On the Hardness of Approximating MAX k-CUT and its Dual

Viggo Kann[*] Sanjeev Khanna[†] Jens Lagergren[*] Alessandro Panconesi[‡]

November 6, 1995

Abstract. We study the MAX k-CUT problem and its dual, the MIN k-PARTITION problem: given $G = (V, E)$ and $w : E \to R^+$, find an edge set of minimum weight whose removal makes G k-colorable. For the MAX k-CUT problem we show that, if P $\neq$ NP, no polynomial time approximation algorithm can achieve a relative error of $1/\sigma k$, where $\sigma = 132$. This should be compared to the well-known fact that a naive randomized heuristics delivers approximations whose relative error is $1/k$.

For the MIN k-PARTITION problem, we show that for $k > 2$ and for every $\epsilon > 0$, there exists a constant α such that the problem cannot be approximated within $\alpha|V|^{2-\epsilon}$, even for dense graphs.

Both problems are directly related to the frequency allocation problem for cellular (mobile) telephones, an industrial application of increasing relevance.

1 Introduction

A good starting point to motivate the problems studied in this paper is to consider the frequency allocation problem for cellular telephones. We are given a *fixed* set of k frequencies, henceforth referred to as *colors*, and a weighted input graph which intuitively models a network of radio transmitters. In the graph, vertices correspond to transmitters, and an edge (u, v) denotes the fact that if u and v get the same color (*i.e.* frequency) the performance locally deteriorates by $w(u, v)$ per cent. We are to allocate colors so to minimize the total disruption. This problem is becoming increasingly relevant in industrial applications.[1] One way to tackle it is to consider the following combinatorial problem:

MIN k-PARTITION.
Input: a graph $G = (V, E)$, with weighted edges.
Goal: find a color assignment $c : V \to [k]$ that minimizes the total weight of monochromatic edges (an edge is monochromatic if its endpoints have the same color).

*Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden. Email: {jensl,viggo}@nada.kth.se. Supported by grants from NFR and TFR.

†Department of Computer Science, Stanford University, Stanford, CA 94305, USA. Email: sanjeev@theory.stanford.edu. Supported by an OTL grant, NSF Young Investigator Award CCR-9357849, and a Schlumberger Foundation Fellowship.

‡The Swedish Institute of Computer Science, Box 1263, S-164 28 Kista, Sweden. Supported by an ERCIM post-doctoral fellowship. Current address: Facbereich Informatik, Freie Universität, Takustr. 9, 14195 Berlin, Deutschland. Supported by an Alexander von Humboldt fellowship. Email: ale@inf.fu-berlin.de

[1] This formulation is simpler than the real problem. However, in this paper we are interested in lower bounds and therefore, our results apply directly to the real problem.

This problem can be conveniently reformulated as: find an edge set of minimum weight whose removal makes the graph k-colorable. When $k = 2$, Garg, Vazirani and Yannakakis use multicommodity flow techniques to give a polynomial time $O(\log n)$-approximation algorithm, where n denotes the number of vertices in G [9]. The problem is MAX SNP-hard [9]; the best lower bound known for the approximation of MAX CUT translates into a 1.0236 lower bound for MIN 2-PARTITION. Shrinking this gap between upper and lower approximation bounds appears to be a challenging open problem.

When $k \geq 3$, a result of Petrank implies that it is NP-hard to approximate MIN k-PARTITION within $O(n)$ [12]. This result however holds only for "sparse" graphs, *i.e.* graphs with $O(n)$ edges. Petrank left the approximability of MIN k-PARTITION for non-sparse graphs as an open question. Edwards [7] and Arora *et al.* [1] have shown that 3-colorability is polynomially solvable for dense graphs where each vertex in the graph has degree at least pn for some constant p.

