Improved Approximation Algorithms for Label Cover Problems

Moses Charikar[1]*, MohammadTaghi Hajiaghayi[2], and Howard Karloff[2]

[1] Department of Computer Science, Princeton University,
Princeton, NJ 08540, USA, moses@cs.princeton.edu
[2] AT&T Labs — Research, 180 Park Ave.,
Florham Park, NJ 07932, USA, {hajiagha,howard}@research.att.com

Abstract In this paper we consider both the maximization variant Max Rep and the minimization variant Min Rep of the famous Label Cover problem, for which, till now, the best approximation ratios known were $O(\sqrt{n})$. In fact, several recent papers reduced Label Cover to other problems, arguing that if better approximation algorithms for their problems existed, then a $o(\sqrt{n})$-approximation algorithm for Label Cover would exist.
We show, in fact, that there are a $O(n^{1/3})$-approximation algorithm for Max Rep and a $O(n^{1/3}\log^{2/3} n)$-approximation algorithm for Min Rep. In addition, we also exhibit a randomized reduction from Densest k-Subgraph to Max Rep, showing that any approximation factor for Max Rep implies the same factor (up to a constant) for Densest k-Subgraph.

1 Introduction

Label Cover was first introduced in Arora et al. [2] and is a canonical problem used to show strong hardness results for many NP-hard problems [12]. It is known that for Label Cover, there is no approximation algorithm achieving a ratio $2^{\log^{1-\varepsilon} n}$, for any $0 < \varepsilon < 1$, unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\mathrm{polylog}(n)})$ [2,12]. Label Cover has both maximization and minimization variants for both of which the above hardness holds. Kortsarz [14] introduced slight variants of these two problems called Max Rep and Min Rep. (See the end of this section for formal definitions of both problems.) Indeed Max Rep is equivalent to the maximization version of Label Cover, but Min Rep is slightly different from the minimization version of Label Cover. Kortsarz [14] showed that for both Max Rep and Min Rep, there is the same hardness of $2^{\log^{1-\varepsilon} n}$, for $0 < \varepsilon < 1$, unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\mathrm{polylog}(n)})$. The simpler definitions of Max Rep and Min Rep make them particularly attractive for use in hardness reductions.

For the upper bound, it is known that both Max Rep and Min Rep admit relatively simple $O(\sqrt{n})$ approximation algorithms [6,16]. Recently some authors

* Supported by NSF ITR grant CCF-0426582, NSF CAREER award CCF-0237113, MSPA-MCS award 0528414, and NSF expeditions award 0832797.

suggested the possibility that $O(\sqrt{n})$ is the best approximation factor for these two problems. See, e.g., [8], in which the authors write, *"This ratio [$O(\sqrt{n})$] seems hard to improve and better ratio algorithms for LABEL-COVER$_{\max}$ are not known even for very simple versions of the problem (e.g., when the structure of the graph obeys the rules of the Unique Game Conjecture...). If LABEL-COVER$_{\max}$ is indeed $\Omega(\sqrt{n})$ hard to approximate, then so is DSF [Directed Steiner Forest].* Indeed several recent papers reduced Min Rep/Max Rep to other problems in order to obtain hardness results; therefore studying the approximability of Min Rep/Max Rep is an important goal. See [8] for Directed Steiner Forest, [16] for Red-Blue Set Cover, [4,11] for Set Cover with Pairs, [3] for Sparsest k-Transitive-Closure-Spanner, [10] for Min-Power k-Edge-Disjoint Paths, [1] for ℓ-Round Power Dominating Set, [5] for Target Set Selection, [15] for Vertex Connectivity Survivable Network Design, and [9] for Stochastic Steiner Tree with Non-uniform Inflation.

In this paper, we refute the possibility of $\Omega(\sqrt{n})$ hardness for both Max Rep and Min Rep by developing a $O(n^{1/3})$-approximation algorithm for Max Rep and a $O(n^{1/3}\log^{2/3} n)$-approximation algorithm for Min Rep. Our result for Min Rep (see Section 2) uses a natural LP relaxation for the problem. We round this LP based on an interesting generalization of the birthday paradox. Our result for Max Rep (see Section 3) uses a direct combinatorial approach. Indeed, we show that for Max Rep the integrality ratio for a natural LP relaxation is $\Omega(\frac{\sqrt{n}}{\ln n})$ (in contrast to Min Rep for which the integrality ratio is $\Omega(n^{1/3-\varepsilon})$, for all $\varepsilon > 0$.)

Our $O(n^{1/3})$- and $O(n^{1/3}\log^{2/3} n)$-approximation algorithms for MaxRep and MinRep might suggest a connection between these problems and the related well-studied problem Densest k-Subgraph, for which the best approximation factor so far is $O(n^{1/3-\delta})$ [7], for some small *fixed* $\delta > 0$. The current best inapproximability result only rules out a polynomial time approximation scheme (PTAS) under the assumption that $\mathbf{NP} \not\subseteq \mathbf{BPTIME}(2^{n^{\varepsilon}})$ [13]. We show indeed that there is a randomized reduction from Densest k-Subgraph to Max Rep, which preserves the approximation factor up to a constant factor (see Section 4).

We end this section with exact definitions of Max Rep and Min Rep.

