On the Approximability of Influence in Social Networks

Ning Chen*

Abstract
In this paper, we study the spread of influence through a social network, in a model initially studied by Kempe, Kleinberg and Tardos [14, 15]: We are given a graph modeling a social network, where each node v has a (fixed) threshold t_v, such that the node will adopt a new product if t_v of its neighbors adopt it. Our goal is to find a small set S of nodes such that targeting the product to S would lead to adoption of the product by a large number of nodes in the graph.

We show strong inapproximability results for several variants of this problem. Our main result says that the problem of minimizing the size of S, while ensuring that targeting S would influence the whole network into adopting the product, is hard to approximate within a polylogarithmic factor. This implies similar results if only a fixed fraction of the network is ensured to adopt the product. Further, the hardness of approximation result continues to hold when all nodes have majority thresholds, or have constant degree and threshold two. The latter answers a complexity question proposed in [10, 29]. We also give some positive results for more restricted cases, such as when the underlying graph is a tree.

1 Introduction

It is well-documented that information spreads via social networks. The dynamic processes governing the diffusion of information and "word-of-mouth" effects have been studied in many fields, including Epidemiology [7, 23, 28], Sociology [24, 25, 31, 21, 5], Economics, and Computer Science [9, 27, 14, 13, 15, 11, 4, 6, 22, 12]. For example, a recently studied problem in the area of viral marketing is the following: Suppose we would like to market a new product and hope it will be adopted by a large fraction of individuals in the network. Which set of individuals should we "target" (for instance, one form of "targeting" involves offering free samples of the product)? The answer to the question depends crucially on the network structure and the extent to which "word-of-mouth" effects will take hold.

One simple way to model diffusion with discrete dynamics is to assume that each individual in the network has a "threshold": The individual becomes *influenced* (i.e. adopts the new product) if a certain pre-specified number of its neighbors have adopted the product. A natural algorithmic problem arises: Given knowledge of these thresholds, which individuals should be targeted so as to create a large wave of adoptions? Domingos and Richardson [9, 27] studied this problem in a probabilistic setting and heuristic solutions were given. Kempe, Kleinberg and Tardos [14, 15] modeled the question as an optimization problem and showed that it is NP-hard to compute the optimal subset to target, and developed approximation algorithms in a submodular framework. Other related literature about diffusion with thresholds includes, e.g. [20, 24, 25, 10, 5, 22].

In many studies, e.g. [20, 24, 10, 7, 28], researchers are interested in the long term effects of diffusion and whether some consensus can be reached. For example, in a virus propagation network, what nodes should we immunize so that the whole network is protected? Related work can be found in, e.g. [7, 23, 11]. In applications like this, a key requirement is that all (or a large fraction of) individuals in the network are influenced. In the current paper, we focus on this problem in the threshold model. In particular, we address the question as an optimization problem: Find a small set S of individuals such that targeting S would lead to influence to a large fraction of individuals in the network.

1.1 The Model We now define the problem formally. Given a connected undirected graph $G = (V, E)$, let $d(v)$ be the degree of $v \in V$. For each $v \in V$, there is a *threshold* value $t(v) \in \mathbb{N}$, where $1 \leq t(v) \leq d(v)$. Initially, the states of all vertices are *inactive*. We pick a subset of vertices, the *target set*, and set their state to be *active*. After that, in each discrete time step, the states of vertices are updated according to following rule: An inactive vertex v *becomes* active if at least $t(v)$ of its neighbors are active. The process runs until either all vertices are active or no additional vertices can update states from inactive to active (it is easy to verify the process runs at most $n-1$ rounds, where $n = |V|$ is

*Department of Computer Science & Engineering, University of Washington, Seattle, WA. Email: ning@cs.washington.edu.
Research supported in part by NSF grant CCF-0635147.

the number of vertices in the graph). The process we consider is *progressive*, i.e. a vertex can only become active from inactive but not vice versa.

We are interested in the following optimization problem, which we call TARGET SET SELECTION: Which subset of vertices should we target at the beginning such that all (or a fixed fraction of) vertices in the graph are active at the end? Observe that a trivial solution is to target all vertices in the graph. The goal we consider in this paper is to minimize the size of the target set.

Our model is different from Kempe et al. [14, 15] in the following two respects: First, Kempe et al. [14, 15] focused on the maximization problem– for any given k, find a target set of size k to maximize the (expected) number of active vertices at the end of process. Here, however, we ask for a target set of minimum size that guarantees that all (or a fixed fraction of) vertices are eventually active. Secondly, here we consider deterministic, explicitly given thresholds. Kempe et al. [14] showed strong hardness of approximation results for the maximization problem when the thresholds are deterministic. Thus, the main focus of their papers was on probabilistic thresholds where all thresholds are drawn randomly from a given range.

1.2 Our Results

For the general setting of the TARGET SET SELECTION problem, we show a polylogarithmic lower bound on the approximation ratio. Specifically, the TARGET SET SELECTION problem can not be approximated within a ratio of $O(2^{\log^{1-\epsilon} n})$, for any fixed constant $\epsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$. Our proof is based on a reduction from the Minimum Representative problem [18, 19].

Our result gives further evidence that without additional assumptions such as the probabilistic thresholds in [14, 15], the problem is completely intractable (even in constant degree graphs with thresholds at most two). Indeed, in the maximization problem studied in [14, 15], for deterministic thresholds, the problem is NP-hard to approximate within a ratio of $n^{1-\epsilon}$ [14]. In related work, Aazami et al. [1] studied a non-threshold propagation process called power dominating set and showed a similar hardness of approximation result.