In this paper we fill in this picture by showing that, for $k \geq 3$: (a) for any fixed $\epsilon > 0$, it is NP-hard to approximate MIN k-PARTITION within $O(n^{2-\epsilon})$, even when restricting to dense graphs with $|E| = \delta n^2$ for any $\delta < (1 - 1/k)/2$. If $\delta > (1 - 1/k)/2$ the graph is certainly not k-colorable by Turán's theorem, and the whole graph is trivially a constant approximation; (b) it is NP-hard to approximate MIN k-PARTITION within $O(|E|)$, even when restricting to graphs with $|E| = \Omega(n^{2-\epsilon})$; (c) MIN 3-PARTITION can be approximated within ϵn^2, for any $\epsilon > 0$. This completes the picture as far as densities are concerned and answers Petrank's open question completely.

In view of the above, it seems natural to consider the dual problem:

MAX k-CUT.
Input: a graph $G = (V, E)$, with weighted edges.
Goal: find a color assignment $c : V \rightarrow [k]$ that maximizes the total weight of properly colored edges (an edge is properly colored if its endpoints have different colors).

A lot of interesting theoretical work has focused on this problem, especially on the special case with $k = 2$, known as the MAX CUT problem (see for example [10, 14]). In a seminal paper, Papadimitriou and Yannakakis showed that the problem is MAX SNP-complete (in their paper the problem is called MAX k-COLORABLE SUBGRAPH) [11]. Recent results in the theory of probabilistic checking of proofs provide new powerful tools for proving hardness results for approximation, under suitable complexity theoretic assumptions [2]. The best known lower bound for the relative error of MAX CUT is 0.015, under the assumption P $\neq$ NP [3]. For MAX k-CUT it is well known that a naive randomized heuristic achieves a relative error of $1/k$: each vertex picks a color uniformly at random, independently of other vertices. This simple procedure can be derandomized by a straightforward application of the *method of conditional probabilities* (see for example [15]). It is natural to ask whether better approximations are possible, for example, $1/k^{1+\epsilon}$ for some $\epsilon > 0$. For the case $k = 2$, the heuristics gives the relative error 0.5, a bound that held for about two decades until the recent ground breaking work of Goemans and Williamson [10]. Their techniques – a beautiful blend of mathematical programming and probabilistic techniques – yield a relative error of 0.122 [10]. More recently, Frieze and Jerrum have tried to generalize the Goemans and Williamson technique to the MAX k-CUT problem [8]. Their results are quite interesting from a technical point of view, but yield only a very marginal improvement, from $1/k$ to (roughly) $1/k - 2k^{-2} \ln k$. In this paper, we show that there might be intrinsic, insurmountable reasons for this disappointing behavior. We show that, if P $\neq$ NP, the true value for approximating MAX k-CUT lies between $1/k$ and $1/\sigma k$, where currently $\sigma = 132$ (the value of σ depends on the best lower bound available on the approximation of MAX CUT). Hence, unless P = NP, we cannot hope for relative errors like $1/k^{1+\epsilon}$ or even, say, $1/(k \log\log \ldots \log k)$. On the other hand, the naive randomized heuristics comes very close to the best possible approximation (assuming P $\neq$ NP); may be this is because it exploits some fundamental combinatorial properties of MAX k-CUT. Can this shed some light on the combinatorial structure of MAX k-CUT and similar problems?

A few words on the techniques used. The MAX k-CUT result is obtained by making use of an approximation preserving reduction from MAX CUT. The difficulty of the proof is to exhibit an approximation preserving reduction that grows only linearly in k. The proof uses *the probabilistic method* and a construction involving the Hamming distance between characteristic vectors. Our reduction to MAX k-CUT increases the relative error with about $2k$, which can be compared with the original reduction in [11] which increases the relative error with more than $1000k^2$. The MIN k-PARTITION lower bound of $\Omega(n^{2-\epsilon})$ is obtained by amplifying the hardness of k-coloring into a zero versus $\Omega(n^{2-\epsilon})$ separation. The ϵn^2-approximation algorithm is based on the fact that 3-colorability can be solved in polynomial time for dense instances [7, 1].

We emphasize that our hardness results do not depend on the existence of exponentially large weights in the input instances, in fact they hold for the unweighted case too (see also [5]).

