On the Hardness of Approximating Minimization Problems

(Extended abstract)

Carsten Lund
Mihalis Yannakakis

AT&T Bell Laboratories
Murray Hill, NJ 07974

Abstract

We prove results indicating that it is hard to compute efficiently good approximate solutions to the Graph Coloring, Set Covering and other related minimization problems. Specifically, there is an $\epsilon > 0$ such that Graph Coloring cannot be approximated with ratio n^ϵ unless P=NP. Set Covering cannot be approximated with ratio $c \log n$ for any $c < 1/4$ unless NP is contained in DTIME[$n^{\text{poly} \log n}$]. Similar results follow for related problems such as Clique Cover, Fractional Chromatic Number, Dominating Set and others.

1 Introduction

Graph Coloring and Set Covering are two important minimization problems that have been extensively studied over the last 25 years. They are both conceptually simple and have served as paradigms for problems from many application areas. In the Graph Coloring problem we are given a graph G and wish to color its nodes with as few colors as possible so that no two adjacent nodes receive the same color. In the Set Covering problem we are given a finite collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of a finite set U, and we wish to compute a subcollection that contains as few sets as possible and covers U.

Both problems were shown to be NP-hard in Karp's original paper [19]. Since it is unlikely that they can be solved optimally in polynomial time, there has been a lot of work in exploring the possibility of obtaining efficiently near-optimal solutions. The usual metric for measuring the nearness to optimality of a solution is by the ratio of its cost to that of an optimal solution. The performance of an approximation algorithm is measured then by its worst-case ratio over all inputs of a given size. An algorithm achieves ratio r if for every instance it computes a solution whose cost is at most r times the cost of the optimal solution. (This is for minimization problems; in the case of maximization problems, the value of the computed solution must be at least $1/r$ times the optimal value.) The approximability status of Graph Coloring and Set Covering have been long-standing open problems.

A number of heuristics were developed for the Graph Coloring problem already from the 60's even before the discovery of NP-completeness. Johnson analyzed in 1974 the performance of many such heuristics in [18] and proved that their worst-case ratio is very poor, $\Omega(n)$ where n is the number of nodes. He also proposed another heuristic with slightly better performance, $\Theta(n/\log n)$. This has been improved since then, but still the currently best known ratio achievable by a polynomial-time approximation algorithm is only $O(n(\log\log n)^2/(\log n)^3)$ [16], a ratio that is worse than n^ϵ for any $\epsilon < 1$. Somewhat better ratios have been obtained in the special case of graphs that can be colored with a small number of colors (for example, for 3-colorable graphs) [6], [28]. On the negative side, the best known result is still that of Garey and Johnson from 1976 showing that if a polynomial-time approximation algorithm achieves a constant ratio smaller than 2 then P=NP [14].

In the case of the Set Covering problem, Johnson and Lovász proved in the mid 70's that a simple greedy heuristic achieves a ratio of $\ln(N) + 1 \approx 0.7\log_2 N$, where $N = |U|$ is the number of different elements [17], [21]. This result has served as an important paradigm in many contexts and has been extended several times. Chvátal generalized it to the weighted case, in which each set of the given collection has an associated weight and we wish to find a cover with the minimum total weight [8]. The above logarithmic factor bounds the ratio of the cost of the solution obtained by the greedy

25th ACM STOC '93-5/93/CA,USA

heuristic not only to the cost of the optimal solution, but also to the cost of the best *fractional* cover, i.e., the best solution to the Linear Programming relaxation of a standard formulation of the Set Covering problem as an Integer Program. The result was further generalized in different directions, for example, to the Integer Programming problem of minimization type with non-negative coefficients [10], to the case of submodular constraints [29], and to a continuous version [13]. On the negative side, essentially nothing was known until very recently. The problem was shown MAX SNP-hard in [26]. The results of [2] imply then that it does not have a polynomial time approximation scheme unless P=NP; i.e., there exists a constant $\epsilon > 0$ such that the problem can not be approximated in polynomial time with ratio smaller than $1 + \epsilon$ unless P=NP.

