Approximation Algorithms for Graph Homomorphism Problems

Michael Langberg*[1], Yuval Rabani**[2], and Chaitanya Swamy[3]

[1] Dept. of Computer Science, Caltech, Pasadena, CA 91125. mikel@cs.caltech.edu
[2] Computer Science Dept., Technion — Israel Institute of Technology, Haifa 32000, Israel. rabani@cs.technion.ac.il
[3] Center for the Mathematics of Information, Caltech, Pasadena, CA 91125. cswamy@ist.caltech.edu

Abstract. We introduce the *maximum graph homomorphism* (MGH) problem: given a graph G, and a target graph H, find a mapping $\varphi : V_G \mapsto V_H$ that maximizes the number of edges of G that are mapped to edges of H. This problem encodes various fundamental NP-hard problems including Maxcut and Max-k-cut. We also consider the *multiway uncut* problem. We are given a graph G and a set of terminals $T \subseteq V_G$. We want to partition V_G into $|T|$ parts, each containing exactly one terminal, so as to maximize the number of edges in E_G having both endpoints in the same part. Multiway uncut can be viewed as a special case of *prelabeled* MGH where one is also given a prelabeling $\varphi' : U \mapsto V_H,\ U \subseteq V_G$, and the output has to be an extension of φ'.
Both MGH and multiway uncut have a trivial 0.5-approximation algorithm. We present a 0.8535-approximation algorithm for multiway uncut based on a natural linear programming relaxation. This relaxation has an integrality gap of $\frac{6}{7} \simeq 0.8571$, showing that our guarantee is almost tight. For maximum graph homomorphism, we show that a $\left(\frac{1}{2} + \varepsilon_0\right)$-approximation algorithm, for any constant $\varepsilon_0 > 0$, implies an algorithm for distinguishing between certain average-case instances of the *subgraph isomorphism* problem that appear to be hard. Complementing this, we give a $\left(\frac{1}{2} + \Omega(\frac{1}{|H| \log |H|})\right)$-approximation algorithm.

1 Introduction

We introduce the *maximum graph homomorphism* (MGH) problem: given a graph $G = (V_G, E_G)$ and a target or "label" graph $H = (V_H, E_H)$, find a mapping $\varphi : V_G \mapsto V_H$ that maximizes the number of edges of G that are mapped to edges of H. This problem is trivially NP-hard; for example, deciding if G is k-colorable is equivalent to checking if the solution to MGH with graph G and the target graph H being a k-clique, has value $|E_G|$. Several fundamental NP-hard optimization problems can be encoded easily as special cases of MGH.

* Research supported in part by NSF grant CCF-0346991.
** Supported in part by ISF 52/03, BSF 2002282, and the Fund for the Promotion of Research at the Technion. Part of this work was done while visiting Caltech.

For example, *Maxcut* is equivalent to MGH where the target graph H is a single edge; similarly *Max-k-cut* is the problem where H is a k-clique. This also shows that MGH is APX-hard even when H is fixed (i.e., not part of the input), that is, there is some absolute constant $\varepsilon_0 > 0$ such that it is NP-hard to approximate MGH better than a factor of $1 - \varepsilon_0$. The maximum graph homomorphism problem is an optimization version of the well-studied *H-coloring* problem [20], which is the problem of *deciding* whether there exists a mapping φ of value equal to $|E_G|$ (such a mapping is called a *homomorphism*).

We also consider a *prelabeled* version of the maximum graph homomorphism problem (prelabeled MGH), where the input also includes a partial mapping $\varphi' : U \mapsto V_H$ where $U \subseteq V_G$, and the output is restricted to extensions $\varphi : V_G \mapsto V_H$ of φ'. This problem, too, includes some natural NP-hard problems as special cases. For example, consider the *multiway uncut* problem (the complement of *multiway cut*): given a graph G and a set of terminals $T \subseteq V_G$, partition V_G into $|T|$ parts, each containing exactly one element of T, so as to maximize the number of edges in E_G whose both endpoints lie in the same part. This is precisely prelabeled MGH where H consists of $|T|$ disconnected self-loops, and the prelabeling $\varphi' : T \mapsto V_H$ is a bijection.

Our Results. We present a 0.8535-approximation algorithm for the *multiway uncut* problem in Section 3. To the best of our knowledge, this is the first time anyone has considered this problem. From an exact optimization point of view, multiway uncut is equivalent to the complementary problem of multiway cut introduced by Dahlhaus et al. [9], and the APX-hardness reduction for multiway cut in [9] also shows that our problem is APX-hard. However, approximation results for multiway cut [9, 5, 23] do not directly yield guarantees for the maximization objective of multiway uncut. Our algorithm is based on a natural linear programming (LP) relaxation and rounding procedure that are motivated by the work of Calinescu, Karloff and Rabani [5] on multiway cut, and Kleinberg and Tardos [24] on the related uniform labeling problem.

In Section 4, we consider the *maximum graph homomorphism* (MGH) problem. MGH admits a simple 0.5-approximation algorithm: take any edge (i, j) of H, run the randomized/greedy algorithm for Maxcut on G to obtain a cut of value $\frac{1}{2}|E_G|$, and map the two sides of the cut to i and j. (The problem is trivial if H contains no edges, or self-loops.) This gives a solution of value at least $\frac{1}{2}|E_G|$. Our work focuses on the question of improving upon the ratio of 0.5.

