On DkS-hardness for MinRep-hard problems

Moses Charikar * Yonatan Naamad † Anthony Wirth ‡

Abstract

In this paper, we explore the connection between the DENSEST k-SUBGRAPH (DkS) problem and MINREP. Both have been widely conjectured to be hard to approximate to within a sub-polynomial factor and these assumptions have been used as a basis to show hardness for a variety of problems. However, establishing such hardness for the DENSEST k-SUBGRAPH problem and MINREP seems out of reach of current techniques, even assuming other strong complexity conjectures, such as ETH. Additionally, it is not known even if either of these conjectures implies the other. We show that a generic technique can transform many proofs of MINREP-hardness into proofs of DkS hardness, up to polynomial factors. Additionally, we show that MINREP is (up to polynomial factors) at least as hard to approximate as DENSEST k-SUBGRAPH with Perfect Completeness. As a corollary of recent work by Manurangsi, this establishes an $n^{1/\text{poly}\log\log n}$ hardness for MINREP, assuming ETH.

1 Introduction

Understanding the approximation complexity of optimization problems is an important goal in theoretical computer science. The challenge here is to establish hardness results for optimization problems (assuming $P \neq NP$ and close variants) that match the performance of the best known approximation algorithms. While there have been many successes in this realm, obtaining tight upper and lower bounds for several problems of interest has proved to be elusive. In order to better map the complexity landscape and understand the relationships between seemingly hard optimization problems, researchers have introduced additional hardness assumptions beyond $P \neq NP$, e.g., Khot's Unique Games Conjecture [27].

In this paper, we explore the connection between two optimization problems: DENSEST k-SUBGRAPH (DkS) and MINREP. Both have embarrassingly large gaps between upper and lower bounds. Both have been widely conjectured to be hard to approximate to within a sub-polynomial factor; these assumptions have been used as a basis to show hardness for a variety of optimization problems of interest. However, establishing such hardness for DkS and MINREP seems out of reach of current techniques, even assuming other strong complexity conjectures such as the Exponential-Time Hypothesis (ETH) [25]. Presently, it is not known even if either of these conjectures implies the other.

The starting point for our work is attempting to explain the intriguing observation that there seems to be a connection between the two: *In many instances in the literature,* MINREP-*hard problems appear to be* DkS-*hard*. Examples include the minimum rainbow subgraph problem studied by Tirodkar and Vishwanathan [38], and k-steiner forest [24] (for which MINREP-hardness was established by Antonakopoulos and Kortsarz [3]).

*Stanford University, Stanford, CA, USA. Email: moses@cs.stanford.edu

†Princeton University, Princeton, NJ, USA. Email: ynaamad@cs.princeton.edu

‡School of Computing and Information Systems, The University of Melbourne, Parkville, Vic, Australia. Email: awirth@unimelb.edu.au

Our Results

In Section 3, we show that a generic technique can transform many proofs of MINREP-hardness into proofs of DkS hardness, up to polynomial factors. We introduce the k-MINREP problem, a natural generalization of MINREP, and establish hardness of this problem based on DkS. Further, we show that many problems previously shown to be MINREP-hard are, in fact, k-MINREP-hard. This immediately implies DkS-hardness for several problems such as, TARGET SET SELECTION (TSS), MONOTONE MINIMUM (CIRCUIT) SATISFYING ASSIGNMENT (MMSA/ MMCSA) and DEPENDENCY MIN MIDDLEBOX NODE PURCHASE (DMMNP), a problem that arises in middlebox optimization in software defined networking.

We then show, in Section 4, that MINREP is (up to polynomial factors) at least as hard to approximate as Densest k-Subgraph with Perfect Completeness (DkS$_C$). As a corollary of a recent breakthrough by Manurangsi [32], this establishes an $n^{1/\mathrm{poly}\log\log n}$ hardness for MINREP, assuming ETH. This reduction immediately implies $n^{1/\mathrm{poly}\log\log n}$ ETH hardness for any MINREP-hard problem, improving on the previous bound of $2^{\log^{1-\varepsilon} n}$.

Related Work

Kortsarz introduced MINREP (a variant of LABELCOVER, both defined fully in Section 2) as a useful problem to reason about the hardness of approximating k-spanners [29].

