数理解析研究所講究録
第 1554 巻 2007 年 78-84

An Improved Approximation Algorithm for Maximum Edge 2-Coloring in Simple Graph

Graduate School of Science and Engineering, Tokyo Denki University
Zhi-Zhong Chen　　Ruka Tanahashi

Abstruct

We present a polynomial-time approximation algorithm for legally coloring as many edges of a given simple graph as possible using two colors. It achieves an approximation ratio of $\frac{468}{575}$. This improves on the previous best (trivial) ratio of $\frac{4}{5}$.

1 Introduction

Given a graph G and a natural number t, the *maximum edge t-coloring problem* (called MAX EDGE t-COLORING for short) is to find a maximum set F of edges in G such that F can be partitioned into at most t matchings of G. Motivated by call admittance issues in satellite based telecommunication networks, Feige et al. [1] introduced the problem and proved its APX-hardness. Their APX-hardness proof indeed shows that the problem remains APX-hard even if we restrict the input graph to a simple graph and fix the input integer t to 2. We call this restriction (special case) of the problem MAX SIMPLE EDGE 2-COLORING.

Since MAX EDGE t-COLORING and its special cases are hard, it is interesting to design approximation algorithms for them. As observed by Feige et al. [1], MAX EDGE t-COLORING is obviously a special case of the well-known maximum coverage problem (see [4]). Since the maximum coverage problem can be approximated by a greedy algorithm within a ratio of $1-(1-\frac{1}{t})^t$ [4], so can be MAX EDGE t-COLORING. In particular, the greedy algorithm achieves an approximation ratio of $\frac{3}{4}$ for MAX EDGE 2-COLORING which is the special case of MAX EDGE t-COLORING where the input number t is fixed to 2. Feige et al. [1] has improved the trivial ratio $\frac{3}{4}$ to $\frac{10}{13}$ by an LP approach. They also pointed out that for MAX SIMPLE EDGE 2-COLORING, the ratio $\frac{10}{13}$ can be further improved to $\frac{4}{5}$ by the following simple algorithm:

Input: A simple graph G.

1. Compute a maximum subgraph H of G such that the degree of each vertex in H is at most 2 and there is no 3-cycle in H. (Comment: This step can be done in $O(n^2m^3)$ time [3].)
2. Remove one edge from each odd cycle of H.

Output: H.

Kosowski et al. [7] also considered MAX SIMPLE EDGE 2-COLORING. They presented an approximation algorithm that achieves a ratio of $\frac{28\Delta-12}{35\Delta-21}$, where Δ is the maximum degree of a vertex in the input simple graph. This ratio can be arbitrarily close to the trivial ratio $\frac{4}{5}$ because Δ can be very large.

In this paper, we present a polynomial-time approximation algorithm for MAX SIMPLE EDGE 2-COLORING which achieves a ratio of $\frac{468}{575}$. To achieve this, we first design a randomized algorithm and then derandomize it. The analysis of our algorithm is quite nontrivial.

Kosowski et al. [7] showed that approximation algorithms for MAX SIMPLE EDGE 2-COLORING can be used to obtain approximation algorithms for certain packing problems and fault-tolerant guarding problems. Combining their reductions and our improved approximation algorithm for MAX SIMPLE EDGE 2-COLORING, we can obtain improved approximation algorithms for their packing problems and fault-tolerant guarding problems immediately.

2 Basic Definitions

Throughout the remainder of this paper, a graph means a simple undirected graph (i.e., it has neither parallel edges nor self-loops).

Let G be a graph. We denote the vertex set of G by $V(G)$, and denote the edge set of G by $E(G)$. The *degree* of a vertex v in G, denoted by $d_G(v)$, is the number of edges incident to v in G. A vertex v of G with $d_G(v)=0$ is called an *isolated vertex*. An *independent set* of G is a set

S of vertices of G such that no two vertices of S are adjacent in G.

A *cycle* in G is a connected subgraph of G in which each vertex is of degree 2. A *path* in G is a connected subgraph of G in which exactly two vertices are of degree 1 and the others are of degree 2. The *length* of a cycle or path C is the number of edges in C. A cycle of odd (respectively, even) length is called an *odd* (respectively, *even*) cycle. A *k-cycle* is a cycle of length k. Similarly, a *k^+-cycle* is a cycle of length at least k. A *path component* (respectively, *cycle component* of G) is a connected component of G that is a path (respectively, cycle). Note that an isolated vertex of G is not a path component of G.

