Theoretical Computer Science 584 (2015) 53–66

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Influence diffusion in social networks under time window constraints ☆

Luisa Gargano [a,*], Pavol Hell [b], Joseph G. Peters [b], Ugo Vaccaro [a]

[a] Dipartimento di Informatica, University of Salerno, Italy
[b] School of Computing Science, Simon Fraser University, Canada

A R T I C L E I N F O

Article history:
Received 20 November 2013
Received in revised form 6 December 2014
Accepted 10 February 2015
Available online 14 February 2015

Keywords:
Target set selection
Influence diffusion

A B S T R A C T

We study a combinatorial model of the spread of influence in networks that generalizes existing schemata recently proposed in the literature. In our model, agents change behaviours/opinions on the basis of information collected from their neighbours in a time interval of *bounded size* whereas agents are assumed to have *unbounded memory* in previously studied scenarios. In our mathematical framework, one is given a network $G = (V, E)$, an integer value $t(v)$ for each node $v \in V$, and a time window size λ. The goal is to determine a small set of nodes (*target set*) that influences the whole graph. The spread of influence proceeds in rounds as follows: initially all nodes in the target set are influenced; subsequently, in each round, any uninfluenced node v becomes influenced if the number of its neighbours that have been influenced in the previous λ rounds is greater than or equal to $t(v)$. We prove that the problem of finding a minimum cardinality target set that influences the whole network G is hard to approximate within a polylogarithmic factor. On the positive side, we design exact polynomial time algorithms for paths, rings, and trees.

1. Introduction

Many phenomena can be represented by dynamical processes on networks. Examples include cascading failures in physical infrastructure networks [24], information cascades in social and economic systems [9], spreads of infectious diseases [3], and the spreading of ideas, fashions, or behaviours among people [12,41]. Therefore, it comes as no surprise that the study of dynamical processes on complex networks is an active area of research, crossing a variety of different disciplines. Epidemiologists, social scientists, physicists, and computer scientists have studied diffusion phenomena using very similar models to describe the spreading of diseases, knowledge, behaviours, and innovations among individuals of a population (see [5,10,27] for surveys of the area).

A particularly important diffusion process is that of *viral marketing* [32], which refers to the spread of information about products and behaviours and their adoption by people. Recently, it has also become an important tool in the communication

☆ An extended abstract of a preliminary version of this paper appeared in: Proceedings of 20th International Colloquium on Structural Information and Communication Complexity (Sirocco 2013), Lectures Notes in Computer Science vol. 8179, T. Moscibroda and A.A. Rescigno (Eds.), pp. 141–152, 2013. Parts of this work were done while the first and last author were visiting Simon Fraser University and while the third author was visiting the University of Salerno. This research was supported in part by the Ebco Eppich Endowment Fund at Simon Fraser University and by NSERC of Canada.

* Corresponding author.
E-mail address: lg@dia.unisa.it (L. Gargano).

http://dx.doi.org/10.1016/j.tcs.2015.02.015
0304-3975/

strategies of politicians [33,40] (see also [38] for a nice survey of the area). Although there are many similarities between social and epidemiological contagion [26], social contagion is usually an intentional act on the part of the transmitter and/or the adopter, unlike a pathogen contagion. The spread of ideas requires extra mechanisms in addition to mere exposure, e.g., some kind of "social pressure". More importantly, in the marketing scenario one is interested in *maximizing* the spread of information [25], while this is not likely to happen in the spread of pathogenic viruses. The intent of maximizing the spread of viral information across a network naturally suggests many optimization problems. Some of them were first articulated in the seminal papers [30,31], under various adoption paradigms. The recent monograph [15] contains an excellent description of the area. In the next section, we will explain and motivate our model of information diffusion, state the problem that we are investigating, describe our results, and discuss how they relate to the existing literature.

2. The model, the context, and the results

The network is represented by a pair (G,t), where $G=(V,E)$ is an undirected graph and $t:V \longrightarrow \mathbb{N}=\{1,2,\ldots,\}$ is a function assigning integer thresholds to nodes. We assume that $1 \le t(v) \le \deg(v)$ for each $v \in V$, where $\deg(v)$ is the degree of v. For a given set $S \subseteq V$ and a time window size $\lambda \in \mathbb{N}$, we consider a dynamical process of influence diffusion in G defined by two sequences of node subsets, $\mathtt{Influenced}[S,r]$ and $\mathtt{Active}[S,r]$, $r=0,1,\ldots,$ where

$$\mathtt{Influenced}[S,0]=S, \qquad \mathtt{Active}[S,0]=\emptyset,$$

and for any $r \ge 1$ it holds that

$$\mathtt{Influenced}[S,r]=\mathtt{Influenced}[S,r-1] \cup \left\{v: \left|N(v)\cap \mathtt{Active}[S,r]\right| \ge t(v)\right\}$$

$$\mathtt{Active}[S,r]=\begin{cases}\mathtt{Influenced}[S,r-1] & \text{if } r \le \lambda\\ \mathtt{Influenced}[S,r-1] \setminus \mathtt{Influenced}[S,r-1-\lambda] & \text{if } r>\lambda\end{cases} \tag{1}$$

Intuitively, the set S might represent a group of people who are initially influenced/convinced to adopt a product or an idea. Then the cascade proceeds in rounds. In each round r, the set of influenced nodes is augmented by including each node v that has a number of influenced and *still active* neighbours greater than or equal to its threshold $t(v)$. A node is active for λ rounds after it becomes influenced and then it becomes inactive.

Our model is based on the models in [23,34] which assume that people can be divided into three classes at any time instant. *Ignorants* are those not aware of a rumour/not yet influenced, *spreaders* are those who are spreading it, and *stiflers* are those who know the rumour/have been influenced but have ceased to spread the rumour/influence.[1] Several rules have been proposed to govern the transition from ignorants to spreaders and from spreaders to stiflers, and many papers have studied the dynamics of these systems, mostly in stochastic scenarios (see [8,35] and references quoted therein). Here, we posit that any ignorant node becomes a spreader if the number of its neighbours who are spreaders is above a certain threshold (i.e., the node is subject to a large enough amount of "social pressure"), and any spreader becomes a stifler after λ rounds (because the spreader loses interest in the rumour, for instance). Other papers have studied information diffusion under similar assumptions [21,29].

Our model also captures another important characteristic of influence diffusion. Indeed, research in Behavioural Economics shows that humans take decisions mostly on the basis of very recent events, even though they might hold much more in their memory [2,14]. Moreover, it is well known (e.g. [4]) that people are more inclined to react to pieces of information cumulatively heard during a "short" time interval than to information heard during a considerably longer period of time. In other words, one is more likely to be convinced of an opinion heard from a certain number of friends during the last few days than by an opinion heard sporadically during the last year from the *same* number of people. Therefore, it seems reasonable to study diffusion processes in which people have *bounded memory*, and only the number of spreaders heard during the last λ rounds may contribute to the change of status of an ignorant node.[2] Formally, one has a dynamical process of influence diffusion on G described by the sequence of node subsets $\mathtt{Influenced}'[S,r]$, $r=0,1,\ldots,$ where $\mathtt{Influenced}'[S,0]=S$, and for any $r \ge 1$ it holds that

$$\mathtt{Influenced}'[S,r]=\mathtt{Influenced}'[S,r-1] \cup \left\{v: \left|N(v)\cap \mathtt{Influenced}'[S,r-1]\right| \ge t(v)\right\} \tag{2}$$

if $r \le \lambda$, and

$$\begin{aligned}\mathtt{Influenced}'[S,r]=\ &\mathtt{Influenced}'[S,r-1]\\ &\cup \left\{v: \left|N(v)\cap (\mathtt{Influenced}'[S,r-1] \setminus \mathtt{Influenced}'[S,r-1-\lambda])\right| \ge t(v)\right\}\end{aligned} \tag{3}$$

if $r > \lambda$.

[1] The reader will notice an analogy with the well known SIR model of mathematical epidemiology [3], in which individuals can be classified as Susceptible, Infected, and Recovered.

[2] Another model in which individuals carry a memory of the "amount of influence" received during a bounded time interval has been studied in [26].

It is immediate that (2) and (3) are an equivalent way to write (1): for any $S \subseteq V$ and $r \geq 1$, $\texttt{Influenced}'[S,r] = \texttt{Influenced}[S,r]$, so we get that the spreading process with "stiflers" also describes the spreading process with "bounded memory" governed by (2) and (3).

Summarizing, the specific problem that we shall study in this paper is the following:

Time Window Constrained Target Set Selection (TWC–TSS)
Input: A graph $G = (V, E)$, a threshold function $t : V \longrightarrow \mathbb{N}$, and a time window size λ.
Output: A minimum size set $S \subseteq V$ such that $\texttt{Influenced}[S,r] = V$, for some $r \geq 0$.

In this paper, we will use the term TWC target set to indicate a set $S \subseteq V$ such that $\texttt{Influenced}[S,r] = V$ for some $r \geq 0$.

