Chicago Journal of Theoretical Computer Science

The MIT Press

Volume 1997, Article 2

3 June 1997

ISSN 1073–0486. MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142-1493 USA; (617)253-2889; *journals-orders@mit.edu, journals-info@mit.edu*. Published one article at a time in LaTeX source form on the Internet. Pagination varies from copy to copy. For more information and other articles see:

- *http://www-mitpress.mit.edu/jrnls-catalog/chicago.html*
- *http://www.cs.uchicago.edu/publications/cjtcs/*
- *ftp://mitpress.mit.edu/pub/CJTCS*
- *ftp://cs.uchicago.edu/pub/publications/cjtcs*

The *Chicago Journal of Theoretical Computer Science* is abstracted or indexed in *Research Alert,*® *SciSearch,*® *Current Contents*®*/Engineering Computing & Technology*, and *CompuMath Citation Index.*®

The *Chicago Journal of Theoretical Computer Science* is a peer-reviewed scholarly journal in theoretical computer science. The journal is committed to providing a forum for significant results on theoretical aspects of all topics in computer science.

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

Viggo Kann   Sanjeev Khanna   Jens Lagergren
Alessandro Panconesi

3 June, 1997

Abstract

Abstract-1 We study the MAX k-CUT problem and its dual, the MIN k-PARTITION problem. In the MIN k-PARTITION problem, given a graph $G = (V, E)$ and positive edge weights, we want to 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 better than $1/(34k)$. It is well known that a relative error of $1/k$ is obtained by a naive randomized heuristic.

Abstract-2 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 application of industrial relevance.

1 Introduction

1-1 To motivate the problems studied in this paper, consider the frequency allocation problem for cellular telephones. We are given a *fixed* set of k frequencies, and a set of radio transmitters. Moreover, for each pair of transmitters, we are given a number that represents the percentage of local performance deterioration incurred if the pair receives the same frequency. The problem is to allocate frequencies to radio transmitters so that the global performance deterioration is minimized. It is indeed natural to model this problem by a coloring problem for edge-weighted graphs. Vertices of the graph correspond to transmitters, and colors correspond to frequencies. Edges indicate nonzero local performance deterioration, and their weights indicate the percentages of local performance deterioration. Formally, the graph problem can be stated as follows:

Min *k*-Partition

Input: A graph $G = (V, E)$, with weighted edges.

Goal: Find a color assignment $c\colon V \to [k]$ that minimizes the total weight of monochromatic edges (an edge is monochromatic if its endpoints have the same color).

In other words, we seek an edge set of minimum weight whose removal makes the graph k-colorable. This problem is relevant in industrial applications.[1]

1-2 For $k = 2$, Garg, Vazirani, and Yannakakis [GVY96] use multicommodity flow techniques to give a polynomial time $O(\log n)$-approximation algorithm, where n denotes the number of vertices in the input graph G. The problem is MAX SNP-hard; the best lower bound known for the approximation of Max Cut (i.e., Max k-Cut with $k = 2$) translates into a 1.058 lower bound for Min 2-Partition. Shrinking this gap between upper and lower approximation bounds appears to be a challenging open problem.

1-3 When $k \geq 3$, a result of Petrank implies that it is NP-hard to approximate Min k-Partition within $O(n)$ [Pet94]. This result, however, holds only for "sparse" graphs, that is, graphs with $O(n)$ edges. Petrank left the approximability of Min k-Partition for nonsparse graphs as an open question. Edwards [Edw86] and Arora, Karger, and Karpinski [AKK95] have shown that 3-colorability is polynomial time solvable on graphs where each vertex has degree at least ϵn for some constant $\epsilon > 0$.

1-4 In this paper, we continue this line of research by showing that, for $k \geq 3$:

- For every fixed $\epsilon > 0$ and every $\delta < (1 - 1/k)/2$, it is NP-hard to approximate Min k-Partition within $O(n^{2-\epsilon})$, even when the problem is restricted to dense graphs with $|E| = \delta n^2$. If $\delta > (1 - 1/k)/2$, the graph is certainly not k-colorable by Turán's theorem [Tur41] (see

[1] Many problems that appear in applications have the problem formulated here as a special case. However, in this paper we are interested in lower bounds, and therefore our results apply directly to these problems.

a standard graph theory text such as [Wes96], for instance), and the whole graph is trivially a constant factor approximation.