Our result implies the same hardness of approximation ratio if, instead of ensuring all vertices in the network are active, we only need to activate a fixed fraction of vertices. Our hardness result gives a negative answer to the problem proposed by Roberts [28]– what vertices need to be "vaccinated" to make sure a virus does not spread to a fixed fraction of the whole network.

By considering different types of thresholds and network structures, we show the following additional results.

Majority Thresholds. One important and well studied threshold is majority, where a vertex becomes active if at least half of its neighbors are active. It has many applications in distributed computing, voting systems, etc. [24]. For example, Peleg [25] proposed the use of a majority update rule for maintaining data consistency in a distributed system. Peleg [24] proved that it is NP-hard to compute the optimal target set for majority thresholds. For different variants of majorities and progressive processes, different lower bounds on the size of the target set were obtained [20, 25, 5]. For further information about majority thresholds, see [24].

For the majority thresholds setting, we show that the problem shares the same hardness of approximation ratio as the general setting. In particular, this implies that the majority thresholds setting does not admit any approximation algorithm of ratio better than $O(2^{\log^{1-\epsilon} n})$, for any fixed constant $\epsilon > 0$. To the best of our knowledge, this is the first inapproximability result for majority thresholds.

Small Thresholds. Another interesting special case is when all thresholds are small, say constant [29]. Dreyer [10] showed that if the threshold of any vertex is k for any $k \geq 3$, the TARGET SET SELECTION problem is NP-hard. However, it leaves as an open problem [29] the case of $k = 2$. Note that the problem can be solved trivially for the case of $k = 1$: Target an arbitrary vertex in each connected component.

In this paper, we solve the problem by proving it is NP-hard as well when $k = 2$. Indeed, we show a much stronger and surprising result: Approximating the TARGET SET SELECTION problem in the threshold 2 setting is as hard as approximating the problem in the general setting, even for constant degree graphs. Our result implies that to study upper or lower bounds on the approximation ratio of the TARGET SET SELECTION problem, it suffices to consider the threshold 2 setting.

Our proof is based on our hardness result for majority thresholds and the simulation of monotone boolean circuits. Specifically, observe that the state of each vertex can be viewed as a boolean variable and written as a majority boolean function of the states of its neighbors. By the results built on sorting network [17], e.g. the seminal work by Ajtai, Komlós and Szemerédi [2], a majority boolean function can be simulated by a polynomial size monotone circuit. Thus, the influence propagation in a social network can be viewed as running a polynomial size monotone circuit on each vertex locally. Given this idea, we construct gadgets composed of vertices with thresholds at most 2 to simulate each AND and OR gate in the circuit.

Unanimous Thresholds. The most influence-resistant setting is the unanimous threshold setting, i.e. the threshold of each node is equal to its degree. For example, in an ideal virus-resistant network, a vertex is infected only if all of its neighbors are infected. Understanding this particular case can help us to construct robust virus-resistant network structures. We show that the problem with unanimous thresholds is equivalent to vertex cover, which implies it admits a 2-approximation algorithm.

Tree Structure. One simple, but important, class of social network structures is trees. For example in query incentive networks [16, 3], the graph is modeled by a tree structure. When the underlying social network is a tree, Dreyer [10] gave a polynomial time algorithm to compute the optimal target set if all thresholds are same. We generalize the result to arbitrary thresholds, using dynamic programming.

2 A Polylogarithmic Hardness Result

In general, the TARGET SET SELECTION problem is hard to approximate within a near polynomial ratio. Formally,

THEOREM 2.1. *The* TARGET SET SELECTION *problem can not be approximated within the ratio of* $O(2^{\log^{1-\epsilon} n})$*, for any fixed constant* $\epsilon > 0$*, unless* $NP \subseteq DTIME(n^{polylog(n)})$.

We will prove the theorem by a reduction from the Minimum Representative (MINREP) problem [18, 19]. We begin by describing the MINREP problem and then show the reduction.

2.1 The MINREP Problem

Given a bipartite graph $G = (A, B; E)$, where A and B are disjoint sets of vertices, there are explicit partitions of A and B into equal-sized subsets. That is, $A = \bigcup_{i=1}^{\alpha} A_i$ and $B = \bigcup_{j=1}^{\beta} B_j$, where all sets A_i have the same size $|A|/\alpha$ and all sets B_j have the same size $|B|/\beta$. The partition of G induces a super-graph H as follows: There are $\alpha + \beta$ super-vertices, corresponding to each A_i and B_j respectively, and there is a super-edge between A_i and B_j if there exist some $a \in A_i$ and $b \in B_j$ that are adjacent in G. Figure 1 gives an example of the MINREP problem, where each set A_i has 3 vertices and B_j has 4 vertices.

We say a pair (a, b) *covers* a super-edge (A_i, B_j) if $a \in A_i$ and $b \in B_j$ are adjacent in G. For example, in Figure 1, (u, v) covers super-edge (A_1, B_j). We say $S \subseteq A_i \cup B_j$ *covers* a super-edge (A_i, B_j) if there exist $a, b \in S$ such that (a, b) covers (A_i, B_j).