When λ is large enough, for instance equal to the number n of nodes, our Time Window Constrained Target Set Selection problem is equivalent to the classical Target Set Selection problem studied in [1,6,7,11,13,16–20,22,37,42], among others. Strictly related is also the area of dynamic monopolies (see [28,36], for instance). In terms of our second formulation of the TWC–TSS problem, the classical Target Set Selection problem can be viewed as an extreme case in which it is assumed that people have unbounded memory. In general, the TWC–TSS and the TSS problems are quite different. One of the main difficulties of the new TWC–TSS problem is that the sequence of sets $\texttt{Active}[S,r]$, $r = 0, 1, \ldots$ is not necessarily monotonically non-decreasing: it is possible that $\texttt{Active}[S,r]$ is larger than $\texttt{Active}[S,r+1]$ for some values of r. When $\lambda = n$, we have $\texttt{Active}[S,r] = \texttt{Influenced}[S,r-1]$ for any r, and monotonicity is restored. At the other extreme, when $\lambda = 1$, a node v becomes influenced at time r only if at least $t(v)$ of its neighbours become influenced at *exactly* time $r-1$. This sort of synchronization in the propagation of influence poses new challenges, both in the assessment of the computational complexity of the TWC–TSS problem and, especially, in the design of algorithms for its solution. The example in which the graph G is a path is particularly illuminating. As we shall see in Section 4.1, the Target Set Selection problem is trivial to solve on a path; it is far from being so when there is a fixed time window size λ.

Example 1. Consider the tree T in Fig. 1. The number inside each circle is the node threshold. For convenience, we consider T to be rooted at v_0. If the time window size is $\lambda = 1$, then one set that influences all nodes of T is $S = \{v_{1,2}, v_{2,2}, v_{2,6}, v_{3,4}\}$. Indeed we have

$$\begin{aligned}
\texttt{Influenced}[S,0] &= \{v_{1,2}, v_{2,2}, v_{2,6}, v_{3,4}\} = \texttt{Active}[S,1] \\
\texttt{Influenced}[S,1] &= \texttt{Influenced}[S,0] \cup \{v_{1,1}, v_{1,3}, v_{2,1}, v_{2,3}, v_{2,5}, v_{3,3}, v_{3,5}\} \\
\texttt{Active}[S,2] &= \texttt{Influenced}[S,1] \setminus \texttt{Influenced}[S,0] = \{v_{1,1}, v_{1,3}, v_{2,1}, v_{2,3}, v_{2,5}, v_{3,3}, v_{3,5},\} \\
\texttt{Influenced}[S,2] &= \texttt{Influenced}[S,1] \cup \{v_0, v_{2,4}, v_{3,2}\} \\
\texttt{Active}[S,3] &= \texttt{Influenced}[S,2] \setminus \texttt{Influenced}[S,1] = \{v_0, v_{2,4}, v_{3,2}\} \\
\texttt{Influenced}[S,3] &= \texttt{Influenced}[S,2] \cup \{v_{3,1}\} = V
\end{aligned}$$

In particular we notice that

- $v_{2,4} \in \texttt{Influenced}[S,2]$ since both its child $v_{2,5}$ and its parent $v_{2,3}$ are active in round 2;
- $v_0 \in \texttt{Influenced}[S,2]$ since $t(v_0) = 2$ of its children are active in round 2;
- $v_{3,1} \in \texttt{Influenced}[S,3]$ since both the root v_0 and its child $v_{3,2}$ are active in round 3.

If we assume a larger value of λ, then a smaller set can be chosen; in particular if $\lambda \geq 3$ then the set $S = \{v_{1,2}, v_{2,4}, v_{3,4}\}$ influences all nodes of T. Indeed for $\lambda = 3$ we have

$$\begin{aligned}
\texttt{Influenced}[S,0] &= \{v_{1,2}, v_{2,4}, v_{3,4}\} = \texttt{Active}[S,1] \\
\texttt{Influenced}[S,1] &= \texttt{Influenced}[S,0] \cup \{v_{1,1}, v_{1,3}, v_{2,3}, v_{2,5}, v_{3,3}, v_{3,5}\} = \texttt{Active}[S,2] \\
\texttt{Influenced}[S,2] &= \texttt{Influenced}[S,1] \cup \{v_{2,2}, v_{2,6}, v_{3,2}\} \\
\texttt{Active}[S,3] &= \texttt{Influenced}[S,2] = \{v_{1,1}, v_{1,2}, v_{1,3}, v_{2,2}, v_{2,3}, v_{2,4}, v_{2,5}, v_{2,6}, v_{3,2}, v_{3,3}, v_{3,4}, v_{3,5}\} \\
\texttt{Influenced}[S,3] &= \texttt{Influenced}[S,2] \cup \{v_{2,1}\} \\
\texttt{Active}[S,4] &= \texttt{Influenced}[S,3] \setminus \texttt{Influenced}[S,0] \\
&= \{v_{1,1}, v_{1,3}, v_{2,1}, v_{2,2}, v_{2,3}, v_{2,5}, v_{2,6}, v_{3,2}, v_{3,3}, v_{3,5}\} \\
\texttt{Influenced}[S,4] &= \texttt{Influenced}[S,3] \cup \{v_0\} \\
&= \{v_0, v_{1,1}, v_{1,2}, v_{1,3}, v_{2,1}, v_{2,2}, v_{2,3}, v_{2,4}, v_{2,5}, v_{2,6}, v_{3,2}, v_{3,3}, v_{3,4}, v_{3,5}\} \\
\texttt{Active}[S,5] &= \texttt{Influenced}[S,4] \setminus \texttt{Influenced}[S,1] = \{v_0, v_{2,1}, v_{2,2}, v_{2,6}, v_{3,2}\} \\
\texttt{Influenced}[S,5] &= \texttt{Influenced}[S,4] \cup \{v_{3,1}\} = V
\end{aligned}$$

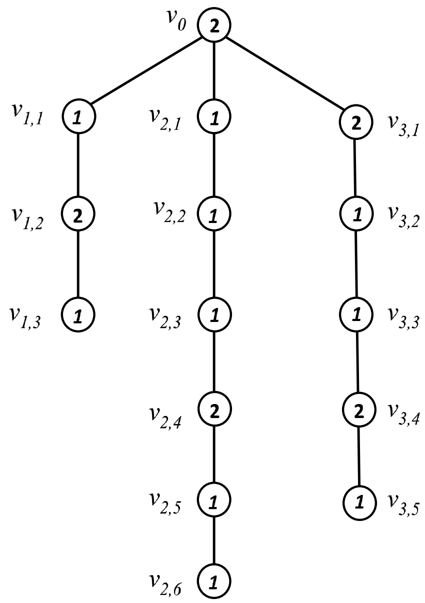

Fig. 1. The tree T, rooted at node v_0, considered in Example 1. The number inside each node is its threshold.

Our results In Section 3, we prove a strong inapproximability result for the TWC–TSS problem under a plausible computational complexity assumption. The result is obtained by a modification of a proof of the inapproximability of TSS by Chen [13]. In view of the strong inapproximability of the TWC–TSS problem, we then turn our attention to special cases of the problem. In Section 4 we present the main results of the paper: exact polynomial time algorithms for paths, rings, and trees. The algorithms for paths and rings are based on dynamic programming. The algorithm for trees is also based on dynamic programming and requires the solution of polynomially many integer linear programs. The polynomial time solvability of each integer linear program is guaranteed by the unimodularity of the associated matrix of coefficients.

3. Hardness of TWC–TSS

In general, our optimization problem TWC–TSS is unlikely to be efficiently approximable, as the following result shows.

The following theorem is a generalization of a similar inapproximability result given in [13] for the Target Set Selection problem that, as said before, corresponds to our Time Window Constrained Target Set Selection problem when the time window size λ is unbounded. Our result holds for any fixed value of λ.

Theorem 1. *For any fixed value of the time window size λ, the TWC–TSS problem cannot be approximated within a ratio of $O(2^{\log^{1-\epsilon} n})$ for any fixed $\epsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$.*

Proof. We prove the theorem by a polynomial time reduction from the same `MIN REP` problem used in [13]. Let $G = (V, E)$ be a bipartite graph, where $V = V_A \cup V_B$, $V_A \cap V_B = \emptyset$, and $E \subseteq V_A \times V_B$. Let $\mathcal{A}$ be a family of subsets of V_A that partitions V_A into $|\mathcal{A}|$ equally sized subsets, and analogously let the family $\mathcal{B}$ be a partition of V_B into $|\mathcal{B}|$ equally sized subsets. Given graph G and partitions $\mathcal{A}$, $\mathcal{B}$, the `MIN REP` problem asks for a subset $U \subseteq V$ of minimum size such that for each $A \in \mathcal{A}$ and $B \in \mathcal{B}$

$$E \cap (A \times B) \neq \emptyset \text{ implies } [E \cap (A \times B)] \cap (U \times U) \neq \emptyset. \tag{4}$$

Given an instance $\mathcal{I}$ of `MIN REP` consisting of the bipartite graph $G = (V, E)$ and the pair of partitions $(\mathcal{A}, \mathcal{B})$, we construct an instance $\mathcal{I}'$ of the TWC–TSS problem. More precisely, we will construct a suitable graph $G' = (V', E')$ and threshold function $t : V' \longrightarrow \mathbb{N} = \{1, 2, \ldots, \}$, but we will not fix λ since our aim is to prove inapproximability for *any* value of λ.

We denote by Γ_ℓ the gadget shown in Fig. 2(a), which consists of ℓ paths of length 2 connecting the same pair of nodes. If $\lambda \leq 8$, we need another gadget Γ_ℓ^λ which is shown in Fig. 2(b); it consists of ℓ paths, each having length $11 - \lambda$ and connecting the same pair of extremal nodes. All internal nodes of the gadgets have threshold 1.