The best approximation for DkS obtains an $O(n^{1/4+\varepsilon})$-approximation in $n^{O(1/\varepsilon)}$ time [8]. The problem has been conjectured to be hard to within polynomial factors by several authors [4, 6, 16, 17]. Feige [22] showed constant-factor hardness for the problem, assuming that random 3-SAT instances are hard to refute. Khot [28] showed constant-factor hardness, assuming that NP is not contained in subexponential time. Alon et al. [2] ruled out any constant-factor approximation, assuming that random k-AND formulas are hard to refute. Raghavendra and Steurer [36] ruled out any constant-factor approximation, based on a strengthened version of the Unique Games conjecture. Braverman et al. [10] ruled out any constant-factor approximation by an $n^{\tilde{O}(\log n)}$ time algorithm, assuming ETH. A remarkable recent result of Manurangsi [32] established an $n^{-1/\mathrm{poly}\log\log n}$ hardness for DkS$_C$, assuming ETH.

In fact, MAXREP, a problem closely related to MINREP, has previously been shown to be DkS-hard [11].

2 Preliminaries

The results in this paper rely on careful consideration of several labelling/covering problems, designed to model Probabilistically Checkable Proofs and Interactive Proof systems [18]. A family of related problems, LABELCOVER, MAXREP, and MINREP, were introduced to explain the hardness of several other optimization problems, initially those on lattices [5], later on graph k-spanners [29]. In their categorization of problems by hardness of approximation [7], Arora and Lund refer to LABELCOVER-hard problems as Category III: hard to approximate within $O(2^{\log^{1-\gamma} n})$, for all fixed $\gamma > 0$, unless NP problems can be solved in quasi-polynomial time ($2^{\mathrm{poly}\log n}$ time).

Subsequently, the list of problems that have been proven LABELCOVER-hard is vast. Although Dinur and Safra showed how to obtain *NP-hardness* for a variant of LABELCOVER within a factor of $2^{(\log n)^{1-1/(\log\log n)^c}}$, no version of LABELCOVER has to date been proven inapproximable to within a polynomial, $1/n^{\varepsilon}$, factor. The belief that LABELCOVER admits polynomial hardness is known as the PROJECTION GAMES CONJECTURE [33].

2.1 Min and Max Rep

To define MINREP and MAXREP, we follow the style of Charikar, Hajiaghayi, and Karloff [11]. For either problem, we are given as input a bipartite graph $G = (A, B; E)$, with A and B partitioned into equal-sized sets $(A_1, A_2, \ldots, A_m)$ and $(B_1, B_2, \ldots, B_m)$, respectively. We regard the sets A_i the sets B_j as *supervertices*, and we say that the pair A_i and B_j induce a *superedge* if and only if there exists an $a_{i'} \in A_i$ and $b_{j'} \in B_j$ such that $(a_{i'}, b_{j'}) \in E$. Thus, the family of superedges partition E.

The MINREP question asks: "What is the minimum-size set of vertices for which the induced bipartite subgraph contains at least one edge for every superedge in the graph?" That is, what is the minimum number of vertices so that every superedge is *covered*?

In contrast, the MAXREP question asks "If exactly one vertex is chosen from each supervertex, what is the maximum number of superedges that can be covered?" Charikar et al. observe that without the restriction of one vertex per set, MAXREP is essentially DENSEST-2κ SUBGRAPH with κ vertices on each side of the bipartition.

2.2 Label Cover

In the original LABELCOVER problem [5], there was an antisymmetry between A and B. Specifically, in LABELCOVER$_{\min}$, the superedge (A_i, B_j) is "covered" if and only if *every* selected $b_{j'} \in B_j$ has some adjacent selected $a_{i'} \in A_i$. Recall that in MINREP, we only needed one matched pair to be selected to cover the superedge.

For the maximization version of the original LABELCOVER problem, this is no variation, as there is only one $a_{i'}$ and one $b_{j'}$ chosen from each supervertex anyway. Dinur and Safra [19] point out that the original LABELCOVER problem of Arora et al. [5] had the additional restriction that each vertex in A has degree one, and the "supervertices" all have the same "superdegree". Without this restriction, they state that it is unclear whether reductions to certain "lattice" problems follow.

Kortsarz identifies an $O(n^{\text{poly}\log n})$-time reduction from SATISFIABILITY in which `NO` instances cover only a fraction $2^{-\log^{1-\gamma} n}$ superedges of a MAXREP instance, while `YES` instances cover them all [29]. He thence provides a reduction to MINREP with essentially the same approximation hardness.