This paper is organized as follows. Section 2 defines the notion of approximation preserving reductions used in this paper. In Section 3, we present the hardness result for MAX k-CUT and a precise lower bound computation using the currently best-known lower bound for MAX 3SAT. Finally, in Section 4, we present our upper and lower bounds for approximating MIN k-PARTITION.

2 Definitions

Solving an optimization problem F given the input instance x means finding a solution y such that the value of the objective function $m_F(x,y)$ is optimum. Let $opt_F(x)$ denote this optimum value of m_F.

Approximating a optimization problem F given the input x means finding some feasible solution y. How good the approximation is depends on the relation between $m_F(x,y)$ and $opt_F(x)$. There are two equivalent measures of this relation: the performance ratio and the relative error. The *performance ratio* of a solution y with respect to the optimum of a optimization problem F is defined as

$$R_F(x,y) = \max\left\{\frac{opt_F(x)}{m_F(x,y)}, \frac{m_F(x,y)}{opt_F(x)}\right\}.$$

The *relative error* is defined as

$$\mathcal{E}_F(x,y) = \frac{|opt_F(x) - m_F(x,y)|}{opt_F(x)}.$$

An optimization problem F can be *approximated within* $f(n)$ for a function f if there exists a polynomial time algorithm A such that for all input instances x, $A(x)$ is a feasible solution and $R_F(x, A(x)) \leq f(|x|)$. $A(x)$ is a $f(n)$-*approximation algorithm.* If the function $f(n)$ is constant we say that A is *approximable.*

Although various reductions preserving approximability within constants have been proposed (see [4]), the L-reduction defined in [11] is perhaps the easiest to use. Given two NP optimization problems F and G and a polynomial time transformation f from instances of F to instances of G, we say that f is an *L-reduction* if there are positive constants α and β such that for every instance x of F

i) $opt_G(f(x)) \leq \alpha \cdot opt_F(x)$,

ii) for every feasible solution y of $f(x)$ with objective value $m_G(f(x), y) = c_2$ we can in polynomial time find a solution y' of x with $m_F(x, y') = c_1$ such that $|opt_F(x) - c_1| \leq \beta\, |opt_G(f(x)) - c_2|$.

The composition of L-reductions is an L-reduction. If F L-reduces to G with constants α and β and there is a polynomial time approximation algorithm for G with worst-case relative error ϵ, then there is a polynomial time approximation algorithm for F with worst-case relative error $\alpha\beta\epsilon$ [11]. Conversely, if δ is a lower bound for the approximation of F then, $\delta/(\alpha\beta)$ is a lower bound for G. An L-reduction from F to G therefore can be used both to show hardness of approximability (for G) and to find a new approximation algorithm (for F).

3 Hardness of Approximating Max k-Cut

In this section, we give an L-reduction from Max Cut instance G to a Max k-Cut instance H such that

- $opt_{\text{Max } k\text{-Cut}}(H) \leq k(k-1) opt_{\text{Max Cut}}(G)$, *i.e.* $\alpha = k(k-1)$,
- given any Max k-Cut solution h we can find in polynomial time a Max Cut solution g satisfying $|opt_{\text{Max Cut}}(G) - g| \leq \frac{k}{2}|opt_{\text{Max } k\text{-Cut}}(H) - h|$, *i.e.* $\beta = 2/k$.

This reduction together with a lower bound of the approximability of Max Cut gives the following theorem.

Theorem 1 *There is a constant σ between 1 and 132 such that it is NP-hard to approximate* Max k-Cut *within* $1 + \frac{1}{\sigma(k-1)}$.

Proof: A lower bound of δ for the relative error of Max Cut will, using the L-reduction, give us a lower bound of $\frac{\delta}{\alpha\beta} = \frac{\delta}{2(k-1)}$ for Max k-Cut. Bellare, Goldreich and Sudan [3] have shown that it is NP-hard to approximate Max Cut within 66/65, which gives us the lower bound 1/66 for the relative error, and the theorem follows. Better lower bounds for Max Cut will automatically give better lower bounds for Max k-Cut. It is also possible to get a better value of α by taking advantage of the specific structure of the used Max Cut instances.