The goal of the MINREP problem is to select the minimum number of representatives from each set A_i and B_j such that all super-edges are covered. That is, we wish to find subsets $A' \subseteq A$ and $B' \subseteq B$ with the minimum total size $|A'|+|B'|$ such that, for every super-edge (A_i, B_j), there exist representatives $a \in A' \cap A_i$ and $b \in B' \cap B_j$ that are adjacent in G.

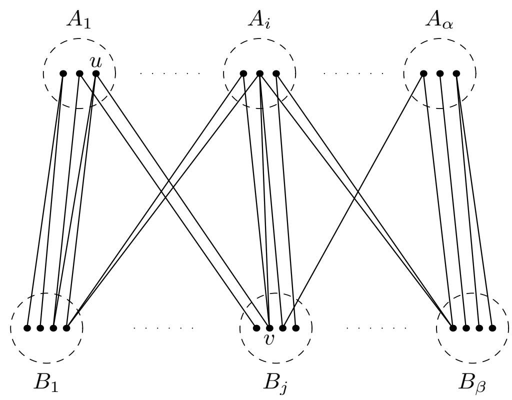

Figure 1: An instance of the MINREP problem.

The MINREP problem is closely related to the label cover problem that models two-prover one-round proof systems, and the following result follows directly from the parallel repetition theorem [26].

THEOREM 2.2. *For any fixed* $\epsilon > 0$*, the* MINREP *problem can not be approximated within the ratio of* $O(2^{\log^{1-\epsilon} n})$*, unless* $NP \subseteq DTIME(n^{polylog(n)})$.

2.2 Proof of Theorem 2.1

For any given MINREP instance $G = (A, B; E)$, let M be the number of super-edges and N be the total input size. In the reduction, we will use a number of the following basic gadgets Γ_ℓ, where $t(v_i) = 1$, for $i = 1, \ldots, \ell$.

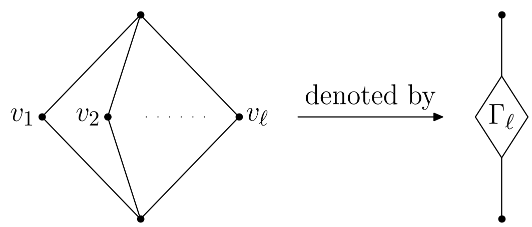

Figure 2: The basic gadget Γ_ℓ.

We next describe the construction of graph G' for the TARGET SET SELECTION problem. Basically, G' consists of four different groups of vertices V_1, V_2, V_3, V_4, where the vertices between two groups are connected by the basic gadgets described above. Formally,

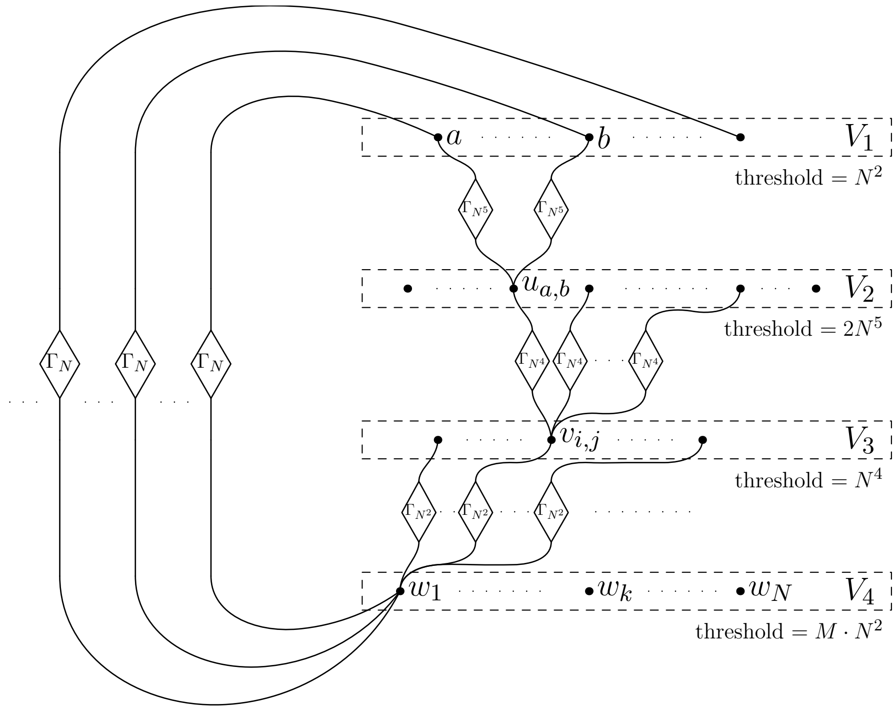

Figure 3: The structure of graph G'.

- $V_1 = \{a \mid a \in A\} \cup \{b \mid b \in B\}$ and each vertex has threshold N^2.
- $V_2 = \{u_{a,b} \mid (a,b) \in E\}$ and each vertex has threshold $2N^5$. Vertex $u_{a,b} \in V_2$ is connected to each of $a, b \in V_1$ by a basic gadget Γ_{N^5}.
- $V_3 = \{v_{i,j} \mid A_i, B_j$ is connected by a super-edge$\}$ and each vertex has threshold N^4. Vertex $u_{a,b} \in V_2$ is connected to $v_{i,j} \in V_3$ by a basic gadget Γ_{N^4} if $a \in A_i$ and $b \in B_j$.
- $V_4 = \{w_1, \ldots, w_N\}$ and each vertex has threshold $M \cdot N^2$. Each vertex $v_{i,j} \in V_3$ is connected to each $w_k \in V_4$ by a basic gadget Γ_{N^2}, and each vertex $a, b \in V_1$ is connected to each $w_k \in V_4$ by a basic gadget Γ_N.