Peleg [35] and Elkin and Peleg [21] introduced $O(\sqrt{n})$ approximation algorithms for MAXREP and MINREP. These were thought perhaps to be optimal until Charikar et al. introduced factor-$O(n^{1/3})$ and $\tilde{O}(n^{1/3})$ algorithms, respectively. In contrast, the problem of interest in this paper, DkS (defined below), admits a factor-$O(n^{1/4+\varepsilon})$ algorithm, running in $O(n^{1/\varepsilon})$ time.

2.3 DkS Hardness

The DENSEST k-SUBGRAPH problem (DkS) asks: given a graph G and integer k, find a subset $S \subseteq V(G)$ of k vertices, S, such that $|E(G) \cap S \times S|$ is maximized. This problem is conjectured to be hard to approximate to some polynomial factor [6, 9, 30].

We also consider an important special case of DkS, DENSEST k-SUBGRAPH WITH PERFECT COMPLETENESS, which we denote by DkS$_C$. Here, `YES`-instance graphs are promised to contain a k-clique. The gap version of DkS$_C$, in which `NO`-instances are promised to contain no k-vertex subgraph with more than $\beta\binom{k}{2}$ edges, is denoted DkS$_C[1, \beta]$. Manurangsi [31] showed that this problem admits no polynomial time algorithm, even with $\beta = O(n^{-1/\text{poly}\log\log n})$, unless NP $\subseteq$ **DTIME** $\left(2^{\tilde{O}(n^{3/4})}\right)$. When $\beta = \Omega(n^{\varepsilon})$, Bhaskara et. al. (implicitly) provide an algorithm for DkS$_C[1, \beta]$ with running time $O(n^{1/\varepsilon})$ [9].

The SMALLEST k-EDGE SUBGRAPH problem (SkES), first explicitly defined by Chlamtac, Dinitz, and Krauthgamer [16], is as follows. Given a graph G and integer k, determine the smallest r for which there exists a subset $R \subseteq V(G)$ of r vertices whose induced subgraph contains at least k edges. As Chlamtac et al.

point out, Nutov's results on the hardness of approximating Node-Weighted Steiner Network [34] imply that if DkS is $\alpha(n)$-hard to approximate, then SkES is $O(\sqrt{\alpha(n)})$ hard.

2.4 Problems Discussed

We now describe the various problems whose DkS-hardness will be shown in Section 3.

TARGET SET SELECTION (TSS) Motivated by work of Domingos and Richardson [20, 37], Kempe, Kleinberg, and Tardos [26] studied the following model of influence propagation in social networks. Chen showed this problem to be MINREP-hard to approximate [14]; in previous work, we showed that it is somewhat harder to approximate under a conjecture regarding the *average-case* hardness of DkS [13].

The input is a graph $G = (V, E)$ along with a threshold function $\tau : V \to \mathbb{Z}^+$. Vertices in this graph can be either *active* or *inactive*. Given a *seed set* of initially activated vertices, influence proceeds in a sequence of rounds. A previously inactive vertex v becomes active in a particular round whenever at least $\tau(v)$ of its neighbors were active in the previous round. Once a vertex becomes active, it remains active in all subsequent rounds. The objective is to find the smallest-cardinality seed set such that every vertex in V eventually activates. (For the process to continue, there must be at least one new active vertex in each round, so there are at most $n-1$ rounds.)

MONOTONE MINIMUM (CIRCUIT) SATISFYING ASSIGNMENT (MMSA/ MMCSA) The MONOTONE MINIMUM SATISFYING ASSIGNMENT class of problems was first introduced by Alekhnovich et al. [1] and Goldwasser and Motwani [23]. Inputs to these problems consist of either a monotone formula (in MMSA) or a monotone circuit (in MMCSA) on a set $\{x_i\}$ of inputs[1]. The goal is to find the smallest *minterm* of the formula or circuit, i.e., the smallest collection of inputs such that the formula/circuit evaluates to `True` even when when all other inputs are set to `False`. It was previously shown that these problems are LABELCOVER$_{\text{min}}$-hard to approximate even for bounded-depth circuits [1, 23]. The hardness of these problems has, in turn, been used as a starting point for other lower bounds [15]. As MMSA is a special case of MMCSA, a hardness result shown for MMSA translates to equivalent hardness for MMCSA.

DEPENDENCY MIN MIDDLEBOX NODE PURCHASE (DMMNP) In the DEPENDENCY MIN MIDDLEBOX NODE PURCHASE problem, first introduced by Charikar et al [12], we are asked to distribute sufficient processing power in a network so that packets sent between terminals can each be processed along the way. The problem is formally described as follows. We are given an edge-capacitated graph, as well as a nonnegative price p_i and nonnegative potential capacity c_i (which might be ∞) for each vertex v_i. At our discretion, we may spend p_i to purchase exactly c_i units of processing capacity at node v_i. Otherwise, v_i has 0 processing capacity.