Definition 1. **(Max Rep)**
Instance: *A bipartite graph $G = (A, B, E)$, where $|A| = |B| = n$, and an equitable partition $\mathcal{A}$ of A and $\mathcal{B}$ of B into k sets of same size $q = \frac{n}{k}$ each (assuming that $n \bmod k = 0$).*
Objective: *Choose $A' \subseteq A$ and $B' \subseteq B$ with $|A' \cap A_i| = |B' \cap B_j| = 1$ for each $i, j = 1, \ldots, k$ such that the subgraph induced by $A' \cup B'$ has the maximum number of edges.*

In Definition 1 the bipartite graph and the partition of A and B induce a "supergraph" $\mathcal{H}$ in the following way: The vertices of $\mathcal{H}$ are the sets A_i and B_j. Two sets A_i and B_j are adjacent by a "superedge" in $\mathcal{H}$ if and only if there exist $a_i \in A_i$ and $b_j \in B_j$ which are adjacent in G. In this case, we say pair

(a_i, b_j) *covers* the superedge (A_i, B_j). In the MAX REP problem the goal is to select one element, called a *representative*, from each A_i and each B_j such that the number of covered superedges in $\mathcal{H}$ is maximized. Another natural objective function considered in the literature is as follows:

Definition 2. *(*MIN REP**)**
Instance: *A bipartite graph $G = (A, B, E)$, where $|A| = |B| = n$, and equitable partitions $\mathcal{A}$ of A and $\mathcal{B}$ of B into k sets of same size $q = \frac{n}{k}$.*
Objective: *Choose $A' \subseteq A$ and $B' \subseteq B$ such that pairs (a, b), $a \in A'$ and $b \in B'$, cover all the superedges of $\mathcal{H}$, while minimizing $|A'| + |B'|$.*

2 $O(n^{1/3} \log^{2/3} n)$-Approximation Algorithm for MIN REP

There is a trivial k-approximation algorithm for MIN REP, namely, select both vertices of one edge corresponding to each superedge. (The optimum selects at least one vertex of each A_i (B_j) to which there is a superedge attached and we choose at most k since there are at most k superedges attached to each A_i (B_j).) In this section, we present a $O(\sqrt{q} \log k)$ approximation algorithm for the MIN REP problem using a natural LP relaxation and a rounding scheme whose analysis is based on a generalization of the birthday paradox. By using the better of these two algorithms, and remembering that $q = n/k$, we obtain an $O(n^{1/3} \log^{2/3} n)$-approximation algorithm.

First, we start with an LP relaxation as follows:

$$\text{OPT} = \text{minimize} \quad \sum_{u \in A} p_u + \sum_{v \in B} p_v \tag{1}$$

subject to

$$\sum_{u \in A_i, v \in B_j \text{ s.t. } (u,v) \in E(G)} f_{uv} = 1 \quad \forall i, j : (A_i, B_j) \text{ is a superedge}$$

$$\sum_{v \in B_j \text{ s.t. } (u,v) \in E(G)} f_{uv} \leq p_u \qquad \forall 1 \leq i, j \leq k, \forall u \in A_i$$

$$\sum_{u \in A_i \text{ s.t. } (u,v) \in E(G)} f_{uv} \leq p_v \qquad \forall 1 \leq i, j \leq k, \forall v \in B_j$$

$$f_{uv} \geq 0 \qquad \forall u \in A, v \in B \text{ s.t. } (u, v) \in E(G).$$

In the IP corresponding to LP 1, p_x for $x \in A \cup B$ is a binary variable which specifies whether vertex x has been chosen or not in our integral solution. (In the LP, intuitively it specifies the fraction of vertex x that is chosen.) In the IP, for all i, j such that (A_i, B_j) is a superedge, choose $u \in A_i, v \in B_j$ such that u, v are both chosen and set $f_{uv} = 1$; set $f_{u'v'} = 0$ for all other $u' \in A_i, v' \in B_j$. (In the LP, f specifies the "flow" from u to v and satisfies capacity constraint p_x on each *vertex* $x \in A \cup B$.)

Our algorithm, called MinRepAlg, for rounding LP 1 is relatively simple, though its proof is involved and is based on an interesting generalization of the birthday paradox. The algorithm is as follows.

1. Find an optimal solution f^*, p^* to LP 1.
2. For each $x \in A \cup B$, let $p_x^1 = \min\{1, \sqrt{q}p_x\}$.
3. Let $S_1 = \emptyset$ be the current set of selected elements.
4. Repeat the following $O(\log k)$ times: for each vertex $x \in (A \cup B) - S_1$, flip an independent biased coin and put x into S_1 with probability p_x^1.

Since in MinRepAlg, we amplify each (probability) variable p by a factor $\sqrt{q}$, the objective function would be at most $\sqrt{q}$ times the optimum solution to LP 1. Next, we show that, for each currently-uncovered superedge, the probability that one iteration will cover that superedge is boundedly away from 0. Since there are at most k^2 such pairs, with high probability after $O(\log k)$ iterations of the while loop, with total cost $O(\sqrt{q}\log k)z_{LP}^*$, we cover all superedges in supergraph $\mathcal{H}$.

Theorem 1. *If we choose each vertex $x \in A \cup B$ with probability p_x^1, any single superedge (A_i, B_j) is covered with constant probability.*

The proof of the above theorem uses the following lemma.