Figure 3 displays the structure of the construction.

We claim that the size of the optimal MinRep solution of G is within a factor of two of the size of the optimal Target Set Selection solution of G'. Thus, any approximation algorithm for Target Set Selection essentially gives the same approximation ratio (up to at most a constant factor) for MinRep.

Assume $A' \subseteq A$ and $B' \subseteq B$ is an optimal MinRep solution of G. We claim that $A' \cup B' \subseteq V_1$ is a Target Set Selection solution of G'. Since $A' \cup B'$ is a MinRep solution, for any super-edge (A_i, B_j), there exist $a \in A' \cap A_i$ and $b \in B' \cap B_j$ such that $(a,b) \in E$. Thus, vertex $u_{a,b} \in V_2$ can be active, which implies that $v_{i,j} \in V_3$ can be active as well. This is true for all super-edges, and thus all vertices in V_3 are active, which implies that all vertices in V_4 are active. Therefore, all vertices in V_1 can be active, which induces all vertices in G' to be active at the end.

On the other hand, let S be an optimal Target Set Selection solution of G'. First of all, it is safe to assume that no middle vertices $v_1, \ldots, v_\ell$ from any basic gadget Γ_ℓ are in S. Secondly, we can assume without loss of generality that no vertices in V_3 are in S. This is because if a vertex $v_{i,j} \in S \cap V_3$, then we can remove $v_{i,j}$ from S and include $u_{a,b} \in V_2$ to S, where $a \in A_i$ and $b \in B_j$, which gives a solution of the same size. Finally, if a vertex $u_{a,b} \in S \cap V_2$, we can remove $u_{a,b}$ from S and include $a, b \in V_1$ to S. By doing this, the size of S is increased by at most a factor of two. Now S only contains vertices from V_1 and V_4, i.e. $S \subseteq V_1 \cup V_4$. According to our construction, those vertices in $S \cap V_4$ can not affect any other vertices until all vertices in V_4 are active. Therefore, the only direction for influence to flow in G' is through the channel $V_1 \to V_2 \to V_3 \to V_4$. However, to activate any vertex $w \in V_4 \setminus S$, all vertices in V_3 have to be activated. This implies that $S \cap V_1$ is a MinRep solution of G.

By Theorem 2.2, we have the same hardness of approximation result for the TARGET SET SELECTION problem, which completes the proof of Theorem 2.1.

2.3 Extensions We observe that Theorem 2.1 continues to hold for a few extensions:

- The optimal solution influences each vertex in a constant number of rounds. This follows directly from the above construction, i.e. Figure 3.

- Instead of ensuring all vertices in the network are active, only a fixed fraction of vertices is needed to be activated. This can be done by the following simple reduction: For the given graph $G = (V, E)$, let $n = |V|$. We construct a new graph G' as follows: Replace each edge in E by a basic gadget Γ_n and define the new threshold of each $v \in V$ to be $t'(v) = n \cdot t(v)$. It is easy to see that G and G' have the same optimal solution. On graph G', by adding many dummy vertices (with thresholds being equal to their degrees) and connecting to all original vertices in V, it can be seen that to activate a fixed fraction of vertices, all vertices in V have to be activated.

3 Majority Thresholds

In this section, we consider *majority thresholds*, i.e. a vertex becomes active if at least half of its neighbors are active. Formally, for each $v \in V$, $t(v) = \left\lceil \frac{d(v)}{2} \right\rceil$.[1]

THEOREM 3.1. *Assume the* TARGET SET SELECTION *problem with arbitrary thresholds can not be approximated within the ratio of $f(n)$, for some polynomial time computable function $f(n)$. Then the problem with majority thresholds can not be approximated within the ratio of $O(f(n))$.*

Proof. For any graph $G = (V, E)$ with arbitrary thresholds, we will construct another graph G' with majority thresholds such that the size of the optimal TARGET SET SELECTION solution of G' and G differs by at most 1. The basic idea is, for each $v \in V$ with $t(v) \neq \left\lceil \frac{d(v)}{2} \right\rceil$, to add some dummy vertices incident to v (and change the threshold of v, if necessary) such that the threshold of v in the new setting is majority.

To be specific, for any $v \in V$ with threshold $t(v)$ and degree $d(v)$, there are the following two cases:

[1] In our discussions, we assume that ties are broken in favor of "prefer-to-change". All our results continue to hold for other ties-breaking rules.

`Case 1.` $t(v) > \left\lceil \frac{d(v)}{2} \right\rceil$. For this case, we add $2t(v) - d(v)$ isolated dummy vertices incident to v and with threshold 1 each.

`Case 2.` $t(v) < \left\lceil \frac{d(v)}{2} \right\rceil$. For this case, we add $d(v) - 2t(v)$ isolated dummy vertices incident to v and with threshold 1 each. Furthermore, let the new threshold of v be $d(v) - t(v)$.

In addition, we add a "super" vertex u and connect u to all dummy vertices added in the above `Case 2`. Let the threshold of u be its majority. Denote the resulting graph by G'. Note that the thresholds of all vertices in G' are majority.