For a function b mapping each vertex v of G to a nonnegative integer, a *b-matching* of G is a subgraph H of G such that $d_H(v) \le b(v)$ for all vertices v of H. When $b(v) \le 1$ for all vertices v of G, a b-matching of G is called a *matching* of G. The *size* of a b-matching M of G, denoted by $|M|$, is the number of edges in M. Given a graph G and a function b mapping each vertex v to a nonnegative integer, the *maximum b-matching problem* is to find a b-matching of G of maximum size. Similarly, given a graph G, the *maximum matching problem* is to find a maximum matching of G. G is *edge-2-colorable* if $E(G)$ can be partitioned into two matchings. Thus, Max Simple Edge 2-Coloring is the problem of finding a maximum edge 2-colorable subgraph in a given graph. Note that G is edge 2-colorable if and only if each connected component of G is an isolated vertex, a path, or an even cycle.

For a random event A, $\Pr[A]$ denotes the probability that A occurs. For two random events A and B, $\Pr[A \mid B]$ denotes the probability that A occurs given the occurrence of B. For a random variable X, $\mathcal{E}[X]$ denotes the expected value of X.

3 The Algorithm

Throughout this section, fix a graph G and a maximum edge 2-colorable subgraph Opt. Given G, our algorithm finds an edge 2-colorable subgraph of G as follows:

1. Compute a maximum subgraph H of G such that each connected component of H is an isolated vertex, a path, or a 4^+-cycle. (*Comment:* The set of vertices v of H with $d_H(v) \le 1$ is an independent set of G because of the maximality of H.)
2. Remove one (arbitrary) edge from each odd 7^+-cycle of H.
3. For $i \in \{0, 1\}$, let T_i be the set of vertices v of H with $d_H(v) = i$.
4. Let V_{5c} be the set of vertices on 5-cycles of H.
5. Construct an auxiliary graph A, where $V(A) = T_0 \cup T_1 \cup V_{5c}$ and $E(A)$ consists of those $\{u, v\} \in E(G)$ such that no connected component of H contains both u and v.
6. Compute a maximum b-matching M in A, where $b(v) = 2 - d_H(v)$ for each $v \in T_0 \cup T_1$ and $b(v) = 1$ for each $v \in V_{5c}$.
7. Choose one edge from each 5-cycle of H uniformly and independently at random and remove it from H.
8. Let M' be the set of all edges $\{u, v\}$ in M such that $d_H(u) + d_M(u) \le 2$ and $d_H(v) + d_M(v) \le 2$.
9. Add the edges in M' to H.
10. For each odd cycle C in H, select one edge in $E(C) \cap M'$ uniformly and independently at random and delete it from H.
11. Output H.

3.1 The First Analysis

For each $i \in \{1, 2, 7, 9, 10\}$, let H_i be the content of graph H immediately after Step i of our algorithm. Note that H_{10} is the output of our algorithm. Related to H_1, we define two sets and two numbers as follows:

- E_{5c} is the set of edges on the 5-cycles of H_1.
- $E_{\overline{5c}} = E(H_1) - E_{5c}$.
- n_{7c+} is the number of 7^+-cycles of H_1.
- n_{pc} is the number of path components of H_1.

Lemma 3.1 $|V(H_1) - V(A)| = |E_{\overline{5c}}| - 2n_{7c+} - n_{pc}$.

Proof. For each path component P of H_1, $|E(P)| = |V(P)| - 1$ and two vertices of P are contained in A. So, each path component of H_1 contributes 1 to the value of $|E_{\overline{5c}}| - |V(H_1) - V(A)|$. Similarly, for each cycle component C of H_1, $|E(C)| = |V(C)|$ and two vertices of C are contained in A. Thus, each cycle component of H_1 contributes 2 to the value of $|E_{\overline{5c}}| -$

$|V(H_1) - V(A)|$. This completes the proof of the lemma. □

Related to Opt, we define five sets of edges as follows:

- E^A_{opt} is the set of edges $\{u, v\}$ in Opt such that both u and v are vertices of A.
- $E^{\overline{A}}_{opt} = E(Opt) - E^A_{opt}$.
- $E^{A,5c}_{opt}$ (respectively, $E^{A,7c+}_{opt}$) is the set of edges $\{u, v\} \in E^A_{opt}$ such that some 5-cycle (respectively, path component) of H_2 contains both u and v. (*Comment:* By the maximality of $E(H_1)$, there is no edge $\{u, v\} \in E^A_{opt}$ such that some path component of H_1 contains both u and v. So, the endpoints of each edge in $E^{A,7c+}_{opt}$ must appear on the same 7^+-cycle of H_1.)
- $E^{A,ex}_{opt} = E^A_{opt} - (E^{A,5c}_{opt} \cup E^{A,7c+}_{opt})$

Lemma 3.2 $|E^A_{opt}| \geq |E(Opt)| - 2|E_{\overline{5c}}| + 4n_{7c+} + 2n_{pc}$.

Proof. Since, each vertex can be adjacent to at most two edges in Opt, $|E(Opt) - E^A_{opt}| \leq 2|V(H_1) - V(A)|$. So, the lemma follows from Lemma 3.1 immediately. □

Corollary 3.3 $|E^{A,ex}_{opt}| \geq |E(Opt)| - 2|E_{\overline{5c}}| + 4n_{7c+} + 2n_{pc} - |E^{A,5c}_{opt}| - |E^{A,7c+}_{opt}|$.

Proof. Obviously, $|E^{A,ex}_{opt}| = |E^A_{opt}| - (|E^{A,5c}_{opt}| + |E^{A,7c+}_{opt}|)$. So, the corollary follows from Lemma 3.2 immediately. □

Lemma 3.4 *For each edge* $e \in M$, $\Pr[e \in E(H_{10})] \geq \frac{8}{75}$.

Proof. Fix an arbitrary edge $e = \{u, v\}$ in M. We distinguish three cases as follows:

Case 1: Both $u \in V_{5c}$ and $v \in V_{5c}$. In this case, since $\Pr[d_{H_7}(u) = 1] = \frac{2}{5}$, $\Pr[d_{H_7}(v) = 1] = \frac{2}{5}$, $d_M(u) \leq 1$, and $d_M(v) \leq 1$, we have $\Pr[e \in M'] = \frac{4}{25}$. Thus, it remains to show that $\Pr[e \in E(H_{10}) \mid e \in M'] \geq \frac{2}{3}$. Assume that $e \in M'$. If no odd cycle of H_9 contains e, then we are done. So, further assume that some odd cycle C of H_9 contains e. We claim that C contains at least three edges of M'. For a contradiction, assume that C contains only one edge e' of M' other than e. Obviously, if we delete e and e' from C, we obtain two paths P_1 and P_2 both of which are connected components of H_7. Moreover, one of u and v is an endpoint of P_1 and the other is an endpoint of P_2. Now, since $u \in V_{5c}$ and $v \in V_{5c}$, P_1 and P_2 must have been obtained in Step 7 by deleting one edge from each 5-cycle of H_1. So, both P_1 and P_2 are of length 4. However, this implies that the length of C is 10 which is even, a contradiction. Hence, the claim holds. By the cliam, we have $\Pr[e \in E(H_{10}) \mid e \in M'] \geq \frac{2}{3}$ immediately.

Case 2: Exactly one of u and v is contained in V_{5c}. We assume that $u \in V_{5c}$ but $v \notin V_{5c}$; the other case is similar. Then, $\Pr[d_{H_7}(u) = 1] = \frac{2}{5}$ and $d_M(u) \leq 1$. Moreover, $v \in T_0$ or $v \in T_1$. In the former case, $d_{H_7}(v) = 0$ and $d_M(v) \leq 2$. In the latter case, $d_{H_7}(v) = 1$ and $d_M(v) \leq 1$. So, in both cases, $d_{H_7}(v) + d_M(v) \leq 2$. Consequently, $\Pr[e \in M'] \geq \frac{2}{5} \cdot 1 = \frac{2}{5}$. Thus, it suffices to show that $\Pr[e \in E(H_{10}) \mid e \in M'] \geq \frac{1}{2}$. Assume that $e \in M'$. If no odd cycle of H_9 contains e, then we are done. On the other hand, if some odd cycle C of H_9 contains e, then the assumption $u \in V_{5c}$ guarantees that C contains at least two edges of M' and hence $\Pr[e \in E(H_{10}) \mid e \in M'] \geq \frac{1}{2}$.