Additionally, we are given a set of *demands*, each of which consists of a source s_i, a sink t_i, and an intermediate sequence of processing-vertex subsets $A_i^1, A_i^2, \ldots$. These denote which vertices are allowed to execute each of a sequence of processing tasks on the packets as they are routed from s_i to t_i.

For each demand d_i, we must route one unit of flow along a walk from its source to some vertex in A_i^1 (which we call its first stop), then to some vertex in A_i^2 (its second stop), and so on, and finally route it to its sink, t_i. Each time the flow stops at a node, it consumes one unit of that node's processing power.

A flow is feasible if, aggregated over all walks, (i) no edge is traversed more times than its capacity and (ii) no vertex is chosen as a stopping vertex more times than its purchased *processing capacity*. The goal is

[1] Monotone here means that the formula or circuit is computed using only a polynomial number of $\wedge$ and $\vee$ gates. In constrast, Umans [39] shows that the circuit problem is $n^{1-\varepsilon}$-hard to approximate (unless NP $\subseteq$ QuasiP) when the circuit can use also use $\neg$ gates internally, despite computing a monotone function.

then to purchase a minimal-spend subset of nodes at which to place processing capacity so that there exists a feasible flow.

2.5 Notation

Many of the reductions in this paper only show that two problems are *polynomially related*. In particular, they show that if PROBLEM A has an $f(n)$-approximation algorithm, then PROBLEM B admits an $O(f(n)^c)$ approximation, for some constant c, (or, conversely, $g(n)$-hardness for PROBLEM B implies $O(g(n)^{1/c})$-hardness for PROBLEM A). Thus, if PROBLEM B is assumed (or known) to have no approximation algorithm with a sub-polynomial guarantee, then the same must hold for PROBLEM A. We denote such a relationship PROBLEM B $\ll_p$ PROBLEM A. For example, the last paragraph of Section 2.3 implies that DkS $\ll_p$ SkES.

3 DkS hardness for MinRep-hard Problems

The starting point for all of our reductions will be the k-MINREP problem, a natural generalization of MINREP (see Section 2.1). As in MINREP, we are given a bipartite graph $G = (A, B; E)$, with A and B partitioned into supervertices $\{A_i\}$s and $\{B_j\}$s, respectively. As an additional input to the problem, we are given an integer k. Using the same definition of a superedge as before, our goal is to find the smallest subset $S \subseteq V$ such that the edges induced by S belong to at least k different superedges. The decision version asks, for some integer ℓ, whether or not there exists a feasible solution of size no more than ℓ.

Indeed, the MINREP problem is simply k-MINREP with k equal the number of superedges in G; k-MINREP is hence MINREP-hard. As it turns out, the hardness of k-MINREP is also polynomially-related to that of DkS.

Theorem 3.1. DkS $\ll_p$ k-MINREP

Proof. As discussed in Section 2.3, DkS $\ll_p$ SkES. It thus suffices to show that SkES $\ll_p$ k-MINREP. We show this via a direct reduction: given an instance $(G(V, E); k)$ of SkES, we choose a random bipartition of V into A and B by placing each vertex in A independently with probability $1/2$, and in B otherwise. The k-MINREP instance returned comprises the bipartition (A, B), those edges in E that cross the bipartition, and one supervertex for each vertex in V.

For a subset $S \subseteq V$ of size k, let $v^1(S)$ be the objective value of S for the given SkES instance, and $v^2(S)$ be the objective value of S under the constructed k-MINREP instance. Clearly, $v^2(S) \leq v^1(S)$, as covering an edge coincides with covering a unique superedge. Additionally, by the random construction, $\mathbb{E}[v^2(S)] = v^1(S)/2$, and, by a standard application of the Chernoff bound, with high probability, $v_2(S) \geq v^1(S)/3$. Thus, the two problems differ only by a constant factor in their objective scores for every set S, meaning that the two problems are polynomially related.

□

We now proceed to show that various problems previously shown to be MINREP hard are, in fact, k-MINREP-hard, and thus (as a consequence of Theorem 3.1), have a hardness polynomially related to that of DkS.

3.1 TARGET SET SELECTION

In this section, we show how a simple modification to Chen's result [14] that MINREP $\ll_p$ TSS leads to k-MINREP-hardness for TSS.