We claim that the size difference between the optimal TARGET SET SELECTION solution of G' and G is at most 1. For any given solution S of G, it can be seen that $S \cup \{u\}$ is a TARGET SET SELECTION solution of G'. On the other hand, consider any optimal solution S' of G'. Assume without loss of generality that no dummy vertices added in `Case 1` and `Case 2` are in S'. If $u \in S'$, then $S' \setminus \{u\}$ is a solution of G. Otherwise, S' itself is a solution of G.

Therefore, the size of the optimal solution of G' and G differs by at most 1. Thus, essentially they share the same hardness of approximation ratio. ■

Given the hardness result of Theorem 2.1 and the above result, the following conclusion follows immediately.

COROLLARY 3.1. *The* TARGET SET SELECTION *problem with majority thresholds can not be approximated within the ratio of $O(2^{\log^{1-\epsilon} n})$, for any fixed constant $\epsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$.*

4 Small Thresholds

When the thresholds are small, say all equal to one, i.e. $t(v) = 1$ for any $v \in V$, the problem can be solved trivially: For each connected component of the graph, we target a vertex arbitrarily. Surprisingly, the problem becomes hard to approximate within a ratio of $O(2^{\log^{1-\epsilon} n})$ when $t(v) = 2$ (or $t(v) \leq 2$), even for constant degree graphs. In particular, we will show the following result.

THEOREM 4.1. *Assume the* TARGET SET SELECTION *problem with arbitrary thresholds can not be approximated within the ratio of $f(n)$, for some polynomial time computable function $f(n)$. Then the problem can not be approximated within the ratio of $O(f(n))$ when all thresholds are at most 2.*

We have the following corollary, which answers a complexity question proposed in [10, 29].

Corollary 4.1. *Given any graph where $t(v) = 2$ (or $t(v) \leq 2$) for any vertex v, the* Target Set Selection *problem can not be approximated within the ratio of $O(2^{\log^{1-\epsilon} n})$, for any fixed constant $\epsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$.*

Proof. The case where $t(v) \leq 2$ follows directed from Theorem 2.1 and Theorem 4.1. It remains to consider the case where $t(v) = 2$ for any vertex v.

We will prove the $t(v) = 2$ case by a reduction from the $t(v) \leq 2$ case. Given a graph $G = (V, E)$ where $t(v) \leq 2$ for any $v \in V$, we add a "super" vertex u and connect u to each $v \in V$ with $t(v) = 1$. Let the resulting graph be G' and all thresholds in G' be 2. We claim that the size difference between the optimal solution of G' and G is at most 1. Then the claim follows from the hardness of the $t(v) \leq 2$ case.

For any Target Set Selection solution S of G, it is easy to see $S \cup \{u\}$ is a solution of G'. On the other hand, assume S' is an optimal solution of G'. If $u \in S'$, then $S' \setminus \{u\}$ is a solution of G. If $u \notin S'$, then S' itself is a solution of G. This completes the proof. ■

Next we will prove Theorem 4.1. Our reduction is built on (1) the hardness result of majority thresholds given by Theorem 3.1, (2) the boolean circuits of computing majority functions and (3) the gadgets of simulating majority boolean circuit. We begin by describing how to do the simulation and then show the reduction.

4.1 Simulating Majority Boolean Circuit A boolean function $f : \{0,1\}^n \to \{0,1\}$ is called a *majority* function if

$$f(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } x_1 + \cdots + x_n \geq \lceil \frac{n}{2} \rceil \\ 0 & \text{otherwise} \end{cases}$$

We will use the following result by Ajtai, Komlós and Szemerédi [2].

Theorem 4.2. ([2]) *There exist polynomial size monotone circuits to compute majority boolean functions, where monotone means only AND and OR gates are in the circuit.*

The basic idea is to construct small gadgets composed of vertices of thresholds at most 2 to simulate AND and OR gates in a circuit. For a circuit that computes a majority function $f(x_1, \ldots, x_n)$, let us denote the gates in the circuit by u_i. Denote the final output gate by u_0 and input gates by $u_1, \ldots, u_n$ (corresponding to $x_1, \ldots, x_n$, respectively). Thus each u_i, $i > n$, is the output of an AND or an OR gate with other u_j's as inputs. The graph we construct has a vertex w_i with threshold 2 for each u_i and a gadget for each AND and OR gate in the circuit. We consider AND and OR gates respectively as follows.

For any AND gate, we construct the following gadget, where the value on each vertex is its threshold.

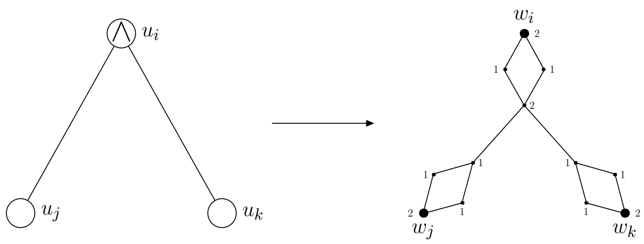

Figure 4: Gadget for AND gate.

It can be seen that for the "bottom-to-top" channel (i.e. $w_j, w_k \to w_i$), w_i is active (corresponding to the output u_i being 1) only if both w_j and w_k are active (corresponding to the inputs u_j and u_k being 1). In addition, if only one of w_j and w_k is active (say w_j), the center vertex of threshold 2 ensures that neither w_i nor w_k can get active due to influence from w_j. On the other hand, considering the channel from "top-to-bottom", once w_i is active, both w_j and w_k become active as well.

