Whom to befriend to influence people [*]

Manuel Lafond[1], Lata Narayanan[2], Kangkang Wu[2]
[1]Department of Computer Science, Université de Montréal, Montréal, Canada
[2]Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada

Abstract

Alice wants to join a new social network, and influence its members to adopt a new product or idea. Each person v in the network has a certain threshold $t(v)$ for *activation*, i.e adoption of the product or idea. If v has at least $t(v)$ activated neighbors, then v will also become activated. If Alice wants to activate the entire social network, whom should she befriend? We study the problem of finding the minimum number of links that Alice should form to people in the network, in order to activate the entire social network. This *Minimum Links* Problem has applications in viral marketing and the study of epidemics. We show that the solution can be quite different from the related and widely studied Target Set Selection problem. We prove that the Minimum Links problem is NP-complete, in fact it is hard to approximate to within an $\varepsilon \ln n$ factor for some constant ε, even for graphs with degree 3 and with threshold at most 2. In contrast, we give linear time algorithms to solve the problem for trees, cycles, and cliques, and give precise bounds on the number of links needed.

[*]Research supported by NSERC, Canada

1 Introduction

The increasing popularity and proliferation of large online social networks, together with the availability of enormous amounts of data about customer bases, has contributed to the rise of *viral marketing* as an effective strategy in promoting new products or ideas. This strategy relies on the insight that once a certain fraction of a social network adopts a product, a larger cascade of further adoptions is predictable due to the *word-of-mouth* network effect [14, 22, 3]. Inspired by social networks and viral marketing, Domingos and Richardson [11, 27] were the first to raise the following important algorithmic problem in the context of social network analysis: If a company can turn a subset of customers in a given network into early adopters, and the goal is to trigger a large cascade of further adoptions, which set of customers should they target?

We use the well-known threshold model to study the influence diffusion process in social networks from an algorithmic perspective. The social network is modelled by a node-weighted graph $G = (V,E,t)$ with $V(G)$ representing individuals in the social network, $E(G)$ denoting the social connections, and t an integer-valued *threshold function*. Starting with a *target set*, that is, a subset $S \subseteq V$ of nodes in the graph, that are *activated* by some external incentive, influence propagates deterministically in discrete time steps, and *activates* nodes. For any unactivated node v, if the number of its activated neighbors at time step $t-1$ is at least $t(v)$, then node v will be activated in step t. A node once activated stays activated. It is easy to see that if S is non-empty, then the process terminates after at most $|V|-1$ steps. We call the set of nodes that are activated when the process terminates as the *activated set*. The problem proposed by Domingo and Richardson [11, 27] can now be formulated as follows: Given a social network $G = (V,E,t)$, and an integer k, find a subset $S \subseteq V$ of size k so that the resulting activated set is as large as possible. In the context of viral marketing, the parameter k corresponds to the budget, and S is a target set that maximizes the size of the activated set. One question of interest is to find the cheapest way to activate the *entire network*, when possible. The optimization problem that results has been called the *Target Set Selection Problem*, and has been widely studied (see for eg. [4, 1, 25]): the goal is to find a minimum-sized set $S \subseteq V$ that activates the entire network (if such a set exists). In a certain sense, the elements of this minimum target set S are the most influential people in the network; if they are activated, the entire network will eventually be activated.

There are, however, two hidden flaws in the formulation of the target set problem. First, the nodes in the target set are assumed to be activated immediately by external incentives, *regardless of their own thresholds of activation*. This is not a realistic assumption; in the context of viral marketing, it is possible, perhaps even likely, that highly influential nodes have high thresholds, and cannot be activated by external incentives alone. Secondly, there is no possibility of giving *partial* external incentives; indeed the target set is activated *only* by external incentives, and the remaining nodes *only* by the internal network effect.

In this paper, we address the flaws mentioned above. We study a related but different problem. Suppose Alice wants to join a new social network, whom should she befriend if her goal is to influence the entire social network? In other words, to whom should Alice create links, so that she can activate the entire network? If Alice creates a link to a node v, the threshold of v is only effectively reduced by one, and so v in turn is activated only if its threshold is one. We call our problem the *Minimum Links* problem (Min-Links).

The Min-Links problem provides a new way to model a viral marketing strategy, which addresses the flaws described in the target set problem formulation. Indeed, Alice can represent the external initiator of a viral marketing strategy. The links added from the external node correspond to the external incentive given to the endpoints of these links. The nodes that are the endpoints of these new links may not be immediately completely activated, but their thresholds are effectively reduced; this corresponds to their receiving partial incentives. One way of seeing this is that every individual to whom we link is given a $10 coupon; for some people this may be enough for them to buy the product, for others, it reduces their resistance to buying it. Individuals with high thresholds cannot be activated only by external incentives. The Min-Links problem also has important applications in epidemiology or the spread of epidemics: in the spread of a new disease, where an infected person arrives from *outside* a community, the Min-Links problem corresponds to identifying the smallest set of people such that if the infected external person has contact with this set, the entire community could potentially be infected.

Observe that the solution to the Min-Links problem can be quite different from the solution to the Target Set Selection problem for a given network. For example, consider a star network, where the leaves all have threshold 1, while the central node has degree $n-1$ and has threshold n. The optimal target set is the central node, while the only solution to the Min-Links problem is to create links to all nodes in the network. Thus, a solution to the Min-Links problem can be arbitrarily larger than one to the Target Set Selection problem for the same social network. However, any solution to the Min-Links problem is clearly also a feasible solution to the Target Set Selection problem.

1.1 Our Results

We prove that the Min-Links problem is NP-hard, and is in fact, hard to approximate to within an $\varepsilon \log n$ factor for some $\varepsilon < 1$. In light of the hardness results, we study the complexity of the problem for social networks that can be represented as trees, cycles, and cliques. In each case, we give a necessary and sufficient condition for the feasibility of the Min-Links problem, based on the structural properties and an observation of the threshold function. We then give $O(|V|)$ algorithms to solve the Min-Links problem for all the studied graph topologies. Finally, we give exact bounds on the number of links needed to activate the entire network for all the above specific topologies, as a function of the threshold values.

1.2 Related work

The problem of identifying the most influential nodes in a social network has received a tremendous amount of attention [15, 18, 23, 17, 5, 16, 2, 12]. The algorithmic question of choosing the target set of size k that activates the most number of nodes in the context of viral marketing was first posed by Domingos and Richardson [11]. Kempe et al [20] started the study of this problem as a discrete optimization problem, and studied it in both the probabilistic independent cascade model and the threshold model of the influence diffusion process. They showed the NP-hardness of the problem in both models, and showed that a natural greedy strategy has a $(1-1/e-\varepsilon)$-approximation guarantee in both models; these results were generalized to a more general cascade model in [21].