We show that in general, any $\left(\frac{1}{2}+\varepsilon_0\right)$-approximation algorithm for a constant $\varepsilon_0 > 0$, would imply an algorithm for deciding certain average-case instances of the *subgraph isomorphism* problem that appear to be hard. This suggests an inherent difficulty in obtaining such an improvement. This result falls into the line of research, initiated by Feige [14], of using average-case complexity assumptions to derive hardness of approximation results. The basis of our reduction is the following key fact (that we prove): if H is a *triangle-free* graph, and G is a *random graph* drawn from the distribution $\mathcal{G}_{n,p}$ where $p = \Theta\left(\frac{\ln |V_H|}{n}\right)$, then with high probability, no mapping φ maps more than $\frac{|E_G|}{2}(1 + \epsilon)$ edges of G (the constant in $\Theta(.)$ depends on ϵ). So when G and H are drawn from a suitable distribution on triangle-free graphs, this establishes a factor 2 gap between the cases when G is a subgraph of H (so there is a mapping of value $|E_G|$), and when it is not. Thus, a $\left(\frac{1}{2}+\varepsilon_0\right)$-approximation algorithm would allow us to distinguish between these two cases.

Motivated by the known better bounds for some special cases of MGH (e.g., Maxcut [18]), we also study special families of label graphs H. We present a $\left(\frac{1}{2} + \Omega(\frac{1}{|V_H| \log |V_H|})\right)$-approximation algorithm for MGH, by using an algorithm of Charikar and Wirth [6] for Maxcut that is based on rounding the semidefinite program for Maxcut used by Goemans and Williamson [18]. This gives an improvement over the approximation ratio of 0.5 for any fixed graph H. We obtain better improvements for some structured classes of graphs H. For the prelabeled problem, we show that an α-approximation algorithm for unlabeled MGH with label graph H yields an $\frac{\alpha}{1+\alpha}$-approximation algorithm for prelabeled MGH with graph H. Finally, we consider the problem on dense graphs G and obtain a PTAS for any fixed H, and a quasi-PTAS when H is part of the input.

Related Work. We are not aware of any previous work on the maximum graph homomorphism (or the prelabeled version) or the multiway uncut problems.

As mentioned earlier, the maximum graph homomorphism problem is an optimization version of the *H-coloring* problem, which is the problem of deciding if there exists a mapping $\varphi : V_G \mapsto V_H$ (called a homomorphism or H-coloring) that maps each edge of G to an edge of H. Homomorphisms, and the H-coloring problem and its variants have been extensively studied from various perspectives; see, e.g., [21] and the references therein. Hell and Nešetřil [20] showed that H-coloring is in P if H contains a self loop or is bipartite, and NP-complete otherwise. Dyer and Greenhill [12] established a similar dichotomy for the problem of counting the number of H-colorings, namely, that the problem is either in P or is #P-complete. Various variants of the H-coloring problem and their counting versions have also been studied; see, e.g., [13, 11]. Cooper et al. [8] considered the problem of sampling a random H-coloring.

Minimization versions of the H-coloring problem have been considered in [19, 7, 1]. Here there is a cost for assigning a label to a node of G and/or weights associated with the edges of H, and one seeks a mapping/homomorphism φ that minimizes the sum of the labeling costs and the weights of the images of the edges of G. (If the edge weights form a metric, then this is precisely the metric labeling problem [24].) Cohen et al. [7] consider the setting where the weight of assigning an edge $e \in E_G$ to an edge of H may even depend on e, and identify a class of cost functions for which the problem is in P. Aggarwal et al. [1] consider the problem with edge weights where H is a complete graph with self-loops at every node, and present various approximation and inapproximability results. Gutin et al. [19] consider the problem with only labeling costs, restricting φ to be a homomorphism, and classify the polynomial-time solvable and NP-hard cases.

A closely related problem is the *maximum common subgraph* problem: given two graphs G and H we want to find a subgraph of G with maximum number of edges that is isomorphic to a subgraph of H. MGH can be reduced to the maximum common subgraph problem by replacing each node of H by an independent set of size $|V_G|$, and each edge of H by the corresponding complete bipartite graph. Kann [22] presented a $B+1$-approximation algorithm, where B is the maximum degree in G and H. Notice that the reduction outlined above does not preserve the degrees in the target graph H.

The complement of the multiway uncut problem, namely the *multiway cut* problem, was introduced by Dahlhaus et al. [9]. They showed that multiway cut is APX-hard, and gave a $\left(2 - \frac{2}{|T|}\right)$-approximation algorithm. Calinescu, Karloff and Rabani [5] proposed a new LP relaxation for the problem and used this to improve the factor to $\left(1.5 - \frac{1}{|T|}\right)$. The current best factor is 1.3438 due to Karger et al. [23]. Our LP-relaxation for multiway uncut is the same as the one in [5] (but with a maximization objective), and our algorithm uses a rounding procedure of Kleinberg and Tardos [24] for the uniform labeling problem (which is a generalization of the multiway cut problem).

Basing hardness of approximation results on average-case complexity is an evolving field of research which was initiated by the work of Feige [14]. Feige gave the first inapproximability results for various NP-hard optimization problems assuming the complexity of refuting random-3CNF formulas. Subsequently, results of a similar nature (for other optimization problems, based on other hardness assumptions) were obtained by Alekhnovich [2] and Demaine et al. [10].

2 Definitions and Preliminaries

Maximum Graph Homomorphism. The input to the *maximum graph homomorphism* (MGH) problem consists of two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$. The objective is to find a mapping $\varphi : V_G \mapsto V_H$ that maximizes the number of edges of G that are mapped to edges of H. More formally, we want to maximize $|\{(u, v) \in E_G : \ (\varphi(u), \varphi(v)) \in E_H\}|$. We will often refer to the mapping φ as a labeling, $\varphi(u)$ as the label of u, and H as the label graph or target graph. Let $OPT(G, H)$ denote the value of an optimal solution. Throughout, n will denote $|V_G|$ and k will denote $|V_H|$. We use variables u, v, w to denote vertices in V_G and i, j, ℓ to denote vertices in V_H.