In this paper we show strong negative results on the approximability of both these problems. For the Graph Coloring problem we show that there is a constant $\epsilon > 0$ such that no polynomial time approximation algorithm can achieve ratio n^{ϵ} unless P=NP. Similar results follow for other related minimization problems. The Graph Coloring problem can be stated as the problem of covering the nodes of a graph by the minimum number of independent sets. Other graph covering problems of this type that are known to be (almost) equivalent to coloring with respect to approximability include: covering the nodes of a graph by the minimum number of cliques, covering the edges of a graph by cliques, and covering the edges of a bipartite graph by complete bipartite subgraphs [30], [24], [27]. Also a similar negative result holds for the approximation of the *fractional* coloring problem (we give the definition in the next section).

For the Set Covering problem we show that it cannot be approximated within ratio $c \log_2 N$ for any constant $c < 1/4$ unless NP is contained in DTIME[$n^{\text{poly} \log n}$]. Of course the same result follows for the generalizations of Set Covering mentioned above. Furthermore, similar results follow again for closely related minimization problems: Hypergraph Transversal (node cover), minimum Hitting Set, minimum Dominating Set in a graph, and the variant of the Set Covering problem where we wish to minimize the sum of the cardinalities of the sets in the cover.

Our proofs use the recent results from interactive proof systems and probabilistically checkable proofs and their surprising connection to approximation. This connection was first observed by Feige et al. [11]. They used it to prove a negative result on the approximation of the maximum clique and the maximum independent set problems, which was then tightened in [3] and [2] to show that these problems cannot be approximated with ratio n^{ϵ} for some constant $\epsilon > 0$ unless P=NP. Arora et al. [2] developed new interactive proof systems for NP which allowed them to prove that unless P=NP, Maximum 3-Satisfiability and a host of other problems that are hard for the class MAX SNP do not have a polynomial time approximation scheme. MAX SNP is a class of maximization problems defined syntactically in [26]; minimization problems can be "placed" in the class or shown hard for it indirectly through approximation-preserving reductions from their complementary maximization problems.

Other connections between interactive proof systems and approximability were observed in [9, 12, 5].

The interactive proof techniques can be most naturally applied to maximization problems. Interactive proof systems themselves can be viewed as maximization problems in which the goal is to find strategies for the provers that maximize the probability that the verifier accepts its input. To prove intractability for the approximation of minimization problems we must relate them to appropriate maximization problems.

We prove the nonapproximability of Graph Coloring by a reduction from the maximum Independent Set problem. It has been known for a long time (see [17]) that one can use an algorithm for the Independent Set problem with approximation ratio r to obtain an algorithm for Graph Coloring with ratio $r \log n$; our reduction shows a relationship between the two problems in the reverse direction. The proof draws in spirit from the recent techniques in that the construction has a (linear) algebraic flavor instead of the more usual graph-theoretic one. Our proof for the Set Covering problem is completely different. It uses a reduction from two-prover one-round interactive proof systems [22, 12].

In the rest of this abstract we sketch the constructions. In Section 2 we address the Graph Coloring problem and in Section 3 the Set Covering problem. Because of space limitations, most of the proofs are omitted from this extended abstract.

2 The Difficulty of Coloring

We shall give first preliminary definitions and notation. Then we shall describe in Section 2.1 the construction proving the main result stated in the Introduction on the nonapproximability of coloring. In Section 2.2 we prove a result for the case of graphs with bounded chromatic number: if there is a constant c such that for every k there is a polynomial time algorithm (possibly depending on k) that colors every k-colorable graph with $c \cdot k$ colors then P=NP. Finally, in Section 2.3 we discuss some other minimization problems related to Coloring.

For a graph G, we use $n(G)$ to denote its number of nodes, $\alpha(G)$ to denote the size of its largest independent (stable) set, and $\chi(G)$ to denote its chromatic num-

ber. Our reduction is from the maximum independent set problem. Arora et al [2] showed that it is NP-hard to approximate the maximum independent set problem within a factor n^ϵ for some $\epsilon > 0$. It will be convenient to take advantage of the structure of the graphs that are produced in their reduction (which is the same as the reduction of Feige et al [11]). From a Boolean 3CNF formula φ with m variables, they construct a graph G_φ on $n = m^{O(1)}$ nodes which are partitioned into r cliques $C_1, \ldots, C_r$ with the following properties:

1. If φ is satisfiable, then $\alpha(G_\varphi) = r$, i.e., G_φ has an independent set that contains exactly one node from each clique C_i (obviously an independent set cannot contain more than one node from a clique);

2. if φ is not satisfiable, then $\alpha(G_\varphi) < r/g$ where the "gap" g is n^ϵ for some $\epsilon > 0$.