Theorem 3.2. k-MINREP $\ll_p$ TARGET SET SELECTION

Proof. A slightly simplified presentation of Chen's result [14] is as follows. Given an instance $G = ((\mathcal{A},\mathcal{B});E)$ of MINREP, let S denote the set of superedges. We construct a graph with four layers:

1. A *vertex layer*, V_1, with one threshold-n vertex v_u^1 for each vertex $u \in A \cup B$.
2. An *edge layer*, V_2, with one threshold-2 vertex v_e^2 for each edge $e \in E$.
3. An *superedge layer*, V_3, with one threshold-1 vertex v_s^3 for each superedge $s \in S$.
4. A *final layer*, V_4, with one threshold-$|S|$ vertex v_u^4 for each vertex $u \in A \cup B$.

Using *directed-edge gadgets* (this gadget simulates a single directed edge with several undirected edges, as in [13]) we connect the layers as follows.

- Each vertex $v_u^1 \in V_1$ has a directed edge to every vertex $v_e^2 \in V_2$ such that edge e is incident on u.
- Each vertex $v_e^2 \in V_2$ has a directed edge to the vertex v_s^3 such that edge e is in superedge s.
- Each vertex $v_s^3 \in V_3$ has a directed edge to every vertex in V^4.
- Each vertex $v_u^4 \in V_4$ has a directed edge to every vertex in V^1.

If we activate exactly the vertices from V_1 corresponding to a feasible solution to the given MINREP instance, we easily verify that they activate all vertices in V_2 corresponding to edges in their induced subgraph, in turn activating all vertices in V_3 corresponding to covered superedges, and then activating all of V_4. Once all of V_4 is activated, the activation of all of V_1, V_2, and V_3 follows in subsequent rounds. He now proves the following:

Claim 3.3 (essentially from Chen [14])**.** *There exists a* 2*-approximation to the optimal solution that contains only vertices from* V_1*. Consequently,* MINREP $\ll_p$ TARGET SET SELECTION.

First, it can be shown without loss of generality that no vertices from the directed-edge gadgets are picked. Next, it never helpful to activate a proper subset of V_4: only V_4 as a whole can activate other vertices, and the graph structure insures that the spreading process influences all vertices of V_4 identically. Next, with threshold 1 the choice of a vertex from V^3 is dominated by the choice of one of its in-neighbors in V^2. Finally, the choice of a vertex in V^2 is in turn dominated (up to a factor two) by choosing both of its in-neighbors in V^1, proving the first part of the claim. Since the graph constructed contains polynomially many vertices, the second part of the claim follows.

To convert this into k-MINREP hardness, we can simply change the threshold of vertices in V^4 from $|S|$ to k. These final-layer vertices become active if and only if at least k superedges are covered, as opposed to all $|S|$ superedges, required for Chen's construction. Thus, with the analysis proceeding exactly as before, we can conclude that $k -$ MINREP $\ll_p$ TSS. □

Combining this with Theorem 3.1, DkS hardness for TSS follows immediately.

Corollary 3.4. DkS $\ll_p$ k-MINREP $\ll_p$ TARGET SET SELECTION.

3.2 Monotone Minimum (Circuit) Satisfying Assignment

We now show how to convert previous proofs that $\text{LABELCOVER}_{\min} \ll_p$ MMSA into k-MINREP (and, thus, DkS) hardness for MMSA.

Theorem 3.5. k-MINREP $\ll_p$ MMSA $\ll_p$ MMCSA

Proof. Previously, it was proven that MMSA is polynomially related to $\text{LABELCOVER}_{\min}$ [1, 23]. Although $\text{LABELCOVER}_{\min}$ is slightly different from MINREP, these previous approaches can be directly adapted to MINREP. Given a MINREP instance $G((A,B);E)$, the input layer to the circuit consists of vertices in $A \cup B$. For each edge $e \in E$, we add one $\wedge$ gate, $\wedge_e$, that takes as input the two inputs corresponding to its endpoints. For each superedge s, we add an $\vee$ gate, $\vee_s$, that takes as input all $\wedge_e$ gates for which e is part of superedge s. Finally, we add one more $\wedge$ gate $\wedge_{out}$ whose input comprises every $\vee_s$ gate.

This circuit evaluates to `True` if and only if the `True` inputs correspond to a set of vertices that collectively cover all superedges. Thus, the optimal objective value for this problem is exactly that of the original MINREP instance. Because the number of gates produced is $\text{poly}(n)$, the hardness of MMSA remains polynomially related to that of MINREP, regardless of whether we the "n" in an $f(n)$-approximation factor refers to the number of inputs or to the number of gates in the circuit.