In the Target Set Selection problem, the size of the target set is not specified in advance, but the goal is to activate the entire network. Chen [4] showed that it is hard to approximate the optimal Target Set to within a polylogarithmic factor, even when all nodes have majority thresholds, or have constant degrees and thresholds two. A polynomial-time algorithm for trees was given in the same paper. Ben-Zwi et al. [1] generalized the result on trees to show that target set selection can be solved in $n^{O(w)}$ time where w is the treewidth of the input graph. The effect of parameters such as diameter, vertex cover number etc. of the input graph on the complexity of the problem are studied in [25]. The Minimum Target Set has also been studied from the point of view of the spread of disease or epidemics. For eg., in [19], the case when all nodes have a threshold k is studied; the authors showed that the problem is NP-complete for fixed $k \geq 3$.

Influence diffusion under time window constraints were studied in [13]. Maximizing the number of nodes activated within a specified number of rounds has also been studied [9, 24]. The problem of dynamos or dynamic monopolies in graphs (eg. [26]) is essentially the target set problem restricted to the case when every node's threshold is half its degree.

The paper closest to our work is [8], in which Demaine *et al* introduce a model to *partially incentivize* nodes to maximize the spread of influence. Our work differs from theirs in several ways. First, they study the maximization of influence given a fixed budget, while we study in a sense the budget (number of links) needed to activate the entire network. Second, they consider thresholds chosen uniformly at random, while we study arbitrary thresholds. Finally, they allow arbitrary fractional influence to be applied externally on any node, while in our model, every node that receives a link has its threshold reduced by the same amount.

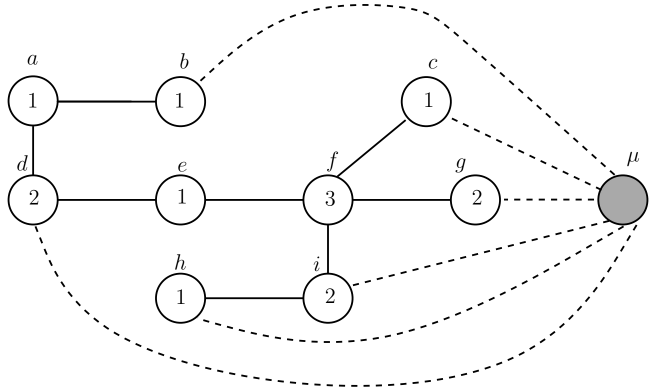

Figure 1: Node μ is the external influencer and is assumed to be activated. Links in the link set are shown with dashed edges. The given link set activates the entire network and is an optimal pervading link set.

2 Notation and preliminaries.

Given a social network represented by an undirected graph $G = (V, E, t)$, we introduce a set of external nodes U that are assumed to be already activated. We assume that all edges have unit weight; this is generally called the *uniform weight assumption*, and has previously been considered in many papers [4, 13, 6, 7]. A *link set* for (G, U) is a set S of links between nodes in U and nodes in V, i.e $S \subseteq \{(u, v) \mid u \in U; v \in V\}$. For a link set S, we define $E(S) = \{v \in V \mid \exists (u, v) \in S\}$, that is, $E(S)$ is the set of V-endpoints of links in S. For a node v, define $r(v)$ to be the number of links in S for which v is an endpoint. Since the set of external nodes U is already activated, observe that adding the link set S to G is equivalent to reducing the threshold of the node v by $r(v)$. In the viral marketing scenario, the link set S represents giving v a partial incentive of $r(v)$.

Given a link set S for a graph G, we define $I(G, S)$ to be the set of nodes in G that are eventually activated as a result of adding the link set S, that is, by reducing the threshold of each node $v \in E(S)$ by $min\{r(v), t(v)\}$, and then running the influence diffusion process. See Figure 1 for an illustration. Observe that in the target set formulation, this is the same as the set of nodes activated by using U as the target set in the graph G', the graph obtained from G by adding the set U to the vertex set and the set S to the set of edges.

A link set S such that $I(G, S) = V$, that is, S activates the entire network, is called a *pervading link set*. A pervading link set of minimum size is called an *optimal pervading link set.*

Definition 1 Minimum Links (Min-Links) problem: *Given a social network $G = (V, E, t)$, where t is the threshold function on V, and a set of external nodes U, find an optimal pervading link set for (G, U).*

In this paper, we consider the case of a *single influencer*, that is, $U = \{\mu\}$. In this case, a link given to a vertex v reduces its threshold by 1. Since μ must be an endpoint of each edge in the link set S, each such edge can be uniquely specified by a vertex in V. We therefore generally omit mention of μ in the rest of the paper. For each node $v \in E(S)$, we say we *give* v a link, or that v *receives* a link. If activating $X \subseteq V$ activates, directly or indirectly, the set of vertices Y, we write $X \sim Y$ (note that there may be vertices outside Y that X activates). We write $x \sim Y$ instead of $\{x\} \sim Y$. The minimum cardinality of a link set for a Min-Links instance G is denoted $ML(G)$.

Observe that for some graphs, a pervading link set may not exist; for example, consider a singleton node of threshold greater than 1. The existence of a feasible solution can be verified in $O(E)$ time by giving a link to every node in V, and simulating the influence diffusion process. The following simple observation stating two conditions under which no pervading link set exists, is used throughout the paper:

Observation 1 *A graph G does not have a pervading link set if it has a node v such that $t(v) > degree(v) + 1$, or if there is no node with threshold 1.*

3 NP-hardness

In this section, we prove that the Min-Links problem is NP-hard; in fact, it is almost as hard as Set-Cover to approximate, even if G has degree bounded by 3 and thresholds bounded by 2. Given a collection of n sets $\mathcal{S} = \{S_1, \ldots, S_n\}$ whose union is the universe $\mathcal{U}$ of cardinality m, with $n \leq m^k$ for some constant k, the Set-Cover problem is to find a minimum set cover, that is, a sub-collection of minimum cardinality $\mathcal{S}' \subseteq \mathcal{S}$ such that $\bigcup_{S \in \mathcal{S}'} S = \mathcal{U}$. The cardinality of $\mathcal{S}'$ is denoted $MSC(\mathcal{S})$. We shall make use of rooted binary trees. For such a tree T, denote the root by $r(T)$, and the set of leaves by $\mathcal{L}(T)$.

Constructing G from $\mathcal{S}$: Given a Set-Cover instance $\mathcal{S}$, we describe the construction of a corresponding Min-Links instance $G = (V, E, t)$ in polynomial time, which is used for our reduction. Figure 2 illustrates our construction. For each set in $\mathcal{S}$ and each element in $\mathcal{U}$, we introduce two binary trees in G, and then describe how to connect these trees. For each $S \in \mathcal{S}$, add to G a binary tree B_S with $|S|$ leaves $\mathcal{L}(B_S) = \{b_{S,u_1}, \ldots, b_{S,u_{|S|}}\}$, one for each element $u_i \in S$. Add another binary tree B'_S with $|S|$ leaves $\mathcal{L}(B'_S) = \{b'_{S,u_1}, \ldots, b'_{S,u_{|S|}}\}$, again one for each element $u_i \in S$. Then, add an edge between $r(B_S)$ and $r(B'_S)$.

