquaintance ties between people, one may observe that usually people listen to their best friends more than to their colleagues. We model the influence between a pair of nodes by assigning a weight to the corresponding edge. It is then assumed that a node changes its opinion if the *weighted majority* of its neighbors have a different opinion. We are interested in the influence of adding weights to our graph on the stabilization time. We start by proving the following lemma.

Lemma 3. *An n-node weighted influence network G stabilizes in $\min\{2\omega(G), 2^n\}$ time steps.*

Proof. Note that there is a bijective relation between weighted graphs and multigraphs. A weighted graph can be modeled as a multigraph by replacing each edge e of weight $\omega(e) = k$ by k edges of weight 1 each. Conversely, each multigraph can be modeled as a weighted graph with weights $\omega(e) \in \mathbb{N}$ by substituting k multiedges by a single edge of weight k. The transformation does not influence the behavior of the nodes as in both situations the weight of the influence is not changed. For the multigraph we can apply Theorem 4 to conclude that the multigraph stabilizes in $2|E|$ tie steps. As the number of edges in the multigraph corresponds to the weight $\omega(G)$ of the weighted graph, we conclude that a weighted IN stabilizes in $2\omega(G)$ time. Moreover, this process is deterministic, and the execution enters a cycle once some global state repeats itself. Consequently, since the IN has 2^n global states, it must stabilizes in at most 2^n time step. □

The stabilization time of an IN can not be prolonged arbitrarily by just setting the edge weights higher. A path network, for example, will stabilize in $O(n)$ rounds no matter how the edge weights are chosen. It is an intriguing question if the weights do indeed have an influence on the stabilization time or if there is another mechanism which may prevent INs from having a higher stabilization time than $O(n^2)$. As we show in the next paragraphs, edge weights can significantly increase the stabilization time of INs. We do this by presenting a family of graphs with stabilization time $2^{\Omega(n)}$

Lemma 4. *There is a family of n-node weighted influence networks with stabilization time $2^{\Omega(n)}$.*

To provoke as many changes as possible we build a graph consisting of 3 different component types: Two different colored paths of length l and several levels of a structure to which we refer as *transistor lines*. The transistor lines consist of 2 different colored lines of k transistors. The main idea is that the paths, with a suitable initial coloring, get "discharged" by a lengthy process during which they change their colors node by node as often as possible, and once a path is completely discharged, it gets recharged (i.e., reset to the original color pattern) by the transistor above it. In turn, each transistor in the transistor lines recharges the levels below it. So each level adds another multiplicative factor to the stabilization time. (Let us remark that we have programmed the construction and simulated the influence propagation process on this graph; the interested reader may find a program and a video tracing the simulation as well as a more formal definition of the graph at http://www.disco.ethz.ch/members/barkelle/FUN.zip)
Let us now take a closer look how the two paths, illustrated in Figure 1, work.
At round 0 all the nodes in path P^1 are blue and all the nodes in path P^2 are red. The first nodes of the paths are denoted by F. When they change their color they start a cascade of changes through the path. To achieve this, the weights of the edges between the path nodes are decreasing from the first to the last node. This ensures that the change of the first node is cascaded through the path without any influence going the opposite direction. The summed up influence to change the color of the first node has therefore also to be higher than the weight of the edge between the first and the second node.

Definition 2. *We define our path graph $P = (V_P, E_P)$ as an undirected weighted graph, where $V_P = \{p_1, \ldots, p_k\}$ and*

$$E_P = \{(p_i, p_{i+1}, 2k + l - i) \mid i = 0, 1, \ldots, k - 2\}.$$

The levels above the paths consist of two transistor lines. At time 0, line L^1 is blue and L^2 is red. Each transistor line is composed of k transistors. The basic function of the transistor is to change the color in the level below it in a controlled manner. A transistor (see Figure 2) consists of three nodes: A switch node (Sw), a collector node (Co) and an emitter node (Em). The idea behind this is to control the color of the transistor by the switch node. This is done by using the switch node to change the color of the collector node which in return changes the color of the emitter node. To do so, the switch node