We explain briefly the relationship to the probabilistically checkable proof for 3SAT: r is the number of random choices of the verifier; there is one clique C_i for each random choice, which contains one node for each possible combination of answers from the provers that causes the verifier to accept. (The graph contains more edges connecting nodes from different cliques if they correspond to contradictory answers to a common question.) Thus, for a probabilistically checkable proof with l random bits and q query bits, $r = 2^l$ and the size of each C_i is bounded by 2^q. We refer the reader to the references for more information on probabilistically checkable proofs; they are not needed to understand the rest of this section.

The basic idea of our reduction is as follows. Let G be a partitioned graph as above. We shall construct another graph H such that

(i) $\alpha(H)$ is proportional to $\alpha(G)$, and

(ii) if $\alpha(G) = r$, the chromatic number of H is $\chi(H) = n(H)/\alpha(H)$; specifically, we can transform any independent set I of G with r nodes to a maximum independent set I' of H such that we can color the nodes of H by appropriate "shifts" of I'.

It will follow that a gap can be transferred from the independent set problem in G to the chromatic number problem in H.

2.1 The Construction

Let G be a graph whose nodes are partitioned into cliques $C_1, \ldots, C_r$. Let p be a prime that is at least as large as the size of the largest C_i. We will work in the field Z_p of integers modulo p. We number arbitrarily the nodes of each C_i by distinct elements of Z_p, for example by the numbers $1, \ldots, |C_i|$. Each node v of G can be represented by a pair $< i, k >$, where the first component $i = 1, \ldots, r$ gives the index of the clique containing the node v, and the second component $k \in Z_p$ gives the number of the node in the clique C_i.

Each node of the new graph H will be represented by a 4-tuple $< i, k, y, w >$ where the first component $i = 1, \ldots, r$, the second component k is in Z_p and the other two components y, w are in Z_p^2. There is one node for every possible such 4-tuple. Thus, H has exactly rp^5 nodes.

Before specifying the edges of H, it will be helpful to describe how an independent set I of G is transformed to an independent set I' of H so that the above mentioned property (ii) holds. Let $v =< i, k_i >$ be a node of I from the ith clique C_i. Corresponding to this node v, the independent set I' of H contains a set of p^2 nodes $< i, k_i, y, k_i y >$, one for every possible third component $y \in Z_p^2$; that is, the first two components agree with v and the fourth component is equal to the product of the second and the third. A shift of I' is obtained by adding to the second component of each node of I' the same amount $s \in Z_p$ and adding to the fourth component the same vector $z \in Z_p^2$. The edges of H are specified so that every shift of I' is also an independent set.

More specifically, let $u_A =< i_A, k_A, y_A, w_A >$, $u_B =< i_B, k_B, y_B, w_B >$ be two nodes of H. The two nodes are *not* adjacent iff there is an element $s \in Z_p$ and a vector $z \in Z_p^2$ (the "shifts") such that

- both $< i_A, k_A - s >$ and $< i_B, k_B - s >$ are nodes of G and they are not adjacent,

- $w_A = (k_A - s)y_A + z$ and $w_B = (k_B - s)y_B + z$.

Note in particular that if two nonadjacent nodes of H have the same first component $i = i_A = i_B$, then they must also have equal second components $k_A = k_B$, because of the first condition and the fact that C_i is a clique in G. Furthermore, they cannot also have equal third components $y_A = y_B$ because then by the second condition they should also agree on the fourth components $w_A = w_B$, i.e., they must be the same node. The following lemma shows the basic properties of the construction.

Lemma 2.1

1. $p^2\alpha(G) \leq \alpha(H) \leq \max(p^2\alpha(G), p^2(\alpha(G) - 1) + r, pr)$.

2. *If* $\alpha(G) = r$ *then* $\alpha(H) = p^2 r$ *and the chromatic number of* H *is* $\chi(H) = n(H)/\alpha(H) = p^3$.

