On Approximating Target Set Selection

Moses Charikar[1], Yonatan Naamad[2], and Anthony Wirth[3]

1 Computer Science Department, Stanford University, Stanford, CA, USA
moses@cs.stanford.edu

2 Department of Computer Science, Princeton University, Princeton, NJ, USA
ynaamad@cs.princeton.edu

3 Department of Computing and Information Systems, The University of Melbourne, Parkville, Vic, Australia
awirth@unimelb.edu.au

Abstract

We study the Target Set Selection (TSS) problem introduced by Kempe, Kleinberg, and Tardos (2003). This problem models the propagation of influence in a network, in a sequence of rounds. A set of nodes is made "active" initially. In each subsequent round, a vertex is activated if at least a certain number of its neighbors are (already) active. In the minimization version, the goal is to activate a small set of vertices initially – a seed, or target, set – so that activation spreads to the entire graph. In the absence of a sublinear-factor algorithm for the general version, we provide a (sublinear) approximation algorithm for the bounded-round version, where the goal is to activate all the vertices in r rounds. Assuming a known conjecture on the hardness of Planted Dense Subgraph, we establish hardness-of-approximation results for the bounded-round version. We show that they translate to general Target Set Selection, leading to a hardness factor of $n^{1/2-\varepsilon}$ for all $\varepsilon > 0$. This is the first polynomial hardness result for Target Set Selection, and the strongest conditional result known for a large class of monotone satisfiability problems. In the maximization version of TSS, the goal is to pick a target set of size k so as to maximize the number of nodes eventually active. We show an $n^{1-\varepsilon}$ hardness result for the undirected maximization version of the problem, thus establishing that the undirected case is as hard as the directed case. Finally, we demonstrate an SETH lower bound for the exact computation of the optimal seed set.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases target set selection, influence propagation, approximation algorithms, hardness of approximation, planted dense subgraph

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2016.4

1 Introduction

In this paper, we address the problem of targeting individuals to spread influence (or infection) in a network. Based on an average-case assumption about finding a planted dense subgraph, we develop the first polynomial-factor lower bound for a key minimization problem. Also, for a fixed-round version, we introduce the first sub-linear-factor approximation algorithm.

Motivated by work of Domingos and Richardson [14, 20], Kempe, Kleinberg, and Tardos [18] introduced the following model. A vertex is either active (infected) or inactive (uninfected). Given an initial *seed set* of active vertices, influence proceeds in a sequence of rounds. Every vertex v has a known, deterministic threshold $\tau(v)$. A previously inactive vertex v becomes active in a particular round if in the previous round at least $\tau(v)$ neighbors of v were active. Once a vertex is active, it remains active in all subsequent rounds. Since the process (essentially) stops if there is no new active vertex in some round, there are at most n rounds.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016).
Editors: Klaus Jansen, Claire Matthieu, José D. P. Rolim, and Chris Umans; Article No. 4; pp. 4:1–4:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Both directed and undirected versions have been considered. In the directed case, the head vertex of an edge directly contributes to activating the tail vertex, but not vice versa.

The key question that arises in the study of viral marketing is Target Set Selection. Given the graph and the activation thresholds for vertices, which nodes should be initially targeted so as to spread the activation widely in the network? Specifically, we have these two problems,

- Min-TSS: Find a minimum-size seed set that leads to all vertices eventually being active;
- Max-TSS: Given budget k, find a size-k seed set that maximizes the final active set size.

Kempe, Kleinberg, and Tardos focused on the maximization problem where all thresholds are drawn randomly from a given range [18]. For the *directed* Max-TSS problem, with deterministic thresholds, they showed that obtaining an $n^{1-\varepsilon}$ approximation is **NP**-hard. By reducing from Label Cover, Chen [7] showed that the minimization problem, Min-TSS, cannot be approximated to within $2^{\log^{1-\varepsilon} n}$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\mathrm{polylog}(n)})$, even for instances with uniform thresholds of 2. In fact [13], it is **NP**-hard to approximate Label Cover within $2^{\log^{1-\varepsilon} n}$, and this hardness bound extends to Min-TSS.

Cicalese et al. [11, 10], considered versions of the problem in which the number of rounds is bounded. For graphs of bounded clique-width, given parameters α, β, and λ, they gave polynomial-time algorithms to determine whether there exists a target set of size β, such that at least α vertices are activated in at most λ rounds. Various other aspects of target set selection have been studied. For example, Coja-Oghlan et al. [12] obtained combinatorial bounds for the size of target sets in expanders, while Ben-Zwi et al. [5] obtained upper and lower bounds for this problem on low-treewidth graphs.

Our Results

We seek a polynomial-factor lower-bound for approximating Min-TSS. Obtaining such a result by reduction from known **NP**-hard problems would be a breakthrough. As we point out in Section 7, (the bounded-round version of) Min-TSS belongs to the *MMSA hierarchy* [13]. A prevailing definition of this class of problems is: Given a monotone Boolean circuit, minimize the number of inputs set to `True` so that the circuit evaluates to `True`. No problem in the MMSA hierarchy is currently known to have a polynomial hardness result.[1] We derive a polynomial-factor hardness result for Min-TSS from an average-case hardness conjecture for the Planted Dense Subgraph problem (see Section 3.2)

In Section 4.2, we show the following hardness result for the bounded-round version of Min-TSS. Assuming the Planted Dense Subgraph Conjecture, for every $\varepsilon > 0$, r-round Min-TSS is $n^{1/2-1/2^{\lfloor r/2 \rfloor}-\varepsilon}$ hard to approximate. In Section 4.3, we prove the corollary that, assuming the Planted Dense Subgraph Conjecture, Min-TSS (with unbounded number of rounds) is **NP**-hard to approximate within $n^{1/2-\varepsilon}$ for every $\varepsilon > 0$.

After this, in Section 5, we provide an $O((\tau_{\max}/\tau_{\min})^{1-1/r} n^{1-1/r} \log^{1/r} n)$ approximation algorithm for r-round Min-TSS, where $\tau_{\max}$ and $\tau_{\min}$ are the maximum and minimum thresholds in the instance, respectively. Subsequently, we show in Section 6 that *undirected* Max-TSS is as hard as the *directed* version, giving an $n^{1-\varepsilon}$ hardness for undirected Max-TSS. Finally, in Section 8, we reuse ideas from Section 4.3 to give an $O(n^k)$ SETH-lower

[1] As explained in Section 7, another variant of MMSA, where the circuits may use non-monotone gates in computing a monotone function, is known to admit problems that are $O(n^{1-\varepsilon})$ hard to approximate unless $\mathbf{NP} \subseteq \mathbf{DTIME}[n^{O(\log n)}]$ [21].

bound on the time needed to find contagious sets of size k, for $k = O(1)$, which is tight up to near-linear factors.

2 Outline of Key Technical Ideas

The main goal of our paper is to obtain a better understanding of the complexity of MIN-TSS. We focus in particular on the bounded-round variant, which has previously only been studied from the point of view of fixed-parameter tractability. As we show, understanding the hardness of the bounded-round variant is a good stepping stone in obtaining hardness results for (unbounded) MIN-TSS.

2.1 Hardness of Min-TSS

A $2^{\log^{1-\varepsilon} n}$ hardness factor is known for MIN-TSS, but there are no non-trivial approximation upper bounds for this problem. We obtain a polynomial hardness factor based on the planted dense subgraph conjecture by introducing and exploiting a recursive version. Roughly speaking, the planted dense subgraph conjecture says that it is hard to distinguish random graphs of degree n^α from such graphs where $k = \sqrt{n}$ vertices have a planted dense subgraph of degree k^β, for $\beta < \alpha$.