We also consider a prelabeled version of maximum graph homomorphism where some of the nodes of G are already labeled, and we want to label the remaining vertices so as to maximize the objective function. More precisely, in the *prelabeled maximum graph homomorphism* problem, in addition to the graphs G and H, we are given a prelabeling $\varphi' : U \mapsto V_H$ where $U \subseteq V_G$, and the goal is to find an extension φ of φ' that maximizes $|\{(u, v) \in E_G : \ (\varphi(u), \varphi(v)) \in E_H\}|$. In general, the label graph H may also contain self-loops. However, note that if H has a self-loop, say at node i, then the unlabeled problem becomes trivial: we can simply map every vertex of G to label i to obtain $OPT(G, H) = |E_G|$. Thus, the problem with self-loops is only interesting in the prelabeled setting.

The Multiway Uncut Problem. In the *multiway uncut* problem, we are given a graph $G = (V, E)$ and a set of k terminals $T \subseteq V$. We want to find a partition of V into k subsets $V_1, \ldots, V_k$ such that each part V_i contains a distinct terminal, so as to maximize the number of *uncut* edges, that is, the quantity $\sum_{i=1}^{k} |\{(u,v) \in E : \; u,v \in V_i\}|$. Notice that the multiway uncut problem is a special case of the prelabeled MGH problem, where the label graph H consists of k disconnected self loops and the prelabeling is a bijection $\varphi' : T \mapsto V_H$.

3 The Multiway Uncut Problem

In this section, we consider the multiway uncut problem and present a 0.8535-approximation algorithm based on a natural linear programming (LP) relaxation. The integrality gap of this relaxation is at least $\frac{6}{7} \simeq 0.8571$, which shows that our guarantee is almost tight. Since multiway uncut is a special case of the prelabeled maximum graph homomorphism problem, we will use the terminology of MGH for consistency: we have k labels $i = 1, \ldots, k$, and the prelabeling φ' is given by $\varphi'(t_i) = i$ for the i-th terminal $t_i \in T$. Note that we may assume that there are no edges between two labeled vertices since such edges contribute 0 to the value of any solution. We consider the following LP relaxation. We use u to index the vertices of $G = (V, E)$, and i to index the labels.

$$\begin{aligned}
\max \quad & \sum_{(u,v)\in E} \sum_{i} c^i_{uv} & & & \text{(MU-LP)} \\
\text{s.t.} \quad & \sum_{i} x^i_u = 1 & & \text{for all } u, \\
& x_t^{\varphi'(t)} = 1 & & \text{for all } t \in T, \\
& c^i_{uv} = \min(x^i_u, x^i_v) & & \text{for all } (u,v) \in E & (1) \\
& x^i_u, c^i_{uv} \geq 0 & & \text{for all } u, v, i.
\end{aligned}$$

Here x^i_u indicates if vertex u is assigned label i, and c^i_{uv} indicates if both endpoints of edge (u,v) are assigned label i. The first constraint states that every node must be assigned a label, and the second enforces that this labeling is an extension of φ' (i.e., the label of a terminal does not change). The term $\sum_i c^i_{uv}$ measures the *similarity* along edge (u,v). Although (1) is not written as a linear constraint, it is easy to see that one can encode (1) using linear constraints.

One can show that the LP relaxation (MU-LP) is identical to the relaxation introduced by Calinescu et al. [5] for the multiway cut problem, i.e., any solution of value *Val* to (MU-LP) is a solution of value $|E| - \mathit{Val}$ to the relaxation in [5]. For the multiway cut problem, Calinescu et al. showed that the integrality gap of the relaxation is at most $1.5 - \frac{1}{k}$, which was improved to 1.3438 [23], whereas Freund and Karloff [16] showed that the integrality gap is at least $\frac{8}{7+1/(k-1)}$.

Our result shows that the integrality gap of (MU-LP) (which is now less than 1) is at most 0.8535, that is, there is always an integer solution of value at least 0.8535 times the optimum of (MU-LP). This also holds in the weighted setting (non-negative edge weights) where the goal is to maximize the weight of the uncut edges. The integrality-gap example in [16] also yields an integrality gap of $\frac{6k^2-10k+4}{7k^2-13k+6} \to \frac{6}{7} \simeq 0.8571$, as $k \to \infty$, for (MU-LP) (with weighted edges). Thus, our guarantee is very close to the best possible using this LP relaxation.

A similar LP relaxation was used by Kleinberg and Tardos [24] for the uniform labeling problem. We will use a randomized rounding procedure from [24], but we will need a more refined analysis of this procedure than that in [24]. The algorithm is simple: we return the better of the following two labelings.

1. The first labeling picks an arbitrary label i, and sets $\varphi(u) = i$ for every vertex $u \notin T$. We call this the "trivial labeling".
2. The second labeling is obtained via the randomized rounding procedure of Kleinberg and Tardos, which we describe below for completeness. They also show how to derandomize the rounding, so we could use this and obtain a deterministic algorithm with the same performance guarantee. We consider the randomized version for ease of exposition and analysis.
   Let $\{x, c\}$ be an optimal solution to (MU-LP). The rounding proceeds in several rounds. Initially all vertices in $V \setminus T$ are unassigned. In each round, we independently pick a label $i \in \{1, \ldots, k\}$ uniformly at random, and a threshold ρ uniformly in $[0, 1]$. For each unassigned vertex $u \in V$, we assign u the label i (i.e., set $\varphi(u) = i$) if $x_u^i \geq \rho$. We repeat this until all the vertices in $V \setminus T$ are assigned. We call this the "LP labeling".