Proof: Omitted in this extended abstract. ∎

Corollary 2.2 *If we pick* $p \geq \frac{r}{\alpha(G)}, \sqrt{r}$ *(for example if* $p \geq r$*) then*

1. $\alpha(H) = p^2\alpha(G)$, *and*
2. *if* $\alpha(G) = r$ *then* $\chi(H) = n(H)/\alpha(H) = p^3$.

We are ready to prove the main theorem now.

Theorem 2.3 *There is a* $\delta > 0$ *such that it is NP-hard to approximate the chromatic number problem within a factor* n^δ.

Proof: By [2] there is a $\epsilon > 0$ such that given a 3CNF formula φ we can construct in polynomial time a graph G partitioned into r cliques C_i such that,

1. if φ is satisfiable then $\alpha(G) = r$, and
2. if φ is not satisfiable then $\alpha(G) < r/[n(G)]^\epsilon$.

Apply the above transformation, and assume that we choose p as in the above corollary. Let H be the resulting graph. If φ is satisfiable then $\chi(H) = p^3$. If φ is not satisfiable then $\chi(H) \geq n(H)/\alpha(H) = rp^5/p^2\alpha(G) \geq p^3[n(G)]^\epsilon$. The ratio between the two cases is at least $[n(G)]^\epsilon \geq [n(H)]^\delta$ for some constant $\delta > 0$. The precise value of δ (and the best choice for p) depends on the relationship between r, $n(G)$ and ϵ; that is, ultimately it depends on the relationship between the number of random bits, the number of query bits and the error probability that come out of the proof of [2]. ∎

Comments. The above transformation utilizes the structure of the graphs that are constructed in the recent reductions of [2] and the other papers for the non-approximability of the independent set problem. However, every graph can be brought in that form. Let G be an arbitrary graph. Form its product $G' = K_r \times G$ with the complete graph on $r \geq \alpha(G)$ nodes as follows. Every node of G' is a pair $< i, v >$ where $i = 1, \ldots, r$ is a node of K_r and v is a node of G. The nodes with the same first component i induce a clique C_i. Two nodes $< i, v >$, $< j, w >$ from different cliques are adjacent iff their second components v, w are equal or adjacent nodes of G. It is easy to see that $\alpha(G') = \alpha(G)$.

Apply now our transformation (denote it T_p) on G' to obtain a graph $H = T_p(K_r \times G)$. Assume that $p \geq r \geq \alpha(G)$. If r is an integer multiple of $\alpha(G)$, then it is easy to see by Corollary 2.2 that $\chi(H)$ is inversely proportional to $\alpha(G)$, namely, $\chi(H) = n(H)/\alpha(H) = p^3 r/\alpha(G)$. If r is not an integer multiple of $\alpha(G)$, then the chromatic number of $H = T_p(K_r \times G)$ lies between $p^3 r/\alpha(G)$ and $p^3\lceil r/\alpha(G)\rceil$.

2.2 Bounded chromatic number

The gap of n^δ of Theorem 2.3 holds only for graphs with unbounded chromatic number. We can deduce from the reduction also some *constant* gaps for graphs with bounded chromatic number, where the gap increases (going to infinity) as the chromatic number increases (going to infinity).

If G is a graph whose nodes are partitioned into cliques $C_1, \ldots, C_r$, we call r the *height* of the graph and call $\max|C_i|$ the *width* of the graph. If L_1, L_2 are two disjoint languages we say that an algorithm *distinguishes* between L_1 and L_2 if on input x, it outputs "Yes" if $x \in L_1$ and it outputs "No" if $x \in L_2$; the output is irrelevant if x is neither in L_1 nor in L_2. The results of [2] imply the following lemma.

Lemma 2.4 *For every constant* $g > 1$ *(the "gap"), there is a constant* w *such that the following problem is NP-hard. Given a partitioned graph* G *with width at most* w, *distinguish between the case that* $\alpha(G)$ *is equal to the height* r *of* G *and the case that* $\alpha(G) < r/g$.

Theorem 2.5 *For every constant* $g > 1$, *there is a constant* c *such that the following problem is NP-hard. Given a graph* H, *distinguish between the case that* H *is colorable with* c *colors and the case that the chromatic number of* H *is at least* $g \cdot c$.

Proof: Let $g' = g + 1$ be a constant gap for the independent set problem and let w be the implied width for g' from the preceding lemma. Let p be the smallest prime that is at least as large as g' and w, and let $c = p^3$. Clearly, p is a constant that depends on g, and the same is true of c.