□

Remark: In this abstract, we reduce an unweighted graph G to a weighted graph H. The reduction is easy to generalize to the case of weighted G. Also, we assume that k is an even number; the case k is odd can easily be reduced to the former case.

The graph H is defined as follows. Assume that $V(G) = \{v_1, \ldots, v_n\}$. Let $G_1, \ldots, G_{k/2}$ be graphs isomorphic to G such that $V(G_i) = \{v_1^i, \ldots, v_n^i\}$ and for each $1 \leq i \leq k/2$ there is an isomorphism from G to G_i mapping v_l to v_l^i, for each $1 \leq l \leq n$. Let H be the graph obtained by first taking the union of $G_1, \ldots, G_{k/2}$. Besides the edges inside each copy we add edges as follows: if (v_l, v_m) is an edge of G we add all the edges in $\{(v_l^i, v_m^j) : 1 \leq i < j \leq k/2\}$. We also add edges between all copies of the same vertex, *i.e.* $\{(v_l^i, v_l^j) : 1 \leq i < j \leq k/2\}$. We will refer to the latter as the *row edges* and the set of copies of v_l will be referred to as the *row of* v_l. Notice that the row of a vertex v_l induces a clique in H which we call the *row clique of* v_l. The non row edges will be called the *cross edges*. Observe that if we take to different rows, say, the row of u and the row of v such that (u, v) is an edge of G, then the cross edges between these rows induce a complete bipartite graph. Otherwise there are no cross edges. Finally we add weights; all cross edges have weight 1 whereas all edges in the row of v_l have weight equal to $d_G(v_l)$.

Lemma 2 $opt_{\text{MAX } k\text{-CUT}}(H) \le k(k-1) opt_{\text{MAX CUT}}(G)$.

Proof: First we prove that $|E(H)| = \frac{k(k-1)}{2}|E(G)|$. Notice that in H there are $k/2$ copies of each vertex u of G and each such copy has degree $\frac{k}{2}d_G(u) + (\frac{k}{2}-1)d_G(u)$. Hence

$$|E(H)| = \frac{1}{2}\sum_{x\in V(H)} d_H(x) = \frac{k}{4}\sum_{u\in V(G)} \frac{k}{2}d_G(u) + (\frac{k}{2}-1)d_G(u) = \frac{k(k-1)}{2}|E(G)|.$$

Clearly the graph H can be constructed in polynomial time. Since $opt_{\text{MAX CUT}}(G) \ge |E(G)|/2$ and $opt_{\text{MAX } k\text{-CUT}}(H) \le |E(H)|$ for any graphs G and H, the claim follows. □

We now prove that the other requirement of the L-reduction is also satisfied. A coloring of H is said to be in *normal form* if it satisfies the following: (i) each copy of G is bichromatic (ii) no two copies share a color (iii) all copies have exactly the same coloring up to renaming of the colors. Our aim now is to show the following.

Lemma 3 *Given a k-coloring c of H, we can always find in polynomial time another k-coloring c' such that c' is in normal form and induces a cut which is no worse than that induced by c.*

For any graph F and any coloring c of F, we define $cut(c, F)$ to be the the cut induced by c, that is, the set of bichromatic edges. Once the above lemma is established, it follows that the second requirement of the L-reduction is satisfied. We will now show why. Given a coloring of G we can find a coloring of H in normal form; the above lemma effectively means that all k-colorings of H are of this form, including the optimum colorings. If c' is in normal form, it induces a cut whose size can be written

$$|cut(c', H)| = W + \frac{k}{2}|cut(g, G)|$$

where g is the coloring appearing in each copy, and W is the the total sum of the weights of all edges connecting vertices in *different* copies of G. In particular, the optimum coloring in H can be assumed to have this form and value equal to $W + opt_{\text{MAX CUT}}(G)k/2$. This gives that,

$$|opt_{\text{MAX } k\text{-CUT}}(H) - cut(c, H)| \ge |opt_{\text{MAX } k\text{-CUT}}(H) - cut(c', H)| = \frac{k}{2}|opt_{\text{MAX CUT}}(G) - cut(g, G)|$$

and it follows that the second requirement of an L-reduction is satisfied.