Analysis. Let $C_{uv} = \sum_i c_{uv}^i$. We analyze the algorithm by considering a "hybrid labeling", where we choose the LP-labeling with probability λ and the trivial labeling with probability $1-\lambda$, for some $\lambda \in [0, 1]$. We will compare the expected contribution of an edge (u, v) in the hybrid labeling against the LP-value C_{uv}. Let $E_0 = \{(u, v) \in E : \ u, v \notin T\}$ and $E_1 = \{(u, v) \in E : \ u \text{ or } v \in T\}$. Note that $E = E_0 \cup E_1$ since there are no edges with both endpoints in T. The trivial labeling obtains a value of 1 for every edge in E_0. We now analyze the LP-labeling. For an edge (u, v), let X_{uv} denote the random variable that is 1 if u and v are assigned the same label in the LP-labeling, and 0 otherwise. We will use "$u \mapsto i$" and "$u \mapsto *$" as a shorthand to denote that "u is assigned label i", and "u is assigned some label" respectively. Let X_u^i be a random variable that is 1 if $u \mapsto i$ in the LP-labeling, and 0 otherwise.

Fact 3.1. *Suppose u is unassigned before a round. Then,* $\Pr[u \mapsto i \text{ in the round}] = \frac{1}{k} \cdot x_u^i$. *Therefore* $\Pr[u \mapsto * \text{ in the round}] = \frac{1}{k} \cdot \sum_i x_u^i = \frac{1}{k}$.

Claim 3.2. $\Pr[X_u^i = 1] = x_u^i$. *Thus, for an edge $(u, v) \in E_1$,* $\mathrm{E}\big[X_{uv}\big] = C_{uv}$.

Proof. $\Pr[X_u^i = 1] = \sum_{r=1}^{\infty} \big(1 - \Pr[u \mapsto * \text{ before round } r]\big) \cdot \Pr[u \mapsto i \text{ in round } r] = \sum_{r=1}^{\infty} \big(1 - \frac{1}{k}\big)^{r-1} \cdot \frac{x_u^i}{k} = x_u^i$. For an edge $(u, v) \in E_1$, where $v \in T$ has label i, we have $\mathrm{E}\big[X_{uv}\big] = \Pr[X_u^i] = x_u^i = c_{uv}^i = C_{uv}$. □

Lemma 3.3. *For an edge $(u, v) \in E_0$, we have* $\mathrm{E}\big[X_{uv}\big] \geq \frac{C_{uv}}{2 - C_{uv}}$.

Proof. We can lower bound $\mathrm{E}\big[X_{uv}\big]$ by the probability that both u and v are assigned a label in the *same* round. Observe that if both u and v are unassigned before a given round, then (a) the probability that u *and* v are both assigned in the round is $\frac{1}{k}\sum_i \min(x_u^i, x_v^i) = \frac{1}{k} \cdot C_{uv}$, and (b) the probability that u *or* v is assigned in the round is $\frac{1}{k}\sum_{i=1}^k \max(x_u^i, x_v^i) = \frac{1}{k} \cdot (2 - C_{uv})$, since $\sum_i \bigl(\min(x_u^i, x_v^i) + \max(x_u^i, x_v^i)\bigr) = 2$. Thus, $\Pr[u$ and v are assigned in the same round$]$ is exactly

$$\sum_{r=1}^{\infty}\bigl(1 - \Pr[u \mapsto * \text{ or } v \mapsto * \text{ before round } r]\bigr) \cdot \Pr[u \mapsto * \text{ and } v \mapsto * \text{ in round } r]$$

$$= \sum_{r=1}^{\infty}\Bigl(1 - \frac{2 - C_{uv}}{k}\Bigr)^{r-1} \cdot \frac{C_{uv}}{k} = \frac{C_{uv}}{2 - C_{uv}}. \qquad \square$$

Fact 3.1 and Claim 3.2 were proved in [24], but for edges in E_0 their analysis proves the weaker bound $\mathrm{E}\bigl[X_{uv}\bigr] \geq 1 - \|x_u - x_v\|_1 = 2C_{uv} - 1$ which only yields a $\frac{2}{3}$-approximation guarantee for the overall algorithm.

Theorem 3.4. *The solution returned has value at least* $\bigl(\frac{1}{2} + \frac{\sqrt{2}}{4}\bigr) \cdot \bigl(\sum_{(u,v)\in E} C_{uv}\bigr)$. *Thus the approximation ratio of the above algorithm is at least* $\frac{1}{2} + \frac{\sqrt{2}}{4} \simeq 0.8535$.