To convert this construction into one for k-MINREP, we replace $\wedge_{out}$ with a monotone threshold-k circuit. Thus, this circuit simulates the requirement that only k of the superedges need to be covered. As such threshold circuits can be constructed with only polynomially many internal gates [40], we conclude that MMSA is polynomially related to k-MINREP. □

Combining this with Theorem 3.1, we obtain the following corollary.

Corollary 3.6. DkS $\ll_p$ k-MINREP $\ll_p$ MMSA $\ll_p$ MMCSA .

3.3 Middlebox Allocation

We now modify the result of Charikar et al. [12], translating its MINREP-hardness for DEPENDENCY MIN MIDDLEBOX NODE PURCHASE into k-MINREP hardness.

Their reduction from MINREP proceeds as follows. For each supervertex in $\mathcal{A}$ or $\mathcal{B}$, the graph has one corresponding node (a_i or b_i, respectively) as well as one node v_j for each of the (constituent) vertices. Capacity-1 arcs are added from each a_i (supervertex) node to the $\{v_k\}$s representing nodes in A_i, as well as from each $\{v_k\} \in B$ to the b_i corresponding to its containing supervertex. Purchasing (infinite) processing capacity is free at the $\{a_i\}$s and $\{b_k\}$s, but comes at a cost of 1 at the $\{v_k\}$s.

Finally, we add two vertices s and t – the first with an infinite-capacity arc to each a_i and the second with an infinite-capacity arc from every b_k. For each superedge (A_i, B_k), we add one unit of demand from s to t specifying that the flow must be processed at both a_i and b_k. It is not hard to verify that the problem of placing processing at the v_i nodes exactly corresponds to selecting vertices in the source MINREP instance, thus establishing the hardness.

To convert this into hardness for k-MINREP, we add two additional nodes, q_1 and q_2. There is an unbounded-capacity arc from s to q_1 and an unbounded-capacity arc from q_2 to t, and q_1 is linked to q_2 with an arc of capacity $(|\mathcal{S}| - k)$. Unlimited processing capacity can be purchased at either q_1 or q_2 for free. We now modify the demands so that each unit of flow can be processed either by its original required pair (a_i and b_j) or by q_1 and q_2.

This construction allows for $|\mathcal{S}| - k$ units of the demand to get processed for free via the path $s \to q_1 \to q_2 \to t$, meaning that only k units of the demand must be processed as the original construction intended. Thus, the problem is now equivalent to that of solving k-MINREP, giving the desired hardness.

Theorem 3.7. DkS $\ll_p$ k-MINREP $\ll_p$ DMMNP .

4 $\mathrm{D}k\mathrm{S}_C$ hardness for MINREP

We initiate this section with a lemma that will later be helpful in showing that $\mathrm{D}k\mathrm{S}_C \ll_p$ MINREP.

Lemma 4.1. *Let G be a graph in which every k-vertex subgraph has at most s induced edges. Every subgraph of $\mathcal{G}$ containing at least $t \geq s$ induced edges must have more than $(k-1)\sqrt{t/s}$ vertices.*

Proof. Let $\mathcal{R}$ be an r-vertex subgraph of G containing at least t induced edges. Clearly, $r \geq k$. Now let $\mathcal{K}$ be a k-vertex induced subgraph of $\mathcal{R}$ chosen by selecting a uniformly random sample of k vertices from $\mathcal{R}$. For every edge $(u,v) \in \mathcal{R}$, the probability that it also appears in $\mathcal{K}$ is the probability that both of its endpoints survived the sampling process. Namely,

$$\Pr[(u,v) \in \mathcal{K}] = \Pr[u \in \mathcal{K} \wedge v \in \mathcal{K}] = \Pr[u \in \mathcal{K}] \cdot \Pr[v \in \mathcal{K} \mid u \in \mathcal{K}] = \tfrac{k}{r} \cdot \tfrac{k-1}{r-1} > \tfrac{(k-1)^2}{r^2}.$$

By linearity of expectations, $\mathbb{E}[\#\text{ edges in } \mathcal{K}] > t(k-1)^2/r^2$. However, since a k-vertex subgraph of $\mathcal{R}$ has at most s induced edges, $s \geq \mathbb{E}[\#\text{ edges in } \mathcal{K}]$. Combining the two inequalities, $s > t(k-1)^2/r^2$, and, solving for r, we get $r > (k-1)\sqrt{t/s}$. □

With this lemma in hand, we can proceed to prove that MINREP's hardness is polynomially related to that of the parameterized version of $\mathrm{D}k\mathrm{S}_C$.