Lemma 2. *Consider a superedge (A_i, B_j) for which LP 1 routes one unit of flow f from vertices $u \in A_i$ to $v \in B_j$ and satisfies capacity constraints with respect to the p variables. Then there exists a flow $\hat{f}$ from vertices $u \in A_i$ to vertices $v \in B_j$ that*

1. *has value at least $\frac{1}{3}$ and at most 1,*
2. *that satisfies the capacity constraint p_x on each vertex $x \in A_i \cup B_j$, and*
3. *such that every nonzero $\hat{f}_{uv}$ is at least $1/(6q)$.*

Proof. We start with a flow $f' = f$ initially and decrease it in iterations until its flow from A_i to B_j becomes at most $\frac{1}{3}$. We also maintain node capacities $p' = p$ initially and reduce them in each iteration maintaining the property that the modified flow f' is a feasible flow for the modified node capacities p'. We build a new flow $\hat{f} = 0$ initially and increase it iteratively such that in each iteration, we increase flow $\hat{f}$ by at least some α on one edge from A_i to B_j and simultaneously, we decrease flow f' by at most 2α; we will ensure that $\alpha \geq 1/(6q)$. We do this increasing of $\hat{f}$ and decreasing of f' in such a manner that flow $f' + \hat{f}$ always satisfies capacity constraints p. Thus when the flow f' becomes less than $\frac{1}{3}$, the flow $\hat{f}$ is at least $\frac{1}{3}$ and we are done.

Now consider f' whose flow is at least $\frac{1}{3}$ during the process. Let $A_i' = \{x \in A_i, p_x' < \frac{1}{6q}\}$ and $B_j' = \{x \in B_j, p_x' < \frac{1}{6q}\}$. First, we show that there is an edge $(u, v) \in E(G)$ with $u \in A_i - A_i'$ and $v \in B_j - B_j'$. If it is not the case, all flow of f' should pass through either a vertex of A_i' or a vertex B_j' and thus its flow is less than $2q\frac{1}{6q} = \frac{1}{3}$, a contradiction. Let $\alpha = \min\{p_u', p_v'\} \geq \frac{1}{6q}$. We now add a

flow α from u to v in $\hat{f}$ and reduce the flow f' as follows: Assume without loss of generality that $p'_u \leq p'_v$. Then we reduce the flow in f' along all edges incident on u to zero. Note that the total flow reduction in this step is at most α. We also reduce flow arbitrarily along edges incident to v other than (u, v) such that the total flow on these edges is at most $p'_v - \alpha$. The total flow reduction in this step is also bounded by α. Finally, we reduce p'_u and p'_v by α. This maintains the property that flow f' is feasible for capacities p', and that $f' + \hat{f}$ is feasible for the original capacities p. In this way, the flow of f' is decreased by at most 2α, and the flow of $\hat{f}$ on any one edge has been increased by at least $\frac{1}{6q}$. □

We are now ready to prove Theorem 1.

Proof. **[of Theorem 1]** Fix i, j such that (A_i, B_j) is a superedge. First by Lemma 2, we obtain a flow $\hat{f}$ from the flow f in LP 1 and use the properties of $\hat{f}$ instead of f in the statement of the lemma in the rest of the proof. Let $\hat{p}_x \leq p_x$, for each vertex $x \in A_i \cup B_j$, be the total flow of $\hat{f}$ passing through vertex x.

If $\hat{p}_x \geq \frac{1}{\sqrt{q}}$, for $x \in A_i \cup B_j$, then since $\sqrt{q}p_x \geq \sqrt{q}\hat{p}_x \geq 1$, vertex x is chosen in our random selection with probability 1. Without loss of generality, assume that $x \in A_i$. Let N_x be the set of all vertices $y \in B_j$ for which x has positive flow $\hat{f}_{xy}$ to y. If there is a y with $\hat{p}_y \geq \frac{1}{\sqrt{q}}$, then vertex y is also chosen in our random selection with probability 1. Thus in this case we will satisfy the superedge (A_i, B_j) with probability 1 and we are done. If it is not the case, then each vertex $y \in N_x$ will be selected in our random process with probability at least $\sqrt{q}\hat{f}_{xy}$. This means that the probability that we do not select in our random process any vertices in N_x is at most $\Pi_{y \in N_x}(1 - \sqrt{q}\hat{f}_{xy}) \leq e^{-\sqrt{q}\sum_{y \in N_x} \hat{f}_{xy}} \leq e^{-\sqrt{q}\frac{1}{\sqrt{q}}} = \frac{1}{e}$ (since $\sum_{y \in N_x} \hat{f}_{xy} = \hat{p}_x \geq \frac{1}{\sqrt{q}}$). Thus with probability at least $1 - \frac{1}{e}$, we select a vertex in N_x and thus satisfy the superedge (A_i, B_j).

In the rest of the proof, we assume $\hat{p}_x \leq \frac{1}{\sqrt{q}}$, for $x \in A_i \cup B_j$, and thus $\sqrt{q}\hat{f}_{uv} \leq 1$ for $u \in A_i$ and $v \in B_j$.

The outline of the rest of the proof is as follows. Instead of directly analyzing the probability that the randomized rounding chooses both endpoints of some edge in $G[A_i \cup B_j]$, for a general bipartite graph between A_i and B_j with $q = |A_i| = |B_j|$, we first transform the bipartite graph, in a natural way, into a perfect matching graph. We do this by replacing a vertex v of degree $d(v)$ by $d(v)$ "clones," associating a different edge incident to v with each clone, and keeping the flow values on edges unchanged. We then choose each clone with probability $\sqrt{q}$ times the flow on the incident edge. (Note that the scaling factor is the square root of q, not the square root of the number of boys or girls in the perfect matching.) We argue that with at least positive constant probability, there is an edge e of $G[A_i \cup B_j]$ with at least one clone of each endpoint of e chosen. However, this is not what algorithm MinRepAlg does, in fact (it doesn't detour through a perfect matching graph), so we then argue that the probability that algorithm MinRepAlg chooses both endpoints of some edge of $G[A_i \cup B_j]$ is at least as high as it is in the perfect matching, and hence at least a positive constant.