Let G be a partitioned graph with width w and height r. Let $H = T_p(G)$. By Lemma 2.1 if $\alpha(G) = r$ then $\chi(H) = p^3 = c$. Suppose that $\alpha(G) < r/g'$. We can easily bound the three quantities on the right-hand side of the inequality of the first part of Lemma 2.1. The first and the second quantity are smaller than $p^2\alpha(G) + r < p^2(r/g') + r < p^2 r/g$. The third quantity is $pr < p^2 r/g$. Thus, $\alpha(H) < p^2 r/g$, and consequently $\chi(H) \geq n(H)/\alpha(H) > gp^3 = g \cdot c$. ∎

Stronger results are probably true in the case of bounded chromatic number. For instance, it should be possible at least to reverse the order of the two quantifiers in the theorem, i.e., to show that there is a constant number of colors c such that, even for c-colorable graphs one cannot achieve a constant factor approximation in polynomial time unless P=NP.

2.3 Related problems

We define some problems that have similar approximability properties to Graph Coloring.

Clique Partition: Given a graph G partition its nodes into the minimum number of cliques; equivalently, cover the nodes of G with the minimum number of cliques.

Clique Cover: Given a graph G, find the minimum number of cliques that cover the edges of G. This problem was studied and shown NP-hard in [30] and [24]. It is equivalent to the problem of finding a set U of minimum cardinality such that G can be expressed as the intersection graph of a collection of subsets of U.

Biclique Cover: Given a bipartite graph G, find the minimum number of complete bipartite subgraphs that cover the edges of G. This problem was shown NP-hard in [24]. It is related to communication complexity. In particular, the nondeterministic communication complexity of a predicate is equal to the logarithm of the biclique cover number of a bipartite graph associated with the predicate.

Fractional Chromatic Number: Given a graph G, find a collection of independent sets $I_1, \ldots I_t$ of G and corresponding nonnegative (possibly fractional) values $\lambda_1, \ldots \lambda_t$, so that the sum of the λ_i's is minimized, and for every node v of G the sum of values assigned to the independent sets containing v is at least 1. It is well known that we only need to assign nonzero value to at most n independent sets, i.e. there is an optimal solution with $t \leq n$. If we consider the Graph Coloring problem as a special case of the Set Covering problem where the collection of sets is the collection of all independent sets of the graph (it is not listed explicitly), then the fractional chromatic number is the optimal value of the corresponding Linear Programming relaxation. The fractional chromatic number is within a $\log n$ factor of the (ordinary integer) chromatic number. Grötschel, Lóvasz and Scrijver used the Ellipsoid algorithm in an interesting way to show that it is NP-hard to compute a weighted version of the fractional chromatic number [15]. The NP-hardness of the unweighted version appears to be new.

Theorem 2.6 *The following holds for each of the problems Clique Partition, Clique Cover, Biclique Cover and Fractional Chromatic Number. There is a $\delta > 0$ such that there does not exist a polynomial time approximation algorithm that achieves ratio n^δ unless P=NP.*

3 The Difficulty of Set Covering

A set system $\mathcal{S} = (U; S_1, S_2, \ldots, S_m)$ is a collection of sets $S_1, S_2, \ldots, S_m \subset U$. We say that $S_1, S_2, \ldots, S_m$ cover U if and only if $\bigcup_{i=1}^m S_i = U$.

In this section we will study the set covering problem: Given a collection of sets $S_1, S_2, \ldots, S_m \subset U = \{1, 2, \ldots, N\}$ find the minimum subcollection that covers $\{1, 2, \ldots, N\}$. We will show that this problem is hard to approximate within a factor of $c \log N$, for any $0 < c < 1/4$. We will then list some equivalent problems.

3.1 The Construction

Our construction uses a 2-prover 1-round interactive proof system constructed by Feige-Lovász [12] and Lapidot-Shamir [22]. Our construction depends on some specific properties of their proof system and we need to change the proof system slightly.