For any OR gate with output u_i, we construct the following gadget, where once again the value on each vertex is its threshold. Recall that w_0 is the vertex corresponding to the final output u_0 of the circuit.

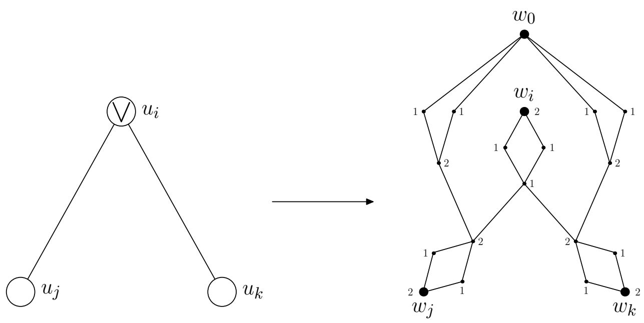

Figure 5: Gadget for OR gate.

As in AND case, for the "bottom-to-top" channel (i.e. $w_j, w_k \to w_i$), w_i is active (corresponding to the output u_i being 1) if at least one of w_j and w_k is active (corresponding to at least one of the inputs u_j and u_k being 1). In addition, if only one of w_j and w_k is active (say w_j), even though w_i can be activated, neither w_0 nor w_k can get active due to influence from w_i, w_j. On

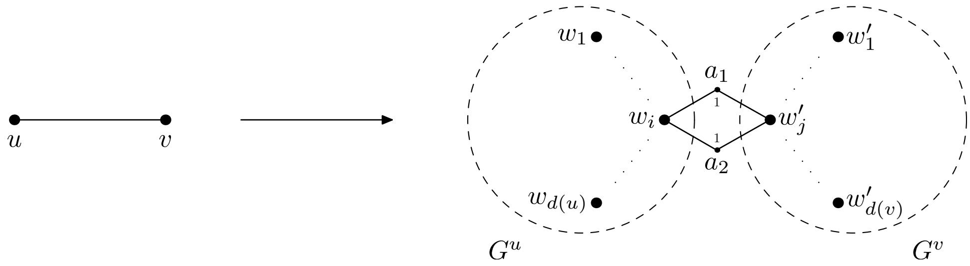

Figure 6: Gadget for edge (u, v).

the other hand, for the channel from "top-to-bottom", when w_i is active, w_j and w_k can be active once w_0 is active as well.

Denote the resulting graph by G_n. From the argument above, we know G_n has the following properties:

- If w_0 is active, then all vertices in G_n can become active. This implies that if there is a vertex targeted in G_n, we can assume without loss of generality that the vertex is w_0. We call w_0 the *output vertex* of G_n and denote it by $r(G_n)$.

- If at least half of vertices in $\{w_1, \ldots, w_n\}$ are active, then w_0 can be active. This holds because the circuit correctly computes the majority function and our simulation of each gate. In the following discussions, we denote $w_1, \ldots, w_n$ by the *input vertices* of G_n.

- If a vertex w_i is inactive, then all its neighbors are still inactive. This is important in that the propagation in G_n can only be through the channel of "top-to-bottom" or "bottom-to-top". In particular, this implies that if less than half of the input vertices are active, then the remaining inactive input vertices can not be activated due to influence from G_n.

4.2 Proof of Theorem 4.1 We are now ready to prove Theorem 4.1. Given a graph $G = (V, E)$, where each $v \in V$ has majority threshold, we will construct a graph $G' = (V', E')$, where $t(v) \leq 2$ for any $v \in V'$, such that the size of the optimal Target Set Selection solution of G is equal to that of G'. The claim then follows from Theorem 3.1.

For each $v \in V$, let $d(v)$ be the degree of v in G. We use a copy G^v of graph $G_{d(v)}$ to replace v and all its incident edges, where $G_{d(v)}$ is the graph constructed above to simulate majority function $f(\cdot)$ with $d(v)$ input variables. Each input vertex in G^v corresponds to an edge incident to v in E. For any edge $(u, v) \in E$, let w_i and w'_j be the two input vertices in G^u and G^v corresponding to (u, v), respectively. We connect w_i and w'_j by a basic gadget Γ_2 (i.e. as Figure 6 shows, we add two vertices a_1 and a_2 with threshold 1 each and connect $(a_1, w_i), (a_1, w'_j), (a_2, w_i), (a_2, w'_j)$). Denote the resulting graph by G'.

For any Target Set Selection solution S of G, let $S' = \{r(G^v) \mid v \in S\}$, i.e. S' contains the output vertex of each G^v for $v \in S$. For any $v \in S$, we consider how its neighbor u could be influenced by v. In graph G, we know u can be influenced from v directly by one unit. In graph G', according to the properties of G^v established in the last subsection, we know all vertices in G^v are active. Thus, as u and v are connected by an edge, one of the input vertices of G^u becomes active. Since the threshold of u in G is majority, u becomes active when at least half of its neighbors are active, which is equivalent to at least half of the input vertices of G^u being active (and thus, all vertices in G^u are active). Hence, the influence propagation in G' follows exactly the same pattern as that in G, and hence S' is a Target Set Selection solution of G'.