First, we construct a bipartite graph $M = (A', B', E')$, for the given i, j, in which for each vertex $a \in A_i$ (resp., $b \in B_j$), we put r vertices $a^1, a^2, \ldots, a^r$ (resp., $b^1, b^2, \ldots, b^r$) in A' (resp., B') called *clones* of vertex a (resp., b), where r is the number of edges incident to a (resp., b) in $G[A_i \cup B_j]$ that carry nonzero flow (and thus a flow of at least $\frac{1}{6q}$) in $\hat{f}$. We associate each clone a^i of a (resp., b^j of b) with a different edge of G incident to a (resp., b). We put edges between vertices (clones) in E' corresponding to edges in $G[A_i \cup B_j]$ that carry a nonzero flow in $\hat{f}$ (and we put this flow as the flow of the new edge). Since each edge carrying positive flow in the bipartite graph between A_i and B_j gives rise to one edge in M whose endpoints have degree 1, M is a bipartite perfect matching, with $|A'| = |B'|$, which is at most the number of edges in $G[A_i \cup B_j]$.

We now consider a random process in which we build a set S by selecting each vertex (clone) c in M independently with probability $\sqrt{q}$ times the $\hat{f}$ flow of the unique edge incident to c in M. Let the subset of $A_i \cup B_j$ chosen by MinRepAlg be called S_1. We will prove two things: (1) first, that the chance that, in the perfect matching graph M, S contains an edge, is at least $1 - e^{-1/54} > 0$, and (2) second, the chance that S_1 contains an edge in G is at least as large as the chance that S contains an edge in the perfect matching graph M.

Now we prove (1), that both endpoints of some edge in M are chosen with constant probability. Consider one fixed edge $d = (c_A, c_B)$ carrying a flow $\hat{f}_d$. We select both c_A and c_B with probability $(\sqrt{q}\hat{f}_d)^2 = q\hat{f}_d^2$. Thus with probability $1 - q\hat{f}_d^2$, edge d will be not selected. The probability that no edges are selected then is at most $\Pi_{d \in E'}(1 - q\hat{f}_d^2)$. (We have independence because the graph is a perfect matching.) Since each edge carries a flow of at least $\frac{1}{6q}$ and the total flow is at most one (by Lemma 2), $|E'| \leq 6q$. Hence, since flow of $\hat{f}$ is at least $\frac{1}{3}$ by Lemma 2, $\Pi_{d \in E'}(1 - q\hat{f}_d^2) \leq e^{-q\sum_{d \in E'} \hat{f}_d^2} \leq e^{-q\frac{(\sum_{d \in E'} \hat{f}_d)^2}{|E'|}} \leq e^{-q\frac{(\frac{1}{3})^2}{6q}} = e^{-\frac{1}{54}}$. Thus with constant probability $1 - e^{-\frac{1}{54}} > 0$ we satisfy any one given superedge. This completes the proof of (1).

Now we prove (2). Build a new probabilistic process as follows. Define p_x^2, for $x \in A_i \cup B_j$, to be the probability that at least one of the clones of x is chosen to be in S. This is, of course, at most the sum of the probabilities that each individual clone is chosen to be in S, which is itself at most the probability that $x \in S_1$ (since the flow values add). Build a set S_2 by choosing each node $x \in A_i \cup B_j$ independently with probability p_x^2. The algorithm, on the other hand, builds S_1 using probabilities $p_x^1 \geq p_x^2$. It is a fairly obvious fact that, since $p_x^1 \geq p_x^2$, the chance that S_1 contains an edge is no smaller than the chance that S_2 contains an edge, but we prove it anyway.

Lemma 3. *Suppose we are given an r-node graph H and two probabilities $p_x^1 \geq p_x^2$ for each vertex x. Consider experiment E_ℓ, for $\ell = 1, 2$, with probability measure P_ℓ, in which we build set S_ℓ by putting each vertex x into S_ℓ with probability p_x^ℓ, independently. Then $P_1[S_1$ contains an edge$] \geq P_2[S_2$ contains an edge$]$.*

Proof. We can pick *one* sequence of r independent random reals ξ_x in $[0,1]$ and put x into S_ℓ if $\xi_x \leq p_x^\ell$. As $S_2 \subseteq S_1$ always, in every run in which S_2 contains an edge, so does S_1. □

But now we can view the construction of S_2 as putting a node x into S_2 if and only if at least one of its clones is chosen for S. It is clear that S contains an edge in M implies that S_2 contains an edge in G (but not the converse), so that the chance that S contains an edge is dominated by the chance that S_2 contains an edge, which itself is dominated by the chance that S_1 contains an edge, and we are done with the proof of Theorem 1. □

2.1 The Integrality Ratio of Min Rep

Next, we show that the integrality ratio of LP 1 is indeed $\Omega(n^{\frac{1}{3}-\varepsilon})$ for all $\varepsilon > 0$ and thus our algorithm in this section is essentially the best that we can hope for using the LP.