The first step in proving Lemma 3 is to show that we may assume *w.l.o.g.* that each row-clique is properly colored, *i.e.* it is colored with $k/2$ colors.

Lemma 4 *For each k-coloring c_0 of H there is a k-coloring c of H such that $|cut(c, H)| \ge |cut(c_0, H)|$ and each row edge is bichromatic. Moreover given c_0 and H the k-coloring c can be obtained in polynomial time.*

Proof: Assume that x and y are two copies in H of the same vertex u of G such that $c_0(x) = c_0(y) = p$. For each color p' which is not used in the row of u, let $A_{p'}$ be the set of vertices z adjacent to x in H such that $c_0(z) = p'$ and (x, z) is a cross edge. Since there are more than $k/2$ such colors p' and the number of cross edges incident to x is $\frac{k}{2}d_G(u)$, it follows by a standard average argument that there is a p' such that $|A_{p'}| \le d_G(u)$. If we recolor x with p' we create at most $|A_{p'}|$ monochromatic edges of weight 1, but get rid of one monochromatic (row) edge of weight $d_G(u)$. Thus, by recoloring x with p' we obtain a coloring which induces a cut at least as good as before. This process can be repeated until, in polynomially many steps, we have a new coloring satisfying the stated properties. □

Now fix a k-coloring c of H; by the previous lemma we may assume that every row edge is bichromatic *w.r.t.* c. For each vertex u of G, let $\chi(u)$ be the vector in $\{0,1\}^k$ defined by $\chi(u)_p = 1$ *iff* the color p is used in the row of u in H. Notice, by the choice of c each vector $\chi(u)$ has a support of cardinality $k/2$. We define h to be the Hamming distance.

Given c we will construct a coloring c' in normal form. The crux of the argument is to show that the number of monochromatic cross edges of c' is no more than that of c and hence that the cut induced by c' is at least as good as that induced by c (being normal, c' properly colors all row edges). The coloring c' will be defined by means of the vectors $\chi(u)$. First we want to count the number of monochromatic edges induced by c as a function of the hamming distance between the $\chi(u)$ vectors.

Lemma 5 *Let u and v be two vertices of G such that (u,v) is an edge of G. The number of monochromatic cross edges between the row of u and the row of v is equal to $(k - h(\chi(u),\chi(v)))/2$.*

Proof: Let P be the set of colors that appear in both rows. We know that $k/2$ colors are used in each row. Since moreover $\chi(u)$ and $\chi(v)$ both have support of cardinality $k/2$ it follows that

$$|P| = \frac{k - h(\chi(u),\chi(v))}{2}.$$

Since each color has at most one occurrence per row and the cross edges between the two rows induce a complete bipartite graph, $|P|$ equals the number of monochromatic edges. □

We now use the vectors to find a good 2-coloring of G which can be duplicated in each copy, using new colors in each copy. The proof uses the probabilistic method. Chose i uniformly at random between 1 and k, let r be the 2-coloring of G defined by $r(v) = \chi(v)_i$. It follows immediately that for any two vertices u and v of G we have $\Pr[r(u) = r(v)] = \frac{k-h(\chi(v_l),\chi(v_m))}{k}$. We also get the following lemma.

Lemma 6

$$E[(\#\textit{monochromatic edges of } G \text{ w.r.t. } r)] = \frac{2}{k}(\#\textit{monochromatic edges of } H \text{ w.r.t. } c)$$

Proof: First notice

$$\begin{aligned}
&(\#\text{monochromatic edges of } H \ \textit{w.r.t.}\ c) = \\
&\quad = \sum_{(u,v)\in E(G)} (\#\text{monochromatic cross edges between the row of } u \text{ and the row of } v) \\
&\quad = \frac{1}{2}\sum_{(u,v)\in E(G)} (k - h(\chi(u),\chi(v))),
\end{aligned}$$

where the last inequality follows from Lemma 5. From this we get

$$\begin{aligned}
&\mathrm{E}[(\#\text{monochromatic edges of } G \ \textit{w.r.t.}\ r)] = \\
&\quad = \sum_{(u,v)\in E(G)} \Pr[r(u) = r(v)] \\
&\quad = \sum_{(u,v)\in E(G)} \frac{k - h(\chi(u),\chi(v))}{k} \\
&\quad = \frac{2}{k}(\#\text{monochromatic edges of } H \ \textit{w.r.t.}\ c).
\end{aligned}$$

□

By cycling through all k colors we are sure to find a 2-coloring g of G whose number of monochromatic edges is at least the expected value. To obtain a c' in normal form just color each copy according to g, each time using new colors. The whole process can be carried out in polynomial time.