The thresholds are $t(b) = 1$ for every $b \in V(B_S) \cup \mathcal{L}(B'_S)$, and $t(b') = 2$ for every internal node b' of $V(B'_S)$, that is for every $b' \in V(B'_S) \setminus \mathcal{L}(B'_S)$. Note that $\mathcal{L}(B'_S) \sim V(B'_S) \sim V(B_S)$.

Then for each element $u \in \mathcal{U}$, add a binary tree C_u with $|\mathcal{S}(u)|$ leaves, where $\mathcal{S}(u) = \{S \in \mathcal{S} : u \in S\}$ consists of the sets containing u. Denote $\mathcal{L}(C_u) = \{c_{u,S_1}, \ldots, c_{u,S_{|\mathcal{S}(u)|}}\}$, each leaf corresponding to a set S_i of $\mathcal{S}(u)$. Next, add yet another binary tree C'_u with $|\mathcal{S}(u)|$ leaves $\{c'_{u,S_1}, \ldots, c'_{u,S_{|\mathcal{S}(u)|}}\}$, again one for each $S_i \in \mathcal{S}(u)$. Add an edge between $r(C_u)$ and $r(C'_u)$. Every node $c \in V(C_u) \cup V(C'_u)$ has $t(c) = 1$.

We now define a gadget called a *heavy link*. Let x, y be two non-adjacent nodes with $t(x) = t(y) = 1$. Adding an $x - y$ *heavy link* consists of adding two nodes z_1, z_2 that are neighbors of x, then adding another node z_3 that is a neighbor of z_1, z_2 and y. We set the thresholds $t(z_1) = t(z_2) = 1$ and $t(z_3) = 2$. Note that the heavy link makes $x \sim y$ but not necessarily $y \sim x$ (thus adding an $x - y$ heavy link is different from adding a $y - x$ heavy link). Also notice that this operation increases the degree of x by 2 and of y by 1, and that z_1, z_2 and z_3 have degree bounded by 3.

To finish the construction, for every set $S \in \mathcal{S}$ and each element $u \in S$, add a $b_{S,u} - c_{u,S}$ heavy link, and a $c'_{u,S} - b'_{S,u}$ heavy link. Denote by H_S the set of nodes added to G by incorporating the heavy links to the B_S leaves, and by H'_u the set of heavy link nodes added to the C'_u leaves. It is not hard to see that G can be constructed in polynomial time. Note that for each $S \in \mathcal{S}$, the nodes of B_S are equivalent, in the sense that if one is activated, then they all get activated. The same holds for the nodes of C_u and C'_u, for every $u \in \mathcal{U}$. We will use their roots as *representatives*, meaning that we will implicitly use the fact that $r(B_S) \sim V(B_S)$ and $r(C_u) \sim V(C_u)$.

The proof of the following lemma is straightforward and is in the appendix:

Lemma 1 *Let $\mathcal{S}$ be an instance of* Set-Cover *over universe $\mathcal{U}$, with $|\mathcal{S}| = n$ and $|\mathcal{U}| = m$, and let $G = (V, E, t)$ be the* Min-Links *instance constructed as above. Then all of the following conditions are met:*

1. *$|V| \leq m^c$ for some constant c;*
2. *each node of G has at most 3 neighbors;*
3. *$t(v) \leq 2$ for every node v of G.*

We now show that both $\mathcal{S}$ and its corresponding instance G have the same optimality value.

Lemma 2 *$MSC(\mathcal{S}) = ML(G)$.*

Proof. First observe that for a given set $S \in \mathcal{S}$,

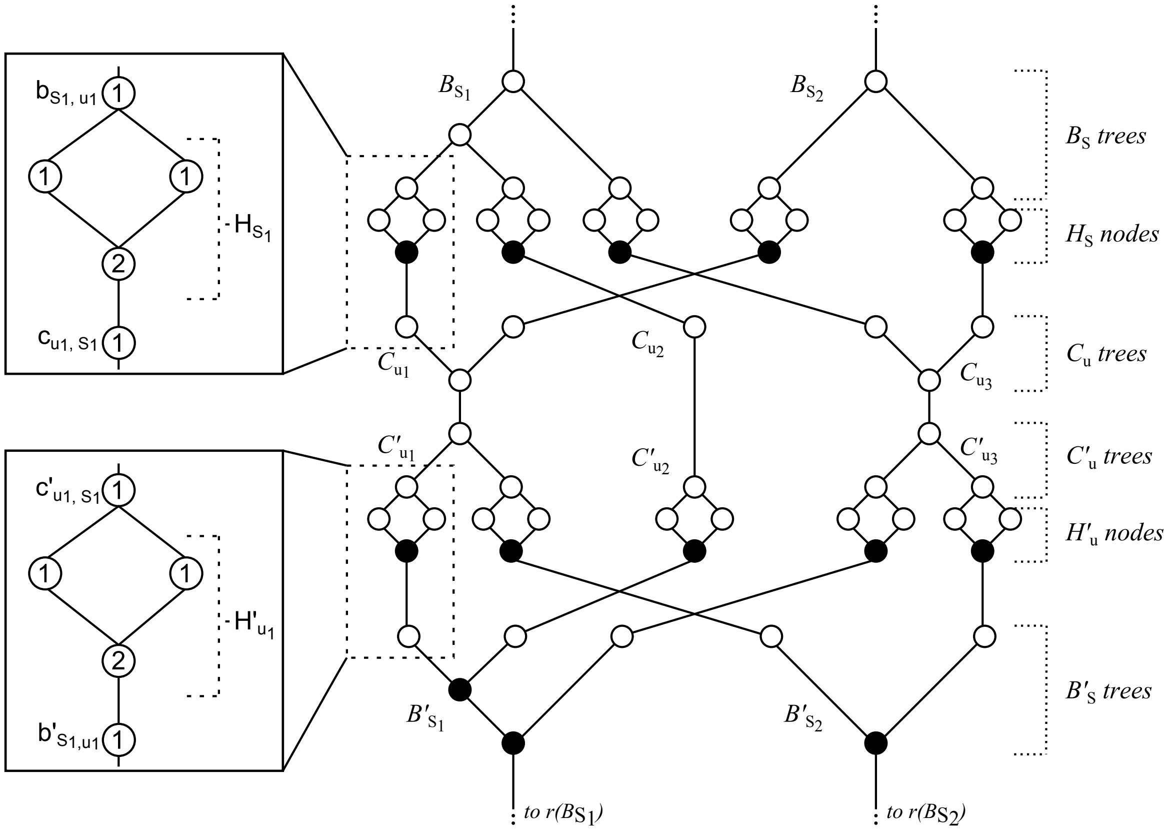

Figure 2: The construction of G from $\mathcal{S}$ consisting of $S_1 = \{u_1, u_2, u_3\}$ and $S_2 = \{u_1, u_3\}$. White nodes have threshold 1, whereas black nodes have threshold 2.

$$\bigcup_{u\in S} r(C_u) \sim \mathcal{L}(B'_S) \sim V(B'_S) \sim V(B_S)$$

which implies that

$$\bigcup_{u\in\mathcal{U}} r(C_u) \sim \bigcup_{S\in\mathcal{S}} V(B'_S) \sim \bigcup_{S\in\mathcal{S}} V(B_S)$$

and it follows that $\bigcup_{u\in\mathcal{U}} r(C_u) \sim V$.