Proof. We prove the stated bound for the expected value of the random hybrid labeling; the theorem then follows. For an edge $(u, v) \in E_0$, we get an expected value of (at least) $\frac{C_{uv}}{2 - C_{uv}}$ in the LP-labeling by Lemma 3.3, and 1 in the trivial labeling. So the expected contribution of this edge in the hybrid labeling is at least $C_{uv} \cdot \bigl(\frac{\lambda}{2 - C_{uv}} + \frac{1-\lambda}{C_{uv}}\bigr) \geq \bigl(\frac{1}{2} + \sqrt{\lambda(1-\lambda)}\bigr)C_{uv}$. The last inequality follows since $\min_{C\in[0,1]}\bigl(\frac{\lambda}{2-C} + \frac{1-\lambda}{C}\bigr) \geq \frac{1}{2} + \sqrt{\lambda(1-\lambda)}$ by simple calculus. For an edge $(u, v) \in E_1$, using Claim 3.2, the (expected) contribution in the hybrid labeling is at least λC_{uv}. Therefore the expected total value of the hybrid labeling is at least $\min\bigl(\lambda, \frac{1}{2} + \sqrt{\lambda(1-\lambda)}\bigr) \cdot \bigl(\sum_{(u,v)\in E} C_{uv}\bigr)$. Taking $\lambda = \frac{1}{2} + \frac{\sqrt{2}}{4} = \frac{1}{2} + \sqrt{\lambda(1-\lambda)} \simeq 0.8535$ maximizes this expression and yields a solution of value at least $0.8535 \cdot \bigl(\sum_{(u,v)\in E} C_{uv}\bigr)$. As mentioned earlier, the rounding procedure can be derandomized to yield a deterministic algorithm with the same guarantee. $\square$

Extensions. We can also handle the weighted case where we have non-negative weights on the edges and we want to maximize the weight of the uncut edges. The algorithm remains unchanged and the analysis requires only notational changes. One can also consider the problem where we have non-negative *profits* $\{p_u^i\}$ for assigning label i to node u, and we want to maximize the sum of the profits and the weight of the uncut edges. This problem is the complement of the uniform labeling problem considered in [24]. We can reduce this to the no-profit setting by adding an edge (u, i) with weight p_u^i for every node $u \in V$ and label i.

4 The Maximum Graph Homomorphism Problem

We now consider the maximum graph homomorphism (MGH) problem (with an arbitrary label graph H). Recall that we are given graphs G and H, and the goal is to find a mapping $\varphi : V_G \mapsto V_H$ that maximizes the number of edges of G

mapped to edges of H. In Section 4.1, we give a $\left(\frac{1}{2}+\Omega(\frac{1}{k\log k})\right)$-approximation algorithm (where $k = |V_H|$) for this problem. In Section 4.2, we present some evidence suggesting that obtaining a $\left(\frac{1}{2}+\Omega(1)\right)$-approximation algorithm may be inherently difficult. We argue that such an approximation algorithm would yield an algorithm for distinguishing between certain average-case instances of the subgraph isomorphism problem. In Section 4.3, we consider some extensions and refinements. We show that any approximation guarantee for the unlabeled problem yields a corresponding guarantee for prelabeled MGH. We also obtain a quasi-PTAS for the problem on dense graphs G (i.e., $|E_G| = \Omega(|V_G|^2)$).

4.1 A $\left(\frac{1}{2}+\Omega(\frac{1}{k\log k})\right)$-Approximation Algorithm

We now present the $\left(\frac{1}{2}+\Omega(\frac{1}{k\log k})\right)$-approximation algorithm. Recall that $k = |V_H|$. We assume that H contains at least one edge and has no self-loops (otherwise the problem is trivial). We start with some simple observations. Observe that any cut $(U, V_G \setminus U)$ of G yields a labeling φ of value equal to the size of the cut, since we can consider any edge $(i,j) \in E_H$ and map all the nodes in U to i, and all the nodes in $V_G \setminus U$ to j. Thus, since one can easily obtain a cut of value at least $\frac{|E_G|}{2}$ (e.g., by using the greedy, or randomized, algorithm where we assign each vertex greedily, or independently and uniformly at random, to one of the two parts), there is a trivial 0.5-approximation algorithm for the maximum graph homomorphism problem. Conversely, for bipartite graphs H, one can show that $MaxCut(G) = OPT(G,H)$.

Fact 4.1. *Any cut of G yields a mapping φ of value equal to the size of the cut. Thus, $OPT(G,H) \geq MaxCut(G) \geq \frac{|E_G|}{2}$.*

Claim 4.2. *If H is bipartite, the MGH problem on graphs G and H is equivalent to the Maxcut problem on G, that is, $MaxCut(G) = OPT(G,H)$.*

We improve upon this factor of 0.5 for any fixed graph H, by using a result of Charikar and Wirth [6]. They used the semidefinite program for Maxcut in [18], along with the RPR^2 rounding technique of [15] to obtain the following theorem.

Theorem 4.3 (Charikar and Wirth). *Let G be a graph with non-negative edge weights, having a cut of weight $|E_G|\left(\frac{1}{2}+\delta\right)$, where $\delta > 0$. One can obtain a cut of G with weight $|E_G|\left(\frac{1}{2}+\frac{c\delta}{\log(1/\delta)}\right)$ in polynomial time, where c is a constant.*

Notice that the algorithm mentioned in the above theorem always returns a cut of value at least $\frac{|E_G|}{2}$. Our algorithm for MGH simply uses the algorithm mentioned in Theorem 4.3 to obtain a cut of G; this induces a labeling of the same value and the algorithm returns this labeling. The idea behind the algorithm is that if $OPT(G,H)$ is small compared to $|E_G|$, then $\frac{|E_G|}{2}$ would be strictly larger than $\frac{OPT(G,H)}{2}$. Otherwise, we will show that there exists a *bipartite subgraph* H' of H that captures more than half the edges of G, which in turn implies that G has a cut of value strictly larger than $\frac{|E_G|}{2}$. Thus, using Theorem 4.3 we obtain a cut of G, and hence a labeling, of value strictly larger than $\frac{|E_G|}{2} \geq \frac{OPT(G,H)}{2}$.