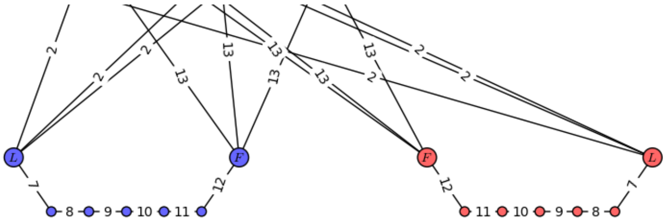

Fig. 1: The path nodes are connected with decreasing weights from the first node (F) to the last (L) in the path. This induces a cascade of color changes through the path once the first node changes its opinion. The edge between the last and second-to-last node has a weight larger than $k \cdot 2$ where k is the number of transistors in a transistor line. The cascade of changes is triggered by changing the color of one external node that is connected to F.

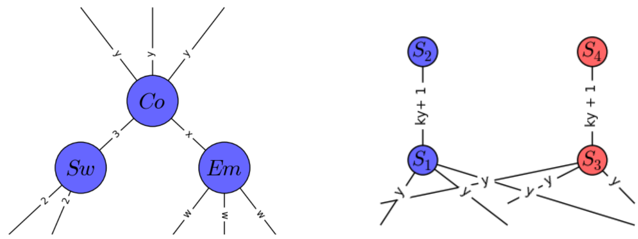

Fig. 2: A transistor on the left consists of the following three nodes: switch (Sw), collector (Co) and an emitter (Em). Its edge weights satisfy $2x > \sum_{(u,Em)\in E} \omega(u, Em)$ and $x < y < x + 3$. The graph with four nodes on the right will never change their color, as they share an edge with a higher weight than all the other adjacent edges combined. With k transistors per line, this weight is set to $k \cdot y + 1$.

shares an edge with the collector node of weight 3 and the collector node is balanced in a way that it will only change its color, if the summed up influence from the level above is the opposite color and the switch node changes to this color. The collector node shares an edge with the emitter node which is heavier than all the other edges adjacent to the emitter node combined. This makes sure that the emitter node changes its color exactly one round after the collector node changed its color, no matter what the color of the other neighboring nodes are.
The order in which the transistors are activated is level by level, and in each level - transistor by transistor. To make sure that a transistor is only activated when the transistor in front of it already finished, we add an edge of weight 2 between the emitter node and the switch node of the next transistor in the transistor line (the switch node of the first transistor in each line is connected to the last emitter of the opposite colored transistor line); see Figure 3.

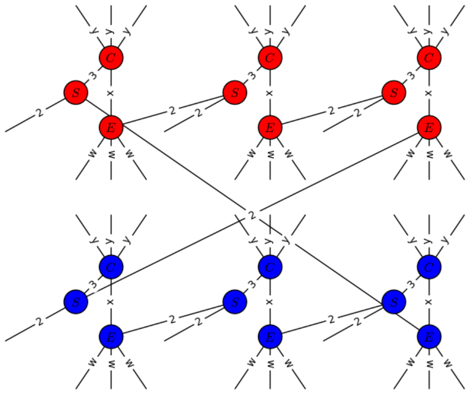

Fig. 3: A level of transistor lines, consisting of k blue and k red transistors. Each switch is wired to the emitter of the transistor in front of it, so that the transistors get activated in the order from left to right.

Above the levels we have 4 special nodes, $S_1 - S_4$. These 4 nodes consist of two pairs of nodes, one (S_1, S_2) blue and the other (S_3, S_4) red. Each pair is connected by an edge which is heavier than all the other edges adjacent to these nodes combined (see Figure 2). This ensures that these nodes never change their color.

Definition 3. *A transistor $T = (V_T, E_T)$ is an undirected weighted graph, where $V_T = \{Sw, Co, Em\}$ and $E_T = \{(Sw, Co, 3), (Co, Em, x)\}$, where x is dependent on the level i of the transistor and the length l of the path.*

$$x_0 = k \cdot (k \cdot 2 + l) + 2k + 3$$
$$x_i = k \cdot (x_{i-1} + 2k + 3) + 3.$$

Definition 4. *A transistor line $L^j = (V_{L^j}, E_{L^j})$ is an undirected weighted graph consisting of k transistors, where*

$$V_{L^j} = \bigcup_{i=0}^{k-1} V_{T_i} \text{ and } E_{L^j} = \bigcup_{i=0}^{k-1} E_{T_i} \cup \{(Em_i, Sw_{i+1}, 2) \mid i = 0, 1, \ldots, k-1\}.$$