Case 3: Both $u \notin V_{5c}$ and $v \notin V_{5c}$. Then, as discussed in Case 2 about v, we have $d_{H_7}(u) + d_M(u) \leq 2$ and $d_{H_7}(v) + d_M(v) \leq 2$. So, $\Pr[e \in M'] = 1$. Thus, it suffices to show that $\Pr[e \in E(H_{10}) \mid e \in M'] \geq \frac{1}{2}$. Assume that $e \in M'$. If no odd cycle of H_9 contains e, then we are done. So, further assume that some odd cycle C of H_9 contains e. We claim that C contains at least two edges of M'. For a contradiction, assume that the claim is false. Then, the path obtained from C by deleting e is a connected component of H_2. However, this contradicts the construction of graph A in Step 5. Thus, the claim holds. Consequently, $\Pr[e \in E(H_{10}) \mid e \in M'] \geq \frac{1}{2}$. □

By Lemma 3.4 and the algorithm, we have the following corollary immediately:

Corollary 3.5 $\mathcal{E}[|E(H_{10})|] \geq |E_{\overline{5c}}| - n_{7c+} + \frac{4}{5}|E_{5c}| + \frac{8}{75}|M|$.

Lemma 3.6 $|M| \geq |E^{A,ex}_{opt}|/2$.

Proof. Let M'' be a maximum matching in graph A. Since Opt has no odd cycle, $E^{A,ex}_{opt}$ can be partitioned into two matchings of A. So, $|M''| \geq |E^{A,ex}_{opt}|/2$. On the other hand, since M is a maximum b-matching of A with $b(v) \geq 1$ for each $v \in V(A)$, we have $|M| \geq |M''|$. Thus, the lemma holds. □

Theorem 3.7 $\mathcal{E}[|E(H_{10})|] \geq \frac{146}{175}|E(Opt)| + \frac{2}{105}|E_{5c}| - \frac{4}{75}|E_{opt}^{A,5c}| - \frac{4}{75}|E_{opt}^{A,7c+}|$.

PROOF. Combining Corollary 3.5 and Lemma 3.6, we have

$$\mathcal{E}[|E(H_{10})|] \geq |E_{\overline{5c}}| - n_{7c+} + \frac{4}{5}|E_{5c}| + \frac{4}{75}|E_{opt}^{A,ex}|.$$

So, by Corollary 3.3 and a simple calculation, we have

$$\mathcal{E}[|E(H_{10})|] \geq \frac{4}{75}|E(Opt)| + \frac{67}{75}|E_{\overline{5c}}| + \frac{4}{5}|E_{5c}| - \frac{59}{75}n_{7c+} + \frac{8}{75}n_{pc} - \frac{4}{75}|E_{opt}^{A,5c}| - \frac{4}{75}|E_{opt}^{A,7c+}|.$$

Consequently, since $n_{7c+} \leq (|E(H_1)| - |E_{5c}|)/7$ and $|E_{\overline{5c}}| = |E(H_1)| - |E_{5c}|$, we have

$$\mathcal{E}[|E(H_{10})|] \geq \frac{4}{75}|E(Opt)| + \frac{82}{105}|E(H_1)| + \frac{2}{105}|E_{5c}| - \frac{4}{75}|E_{opt}^{A,5c}| - \frac{4}{75}|E_{opt}^{A,7c+}|.$$

Now, since $|E(H_1)|$ is at least as large as $|E(Opt)|$, the theorem follows. □

The following corollary shows that our algorithm achieves an expected ratio of $\frac{304}{375}$.

Corollary 3.8 $\mathcal{E}[|E(H_{10})|] \geq \frac{304}{375}|E(Opt)| + \frac{2}{125}|E_{\overline{5c}}|$.

PROOF. Obviously, $|E_{opt}^{A,7c+}| \leq n_{7c+} \leq \frac{1}{7}|E_{\overline{5c}}|$. Moreover, since Opt cannot contain a 5-cycle, $E_{opt}^{A,5c}$ contains at most four edges $\{u,v\}$ with $u \in V(C)$ and $v \in V(C)$ for each 5-cycle C of H_1. Consequently, $|E_{opt}^{A,5c}| \leq \frac{4}{5}|E_{5c}| = \frac{4}{5}|E(H_1)| - \frac{4}{5}|E_{\overline{5c}}|$. Also recall that $|E(H_1)| \geq |E(Opt)|$. Now, by the last inequality in the proof of Theorem 3.7, the corollary follows. □

In the next subsection, we will give another analysis of the algorithm and combine it with the analysis in this section to obtain a better ratio.