Theorem 4.4. *There is a* $\big(\frac{1}{2}+\frac{c}{k\log k}\big)$*-approximation algorithm for MGH, where* $c>0$ *is a constant independent of* k.

Proof. Let G and H be the input graphs. If $OPT(G,H) \le |E_G|\big(1-\frac{1}{2k}\big)$, then our algorithm returns a solution of value at least $\frac{|E_G|}{2} \ge \frac{OPT(G,H)}{2(1-1/2k)} \ge \frac{OPT(G,H)}{2}\big(1+\frac{1}{2k}\big)$. So suppose that $OPT(G,H) \ge |E_G|\big(1-\frac{1}{2k}\big)$. Consider an optimal mapping φ^*. For each edge (i,j) in H, let $m_{ij} = \big|\big\{(u,v)\in E_G:\ \{\varphi^*(u),\varphi^*(v)\}=\{i,j\}\big\}\big|$. Thus, $OPT(G,H)=\sum_{(i,j)\in E_H} m_{ij}$. We claim that there is a bipartite subgraph H' of H such that $OPT(G,H') \ge \sum_{(i,j)\in H'} m_{ij} \ge \frac{|E_G|}{2}\big(1+\frac{1}{4k}\big)$. Consider the cut $(U_H, V_H\setminus U_H)$ where U_H is a random subset of vertices of H of size $k/2$. The probability that an edge is cut by such a partition is $\frac{k^2}{4}/\binom{k}{2}=\frac{1}{2}\big(1+\frac{1}{k-1}\big)$. Therefore, the expected weight of the cut edges is $\big(\sum_{(i,j)\in E_H} m_{ij}\big)\cdot\frac{1}{2}\big(1+\frac{1}{k-1}\big) = \frac{OPT(G,H)}{2}\big(1+\frac{1}{k-1}\big) \ge \frac{|E_G|}{2}\big(1+\frac{1}{4k}\big)$. Thus, there exists such a partition of at least this value, and we can take H' to be the associated bipartite subgraph of H. Now by Claim 4.2, G must have a cut of value at least $\frac{|E_G|}{2}\big(1+\frac{1}{4k}\big)$. So applying Theorem 4.3, our algorithm finds a cut, and hence a labeling, of value at least $|E_G|\big(\frac{1}{2}+\frac{c}{k\log k}\big)$. The theorem follows since $OPT(G,H)\le |E_G|$. □

4.2 Connection to the Subgraph Isomorphism Problem

Given two graphs G and H, the *subgraph isomorphism* problem is the problem of deciding whether G is a subgraph of H. The subgraph isomorphism problem is a well-known NP-complete problem. We show that a $(\frac{1}{2}+\varepsilon_0)$-approximation algorithm for MGH, where $\varepsilon_0>0$ is an absolute constant, implies an algorithm for distinguishing between certain average-case instances of the subgraph isomorphism problem (this is defined precisely below). This hints at an inherent difficulty in obtaining an approximation ratio better than 0.5 for MGH.

The main technical result of this section (Lemma 4.5) is as follows. For any $\epsilon>0$, if H is a *triangle-free graph*, and G is a *random graph* drawn from the distribution $\mathcal{G}_{n,p}$, for a suitable $p=p(\epsilon)\in[0,1]$ and large enough n, then $OPT(G,H)\le\frac{|E_G|}{2}(1+\epsilon)$ with high probability. If however G is a subgraph of H, then $OPT(G,H)=|E_G|$. The gap between these two cases motivates the definition of a refutation problem for certain average-case instances of the subgraph isomorphism problem, which allows us to encode the difficulty of obtaining a better than 0.5-approximation algorithm for MGH. Let $\triangle_n$ be the set of all triangle-free graphs on n vertices. For $p\in[0,1]$, let $\triangle_{n,p}$ be the distribution over $G\in\triangle_n$ obtained by choosing a random graph $G\in\mathcal{G}_{n,p}$, and then considering the edges of G in a random order and deleting any edge that is part of a triangle.

Refutation problem (with parameter $c>0$). Find a polynomial time algorithm $\mathcal{A}$ such that given a pair of random graphs $G\in\triangle_{n,p_G}, H\in\triangle_{n,p_H}$, where $p_G=\frac{c\ln n}{n}$, $p_H \gg p_G$, (a) $\mathcal{A}$ returns "yes" if H contains G as a subgraph, and (b) $\mathcal{A}$ returns "no" on most instances, more precisely $\Pr_{G,H}[\mathcal{A}(G,H)=\text{"no"}]\ge\frac{1}{2}$.

Intuitively, the *refutation algorithm* $\mathcal{A}$ refutes most tuples (G,H) as being "no" instances of the subgraph isomorphism problem, but always announces "yes" when G is a subgraph of H. As mentioned earlier, with very high probability G will not be a subgraph of H, thus conditions (a) and (b) do not conflict. We will show that a $\left(\frac{1}{2}+\varepsilon_0\right)$-approximation algorithm for MGH yields such a refutation algorithm; thus the non-existence of such an algorithm implies that MGH cannot be approximated to a factor better than 0.5.