4 Hardness of Approximating Min k-Partition

Let $\gamma(G,k)$ be the minimum number of edges that must be deleted from G in order to obtain a k-partite graph. We now show that for any fixed $k > 2$, it is NP-hard to approximate Min k-Partition to within a factor of $O(n^{2-\epsilon})$ for any $\epsilon > 0$. An interesting aspect of this result is that it holds even when we restrict ourselves to graphs with $\Omega(n^{2-\epsilon})$ edges.

Definition 1 [Dense and Everywhere Dense Graphs] *An n-vertex graph is called* dense *if it has $\Omega(n^2)$ edges and it is called (everywhere) ϵ-dense if each vertex in the graph has degree at least ϵn for some fixed $\epsilon > 0$.*

We begin with the following observation.

Observation 7 *For any $k \geq 3$, it is NP-hard to decide if a graph is k-colorable over graphs with $\Theta(n^\alpha)$ edges, where $0 < \alpha \leq 2$.*

Proof: It suffices to show that the result holds on the family of n-vertex graphs with $\Omega(n^2)$ edges; the density can be scaled down to $\Theta(n^\alpha)$ simply by adding a disjoint empty graph on $\Theta(n^{\frac{2}{\alpha}})$ vertices. Clearly, this transformation leaves the chromatic number unchanged.

We show only the hardness of 3-coloring on dense graphs, for $k \geq 4$, it follows from a result of Edwards [7]. Given a graph G on n vertices, we construct another graph G' which is simply a disjoint union of G and a complete 3-partite graph H on N vertices such that each partite set is of the same size. If we choose $N = n^2$, for instance, then clearly the graph G' is dense. Furthermore, it is immediate from the construction that G' is 3-colorable iff G is 3-colorable. Hence it is NP-hard to decide if a dense graph is 3-colorable. □

We next establish a lemma which will allow us to amplify the NP-hardness of deciding between $\gamma(G,k) = 0$ and $\gamma(G,k) \geq 1$ to that of distinguishing between $\gamma(H,k) = 0$ and $\gamma(H,k) = \Omega(n^{2-\epsilon})$. This yields the desired hardness result. The precise statement is as follows.

Lemma 8 [Amplification Lemma] *For any positive integers k and s, given a graph G one can in polynomial time construct a graph H such that $|V(H)| = s|V(G)|$, $|E(H)| = s^2|E(G)|$, and $\gamma(H,k) = s^2\gamma(G,k)$.*

The graph H in the Amplification Lemma is constructed as follows:

- for each vertex u in G, the set $V(H)$ contains s copies named $u^1, \ldots, u^s$.
- (u^i, v^j) is an edge in H iff (u,v) is an edge of G.

Notice that the set of copies of any vertex u, that is, $u^1, \ldots u^s$ is an independent set of H'. As easily seen $|V(H')| = s|V(H)|$, $|E(H')| = s^2|E(H)|$, and the construction can be done in polynomial time.

Definition 2 [Semi-coloring] *A coloring of a graph G is called an $\langle m \rangle$ semi-coloring if it induces at most m monochromatic edges.*

We say that an $\langle m \rangle$ semi-coloring of H is *normalized* iff for each $u \in V(G)$, all the s copies of u in H, namely $u^1, \dots, u^s$, have the same color.