To assist our exposition of the intuition, we will sacrifice a little accuracy, and think of $\beta = \alpha = 1/2$. So (pretend that) it is hard to distinguish random graphs of degree $\sqrt{n}$ (unplanted) from such graphs where a subset of $\sqrt{n}$ vertices have a planted random graph of degree $n^{1/4}$ (planted) – a factor of roughly $n^{1/4}$ larger than the expected vertex degree in a random subgraph of that size. Consider these graph families as inputs to MIN-TSS with the threshold set to a large constant. In the unplanted case, we can show that at least $n^{1/2}$ vertices must be initially activated in order to activate all vertices in a constant number of rounds. On the other hand, in the planted case, a target set of roughly $n^{1/4}$ vertices will activate all vertices in at most four rounds: in two rounds, all the vertices in the planted set will be activated, and in two more rounds all vertices in the graph will be activated. This leads to an $\Omega(n^{1/4})$ hardness result for 4-round MIN-TSS.

In order to show stronger hardness results (for more rounds), we recurse. Consider a recursively planted dense subgraph instance where we start with a random graph of degree $n^{1/2}$ and plant a random subgraph of $n^{1/2}$ vertices and degree $n^{1/4}$. Within this, we plant a random subgraph of $n^{1/4}$ vertices and degree $n^{1/8}$, and so on; the last subgraph in this sequence is on $n^{1/2^t}$ vertices with degree $n^{1/2^{t+1}}$. (Again, to facilitate the explanation, the parameters are slightly inaccurate.) We show that the planted dense subgraph conjecture implies the hardness of the recursive version. As before, in the unplanted case, at least $n^{1/2}$ vertices must be initially activated in order to activate all vertices in a constant number of rounds. In the recursively planted case, activating roughly $n^{1/2^{t+1}}$ vertices will activate all vertices in at most $2(t+1)$ rounds: in two rounds, all the vertices of the inner-most planted subgraph will be activated, in a further two rounds, all vertices of the second-deepest planted subgraph will be activated, and so on. This establishes a hardness of $n^{1/2-1/2^{t+1}}$ for $2(t+1)$-round MIN-TSS.

Via a direct reduction, we show that the hardness results for r-round MIN-TSS imply hardness for (unbounded-round) MIN-TSS. For every constant $\varepsilon > 0$, we show that MIN-TSS is $n^{1/2-\varepsilon}$ hard to approximate. The reduction is easiest to describe with directed edges, but these can be simulated with a gadget comprising undirected edges. Given an instance of r-round MIN-TSS on $G(V, E)$, the construction consists of $2r+1$ layers of copies of vertices of V: layers $S_0, S_1, \ldots, S_r$ are interleaved with layers $M_0, \ldots, M_{r-1}$. Layer S_0 contains the seed set and for each $i > 0$, vertices active in S_i represent the set of vertices active in G after i rounds. Each layer M_i contains "memory" vertices: the copy of vertex v in this layer is activated if a copy of v is activated in at least one of the previous layers, $S_0, \ldots, S_i$. Between layers M_i and S_{i+1} we place a bipartite copy of the original graph G, and the copy of v in S_{i+1} has the same threshold as v in the original graph. This simulates one round of the activation process on the original graph. Finally, we place a complete bipartite graph directed from vertices in S_r to those in S_0: each vertex in S_0 has threshold $|V|$, so, unless in the seed set, it is activated if and only if all vertices in S_r are active.

The recursive planted dense subgraph construction and its application are new. It would be interesting to understand what sequences of parameters in the recursive construction imply indistinguishability between the unplanted case and the recursively planted case. Our hybrid-like argument for indistinguishibility relies on the fact that the recursively planted instances are scaled-down copies of hard instances of planted dense subgraph; however, this might not be needed in the recursive construction. Together with a better understanding of planted dense subgraph in regimes where the planted subgraph size $k \gg \sqrt{n}$, this could lead to tight hardness results for Min-TSS. Although we are unaware of an algorithm for Min-TSS with approximation factor $o(n)$, our current construction establishes hardness at most $n^{1/2}$ because the unplanted case has $\mathrm{OPT} = O(n^{1/2})$. Establishing stronger hardness results will need constructions where OPT is much higher; this could be achieved by setting the order of the planted subgraph to be $n^{1-\varepsilon}$, with appropriate choices of degrees for the planted and ambient graphs. Because the assumptions were originally formulated to capture the hard case for approximating densest subgraph, the stated assumptions in the literature about hardness of planted dense subgraph only apply to $k \leq \sqrt{n}$. A more comprehensive understanding of planted dense subgraph in the $k > \sqrt{n}$ regime would be interesting in its own right, and could lead to an almost-tight $n^{1-\varepsilon}$ hardness for Min-TSS.

2.2 Approximation of r-round Min-TSS

Until now, no non-trivial approximation algorithms for bounded-latency Min-TSS is known. When all thresholds are the same, our algorithm follows a greedy approach and obtains an $\tilde{O}(n^{1-1/r})$-approximation for r-round Min-TSS. In general, the approximation factor also depends on the ratio of the largest to smallest thresholds. The challenge in applying a greedy approach to Min-TSS is quantifying the progress made in adding a single vertex to the seed set. Indeed, a single vertex may have negligible impact until several other vertices are picked. The key idea behind our algorithm is a potential function, called *hunger*, that guides the algorithm. Given a seed set S and a bound r on the number of rounds, a vertex v has positive hunger if and only if it remains inactive after r rounds. In this case, v's hunger is the number of additional neighbors that need to be active after $r - 1$ rounds, in order to activate v in the next round.

Our algorithm chooses the seed set in two phases, based on a parameter β that we choose appropriately to obtain the approximation guarantee. In the first phase, we greedily pick vertices that have more than β neighbors that would not otherwise become active within one round. The first phase ensures that in the "residual" graph, degrees of vertices are bounded by β. With this, we can relate the size of the seed set picked in the second phase to the optimum seed set size. In the second phase, given the current seed set, we repeatedly pick vertices greedily to reduce the total hunger. Our analysis shows that one of the vertices of the optimal solution reduces the total hunger by a significant quantity. However, we lose a factor of β^{r-1} in relating the drop in the total hunger to the effect of this vertex on the optimal solution. Consequently, in the second phase, the bound on the size of the picked seed set is (roughly) a factor β^{r-1} times optimal.

3 Preliminaries

3.1 Formal definition and notation

An instance of Target Set Selection (TSS) is an n-vertex (di)graph $G = (V, E)$ coupled with a threshold function $\tau : V \to \mathbb{Z}^+$. Seed set $S \subset V$ comprises the vertices that are *active in round* 0. For all $t > 0$, a vertex $v \in V$ is called *active in round* t if either it is active in round $t - 1$ or at least $\tau(v)$ of its (in)neighbors were active in round $t - 1$. The r^{th}*-round activation family* of a seed set $S \subset V$, denoted by $\mathcal{A}_r(S)$, comprises those vertices active in round $t = r$, conditioned on exactly the vertices in S being active at time $t = 0$. The *activation family* of seed set S is $\mathcal{A}_\infty(S) \equiv \lim_{r\to\infty} \mathcal{A}_r(S)$. By monotonicity, and the Markovian nature of this process, it is easy to verify that $\mathcal{A}_\infty(S)$ is equivalent to $\mathcal{A}_{n-1}(S)$. We study three variants of the Target Set Selection problem.

TSS: Given G and τ, what is the size of the smallest seed set S for which $\mathcal{A}_\infty(S) = V$?

RTSS$_r$: Given G and τ, what is the size of the smallest seed set S for which $\mathcal{A}_r(S) = V$?

MaxTSS: Given G, τ, and $k > 0$, conditioned on $|S| \leq k$, what is the largest value of $|\mathcal{A}_\infty(S)|$?