We mention a few remarks. First, one could also define the refutation problem in terms of an approximation version of subgraph isomorphism by requiring (a'): $\mathcal{A}$ always return "yes" if G contains a subgraph of size $|E_G|(1-\epsilon)$ that is isomorphic to a subgraph of H. Such a modification was also considered by Feige (see Hypothesis 2 in [14]). An algorithm satisfying (a'), (b) refutes average-case instances of the *maximum common subgraph problem* [22], and is also a refutation algorithm for the exact-version of the problem. Thus, the non-existence of an algorithm satisfying (a'), (b) is a weaker hardness assumption (implying a $(\frac{1}{2}+\varepsilon_0)$ inapproximability for MGH). Moreover, this version of the refutation problem might be more robust than the exact-version. Second, we take $p_H \gg p_G$ to avoid the case where $p_H \simeq p_G$. In this setting, the problem is closely related to the graph isomorphism problem on random graphs, which is known to be solvable on average in polynomial time; see [4], and §6 of the survey [17] and its references.

Lemma 4.5. *For any $\epsilon \in (0,1)$, there exist constants $n_0(\epsilon), c_0(\epsilon)$, such that if $G=(V_G,E_G)$ is a random graph in $\mathcal{G}_{n,p}$, where $n \geq n_0(\epsilon)$, $p=\frac{c\ln k}{n}$, $c \geq c_0(e)$, and $H=(V_H,E_H)$ is a simple triangle-free graph with k vertices, then (i) $OPT(G,H) < \frac{cn\ln k}{4}(1+\epsilon/2)$ with probability at least $1-e^{-n\ln k}$, and (ii) $OPT(G,H) < \frac{|E_G|}{2}(1+\epsilon)$ with probability at least $1-2e^{-n\ln k}$.*

Proof. Set $n_0(\epsilon)=\frac{8}{\epsilon}$, $c_0(\epsilon)=\frac{2048}{7\epsilon^2}$. Let $m=p\binom{n}{2}$ be the expected number of edges in G. Fix a mapping $\varphi: V_G \mapsto V_H$. We will show that with very high probability, mapping φ has value at most $\frac{m}{2}(1+\epsilon/2)$. Applying the union bound over all mappings then yields that $OPT(G,H) < \frac{m}{2}(1+\epsilon/2)$ with high probability, proving part (i). Since $|E_G|$ is strongly concentrated around its expectation, this will also prove part (ii).

Given the mapping φ, consider the following graph H': H' also has $n=|V_G|$ vertices, and we include an edge (u,v) in H' iff $(\varphi(u),\varphi(v))$ is an edge in H. It is easy to see that H' is also triangle-free: a triangle (v_1,v_2,v_3) in H' implies that H has edges $(\varphi(v_1),\varphi(v_2))$, $(\varphi(v_2),\varphi(v_3))$, and $(\varphi(v_3),\varphi(v_1))$, and therefore contains a triangle. Since H' is triangle-free, by Turán's Theorem [25] it has at most $\frac{n^2}{4}$ edges. Let $X(\varphi)$ denote the (random) value of the mapping φ for G. Observe that $X(\varphi)$ is simply the number of edges of H' that are also edges of G. For every pair u,v, (u,v) is in E_G with probability p, so we have $\mathrm{E}\big[X(\varphi)\big] = p \cdot |E_{H'}| \leq p \cdot \frac{n^2}{4} = \frac{cn\ln k}{4}$. Since $X(\varphi)$ is the sum of independent indicator random variables, using Chernoff bounds, we get $\Pr\big[X(\varphi) \geq \frac{cn\ln k}{4}(1+\epsilon/2)\big] \leq e^{-(\epsilon^2 cn\ln k)/48} \leq e^{-2n\ln k}$. The number of mappings φ is k^n. So by the union bound, $\Pr\big[OPT(G,H) \geq \frac{cn\ln k}{4}(1+\epsilon/2)\big] = \Pr\big[\exists\varphi,\ X(\varphi) \geq \frac{cn\ln k}{4}(1+\epsilon/2)\big] \leq e^{-n\ln k}$.

The expected number of edges in G is $p\binom{n}{2} = \frac{cn\ln k}{2}(1-\frac{1}{n}) \geq \frac{cn\ln k}{2}(1-\frac{\epsilon}{8})$. Again using Chernoff bounds, we get that $\Pr\big[|E_G| \leq \frac{cn\ln k}{2}(1-\epsilon/4)\big] \leq$

$e^{-(7\epsilon^2 cn\ln k)/2048} \le e^{-n\ln k}$. So using part (i), with probability at least $1-2e^{-n\ln k}$ it is the case that $OPT(G,H) < \frac{cn\ln k}{4}(1+\epsilon/2) < \frac{|E_G|}{2}(1+\epsilon)$. □

Theorem 4.6. *For any $\varepsilon_0 > 0$, a $\left(\frac{1}{2}+\varepsilon_0\right)$-approximation algorithm $\mathcal{A}$ for MGH yields an algorithm for the refutation problem with parameter $c \ge c_0(\varepsilon_0) = \frac{2048}{7\varepsilon_0^2}$.*

Proof Sketch. Let G and H be the two input graphs. Let n be sufficiently large. If we are in case (a), then $OPT(G,H) = |E_G|$, so running $\mathcal{A}$ on (G,H) will produce a solution of value at least $|E_G|\left(\frac{1}{2}+\varepsilon_0\right)$. Otherwise, we can use Lemma 4.5 to show that that $OPT(G,H) < |E_G|\left(\frac{1}{2}+\varepsilon_0\right)$ with high probability; thus, one can use $\mathcal{A}$ to distinguish between the two cases. Let G by obtained by deleting edges from $G' \in \mathcal{G}_{n,p}$. Lemma 4.5 shows that $OPT(G,H) \le OPT(G',H) < \frac{cn\ln n}{4}(1+\varepsilon_0/2)$ and $|E'_G| \ge \frac{cn\ln n}{2}(1-\varepsilon_0/4)$, with high probability. Although we delete edges from G', with high probability, the number of triangles in G' is a negligible fraction of $|E_{G'}|$. So we obtain that $|E_G| \ge \frac{cn\ln n}{2}(1-\varepsilon_0/2)$ and therefore we have $OPT(G,H) < |E_G|\left(\frac{1}{2}+\varepsilon_0\right)$. □

4.3 Extensions and Refinements

Prelabeled MGH. Recall that in prelabeled MGH, we are given a prelabeling $\varphi' : U \mapsto V_H,\ U \subseteq V_G$ and the output has to be an extension of φ'. We can show that for any label graph H, an α-approximation algorithm for MGH on instances (G,H) (α could depend on H) gives an $\frac{\alpha}{1+\alpha}$-approximation algorithm for prelabeled MGH on instances (G,H).