Theorem 4. *The integrality ratio of LP 1 for* Min Rep *is* $\Omega(n^{1/3-\varepsilon})$ *for any* $\varepsilon > 0$*, for all large enough* n*.*

Proof. Consider an instance of Min Rep with $k = n/q$ groups of q boys each and $k = n/q$ groups of q girls each. Between the ith group A_i of boys and the jth group B_j of girls there is a random perfect matching. It is clear that one can assign $f_e = 1/q$ for any edge e and $p_u = 1/q$ for any vertex u. This implies that $z^*_{LP} \leq 2n/q$. To study the integrality ratio, we look at the smallest feasible set S (i.e., the smallest set of vertices such that for all i, j, there is at least one edge between A_i and B_j both of whose endpoints are in S). Let S be a feasible set, $s = |S|$. The size s of S is the sum of $2n/q$ terms, one for each A_i and B_j. Let $a = s/(2n/q) = sq/(2n)$, the average size of the intersection of S with some A_i or B_j. Of the $2n/q$ terms, whose sum is s, fewer than $1/4$ of them (i.e., $(1/2)n/q$) can exceed $4a = 2sq/n$, and hence at least $(3/2)n/q$ of them are at most $4a$. Since at most n/q of them can be intersections with the n/q A_i's, at least $(1/2)n/q$ of them are intersections with n/q B_j's. Similarly, at least $(1/2)n/q$ of them are intersections with the n/q A_i's. Hence there are sets $I \subseteq \{1, 2, ..., n/q\}$ and $J \subseteq \{1, 2, ..., n/q\}$, $|I|, |J| = (1/2)n/q$ (provided that n/q is even), such that $|S \cap A_i| \leq 4a = 2sq/n$ for all $i \in I$ and $|S \cap B_j| \leq 4a = 2sq/n$ for all $j \in J$, and such that there is an edge between $S \cap A_i$ and $T \cap B_j$. Let S_i be any subset of A_i which contains $S \cap A_i$ and which has size exactly $4a$. Analogously, let T_j be any subset of B_j which contains $T \cap B_j$ and which has size exactly $4a$. Clearly there is an edge between S_i and T_j.

Fix an s and let $a = sq/(2n)$. As just shown, the existence of an S, of size s, for graph G implies the existence of $I, J \subseteq \{1, 2, ..., n/q\}$, $|I| = |J| = (1/2)n/q$, $S_i \subseteq A_i$ for all $i \in I$, and $T_j \subseteq B_j$ for all $j \in J$, $|S_i| = |T_j| = 4a$, such that for all $i \in I, j \in J$, the perfect matching in G between A_i and B_j contains an edge between S_i and T_j. (S_i, T_j have size exactly $4a$.) Hence the probability that there is an S is at most the probability that there exist $I, J \subseteq \{1, 2, ..., n/q\}$, both of size $(1/2)n/q$, and $S_i \subseteq A_i, T_j \subseteq B_j$ for all $i \in I, j \in J$, with $|S_i| = |T_j| = 4a$,

such that for all $i \in I, j \in J$, the perfect matching in G between A_i and B_j contains an edge between S_i and T_j.

Given fixed $I, J, (S_i), (T_j)$, what is the probability that the random graph contains, for each $i \in I, j \in J$, an edge whose left endpoint is in S_i and whose right one is in T_j? The chance that the random graph does *not* contain both endpoints of some edge in the random perfect matching between A_i and B_j is the chance that all the edges in the (i,j) perfect matching (the one between A_i and B_j) emanating from S_i end outside T_j. There are exactly $4a$ such edges. We will prove a *lower* bound on the probability that a random perfect matching does *not* contain both endpoints of some edge whose left endpoint is in S_i and whose right one is in T_j. In order for the mate of each vertex in S_i to lie outside of T_j, the mate of the first one must be chosen to be one of $q-4a$ nodes not in T_j among the q vertices in B_j, the mate of the second must be chosen to be one of the remaining $q-4a-1$ nodes not in T_j among the remaining $q-1$ vertices in B_j, etc. Hence the probability is exactly $\frac{q-4a}{q}\frac{q-4a-1}{q-1}\cdots\frac{q-4a-(4a-1)}{q-(4a-1)} \geq \left(\frac{q-8a}{q}\right)^{4a} = \left(1-\frac{8a}{q}\right)^{4a}$. Hence the chance that the (i,j) perfect matching *does* contain an edge whose left endpoint is in S_i and whose right one is in T_j is at most $1-(1-8a/q)^{4a}$. The chance that the random matching works for all $(n/q)^2/4$ pairs (i,j) with $i \in I, j \in J$ is at most $[1-(1-8a/q)^{4a}]^{(n/q)^2/4}$. Let $A = \binom{n/q}{n/(2q)}^2 \binom{q}{4a}^{n/q}\left[1-\left(1-\frac{8a}{q}\right)^{4a}\right]^{(n/q)^2/4}$, the first binomial coefficient representing the choices of I and J, the second representing the subsets S_i of A_i and T_j of B_j. If $A < 1$, then there is a fixed graph for which no set S of size s is good. $A \leq 2^{2n/q} q^{4an/q}\left[1-\left(1-\frac{8a}{q}\right)^{4a}\right]^{(n/q)^2/4}$. We choose $a = \sqrt{q/32}$ so that $q/(8a) = 4a$. Note that $(1-8a/q)^{4a} = (1-1/(q/(8a)))^{q/(8a)} \geq 1/4$ for $q/(8a) = 4a$ sufficiently large. So $[1-(1-8a/q)^{4a}]^{(n/q)^2/4} \leq [1-(1/4)]^{(n/q)^2/4}$. Letting $q = n^\delta$ for a fixed δ, we have $A \leq 2^{2n^{1-\delta}}(n^\delta)^{4an/q}(3/4)^{(n/q)^2/4}$. Since $4an/q = 4n^{1-\delta}\sqrt{q/32} = (1/\sqrt{2})n^{1-\delta/2}$, we have $A \leq (3/4)^{(1/4)n^{2(1-\delta)}} 2^{2n^{1-\delta}} n^{\delta(1/\sqrt{2})n^{1-\delta/2}}$. We have $A \leq 2^{-0.01n^{2(1-\delta)}+2n^{1-\delta}+(\lg n)(\delta/\sqrt{2})n^{1-\delta/2}}$. Since obviously $2(1-\delta) > 1-\delta$, we will have $A < 1$, in fact, $A \to 0$, if $2(1-\delta) > 1-\delta/2$, i.e., $2-2\delta > 1-\delta/2$, i.e., $1 > (3/2)\delta$, i.e., $\delta < 2/3$. Hence if $\delta < 2/3$, then for $q = n^\delta$ and $a = \sqrt{q/32}$, as specified above, there is an instance for which no set S of size $a(2n/q)$ is feasible. For this instance, $z^*_{IP} > \sqrt{q/32}(2n/q)$. Since $z^*_{LP} \leq 2n/q$, the integrality ratio exceeds $\sqrt{q/32}$, which is $\Omega(n^{\delta/2})$. Since $\delta < 2/3$ is arbitrary, for all $\varepsilon > 0$ the integrality ratio is $\Omega(n^{1/3-\varepsilon})$. □