- It is NP-hard to approximate Min k-Partition within $O(|E|)$, even when restricting to graphs with $|E| = \Omega(n^{2-\epsilon})$.
- Min 3-Partition can be approximated within ϵn^2, for every $\epsilon > 0$. This completes the picture as far as the densities are concerned, and answers completely the open question of Petrank.

1-5 In view of the above two negative results, 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 $c\colon V \to [k]$ that maximizes the total weight of properly colored edges (an edge is properly colored if its endpoints have different colors).

There are many interesting theoretical results on this problem. In particular, the Max Cut problem has received much attention (see [GW95, PT95], for example). In a seminal paper, Papadimitriou and Yannakakis [PY91] showed that the problem is MAX SNP-complete. Recent results in the theory of probabilistic checking of proofs provide powerful new tools for proving hardness of approximation results, under suitable complexity theoretic assumptions (see [BGS95] for an overview). The best-known lower bound on the relative error achievable for Max Cut is 0.0588, under the assumption P $\neq$ NP [Hås97]. For Max k-Cut it is well known that a naive randomized heuristic achieves a relative error of $1/k$: Each vertex is assigned one of the k colors uniformly at random. This simple procedure can be derandomized by a straightforward application of the *method of conditional probabilities* (see [MR95], for example).

1-6 For the case $k = 2$, the above heuristic algorithm has a relative error bound of 0.5, a bound that held for about two decades until the recent pioneering work of Goemans and Williamson [GW95]. Their techniques—a beautiful blend of mathematical programming and probabilistic methods—yield a relative error bound of 0.122 [GW95]. More recently, Frieze and

Jerrum [FJ95] have tried to generalize the Goemans and Williamson technique to the Max k-Cut problem. Their results are quite interesting from a technical point of view but yield only a marginal improvement—from $1/k$ to (roughly) $1/k - 2k^{-2}\ln k$. It is natural to ask whether better approximations are possible, say, a relative error of $1/k^{1+\epsilon}$, for some $\epsilon > 0$. In this paper, we give (under the assumption $\mathrm{P} \neq \mathrm{NP}$) a negative answer to this question. We show that if $\mathrm{P} \neq \mathrm{NP}$, for each $k \geq 2$ the true relative error bound for approximating Max k-Cut lies between $1/k$ and $1/\sigma k$, where currently $\sigma = 34$ (the value of σ depends on the best lower bound available on the approximation of Max Cut). Hence, unless $\mathrm{P} = \mathrm{NP}$, we cannot hope for relative error bounds like $1/k^{1+\epsilon}$ or even, say, $1/(k \log\log \ldots \log k)$. On the other hand, the naive randomized heuristics achieve a relative error that is very close to the best possible (assuming $\mathrm{P} \neq \mathrm{NP}$).

1-7 A few words on the techniques used in this paper. The Max k-Cut result is obtained by making use of an approximation preserving reduction from Max Cut. The difficulty in the proof is to exhibit a reduction that increases the relative error only linearly as a function of 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 by approximately a factor of $2k$. In contrast, the original reduction in [PY91] increases the relative error by a factor greater than $1000k^2$. The Min k-Partition lower bound of $\Omega(n^{2-\epsilon})$ is obtained by amplifying the NP-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 [Edw86, AKK95].

1-8 We emphasize that our hardness results do not depend on the existence of exponentially large weights in the input instances; in fact, they also hold for the unweighted case (see also [CST95]).

1-9 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 Cut. Finally, in Section 4, we present our upper and lower bounds for approximating Min k-Partition.

2 Definitions

2-1 We will use the standard approximation terminology. See for example [CK95] for detailed definitions.

2-2 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 equal to the optimum (maximum or minimum). Let $opt_F(x)$ denote this optimum value of m_F.

2-3 Approximating an 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 to an 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

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

2-4 These definitions are only meaningful when $opt_F(x) \neq 0$. This is a problem for Min k-Partition when the input graph is k-colorable. To make the definition robust and to simplify the statement of our results, we define the optimum value for Min k-Partition as $\max\{1, \min_c m(G, c)\}$ where $G = (V, E)$ is the input graph, c ranges over all possible color assignments $c\colon V \to \{1, \ldots, k\}$, and $m(G, c)$ measures the number of monochromatic edges under c.