Definition 5. *A level $L = (V_L, E_L)$ is an undirected weighted graph consisting of two transistor lines L^1, L^2, where $V_L = V_{L^1} \cup V_{L^2}$ and*

$$E_L = E_{L^1} \cup E_{L^2} \cup \{(u, v) \mid u = Sw \in T_0 \in L^1 \wedge v = Em \in T_{k-1} \in L^2\}$$
$$\cup \{(u, v) \mid u = Sw \in T_0 \in L^2 \wedge v = Em \in T_{k-1} \in L^1\}.$$

We now show how the structures and different levels are wired. We start with the two paths and the first transistor lines. We want the blue path P^1 to turn alternately red and

blue. As the blue transistors have the potential to turn red we connect the emitter of the first blue transistor with the first node of the blue path with weight $w = 2 \cdot k + l$. Similarly, we connect the first red transistor with the first node of the red path. In order to turn them back to their original color we connect the emitter of the second red transistor to the first node of P^1 and the emitter of the second blue transistor to the first node of P^2. We continue this until the first node of P^2 is connected with all the emitters of the even transistors in L^1 and all the emitters of the odd transistors in L^2. To inhibit the second transistor from changing P^1 back before the first cascade finished we connect the last node in P^2 with the switch of the second transistor in L^2 with a weight $w = 2$. So the switch node can only switch if the transistor in front of it and the last node of the opposite colored path did switch. As P^1 and P^2 have the same length and start at the same time they will also finish at the same time the cascade and will influence the second transistors to switch. All these edges are added for the first node F in P^2 and the last node L in P^2 respectively. Note that with k being odd, there is always a summed up influence on the first node of the path with value $2k + l$. The edges added between the paths and the first levels are denoted by E_{PL_1}.
The different levels are wired similarly to the first level and the path. The first node of the path corresponds to the collector nodes of the transistors one level below and the last node of the path corresponds to the emitter node of the last transistor. We denote the edges between level L_i and level L_{i+1} as E_{LL_i}.
The last level L_r is wired to the special nodes in the following way:

$$\begin{aligned} E_{SL_r} = \; & \{(u, v, x_r + 2k + 2) \mid u = S_1 \wedge v \in \{Co_j \in L_r^2\}\} \\ & \cup \{(u, v, x_r + 2k + 2) \mid u = S_3 \wedge v \in \{Co_j \in L_r^1\}\}. \end{aligned}$$

The complete asymmetric IN is a union of all these structures and can be seen with $k = 3$ and $l = 3$ and 2 levels in Figure 4.

Definition 6. *Our worst case IN is an undirected weighted graph consisting of 2 paths, r levels and the 4 special nodes. Formally it is defined as* $IN = (V_{IN}, E_{IN})$*, where*

$$\begin{aligned} V_{IN} &= V_{P^1} \cup V_{P^2} \cup \bigcup_{i=1}^{r} V_{L_i} \cup \{S_1, S_2, S_3, S_4\} \\ E_{IN} &= E_{P^1} \cup E_{P^2} \cup E_{PL_1} \cup \bigcup_{i=1}^{r-1} E_{LL_i} \cup E_{SL_r} \end{aligned}$$

Stabilization Time. We analyze the stabilization time of the presented graph. Each time the path is activated it takes l rounds to complete. As the transistors from the first level are connected to the last nodes of the two paths they only change when all the nodes in the path changed their color. As a transistor needs 3 rounds to change (switch $\rightarrow$ collector $\rightarrow$ emitter) and nothing of this happens in parallel, the first level takes $k \cdot (3 + l)$ rounds. Each additional level adds a factor of k to the stabilization time which

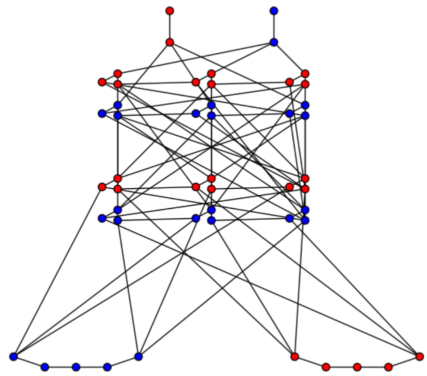

Fig. 4: This is an IN with stabilization time $l \cdot k^{(n-l)/k*6}$ with $k = 3$, $l = 5$ and $r = 2$.