A 2-prover 1-round interactive proof system for a language L consists of three players: a probabilistic (computationally limited) verifier V and two deterministic (all-powerful) provers P_1 and P_2. Fix an input size n. The proof system has an associated finite set Q_i, $i = 1, 2$ of possible queries that the verifier can ask the ith prover, a finite set A_i of possible answers that it can receive from the ith prover, a finite set R of possible random seeds for the verifier, a polynomial-time computable function f from $\Sigma^n \times R$ to $Q_1 \times Q_2$ (where Σ is the input alphabet) and a polynomial-time computable Boolean predicate Π on $\Sigma^n \times R \times A_1 \times A_2$. A prover P_i for $i = 1, 2$ is simply a function from Q_i to A_i. Given an input x of length n, the verifier chooses uniformly at random a seed r from R and computes a pair of queries $(q_1, q_2) = f(x, r)$ to the provers. After receiving answers $a_1 = P_1(q_1)$ and $a_2 = P_2(q_2)$ from the provers, the verifier evaluates the predicate $\Pi(x, r, a_1, a_2)$ and accepts or rejects accordingly.

Feige and Lovász [12] constructed 2-prover 1-round proof systems for any language in $NEXP$. Their construction can easily be scaled down to a 2-prover 1-round proof for SAT such that

- If $\varphi \in$ SAT then there exists provers such that the verifier accepts always.
- If $\varphi \notin$ SAT then for any pair of provers the verifier accepts with probability at most $1/n$, where n is the size of φ.

Furthermore, the verifier uses $\text{poly} \log n$ random bits and all the messages have length $\text{poly} \log n$; i.e., the sets R, Q_i and A_i, $i = 1, 2$, have cardinality bounded by $2^{\text{poly} \log n}$. For a fixed prover P_i, the verifier's queries to that prover are distributed uniformly at random over the set Q_i of all the possible queries, and given a query to one prover the number of queries that can be asked to the second prover is independent of the query to the first

prover. Furthermore, for a fixed choice of the random seed and a fixed answer from the first prover there is at most one answer from the second prover that makes the verifier accept.

We will need that the verifier asks the provers the same number of queries. The Feige-Lovász proof system does not have this property, but a simple variation yields a proof system that has this property as well as all the other ones mentioned above.

Our construction uses as a basic building block a set system $\mathcal{B}_{m,l} = (B; C_1, C_2, \ldots, C_m)$ with the properties stated in the following lemma. Its proof is omitted.

Lemma 3.1 *Given integers m, l there exists a set B and $C_1, C_2, \ldots, C_m \subset B$ such that for any sequence of indices $1 \leq i_1 < i_2 < \cdots < i_l \leq m$, no collection $D_{i_1}, D_{i_2}, \ldots, D_{i_l}$ covers B where D_{i_j} is C_{i_j} or the complement of C_{i_j}. Furthermore, $|B| = O(2^{2l}m^2)$.*

The basic idea of the construction is as follows. Let φ be a CNF formula. We shall construct an instance $\mathcal{S}_\varphi$ of SETCOVER such that if $\varphi \in$ SAT then there is a cover of size $|Q_1| + |Q_2|$, whereas if $\varphi \notin$ SAT the minimum cover has size at least $c \log N(|Q_1| + |Q_2|)$.

The points in $\mathcal{S}_\varphi$ are all pairs $< r, b >$, where r is a random seed to the verifier and b is a point in the set B from the set system $\mathcal{B}_{m,l}$, where $m = |A_2|$ and l is an even integer that will be determined later. $\mathcal{S}_\varphi$ has a set S_{q_i,a_i} for every pair of possible query q_i and possible answer a_i to q_i for each prover $i = 1, 2$.

For $r \in R$ and $i = 1, 2$ let $q[r, i]$ be the query that the verifier asks the ith prover when using the random seed r. Let $a_2[r, a_1]$ be the answer from the second prover such that the verifier accepts when using random seed r and receiving a_1 as the answer from the first prover. Note that from the properties of the protocol there exist at most one such answer. If no such answer exists then let $a_2[r, a_1]$ be undefined. Now we can define the sets, for every $q_i \in Q_i$ and $a_i \in A_i$:

$$S_{q_1,a_1} = \{< r, b > | q_1 = q[r, 1], \\ a_2[r, a_1] \text{ is defined and } b \notin C_{a_2[r,a_1]}\}$$

and

$$S_{q_2,a_2} = \{< r, b > | q_2 = q[r, 2] \text{ and } b \in C_{a_2}\}.$$

Note that for a fixed r the points $\{< r, b >\}_{b \in B}$ can be covered by any two sets $S_{q[r,1],a_1}$ and $S_{q[r,2],a_2}$, where the verifier accepts if using random seed r and receiving the answers (a_1, a_2). Furthermore any covering that uses less than l sets to cover the points $\{< r, b >\}_{b \in B}$ contains two sets that correspond to answers that make the verifier accept. This follows from the property of $\mathcal{B}_{ml}$. If a covering, that covers all the points, uses at most l sets to cover the points $\{< r, b >\}_{b \in B}$ then the covering contains answers to $q[r, 1]$ and $q[r, 2]$ that make the verifier accept. We can show the following:

Lemma 3.2 *Let* SETCOVER($\mathcal{S}_\varphi$) *be the size of the minimum cover of S_φ.*

- *If $\varphi \in$* SAT *then* SETCOVER($\mathcal{S}_\varphi$) $= |Q_1| + |Q_2|$.
- *If $\varphi \notin$* SAT *then* SETCOVER($\mathcal{S}_\varphi$) $\geq c \log N(|Q_1| + |Q_2|)$, *where $N = |R||B|$ and $0 < c < 1/4$.*

Proof: Omitted in this extended abstract. ∎

Theorem 3.3 *For any $0 < c < 1/4$, the Set Covering problem can not be approximated within factor of $c \log N$ in polynomial time unless $NTIME(n^{\text{poly}\log n}) = DTIME(n^{\text{poly}\log n})$.*

Comments. A randomized variation of our construction shows that for any $0 < c < 1/2$, the Set Covering problem can not be approximated within factor of $c \log N$ in polynomial time unless $NP \subset BPTIME(n^{\text{poly}\log n})$. This gives us a very good lower bound, since the upper bound $\ln n \approx 0.7 \log n$ [17, 21].

The construction has size $\max(|R|, |A_1|, |A_2|)^{O(1)}$ and hence $n^{\text{poly}\log n}$. In order to get a polynomial-time reduction showing nonapproximability of Set Covering within a factor of $\Theta(\log N)$ using our construction we need new probabilistic proof systems for NP for which $\max(|R|, |A_1|, |A_2|)$ is polynomial in n, and that these proof systems have similar properties as the Feige-Lovász proof systems. The two most important of these properties are the constant number of provers and the very low error probability. Recently a step in this direction was taken by [4] where a 4-prover interactive proof system is constructed with parameters $\max(|R|, |A_1|, \ldots, |A_4|) = n^{O(\log\log n)}$, yielding an improvement to the above result. It implies that for any $0 < c < 1/8$, the Set Covering problem can not be approximated within factor of $c \log N$ in polynomial time unless $NP \subset DTIME(n^{O(\log\log n)})$. Their proof systems also imply that approximating the Set Covering problem within any constant factor is NP-hard.

3.2 Related problems

We define some problems that are equivalent to Set Covering.

Hitting Set: Given a (finite) collection $\mathcal{S}$ of subsets of a (finite) set U, find a minimum cardinality subset of U that intersects every set in $\mathcal{S}$.

Hypergraph Transversal: Given a hypergraph $H = (V, E)$, find a minimum cardinality set of nodes

S that covers all the edges of H, i.e., every edge contains at least one member of S.

Dominating Set: Given a (directed or undirected) graph G, find a minimum cardinality set S of nodes that dominates all the nodes of the graph, where a node dominates itself and all its adjacent nodes.

Minimum Exact Cover: Given a (finite) collection $\mathcal{S}$ of subsets of U find a subcollection $\mathcal{S}'$ that covers U and which minimizes the sum of the cardinalities of the sets in $\mathcal{S}'$. (This variant was called Set Covering II in [17].)

All the above problems can be approximated within the same logarithmic factor as the Set Covering problem. Kolaitis and Thakur define syntactically a class of problems that exhibit this logarithmic behavior and which have the Set Covering problem as a complete representative [20].

Theorem 3.4 *The following holds for each of the problems Hitting Set, Hypergraph Transversal, Dominating Set, Minimum Exact Cover. For any* $0 < c < 1/4$, *there is no polynomial time approximation algorithm with ratio* $c \log n$ *unless* $NTIME(n^{\text{poly}\log n}) = DTIME(n^{\text{poly}\log n})$.

