

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

Zhi-Zhong Chen [†] Ruka Tanahashi [‡] Lusheng Wang [§]

Abstract

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. They also observed that 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 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 $\frac{3}{4} = 0.75$ to $\frac{10}{13} \approx 0.769$ by an LP approach.

The APX-hardness proof for MAX EDGE t -COLORING given by Feige et al. [1] 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. Feige et al. [1] also pointed out that for MAX SIMPLE EDGE 2-COLORING, an approximation ratio of $\frac{4}{5}$ can be achieved 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 $O(n^2m^2)$ time [3].)

*A preliminary version of this paper appeared in the *Proceedings of 3rd International Conference on Algorithmic Aspects in Information and Management*, Lecture Notes in Computer Science, Vol. 4508, pp. 27–36, 2007.

[†]Supported in part by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture of Japan, under Grant No. 17500012. Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan. Phone: +81-49-296-2911 x 2105. Fax: +81-49-296-7072. Email: chen@r.dendai.ac.jp.

[‡]Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan.

[§]Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong.

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.

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*. For a subset U of $V(G)$, let $G[U]$ denote the graph (U, E_U) where E_U consists of all edges $\{u, v\}$ of G with $u \in U$ and $v \in U$. We call $G[U]$ the *subgraph of G induced by U* . An *independent set* of G is a subset U of $V(G)$ such that $G[U]$ has no edge.

A *cycle* in G is a connected subgraph of G in which each vertex is of degree 2. A *Hamiltonian cycle* of G is a cycle C of G with $V(C) = V(G)$. 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 . A *path-cycle cover* of G is a subgraph H of G such that $V(H) = V(G)$ and $d_H(v) \leq 2$ for every $v \in V(H)$. Note that each connected component of a path-cycle cover of G is a single vertex, path, or cycle. A path-cycle cover C of G is *triangle-free* if C does not contain a 3-cycle. A path-cycle cover C of G is *maximum* if the number of edges in C is maximized over all path-cycle covers 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 $d_H(v) \leq b(v)$ for all vertices v of H . When $b(v) \leq 1$ for all vertices v of G , a b -matching of G is called a *matching* of G . Moreover, when $b(v) \leq 2$ for all vertices v of G , a b -matching of G is a path-cycle cover 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 matching of G of maximum size. 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 of G . For convenience, for each path-cycle cover K of G , we define two numbers as follows:

- For each $i \in \{0, 1\}$, let $n_i(K)$ be the number of vertices v in K with $d_K(v) = i$.

Like the simple algorithm described in Section 1, our algorithm starts by performing the following step:

1. Compute a maximum triangle-free path-cycle cover H of G .

Since $|E(H)| \geq |E(Opt)|$, it suffices to modify H into an edge-2-colorable subgraph of G without significantly decreasing the number of edges in H . The simple algorithm achieves an approximation ratio of $\frac{4}{5}$ because it simply removes an arbitrary edge from each 5^+ -cycle in H . In order to improve this ratio, we have to treat 5-cycles in H more carefully. So, we first define two useful operations on 5-cycles of H as follows.

Suppose that C is a 5-cycle of H . A *killer* for C is an ordered pair (u, v) of vertices of G such that $\{u, v\} \in E(G)$, $u \in V(C)$, $G[V(C) - \{u\}]$ has a Hamiltonian cycle, and $d_H(v) \leq 1$ (hence $v \notin V(C)$). *Killing* C in H by a killer (u, v) for it is the operation of modifying H by performing the following three steps in turn:

- (1) Delete all the edges of C from H .
- (2) Add the edges of a Hamiltonian cycle of $G[V(C) - \{u\}]$ to H .
- (3) Add the edge $\{u, v\}$ of G to H .

Killing C in H means killing it by an arbitrary killer for it. Note that killing C does not change the number of edges in H and decreases the number of 5-cycles in H by 1. So, we say that C is *superb* if there is a killer for it.

Similarly, suppose that C and D are two 5-cycles of H . A *killer* for the ordered pair (C, D) is an ordered pair (u, v) of vertices of G such that $\{u, v\} \in E(G)$, $u \in V(C)$, $v \in V(D)$, and $G[V(C) - \{u\}]$ has a Hamiltonian cycle. *Killing* (C, D) in H by a killer (u, v) for it is the operation of modifying H by performing the following four steps in turn:

- (a) Delete all the edges of C from H .

- (b) Delete one (arbitrary) edge incident to v from H .
- (c) Add the edges of a Hamiltonian cycle of $G[V(C) - \{u\}]$ to H .
- (d) Add the edge $\{u, v\}$ of G to H .

Killing (C, D) in H means killing it by an arbitrary killer for it. Note that killing (C, D) decreases the number of 5-cycles in H by 2 and decreases the number of edges in H by 1. So, we say that (C, D) is a *good* pair if there is a killer for it.