We sometimes refer to TSS as Min-TSS, and RTSS stands for r-Round TSS. For both minimization problems, a seed set whose activation family (within the round limit, if any) is the whole graph is called a *contagious* set. When all thresholds are equal, we may abuse notation and let τ itself be an integer. All tuples in this paper are written thus $\vec{a}$, and are indexed in two ways. If $\vec{\xi} = (\xi_1, \xi_2, \cdots, \xi_m)$ is a tuple and $1 \leq i < j \leq m$, we let $\vec{\xi}_i^j$ denote the contiguous sub-tuple $(\xi_i, \xi_{i+1}, \cdots, \xi_j)$. Sometimes, we are interested in the suffix of the tuple, so we can index the final elements thus: $\vec{\xi}_{(-i)}^{(-1)} = (x_{m-(i-1)}, x_{m-(i-2)}, \ldots, x_m)$.

3.2 Planted Dense Subgraph Conjecture

The Planted Dense Subgraph (PDS) problem is a generalization of Planted Clique, in which the goal is to distinguish a $G(n, p)$, Erdős-Rényi, random graph from one that contains a planted dense Erdős-Rényi component. Formally, an instance of the problem PDS(n, k, α, β) is parameterized by the graph order n, the subgraph order k, and log-densities $\alpha, \beta \in (0, 1)$. We are then asked for an algorithm that with high probability distinguishes between these two families of random graphs:

Unplanted: An Erdős-Rényi random graph $G(n, n^{\alpha-1})$ (i.e., a random graph with expected degree $\approx n^\alpha$).

Planted: An Erdős-Rényi random graph $G(n, n^{\alpha-1})$ from which k vertices are chosen uniformly at random and their induced subgraph is replaced by an instance of $G(k, k^{\beta-1})$.

The input to the PDS(n, k, α, β) problem is a graph, with the *promise* that with probability 1/2 it is drawn from the Planted distribution and with probability 1/2 it is drawn from the Unplanted distribution. The output is True or False, indicating whether the graph has a planted subcomponent. An algorithm *solves* the problem if, for some $\varepsilon > 0$, independent of n, it attains an ε advantage over random guessing. That is, with probability at least $1/2 + \varepsilon$, it correctly states from which of the two distributions the input graph was drawn. This statement about probability is over the joint distribution of the input graph and the random choice sequence of the algorithm.

As observed by Bhaskara et al. [6], PDS(n,k,α,β) admits a simple polynomial-time deterministic algorithm when $\beta > \alpha$; for $k > \sqrt{n}$, an eigenvalue approach works with a weaker condition on β. For $\beta < \alpha$ and $k \le \sqrt{n}$, there is no quasipolynomial-time algorithm (deterministic or randomized) known for PDS, giving rise to the Planted Dense Subgraph Conjecture, viz.

▶ **Conjecture 1.** *For every $\varepsilon > 0$, $k \le \sqrt{n}$, and $\beta < \alpha$, no probabilistic polynomial-time (PPT) algorithm can, with advantage greater than ε, solve* PDS(n,k,α,β).

This conjecture is attractive because the current best-known algorithms for the problem require $2^{n^{\Theta(1)}}$ time (as opposed to $n^{\Theta(\log n)}$ for the well-studied Planted Clique problem).

Similar conjectures have previously been made in different contexts in theoretical computer science [2, 3, 4]. The precise form of the conjecture we state is very similar to the conjecture stated by Awasthi et al. [4]. As we show in Section 4, Planted Dense Subgraph also naturally lends itself to showing hardness for the bounded-round version of Min-TSS, which in turn leads to a hardness result for the (unbounded) Min-TSS problem.

4 Hardness of Min-TSS

In this section, we prove that the Planted Dense Subgraph (PDS) conjecture implies that for all $\varepsilon > 0$, there is no probabilistic polynomial-time algorithm for Min-TSS that achieves an approximation factor of $O(n^{1/2-\varepsilon})$. We first show that the Planted Dense Subgraph conjecture implies the hardness of a recursive version, and we use this recursive version to show hardness for Min-TSS.

4.1 Recursive extension of PDS

To simplify notation, we define a right-associative operator, $\triangleleft$, on distributions of graphs. Suppose there are two distributions on graphs, $\mathcal{G}_1$, on graphs of order n, and $\mathcal{G}_2$, on graphs of order n', with $n > n'$. The distribution $\mathcal{G}_1 \triangleleft \mathcal{G}_2$ is defined thus:

> Draw a graph G_1 from $\mathcal{G}_1$ and a graph G_2 from $\mathcal{G}_2$, then choose uniformly at random a subset S' of vertices of size n' from G_1 and *replace* its induced subgraph with G_2.

Hence PDS(n,k,α,β) asks us to state whether a graph is drawn from $G(n,n^{\alpha-1}) \triangleleft G(k,k^{\beta-1})$ (True) or from $G(n,n^{\alpha-1})$ (False).

In the definition of PDS(n,k,α,β), one consequence of the random construction of both $G(n,n^{\alpha-1})$ and $G(k,k^{\beta-1})$ is that the planting process can be naturally recursed. For every pair of length-m tuples $\vec{n} = (n_1,n_2,\ldots,n_m)$ – the subgraph orders – and $\vec{\alpha} = (\alpha_1,\alpha_2,\ldots,\alpha_m)$ – the subgraph log-densities – with, for each i, $n_i \in \mathbb{Z}^+$, $n_i > n_{i+1}$, and $\alpha_i \in (0,1)$, we define the PDS$^m(\vec{n},\vec{\alpha})$ distribution via the recurrence

$$\mathrm{PDS}^m(\vec{n},\vec{\alpha}) = \begin{cases} G(n_1,{n_1}^{\alpha_1-1}) \triangleleft \mathrm{PDS}^{m-1}(\vec{n}^{(-1)}_{(-m+1)},\vec{\alpha}^{(-1)}_{(-m+1)}) & \text{if } m>1; \\ G(n_1,{n_1}^{\alpha_1-1}), & \text{otherwise.} \end{cases}$$

We also define the (eponymous) PDS$^m(\vec{n},\vec{\alpha})$ problem: distinguish with ε advantage graphs drawn from the PDS$^m(\vec{n},\vec{\alpha})$ distribution from those drawn simply from PDS$^1(n_1,\alpha_1) = G(n_1,n^{\alpha_1-1})$. More formally, under the promise that with probability 1/2 the graph is from the former distribution, and with probability 1/2 from the latter, an algorithm solves the problem if with probability at least $1/2+\varepsilon$ it correctly states which of the distributions the graph came from. Setting $m=2$, $\vec{n}=(n,k)$, and $\vec{\alpha}=(\alpha,\beta)$ recovers exactly PDS(n,k,α,β).

We now show that for monotonically decreasing log-densities, recursive planting of small, but polynomially sized, subgraphs leads to a problem no simpler than PDS.

▶ **Lemma 2.** *Assuming Conjecture 1, if $m \geq 2$ is a constant, $\alpha_i > \alpha_{i+1}$ for every $i < m$, and for some constant $c > 0$, $n_{i+1} \in [n_i^c, \sqrt{n_i}]$, then no PPT algorithm can solve the* $\mathrm{PDS}^m(\vec{n}, \vec{\alpha})$ *problem with ε advantage.*

Proof (by contradiction). Consider the *minimum* m for which some algorithm A solves $\mathrm{PDS}^m(\vec{n}, \vec{\alpha})$ with ε advantage. We show how to construct an algorithm that attains an $\varepsilon' > 0$ advantage for the problem $\mathrm{PDS}^{m-1}(\vec{n}^{(-1)}_{(-m+1)}, \vec{\alpha}^{(-1)}_{(-m+1)})$, contradicting the minimality of m. For a distribution $\mathcal{H}$, let $\mathcal{F}(\mathcal{H})$ be the distribution $G(n_1, n_1^{\alpha_1 - 1}) \triangleleft \mathcal{H}$. Hence

$$\mathcal{F}\left(\mathrm{PDS}^{m-1}(\vec{n}^{(-1)}_{(-m+1)}, \vec{\alpha}^{(-1)}_{(-m+1)})\right)$$

is $\mathrm{PDS}^m(\vec{n}, \vec{\alpha})$ and $\mathcal{F}\left(G(n_2, {n_2}^{\alpha_2 - 1})\right)$ is $\mathrm{PDS}^2\left((n_1, n_2), (\alpha_1, \alpha_2)\right)$. But, of course, this $\mathcal{F}$ operator represents a (randomized) polynomial-time-computable operation on a graph, once drawn from distribution $\mathcal{H}$. Assuming the existence of algorithm A, we propose algorithm $A_{\mathcal{F}}$ for $\mathrm{PDS}^{m-1}(\vec{n}^{(-1)}_{(-m+1)}, \vec{\alpha}^{(-1)}_{(-m+1)})$:

Given graph H, apply algorithm A to $\mathcal{F}(H)$ and return A's answer.