Lemma 9 *Any $\langle m \rangle$ semi-coloring of H can be transformed into a normalized $\langle m \rangle$ semi-coloring in polynomial time.*

Proof: Let c be a given $\langle m \rangle$ semi-coloring. Consider any vertex u in G and let u_{min} be the copy of u in H with the smallest number of monochromatic edges incident on it under the coloring c. Since the u^i's have the same neighbors, the coloring c' which assigns the color of the vertex u_{min} to each copy u^i, and agrees with coloring c everywhere else, is still an $\langle m \rangle$ semi-coloring. Iterating this process over each vertex u in G, we get a normalized $\langle m \rangle$ semi-coloring for H. □

Lemma 10 *The graph H constructed above satisfies $\gamma(H, k) = s^2\gamma(G, k)$*

Proof: Any $\langle m \rangle$ semi-coloring of G can be transformed into an $\langle s^2 m \rangle$ semi-coloring for H by simply assigning for each vertex u in G, its color to all the copies of u in H. Thus $\gamma(H, k) \leq s^2\gamma(G, k)$.

On the other hand, an $\langle m \rangle$ semi-coloring for H implies an $\langle \lfloor m/s^2 \rfloor \rangle$ semi-coloring for G. By Lemma 9, we can assume without loss of generality that we are given a normalized $\langle m \rangle$ semi-coloring, say c_H, of H. Construct a coloring c_G for G by assigning each vertex u the same color as assigned to each of its copies in H by the coloring c_H. By our construction of the graph H, it is easy to see that each monochromatic edge in G corresponds to a unique set of s^2 monochromatic edges in H. We conclude that $\gamma(H, k) \leq s^2\gamma(G, k)$. The lemma follows. □

Hence we have the following theorem.

Theorem 11 *For any fixed $\epsilon > 0$ it is NP-hard to approximate $\gamma(H, k)$ within $O(n^{2-\epsilon})$ over graphs with $\Theta(n^\alpha)$ edges, for any $(2 - \epsilon) < \alpha \leq 2$.*

Proof: Observation 7 shows that it is NP-hard to distinguish between $\gamma(G, k) = 0$ and $\gamma(G, k) \geq 1$ even when restricted to a family of graphs with $\Theta(n^\alpha)$ edges where $0 < \alpha \leq 2$.

To show the desired result, we start with the family of graphs with $\Theta(n^{\alpha+(\alpha-2)(2-\epsilon)/\epsilon})$ edges. Given a graph G in this family, we construct the graph H using the Amplification Lemma where we choose $s = n^{\frac{2}{\epsilon}-1}$. Clearly, $\gamma(H, k) = 0$ iff $\gamma(G, k) = 0$ and is $\Omega(N^{2-\epsilon})$ otherwise where N is the number of vertices in H. This completes the proof. □

Remark: In fact we can show that it is NP-hard to approximate MIN k-PARTITION within $O(n^{2-\epsilon})$, even when restricting to dense graphs with $|E| = \delta n^2$ for any $\delta < (1 - 1/k)/2$. This is a sharp bound, since if $\delta > (1 - 1/k)/2$ Turán's theorem says that the graph cannot be k-colorable, and the whole graph is trivially a $1 + 2\delta/(1 - 1/k)$-approximation.

The following lemma shows that the hardness result established above is almost tight when $k = 3$ as shown by the following lemma.

Lemma 12 *$\gamma(G, 3)$ can be approximated within ϵn^2 for any fixed $\epsilon > 0$.*

Proof: Let $p = 1/\epsilon$. If $\gamma(G,3) > p$, then we are done (as discussed in Section 2, we take $\max(\gamma(G,3),1)$ to compute the performance ratio). So we assume that $\gamma(G,3) \leq p$. Delete from G any vertices of degree at most $(\epsilon/2)n$. We delete at most $(\epsilon/2)n^2$ edges in this process. Now we are left with a graph G' which satisfies the property that each vertex has degree at least $(\epsilon/2)n$. In G', for every subset of at most p edges, we try deleting it and try 3-coloring the remaining graph. Assuming that n is large enough, the resulting graph is still ϵ'-dense for some $\epsilon' < \epsilon$. Hence if the resulting graph is 3-colorable, we can indeed verify so by using the coloring algorithm of Edwards [7] or Arora *et al.* [1]. The result follows. □

Petrank showed in [12] that, for graphs with $\Theta(n)$ edges, it is NP-hard to approximate $\gamma(H,k)$ to within a factor of $O(|E(H)|)$ (whenever $k \geq 3$). We can use the Amplification Lemma to strengthen this result and obtain the following.