Now, given an instance $\mathcal{I}$ of `MIN REP`, we construct an instance $\mathcal{I}'$ of TWC–TSS as follows. Let $N = |V| + |E|$. The graph G' (cf. Fig. 3) consists of the node sets V_1, V_2, V_3, V_4 and the connecting gadgets as follows:

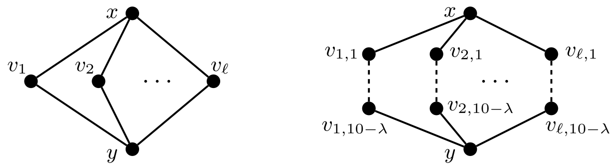

Fig. 2. (a) The gadget Γ_ℓ consisting of ℓ paths of length 2 sharing the extremal nodes. (b) The gadget Γ_ℓ^λ consisting of ℓ paths of length $11-\lambda$ sharing the extremal nodes.

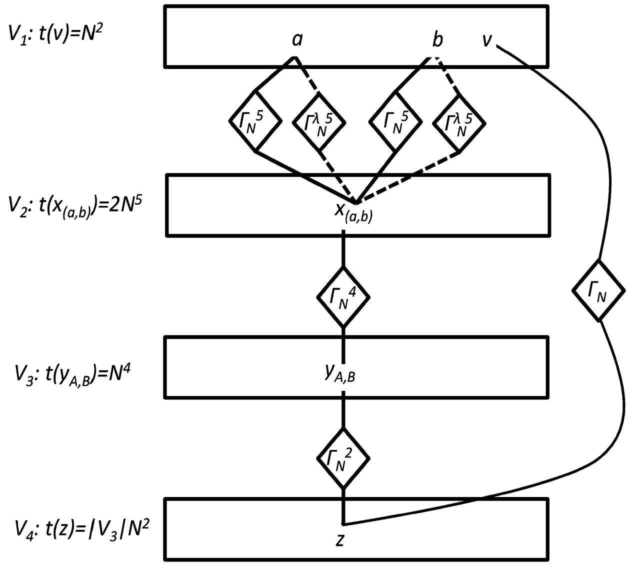

Fig. 3. The construction of the graph G' of instance I'.

- $V_1 = V$ and each node has threshold N^2.
- $V_2 = \{x_{(a,b)} : (a,b) \in E\}$; each node $x_{(a,b)} \in V_2$ has threshold $2N^5$ and is connected to each of $a \in V_1$ and $b \in V_1$ by a gadget Γ_{N^5}. If $\lambda \le 8$, each $x_{(a,b)}$ is also connected to each of $a \in V_1$ and $b \in V_1$ by a gadget $\Gamma^\lambda_{N^5}$.
- $V_3 = \{y_{A,B} : A \in \mathcal{A}, B \in \mathcal{B}, (A \times B) \cap E \neq \emptyset\}$; each node $y_{A,B} \in V_3$ has threshold N^4 and is connected by a gadget Γ_{N^4} to each $x_{(a,b)} \in V_2$ with $a \in A$ and $b \in B$.
- $V_4 = \{z_1, \ldots, z_N\}$; each node $z \in V_4$ has threshold $|V_3|N^2$ and is connected by a gadget Γ_{N^2} to each node in V_3 and by a gadget Γ_N to each node in V_1.

We shall show that any optimal solution U of the $\texttt{MIN REP}$ instance $\mathcal{I}$ gives a solution $U \subseteq V_1$ for the TWC–TSS instance $\mathcal{I}'$ and that the size of any optimal solution for the $\texttt{MIN REP}$ instance $\mathcal{I}$ is at most twice the size of an optimal solution for the TWC–TSS instance $\mathcal{I}'$.

First, suppose that U is an optimal $\texttt{MIN REP}$ solution for $\mathcal{I}$ and consider $U \subseteq V_1$ in the TWC–TSS instance $\mathcal{I}'$. Since U is a $\texttt{MIN REP}$ solution, we know that for each $A \in \mathcal{A}$ and $B \in \mathcal{B}$ it holds that

$$E \cap (A \times B) \neq \emptyset \quad \Rightarrow \quad E \cap ((A \cap U) \times (B \cap U)) \neq \emptyset. \tag{5}$$

Now recall the existence of the gadgets Γ_{N^5} between nodes in V_1 and nodes in V_2 and that each node in V_2 has threshold $2N^5$. Hence, we get that $\texttt{Influenced}[U,2] \supseteq \{x_{(a,b)} \in V_2 |\ a, b \in U\}$. From this and (5), we get that the nodes in $\texttt{Influenced}[U,2] \cap V_2$ can influence (through the Γ_{N^4} gadgets between nodes in V_2 and V_3) each node in V_3 in round 4, that is,

$$V_3 \subseteq \texttt{Influenced}[U,4].$$

This allows all the nodes in V_4 to become influenced in round 6, that is,

$$V_4 \subseteq \texttt{Influenced}[U,6].$$

Hence, $V_4 \subseteq \texttt{Active}[U,7]$ and the remaining nodes in V_1, namely $V_1 - U$, become influenced in round 8.

Therefore, all nodes in V_1 have become influenced either in round 0 or in round 8 and

$$V_1 \subseteq (\texttt{Active}[U,1] \cup \ldots \cup \texttt{Active}[U,\lambda]) \cup \texttt{Active}[U,9].$$

The influence of the nodes in $V_1 - U$ reaches the nodes in V_2 in round 10 through Γ_{N^5} gadgets. If $\lambda > 8$, then the nodes in $U \subseteq V_1$ are still active in round 9 and can influence nodes in V_2 in round 10 through Γ_{N^5} gadgets. If $\lambda \le 8$, then the influence of the nodes in U reaches the nodes in V_2 in round $11 - \lambda$ through $\Gamma^{\lambda}_{N^5}$ gadgets and this influence remains until round $11 - \lambda + (\lambda - 1) = 10$. In either case, each uninfluenced node $x_{(a,b)} \in V_2$ has at least N^5 neighbours that belong to $\texttt{Active}[U,10]$ for both $a, b \in V_1$. Hence, $V_2 \subseteq \texttt{Influenced}[U,10]$. In conclusion, $V_1 \cup V_2 \cup V_3 \cup V_4 \subseteq \texttt{Influenced}[U,10]$. Recalling that each gadget node has threshold 1, we have that each node in V' becomes influenced, thus completing the process.

Now we want to show that if we have a solution S to TWC–TSS, then we can construct a solution to MIN REP of size at most $2|S|$.

We start by showing that we can assume $S \subset V_1 \cup V_4$. To this aim, we first notice that the influence process can only proceed from nodes in V_1 to nodes in V_2, from nodes in V_2 to nodes in V_3, and from nodes in V_3 to nodes in V_4; namely we have that:

a) A node in V_4 can become influenced at some round r only if *all* nodes in V_3 are active in round $r - 2$ (otherwise the threshold $|V_3|N^2$ cannot be reached, independently of which nodes in V_1 are active; indeed if there exists a node in V_3 that is not active at round $r - 2$, then the sizes of the gadgets between a node $z \in V_4$ and nodes in V_3 and V_1 implies that at round $r - 1$ at most $(|V_3| - 1)N^2 + |V_1|N < |V_3|N^2$ neighbours of z can be active).

b) A node $y_{A,B} \in V_3$ can become influenced at some round r only if there exist $a \in A$ and $b \in B$ such that the node $x_{(a,b)} \in V_2$ is active in round $r - 2$ (otherwise the threshold N^4 cannot be reached independently of which nodes in V_4 are active).

c) A node $x_{(a,b)} \in V_2$ can become influenced in some round r only if both a and b ($a, b \in V_1$) are active in round $r - 2$ if $\lambda > 8$ and in round $r - 2$ or $r - (11 - \lambda)$ if $\lambda \le 8$ (otherwise the threshold $2N^5$ cannot be reached independently of which nodes in V_3 are active).

Moreover:

- We can assume that S does not contain any node internal to any gadget; indeed, due to the thresholds of the extremal nodes of the gadgets, such nodes could be useful only if ℓ nodes in a gadget Γ_ℓ (or Γ^{λ}_{ℓ}) are in S, but any gadget has $\ell \ge N > |V|$.
- We can assume that $S \cap V_3 = \emptyset$. Suppose to the contrary that $S \cap V_3 \ne \emptyset$. By c), nodes in $S \cap V_3$ cannot contribute to the influencing of any node in V_2. Moreover, by b) we can replace each $y_{A,B} \in S \cap V_3$ by a node $x_{(a,b)} \in V_2$ for some $a \in A$ and $b \in B$ to obtain a new feasible solution having size at most $|S|$ and empty intersection with V_3.
- If $S \cap V_2 = \emptyset$ then, using c), we can replace each $x_{(a,b)} \in S \cap V_2$ by the two nodes a and b in V_1. This gives a feasible solution with size at most $2|S|$.