Dense Graphs G. We obtain much better results when G is dense, i.e., when $|E_G| = \Omega(n^2)$ ($n = |V_G|$). One can adapt the techniques of Arora, Karger and Karpinski [3] to obtain a solution φ of value $OPT(G,H) - \epsilon n^2$ in time $O\left((nk)^{\log k/\epsilon^2}\right)$ (although MGH does not directly fall into the problem-class detailed in [3]). Since $OPT(G,H) \ge \frac{|E_G|}{2} = \Omega(n^2)$, we can obtain a quasi-PTAS by setting ϵ suitably. This also yields a PTAS for any fixed graph H.

Special graphs H. When H if bipartite, by Claim 4.2 it follows that one can obtain a 0.878-approximation algorithm for MGH using the Maxcut algorithm of Goemans and Williamson [18]. One can also obtain an approximation ratio better than 0.5 if H has a dense subgraph. Let $\rho_H = \max_U \rho(U)$, where $\rho(U) = \left(2|\{(u,v) \in E_H :\ u,v \in U\}|\right)/|U|^2$. Let $U^* \subseteq V_H$ be such that $\rho(U^*) = \rho_H$. The randomized algorithm that maps each node of G to a node of U^* chosen uniformly at random, returns a solution of expected value $\rho(U^*)|E_G|$ and is thus a ρ_H-approximation algorithm. This immediately implies an approximation ratio of at least 2/3 if H contains a triangle.

References

[1] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Channel assignment in wireless networks and classification of minimum graph homomorphism. In *ECCC: TR06-040*, 2006.

[2] M. Alekhnovich. More on average case vs approximation complexity. In *Proceedings, 44th FOCS*, pages 298–307, 2003.
[3] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *J. Comput. Syst. Sci.*, 58:193–210, 1999.
[4] L. Babai, P. Erdös, and S. Selkow. Random graph isomorphism. *SICOMP*, 9:628–635, 1980.
[5] G. Calinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *Journal of Computer and System Sciences*, 60:564–574, 2000.
[6] M. Charikar and A. Wirth. Maximizing quadratic programs: Extending Grothendieck's inequality. In *Proceedings, 45th FOCS*, pages 54–60, 2004.
[7] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. A maximal tractable class of soft constraints. *Journal of Artificial Intelligence Research*, 22:1–22, 2004.
[8] C. Cooper, M. Dyer, and A. Frieze. On Markov chains for randomly H-coloring a graph. *Journal of Algorithms*, 39:117–134, 2001.
[9] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SICOMP*, 23:864–894, 1994.
[10] E. D. Demaine, U. Feige, M. T. Hajiaghayi, and M. Salavatipour. Combination can be hard: Approximability of the unique coverage problem. In *Proceedings, 17th SODA*, pages 162–171, 2006.
[11] J. Díaz, M. J. Serna, and D. M. Thilikos. The complexity of restrictive H-coloring. In *Proceedings, 28th International Workshop (WG 2002)*, pages 126–137, 2002.
[12] M. E. Dyer and C. S. Greenhill. The complexity of counting graph homomorphisms. *Random Structures and. Algorithms*, 25:346–352, 2004.
[13] T. Feder and P. Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B*, 72:236–250, 1998.
[14] U. Feige. Relations between average case complexity and approximation complexity. In *Proceedings, 34th STOC*, pages 534–543, 2002.
[15] U. Feige and M. Langberg. The RPR^2 rounding technique for semidefinite programs. In *Proceedings, 28th ICALP*, pages 213–224, 2001.
[16] A. Freund and H. Karloff. A lower bound of $8/\left(7+\frac{1}{k-1}\right)$ on the integrality ratio of the Calinescu-Karloff-Rabani relaxation for multiway cut. *Information Processing Letters*, 75:43–50, 2000.
[17] A. Frieze and Colin McDiarmid. Algorithmic theory of random graphs. *Random Structures and Algorithms*, 10:5–42, 1997.
[18] 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.
[19] G. Gutin, A. Rafiey, A. Yeo, and M. Tso. Level of repair analysis and minimum cost homomorphisms of graphs. *Discrete Applied Mathematics*, 154:881–889, 2006.
[20] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48:92 – 110, 1990.
[21] P. Hell and J. Nešetřil. *Graphs and Homomorphisms.* Oxford Univ. Press, 2004.
[22] V. Kann. On the approximability of the maximum common subgraph problem. In *Proceedings, 9th STACS*, pages 377–388, 1992.
[23] D. Karger, P. Klein, C. Stein, M. Thorup, and N. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *M. of OR*, 29:436–461, 2004.
[24] J. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. *Journal of the ACM*, 49:616–639, 2002.
[25] P. Turán. On an extremal problem in graph theory. *Mat. Fiz. Lapok*, 48:436–452, 1941.