For the following three distributions, let p_i be the probability that A reports True when the graph is drawn from distribution i.

1. $G(n_1, {n_1}^{\alpha_1 - 1})$;
2. $\mathcal{F}\left(G(n_2, {n_2}^{\alpha_2 - 1})\right) = \mathrm{PDS}^2\left((n_1, n_2), (\alpha_1, \alpha_2)\right)$;
3. $\mathcal{F}\left(\mathrm{PDS}^{m-1}(\vec{n}^{(-1)}_{(-m+1)}, \vec{\alpha}^{(-1)}_{(-m+1)})\right) = \mathrm{PDS}^m(\vec{n}, \vec{\alpha})$

Since we assumed the Planted Dense Subgraph Conjecture, the probability of returning the correct answer when distinguishing between distributions 1 and 2 is at most $1/2 + o(1)$. That is, $(1 - p_1)/2 + p_2/2 \leq 1/2 + o(1)$. On the other hand, since A solves $\mathrm{PDS}^m(\vec{n}, \vec{\alpha})$, with ε advantage, $(1 - p_1)/2 + p_3/2 \geq 1/2 + \varepsilon$.

Consider algorithm $A_{\mathcal{F}}$, it applies $\mathcal{F}$ to its input graph and sends it to A. Although A was promised a graph that was with probability 1/2 from the first distribution and with probability 1/2 from the third[2], its input is merely a graph with nonzero mass in each of the two distributions, and thus it outputs some value in probabilistic polynomial time[3]. The probability of algorithm $A_{\mathcal{F}}$ correctly reporting True is (by definition) $(1 - p_2)/2 + p_3/2$, which, from the two previous inequalities, is easily seen to be at least $1/2 + \varepsilon - o(1)$. Hence, for sufficiently large n, Algorithm $A_{\mathcal{F}}$ is a PPT algorithm with advantage $\varepsilon/2$ for the $\mathrm{PDS}^{m-1}(\vec{n}^{(-1)}_{(-m+1)}, \vec{\alpha}^{(-1)}_{(-m+1)})$ problem, contradicting the minimality of m. ◀

4.2 Hardness of fixed-round TSS

We now prove the following theorem on the hardness of r-round Min-TSS.

[2] This application of a promise oracle to non-promise-satisfying inputs is an example of a *non-smart* reduction [16].

[3] Although the positivity of the mass on some order-n graph might seem like a technicality, it is not intrinsic to the analysis. For other, similar, distributions of graphs where $\mathcal{F}(H)$ may have 0 probability in either distribution, we can run A up to its polynomial-time upper-bound for promise-satisfying instances and return False if it has failed to return by then. The rest of the analysis proceeds identically, up to the necessary modifications needed to handle the new ensemble of distributions.

▶ **Theorem 3.** *Assuming the* Planted Dense Subgraph Conjecture, *for every $\varepsilon > 0$, no PPT algorithm can approximate r-round* Min-TSS *to within a factor of $O(n^{1/2-1/2^{\lfloor r/2 \rfloor}-\varepsilon})$.*

The general idea is as follows. We first prove (in Lemma 4) a lower bound on the size of r-round contagious sets on $G(n_1, {n_1}^{\alpha_1 - 1})$ for particular (ranges of) values of n_1 and α_1. Afterwards, we argue (in Lemma 5) that for some appropriate setting of $\vec{n} = (n_1, n_2, \cdots, n_{\lfloor r/2 \rfloor})$ and $\vec{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_{\lfloor r/2 \rfloor})$, the $n_{\lfloor r/2 \rfloor}$ vertices in the densest component of $\mathrm{PDS}^{\lfloor r/2 \rfloor}(\vec{n}, \vec{\alpha})$ must form a contagious set of size $n_{\lfloor r/2 \rfloor}$. Theorem 3 then follows from a direct application of Lemma 2.

▶ **Lemma 4.** *For all $\alpha \in (0, 1/2]$ and all positive integers r, τ, the r-round* Min-TSS *instance* $\mathrm{RTSS}_r(G(n, n^{\alpha-1}), \tau)$ *has $|OPT| = \tilde{\Omega}(n^{\beta})$, where $\beta = (\tau(1-\alpha) - 1)/(\tau - 1)$.*

Proof. For a seed set S of size $\tilde{O}(n^{\beta})$, chosen uniformly at random (u.a.r.), and a vertex $v \notin S$, chosen u.a.r., the probability that v has k neighbors in S is, by the union bound, at most $\binom{|S|}{k}(n^{\alpha-1})^k = \tilde{O}\left((n^{\beta} n^{\alpha-1})^k\right)$ (for constant k), which decreases geometrically with k. Thus, the number of vertices newly activated after one round, $|\mathcal{A}_1(S) \setminus S|$, follows a binomial distribution with mean $\mu = \tilde{O}(n(n^{\beta} n^{\alpha-1})^{\tau}) = \tilde{O}(n^{1+(\alpha+\beta-1)\tau})$. Because $(\alpha+\beta-1)\tau = \beta - 1$, this expression simplifies to $\tilde{O}(n^{\beta})$. Chernoff bounds tell us that the probability that $|\mathcal{A}_1(S) \setminus S|$ exceeds its expectation by a $\log^2 n$ factor is bounded by

$$\exp[\mu(\log^2 n - 1 - 2\log^2 n(\log\log n))] < \exp[\mu(-\log^2 n)] = n^{-\mu \log n}\,.$$

Given it has size $\tilde{O}(n^{\beta})$, there are $\tilde{O}\left(\binom{n}{n^{\beta}}\right)$ choices of seed set S. By the union bound, the probability that at least one of them activates more than $\tilde{O}(n^{\beta})$ vertices within one round is $1/n^{\Omega(1)}$. By induction, after a constant number of activation rounds, for all $\varepsilon > 0$, the number of active nodes is with high probability at most $o(n^{\beta+\varepsilon})$. Hence, with high probability, no seed set of size $\tilde{O}(n^{\beta})$ is an r-round contagious set. ◀

▶ **Lemma 5.** *Suppose, in an instance of the r-round* Min-TSS *problem, every vertex shares the same constant threshold τ, with*

$$\rho \equiv \lfloor r/2 \rfloor, \qquad \varepsilon_0 \in \left(0, \frac{1}{3(\tau+1)^2}\right), \qquad \vec{n} = \left(n, \sqrt{n}, \sqrt{\sqrt{n}}, \ldots, \sqrt[2^{\rho}]{n}\right), \ \textit{and}$$
$$\vec{\alpha} = \left(\frac{1}{2}, \frac{1}{2} - \frac{\varepsilon_0}{1+\rho}, \frac{1}{2} - \frac{2\varepsilon_0}{1+\rho} \ldots, \frac{1}{2} - \frac{\rho\varepsilon_0}{1+\rho}\right).$$

If $G \sim \mathrm{PDS}^{\rho}(\vec{n}, \vec{\alpha})$, then $\mathrm{RTSS}_r(G, \tau)$ has $|OPT| = O(\sqrt[2^{\rho}]{n})$.