3.2 The Second Analysis

Let K be the graph with vertex set $V(A)$ and edge set $E_{opt}^{A} - (E_{opt}^{A,5c} \cup E_{opt}^{A,7c+})$.

Lemma 3.9 *There are at least* $\frac{5}{4}|E_{opt}^{A,5c}|$ *vertices* $v \in V_{5c}$ *with* $d_K(v) < 2$.

PROOF. Fix an arbitrary 5-cycle C of H_1. Let F be the number of edges $\{u,v\} \in E_{opt}^{A,5c}$ with $\{u,v\} \subseteq V(C)$. Let W be the set of the endpoints of the edges in F. Obviously, for each $v \in W$, $d_K(v) < 2$. We claim that $|W| \geq \frac{5}{4}|F|$. To see this, first observe that we always have $|W| \geq |F| + 1$. Moreover, $|F| \leq 4$ because Opt cannot contain a 5-cycle. Thus, the claim holds. The claim implies the lemma immediately because summing up $\frac{5}{4}|F|$ over all 5-cycles C of H_1 yields the bound $\frac{5}{4}|E_{opt}^{A,5c}|$. □

Besides Lemma 3.6, we have another lower bound on $|M|$.

Lemma 3.10 $|M| \geq |E_{opt}^{A}| - |E_{5c}| - 2n_{7c+} - 2n_{pc} + \frac{1}{4}|E_{opt}^{A,5c}| + |E_{opt}^{A,7c+}|$.

PROOF. Let h be the number of vertices $v \in V_{5c}$ with $d_K(v) = 2$. By Lemma 3.9, $h \leq |E_{5c}| - \frac{5}{4}|E_{opt}^{A,5c}|$.

Let ℓ be the number of vertices $v \in T_1$ with $d_K(v) = 2$. We claim that $\ell \leq 2(n_{7c+} + n_{pc}) - 2|E_{opt}^{A,7c+}|$. To see this, first observe that $|T_1| = 2(n_{7c+} + n_{pc})$. Moreover, if $\{u,v\} \in E_{opt}^{A,7c+}$, then both $d_K(u) \leq 1$ and $d_K(v) \leq 1$. Now, since no two edges of $E_{opt}^{A,7c+}$ can share an endpoint, the claim holds.

Obviously, if we modify K by removing one edge from each $v \in V_{5c} \cup T_1$ with $d_K(v) = 2$, we obtain a b-matching of A. So, since M is a maximum b-matching of A, we have

$$|M| \geq |E_{opt}^{A}| - |E_{opt}^{A,5c}| - |E_{opt}^{A,7c+}| - h - \ell.$$

Thus, by the aforementioned bounds on h and ℓ, the lemma holds. □

Theorem 3.11 $\mathcal{E}[|E(H_{10})|] \geq \frac{82}{105}|E(Opt)| + \frac{2}{105}|E_{5c}| + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,7c+}|$.

PROOF. Combining Corollary 3.5 and Lemma 3.10, we have

$$\mathcal{E}[|E(H_{10})|] \geq |E_{\overline{5c}}| - \frac{91}{75}n_{7c+} + \frac{52}{75}|E_{5c}| + \frac{8}{75}|E_{opt}^{A}| - \frac{16}{75}n_{pc} + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,7c+}|.$$

So, by Lemma 3.2 and a simple calculation, we have

$$\mathcal{E}[|E(H_{10})|] \geq \frac{8}{75}|E(Opt)| + \frac{59}{75}|E_{\overline{5c}}| + \frac{52}{75}|E_{5c}| - \frac{59}{75}n_{7c+} + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,7c+}|.$$

Consequently, since $n_{7c+} \le |E_{\overline{5c}}|/7$, we have

$$\mathcal{E}[|E(H_{10})|] \ge \frac{8}{75}|E(Opt)| + \frac{354}{525}|E_{\overline{5c}}| + \frac{52}{75}|E_{5c}| + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,7c+}|.$$

Since $|E_{5c}| + |E_{\overline{5c}}| = |E(H_1)|$, we have

$$\mathcal{E}[|E(H_{10})|] \ge \frac{8}{75}|E(Opt)| + \frac{354}{525}|E(H_1)| + \frac{10}{525}|E_{5c}| + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,7c+}|.$$

Now, since $|E(H_1)| \ge |E(Opt)|$, the theorem follows. □

Corollary 3.12 $\mathcal{E}[|E(H_{10})|] \ge \frac{1258}{1575}|E(Opt)| + \frac{2}{105}|E_{5c}|$.