To see that $MSC(\mathcal{S}) \geq ML(G)$, if $\mathcal{S}' \subseteq \mathcal{S}$ is a minimum set cover, then giving links to $V' = \bigcup_{S\in\mathcal{S}'} r(B_S)$ suffices to activate G since $V' \sim \bigcup_{u\in\mathcal{U}} r(C_u) \sim V$. Thus $MSC(\mathcal{S}) \geq ML(G)$.

It remains to show that $MSC(\mathcal{S}) \leq ML(G)$. Let $B = \{r(B_S) : S \in \mathcal{S}\}$. Let $V' \subseteq V$ be the set of endpoints of $E(\hat{S})$ for an optimal pervading link set $\hat{S}$ such that $|V' \cap B|$ is maximized among all possible choices. We divide this section of the proof into two claims.

Claim 1 *$V' \subseteq B$.*

Proof. First observe that we may assume that if $x \in V' \setminus B$, then there is no set S such that $r(B_S) \sim x$ (for otherwise, we can replace x by $r(B_S)$ in V', contradicting our choice of V'). But no such x can exist. If $x \in V(C_u)$ for some u, then $r(B_S) \sim x$ for any set S containing u. If x belongs to a $b_{S,u} - c_{u,S}$ heavy link, then $r(B_S) \sim x$. If x belongs to a $c'_{u,S} - b'_{S,u}$ heavy link, then again $r(B_S) \sim r(C_u) \sim x$. Finally if $x \in V(B'_S)$, then $r(B_S) \sim \bigcup_{u\in S} r(C_u) \sim \mathcal{L}(B'_S) \sim x$. We conclude that V' has only nodes from B. ■

Claim 2 *$\mathcal{S}' = \{S \in \mathcal{S} : r(B_S) \in V'\}$ is a set cover.*

Proof. Suppose the claim is false, and let $w \in \mathcal{U}$ be an element not covered by $\mathcal{S}'$. Recall that $\mathcal{S}(w) = \{S_1, \dots, S_{|\mathcal{S}(w)|}\}$ is the collection of sets containing w. Let $S_i \in \mathcal{S}(w)$. Then in B'_{S_i}, there is a leaf $b'_{S_i,w}$. Let P_{S_i} be the set of nodes lying on the unique $b'_{S_i,w} - r(B'_{S_i})$ shortest path in B'_{S_i} (inclusively). Define W as the node set that contains the C_w and C'_w nodes along with the heavy link nodes appended to $\mathcal{L}(C'_w)$, plus for each $S_i \in \mathcal{S}(w)$, the P_{S_i} nodes and the B_{S_i} nodes with the heavy link nodes appended to $\mathcal{L}(B_{S_i})$. Formally,

$$W = V(C_w) \cup V(C'_w) \cup H'_w \cup \left(\bigcup_{S_i \in \mathcal{S}(w)} (V(B_{S_i}) \cup H_{S_i} \cup P_{S_i}) \right)$$

We show that no node of W gets activated by V', contradicting the assertion that $\hat{S}$ is a pervading link set. Suppose instead that some W nodes do get activated. Let z be the first node of W activated by the propagation process (or if multiple nodes of W get simultneously activated first, pick z arbitrarily among them). Then, since $V' \cap W = \emptyset$, z must have $t(z)$ neighbors outside of W that were activated and influenced it. Observe that the only nodes of W that have neighbors outside of W belong to either H_{S_i} or P_{S_i} for some $S_i \in \mathcal{S}(w)$. If $z \in H_{S_i}$, then the only heavy link node with neighbors outside of W is the threshold 2 node. But then, z has only one neighbor outside W (namely a c_{u,S_i} node for some u), which is not enough to activate z. Thus $z \notin H_{S_i}$. If $z \in P_{S_i}$, then $z \neq b'_{S_i,w}$ since $b'_{S_i,w}$ receives no influence from outside of W: it has two neighbors, one is in P_{S_i} and the other is in the $b'_{S_i,w} - c'_{w,S_i}$ heavy link, both of which are in W. If instead z is an interior node of the P_{S_i} path, then z has two neighbors in W (by the definition of a path). But $t(z) = 2$ and z has only three neighbors, i.e. only one outside of W, and so z cannot be activated only by influence from outside W. The last possible case is $z = r(B'_{S_i})$. But again, z has two neighbors in W: one is in P_{S_i} and the other is $r(B_{S_i})$, and the same argument applies. We conclude that z, and hence w, cannot exist, and that S' is a set cover. ■

Since V' yields a set cover $\mathcal{S}$ of size $ML(G)$, we deduce that $MSC(\mathcal{S}) \leq ML(G)$. ■

We can now state the main result of this section.

Theorem 1 *The decision version of* Min-Links *is NP-complete, even when restricted to instances with maximum degree* 3 *and maximum threshold* 2*. Moreover, there exists a constant* $\varepsilon > 0$ *such that the optimization version of* Min-Links*, under the same restrictions, is NP-hard to approximate within a* $\varepsilon \ln n$ *factor, where* n *is the number of nodes of the given graph.*

Proof. *NP*-completeness follows directly from Lemma 2, and observing that Min-Links is in *NP*, as it is easy to check that a given set V' is a pervading link set (because propagation must finish in a polynomial number of steps). As for the inapproximability result, let $\mathcal{S}$ be an instance of set cover over universe $\mathcal{U}$, $|\mathcal{S}| = n$ and $|\mathcal{U}| = m$, and let n' be the number of nodes of G constructed from $\mathcal{S}$ as described above, with $n' \leq m^c$ (c is the constant from Lemma 1). Dinur and Steurer showed that it is NP-hard to approximate set cover within a $d \ln m$ factor for any $0 < d < 1$ [10]. For our purposes, fix $0 < d < 1$, and suppose that some approximation algorithm $\mathcal{A}$ always finds a pervading link set of size at most $APP \leq \frac{d}{c} \ln(n') \cdot ML(G)$. Because $ML(G) = MSC(\mathcal{S})$, we have $APP \geq MSC(\mathcal{S})$, and in the other direction,

$$APP \leq \frac{d}{c} \ln(n') \cdot ML(G) \leq \frac{d}{c} \ln(m^c) \cdot ML(G) = d \ln(m) \cdot MSC(\mathcal{S})$$

and hence $\mathcal{A}$ can approximate Set-Cover to within a factor $d \ln(m)$ using the aforementioned reduction. Therefore, for $\varepsilon = \frac{d}{c}$, it is hard to approximate the Min-Links problem within a $\varepsilon \ln(n')$ factor. ■

4 Trees

In contrast to the NP-completeness of the Min-Links problem shown in the previous section, we now show that there is a linear time algorithm to solve the problem in trees. We start with a necessary and sufficient condition for a tree T to have a valid pervading link set.