Proof. For $i \in 1, 2, \cdots, 1+\rho$, let G_i be the depth-i planted component of G, i.e., the graph of order $k_i = \sqrt[2^{i-1}]{n}$ and average degree ${k_i}^{1/2-(i-1)\varepsilon_0/(1+\rho)}$ planted in G. Further, for every i, define $a_i = {k_i}^{-1/2-\varepsilon_0}$, making a_i a lower bound on the edge probability in G_i. For $i \in \{1, 2, \ldots, \rho\}$, choosing a subset of size $\sqrt{k_i} = k_{i+1}$ of G_i's vertices u.a.r. to be the seed set S, leads to random variable $|\mathcal{A}_1(S) \setminus S|$ following a binomial distribution with $\mu = \Omega(k_i \cdot (\sqrt{k_i} a_i)^{\tau})$. Each indicator of membership in $\mathcal{A}_1(S) \setminus S$ is a Bernoulli random variable with probability (of activation) at least $\binom{\sqrt{k_i}}{\tau}{a_i}^{\tau}$. The expression for the mean is thus $\Omega(k_i(\sqrt{k_i} \cdot {k_i}^{-1/2-\varepsilon_0})^{\tau})$, which is $\Omega({k_i}^{1-\tau\varepsilon_0})$. The probability that, for a randomly chosen seed set of size $\sqrt{k_i}$, fewer than $\mu/2$ vertices become active is, again by the Chernoff bounds, less than $e^{-\Omega(\mu)} = n^{-\Omega(\mu/\log n)} = n^{-\Omega({k_i}^{1-\tau\varepsilon_0}/\log n)} < n^{-k_i^{2/3}} \ll n^{-\sqrt{k_i}-2}$. Every remaining inactive vertex in G_i would now have $\hat{d} = \mu a_i = \tilde{\Omega}({k_i}^{1/2-\varepsilon_0(\tau+1)}) = \tilde{\Omega}({k_i}^{1/2-1/(3(\tau+1))})$ activated neighbors in expectation. Thus, with high probability, no vertex has fewer than $\tilde{\Omega}({k_i}^{1/3})$ active neighbors. Since each vertex has so many active neighbors, with high

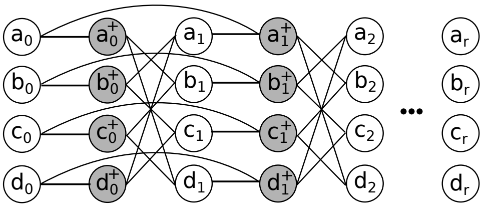

Figure 1 Sample of the first few layers of the r-round Min-TSS to TSS reduction. The i-th stage vertices are white and the memory vertices are grey. All drawn arcs are oriented rightwards. Not shown: directed edges from each vertex in S_r to all those in S_0.

probability, all of G_i's vertices becomes active within two rounds of all of G_{i+1}'s vertices becoming active. By induction, activating all of $G_{r'+1}$'s vertices will with high probability activate all of G_1's vertices after $2\rho \leq r$ rounds. Since $|V(G_{\rho+1})| = n^{1/2^\rho}$, the lemma follows. ◀

Proof of Theorem 3. Given an instance G of the $\text{PDS}^m(\vec{n}, \vec{\alpha})$ problem with $\vec{n}$ and $\vec{\alpha}$ satisfying the conditions of Lemma 5 (and thus Lemma 2), we choose a threshold τ satisfying $(\tau/2 - 1)/(\tau - 1) > 1/2 - \varepsilon/2$ and generate the r-round Min-TSS instance $\text{RTSS}_r(G, \tau)$. If G comes from the unplanted distribution, an application of Lemma 4 provides a lower bound of $\tilde{\Omega}(n^{1/2-\varepsilon/2})$ on the size of some optimal seed set, OPT. On the other hand, if G comes from the planted distribution, Lemma 5 provides an upper bound of $O(n^{1/2^{\lfloor r/2 \rfloor}})$ on the size of OPT. Thus a PPT algorithm with approximation factor in

$$\tilde{O}(n^{1/2-\varepsilon/2}/n^{1/2^{\lfloor r/2 \rfloor}}) = \tilde{O}(n^{1/2-1/2^{\lfloor r/2 \rfloor}-\varepsilon/2}) = O(n^{1/2-1/2^{\lfloor r/2 \rfloor}-\varepsilon})$$

for r-round Min-TSS can distinguish the two cases, contradicting the Planted Dense Subgraph Conjecture. ◀

4.3 Hardness of round-unbounded TSS

We now show that for every constant r the general form of Min-TSS is, up to a constant factor, at least as hard to approximate as r-round Min-TSS.

▶ **Theorem 6.** *An $O(f(n))$-approximation algorithm for* Min-TSS *can be transformed into an $O(f(n))$-approximation algorithm for r-round* Min-TSS.

Proof. Our reduction relies heavily on directed edges. The hardness for undirected Min-TSS follows by simulating each directed edge with a *directed-edge gadget*, as shown in the left part of Figure 2. Given an instance $\text{RTSS}_r(G = (V, E), \tau)$ of r-round Min-TSS, we create an instance $\text{TSS}(G', \tau')$ of TSS as follows, and as depicted in Figure 1.

1. For $i = 0, 1, \ldots, r$, and for each $v \in V$, there is a vertex v_i. Collectively, the vertices $\{v_i\}_{v \in V}$ constitute S_i, the "i-th stage" vertices of G'.
2. For $j = 0, 1, \ldots, r-1$, and for each $v \in V$, there is a vertex v_j^+. Collectively, the vertices $\{v_j^+\}_{v \in V}$ constitute M_j, the "j-th stage memory vertices" of G'.
3. For $i = 0, 1, \ldots, r-1$, and for $j = i, \ldots, r-1$, there is a directed edge (gadget) from v_i to v_j^+.
4. For $i = 0, 1, \ldots, r-1$, and for each pair (u, v) in E, there is a directed edge (gadget) from u_i^+ to v_{i+1}.
5. For each vertex $x \in S_r$ and for each $y \in S_0$ there is a directed-edge (gadget) from x to y.

6. For each $v \in V$, and $i = 0, \ldots, r$ and $j = 0, \ldots, r-1$, $\tau'(v_0) = |V|$, $\tau'(v_i) = \tau(v)$, and $\tau'(v_j^+) = 1$.

The goal of the construction is for the active vertices in S_0 to represent (at least initially) the seed set S and for the active vertices in S_i to represent $\mathcal{A}_i(S)$ in G. If S_r is entirely active – representing $\mathcal{A}_r(S) = V$ – this set in turn activates all of S_0, causing all vertices in G' to become active. With the exception of those from S_r to S_0, the *directed* edges ensure that vertices in some stage cannot activate those in an "earlier" stage.

In designing G', simply linking vertices in S_i to those in S_{i+1} with directed edges is inappropriate. We need to ensure that a vertex in G that is active in round i, represented by an active vertex in stage i of G', is (also) represented as active in stage $i+1$ of G'. That is, if all v_i are active for $v \in \mathcal{A}_i(S)$, then all v_{i+1} are also active for $v \in \mathcal{A}_i(S)$. To assist us in this, the memory vertex v_j^+, in M_j is active whenever there is some v_i, with $i < j$, that is active.

Thus, a contagious set for the RTSS instance corresponds exactly to a contagious set in S_0. To show that G' does not contain smaller contagious sets, observe that activating v_0 weakly dominates activating each of v_i, for $i > 0$, and v_i^+, for $i \geq 0$. Vertex v_i will become active within i rounds anyway, so there is no benefit in activating it earlier; a similar argument applies to v_i^+. Thus, a contagious set containing vertices in $V(G') \setminus S_0$ can be transformed into a no-larger contagious set entirely inside S_0. Hence the optimal values of both this instance G' of TSS and of the initial instance G of RTSS are equal. ◀

Picking r sufficiently large, Theorem 3 and Theorem 6 together imply the following theorem.