2-5 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 $R_F(x, A(x)) \leq f(|x|)$. The algorithm $A(x)$ is called an $f(n)$*-approximation algorithm*.

2-6 Although various reductions preserving approximability within constants have been proposed (see [CKST95]), the L-reduction defined in [PY91] is perhaps the easiest to use. Given two NP optimization problems F and G, an *L-reduction* from F to G consists of two polynomial-time computable functions f and g and two positive constants α and β, such that f transforms instances of F into instances of G, and such that, for every instance x of F,

- $opt_G(f(x)) \leq \alpha \cdot opt_F(x)$, and
- for every feasible solution y of $f(x)$ with objective value $m_G(f(x), y) = s_2$, $y' = g(x, y)$ is a feasible solution of x with $m_F(x, y') = s_1$ such that $|opt_F(x) - s_1| \leq \beta |opt_G(f(x)) - s_2|$.

2-7 The composition of L-reductions is also 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 $\alpha\beta\epsilon$ [PY91]. 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).

2-8 Given two k-dimensional vectors χ_1 and χ_2, the Hamming distance between them is denoted by $h(\chi_1, \chi_2)$. For a graph G and a vertex v of G, we denote the degree of v in G by $d_G(v)$. If the edges of G have positive weights, the degree of a vertex v refers to its *weighted degree*, given by the sum of the weights of the edges incident on it. In this paper, a *coloring* of a graph G is *any* assignment of colors to the vertices, and a *k-coloring* is any coloring using at most k colors.

3 Hardness of Approximating Max k-Cut

3-1 In this section, we give an L-reduction from Max Cut to Max k-Cut. The reduction maps a Max Cut instance G to a Max k-Cut instance H, and is such that:

- $opt_{\text{Max } k\text{-Cut}}(H) \leq k(k-1) opt_{\text{Max Cut}}(G)$, i.e., $\alpha = k(k-1)$, and
- given any Max k-Cut solution h with cost s_h, we can find in polynomial time a Max Cut solution g with cost s_g satisfying

$$|opt_{\text{Max Cut}}(G) - s_g| \leq \frac{2}{k} |opt_{\text{Max } k\text{-Cut}}(H) - s_h|$$

  i.e., $\beta = 2/k$.

3-2 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* 34 *such that it is* NP-*hard to approximate* Max k-Cut *with relative error less than* $\frac{1}{\sigma(k-1)}$.

Prove Theorem 1-1 **Proof of Theorem 1** 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. In Håstad [Hås97] it was shown that it is NP-hard to approximate Max Cut within 17/16, which gives us the lower bound 1/17 for the relative error, and the theorem follows.

Proof of Theorem 1 □

Remark 1 *Better lower bounds for* Max Cut *will automatically give better lower bounds for* Max k-Cut.

3-3 We now give the L-reduction from Max Cut to Max k-Cut. In the proof, we assume that k is even, and we transform an unweighted graph G into a weighted graph H. These two restrictions can be assumed without loss of generality. The case when k is odd follows from the following simple reduction from Max k-Cut to Max $(k+1)$-Cut: Given an instance H of the former, produce an instance H' of the latter by adding an extra node u and connecting it to all vertices in H. All edges in H' have weight 1, except edges of the form (u, x), $x \in V(H)$, which have weight $w(u, x) = d_H(u)/k$. This is an L-reduction with $\alpha = (k+1)/(k-1)$ and $\beta = 1$. The proof is rather straightforward and hence is omitted. Just observe that one can assume without loss of generality that all cuts in H' use color $k+1$ for u and colors 1 through k for the isomorphic copy of H contained in H'. The second restriction (the fact that H is weighted, while G is not) could be eliminated by further refining the reduction to yield an unweighted instance of the Max k-Cut problem. This is not needed, however, in view of the more general result of Crescenzi, Silvestri, and Trevisan [CST95]. They show that for a rather general class of optimization problems, including Max k-Cut, the approximation ratios of the weighted and unweighted cases coincide, provided the weights are polynomially bounded. Our reduction below uses weights bounded by $|V(G)|$, and hence that result applies.