3 $O(n^{\frac{1}{3}})$-Approximation Algorithm for Max Rep

In this section, we provide an $O(n^{\frac{1}{3}})$-approximation algorithm for Max Rep. However, in contrast to Section 2, in which we use a natural LP for the problem, we can show that the integrality gap of a natural LP for Max Rep is $\Omega(\sqrt{n})$. This

forces us to use a combinatorial approach to obtain a non-trivial approximation factor $O(n^{\frac{1}{3}})$.

We consider the best of three algorithms:

1. *Matching:* Find a maximal matching in the supergraph $\mathcal{H}$. For each edge (A_i, B_j) in this matching, pick $a_i \in A_i$ and $b_j \in B_j$ such that $(a_i, b_j) \in E(G)$.
2. *Random-Choice:* For each B_j, pick $b_j \in B_j$ at random. For each A_i, pick $a_i \in A_i$ that has the maximum number of edges to the set of all selected b_j vertices. Repeat, flipping the roles of A and B.
3. *Random-Neighbor:* For each $a \in A$, construct a solution in the following fashion and eventually pick the best such solution: For each B_j, pick $b_j \in B_j$ at random from amongst those vertices that are neighbors of a (if there is no neighbor of a in B_j, pick an arbitrary $b_j \in B_j$). For each A_i, pick $a_i \in A_i$ that has the maximum number of edges to the selected b_j vertices over all j. Repeat, flipping the roles of A and B.

Theorem 5. *The best of these three algorithms is a $2(2n)^{1/3}$-approximation algorithm.*

Proof. Suppose that the maximal matching in $\mathcal{H}$ has size ℓ. Renumber the A_i's and B_j's such that the matching has edges $(A_i, B_i), i = 1, \ldots, \ell$. There are no edges between A_i and B_j for $i, j > \ell$. Let $A' = \cup_{i=1}^{\ell} A_i$, and $B' = \cup_{j=1}^{\ell} B_j$. The edges in the optimal solution can be decomposed into two groups: those that go between A' and B and those that go between A and B'. (Edges between A' and B' appear in both.) Hence the optimal solution restricted to one of these two groups much contain at least half the number of edges in the optimal solution. Without loss of generality, assume that the optimal solution restricted to edges between A and B' contains at least half the number of edges in the optimal solution.

We introduce some notation to facilitate the analysis. Let $X_{ij} = 1$ if there is an edge in the optimal solution from A_i to B_j (and 0 otherwise). Let N_{ij} be the number of edges from the optimal vertex a_i^* in A_i to the remaining vertices in B_j, called "nonoptimal" since they're not in the optimal solution.

Define p and r as follows:

$$\sum_{i=1}^{k} \sum_{j=1}^{\ell} X_{ij} = p(k\ell) \tag{2}$$

$$\sum_{i=1}^{k} \sum_{j=1}^{\ell} N_{ij} = r(\ell n). \tag{3}$$

Thus $OPT \leq 2 \sum_i^k \sum_{j=1}^{\ell} X_{ij} = 2pk\ell$ and algorithm *Matching* gives a $2pk$ approximation.

Next, we analyze algorithm *Random-Choice*. The algorithm picks random vertices in B and picks the best vertices in A for the chosen vertices in B.

In order to obtain a lower bound on the number of superedges covered, we compute the expected number of superedges covered if we pick random vertices in $B_j, j = 1, \ldots, \ell$, and instead of the best vertex in A_i, we use the vertex $a_i^* \in A_i$ which is in the optimal solution.