Proposition 1 *Let T be a tree and let v be a leaf in T. Let $T' = T - \{v\}$ and T'' be the same as T' except that the threshold of w, the neighbor of v in T, is reduced by 1. Then T has a pervading link set if and only if (a) either $t(v) = 1$ and T'' has a pervading link set or (b) $t(v) = 2$ and T' has a pervading link set.*

We now prove a critical lemma that shows that for any node in the tree, there is an optimal solution that gives a link to that node.

Lemma 3 *Let T be a tree with n nodes that has a pervading link set, and let v be a node in T. Then there exists an optimal solution for* Min-Links(T) *in which v gets a link.*

Proof. We prove the lemma by induction on the number of nodes n in the tree. Clearly it is true if $n = 1$. Suppose $n > 1$, and let S be an optimal pervading link set for T. If v gets a link, we are done. If not, v must have a neighbor w that is activated before v, and that contributes to the activation of v. Let T_1 and T_2 be the two trees created by removing the edge between v and w, with T_1 containing w, and let S_1 (respectively S_2) be the links of S with an endpoint in T_1 (respectively T_2). Since T is a tree, and v is activated after w by S, none of the links in S_2 can contribute to the activation of nodes in T_1. It follows that S_1 is a pervading link set for T_1, and in fact is optimal, as a smaller solution for T_1 could be combined with S_2 to yield a better solution for T, contradicting the optimality of S. By the inductive hypothesis, there is an optimal solution S' for T_1 that gives a link to w. Note that $|S'| = |S_1|$, and $S' \cup S_2$ must also be an optimal solution for T. But clearly $S'' = S' \cup S_2 \cup \{(\mu, v)\} - \{(\mu, w)\}$ also activates the entire tree T , and since $|S''| = |S|$, we conclude that S'' is an optimal solution for T, that gives a link to v, as needed to complete the proof by induction. ■

The above lemma suggests a simple way to break up the Min-Links problem for a tree into subproblems that can be solved independently, which yields a linear-time greedy algorithm.

Theorem 2 *The* Min-Links *problem can be solved for trees in linear time.*

Proof. Given a tree T, let v be an arbitrary leaf in the tree. By Lemma 3, there is an optimal solution, say S, to the Min-Links problem for T that gives v a link. Suppose $t(v) = 2$, then the link to v is not enough to activate v, and therefore v's neighbor w must activate v. Also, v's activation cannot help in activating any other nodes in T. Thus $S - \{(\mu, v)\}$ must be an optimal solution to $T' = T - \{v\}$. Suppose instead that $t(v) = 1$. Then the link given to v activates it immediately. Consider the induced subgraph of T containing only nodes of threshold 1, and let C be the connected component (subtree) containing v in this subgraph. Then clearly $v \sim C$. Since S is optimal, S cannot contain any node in C except for v. Construct T' by removing C from T, and subtracting 1 from the threshold of any node x who is a neighbor of a node in C. Observe that any such node x can be a neighbor of exactly one node in C, since T is a tree. Then $S - \{(\mu, v)\}$ must be an optimal solution to T'; if instead there is a smaller-sized solution to T', we can add (μ, v) to that solution to obtain a smaller solution for T than S, contradicting the optimality of S.

The above argument justifies the correctness of the following simple greedy algorithm. Initialize $S = \emptyset$. Take a leaf v in the tree. If $t(v) > 2$ then there is no solution by Observation 1. If $t(v) = 2$, then put the link (μ, v) in S, remove v from the tree, and recursively solve the remaining tree. If $t(v) = 1$, then give a link to v, remove the subtree of T that is connected to v consisting only of nodes of degree 1, reduce the thresholds of all neighbors of the nodes in this subtree by 1, and recursively solve the resulting trees. It is easy to see that the algorithm can be implemented in linear time. ■

For the network in Figure 1, assuming that leaves in the tree are always processed in alphabetical order, the greedy algorithm given in Theorem 2 first picks node b and adds a link to it. We then remove nodes b and a, and reduce the threshold of d by 1. Next we pick c, give it a link, remove it from the tree, and decrement $t(f)$ to 2. The next leaf that is picked and given a link is d; since d's threshold now is 1, we remove d and e from the tree, and reduce f's threshold to 1. Proceeding in this way, we arrive at the link set shown.

We now give an exact bound on $ML(T)$, the number of links required to activate the entire tree T, the proof of which is in the appendix.

Theorem 3 *Let T be a tree that has a pervading link set. Then $ML(T) = 1 + \sum_{v \in T}(t(v) - 1)$*

We remark that in contrast to the intuition for the optimal target set problem, where we would choose nodes of high degree or threshold to be in the target set, in the Min-Links problem, our algorithm gives links to leaves initially, though eventually nodes that were internal nodes in the tree may also receive links. That is, the best nodes to befriend might be the nodes with a single connection to other nodes in the tree!

5 Cycles

In this section, we give a solution for the Min-Links problem on cycles. Let $C_n = (V, E, t)$ be a cycle with n nodes, $V = \{0, 1, ..., n-1\}$, $E = \{((i, i+1) \text{ } mod \text{ } n) \mid 1 \leq i \leq n\}$, and $t : t(v) \to \mathcal{Z}^+$. We define $P_{i,j}$ $(i \neq j)$ to be the sub-path of C_n consisting of all nodes in $\{i, \ldots, j\}$ in the clockwise direction. We may use the $[i, j]$ notation to denote the vertices of $P_{i,j}$. By *consecutive vertices of threshold 3*, we mean two vertices i, j such that the only two vertices in $P_{i,j}$ with threshold 3 are i and j.

Proposition 2 *A cycle has a pervading link set if and only there is at least one node of threshold 1, every node is of threshold at most 3, and between any two consecutive nodes of threshold 3, there is at least one node of threshold 1.*

We note that a similar condition can be stated for paths, with the additional restriction that there must be a node of threshold 1 before (after) the first (last resp.) node of threshold 3.

We give a linear time algorithm for finding a minimum-sized link set for problem Min-Links(C_n). Essentially we reduce the problem to finding an optimal solution for an appropriate path.