3-4 We now turn to the task of defining the L-reduction. The graph H is defined as follows. Assume that $V(G) = \{v_1, \dots, v_n\}$. Let $V_1, \dots, V_{k/2}$ be sets (of vertices) defined by $V_i = \{v_1^i, \dots, v_n^i\}$. That is, the sets $V_1, \dots, V_{k/2}$ are "copies" of the vertex set of G. Actually, we will say that for every $1 \le i \le k/2$, the vertex v_l^i is a *copy* of the vertex v_l of G. Let H be the graph with vertex set $V_1, \cup \cdots \cup V_{k/2}$ and with the following edges:

- If (v_l, v_m) is an edge of G, then (v_l^i, v_m^j) is an edge of H, for $1 \le i \le j \le k/2$. These edges are called *cross edges*.
- All edges between all copies of the same vertex, i.e., for every vertex v_l of G and $1 \le i < j \le k/2$, (v_l^i, v_l^j) is an edge of H. We will refer to these edges as the *row edges*, and the set of copies of v_l will be referred to as the *row of* v_l.

3-5 We make a few observations. First, the row of a vertex v_l induces a clique in H which we call the *row clique of* v_l. Second, if we take two 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. Third, the subgraph of H induced by V_i, call it G_i, is isomorphic to G. Moreover, the map that maps the vertex v_l of G to the vertex v_l^i of G_i (i.e., to its copy in G_i) is an isomorphism. We will call the graphs $G_1, \ldots, G_{k/2}$ the *G-copies*.

3-6 Finally, the edge weights of H are defined as follows: All cross edges have weight 1, whereas all edges in the row of v_l have weight equal to $d_G(v_l)$. We denote the sum of the weights of the edges of H by $w(H)$.

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

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

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

Since $opt_{\text{MAX CUT}}(G) \ge |E(G)|/2$ and $opt_{\text{MAX } k\text{-CUT}}(H) \le w(H)$ for all graphs G and H, the claim follows.

Proof of Lemma 1 □

3-7 Clearly, the graph H can be constructed in polynomial time, and it satisfies the first requirement of the L-reduction. We now prove that the other requirement of the L-reduction is also satisfied. A coloring of H is said to be *canonical* if it satisfies the following:

- each G-copy is bichromatic,
- no two G-copies share a color, and

- all G-copies have exactly the same coloring up to renaming of the colors.

For each graph F and each coloring c of F, we define the *cut-weight* of c to be the sum of the weights of the bichromatic edges of F with respect to c; we denote it by $cw(c, F)$.

Lemma 2 *Given a k-coloring c of H, one can find in polynomial time a canonical k-coloring c' with cut-weight at least as large as that of c.*

3-8 Once the above lemma is established, it follows that the second requirement of the L-reduction is satisfied. We will now prove that this is the case.

Prove Lemma 2-1 **Proof of Lemma 2** Given a coloring of G, we can find a canonical coloring of H; the above lemma effectively means that all k-colorings of H are canonical, including the optimum colorings. If c' is canonical, its cut-weight is given by

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

where g is the coloring appearing in each G-copy, and W is the sum of the weights of all edges connecting vertices in *different* G-copies. In particular, an optimum coloring of H has cut-weight

$$W + \frac{k}{2} opt_{\text{MAX CUT}}(G)$$

It follows that for c and c' as in Lemma 2,

$$\begin{aligned} &|opt_{\text{MAX } k\text{-CUT}}(H) - cw(c, H)| \\ &\qquad \geq |opt_{\text{MAX } k\text{-CUT}}(H) - cw(c', H)| = \frac{k}{2} |opt_{\text{MAX CUT}}(G) - cw(g, G)| \end{aligned}$$

and so the second requirement of the L-reduction is satisfied with $\beta = 2/k$.

Prove Lemma 2-2 The first step in proving Lemma 2 is to show that we may assume without loss of generality that each row is properly colored, that is, it is colored with $k/2$ colors.

Lemma 3 *For each k-coloring c_0 of H, there is a k-coloring c of H such that $cw(c, H) \geq cw(c_0, H)$, and each row edge is bichromatic with respect to c. Moreover, given c_0 and H, the k-coloring c can be obtained in polynomial time.*

Prove Lemma 3-1 **Proof of Lemma 3** 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' that is not used in the row of u, let $A_{p'}$ be the set of vertices z adjacent to x in H such that $c_0(z) = p'$. Notice that for every such set $A_{p'}$, an edge between x and a vertex of $A_{p'}$ 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 that there is a p' such that $|A_{p'}| < d_G(u)$. 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 of greater cut-weight. This process can be repeated until, in polynomially many steps, we have a new coloring that satisfies the stated properties.