leads to the following recursive function for the running time: $T_i = k^i \cdot (3 + l) + T_{i-1}$. Solving the recurrence formula gives

$$T_i = \frac{(l+3)(k^{i+1}-1)}{k-1} \in O(k^i) \ .$$

The running time grows exponentially in the number of levels. Each level consists of $2\dot{k}$ transistors and each transistor consists of 3 nodes. The constant nodes consist of $2 \cdot 2$ nodes. The IN consists of n nodes, therefore we can build an IN with $i = \frac{n-l-4}{6 \cdot k}$ levels. Choosing l to be constant and $k = 3$, we achieve a stabilization time of $\Omega(3^{n/18}) = \Omega(2^{0.088n})$.

4 Asymmetric Weights

In real life, ties between people do not necessarily have the same weight for both adjacent nodes. Although friendships are often symmetrically perceived concerning how strong they are, there are a lot of examples where this is not the case. One example is the student advisor relation. Usually the advisor's opinion has a larger influence on the student than vice versa, hence the edge between advisor and student has a smaller weight for the advisor than for the student. This is even more extreme in the case of celebrities: A famous artist may influence people whom she does not even know, who in return do not influence her at all. We extend the model to allow asymmetric weights. Note that the weight can also be 0 on one side, which is then equivalent to a directed edge. Interestingly, in this new model the "stable states" are not that simple anymore, as the cycle length can be larger than 2. We are interested in the cycle length which can be achieved. An easy lower bound on it is n. One can think of a circle with edges directed in one direction. Initially, one node is red and all the others are blue. This red "token" cycles through the circle with cycle length n. We are interested in how big the cycle length can get in an IN with asymmetric weights. As asymmetric INs are deterministic too, we have an upper bound of 2^n. We show a family of graphs with a cycle length of $2^{\Omega(n)}$

Lemma 5. *There are families of n-node influence networks with a cycle length $2^{\Omega(n)}$.*

We use the same IN as described in Section 3 as a basis for our construction but substitute most of the symmetric edges by directed edges. The main idea is to have the same process as the IN in Section 3, except that in the round where the symmetric IN would stabilize, our IN gets restarted by changing the colors of the special 4 nodes $S_1 - S4$. This leads to a cycle length for our asymmetric IN that's twice as long, as the stabilization time of the previous IN. In order to do so, we add directed edges from the last emitter of each transistor line and each path to the nodes $S_1 - S_4$ with a weight x that sums up to a weight higher than the edge weight $w(S_1, S_2) = w(S_3, S_4)$ but so that each subset of these weights is smaller than w. This will change the special nodes exactly when all the levels have switched. This is achieved by assigning the edges $w(S_1, S_2) = w(S_3, S_4) = 3(r + 1) - 1$, where r is the number of levels. Note that as directed edges can be used, we do not need an exponential growth of the weights anymore which makes the graph simpler.
To build our AIN we change the edges in the IN from Sect. 3 in the following way: The edges in the path graph are directed from the first node (F) to the last (L) with weight 1. The edges between the emitter from the first level and F are directed towards F and have weight 4. For each transistor the two edges of the switch node with weight 2 are now directed towards the switch node. The edge between the switch node and the collector node is substituted by a directed edge to the collector node with weight 3. The edge between a collector and an emitter in the same transistor stays symmetrical but its weight is now 4. All the edges between collector and emitter nodes from different levels are substituted by directed edges from the emitter of the higher level to the collector on the lower level with weight 4. All the edges between the special nodes and the collectors from level r are now directed towards the collector nodes and have weight 4. The edge between S_1 and S_2 (as well as between S_3 and S_4) is symmetric and has an assigned weight of $3(r + 1) - 1$. Additionally we add the previously described special edges.

References

[AG92] Frederick R. Adler and Deborah M. Gordon. Information Collection and Spread by Networks of Patrolling Ants. *The American Naturalist*, 1992.

[Ber08] Eli Berger. Dynamic Monopolies of Constant Size. *Journal of Combinatorial Theory Series B*, 2008.