On the other hand, let S' be an optimal Target Set Selection solution of G'. According to the properties of simulation graph discussed above, we can assume without loss of generality that only output vertices are in S'. Define $S = \{v \in V \mid r(G^v) \in S'\}$. By a similar argument as above, it follows that S is a Target Set Selection solution of G.

Therefore, the size of the optimal Target Set Selection solution of G is equal to that of G', which completes the proof of Theorem 4.1.

4.3 Constant Degree Graphs In this subsection, we will show that Theorem 4.1, as well as Corollary 4.1, continues to hold for constant degree graphs.

By the construction of AKS sorting network [2] and the reduction of proving Theorem 4.1, the only vertex that has non-constant degree in each G^v gadget is its output vertex $r(G^v)$. This is because for each OR gadget, we add four edges incident to $r(G^v)$, as Figure 5 shows (i.e. w_0). To fix this, we make a few "identical" copies of $r(G^v)$ such that each copy has constant degree. More precisely, we replace $r(G^v)$ and all its incident edges with the gadget showed in the following figure:

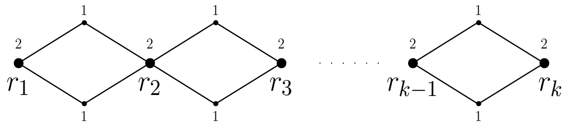

Figure 7: Gadget for the output vertex.

where each r_i, $i = 1, \ldots, k$, is an "identical" copy of $r(G^v)$ and $k-1$ is the number of OR gates in G^v. In particular, r_1 corresponds to the original $r(G^v)$ and each r_i, $i = 2, \ldots, k$, corresponds to an OR gate and is used to add the four edges as Figure 5 shows. If one of r_i's becomes active, all others are active as well and thus the resulting constant degree graph is equivalent to the original graph.

For Corollary 4.1, note that in the proof of Corollary 4.1, we add a "super" vertex u and connect u to all vertices with threshold one. In general, the degree of u can be arbitrary. Instead of adding one such "super" vertex, we add the following gadget:

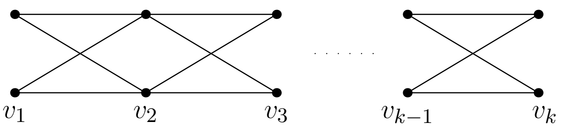

Figure 8: Gadget for the "super" vertex.

where each v_i connects to a vertex in the original graph with threshold one (assume there are k such vertices totally). The thresholds all vertices in the resulting graph are 2. It can be seen that the size of the optimal solution of the original and resulting graph differs by at most 2. Hence, Corollary 4.1 still holds for constant degree graphs.

5 Unanimous Thresholds

The most influence-resistant setting is the *unanimous* thresholds setting. That is, the threshold of each vertex is equal to its degree, i.e. $t(v) = d(v)$ for each $v \in V$. For this case, we have the following hardness result.

Theorem 5.1. *If all thresholds in a graph are unanimous, it is NP-hard to compute the optimal* Target Set Selection *solution.*

Proof. We reduce from vertex cover: Given a graph $G = (V, E)$, we want to find a subset $V' \subseteq V$ such that for each $(u, v) \in E$, $V' \cap \{u, v\} \neq \emptyset$ and $|V'|$ is as small as possible. We consider the same graph for the Target Set Selection problem, and claim that G has a vertex cover of size at most k if and only if Target Set Selection has a solution of size at most k.

For any vertex cover solution V' of G, let the target set of G be V'. Then for each $v \notin V'$, all edges incident to v are covered by the corresponding vertices in V', which implies v can be active. Thus, by targeting V', all vertices are active at the end.

On the other hand, for any Target Set Selection solution V', we argue that V' is a vertex cover as well. For any edge (u, v), if neither u nor v is in V', both u and v can not be activated, since their threshold is equal to their degree, which is a contradiction. ∎

Indeed, as can be seen from the above proof, the Target Set Selection problem with unanimous thresholds is equivalent to vertex cover. Thus, it admits a 2-approximation algorithm [30], and is NP-hard to approximate better than 1.36 [8].

6 Tree Structure

When the underlying graph $G = (V, E)$ is a tree, the Target Set Selection problem can be solved in polynomial time. The basic observation is that for any leaf $v \in V$, $t(v)$ is equal to 1. Thus, at most one of v and its parent u will be targeted in the optimal solution. Hence, we can assume without loss of generality that v is not targeted, otherwise, we can target u instead of v and get a solution of the same size. The algorithm is as follows.

Alg-Tree

```
 1. Let t'(v) = t(v), for v ∈ V
 2. Let x(v) = 0, for each leaf v ∈ V
 3. While there is x(v) not defined yet
 4.    for any vertex u where all x(·)'s of its
       children have been defined
 5.       let w be u's parent
 6.       if t'(u) ≥ 2
 7.          let x(u) = 1
 8.          let t'(w) ← t'(w) − 1
 9.       else
10.          let x(u) = 0
11.          if t'(u) ≤ 0
12.             let t'(w) ← t'(w) − 1
13. Output the target set {v ∈ V | x(v) = 1}
```

We have the following result.

Theorem 6.1. Alg-Tree *computes an optimal solution for the* Target Set Selection *problem when the underlying graph $G = (V, E)$ is a tree.*

7 Acknowledgements

I would like to thank Paul Beame, Venkatesan Guruswami, Anna Karlin and David Kempe for helpful discussions. I am especially grateful to Kunal Talwar and Evimaria Terzi– we were working on a different but related model (for which we got a $\log^{2-\epsilon} n$-hardness result)– and those ideas inspired me to complete this work. I also thank Anna, Kunal and Evimaria for their many suggestions on improving the paper.