▶ **Theorem 7.** *Assuming the* Planted Dense Subgraph Conjecture, *for no $\varepsilon > 0$ can a PPT algorithm approximate* Min-TSS *to within a factor of $O(n^{1/2-\varepsilon})$.*

5 Approximation of r-round Min-TSS

In Section 4.2, we showed that the Planted Dense Subgraph Conjecture implies $O(n^{1/2-1/2^{\lfloor r/2 \rfloor}-\varepsilon})$ hardness for the r-round Min-TSS problem, even in instances where all vertices have thresholds bounded by a constant. In this section, we complement the hardness result with an $\tilde{O}(n^{1-1/r})$ approximation algorithm for such graphs, viz.

▶ **Theorem 8.** *For every $r \in \mathbb{Z}^+$, there is a polynomial-time algorithm approximating r-round TSS to within a factor of $O((\tau_{\max}/\tau_{\min})^{1-1/r} n^{1-1/r} \log^{1/r} n)$.*

Overall, the algorithm follows a two-step process. A *degree-reduction* step – where we greedily add vertices to the seed set S whose neighborhoods contain many vertices that would not be in $\mathcal{A}_1(S)$ for the S thus far – followed by a *greedy* step – where we greedily select vertices that most reduce a potential function we call the *total hunger*. Roughly, if the seed set S would activate vertex v in r rounds, then v's hunger is zero; otherwise, its is the number of active neighbors v still "lacks" at the end of the $(r-1)^{\text{th}}$ round. The total hunger of the graph is the sum of these vertex hungers. We then argue that each vertex chosen in the degree-reduction step necessarily reduces the total hunger by a large amount. After these degree-reducing vertices have been added to the seed set, the residual graph has bounded degree, which means that the greedy algorithm is effective.

Formally, for a vertex v and seed set S, define v's *hunger* thus, where $\delta(v)$ is the set of v's neighbors:

$$h_{S,r}(v) = \begin{cases} 0, & \text{if } v \in \mathcal{A}_r(S) \\ \tau(v) - |\{u : u \in \delta(v), u \in \mathcal{A}_{r-1}(S)\}|, & \text{otherwise.} \end{cases}$$

Algorithm 1 Algorithm for computing a target set: parameter β specified below.

```
S_1 ← ∅, S_2 ← ∅  {Initialization}
while There exists some vertex $u$ with more than $\beta$ $(S_1, 1)$-hungry neighbors do
    Add $u$ to $S_1$. {Degree-reduction step}
end while
while There exists an $(S_1 \cup S_2, r)$-hungry vertex do
    Add to $S_2$ the vertex that most reduces the total hunger of $(S_1 \cup S_2, r)$-hungry vertices.
    {Greedy step}
end while
Return $S_1 \cup S_2$.
```

Likewise, define the total (S, r) hunger in the graph thus, $h_r(S) = \sum_{v \in V} h_{S,r}(v)$. Vertex v is called (S, r)-hungry if $h_{S,r}(v) > 0$. Algorithm 1, including a parameter β that will be defined below, details the construction of the two components of the seed set, S_1 and S_2.

Since Step 5 only terminates when there are no more $(S_1 \cup S_2, r)$-hungry vertices, the algorithm returns a valid contagious set. We now bound the sizes of sets S_1 and S_2.

▶ **Lemma 9.** $|S_1| \leq n\tau_{\max}/\beta$

Proof of Lemma 9. The initial total 1-round hunger is bounded by $n\tau_{\max}$. Each element added to S_1 reduces this quantity by at least β, and since the total 1-round hunger is non-negative, $|S_1|\beta \leq n\tau_{\max}$. ◀

To analyze the size of S_2, we focus on the sub-problem induced by including S_1 in the seed set. The residual problem, Residual(S_1), is an instance of r-round Min-TSS on $V \setminus S_1$ in which, because the process is Markovian, $\tau(v)$ becomes $h_{S_1,1}(v)$. This residual instance has the unusual feature that zero-threshold vertices become active in one round. The degree-reduction step ensures that no vertex in Residual(S_1) has more than β neighbors in $V \setminus S_1$, which leads to the following lemma.

▶ **Lemma 10.** $|S_2| \leq O(|OPT|\beta^{r-1} \log n)$

Proof of Lemma 10. During each iteration of the greedy step, let O be a minimum-cardinality set for which $S_2 \cup O$ is (r-round) contagious for Residual(S_1). We let $\vec{O} = o_1, o_2, o_3, \ldots, o_k$ be an arbitrary, but fixed, ordering over O. Were we to add the elements of $\vec{O}$, in order, to some initially empty set S_3, we would reduce the total hunger $h_r(S_2 \cup S_3)$ from $h_r(S_2)$ down to 0. Therefore, for some o^* in the sequence $\vec{O}$, adding o^* to S_3 causes $h_r(S_2 \cup S_3)$ to decrease by at least $\Delta^* \equiv h_r(S_2)/k$.

We now argue that adding o^* directly to S_2 must also significantly reduce the total amount of hunger amongst (S_2, r)-hungry vertices. Denote the magnitude of this change in total hunger by δ^*, that is, $\delta^* = h_r(S_2) - h_r(S_2 \cup \{o^*\})$. In a t-round activation process, the amount of hunger that can be removed by adding some vertex u to the seed set is bounded above by the number of up-to-length-t simple paths coming out of u. The residual graph has degree bounded by β, and thus the number of such paths is bounded by $1 + \beta + (\beta - 1)^2 + \cdots + (\beta - 1)^t \leq 2\beta^t$.

The addition of o^* to S_3 activates at most δ^* neighbors, each of which may be seen as initiating an $(r - 1)$-round activation process. Therefore, by adding o^* to S_3 in the original ordering $\vec{O}$, $h_r(S_2 \cup S_3)$ drops by at most $\delta^* \cdot 2\beta^{r-1}$. By definition, $\Delta^* \leq 2\delta^*\beta^{r-1}$ and hence $\delta^* \geq \Delta^*/(2\beta^{r-1}) = h_r(S_2)/(2k\beta^{r-1})$.

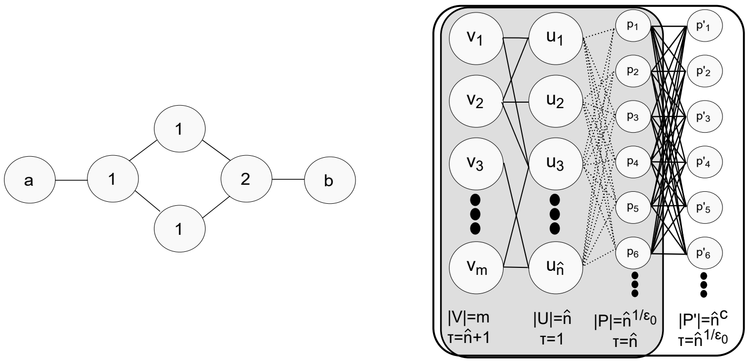

Figure 2 *Left:* This gadget, which has only undirected edges, simulates directed edge $a \to b$. The threshold of each gadget vertex is displayed. An active a "sends" one unit of activation to b, but an active b cannot "send" activation to a. *Right:* To create a hard undirected instance, the unshaded region augments the shaded region, the hard instance of directed MAX-TSS. In the shaded part, all drawn connections are oriented rightwards; in the unshaded part, solid lines represent undirected edges.