Theorem 4.2. *If there is an α-approximation algorithm for* MINREP *(with $\alpha > 1$), then there exists a randomized algorithm for $\mathrm{D}k\mathrm{S}_C[1,\beta]$ whenever $\beta < 1/(9\alpha^2)$. In particular, $\mathrm{D}k\mathrm{S}_C \ll_p$* MINREP.[2]

Proof. To simplify the presentation of the proof, we make the benign assumption that the number of vertices is evenly divisible by $6\log n$. Similar analysis carries through when this is not the case.

The construction here is similar to that of Theorem 6 from [11], but the analysis is different. Given an instance $(G(V,E),k)$ of $\mathrm{D}k\mathrm{S}_C[1,\beta]$, we partition V uniformly at random into $k' = k/(3\log n)$ groups so that each has size at least $(3n/k)\log n$.

We now construct a MINREP instance as follows:

- The first $k'/2$ groups become the *left* supervertices $A_1, A_2, \cdots, A_{k'/2}$, while the the remaining $k'/2$ groups become the *right* supervertices $B_1, B_2, \cdots, B_{k'/2}$.
- The edges in the construction are those edges of G that cross the bipartition, i.e., $\{(u,v) \in E : u \in \bigcup_i A_i \wedge v \in \bigcup_j B_j\}$.

If G is a `YES` instance of $\mathrm{D}k\mathrm{S}_C[1,\beta]$, then it contains a k-clique. The partitioning scheme above ensures that, with high probability, each supervertex contains at least 1 vertex of this clique. Thus, every (A_i, B_j) pair induces at least one superedge with high probability. Consequently, the optimal objective value is exactly k' with high probability, as picking one vertex in each partition is necessary to cover all superedges, but picking exactly one clique vertex from each partition is sufficient.

However, if G came from a `NO` instance, then one of two things can happen:

1. The constructed MINREP instance is missing some superedge (that is, some (A_i, B_j) supervertex pair does not induce a superedge); or
2. The constructed MINREP instance has all $k'^2/4$ potential superedges.

[2] To keep the analysis simpler, no attempt was made to lower the factor of 9 in the bound on β.

In the first case, we can immediately identify that with high probability the input graph came from a NO instance of DkS$_C$ and return accordingly. Thus, we focus our attention on the optimal objective values of NO instances in which all superedges are present.

Such an instance can be identified as a NO instance when its optimal objective value is greater than $\alpha k'$: Applying the factor-α approximation algorithm to MINREP would reveal that the optimal objective value exceeds k', hence, with high probability, that of a YES instance.

To figure out which values of β necessitate imply this $\alpha k'$ lower bound on OPT, we appeal to Lemma 4.1. A feasible solution to the problem must cover all $k'^2/4$ superedges, and thus the subgraph induced by the solution must contain at least $k'^2/4$ edges.

However, the promise problem we are solving, DkS$_C[1,\beta]$, guarantees that no k-vertex subgraph, $k = 3k'\log n$, contains more than $\beta\binom{k}{2} \le \beta k^2 = 9\beta k'^2 \log^2 n$ edges. Applying Lemma 4.1 with $t = k'^2/4$, $s = 9\beta k'^2 \log^2 n$, and $k = 3k'\log n$, we see that a subgraph inducing $k'^2/4$ edges (and, in particular, the smallest one) has at least

$$(3k'\log n - 1)\sqrt{\frac{k'^2/4}{9\beta k'^2 \log^2 n}} = \frac{3k'\log n - 1}{6\sqrt{\beta}\log n} > \frac{k'}{3\sqrt{\beta}}$$

vertices. Thus, we have the sought $\alpha k'$ lower bound on the optimal objective value whenever $\alpha < 1/(3\sqrt{\beta})$ or, equivalently, whenever $\beta < 1/(9\alpha^2)$. □

Combining Theorem 4.2 with the recent result of Manurangsi [31] that, assuming ETH, DkS$_C$ is hard to approximate to within an $O(n^{1/\text{poly}\log\log n})$ factor, we get the following result on the hardness of MINREP.

Theorem 4.3. *For some constant c,* MINREP *has no $O(n^{1/\log\log^c n})$-approximation algorithm unless NP* $\subseteq$ **DTIME** $\left(2^{\tilde{O}(n^{3/4})}\right)$.