References

[1] A. Aazami, M. D. Stilp, *Approximation Algorithms and Hardness for Domination with Propagation*, APPROX 2007, 1-15.

[2] M. Ajtai, J. Komlós, E. Szemerédi. *An $O(n \log n)$ Sorting Network (Sorting in $c \log n$ Parallel Steps)*, Combinatorica, V.3(1), 1-19, 1983.

[3] E. Arcaute, A. Kirsch, R. Kumar, D. Liben-Nowell, S. Vassilvitskii, *On Threshold Behavior in Query Incentive Networks*, EC 2007, 66-74.

[4] J. Aspnes, K. Chang, A. Yampolskiy, *Inoculation Strategies for Victims of Viruses and the Sum-of-Squares Partition Problem*, SODA 2005, 43-52.

[5] E. Berger, *Dynamic Monopolies of Constant Size*, Journal of Combinatorial Theory B, V.83(2), 191-200, 2001.

[6] N. Berger, C. Borgs, J. T. Chayes, A. Saberi, *On the Spread of Viruses on the Internet*, SODA 2005, 301-310.

[7] Z. Dezso, A.-L. Barabasi, *Halting Viruses in Scale-Free Networks*. Physical Review E, V.65, 2002.

[8] I. Dinur, S. Safra, *The Importance of Being Biased*, STOC 2002, 33-42.

[9] P. Domingos, M. Richardson, *Mining the Network Value of Customers*, KDD 2001, 57-66.

[10] P. A. Dreyer, *Applications and Variations of Domination in Graphs*, Ph.D. Thesis, Rutgers University, 2000.

[11] A. Ganesh, L. Massouli, D. Towsley, *The Effect of Network Topology on the Spread of Epidemics*, INFOCOM 2005, 1455-1466.

[12] N. Immorlica, J. M. Kleinberg, M. Mahdian, T. Wexler, *The Role of Compatibility in the Diffusion of Technologies through Social Networks*, EC 2007, 75-83.

[13] M. Kearns, L. Ortiz, *Algorithms for Interdependent Security Games*, NIPS 2003, 288-297.

[14] D. Kempe, J. Kleinberg, É. Tardos, *Maximizing the Spread of Influence through a Social Network*, KDD 2003, 137-146.

[15] D. Kempe, J. Kleinberg, É. Tardos, *Influential Nodes in a Diffusion Model for Social Networks*, ICALP 2005, 1127-1138.

[16] J. M. Kleinberg, P. Raghavan, *Query Incentive Networks*, FOCS 2005, 132-141.

[17] D. E. Knuth, *The Art of Computer Programming*, V.3, Sorting and Searching, 3rd edition, Addison-Wesley, 1997.

[18] G. Kortsarz, *On the Hardness of Approximating Spanners*, Algorithmica, V.30(3), 432-450, 2001.

[19] G. Kortsarz, R. Krauthgamer, J. R. Lee, *Hardness of Approximating Vertex-Connectivity Network Design Problems*, SIAM Journal on Computing, V.33(3), 704-720, 2004.

[20] N. Linial, D. Peleg, Y. Rabinovich, M. Saks, *Sphere Packing and Local Majorities in Graphs*, ISTCS 1993, 141-149.

[21] S. Morris, *Contagion*, Review of Economic Studies, V.67, 57-78, 2000.

[22] E. Mossel, S. Roch, *On the Submodularity of Influence in Social Networks*, STOC 2007, 128-134.

[23] R. Pastor-Satorras, A. Vespignani, *Epidemics and Immunization in Scale-Free Networks*, Handbook of Graphs and Networks: From the Genome to the Internet, edited by S. Bornholdt, H. G. Schuster, 111-130, Wiley-VCH, 2003.

[24] D. Peleg, *Local Majority Voting, Small Coalitions and Controlling Monopolies in Graphs: A Review*, in Proceedings of the 3rd Colloquium on Structural Information & Communication Complexity, 170-179, 1996.

[25] D. Peleg, *Size Bounds for Dynamic Monopolies*, Discrete Applied Mathematics, V.86, 263-273, 1998.

[26] R. Raz, *A Parallel Repetition Theorem*, SIAM J. Computing, V.27(3), 763-803, 1998.

[27] M. Richardson, P. Domingos, *Mining Knowledge-Sharing Sites for Viral Marketing*, KDD 2002, 61-70.

[28] F. S. Roberts, *Challenges for Discrete Mathematics and Theoretical Computer Science in the Defense against Bioterrorism*, Mathematical and Modeling Approaches in Homeland Security, SIAM Frontiers in Applied Mathematics Series, edited by H. T. Banks, C. Castillo-Chavez, 1-34, 2003.

[29] F. S. Roberts, *Graph-Theoretical Problems Arising from Defending Against Bioterrorism and Controlling the Spread of Fires*, DIMACS/DIMATIA/Renyi Combinatorial Challenges Conference, 2006.

[30] V. V. Vazirani, *Approximation Algorithms*, Springer-Verlag, 2001.

[31] D. J. Watts, S. H. Strogatz, *Collective Dynamics of 'Small-World' Networks*, Nature, V.393, 440-442, 1998.