Theorem 4 *The* Min-Links *problem for a cycle C_n can be solved in time $\Theta(n)$.*

Proof. By Observation 1, there is no solution if there is a node with threshold 4 or more. If there exists a node i such that $t(i) = 3$, then clearly i must get a link, and both of its neighbors must be activated before it. That is, i can play no role in activating any node in $P_{i+1,i-1}$. Therefore, $S = \{(\mu, i)\} \cup S'$ is an optimal solution to C_n where S' is an optimal solution to $P_{i+1,i-1}$. In this case, S' can be found in linear time using the tree algorithm of Theorem 2. If there is no node with threshold 3, a single node with threshold 2, and the remaining nodes all have threshold 1, then by giving a link to *any* of the nodes with threshold 1, we can activate the entire cycle.

It remains only to consider the case when there are no nodes of threshold 3, at least two nodes of threshold 2, and at least one node of threshold 1. Fix an arbitrary node i of threshold 1 in C_n. We define $c(i)$ and $cc(i)$ to be the first node with threshold 2 in i's clockwise direction and counter clockwise direction respectively, $c(i) \neq cc(i)$ (see Figure 3). We also define $P_{c(i),cc(i)}$ be the path from $c(i)$ to $cc(i)$ where $t(c(i)) = t(cc(i)) = 2$; and $P'_{c(i),cc(i)}$ to be the same path as $P_{c(i),cc(i)}$ except that we set $t(c(i)) = t(cc(i)) = 1$.

We first claim that there exists an optimal solution that gives a link to i. To see this, let S be an optimal solution that does not give a link to node i. Since all nodes in C_n are activated by S, there must exist some node $j \in [cc(i), c(i)]$ that gets a link. If $t(j) = 1$, we can take the link given to j and give it instead to node i. Otherwise there exists $j \in \{c(i), cc(i)\}$ such that it gets a link and is activated before i, and eventually activates i. Again we can move the link from node j to node i, which clearly has the same effect of giving a link to node j. Therefore, we have a new solution of the same size as S that gives a link to node i.

Consider therefore an optimal solution S that gives a link to the node i. It is not hard to see that that $S - \{(\mu, i)\}$ must be an optimal solution to Min-Links$(P'_{c(i),cc(i)})$, since activating i activates $[cc(i)+1, c(i)-1]$ and lowers the threshold of $cc(i)$ and $c(i)$. Again, since the Min-Links problem for a path can be solved in $\Theta(n)$ according to Theorem 2, we can construct an optimal solution for a cycle in $\Theta(n)$ time as well. ■

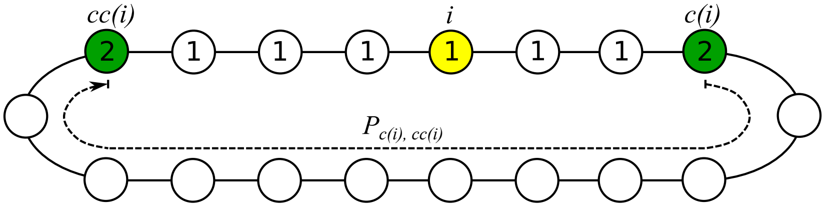

Figure 3: A cycle with no threshold 3 vertices, illustrating the main components of the proof.

We give an exact bound on the number of links required to fully activate a cycle.

Theorem 5 *Given a cycle $C_n = (V, E, t)$ which has a pervading link set, $ML(C_n) = \sum_{i=1}^{n}(t(i) - 1)$*

Proof. If there is a node i of threshold 3, then $ML(C_n) = 1 + ML(P_{i+1,i-1})$. Since by Theorem 3, $ML(P_{i+1,i-1}) = 1 + \sum_{j \neq i}(t(j) - 1)$, we have $ML(C_n) = 1 + (1 + \sum_{j \neq i}(t(j) - 1)) = (t(i) - 1) + \sum_{j \neq i}(t(j) - 1) = \sum_{j=1}^{n}(t(j) - 1)$ as needed. If there is no node of threshold 3 and a single node of threshold 2, then $ML(C_n) = 1 = \sum_{j=1}^{n}(t(j) - 1)$. Finally, if there is no node of threshold 3, and at least two nodes of threshold 2, and at least one of threshold 1, then $ML(C_n) = 1 + ML(P'_{cc(i),c(i)})$ where i is a node of threshold 1. Since the thresholds of $c(i)$ and $cc(i)$ have been reduced by 1 each in $P'_{cc(i),c(i)}$, by Theorem 3, we have $ML(P'_{cc(i),c(i)}) = -1 + \sum_{j \in [cc(i),c(i)]}(t(j) - 1)$. Therefore $ML(C_n) = 1 - 1 + \sum_{j \in [cc(i),c(i)]}(t(j) - 1) = \sum_{i=1}^{n}(t(i) - 1)$. ■

6 Cliques

In this section, we give an algorithm to solve the Min-Links problem on cliques. Let $K_n = (V, E, t)$ be a clique with n nodes, $V = \{1, 2, ..., n\}$ and $E = \{(i, j) : 1 \leq i < j \leq n\}$ and $t : t(v) \to \mathcal{Z}^+$. We first show a necessary and sufficient condition for the Min-Links problem to have a feasible solution:

Proposition 3 *Let K_n be a clique with $t(i) \leq t(i+1)$, for all $1 \leq i < n$. Then K_n has a pervading link set if and only if $t(i) \leq i$ for all $1 \leq i \leq n$.*

Proof. If $t(i) \leq i$ for all $1 \leq i \leq n$, it is easy to see that there exists a solution S by giving a link to every node i; we claim that node i is activated in or before round i. Since $t(1) \leq 1$, node 1 is activated in round 1. Inductively, node 1 to $i - 1$ are already activated in round $i - 1$, the effective threshold of node i has been reduced to ≤ 1. Node i receives a link, therefore, node i must be activated in the i^{th} round, if it is not already activated. Conversely, suppose there exist nodes j such that $t(j) > j$ and there exists a solution S to the Min-Links problem; let p be the smallest such node with $t(p) > p$. In order to activate any node $q \geq p$, at least p nodes have to be activated before q, since $t(q) \geq t(p) > p$. However, there are only $p - 1$ nodes that can be activated before any such node $q \geq p$. Thus no node q with $q \geq p$ can be activated, a contradiction. ■

We now give a greedy algorithm to solve the Min-Links problem on a clique.

Theorem 6 *The* Min-Links *problem for a clique K_n can be solved in time $\Theta(n)$.*

Proof. First sort the nodes in order of threshold. By Observation 1, there is no solution if any node has a threshold $> n$, therefore, we can use counting sort and complete the sorting in $\Theta(n)$ time. Clearly, the condition given in Proposition 3 can easily be checked in linear time. We now give the following greedy linear time algorithm for a clique which has a feasible solution: give a link to node 1, and let j be the maximum value such that $t(i) < i$ whenever $2 \leq i < j$. Remove all nodes in $\{1,\ldots,j-1\}$, decrement by $j-1$ the thresholds of all nodes $\geq j$, and solve the resulting graph recursively. It is easy to see that this algorithm can be implemented in linear time, in an iterative fashion as follows: we examine the nodes in order. When we process node i, if $t(i) < i$, we simply increment i and continue; if $t(i) = i$, we give a link to node i. We now show that the link set produced by this greedy algorithm is optimal.