[BP98] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *WWW*, 1998.

[Car59] D. Cartwright. *Studies in Social Power*. Publications of the Institute for Social Research: Research Center for Group Dynamics Series. 1959.

[CYZ10] Wei Chen, Yifei Yuan, and Li Zhang. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *ICDM*, 2010.

[DFF11] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. Social Networks Spread Rumors in Sublogarithmic Time. In *STOC*, 2011.

[FKW13] Silvio Frischknecht, Barbara Keller, and Roger Wattenhofer. Convergence in (Social) Influence Networks. In *DISC*, October 2013.

[FLL+04] Paola Flocchini, Elena Lodi, Fabrizio Luccio, Linda Pagli, and Nicola Santoro. Dynamic Monopolies in Tori. *IWACOIN*, 2004.

[FPRU90] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Randomized Broadcast in Networks. *Random Structures and Algorithms*, 1990.
[GM79] E. Garfield and R. K. Merton. Citation Indexing-Its Theory and Application in Science, Technology, and Humanities. 1979.
[GO80] E. Goles and J. Olivos. Periodic Behaviour of Generalized Threshold Functions. *Discrete Mathematics*, 1980.
[GT83] E. Goles and M. Tchuente. Iterative Behaviour of Generalized Majority Functions. *Mathematical Social Sciences*, 1983.
[JTA+00] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabsi. The Large-Scale Organization of Metabolic Networks. *Nature*, 2000.
[Kel58] H.C. Kelman. Compliance, Identification, and Internalization: Three Processes of Attitude Change. *Journal of Conflict Resolution*, 1958.
[KKT05] David Kempe, Jon M. Kleinberg, and Éva Tardos. Influential Nodes in a Diffusion Model for Social Networks. In *ICALP*, 2005.
[Kle99] Jon M. Kleinberg. Hubs, Authorities, and Communities. *CSUR*, 1999.
[KMLA13] Christel Kamp, Mathieu Moslonka-Lefebvre, and Samuel Alizon. Epidemic Spread on Weighted Networks. *PLoS Computational Biology*, 2013.
[Koh89] T. Kohonen. *Self-Organization and Associative Memory*. Springer, 1989.
[KOW08] Jan Kostka, Yvonne Anne Oswald, and Roger Wattenhofer. Word of Mouth: Rumor Dissemination in Social Networks. In *SIROCCO*, 2008.
[KSSV00] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized Rumor Spreading. In *FOCS*, 2000.
[LHK10] Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. Signed Networks in Social Media. In *CHI*, 2010.
[Man12] Zaker Manouchehr. On Dynamic Monopolies of Graphs with General Thresholds. *Discrete Mathematics*, 2012.
[MJ34] J.L. Moreno and H.H. Jennings. *Who Shall Survive?: A New Approach to the Problem of Human Interrelations*. Nervous and mental disease monograph series. Nervous and mental disease publishing co., 1934.
[MMG+07] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and Analysis of Online Social Networks. In *SIGCOMM*, 2007.
[Neu66] John Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.
[Pel96] David Peleg. Local Majority Voting, Small Coalitions and Controlling Monopolies in Graphs: A Review. In *SIROCCO*, 1996.
[Pel98] David Peleg. Size Bounds for Dynamic Monopolies. *Discrete Applied Mathematics*, 1998.
[PS83] Svatopluk Poljak and Miroslav Sra. On Periodical Behaviour in Societies with Symmetric Influences. *Combinatorica*, 1983.
[RT98] Edmund T. Rolls and Alessandro Treves. *Neural Networks and Brain Function*. Oxford University Press, USA, 1998.
[SS10] Thomas Sauerwald and Dirk Sudholt. A Self-Stabilizing Algorithm for Cut Problems in Synchronous Networks. *Theoretical Computer Science*, 2010.
[SS11] Thomas Sauerwald and Alexandre Stauffer. Rumor Spreading and Vertex Expansion on Regular Graphs. In *SODA*, 2011.
[Win08] Peter Winkler. Puzzled: Delightful Graph Theory. *CACM*, 2008.
[Wol02] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002.
[WS98] D.J. Watts and S.H. Strogatz. Collective Dynamics of 'Small-World' Networks. *Nature*, 1998.