We cannot identify o^*: it depends both on the unknown optimal solution and some arbitrary ordering. However, we know that there exists *some* o^* for which

$$h_r(S_2 \cup \{o^*\}) = h_r(S_2) - \delta^* \leq \left(1 - 1/(2\beta^{r-1}k)\right) h_r(S_2)\,. \tag{1}$$

By iterating through each vertex, we can choose (in polynomial time) the vertex o^+ that minimizes $h_r(S_2 \cup \{o^+\})$: this latter expression is clearly also bounded by the right-hand side of inequality (1). Therefore, after $\Theta(k\beta^{r-1}\log n)$ iterations of greedily choosing such o^+, and adding it to S_2, the total hunger $h_r(S_2)$, in the residual graph, drops below 1 and so Algorithm 1 terminates. ◀

Proof of Theorem 8. Given Lemmas 9 and 10, all that remains is to find the optimal choice of β for Algorithm 1. If we knew $|\mathrm{OPT}|$, we would let β be $\Theta([n\tau_{\max}/(|\mathrm{OPT}|\log n)]^{1/r})$, and we would have $|S_1| + |S_2| = O((n\tau_{\max})^{1-1/r}|\mathrm{OPT}|^{1/r}\log^{1/r} n)$, so the ratio of $|S_1| + |S_2|$ to $|\mathrm{OPT}|$ would be

$$O((n\tau_{\max})^{1-1/r}\log^{1/r} n/|\mathrm{OPT}|^{1-1/r}) = O((\tau_{\max}/\tau_{\min})^{1-1/r} n^{1-1/r}\log^{1/r} n)\,, \tag{2}$$

where this "inequality" follows from the fact that $|\mathrm{OPT}| \geq \tau_{\min}$. Although we do not know $|\mathrm{OPT}|$, we can run Algorithm 1 with each β in $1, 2, \ldots, n$, and return the best solution; our approximation ratio will be at most the right-hand side of "inequality" (2). ◀

6 Hardness of Undirected Max-TSS

In this section, we show that the hardness of the undirected variant of MAX-TSS matches the $O(n^{1-\varepsilon})$-hardness of the directed variant studied by Kempe, Kleinberg, and Tardos [18].

▶ **Theorem 11.** *For every $\varepsilon > 0$, it is* **NP**-*hard to approximate the undirected version of* MAX-TSS *to within an $O(n^{1-\varepsilon})$ multiplicative factor.*

6.1 Revisiting the directed-edge construction

First, we recall the $O(n^{1-\varepsilon_0})$ hardness construction for the directed variant [18], as depicted in the shaded region of Figure 2. Given an instance of the **NP**-hard MAX COVERAGE problem with set system $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$, universe of $\hat{n}$ elements $\mathcal{X} = \bigcup_{i=1}^{m} S_i = \{x_1, x_2, \ldots, x_{\hat{n}}\}$ with $|\mathcal{X}| \geq |\mathcal{S}|$, budget k, and coverage goal g, construct a graph containing a vertex $v_i \in V$ for each set S_i, a vertex $u_j \in U$ for each element x_j, and a collection P of $\hat{n}^{1/\varepsilon_0}$ additional "padding" vertices $\{p_i\}$. Whenever $x_j \in S_i$, add a directed edge from v_i to u_j. Next, add a directed edge from each vertex in U to each vertex in P. Every vertex in V has threshold $m+1$, every vertex in U has threshold 1, and every vertex in P has threshold g.

Now, a seed set that has a vertex y in either U or P is weakly dominated by one that instead has a vertex $v \in V$ with a path to y. If no such v is available, choose an arbitrary unactivated element of V for the seed set. Therefore, consider only solutions in which the seed set S is a subset of V. Since the edges are directed, the vertices in U that are in the activation family are exactly those that have an in-edge from some vertex in the seed set $S \subset V$. Hence the vertex u_i becomes active if and only if the corresponding x_i is in some set S_j for which v_j is active. Thus, it is possible to activate $\geq g$ vertices in U if and only if the Max Coverage instance is satisfiable.

Each $p \in P$ has threshold g and there is the full family of directed edges in $U \times P$. So, if at least g vertices of U become active at some stage, then all $\hat{n}^{1/\varepsilon_0}$ vertices in P become active. If not, then since $m \leq \hat{n}$, no more than $O(\hat{n})$ vertices in the construction become active. Therefore, an $O(\hat{n}^{1/\varepsilon_0}/\hat{n}) = O(\hat{n}^{1/\varepsilon_0 - 1})$-approximation algorithm to Max-TSS can distinguish these two cases, and thus solve the initial Max Coverage instance. As the size of the instance is $n = O(\hat{n}^{1/\varepsilon_0})$, this translates to an $O(n^{1-\varepsilon_0})$ hardness result.

6.2 Translating to undirected edges

Unfortunately, naively replacing the construction's directed edges with undirected edges fails. A single active vertex in P would then activate all of U in a single time step. Instead, replacing each directed edge with the "directed-edge gadget", shown at the left of Figure 2, simulates the previous activation process using only undirected edges. However, these gadgets introduce $O(\hat{n} \cdot \hat{n}^{1/\varepsilon_0})$ extra vertices. To account for the larger problem size, we add a second padding set P' of size $\hat{n}^c$, for some $c \gg 1/\varepsilon_0$ to be chosen later. Each vertex in P', the unshaded region in the right of Figure 2, has threshold $\hat{n}^{1/\varepsilon_0}$.

Finally, we add an undirected edge for each $(p, p') \in P \times P'$. Due to their high thresholds, no vertex in P' will become active before all P are active, so these undirected edges in $P \times P'$ are "safe". As before, including a vertex outside V in the seed set S is dominated by activating one in V, so the analysis in Section 6.1 translates to the construction with set P'. Hence it is **NP**-hard to distinguish $|\mathcal{A}_\infty(S)| = \Omega(\hat{n}^c)$ from $|\mathcal{A}_\infty(S)| = O(\hat{n}^{1+1/\varepsilon_0})$, with the potential activation of gadget vertices being the dominating term in the latter number. As $n = \Theta(\hat{n}^c)$, this translates to an inapproximability factor of $O(n^{1-(1+1/\varepsilon_0)/c})$. For every $\varepsilon > 0$, choosing $c \geq (1 + 1/\varepsilon_0)/\varepsilon$ gives a hardness result of $O(n^{1-\varepsilon})$.

7 Application to Minimum Monotone Satisfying Assignment

Intuitively, the MMSA class of problems asks for the smallest *minterm* of a monotone Boolean function f, i.e., the Boolean vector $\vec{x}$ with $f(\vec{x}) = 1$ that minimizes $|\vec{x}|_1$. However, there is no single agreed definition of MMSA: both the exact description of function f and the measure of quality of a candidate solution $\vec{x}$ have been defined in several ways. The input f has varyingly been given as a monotone formula [17, 13, 8], a monotone circuit [1], and as a circuit that evaluates a monotone function (though it may use $\neg$ gates internally) [21]. The quality of an approximate solution (the "n" in the approximation factor) has also been measured either in terms of the formula/circuit size or in terms of the length of the input vector, in which case the circuit size is assumed to be poly(n). To standardize terminology, we rename MMSA by applying the prefix "MF", "MC", or "NMC" when f is a monotone formula, monotone circuit, or nonmonotone circuit, respectively, and add the superscript "f" or "x" to denote whether it is the description of f or the length of x, respectively, that determines n (in other contexts, previously, subscripts have denoted bounded-depth formulas and circuits).

While each of these models has been shown to give a LabelCover-like hardness of $2^{\log^{1-\varepsilon} n}$, only Umans [21] manages to establish a stronger hardness for the nonmonotone form, namely $n^{1-\varepsilon}$-hardness for both NMC-MMSA^f and NMC-MMSA^x. For MC-MMSA^x, the strongest known hardness results (to our knowledge) conditioned on an established conjecture are only implicitly described by Chlamtac, Dinitz, and Krauthgamer [9]; these lead to a hardness of $n^{3-2\sqrt{2}} \approx n^{.172}$ for the Smallest m-Edge Subgraph problem, assuming some slight modification of Conjecture 1. Despite the weak lower bounds in the case where f is given as a monotone formula/circuit, there is no MMSA problem for which we are aware of a sublinear factor approximation algorithm. A related observation has been previously made by Coja-Oghlan et al. in the context of Target Set Selection [12].