First we show that there must be an optimal solution that contains the node 1. Consider an optimal solution S and let i be the smallest index of a node that receives a link in S. If $i = 1$, then we are done. If not, since there must always be a node with threshold 1 that receives a link, it must be that $t(i) = 1$. But then we can move the link from i to 1, to create a new solution S' which will activate node i in the next step. Since $|S'| = |S|$ and $I(K_n,S) = I(K_n,S')$, S' is an optimal solution to the Min-Links problem that contains the node 1. Thus, we can assume that the optimal solution S contains the node 1.

Next we claim that $S - \{1\}$ is an optimal solution to the clique C' which is the induced sub-graph on the nodes $\{j, j+1,\ldots,n\}$ where $j > 1$ is the smallest index with $t(j) = j$, and with thresholds of all nodes reduced by $j-1$. Suppose there is a smaller solution S' to C'. We claim that $S' \cup \{1\}$ activates all nodes in the clique K_n. Since for any node $1 < k < j$, we have $t(k) < k$, it can be seen inductively that the link given to node 1 suffices to activate node k. Thus, all nodes in $\{1,2,\ldots j-1\}$ are activated. Furthermore, the thresholds of all nodes in $\{j, j+1,\ldots,n\}$ are effectively reduced by $j-1$. Thus using the links in S' suffices to activate them. Finally, since $|S'| < |S|-1$, $S' \cup \{1\}$ is a smaller solution than S to the clique K_n, contradicting the optimality of S for K_n. We conclude that the greedy algorithm described above produces a minimum sized solution to the Min-Links problem. ■

The following tight bound on the minimum number of links to activate an entire clique is immediate:

Theorem 7 *Given a clique K_n which has a feasible solution, $ML(K_n) = |\{j \mid t(j) = j\}|$*

The greedy algorithm from Theorem 6 can be extended to complete multi-partite graphs:

Theorem 8 *The* Min-Links *problem for a complete multi-partite graph G can be solved in time $O(|E(G)|)$.*

7 Discussion

In this paper, we introduced and studied the Min-Links problem: given a social network G where every node v has a threshold $t(v)$ to be activated, which minimum-sized set of nodes should an already acivatated external influencer μ befriend, so as to influence the entire network? We showed that the problem is NP-complete, in fact it is hard to approximate to within an $\varepsilon \ln n$ factor (for some constant $0 < \varepsilon < 1$) even for graphs with maximum degree 3, and with maximum threshold 2. In contrast, we show linear time algorithms for the problem for trees, cycles, cliques, and complete k-partite graphs, and give an exact bound (as a function of the thresholds) on the number of links needed for such graphs.This leaves open the question of a polynomial time algorithm for graphs of bounded treewidth, as well as the best possible approximation algorithm for general graphs. It would be interesting to generalize these algorithms to find the minimum number of links required to influence a specified fraction of the nodes. Other directions include studying the *multiple influencer case*, and the case with non-uniform weights on the edges. Clearly, the problem remains NP-complete in general, but the complexity for special classes of graphs remains open. Another interesting question is that of maximizing the number of activated nodes, given a fixed budget of k links.

References

[1] Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. Treewidth governs the complexity of target set selection. *Discrete Optimization*, 8:702–715, 2011.

[2] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 946–957, 2014.

[3] Jacqueline Johnson Brown and Peter H Reingen. Social ties and word-of-mouth referral behavior. *Journal of Consumer research*, pages 350–362, 1987.

[4] Ning Chen. On the approximability of influence in social networks. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 1029–1037, 2008.

[5] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 199–208, 2009.

[6] Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, M. Milanic, Joseph Peters, and Ugo Vaccaro. How to go viral: Cheaply and quickly. In *Proceedings of the Seventh International Conference on Fun with Algorithms*, volume 8496 of *Lecture Notes in Computer Science*, pages 100–112. 2014.

[7] Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, Martin Milanic, and Ugo Vaccaro. Latency-bounded target set selection in social networks. In *The Nature of Computation. Logic, Algorithms, Applications*, volume 7921 of *Lecture Notes in Computer Science*, pages 65–77. 2013.

[8] Erik D. Demaine, Mohammad Taghi Hajiaghayi, Hamid Mahini, David L. Malec, S. Raghavan, Anshul Sawant, and Morteza Zadimoghadam. How to influence people with partial incentives. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 937–948, 2014.

[9] Thang Dinh, Huiyuan Zhang, Dzung Nguyen, and My Thai. Cost-effective viral marketing for time-critical campaigns in large-scale social networks. *IEEE/ACM Transactions on Networking*, PP(99):1–1, 2014.

[10] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 624–633. ACM, 2014.

[11] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 57–66, 2001.

[12] MohammadAmin Fazli, Mohammad Ghodsi, Jafar Habibi, Pooya Jalaly Khalilabadi, Vahab Mirrokni, and SinaSadeghian Sadeghabad. On the non-progressive spread of influence through social networks. In *Latin American Theoretical Informatics Symposium, LATIN 12*, volume 7256 of *Lecture Notes in Computer Science*, pages 315–326. 2012.

[13] Luisa Gargano, Pavol Hell, Joseph Peters, and Ugo Vaccaro. Influence diffusion in social networks under time window constraints. In *Proceedings of the 20th International Colloquium on Structural Information and Communication Complexity, SIROCCO 13*, volume 8179 of *Lecture Notes in Computer Science*, pages 141–152. 2013.

[14] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.

[15] Amit Goyal, Francesco Bonchi, Laks Lakshmanan, and Suresh Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, 3:179–192, 2013.

[16] Amit Goyal, Francesco Bonchi, and Laks V.S. Lakshmanan. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5:73–84, 2011.

[17] Amit Goyal, Wei Lu, and Laks V.S. Lakshmanan. Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 47–48, 2011.

[18] Jing He, Shouling Ji, Raheem Beyah, and Zhipeng Cai. Minimum-sized influential node set selection for social networks under the independent cascade model. In *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '14, pages 93–102, 2014.

[19] Paul A. Dreyer Jr. and Fred S. Roberts. Irreversible -threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Applied Mathematics*, 157(7):1615 – 1627, 2009.

[20] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, 2003.

[21] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32rd International Conference on Automata, Languages and Programming*, ICALP'05, pages 1127–1138, 2005.

[22] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 228–237. ACM, 2006.

[23] Wei Lu, Francesco Bonchi, Amit Goyal, and Laks V.S. Lakshmanan. The bang for the buck: Fair competitive viral marketing from the host perspective. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 928–936, 2013.