Proof of Lemma 3 □

Prove Lemma 2-3 Now fix a k-coloring c of H. By the previous lemma, we may assume that every row edge is bichromatic with respect to c. For each vertex u of G, let $\chi(u)$ be the vector in $\{0,1\}^k$ defined by $\chi(u)_p = 1$ *if and only if* the color p is used in the row of u in H. Notice that by the choice of c, each vector $\chi(u)$ has a Hamming weight of cardinality $k/2$ (recall that the Hamming weight of a vector is the number of its nonzero coordinates).

Prove Lemma 2-4 Given c, we now construct a canonical coloring c'. The crux of the argument is to show how to compute a canonical coloring c' whose number of monochromatic edges is no more than that of c. This implies that the cut-weight of c' is no less than the cut-weight of c.

Prove Lemma 2-5 Roughly speaking, the difficulty of the proof, and what makes it interesting, is to show how, given c, one can get rid of the bichromatic cross edges connecting vertices in *different* G-copies. Local exchange arguments such as those used in Lemma 3 do not seem applicable, and we solve the problem by making use of the vectors $\chi(u)$ and the probabilistic method.

Prove Lemma 2-6 Our next step is to count the number of monochromatic edges with respect to c as a function of the Hamming distance between the $\chi(\cdot)$ vectors.

Lemma 4 *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 $(k - h(\chi(u),\chi(v)))/2$.*

Prove Lemma 4-1 **Proof of Lemma 4** Let P be the set of colors that appear in both rows. Since $\chi(u)$ and $\chi(v)$ are both k-dimensional and have Hamming weight $k/2$

each, 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.

Proof of Lemma 4 □

Prove Lemma 2-7

We now use the $\chi(\cdot)$ vectors to find a good 2-coloring of G that can be duplicated in each G-copy, using new colors in each G-copy. Chose i uniformly at random between 1 and k, and let r be the 2-coloring of G defined by $r(v) = \chi(v)_i$. It follows immediately that for every 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 for the expectation of the number of monochromatic edges of G with respect to r.

Lemma 5

$$E[|\textsc{Mono}(G, r)|] = \frac{2}{k}|\textsc{Mono}(H, c)|$$

where $\textsc{Mono}(G, r)$ *is the set of monochromatic edges of* G *with respect to* r*.*

Prove Lemma 5-1

Proof of Lemma 5 First notice that

$$\begin{aligned}|\textsc{Mono}(H, c)| &= \sum_{(u,v)\in E(G)} |\textsc{Mono}(G, c) \cap \textsc{Cross}(G, u, v)| \\ &= \frac{1}{2} \sum_{(u,v)\in E(G)} (k - h(\chi(u), \chi(v)))\end{aligned}$$

where $\textsc{Cross}(G, u, v)$ is the set of cross edges in G between the row of u and the row of v. The last equality follows from Lemma 4. From this we get

$$\begin{aligned}E[|\textsc{Mono}(G, r)|] &= \sum_{(u,v)\in E(G)} \Pr[r(u) = r(v)] \\ &= \sum_{(u,v)\in E(G)} \frac{k - h(\chi(u), \chi(v))}{k} = \frac{2}{k}|\textsc{Mono}(H, c)|\end{aligned}$$

Proof of Lemma 5 □

Prove Lemma 2-8 Clearly, for at least one 2-coloring g of G, the number of monochromatic edges is at least the expected value. We can find such a 2-coloring simply by computing, for all values of i, the number of monochromatic edges given by the coloring $r(v) = \chi(v)_i$. To obtain a canonical c', just color each G-copy according to g, each time using new colors. The whole process can be carried out in polynomial time.

Proof of Lemma 2 □

4 Hardness of Approximating Min *k*-Partition

4-1 We now show that for each fixed $k > 2$, it is NP-hard to approximate Min k-Partition to within a factor of $O(n^{2-\epsilon})$ for every $\epsilon > 0$. 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 $\epsilon > 0$.*

4-2 We begin with the following simple lemma.