References

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

[2] Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. Inapproximability of densest κ-subgraph from average case hardness. *Unpublished manuscript*, 1, 2011.

[3] S. Antonakopoulos and G. Kortsarz. Lower bounds for non-preemptive *k-dial a ride* and for the *k steiner forest* problem, August 2013. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.412.318.

[4] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180. ACM, 2010.

[5] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[6] Sanjeev Arora, Boaz Barak, Markus Brunnermeier, and Rong Ge. Computational complexity and information asymmetry in financial products. In *ICS*, pages 49–65, 2010.

[7] Sanjeev Arora and Carsten Lund. Hardness of approximations. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 10. PWS, 1997.

[8] 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 *STOC*, pages 201–210, 2010.

[9] Aditya Bhaskara, Moses Charikar, Aravindan Vijayaraghavan, Venkatesan Guruswami, and Yuan Zhou. Polynomial integrality gaps for strong SDP relaxations of densest k-subgraph. In *SODA*, pages 388–405, 2012.

[10] Mark Braverman, Young Kun Ko, Aviad Rubinstein, and Omri Weinstein. ETH hardness for densest-k-subgraph with perfect completeness. In *SODA*, pages 1326–1341, 2017.

[11] Moses Charikar, MohammadTaghi Hajiaghayi, and Howard Karloff. Improved approximation algorithms for label cover problems. *Algorithmica*, 61(1):190–206, 2011.

[12] Moses Charikar, Yonatan Naamad, Jennifer Rexford, and X. Kelvin Zou. Multi-commodity flow with in-network processing, submitted.

[13] Moses Charikar, Yonatan Naamad, and Anthony Wirth. On approximating target set selection. In *APPROX*, pages 4:1–4:16, 2016.

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

[15] 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.

[16] Eden Chlamtac, Michael Dinitz, and Robert Krauthgamer. Everywhere-sparse spanners via dense subgraphs. In *FOCS*, pages 758–767, 2012.

[17] Eden Chlamtáč, Michael Dinitz, and Yury Makarychev. Minimizing the union: Tight approximations for small set bipartite vertex expansion. In *SODA*, pages 881–899, 2017.

[18] Michael Dinitz, Guy Kortsarz, and Ran Raz. Label cover instances with large girth and the hardness of approximating basic k-spanner. *ACM Transactions on Algorithms (TALG)*, 12(2):25, 2016.

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

[20] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *KDD*, pages 57–66, 2001.

[21] Michael Elkin and David Peleg. The hardness of approximating spanner problems. *Theory of Computing Systems*, 41(4):691–729, 2007.

[22] Uriel Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543, 2002.

[23] Michael Goldwasser and Rajeev Motwani. Intractability of assembly sequencing: Unit disks in the plane. In *WADS*, pages 307–320, 1997.

[24] Mohammad Taghi Hajiaghayi and Kamal Jain. The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. In *SODA*, pages 631–640, 2006.

[25] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[26] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory OF Computing*, 11(4):105–147, 2015.

[27] Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC*, pages 767–775, 2002.

[28] Subhash Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

[29] Guy Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

[30] Guy Kortsarz, Vahab S. Mirrokni, Zeev Nutov, and Elena Tsanko. Approximating minimum-power degree and connectivity problems. *Algorithmica*, 60(4):735–742, 2011.

[31] Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. *arXiv preprint arXiv:1611.05991*, 2016.

[32] Pasin Manurangsi and Dana Moshkovitz. Improved approximation algorithms for projection games. *Algorithmica*, 77(2):555–594, 2017.

[33] Dana Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. In *APPROX*, pages 276–287, 2012.

[34] Zeev Nutov. Approximating steiner networks with node-weights. *SIAM Journal on Computing*, 39(7):3001–3022, 2010.

[35] David Peleg. Approximation algorithms for the Label-Cover$_{\text{MAX}}$ and Red-Blue Set Cover problems. *Journal of Discrete Algorithms*, 5(1):55–64, 2007.

[36] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *STOC*, pages 755–764, 2010.

[37] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[38] Sumedh Tirodkar and Sundar Vishwanathan. On the approximability of the minimum rainbow subgraph problem and other related problems. In *ISAAC*, pages 106–115, 2015.

[39] Christopher Umans. Hardness of approximating Σ_2^p minimization problems. In *FOCS*, pages 465–474, 1999.

[40] Leslie G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5(3):363–366, 1984.