[24] Shunming Lv and Li Pan. Influence maximization in independent cascade model with limited propagation distance. In *Web Technologies and Applications*, volume 8710 of *Lecture Notes in Computer Science*, pages 23–34. 2014.

[25] Andre Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller. On tractable cases of target set selection. *Social Network Analysis and Mining*, 3(2):233–256, 2012.

[26] David Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. *Theoretical Computer Science*, 282:231–257, 2002.

[27] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 61–70, 2002.

Appendix

Lemma 1 *Let $\mathcal{S}$ be an instance of* Set-Cover *over universe $\mathcal{U}$, with $|\mathcal{S}| = n$ and $|\mathcal{U}| = m$, and let $G = (V, E, t)$ be the* Min-Links *instance constructed as above. Then all of the following conditions are met:*

1. *1 $|V| \leq m^c$ for some constant c;*

2. *2 each node of G has at most 3 neighbors;*

3. *3 $t(v) \leq 2$ for every node v of G.*

Proof. For 1, there are $2n+2m$ binary trees in G, which together contain at most $\ell = 2n \cdot m + 2m \cdot n = 4nm$ leaves. Thus the binary trees contain less than 2ℓ nodes in total. The heavy links account for at most 3ℓ nodes in total, and so $|V| \leq 5\ell \leq 20nm \leq m^c$ for some c (because $n \leq m^k$). To see that 2 holds, i.e. that the maximum degree is 3, observe that G consists of binary trees to which we add at most neighbor per root ($r(B_S)$ with $r(B'_S)$, and $r(C_u)$ with $r(C'_u)$), plus at most two neighbors per leaf (the heavy links). In the case that a node is both a root and a leaf (e.g. B_{S_i} is a single node because S_i has only one element), three neighbors are added to it, but it has zero neighbors initially. As for 3, it is easy to see that $t(v) \leq 2$ for every node $v \in V$ created. ■

Theorem 3 *Let T be a tree that has a feasible solution. Then $ML(T) = 1 + \sum_{v \in T}(t(v) - 1)$*

Proof. We give a proof by induction on the number of nodes n in the tree. Clearly if the tree consists of a single node x, there is a solution if and only if $t(x) = 1$, and the number of links needed is 1 which is equal to $1 + \sum_{v \in V}(t(v) - 1)$ as needed. Now consider a tree T with $n > 1$ nodes and let x be a leaf in the tree. Then by Lemma 3, there is an optimal solution S in which x gets a link. By Observation 1, there is a solution only if $t(x) = 1$ or $t(x) = 2$. Let $T' = T - \{x\}$ (all nodes keep the same thresholds as in T) and let T'' be the tree derived from T by removing x and reducing the threshold of w, the neighbor of x in T by 1.

First we consider the case when $t(x) = 2$. Then giving x a link is not sufficient to activate it. By the usual cut-and-paste argument, $S - \{(\mu, x)\}$ must be an optimal solution for tree T'.

$$\begin{aligned} ML(T) &= 1 + ML(T') \\ &= t(x) - 1 + (1 + \sum_{v \in T'}(t(v) - 1)) \text{ by the inductive hypothesis} \\ &= 1 + \sum_{v \in T}(t(v) - 1) \end{aligned}$$

Next we consider the case when $t(x) = 1$, and $t(w) > 1$. Then x is immediately activated by the link it receives in S, and the link given to x effectively reduces the threshold of w. Therefore, $S - \{(\mu, x)\}$ must be an optimal solution for the tree T'' in which the threshold of w is $t(w) - 1$. It follows that

$$\begin{aligned} ML(T) &= 1 + ML(T'') \\ &= 1 + (1 + \sum_{v \in T''}(t(v) - 1)) \text{ by the inductive hypothesis} \\ &= 2 + (t(w) - 2) + \sum_{v \in T'' - \{w\}}(t(v) - 1) \\ &= 1 + \sum_{v \in T}(t(v) - 1) \end{aligned}$$

Finally suppose $t(x) = t(w) = 1$. Then it is impossible that S contains w, as this would contradict the optimality of S. Therefore, we can move the link from node v to node w, to get a new optimal pervading link set S' for T. Furthermore, S' must also be an optimal pervading link set for T'. It follows that

$$\begin{aligned} ML(T) &= ML(T') \\ &= t(x) - 1 + (1 + \sum_{v \in T'} (t(v) - 1)) \text{ by the inductive hypothesis} \\ &= 1 + \sum_{v \in T} (t(v) - 1) \end{aligned}$$

■

Proposition 2 *A cycle has a pervading link set if and only if all the conditions below are met:*

1. *there is at least one node of threshold 1*
2. *every node is of threshold at most 3*
3. *between any two consecutive nodes of threshold 3, there is at least one node of threshold 1.*

Proof. The necessity of the first two conditions follows from Observation 1. Suppose there are two consecutive nodes i and j with $j > 1$, and with threshold 3, such that all nodes between them have threshold 2. Then both nodes i and j needs both their neighbors to be activated before them, but meanwhile, since there is no node of threshold 1, no node in the the sub-path $P_{i+1,j-1}$ can be activated. Therefore none of the nodes in the sub-path $P_{i,j}$ can be activated. Conversely, if all three conditions listed in the statement are met, it is easy to see that by giving a link to every node in the cycle, all the nodes in the cycle can be activated. ■

Theorem 8 *The* Min-Links *problem for a complete multi-partite G can be solved in time $O(|E(G)|)$.*

Proof. Let $V_1, \dots, V_k$ be the vertex sets of G. Consider the following algorithm: give a link to any vertex of threshold 1, activate it and propagate its influence, remove activated vertices, and repeat until no vertex of threshold 1 remains. In order to propagate the influence of an activated vertex, each of its neighbors may need to be visited, hence the $O(|E|)$ complexity. If the resulting graph is non-empty, then G clearly cannot have a feasible solution. If it is empty, we claim that the vertices activated during this process form a minimum link set.

We prove our claim by induction on n. If G has 1 vertex, this is trivial. Assume $n > 1$, and let S be a minimum link set for G. Let v be the first vertex picked by the algorithm. We show that some optimal solution contains the link (μ, v). Assume that $(\mu, v) \notin S$. Let w be a vertex of threshold 1 in S. If $v \in V_i$ and $w \in V_j$, $i \neq j$, then $v \sim w$ and $w \sim v$, implying that $(S \setminus \{(\mu, w\}) \cup \{(\mu, v)\}$ is also optimal. Otherwise, if $v, w \in V_i$ for some i, then v and w have the same set of neighbors. Then w can be replaced by v in S, since both vertices have the same influence on G after activation. Therefore, we may now assume that $(\mu, v) \in S$. Let G' be the graph obtained after activating v, propagating its influence, and removing the activated vertices. The property of being complete multi-partite is preserved by vertex deletion, and so G' is still complete multi-partite. By induction, the algorithm finds a minimum link set S' on G'. It is not hard to see that $S' \cup \{(\mu, v)\}$ is an optimal link set. ■