Summarizing, we can assume the existence of a feasible solution S' of size $|S'| \le 2|S|$ such that $S' \subset V_1 \cup V_4$. Notice that V_1 is a trivial solution, so we can assume that $|S| \le |V| < N$, and that each node in V_1 has threshold N^2. Hence, we get that nodes in $S' \cap V_4 = S \cap V_4$ cannot influence any node in $V_1 - S$. By a), b), and c), this implies that $S' \cap V_1$ can influence all nodes in V' and is a solution for the MIN REP instance $\mathcal{I}$. $\square$

4. Polynomially solvable cases of TWC–TSS

We now present exact polynomial time algorithms to solve the TWC–TSS problem in some classes of graphs.

4.1. Paths

Let $L^n = (V, E)$ be a path on n nodes, with $V = \{0, \ldots, n-1\}$ and $E = \{(i, i+1) : 0 \le i \le n-2\}$. Since the threshold of each node cannot exceed its degree, we have that $t(0) = t(n-1) = 1$ and $t(i) \in \{1, 2\}$, for each $i = 1, \ldots, n-2$.

The TWC–TSS problem is trivial to solve in case λ is unbounded. Letting $\{i_1, i_2, \ldots, i_m\}$ be the nodes of L^n having threshold equal to 2, one can see that $\{i_1, i_3, \ldots, i_{m-2}, i_m\}$ is an optimal solution when m is odd, whereas the subset $\{i_1, i_3, \ldots, i_{m-1}, i_m\}$ is optimal when m is even. In case λ has some fixed value, the situation is much more complicated. Indeed, because of the time window constraint, one must judiciously choose the initial target set in such a way that, for every node with threshold 2 that does not belong to the initial target set, its two neighbours become influenced at the correct times.

To avoid trivialities, we assume that L^n has at least two nodes with threshold equal to 2. Should it be otherwise, for instance all nodes have threshold 1, then any subset S of V with $|S| = 1$ is an optimal solution. If exactly one node, say i, has threshold 2, then $\{i\}$ is an optimal solution.

Lemma 1. *If $\ell = \min\{i \in V : t(i) = 2\}$ and $s = \max\{i \in V : t(i) = 2\}$, then there exists an optimal solution S such that*

a) $S \cap \{0, \dots \ell - 1\} = \emptyset = S \cap \{s + 1, \dots n - 1\}$;
b) $\ell, s \in S$.

Proof. Since $t(\ell) = 2 = t(s)$, we immediately get that

$$S \cap \{0, \dots, \ell\} \neq \emptyset \text{ and } S \cap \{s, \dots, n-1\} \neq \emptyset. \tag{6}$$

It is easy to see that if $i \in S$ with $i < \ell$ (resp. $i > s$), then the set $S' = S - \{i\} \cup \{\ell\}$ (resp. $S' = S - \{i\} \cup \{s\}$) is also an optimal solution. Therefore, we can assume that S satisfies a). If S is an optimal solution satisfying a), then a) and (6) imply that S satisfies b). □

Lemma 1 implies that we can ignore all nodes in L^n that are either to the left of the lowest numbered node with threshold 2, or to the right of the highest numbered node with threshold 2, when constructing an optimal solution. To simplify the notation in the following proofs, we will remove these nodes leaving a path L^n with left and right extremal nodes having degree 1 and threshold 2, that is,

$$t(0) = t(n-1) = 2.$$

For each $i = 0, \dots, n-1$, let L^n_i denote the sub-path consisting of the last $n - i$ nodes $\{i, i+1, \dots, n-1\}$ of L^n. We denote by $\mathtt{s}(i)$ the minimum size of a TWC target set for L^n_i that contains both of the extremal nodes, that is, i and $n-1$. Our goal is to compute $\mathtt{s}(0)$, the size of an optimal solution for $L^n_0 = L^n$.

Let us define the array $D[0 \dots (n-1)]$, where $D[n-1] = n-1$ and for each $0 \le i < n-1$,

$$D[i] = \min\{j : i < j \le n-1 \text{ and } t(j) = 2\}. \tag{7}$$

Since $t(n-1) = 2$, the value $D[i]$ is always well defined.

One can check that the following algorithm computes an array D satisfying (7).

Algorithm ARRAY(L^n) *[Input: A path L^n with threshold function $t(\cdot)$]*
1. $j = n-1$
2. $D[n-1] = n-1$
3. for $i = (n-2)$ **down to** 0 **do**
4. $D[i] = j$
5. **if** $t(i) = 2$ **then** $j = i$

Lemma 2. *Fix the time window size λ and consider the family of all TWC target sets for L^n_i that include both node i and node $n-1$. If $i < n-1$, then such a family contains a minimum size TWC target set whose second smallest element belongs to the set*

$$\big\{D[i]\big\} \cup \big\{x : \max\{D[i]+1,\ 2D[i]-i-\lambda+1\} \le x \le \min\{2D[i]-i+\lambda-1,\ D[D[i]]\}\big\}. \tag{8}$$

Proof. Let S denote a minimum size solution for L^n_i among all the TWC target sets containing both i and $n-1$. Clearly, the smallest element in S is i. Let j be the second smallest element of S and assume by contradiction that j does not belong to the set in (8). We distinguish four cases based on the value of j. These cases cover all possible values of j for each possible ordering of the quantities defining the set in (8); in each case we derive a contradiction to the assumption that S is a minimum size solution for L^n_i.

Case $0 < j < D[i]$: In this case, the set $S' = S - \{j\} \cup \{D[i]\}$ is a TWC target set of size at most $|S|$ satisfying the claim. Indeed, by definition of the vector D, any node x, $i < x < D[i]$, has $t(x) = 1$. Since $i \in S$, we have that x is influenced no later than round $x - i$, that is, $x \in \mathtt{Influenced}[S', x-i]$. Moreover, the influencing of any node $y > D[i]$ cannot depend on the presence of j in the target set, that is, $S - \{j\} = S' - \{D[i]\}$ is a TWC target set for $L^{n-1}_{D[i]+1}$.

Case $D[i] < j < 2D[i] - i - \lambda + 1$: In this case, we have that $D[i] \notin S$ (since $0 < D[i] < j$) and $t(D[i]) = 2$. Therefore, to become influenced, node $D[i]$ needs two neighbours that are active in the same round. Each node $x = i+1, \dots, D[i]-1$ has threshold 1 and becomes influenced in round $x - i$ because $i \in S$, that is, $x \in \mathtt{Influenced}[S, x-i]$. In particular, node $D[i]-1$ becomes influenced in round $D[i]-1-i$ and remains active for the λ subsequent rounds $D[i]-i, \dots, D[i]-i+\lambda-1$, that is,

$$(D[i]-1) \in \bigcap_{r=D[i]-i}^{D[i]-i+\lambda-1} \mathtt{Active}[S, r], \text{ and} \tag{9}$$

$$(D[i]-1) \notin \mathtt{Active}[S, r] \text{ for } r \le D[i]-i-1 \text{ and } r \ge D[i]-i+\lambda. \tag{10}$$

Analogously, each node y with $D[i] < y < j$ becomes influenced because $j \in S$, so $y \in \mathtt{Influenced}[S, j-y]$. In particular, node $D[i]+1$ becomes influenced in round $j - D[i] - 1$ and remains active for λ rounds, that is,

$$(D[i]+1) \in \bigcap_{r=j-D[i]}^{j-D[i]+\lambda-1} \mathtt{Active}[S,r] \text{ and } (D[i]+1) \notin \mathtt{Active}[S, j-D[i]+\lambda]. \tag{11}$$

The hypothesis of this case implies that $j - D[i] + \lambda - 1 < (2D[i] - i - \lambda + 1) - D[i] + \lambda - 1 = D[i] - i$. This inequality, together with (9) and (10), imply that there is no round r such that both $D[i]-1$ and $D[i]+1$ belong to $\mathtt{Active}[S,r]$; thus node $D[i]$ cannot become influenced and we have reached a contradiction.

Case $2D[i] - i + \lambda - 1 < j < D[D[i]]$: As already seen in the previous case, node $D[i]-1$ becomes influenced in round $D[i] - 1 - i$ and remains active for the subsequent λ rounds as stated in (9) and (10). On the other side, node $D[i]+1$ becomes influenced in round $j - D[i] - 1$, and

$$(D[i]+1) \in \mathtt{Active}[S, j-D[i]] \text{ but } (D[i]+1) \notin \mathtt{Active}[S,r], \forall r \le j - D[i] - 1.$$

The hypothesis of this case implies that $j - D[i] > (2D[i] - i + \lambda - 1) - D[i] = D[i] - i + \lambda - 1$. This, together with (9) and (10), imply that there is no round when both neighbours of $D[i]$ are active; thus node $D[i]$ cannot become influenced. Again a contradiction.

Case $j > D[D[i]]$: We have that no node x with $i < x < j$ belongs to S. In particular, $D[i]$ and $D[D[i]]$ are two nodes with threshold 2 such that $i < D[i] < D[D[i]] < j$. This clearly prevents the two nodes $D[i]$ and $D[D[i]]$ from becoming influenced. □

Clearly, node i must belong to any target set for L_i^n. Hence, $\mathtt{s}(n-1) = 1$. Moreover, Lemma 2 implies that there is an optimal target set that contains the node $\min\left\{\mathtt{s}(D[i]), \min_j \mathtt{s}(j)\right\}$, corresponding to the minimum value in the set in (8), and does not contain any node from $i+1$ to $\min\left\{\mathtt{s}(D[i]), \min_j \mathtt{s}(j)\right\} - 1$. Hence,

$$\mathtt{s}(i) = 1 + \min\left\{\mathtt{s}(D[i]), \min_j \mathtt{s}(j)\right\} \tag{12}$$

where j satisfies $\max\left\{D[i]+1, 2D[i] - i - \lambda + 1\right\} \le j \le \min\left\{2D[i] - i + \lambda - 1, D[D[i]]\right\}$.