PROOF. By Theorem 3.7, we have

$$\frac{1}{3}\mathcal{E}[|E(H_{10})|] \ge \frac{146}{525}|E(Opt)| + \frac{2}{315}|E_{5c}| - \frac{4}{225}|E_{opt}^{A,5c}| - \frac{4}{225}|E_{opt}^{A,7c+}|.$$

On the other hand, by Theorem 3.11, we have

$$\frac{2}{3}\mathcal{E}[|E(H_{10})|] \ge \frac{164}{315}|E(Opt)| + \frac{4}{315}|E_{5c}| + \frac{4}{225}|E_{opt}^{A,5c}| + \frac{16}{225}|E_{opt}^{A,7c+}|.$$

So, summing up the left sides and the right sides of the above two inequalities respectively, we have

$$\mathcal{E}[|E(H_{10})|] \ge \frac{1258}{1575}|E(Opt)| + \frac{2}{105}|E_{5c}|.$$

□

Theorem 3.13 $\mathcal{E}[|E(H_{10})|] \ge \frac{468}{575}|E(Opt)|$.

PROOF. By Corollary 3.8, we have

$$\frac{25}{46}\mathcal{E}[|E(H_{10})|] \ge \frac{152}{345}|E(Opt)| + \frac{1}{115}|E_{\overline{5c}}|.$$

By Corollary 3.12, we have

$$\frac{21}{46}\mathcal{E}[|E(H_{10})|] \ge \frac{629}{1725}|E(Opt)| + \frac{1}{115}|E_{5c}|.$$

So, summing up the left sides and the right sides of the above two inequalities respectively, we have

$$\mathcal{E}[|E(H_{10})|] \ge \frac{463}{575}|E(Opt)| + \frac{1}{115}|E(H_1)| \ge \frac{468}{575}|E(Opt)|.$$

□

3.3 Derandomization

Our algorithm makes random choices only in Step 7 and 10. To derandomize Step 10, we just modify it as follows:

10. For each odd cycle C in H, select an arbitrary edge of C and delete it from H.

Because the input graph is unweighted, it does not matter which edge is deleted from each odd cycle in Step 10. So, it should be clear that the above modification of Step 10 does not affect the approximation ratio achieved by the algorithm.

In Step 7, we make a random choice for each 5-cycle. In our above analysis of the algorithm, only the proof of Lemma 3.4 is based on the mutual independence between these random choices. Indeed, by carefully inspecting the proof, we can see that the proof is still valid even if the random choices made in Step 7 are only pairwise independent. So, we can derandomize it via conventional approaches. Therefore, we have the following theorem:

Theorem 3.14 *There is an $O(n^2m^3)$-time approximation algorithm for* MAX SIMPLE EDGE 2-COLORING *achieving a ratio of $\frac{468}{575}$, where n (respectively, m) is the number of vertices (respectively, edges) in the input graph.*

PROOF. We estimate the running time of the derandomize algorithm as follows. Step 1 can be done in $O(n^2m^3)$ time [3]. Obviously, Steps 2 through 4 can be done in $O(n)$ time. Step 5 can be trivially done in $O(n^2)$ time. Since $b(v) \le 2$ for each vertex v, Step 6 can be done in $O(\sqrt{n}m)$ time [2]. In Step 7, we need to generate $O(n)$ pairwise independent random integers. A conventional way to do this uses two random seeds s_1 and s_2 both of value $O(n)$. So, the sample space of (s_1, s_2) is of size $O(n^2)$. For each sample (s_1, s_2) in the space, we perform Steps 8 through 11 to obtain an output $H(s_1, s_2)$. This takes a total time of (n^3) because Steps 8 through 11 can be done in $O(n)$ time. We then find the sample (s_1, s_2) in $O(n^2)$ time such that $|H(s_1, s_2)|$ is maximized, and further output $H(s_1, s_2)$. □

4 An Application

Let G be a graph. An *edge cover* of G is a set F of edges of G such that each vertex of G is incident to at least one edge of F. For a natural number k, a *[1, Δ]-factor k-packing* of G is a collection of k disjoint edge covers of G. The *size* of a $[1, \Delta]$-factor k-packing $\{F_1, ..., F_k\}$ of G is $|F_1| + \cdots + |F_k|$. The problem of deciding whether a given graph has a $[1, \Delta]$-factor k-packing was considered in [5, 6]. In [7], Kosowski *et al.* defined the *minimum [1, Δ]-factor k-packing problem* (MIN-k-FP) as follows: Given a graph G, find a $[1, \Delta]$-factor k-packing of G of minimum size or decide that G has no $[1, \Delta]$-factor k-packing at all.