References

[1] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. In *Proc. of the 31st IEEE Symp. on Foundations of Computer Science*, pages 544–553, 1990.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. of the 33rd IEEE Symp. on Foundations of Computer Science*, 1992.

[3] S. Arora and S. Safra. Approximating clique is NP-complete. In *Proc. of the 33rd IEEE Symp. on Foundations of Computer Science*, 1992.

[4] M. Bellare, S. Goldwasser, C. Lund and A. Russell. Efficient Probabilistically Checkable Proofs and Applications to Approximation. To appear in *Proc. of the 25th ACM Symp. on the Theory of Computing*, 1993.

[5] M. Bellare. Interactive proofs and Approximation. IBM Research Report RC 17831, 1992.

[6] A. Blum. Some tools for approximate 3-coloring. In *Proc. of the 31st IEEE Symp. on Foundations of Computer Science*, pages 554–562, 1990.

[7] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. In *Proc. 7th Symp. on Theoretical Aspects of Comp. Sci.*, pages 256–267. LNCS 349, 1989.

[8] V. Chvatal. A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4:233–235, 1979.

[9] A. Condon. The Complexity of the Max Word Problem. In *Proc. 8th Symp. on Theoretical Aspects of Comp. Sci.*, pages 456–465, 1991.

[10] G. Dobson. Worst-case analysis of greedy heuristics for integer programming with nonnegative data. *Mathematics of Operations Research*, 7:515–531, 1982.

[11] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proc. of the 32nd IEEE Symp. on Foundations of Computer Science*, pages 2–12, 1991.

[12] U. Feige and L. Lóvasz. Two-prover one-round proof systems: Their power and their problems. In *Proc. of the 24th ACM Symp. on the Theory of Computing*, pages 733–744, 1992.

[13] M. L. Fisher and L. A. Wolsey. On the greedy heuristic for continuous covering and packing problems. *SIAM J. Algebraic Discrete Methods*, 3:584–591, 1982.

[14] M. R. Garey and D. S. Johnson. The complexity of near-optimal graph coloring. *J. of the ACM*, 23:43–49, 1976.

[15] M. Grotschel, L. Lóvasz, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[16] M. M. Halldórsson. A still better performance guarantee for approximate graph coloring. Technical Report 90-44, DIMACS, 1990.

[17] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. of Computer and System Sciences*, 9:256–278, 1974.

[18] D. S. Johnson. Worst case behavior of graph coloring algorithms. In *Proc. 5th Southeastern Conf. on Combinatorics, Graph Theory and Computing*, pages 513–527. Utilitas Mathematica Publ., Winnipeg, Ontario, 1974.

[19] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher,

editors, *Complexity of Computer Computations*, Advances in Computing Research, pages 85–103. Plenum Press, 1972.

[20] P. G. Kolaitis and M. N. Thakur. Approximation properties of NP minimization classes. In *Proc. of the 6th Conference on Structure in Complexity Theory*, pages 353–366, 1991.

[21] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[22] D. Lapidot and A. Shamir. Fully parallelized multi prover protocols for NEXP-time. In *Proc. of the 32nd IEEE Symp. on Foundations of Computer Science*, pages 13–18, 1991.

[23] N. Linial and U. Vazirani. Graph products and chromatic numbers. In *Proc. of the 30th IEEE Symp. on Foundations of Computer Science*, pages 124–133, 1989.

[24] J. Orlin. Contentment in graph theory: covering graphs with cliques. In *Proc. Konik. Neder. Akad. Wet.*, volume 80, pages 406–424, 1977.

[25] A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximations. *Theoretical Computer Science*, 15:251–277, 1981.

[26] C. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. In *Proc. of the 20th ACM Symp. on the Theory of Computing*, pages 229–234, 1988.

[27] H. U. Simon. On approximate solutions for combinatorial optimization problems. *SIAM J. Algebraic Discrete Methods*, 3:294–310, 1990.

[28] A. Wigderson. Improving the performance guarrantee for approximate graph coloring. *J. of the ACM*, 30:729–735, 1983.

[29] L. A. Wolsey. An analysis of the greedy algorithm for the submodulat set covering problem. *Combinatorica*, 2:385–393, 1982.

[30] C. K. Wong L. T. Kou, L. J. Stockmeyer. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Communications of the ACM*, 21:135–139, 1978.