Lemma 3. *For each node $d \in \{0, \dots, n-1\}$ of L^n with $t(d) = 2$ there exists an integer $\mathtt{s}_d$ such that for each $i = d, \dots, D[d]$ it holds $\mathtt{s}(i) \in \{\mathtt{s}_d, \mathtt{s}_d - 1\}$.*

Proof. Let $d_1 = 0 < d_2 < \dots < d_m = (n-1)$ be the nodes of L^n with threshold 2. For each $\ell = 2, \dots, m$ and $i = d_{\ell-1}, \dots, d_\ell - 1$, we have that $D[i] = D[d_{\ell-1}] = d_\ell$ and the value of the function $\mathtt{s}(i)$ is obtained as in (12) where the minimum is computed in the range

$$\max\left\{d_\ell + 1, 2d_\ell - i - \lambda + 1\right\} \le j \le \min\left\{2d_\ell - i + \lambda - 1, d_{\ell+1}\right\}. \tag{13}$$

By definition we have $\mathtt{s}(d_m) = \mathtt{s}(n-1) = 1$ and we can fix $\mathtt{s}_{d_m} = 1$. We prove now, by induction on ℓ from $m-1$ down to 1, that there exists a value $\mathtt{s}_{d_\ell}$ such that

$$\mathtt{s}_{d_\ell} - 1 \le \mathtt{s}(i) \le \mathtt{s}_{d_\ell} \tag{14}$$

for each integer $i = d_\ell, \dots, d_{\ell+1}$.

By (12), we have that $\mathtt{s}(i) = \mathtt{s}(d_m) + 1 = 2$ for each $i = d_{m-1}, \dots, d_m - 1$. Hence we can fix $s_{d_{m-1}} = 2$ and (14) holds for $\ell = m-1$.

Now suppose that (14) is true for each integer from ℓ to m; we prove it true for $\ell - 1 \le m - 2$.

By (13) and using the inductive hypothesis on ℓ, we have

$$\mathtt{s}_{d_\ell} \le \mathtt{s}(i) \le \mathtt{s}_{d_\ell} + 1$$

for any $i = d_{\ell-1}, \dots, d_\ell - 1$. It follows that the value $\mathtt{s}_{d_{\ell-1}} = \max\{\mathtt{s}(i) \mid d_{\ell-1} \le i \le d_\ell - 1\}$ satisfies (14) for $i = d_{\ell-1}, \dots, d_\ell - 1$.

It remains to show that $\mathtt{s}_{d_{\ell-1}} - 1 \le \mathtt{s}(d_\ell) \le \mathtt{s}_{d_{\ell-1}}$. We notice that if $\mathtt{s}(d_\ell) = \mathtt{s}_{d_\ell} - 1$, then by (13) we have $\mathtt{s}(i) = \mathtt{s}(d_\ell) + 1 = \mathtt{s}_{d_\ell}$ for each $i = d_{\ell-1}, \dots, d_\ell - 1$. Hence, we have that $\mathtt{s}_{d_{\ell-1}} = \mathtt{s}_{d_\ell} = \mathtt{s}(d_\ell) + 1$ and (14) holds for $i = d_\ell$.

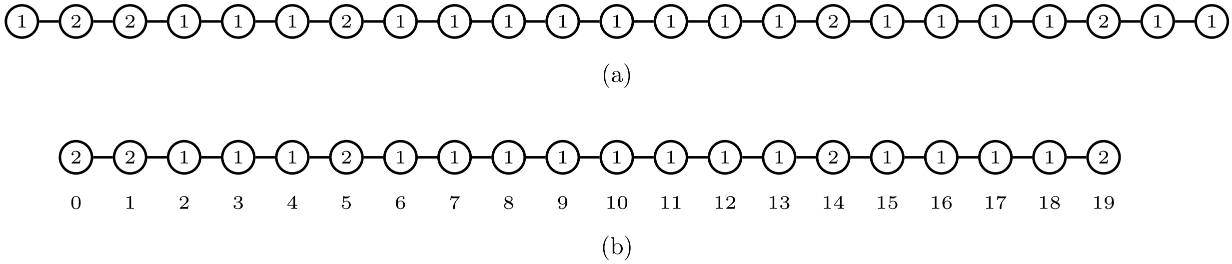

Fig. 4. (a) A path on 23 nodes; (b) the pruned path L^{20}.

Theorem 2. *For any time window size λ, an optimal TWC target set for the path L^n can be computed in time $O(n)$.*

Proof. The size of an optimal target set for L_n can be computed as $\mathtt{s}(0)$. By Lemma 3, one can implement the computation in (13) as shown in the following algorithm LINE(L^n). In the algorithm, the variable d represents the current node of threshold 2 and the variable $\mathtt{s}$ has value $\mathtt{s}_d$ (cf. Lemma 3). To carry out the computation, we consider the array $\mathtt{J}$ such that for any i the value $\mathtt{J}[i]$ represents the maximum index j with $d \le j \le i$ such that $\mathtt{s}(j) = \mathtt{s} - 1$ (if none exists, we define $\mathtt{J}[i] = d$). This value is used to check if any of the values needed to compute $\min_j \mathtt{s}(j)$ in (12) is $\mathtt{s}_d - 1$, that is, the smaller of the two possibilities according to Lemma 3.

Algorithm: LINE(L^n) *[Input: A path L^n and the array D satisfying (7)]*

1. $\mathtt{J}[n-1] = n-1,\ \ \mathtt{s}(n-1) = 1$ *[Here d is the last node $n-1$]*

2. $d = \min\{i \mid D[i] = n-1\}$ *[Now d is the penultimate node of threshold 2*
3. $\mathtt{s}(d) = 2,\ \ \mathtt{s} = 2$ *and we can set $\mathtt{s}(i)$ to 2 for each i up to $n-2$;*
4. for $i = d+1$ **to** $D[d]-1$ *moreover, we set $\mathtt{J}[i] = i$ since all $\mathtt{s}(i)$ have*
5. $\mathtt{s}(i) = 2,\ \ \mathtt{J}[i] = i$ *the same value s]*

6. while $d > 0$ **do**
7. $d' = d$
8. $d = \min\{i \mid D[i] = d'\}$
9. **for** $i = d$ **to** $D[d]-1$
10. *[We compute $\mathtt{s}(i) \in \{\mathtt{s}, \mathtt{s}+1\}$ as in (13). Recall that here $D[i] = D[d]$]*
11. **If** $\mathtt{s}(D[i]) = \mathtt{s} - 1$ **then** $\mathtt{s}(i) = \mathtt{s}$
12. **else**
13. Set $j_{\min} = \max\left\{D[i]+1,\ 2D[i] - i - \lambda + 1\right\}$
14. Set $j_{\max} = \min\left\{2D[i] - i + \lambda - 1,\ D[D[i]]\right\}$
15. **If** $(\mathtt{s}(j_{\max}) = \mathtt{s} - 1$ OR $j_{\min} \le \mathtt{J}[j_{\max}])$ **then** $\mathtt{s}(i) = \mathtt{s}$ **else** $\mathtt{s}(i) = \mathtt{s} + 1$
16. $\mathtt{s} = \max\{\mathtt{s}(i) \mid d \le i \le D[d]\}$
17. *[We set the values of the vector L in the range $i = d, \ldots, D[d]$.*
18. *Note that all values in the initial part of the interval are set to d as long as $\mathtt{s}(i) = \mathtt{s}$]*
19. $\mathtt{J}[d] = d$
20. **For** $i = d+1$ **to** $D[d]$
21. **If** $\mathtt{s}(i) = \mathtt{s} - 1$ **then** $\mathtt{J}[i] = i$ **else** $\mathtt{J}[i] = \mathtt{J}[i-1]$.

The algorithm takes time $O(1)$ to compute each $\mathtt{s}(i)$, $i = n-1, \ldots, 0$, for a total of $O(n)$. The actual TWC target set of optimal size $\mathtt{s}(0)$ can be constructed using standard backtracking techniques. $\square$

Example 2. Consider the path in Fig. 4(a). The threshold of each node is indicated inside the circle representing the node. Pruning the extremal nodes of threshold 1, we get a path of 20 nodes. Naming them from 0 to 19, we get the path L^{20} in Fig. 4(b) where the nodes of threshold 2 are $d_1 = 0$, $d_2 = 1$, $d_3 = 5$, $d_4 = 14$, and $d_5 = 19$; hence, $D[18] = D[17] = \ldots = D[14] = 19$, $D[13] = \ldots = D[5] = 14$, $D[4] = \ldots = D[1] = 5$, $D[0] = 1$.