For a superedge (A_i, B_j), $i = 1, \ldots, k$ and $j = 1, \ldots, \ell$, the probability that this edge is covered by *Random-Choice* is $(X_{ij} + N_{ij})/(n/k)$. Hence the expected number of superedges covered is at least $\frac{k}{n} \sum_{i=1}^{k} \sum_{j=1}^{\ell} (X_{ij} + N_{ij}) = \frac{k}{n}(pk\ell + r\ell n)$. Hence the approximation ratio of algorithm *Random-Choice* is at most $\frac{2pk\ell}{\frac{k}{n}(pk\ell + r\ell n)} \leq \min\left\{\frac{2n}{k}, \frac{2p}{r}\right\}$.

Finally, we analyze algorithm *Random-Neighbor*. Suppose the vertex a chosen by the algorithm in the first step is in, say, A_h, and also is in the optimal solution. Consider set B_j and the vertex $b_j^* \in B_j$ in the optimal solution. The number of edges from a to B_j is $X_{hj} + N_{hj}$. The algorithm picks a random neighbor of a in B_j. Thus the probability that b_j^* is chosen is $\frac{X_{hj}}{X_{hj}+N_{hj}}$. As before, instead of picking the best choice of vertices in A for the chosen vertices in B, we lower bound the expected number of superedges covered by replacing the vertex a_i by the vertex $a_i^* \in A_i$ in the optimal solution. If $b_j^* \in B$ is chosen, the number of edges from the set of a_i^*'s is $\sum_{i=1}^{k} X_{ij}$. Thus the expected number of superedges covered is at least $\sum_{j=1}^{\ell} \frac{X_{hj}}{X_{hj}+N_{hj}} (\sum_{i=1}^{k} X_{ij})$. In this calculation, we assumed that $a = a_h^*$ was chosen in the first step. We average over $h = 1, \ldots, k$. Thus the expected number of covered edges is at least $\frac{1}{k} \sum_{h=1}^{k} \sum_{j=1}^{\ell} \frac{X_{hj}}{X_{hj}+N_{hj}} (\sum_{i=1}^{k} X_{ij})$. Let $C_j = \sum_{i=1}^{k} X_{ij}$ and let $N_j = \sum_{i=1}^{k} N_{ij}$. Then the previous expression is $\frac{1}{k} \sum_{j=1}^{\ell} C_j \sum_{h=1}^{k} \frac{X_{hj}}{X_{hj}+N_{hj}}$.Note that $\sum_{h=1}^{k} \frac{X_{hj}}{X_{hj}+N_{hj}} \geq C_j \left(\frac{1}{1+\frac{N_j}{C_j}}\right)$ by the arithmetic-geometric-harmonic means inequality. In order to obtain a lower bound for this expression, consider the minimum value of $\sum_j \frac{C_j^3}{C_j + N_j}$ over all choices of C_j and N_j subject to the constraint that $\sum_j C_j$ and $\sum_j N_j$ are fixed. Now let C_j, N_j be the respective values that minimize this expression. Then for any indices $f \neq g$, the function (of x) $\frac{{C_f}^3}{C_f + (N_f - x)} + \frac{{C_g}^3}{C_g + (N_g + x)}$ must be minimized for $x = 0$. Thus, the derivative of this function at $x = 0$ must be zero. Hence ${C_f}^3/(C_f + N_f)^2 = {C_g}^3/(C_g + N_g)^2$. Hence there is a constant α such that for all indices f, $(C_f + N_f)^2 = \alpha {C_f}^3$. Hence $\alpha \sum_j C_j^{3/2} = \sum_j C_j + \sum_j N_j = pk\ell + r\ell n$. Thus the expected number of superedges covered is at least $\frac{1}{k} \sum_{j=1}^{\ell} \frac{C_j^3}{\alpha C_j^{3/2}} = \frac{1}{k} \sum_{j=1}^{\ell} \frac{C_j^{3/2}}{\alpha} \geq \frac{1}{k} \frac{(\sum_{j=1}^{\ell} C_j^{3/2})^2}{(pk\ell + r\ell n)}$. Convexity of $f(x) = x^{3/2}$ shows that this expression is minimized when all C_j are equal. Hence a lower bound on the expected number of superedges covered is given by $\frac{1}{k} \frac{(\ell(pk)^{3/2})^2}{(pk\ell + r\ell n)} = \frac{\ell^2 p^3 k^2}{pk\ell + r\ell n}$. Thus the approximation ratio of this procedure is at most $\frac{2pk\ell(pk\ell + r\ell n)}{\ell^2 p^3 k^2} = \frac{2(1 + (\frac{r}{p})\frac{n}{k})}{p}$.

Thus we have the following upper bounds on the approximation ratio of the algorithm: $2pk, 2\frac{n}{k}, 2\frac{p}{r}, 2\frac{1+(\frac{r}{p})\frac{n}{k}}{p}$. We consider two cases:

Case 1: $(\frac{r}{p})\frac{n}{k} \geq 1$. In this case, the fourth bound is at most $\frac{4(\frac{r}{p})\frac{n}{k}}{p}$. The product of the first, third and fourth bounds is $16pk \times \frac{p}{r} \times \frac{(\frac{r}{p})\frac{n}{k}}{p} = 16n$. Hence at least one upper bound is at most $2(2n)^{1/3}$.