Lemma 6 *For each $k \geq 3$, it is* NP*-hard to decide if a graph is k-colorable over graphs with $\Theta(n^\alpha)$ edges, where $0 < \alpha \leq 2$.*

Prove Lemma 6-1 **Proof of Lemma 6** It suffices to show that the result holds on the family of n-vertex graphs with $\Omega(n^2)$ edges (i.e., dense graphs): We can obtain a graph with $\Theta(n^\alpha)$ edges by adding a disjoint empty graph on $\Theta(n^{2/\alpha})$ vertices. Clearly, this transformation leaves the chromatic number unchanged.

Prove Lemma 6-2 We show only the hardness of 3-coloring on dense graphs: Edwards [Edw86] proved that for $k \geq 4$, k-coloring a dense graph is NP-hard. 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 part of the partition has the same size. If we choose N to be the smallest multiple of 3 greater than or equal to n^2, then clearly the graph G' is dense. Furthermore, it is immediate from the construction that G' is 3-colorable if and only if G is 3-colorable. Hence it is NP-hard to decide if a dense graph is 3-colorable.

Proof of Lemma 6 □

4-3 From here on, we use $\gamma(G, k)$ to denote the minimum number of edges that must be deleted from G to obtain a k-partite graph. The next lemma shows that the NP-hardness of distinguishing between $\gamma(G, k) = 0$ and $\gamma(G, k) \geq 1$ can be amplified to that of distinguishing between $\gamma(H, k) = 0$ and $\gamma(H, k) = \Omega(n^{2-\epsilon})$ for each fixed $\epsilon > 0$. This yields the desired hardness result. The precise statement is as follows.

Lemma 7 (Amplification Lemma) *For all 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 $\gamma(H, k) = s^2\gamma(G, k)$.*

Prove Lemma 7-1

Proof of Lemma 7 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 if and only if (u, v) is an edge in G.

Notice that the set of copies of each vertex u, that is, $u_1, \ldots, u_s$, is an independent set of H. It is easily seen that $|V(H)| = s|V(G)|$, $|E(H)| = s^2|E(G)|$, and that the construction can be done in polynomial time. In the remainder of this section, H will denote the graph obtained from G by the construction above.

Prove Lemma 7-2

Definition 2 (Quasi-coloring) *A coloring of a graph G is called an $\langle m \rangle$ quasi-coloring if it induces at most m monochromatic edges.*

We say that an $\langle m \rangle$ quasi-coloring of H is *normalized* if and only if, for each $u \in V(G)$, all the s copies of u in H, that is, $u_1, \ldots, u_s$, have the same color.

Lemma 8 *Each $\langle m \rangle$ quasi-coloring of H can be transformed into a normalized $\langle m \rangle$ quasi-coloring in polynomial time.*

Prove Lemma 8-1

Proof of Lemma 8 Let c be a given $\langle m \rangle$ quasi-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_is have the same neighbors, the coloring c' that assigns the color of the vertex $u_{\min}$ to each copy u_i, and agrees with coloring c everywhere else, is still an $\langle m \rangle$ quasi-coloring. Iterating this process over each vertex u in G, we get a normalized $\langle m \rangle$ quasi-coloring for H.

Proof of Lemma 8 □

Prove Lemma 7-3 We can now prove that he graph H constructed above satisfies $\gamma(H,k) = s^2\gamma(G,k)$. Every $\langle m \rangle$ quasi-coloring of G can be transformed into an $\langle s^2 m \rangle$ quasi-coloring for H. For each vertex u in G, assign its color to all the copies of u in H. Thus $\gamma(H,k) \le s^2\gamma(G,k)$.

Prove Lemma 7-4 On the other hand, an $\langle m \rangle$ quasi-coloring for H implies an $\langle \lfloor m/s^2 \rfloor \rangle$ quasi-coloring for G. By Lemma 8, we can assume without loss of generality that we are given a normalized $\langle m \rangle$ quasi-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 $\gamma(H,k) \le s^2\gamma(G,k)$. The lemma follows.

Proof of Lemma 7 □

4-5 Hence we have the following theorem.

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

Prove Theorem 2-1 **Proof of Theorem 2** Lemma 6 shows that it is NP-hard to distinguish between $\gamma(G,k) = 0$ and $\gamma(G,k) \ge 1$, even when we are restricted to a family of graphs with $\Theta(n^\alpha)$ edges where $0 < \alpha \le 2$.