Let the time window size be $\lambda = 2$. Denote by $\mathtt{s}(i)$ the cardinalities of the optimal solutions to the subproblems and by S_i a corresponding optimal set (recall that in general S_i is not unique). We have

$$\begin{array}{lll}
\mathtt{s}(19) = 1 & \text{and} & S_{19} = \{19\}, \\
\mathtt{s}(i) = 1 + \mathtt{s}(19) = 2, & \text{and} & S_i = \{i, 19\}, \ \text{ for } i = 18, 17, \ldots, 14, \\
\mathtt{s}(i) = 1 + \mathtt{s}(14) = 3, & \text{and} & S_i = \{i, 14, 19\}, \ \text{ for } i = 13, 12, 11, \\
\mathtt{s}(i) = 1 + \mathtt{s}(19) = 2, & \text{and} & S_i = \{i, 19\}, \ \text{ for } i = 10, 9, 8, \\
\mathtt{s}(i) = 1 + \mathtt{s}(14) = 3, & \text{and} & S_i = \{i, 14, 19\}, \ \text{ for } i = 7, 6, 5, \\
\mathtt{s}(4) = 1 + \mathtt{s}(5) = 4, & \text{and} & S_4 = \{4, 5, 14, 19\}, \\
\mathtt{s}(3) = 1 + \mathtt{s}(8) = 3, & \text{and} & S_3 = \{3, 8, 19\}, \\
\mathtt{s}(2) = 1 + \mathtt{s}(8) = 3, & \text{and} & S_2 = \{2, 8, 19\}, \\
\mathtt{s}(1) = 1 + \mathtt{s}(9) = 3, & \text{and} & S_1 = \{1, 9, 19\}, \\
\mathtt{s}(0) = 1 + \mathtt{s}(3) = 4, & \text{and} & S_0 = \{0, 3, 8, 19\}.
\end{array}$$

Hence, the optimal size of a TWC target set for the path is $\mathtt{s}(0) = 4$ and an optimal set is $S_0 = \{0, 3, 8, 19\}$.

4.2. Rings

We can use Theorem 2 above to design an algorithm for the TWC–TSS problem on rings. Let R^n denote the ring on n nodes $\{0, \ldots, n-1\}$ with edges $(i, (i+1) \bmod n)$ and thresholds $t(i)$, for $i = 0, \ldots, n-1$.

We first notice that if all nodes have threshold 2, then an optimal TWC target set for R^n trivially has size $\lceil n/2 \rceil$, so let us now assume that there exists a node j that has threshold $t(j) = 1$. Either j is in an optimal TWC target set for R^n or it is not. Consider the path $R^n_{j,2}$ obtained by "breaking" the ring R^n at node j, duplicating node j into j and j', and assigning threshold 2 to both j and j' (regardless of the original threshold value $t(j) = 1$ in R^n). Therefore, the edges of $R^n_{j,2}$ are $(j, j+1), (j+1, j+2), \ldots, (n-2, n-1), (n-1, 0), \ldots (j-2, j-1), (j-1, j')$. The thresholds of $R^n_{j,2}$ are

$$t_{j,2}(i) = \begin{cases} t(i) & \text{if } 0 \le i \le n-1 \text{ and } i \ne j \\ 2 & \text{if } i = j \text{ or } i = j'. \end{cases}$$

We can use the algorithm of Section 4.1 to compute the size of an optimal TWC target set $S_{j,2}$ for the path $R^n_{j,2}$. Notice that both j and j' must be in $S_{j,2}$, so $S_{j,2} - \{j'\}$ is a TWC target set for the ring R^n, optimal among all TWC target sets that include node j.

Now we want to compute a TWC target set for the ring R^n that is optimal among all TWC target sets that do not include node j. To do this, consider the path $R^n_{j,1}$ that has the same nodes and edges as $R^n_{j,2}$ but has thresholds

$$t_{j,1}(i) = \begin{cases} t(i) & \text{if } 0 \le i \le n-1 \text{ and } i \ne j \\ 1 & \text{if } i = j \text{ or } i = j'. \end{cases}$$

In particular, the endpoints of $R^n_{j,1}$ have thresholds $t_{j,1}(j) = t_{j,1}(j') = 1$. First, we apply Lemma 1 to $R^n_{j,1}$ and then we use the algorithm of Section 4.1 to compute the size of an optimal TWC target set $S_{j,1}$. Since $j, j' \notin S_{j,1}$, we have that $S_{j,1}$ is a TWC target set for the ring R^n, optimal among all TWC target sets that do not include node j.

An optimal solution for the ring R^n is then obtained by choosing the smaller of $S_{j,2} - \{j'\}$ and $S_{j,1}$. In conclusion we have the following result.

Theorem 3. *For any value of the time window size λ, an optimal TWC target set for the ring R^n can be computed in time $O(n)$.*

4.3. Trees

Let $T = (V, E)$ be a tree with threshold function $t : V \longrightarrow \mathbb{N}$, and let $\lambda \ge 1$ be a fixed value of the time window size. We consider T to be rooted at some arbitrary node $p \in V$. For each node $v \in V$, we denote by $T_v = (V_v, E_v)$ the subtree of T rooted at v. Moreover, we denote by $\mathtt{Ch}(v)$ the set of all children of node v in T_v.

Definition 1. Given node $v \in V$ and integers t, r, with $t \in \{t(v), t(v) - 1\}$ and $r \ge 0$, we denote by $\mathtt{s}(v, t, r)$ the minimum size of a TWC target set $S \subseteq V_v$ for subtree T_v that influences node v in round r (that is, $v \in \mathtt{Influenced}[S, r] \setminus \mathtt{Influenced}[S, r-1]$), under the assumption that v has threshold t in T_v. The threshold of each other node $w \ne v$ in T_v is its original one $t(w)$.

The size of an optimal TWC target set for the tree T can be computed as

$$\min_r \mathtt{s}(p, t(p), r), \tag{15}$$

where r ranges between 0 and the maximum possible number of rounds needed to complete the influence diffusion process. The number of rounds is always upper bounded by the number of nodes in the graph (since at least one new node must be influenced in each round before the diffusion process stops). However, for a tree T, this value is upper bounded by the length of the longest path in T. In other words, the parameter r in Definition 1 is bounded by the diameter $diam(T)$ of T.

We use a dynamic programming approach to compute the value in (15). Then, the corresponding optimal TWC target set S can be built using standard backtracking techniques. In our dynamic programming algorithm we compute all of the values

$$\mathtt{s}(v,t,r) \text{ for each } v \in V, \text{ each } t \in \{t(v), t(v)-1\}, \text{ and } r = 0,\dots,diam(T),$$

and the computation is performed according to a reverse breadth-first search (BFS) ordering of the nodes of T, so that each node v is considered only when all of the values $\mathtt{s}(\cdot,\cdot,\cdot)$ for all of its children are known. The rationale behind the computation of both $\mathtt{s}(v,t(v),r)$ and $\mathtt{s}(v,t(v)-1,r)$ is the following:

i) $\mathtt{s}(v,t(v),r)$ corresponds to the case of a target set S for tree T such that
   - $v \in \mathtt{Influenced}[S,r] \setminus \mathtt{Influenced}[S,r-1]$ and
   - at least $t(v)$ of v's children belong to $\mathtt{Active}[S \cap V_v, r] \subseteq \mathtt{Active}[S,r]$;

ii) $\mathtt{s}(v,t(v)-1,r)$ is the size of an optimal target set S for T satisfying
   - $v \in \mathtt{Influenced}[S,r] \setminus \mathtt{Influenced}[S,r-1]$,
   - $\mathtt{Active}[S,r]$ contains v's parent in T, and
   - at least $t(v)-1$ of v's children belong to $\mathtt{Active}[S \cap V_v, r] \subseteq \mathtt{Active}[S,r]$.

In the following, we show how to compute the above values $\mathtt{s}(\cdot,\cdot,\cdot)$. The procedure is summarized in algorithm TREE.

Algorithm TREE(T, p, λ, t) *[*$\mathtt{Input:}$ *Tree T rooted at p, time window size λ, threshold function t.]*

1. For each $v \in T$ in reverse order to a BFS of T
2. *[We compute $\mathtt{s}(v,t,r)$ for each $t \in \{t(v), t(v)-1\}$ and $0 \le r \le diam(T)$]*
3. **If** v is a leaf of T **then** *[here $t(v)=1$]*
4. **For** $r = 0,\dots,diam(T)$
5. Set $\mathtt{s}(v,0,r)=0$ and $\mathtt{s}(v,1,r) = \begin{cases} 1 & \text{if } r=0 \\ \infty & \text{otherwise} \end{cases}$
6. **If** v is NOT a leaf of T **then**
7. **For** $r=0,\dots,diam(T)$ AND $t \in \{t(v), t(v)-1\}$ (only $t=t(v)$ if $v=p$)
8. **If** $r=0$ OR $t \le 1$ **then**
9. **For** each $w \in \mathtt{Ch}(v)$ compute
10. $min(w) = \min\left\{\min_{r+1 \le j \le r+\lambda} \mathtt{s}(w,t(w)-1,j), \min_{r-1 \le j \le diam(T)} \mathtt{s}(w,t(w),j)\right\}$
11. **If** $r=0$ **then** Set $\mathtt{s}(v,t,0) = 1 + \sum_{w \in \mathtt{Ch}(v)} min(w)$
12. **If** $r \ge 1$ and $t=0$ **then** Set $\mathtt{s}(v,0,r) = \sum_{w \in \mathtt{Ch}(v)} min(w)$
13. **If** $r \ge 1$ and $t=1$ **then**
14. Set $z = \operatorname{argmin}_{w \in \mathtt{Ch}(v)}\{\mathtt{s}(w,t(w),r-1) - min(w)\}$
15. Set $\mathtt{s}(v,1,r) = \sum_{w \in \mathtt{Ch}(v)\setminus\{z\}} min(w) + \mathtt{s}(z,t(z),r-1)$
16. **If** $r \ge 1$ and $t > 1$ **then**
17. Set $\mathtt{s}(v,t,r) = \min \sum_{w \in \mathtt{Ch}(v)} m(w)$, where
18. $m(w) \in \{\mathtt{s}(w,\tau,j) : (\tau = t(w) \text{ AND } j \ge 0) \text{ OR } (\tau = t(w)-1 \text{ AND } r+1 \le j \le r+\lambda)\}$
19. $|\{w : m(w) = \mathtt{s}(w,t(w),j),\ \max\{0, r-\lambda\} \le j \le r-1\}| \ge t$
20. $|\{w : m(w) = \mathtt{s}(w,t(w),j),\ \max\{0, \ell-\lambda\} \le j \le \ell-1\}| < t, \quad \forall \ell = 1,\dots,r-1$

First, consider the computation of $\mathtt{s}(v,t,r)$ when v is a leaf of T in lines **3.** to **5.** In this case we have $t(v) = \deg(v) = 1$.