Here, we discuss the scenario in which f is provided as a monotone circuit, and n is either the length of the input or the size of the circuit description. We prove the following conditional results:

▶ **Theorem 12.** *Assuming Conjecture 1, for every $\varepsilon > 0$, it is hard to approximate* MC-MMSA^x *to within an $O(n^{1/2-\varepsilon})$ factor. .*

▶ **Theorem 13.** *Assuming Conjecture 1, for every $\varepsilon > 0$, it is hard to approximate* MC-MMSA^f *to within an $O(n^{1/3-\varepsilon})$.*

We begin with the proof of Theorem 12; the proof of Theorem 13 follows naturally.

Proof of Theorem 12. By the construction in Theorem 3, it is hard (assuming PDS) to approximate constant-round, constant-threshold, degree-$O(\sqrt{n})$ versions of TSS to within a factor of $n^{1/2-\varepsilon}$. Therefore, it suffices to show that we can transform these instances into monotone circuits with size polynomial in n. The reduction is quite simple: for each round $t = 1, 2, \ldots, r$ we have n gates, one for each vertex. The gate corresponding to vertex v is the output of a monotone circuit evaluating the threshold function $\mathrm{Th}_\tau^d(N_{t-1}(v))$, where τ is the threshold of v, d is its degree, and $N_{t-1}(v)$ comprises the gates corresponding to the previous timestep's gates that are in-neighbors of v (for the first timestep, these are just the input gates). Finally, the gates corresponding to the last round all feed into an $\wedge$ gate, forming our output.

The correctness of this circuit is easy to verify. It simulates running the activation process for all r rounds, and checking whether all vertices are active by the end of round r. As the in-degree of each vertex in the RTSS$_r$ instance is $\tilde{O}(\sqrt{n})$, the monotone circuit construction of Friedman [15] requires at most $O(\tau^{12.6}\sqrt{n}\log\sqrt{n}) = O(\sqrt{n}\log n)$ gates. Since we have rn of these circuits, the total number of gates is $O(rn\sqrt{n}\log n) = \mathrm{poly}(n)$. ◀

Proof of Theorem 13. The construction used in the proof of Theorem 12 provides a circuit with $O(m^{3/2}\log m)$ gates for which it is hard to approximate the optimal value to within $m^{1/2-\varepsilon'}$ for every $\varepsilon' > 0$. Choosing n such that $n = m^{3/2}\log m$, for every $\varepsilon > 0$ we have a circuit with n gates whose hardness is $\tilde{O}(n^{(1/2-\varepsilon')/(3/2)}) = O(n^{1/3-\varepsilon})$. ◀

8 Lower Bounds for k-Contagious Set

When a MIN-TSS instance contains a contagious set of size k, brute force search (plus simulation of the activation process) can identify such a set in $O(mn^k)$ time, where m is the number of edges in the graph. In this section, we show that improving on this naive approach on directed graphs by even slightly super-linear factors is difficult. Namely, using a construction similar to that in Theorem 6, we show that for $k \geq 3$ an $O(n^{k-\varepsilon})$-time algorithm for k-CONTAGIOUS SET implies an $O(2^{(1-\varepsilon')n})$-time algorithm for CNF-SAT, violating the STRONG EXPONENTIAL TIME HYPOTHESIS (SETH).

▶ **Theorem 14.** *For $k \geq 3$, there is no $O(n^{k-\varepsilon})$-time algorithm for k-*CONTAGIOUS SET *unless* SETH *is false.*

Proof. We reduce from the k-DOMINATING SET problem, which has been shown not to admit an $O(n^{k-\varepsilon})$-time algorithm unless SETH is false [19]. Suppose we are given an instance graph $G = (V, E)$ of k-DOMINATING SET, from which we derive an instance G' of k-CONTAGIOUS SET. Graph G' contains two vertices, a_v and b_v, for each vertex $v \in V$. A directed edge exists from a_u to b_v whenever either $u = v$ or when v is a neighbor of u in G. Additionally, a directed edge exists from b_u to a_v for every pair u and v, regardless of their adjacency in G. Finally, for all $v \in V$, we set $\tau(a_v) = n$ and $\tau(b_v) = 1$.

It is easy to verify that a size-k dominating set S in G corresponds to a size-k contagious set in G'. The seed set $\{a_v \mid v \in S\}$ activates all of the b vertices, which in turn activates the rest of the a vertices. Conversely, a size-k contagious set S' in G' can be converted into a size-k dominating set for G: $\{v \in V \mid a_v \in S' \text{ or } b_v \in S'\}$. Without loss of generality, we can assume that $S' \subset A$. The choice of a b_v in seed set S' is dominated by the selection of a_v, and no vertex in $A \setminus S'$ could become active unless S' activates all of $\{b\}$ in one round. Hence S' corresponds to a dominating set in G.

Thus, in $O(n^2)$ time, we can reduce finding a k-DOMINATING SET in G to finding a k-CONTAGIOUS SET in some G' that has $O(n)$ vertices. Since $k > 2$, it follows that a $O(n^{k-\varepsilon})$-time algorithm for k-CONTAGIOUS SET on G' implies the existence of such an algorithm for k-DOMINATING SET, violating SETH. ◀

Acknowledgements. The authors would like to thank Andrew Chester for many fruitful discussions, especially those relating to Section 5.

References

1 Michael Alekhnovich, Sam Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is **NP**-hard to linearly approximate. *The Journal of Symbolic Logic*, 66(01):171–191, 2001.

2 Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 171–180. ACM, 2010.

3 Sanjeev Arora, Boaz Barak, Markus Brunnermeier, and Rong Ge. Computational complexity and information asymmetry in financial products. In *Proceedings of the Innovations in (Theoretical) Computer Science Conference (ICS)*, pages 49–65, 2010.

4 Pranjal Awasthi, Moses Charikar, Kevin A. Lai, and Andrej Risteski. Label optimal regret bounds for online local learning. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, pages 150–166, 2015.

5 Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. Treewidth governs the complexity of target set selection. *Discrete Optimization*, 8(1):87–96, 2011.

6 Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 201–210. ACM, 2010.

7 Ning Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009.

8 Ramkumar Chinchani, Duc Ha, Anusha Iyer, Hung Q Ngo, and Shambhu Upadhyaya. On the hardness of approximating the min-hack problem. *Journal of Combinatorial Optimization*, 9(3):295–311, 2005.

9 Eden Chlamtac, Michael Dinitz, and Robert Krauthgamer. Everywhere-sparse spanners via dense subgraphs. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 758–767. IEEE, 2012.

10 Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, Martin Milanič, Joseph Peters, and Ugo Vaccaro. Spread of influence in weighted networks under time and budget constraints. *Theoretical Computer Science*, 586:40–58, 2015.

11 Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, Martin Milanič, and Ugo Vaccaro. Latency-bounded target set selection in social networks. *Theoretical Computer Science*, 535:1–15, 2014.

12 Amin Coja-Oghlan, Uriel Feige, Michael Krivelevich, and Daniel Reichman. Contagious sets in expanders. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1953–1987. SIAM, 2015.

13 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Information Processing Letters*, 89(5):247–254, 2004.

14 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*, pages 57–66. ACM, 2001.

15 Joel Friedman. Constructing $O(n \log n)$ size monotone formulae for the kth threshold function of n Boolean variables. *SIAM Journal on Computing*, 15(3):641–654, 1986.

16 Oded Goldreich and Shafi Goldwasser. On the possibility of basing cryptography on the assumption that $\mathbf{P} \neq \mathbf{NP}$., 1998.

17 Michael Goldwasser and Rajeev Motwani. *Intractability of assembly sequencing: Unit disks in the plane.* Springer, 1997.

18 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*, pages 137–146. ACM, 2003.

19 Mihai Pătraşcu and Ryan Williams. On the possibility of faster sat algorithms. In *Proceedings of the 21st Annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pages 1065–1075. Society for Industrial and Applied Mathematics, 2010.

20 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*, pages 61–70. ACM, 2002.

21 Christopher Umans. Hardness of approximating Σ_2^p minimization problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 465–474. IEEE, 1999.