Prove Theorem 2-2 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, $\gamma(H,k)$ is 0 if and only if $\gamma(G,k) = 0$, and $\gamma(H,k)$ is $\Omega(N^{2-\epsilon})$ otherwise, where N is the number of vertices in H. This completes the proof.

Proof of Theorem 2 □

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

4-6 The following lemma shows that the hardness result established above is almost tight when $k = 3$.

Lemma 9 *$\gamma(G,3)$ can be approximated within ϵn^2 for every fixed $\epsilon > 0$.*

Prove Lemma 9-1 **Proof of Lemma 9** Let $p = 1/\epsilon$. If $\gamma(G,3) > p$, then the set of all edges constitutes a sufficiently good solution (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 all 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' that satisfies the property that each vertex has degree at least $(\epsilon/2)n$. In G', for every subset of at most p edges, we test whether the graph obtained by deleting F from G' is 3-colorable. 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 [Edw86] or Arora et al. [AKK95]. The result follows.

Proof of Lemma 9 □

4-7 Petrank showed in [Pet94] 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 3 *For each $k > 2$ and $0 < \epsilon < 1$, it is NP-hard to approximate $\gamma(H,k)$ within $O(|E(H)|)$ over the graphs with $\Omega(n^{2-\epsilon})$ edges.*

Prove Theorem 3-1 **Proof of Theorem 3** Petrank showed in [Pet94] the following. For each $k \geq 3$, 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 $\Theta(n) = c'n$ 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: $|V(H)| = sn$, $|E(H)| = c's^2n$, $\gamma(G,k) = 0$ if and only if $\gamma(H,k) = 0$, and $\gamma(G,k) \geq cn$ if and only if $\gamma(H,k) \geq s^2cn$. 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.

Proof of Theorem 3 □

5 Acknowledgments

5-1 We thank Naveen Garg and Johan Håstad for several useful discussions.

Acknowledgement of support: Viggo Kann and Jens Lagergren were supported by grants from the Swedish Natural Science Research Council (NFR) and the Swedish Research Council for Engineering Sciences (TFR). Alessandro Panconesi was supported by a postdoctoral fellowship from the European Research Consortium in Informatics and Mathematics (ERCIM) and by an Alexander von Humboldt research fellowship and by the Swedish Institute of Computer Science. He enjoyed the hospitality of the Swedish Institute of Computer Science and the Royal Institute of Technology (KTH) in Stockholm while performing the research reported in this article.

References

[AKK95] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 284–293. ACM, 1995.

[BGS95] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability—towards tight results. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 422–431. IEEE, 1995.

[CK95] P. Crescenzi and V. Kann. A compendium of NP optimization problems. Technical Report SI/RR-95/02, Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza", 1995. The list is updated continuously. The latest version is available at `http://www.nada.kth.se/theory/problemlist.html`.

[CKST95] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. In *Proceedings of the 1st Annual International Conference on Computing and Combinatorics*, volume 959 of *Lecture Notes in Computer Science*, pages 539–548. Springer-Verlag, 1995.

[CST95] P. Crescenzi, R. Silvestri, and L. Trevisan. To weight or not to weight: Where is the question? In *Proceedings of the 4th Israel Symposium on Theory of Computing and Systems*, pages 66–77. IEEE, 1995.

[Edw86] K. Edwards. The complexity of colouring problems. *Theoretical Computer Science*, 43:337–343, 1986.

[FJ95] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In *Proceedings of the 4th International Conference on Integer Programming and Combinatorial Optimization*, volume 920 of *Lecture Notes in Computer Science*, pages 1–13. Springer-Verlag, 1995.

[GVY96] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25:235–251, 1996.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 1–10. ACM, 1997.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Pet94] E. Petrank. The hardness of approximation: Gap location. *Computational Complexity*, 4:133–157, 1994.

[PT95] S. Poljak and Z. Tuza. Maximum cuts and largest bipartite subgraphs. In W. Cook, L. Lovász, and P. Seymour, editors, *Combinatorial Optimization*, volume 20 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1995.

[PY91] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computing and System Sciences*, 43:425–440, 1991.

[Tur41] P. Turán. An extremal problem in graph theory. *Matematikai Fizikai Lapok*, 48:436–452, 1941. Written in Hungarian.

[Wes96] D. B. West. *Introduction to Graph Theory.* Prentice-Hall Inc., NJ, 1996.