- If $r=0$, v trivially must belong to the target set since v needs to be active at time 0; hence $\mathtt{s}(v,t,0)=1$.
- If $r>0$ and $t=t(v)=1$, we observe that any TWC target set that influences leaf v at time *exactly* r cannot contain v and, therefore, must influence v's parent at time $r-1$. To do so, we set $\mathtt{s}(v,1,r)=\infty$ in the algorithm; this forces the minimum at line **10.** or **17.** to be reached with threshold $t(v)-1=0$, thus forcing v's parent to be active in round $r-1$.
- If $r>0$ and $t=t(v)-1=0$ then, trivially, $\mathtt{s}(v,0,r)=0$.

Now consider an arbitrary internal node v. Since we process nodes in a reverse BFS order, each child of v has already been processed when the algorithm processes v. If $r=0$, then v must necessarily be in the target set and any $w \in \mathtt{Ch}(v)$ can benefit from this. Therefore, the size $\mathtt{s}(v,t,0)$ of an optimal solution for the subtree T_v (line **11.**) for both $t=t(v)$ and $t=t(v)-1$ is equal to

$$\mathtt{s}(v,t,0) = 1 + \sum_{w \in \mathtt{Ch}(v)} min(w) = 1 + \sum_{w \in \mathtt{Ch}(v)} \min\left\{\min_{1 \le j \le \lambda} \mathtt{s}(w,t(w)-1,j), \min_{0 \le j \le diam(T)} \mathtt{s}(w,t(w),j)\right\}.$$

Notice that we have constrained j to be in the range $1, \dots, \lambda$ in the formula above when w's threshold is $t(w) - 1$. This is correct since v is active and able to influence w only in rounds $j = 1, \dots, \lambda$.

Now, let us consider the computation of $\mathtt{s}(v, t, r)$ with $r \geq 1$, that is, when v is not part of the target set and v is influenced at time r by t of its children (plus its parent if $t = t(v) - 1$). To determine the optimal solution, we need to know the best among the values $\mathtt{s}(w, \tau, j)$ for each $w \in \mathtt{Ch}(v)$ and for all possible values of parameters τ and j, subject to the following two constraints:

1) if $\tau = t(w) - 1$, then $r + 1 \leq j \leq r + \lambda$ (indeed v is active and can influence w only during the λ rounds after it has become influenced, that is, in rounds $j = r + 1, \dots, r + \lambda$),
2) at least t nodes in $\mathtt{Ch}(v)$ are active in round r but at most $t - 1$ are active in any previous round $j \leq r - 1$ (otherwise v would become influenced before the required round r).
   The special case $t = 0$ (line **12.**) can hold only if $t(v) = 1$ and $t = t(v) - 1 = 0$; hence node v must be influenced by its parent at round r and none of its children can be active before round r. Recall that $\mathtt{s}(v, t, r)$ asks for the minimum size of a target set in T_v (or equivalently a target set for T having the smallest possible intersection with V_v) which makes v have at least t active children at exactly round r.

More formally, in the algorithm we compute

$$\mathtt{s}(v, t, r) = \min \sum_{w \in \mathtt{Ch}(v)} m(w)$$

where the following three conditions must be satisfied

$$\begin{aligned}
&m(w) \in \{\mathtt{s}(w, \tau, j) : (\tau = t(w) \text{ AND } j \geq 0) \text{ OR } (\tau = t(w) - 1 \text{ AND } r + 1 \leq j \leq r + \lambda)\},\\
&|\{w : m(w) = \mathtt{s}(w, t(w), j),\ \max\{0, r - \lambda\} \leq j \leq r - 1\}| \geq t,\\
&|\{w : m(w) = \mathtt{s}(w, t(w), j),\ \max\{0, \ell - \lambda\} \leq j \leq \max\{0, \ell - 1\}\}| < t, \forall \ell = 1, \dots, r - 1.
\end{aligned} \tag{16}$$

Lemma 4. *The value in (16) can be computed in polynomial time.*

Proof. Consider the computation of $\mathtt{s}(v, t, r)$. Define the variables $x(w, t, j)$, for each $w \in \mathtt{Ch}(v)$ and for each choice of $t \in \{t(w) - 1, t(w)\}$ and $j \geq 0$ such that if $t = t(w) - 1$ then $r + 1 \leq j \leq r + \lambda$. We can re-write (16) as

Compute

$$\min \sum_{w \in \mathtt{Ch}(v)} \left(\sum_{j \geq 0} x(w, t(w), j) \mathtt{s}(w, t(w), j) + \sum_{j=r+1}^{r+\lambda} x(w, t(w) - 1, j) \mathtt{s}(w, t(w) - 1, j) \right)$$

subject to

$$x(w, t, j) \in \{0, 1\}, \qquad \text{for } w \in \mathtt{Ch}(v),\ t = t(w), t(w) - 1,\ j = 0, \dots, diam(T)$$

$$\sum_{j=0}^{diam(T)} x(w, t(w), j) + \sum_{j=r+1}^{r+\lambda} x(w, t(w) - 1, j) = 1, \qquad \text{for each } w \in \mathtt{Ch}(v)$$

$$\sum_{w \in \mathtt{Ch}(v)} \sum_{j=\ell-\lambda}^{\ell-1} -x(w, t(w), j) \geq -(t - 1), \qquad \text{for each } \ell = \lambda, \dots, r - 1$$

$$\sum_{w \in \mathtt{Ch}(v)} \sum_{j=r-\lambda}^{r-1} x(w, t(w), j) \geq t$$

The coefficient matrix M of the above linear program is shown in Fig. 5. We shall prove that M is totally unimodular. (As a matter of fact, one can also show that the above linear program corresponds to a minimum cost flow problem in an associated auxiliary network). Therefore, the linear program can be optimally solved in polynomial time [39]. It is known that M is totally unimodular iff it satisfies the Ghouila-Houri property: each subset R of rows of M can be partitioned into R_1 and R_2 so that each entry of the vector $\sum_{\mathbf{r} \in R_1} \mathbf{r} - \sum_{\mathbf{r} \in R_2} \mathbf{r}$ belongs to $\{-1, 0, 1\}$ [39].

We first notice that

- the sub-matrix formed by the columns corresponding to variables $x(w, t, j)$ for each fixed node $w \in \mathtt{Ch}(v)$ does not depend on w,
- for any fixed row, all of the non-zero entries have the same sign,

$$
\begin{array}{ccccccccccccc}
\dots & w(0) & w(1) & \dots & w(\lambda-1) & w(\lambda) & \dots & w(r-\lambda-1) & w(r-\lambda) & \dots & w(r-2) & w(r-1) & \underset{j\ge r}{\dots w(j) \dots} & \underset{r\le j\le r+\lambda}{\dots w'(j) \dots}
\end{array}
$$
$$
\begin{pmatrix}
\dots & 1 & 1 & \dots & 1 & 1 & \dots & 1 & 1 & \dots & 1 & 1 & 1 & 1 \\
\dots & -1 & -1 & \dots & -1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
\dots & 0 & -1 & \dots & -1 & -1 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
\ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\dots & 0 & 0 & \dots & 0 & 0 & \dots & -1 & -1 & \dots & -1 & 0 & 0 & 0 \\
\dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & 1 & 1 & 0 & 0
\end{pmatrix}
$$

Fig. 5. The portion of the matrix M made of columns corresponding to variables $x(w, t, j)$ for a fixed node w: here $w(j) = x(w, t(w), j)$ and $w'(j) = x(w, t(w) - 1, j)$.

- some columns of M have exactly one non-zero component and are therefore influential in deciding the Ghouila-Houri property.