Theorem 13 *For any $k > 2$, it is NP-hard to approximate $\gamma(H,k)$ within $O(|E(H)|)$ over the graphs with $\Omega(n^{2-\epsilon})$ edges.*

Proof: Petrank showed in [12] the following. For any $k \geq 3$, there is some constant c such that it is NP-hard to tell whether a given graph G, from a family $\mathcal{F}$ of graphs with $\Theta(n)$ edges, satisfies $\gamma(G,k) = 0$ or $\gamma(G,k) \geq cn$. Applying the Amplification Lemma with $s = n^{(1-\epsilon)/\epsilon}$ to a graph G of $\mathcal{F}$ gives a graph H such that: $|V(H)| = sn$, $|E(H)| = c's^2n$, $\gamma(G,k) = 0$ iff $\gamma(H,k) = 0$, and $\gamma(G,k) \geq cn$ iff $\gamma(H,k) = s^2cn$. That is, it is NP-hard to approximate $\gamma(H,k)$ within $O(|E(H)|)$. Since $sn = n^{1/\epsilon}$ and $c's^2n = c'n^{(2-\epsilon)/\epsilon}$, the result follows. □

Acknowledgements

We thank Naveen Garg and Johan Håstad for several useful discussions.

References

[1] S. Arora, D. Karger, and M. Karpinki, Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proc. Twenty Seventh Ann. ACM Symp. on Theory of Comp.*, pages 284–293. ACM, 1995.

[2] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability – towards tight results. In *Proc. 36th Ann. IEEE Symp. on Foundation of Comput. Sci.*, pages 422–431. IEEE Computer Society, 1995.

[3] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability – towards tight results. Technical Report ECCC TR95-24, Revised version, September 1995.

[4] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. In *Proc. of 1st Ann. Int. Conf. on Computing and Combinatorics*, Lecture Notes in Comput. Sci. 959, pages 539–548. Springer-Verlag, 1995.

[5] P. Crescenzi, R. Silvestri, and L. Trevisan. To weight or not to weight: Where is the question? Manuscript, 1995.

[6] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, and M. Yannakakis, The complexity of multiway cuts. In *Proc. 24th Ann. ACM Symp. on Theory of Comp. (STOC 92)*, pages 241–251. ACM, 1992.

[7] K. Edwards, The complexity of colouring problems, Theoretical Comput. Sci., 43:337–343, 1986.

[8] A. Frieze and M. Jerrum, Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In *Proc. of 4th Int. Conf. on Integer Prog. and Combinatorial Optimization*, Lecture Notes in Comput. Sci. 920, pages 1–13. Springer-Verlag, 1995.

[9] N. Garg, V. V. Vazirani, and M. Yannakakis, Approximate max-flow min-(multi)cut theorems and their applications. In *Proc. 25th Ann. ACM Symp. on Theory of Comp. (STOC 93)*, pages 698–707. ACM 1993.

[10] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. In *Proc. 26th Ann. ACM Symp. on Theory of Comp. (STOC 94)*, pages 241–251. ACM, 1994.

[11] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. J. Comput. System Sci., 43:425–440, 1991.

[12] E. Petrank. The hardness of approximation: gap location. Computational Complexity, 4:133–157, 1994.

[13] S. Phillips and S. Safra. PCP and tighter bounds for approximating MAX SNP. Manuscript, April 1992.

[14] S. Poljak and Z. Tuza, The max-cut problem – a survey. In L.Lovasz and P.Seymour editors, *Special Year on Combinatorial Optimization*, DIMACS series in Discrete Mathematics and Theoretical Comput. Sci. American Mathematical Society, 1993 (to be published).

[15] P. Raghavan, Lecture notes on randomized algorithms, Tech. report RC 15340 (#68237), IBM T. J. Watson Research Center, January 1990.