According to [7], MIN-2-FP is of special interest because it can be used to solve a fault tolerant variant of the guards problem in grids (which is one of the art gallery problems [8, 9]). Indeed, they proved the following:

Lemma 4.1 *If* MAX SIMPLE EDGE 2-COLORING *admits an approximation algorithm A achieving a ratio of α, then* MIN-2-FP *admits an approximation algorithm B ahieving a ratio of $2 - \alpha$. Moreover, if the time complexity of A is $T(n)$, then the time complexity of B is $O(T(n))$.*

So, by Theorem 3.14, we have the following immediately:

Lemma 4.2 *There is an $O(n^2m^3)$-time approximation algorithm for* MIN-2-FP *achieving a ratio of $\frac{682}{575}$, where n (respectively, m) is the number of vertices (respectively, edges) in the input graph.*

Previously, the best ratio achieved by a polynomial-time approximation algorithm for MIN-2-FP was $\frac{6}{5}$ [7], although MIN-2-FP admits a polynomial-time approximation algorithm achieving a ratio $\frac{42\Delta-30}{35\Delta-21}$, where Δ is the maximum degree of a vertex in the input graph [7].

5 Final Remarks

When the input graph is restricted to simple graphs, MAX EDGE t-COLORING is easier to approximate for large values of t. In more detail, the following algorithm achieves a ratio of $\frac{t}{t+1}$:

Input: A simple graph G and a natural number t.

1. Compute a maximum b-matching M of G, where $b(v) = t$ for all vertices v of G. (*Comment:* Since an optimal solution F can be partitioned into t matchings of G, it is a b-matching of G. Hence, $|E(M)|$ is at least as large as the size of an optimal solution.)
2. Partition $E(M)$ into $t + 1$ matchings $M_1, ..., M_{t+1}$. (*Comment:* By Vizing's Theorem [10], this can be done in polynomial time.)

Output: The largest t matchings among $M_1, ..., M_{t+1}$.

When $t = 2$, the above algorithm only achieves a ratio of $\frac{2}{3}$. The simple algorithm for MAX SIMPLE EDGE 2-COLORING pointed out by Feige *et al.* (see Section 1) can be viewed as an improvement over the above algorithm. Our new algorithm can be viewed as a furhter improvement.

References

[1] U. Feige, E. Ofek, and U. Wieder. Approximating Maximum Edge Coloring in Multigraphs. *Proceedings of the 10th International Conference on Integer Programming and Combinational Optimization* (IPCO), Lecture Notes in Computer Science, **2462** (2002) 108-121.

[2] H.N. Gabow. An Efficient Reduction Technique for Degree-Constrained Subgraph and Bidirected Network Flow Problems. *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, 448-456, 1983.

[3] D. Hartvigsen. *Extensions of Matching Theory*. Ph.D. Thesis, Carnegie-Mellon University, 1984.

[4] D. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, Boston, 1997.

[5] D.P. Jacobs and R.E. Jamison. Complexity of Recognizing Equal Unions in Families of Sets. *Journal of Algorithms*, **37** (2000) 495-504.

[6] K. Kawarabayashi, H. Matsuda, Y. Oda, and K. Ota. Path Factors in Cubic Graphs. *Journal of Graph Theory,* **39** (2002) 188-193.

[7] A. Kosowski, M. Malafiejski, and P. Zylinski. Packing Edge Covers in Graphs of Small Degree. *Manuscript,* 2006.

[8] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, 1987.

[9] J. Urrutia. *Art Gallery and Illumination Problems.* Handbook on Computational Geometry, Elsevier Science, Amsterdam, 2000.

[10] V.G. Vizing. On an Estimate of the Chromatic Class of a P-Graph (in Russian). *Diskret. Analiz,* **3** (1964) 23-30.