We can therefore limit ourselves to considering the simpler matrix M':

$$
M' = \begin{pmatrix}
1 & 1 & \dots & 1 & 1 & \dots & 1 & 1 & \dots & 1 & 1 \\
1 & 1 & \dots & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\
0 & 1 & \dots & 1 & 1 & \dots & 0 & 0 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\
\vdots & \vdots & & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & \dots & 1 & 1 & \dots & 1 & 0 \\
0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & 1 & 1
\end{pmatrix}
$$

We prove that the transpose of M' satisfies the Ghouila-Houri property since the transpose of a totally unimodular matrix is also totally unimodular. Namely, we show that each subset C of columns of M' can be partitioned into C_1 and C_2 so that each entry of the vector $\sum_{\mathbf{c}\in C_1} \mathbf{c} - \sum_{\mathbf{c}\in C_2} \mathbf{c}$ belongs to $\{-1, 0, 1\}$. Let M'' denote the submatrix of M' consisting of the columns in C and let $\mathbf{c}_i$ be the i-th column of M'', for $i = 1, \dots, |C|$. We can simply partition as follows: $\mathbf{c}_1 \in C_1$ and for $i = 2, \dots, |C|$ we add $\mathbf{c}_i$ to C_1 if $\mathbf{c}_{i-1} \in C_2$ and we add $\mathbf{c}_i$ to C_2 if $\mathbf{c}_{i-1} \in C_1$. $\square$

The algorithm TREE(T, p, λ) requires $O(n^2)$ computations of values $\mathtt{s}(v, t, r)$ using (16). Hence we get the following result.

Theorem 4. *For any tree T, the optimal TWC target set can be computed in polynomial time.*

5. Concluding remarks

In this paper we have introduced a new model of information diffusion in social networks in which agents change behaviours on the basis of information collected from their neighbours in a time interval of bounded size, unlike previous models in which agents have unbounded memory. A number of interesting problems remain open. First of all, we would like to know whether our result on trees can be extended to graphs of bounded tree-width. Secondly, it seems to us that a polynomial time algorithm for complete graphs is possible, but it has so far escaped our attempts. It would also be interesting to find sharp upper and lower bounds on the cardinality of a minimum TWC target set, in terms of λ and easily computable parameters of the graph. This goal seems to be ruled out for arbitrary graphs by our Theorem 1, but it could be achievable for interesting classes of graphs (cf. the work for the classical TSS problem in [1]). Finally, the motivation for studying the TWC-TSS problem suggests that it would be interesting to consider a generalization of the model that takes into account different time windows $\lambda(v)$ for the nodes in the graph.

References

[1] E. Ackerman, O. Ben-Zwi, G. Wolfovitz, Combinatorial model and bounds for target set selection, Theoret. Comput. Sci. 411 (2010) 4017–4022.
[2] J. Alba, J.W. Hutchinson, J. Lynch, Memory and decision making, in: T.S. Robertson, H. Kassarjian (Eds.), Handbook of Consumer Behavior, 1991.
[3] R.M. Anderson, R.M. May, Infectious Diseases of Humans: Dynamics and Control, Oxford University Press, 1991.
[4] R.A. Banos, J. Borge-Holthoefer, Y. Moreno, The role of hidden influentials in the diffusion of online information cascades, arXiv:1303.4629v1 [physics.soc-ph], 2013.
[5] A. Barrat, M. Barthélemy, A. Vespignani, Dynamical Processes on Complex Networks, Cambridge University Press, 2012.
[6] C. Bazgan, M. Chopin, A. Nichterlein, F. Sikora, Parameterized approximability of maximizing the spread of influence in networks, in: Proceedings of Computing and Combinatorics, COCOON 2013, in: Lectures Notes in Computer Science, vol. 7936, 2013, pp. 543–554.
[7] O. Ben-Zwi, D. Hermelin, D. Lokshtanov, I. Newman, Treewidth governs the complexity of target set selection, Discrete Optim. 8 (2011) 87–96.
[8] L.M.A. Bettencourt, A. Cintron-Arias, D.I. Kaiser, C. Castillo-Chavez, The power of a good idea: quantitative modeling of the spread of ideas from epidemiological models, Phys. A 364 (2006) 513–536.
[9] S. Bikhchandani, D. Hirshleifer, I. Welch, A theory of fads, customs, and cultural change as informational cascades, J. Polit. Econ. 100 (1992) 992–1026.

[10] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: structure and dynamics, Phys. Rep. 472 (2006) 175–308.
[11] C.C. Centeno, M.C. Dourado, L.D. Penso, D. Rautenbach, J.L. Szwarcfiter, Irreversible conversion of graphs, Theoret. Comput. Sci. 412 (2011) 3693–3700.
[12] D. Centola, M. Macy, Complex contagions and the weakness of long ties, Amer. J. Sociol. 113 (2007) 702–734.
[13] N. Chen, On the approximability of influence in social networks, SIAM J. Discrete Math. 23 (2009) 1400–1415.
[14] J. Chen, G. Iver, A. Pazgal, Limited memory, categorization and competition, Mark. Sci. 29 (July/August) (2010) 650–670.
[15] W. Chen, L.V.S. Lakshmanan, C. Castillo, Information and Influence Propagation in Social Networks, Morgan & Claypool, 2013.
[16] C.Y. Chiang, L.H. Huang, W.T. Huang, H.G. Yeh, The target set selection problem on cycle permutation graphs, generalized Petersen graphs and torus cordalis, arXiv:1112.1313, 2011.
[17] M. Chopin, A. Nichterlein, R. Niedermeier, M. Weller, Constant thresholds can make target set selection tractable, in: Proceedings First Mediterranean Conference on Algorithms, MedAlg 2012, in: LNCS, vol. 7659, 2012, pp. 120–133.
[18] C.-Y. Chiang, L.-H. Huang, B.-J. Li, J. Wu, H.-G. Yeh, Some results on the target set selection problem, J. Comb. Optim. 25 (4) (2013) 702–715.
[19] C.-Y. Chiang, L.-H. Huang, H.-G. Yeh, Target set selection problem for honeycomb networks, SIAM J. Discrete Math. 27 (1) (2013) 310–328.
[20] F. Cicalese, G. Cordasco, L. Gargano, M. Milanič, U. Vaccaro, Latency-bounded target set selection in social networks, in: P. Bonizzoni, V. Brattka, B. Loewe (Eds.), Proc. of Computability in Europe 2013, CiE 2013, in: Lectures Notes in Computer Science, vol. 7921, 2013, pp. 65–77.
[21] F. Comellas, M. Mitjana, J.G. Peters, Broadcasting in small-world communication networks, in: SIROCCO, in: Proceedings in Informatics, vol. 13, Carleton Scientific, 2002, pp. 73–85.
[22] A. Coja-Oghlan, U. Feige, M. Krivelevich, D. Reichman, Contagious sets in expanders, arXiv:1306.2465 [cs.DM], 2013.
[23] D.J. Daley, D.G. Kendall, Epidemics and rumours, Nature 204 (1964) 1118.
[24] L. Duenas-Osorio, S.M. Vemuru, Cascading failures in complex infrastructure systems, Struct. Saf. 31 (2) (2009) 157–167.
[25] P. Domingos, M. Richardson, Mining the network value of customers, in: Proc. of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 57–66.
[26] P.S. Dodds, D.J. Watts, A generalized model of social and biological contagion, J. Theoret. Biol. 232 (2005) 587–604.
[27] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.
[28] P. Flocchini, R. Královic, P. Ruzicka, A. Roncato, N. Santoro, On time versus size for monotone dynamic monopolies in regular topologies, J. Discrete Algorithms 1 (2003) 129–150.
[29] F. Karimi, P. Holme, Threshold model of cascades in temporal networks, arXiv:1207.1206v2 [physics.soc-ph], 2012.
[30] D. Kempe, J.M. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proc. of the Ninth ACM SIGKDD International Conference on Knowledge, Discovery and Data Mining, 2003, pp. 137–146.
[31] D. Kempe, J.M. Kleinberg, E. Tardos, Influential nodes in a diffusion model for social networks, in: Proc. of the 32nd International Conference on Automata, Languages and Programming, ICALP'05, in: LNCS, vol. 3580, 2005, pp. 1127–1138.
[32] H. Leskovic, L.A. Adamic, B.A. Huberman, The dynamic of viral marketing, ACM Trans. Web 1 (2007).
[33] M. Leppaniemi, H. Karjaluoto, H. Lehto, A. Goman, Targeting young voters in a political campaign: empirical insights into an interactive digital marketing campaign in the 2007 Finnish general election, J. Nonprofit Public Sect. Mark. 22 (2010) 14–37.
[34] D.P. Maki, M. Thompson, Mathematical Models and Applications, with Emphasis on the Social, Life and Management Sciences, Prentice Hall, 1973.
[35] M. Nekovee, Y. Moreno, G. Bianconi, M. Marsili, Theory of rumour spreading in complex social networks, Phys. A 374 (2007) 467–470.
[36] D. Peleg, Local majorities, coalitions and monopolies in graphs: a review, Theoret. Comput. Sci. 282 (2002) 231–257.
[37] T.V.T. Reddy, C.P. Rangan, Variants of spreading messages, J. Graph Algorithms Appl. 15 (5) (2011) 683–699.
[38] J.-B. Rival, J. Walach, The use of viral marketing in politics: a case study of the 2007 French presidential election, Master Thesis, Jönköping University, Jönköping International Business School, 2009, permanent link: http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-9664.
[39] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Springer, 2004.
[40] K. Tumulty, Obama's viral marketing campaign, TIME Mag. (2007), July 5.
[41] J. Ugander, L. Backstrom, C. Marlow, J. Kleinberg, Structural diversity in social contagion, Proc. Nat. Acad. Sci. 109 (2012) 5962–5966.
[42] M. Zaker, On dynamic monopolies of graphs with general thresholds, Discrete Math. 312 (6) (2012) 1136–1143.