Case 2: $(\frac{r}{p})\frac{n}{k} < 1$. In this case, the fourth bound is at most $\frac{4}{p}$. Now the product of the first, second and fourth bound is $2pk \times \frac{2n}{k} \times \frac{4}{p} = 16n$. Hence at least one upper bound is at most $2(2n)^{1/3}$. □

4 Reduction From Densest k-Subgraph to Max Rep

In this section, we consider the Densest k-Subgraph (DkS) problem, in which the goal is to find an induced subgraph of order k of a given graph with the maximum number of edges.

Theorem 6. *An $f(n)$-approximation algorithm for* Max Rep *implies the existence of a randomized $O(f(n))$-approximation algorithm for DkS.*

Proof. From an instance of DkS, we produce an instance of Max Rep by randomly dividing vertices of the given graph for DkS into k groups of equal size $s = \lfloor \frac{n}{k} \rfloor$, e.g., by using a random permutation of all vertices, and disregard the rest of vertices. Next we place $\lfloor k/2 \rfloor$ groups on one side (call this L) and the other $\lceil k/2 \rceil$ groups on the other side (call this R) of the instance for Max Rep. Any feasible solution to the Max Rep instance obtained directly gives a solution to the original DkS instance of the same value. Both instances have the same number n of vertices.

We claim that the expected value of the optimal solution to the Max Rep instance obtained thus is at least a constant times the optimal value for DkS. Consider the optimal solution S of size k to the DkS instance. We produce a solution to the Max Rep instance as follows. For every group in the instance, if the group contains a unique vertex of S, then this unique vertex is picked as the group representative. If there are zero or at least 2 vertices from S then an arbitrary vertex is picked as the group representative (and we don't count edges incident to that vertex). We show that the expected value of this solution is at least a constant times the value of the DkS optimal solution. For any vertex $v \in S$, with constant probability v is placed alone in its group. Furthermore, for two distinct vertices $u, v \in S$, the probability that u and v are both alone in their groups, u is in the L side and v is on the R side, is bounded below by a constant greater than 0. Hence $E[z^*_{MaxRep}] \geq cz^*_{DkS}$, for a positive constant c.

Now the reduction is apparent. Given an $f(n)$-approximation algorithm A for Max Rep, take an n-node instance I of DkS, randomly convert it as above into an n-node instance I' of Max Rep, use A to generate a solution $A(I')$ of value at least $f(n)z^*_{MaxRep}$, and report $A(I')$ as a feasible solution to the DkS instance. That $E[z^*_{MaxRep}] \geq cz^*_{DkS}$ implies that the expected size of the DkS solution returned is at least $cf(n) \cdot z^*_{DkS}$. □

5 Conclusion

Obtaining improvements over the approximation guarantees in this paper would be instructive. Given the reduction demonstrated in Section 4, possibly one can use ideas from the Densest k-Subgraph algorithm to build an $n^{1/3-\delta}$-approximation algorithms for some fixed $\delta > 0$. However, the main remaining open problem is whether, for Max Rep or Min Rep, there is a $O(n^{\varepsilon})$-approximation algorithm for each $\varepsilon > 0$.

References

1. A. Aazami and M. D. Stilp, *Approximation algorithms and hardness for domination with propagation*, in APPROX 2007, pp. 1–15.
2. S. Arora, L. Babai, J. Stern, and Z. Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, J. Comput. System Sci., 54 (1997), pp. 317–331.
3. A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. P. Woodruff, *Transitive-Closure Spanners*, ArXiv e-prints, (2008).
4. L. Breslau, I. Diakonikolas, N. Duffield, Y. Gu, M. Hajiaghayi, D. Johnson, H. Karloff, M. Resende, and S. Sen, *Optimal Node Placement For Path-Disjoint Network Monitoring*, 2008, manuscript.
5. N. Chen, *On the approximability of influence in social networks*, in SODA 2008, pp. 1029–1037.
6. M. Elkin and D. Peleg, *The hardness of approximating spanner problems*, Theory Comput. Syst., 41 (2007), pp. 691–729.
7. U. Feige, G. Kortsarz, and D. Peleg, *The dense k-subgraph problem*, Algorithmica, 29 (2001), pp. 410–421.
8. M. Feldman, G. Kortsarz, and Z. Nutov, *Improved approximation for the directed steiner forest problem*, in SODA 2009, pp. 922-931.
9. A. Gupta, M. Hajiaghayi, and A. Kumar, *Stochastic steiner tree with non-uniform inflation*, in APPROX 2007, pp. 134–148.
10. M. T. Hajiaghayi, G. Kortsarz, V. S. Mirrokni, and Z. Nutov, *Power optimization for connectivity problems*, Math. Program., 110 (2007), pp. 195–208.
11. R. Hassin and D. Segev, *The set cover with pairs problem*, in FSTTCS 2005, pp. 164–176.
12. D. S. Hochbaum, ed., *Approximation algorithms for NP-hard problems*, PWS Publishing Co., Boston, MA, USA, 1997. see the section written by Arora and Lund.
13. S. Khot, *Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique*, SIAM J. Comput., 36 (2006), pp. 1025–1071.
14. G. Kortsarz, *On the hardness of approximating spanners*, Algorithmica, 30 (2001), pp. 432–450.
15. G. Kortsarz, R. Krauthgamer, and J. R. Lee, *Hardness of approximation for vertex-connectivity network design problems*, SIAM Journal on Computing, 33 (2004), pp. 185–199.
16. D. Peleg, *Approximation algorithms for the Label-Cover$_{\mathrm{MAX}}$ and Red-Blue Set Cover problems*, J. Discrete Algorithms, 5 (2007), pp. 55–64.