Using the above two operations, our algorithm then proceeds to modifying H by performing the following steps:

2. Repeat killing an (arbitrary) superb 5-cycle in H , until there is none in H . (*Definition:* After this step, let n , m , n_{7c^+} , and n_{pc} be the numbers of vertices, edges, odd 7^+ -cycles, and path components in H , respectively. Note that $m \geq |E(Opt)|$.)
3. Repeat killing an (arbitrary) good pair of 5-cycles in H , until there is none in H . (*Definition:* Let n_{gp} be the number of good pairs killed in this step.)
4. Remove one (arbitrary) edge from each odd 7^+ -cycle of H .
5. Repeat killing an (arbitrary) superb 5-cycle in H , until there is none in H . (*Definition:* After this step, let n_{5c} be the number of vertices on 5-cycles in H .)

Before proceeding to the other steps of the algorithm, we first prove three lemmas that will be very useful later.

Lemma 3.1 $n_{gp} + n_{7c^+} \leq \frac{1}{7}(m - 5n_{5c})$.

PROOF. For convenience, let H_2 be the content of graph H immediately after Step 2. Note that the algorithm never creates a new odd cycle in Steps 3, 4, and 5. Thus, each 5-cycle still remaining in H immediately after Step 5 is also a 5-cycle in H_2 , each good pair killed in Step 3 is also a good pair in H_2 , and each 7^+ -cycle destroyed in Step 4 (by removing one edge from it) is also a 7^+ -cycle in H_2 . Moreover, a 5-cycle contains exactly 5 edges, a 7^+ -cycle contains at least 7 edges, and the total number of edges in a good pair of 5-cycles is 10. Therefore, $5n_{5c} + 10n_{gp} + 7n_{7c^+} \leq |E(H_2)| = m$. This establishes the lemma. $\square$

Lemma 3.2 *Immediately after Step 5, the following statements hold:*

- (1) $n - n_0(H) - n_1(H) = m - n_{pc} - 2n_{gp} - 2n_{7c^+}$.
- (2) $|E(H)| = m - n_{gp} - n_{7c^+}$.
- (3) H is a path-cycle cover of G in which each odd cycle is a 5-cycle.
- (4) The number of path components in H is $n_{pc} + n_{gp} + n_{7c^+}$.

PROOF. Statement (3) is clear because the algorithm never creates a new odd cycle in Steps 3, 4, and 5. For convenience, let $p(H)$ denote the number of path components in H . To prove the other statements, let us see how the values of $n - n_0(H) - n_1(H)$, $|E(H)|$, and $p(H)$ change in Steps 3, 4, and 5. Immediately before Step 3, $|E(H)| = m$, $p(H) = n_{pc}$, and $n - n_0(H) - n_1(H) = m - n_{pc}$ because $n_1(H) = 2n_{pc}$ and the number of vertices on a path is 1 plus the number of edges on the path. In Step 3, killing a good pair in H decreases the value of $n - n_0(H) - n_1(H)$ by 2, decreases that of $|E(H)|$ by 1, increases that of $p(H)$ by 1, and does not create a new odd cycle in H . Thus, immediately after Step 3, $n - n_0(H) - n_1(H) = m - n_{pc} - 2n_{gp}$, $|E(H)| = m - n_{gp}$, $p(H) = n_{pc} + n_{gp}$, and the number of odd 7^+ -cycles in H is n_{7c^+} . In Step 4, removing one edge from an odd 7^+ -cycle of H decreases the value of $n - n_0(H) - n_1(H)$ by 2, decreases that of $|E(H)|$ by 1, and increases that of $p(H)$ by 1. Hence, immediately after Step 4, $n - n_0(H) - n_1(H) = m - n_{pc} - 2n_{gp} - 2n_{7c^+}$, $|E(H)| = m - n_{gp} - n_{7c^+}$, and $p(H) = n_{pc} + n_{gp} + n_{7c^+}$. Note that during Steps 3, 4, and 5, the algorithm does not create a new isolated vertex or a new odd cycle in H . So, if a superb 5-cycle C in H is killed by a killer (u, v) in Step 5, then v cannot be an isolated vertex of H or else C would have been killed in Step 2. Consequently, if a superb 5-cycle C in H is killed by a killer (u, v) in Step 5, then $d_H(v) = 1$ immediately before the killing and hence the killing does not change the values of $n - n_0(H) - n_1(H)$, $|E(H)|$, and $p(H)$. Therefore, immediately after Step 5, $n - n_0(H) - n_1(H) = m - n_{pc} - 2n_{gp} - 2n_{7c^+}$, $|E(H)| = m - n_{gp} - n_{7c^+}$, and $p(H) = n_{pc} + n_{gp} + n_{7c^+}$. This completes the proof of the lemma. $\square$

Lemma 3.3 *Immediately after Step 5, if C is a 5-cycle in H such that Opt contains a cycle C' with $V(C') \subseteq V(C)$, then C' is a 4-cycle and the unique vertex u in $V(C) - V(C')$ has no neighbor v in Opt such that $d_H(v) \leq 1$ or v appears on a 5-cycle of H other than C .*

PROOF. Suppose that immediately after Step 5, C is a 5-cycle in H such that Opt contains a cycle C' with $V(C') \subseteq V(C)$. Then, since Opt contains no odd cycle, C' must be a 4-cycle. Let u be the unique vertex in $V(C) - V(C')$. If u has a neighbor v in Opt such that $d_H(v) \leq 1$ after Step 5, then C would have been killed in Step 5, a contradiction. Similarly, if u has a neighbor v in Opt such that v appears on a 5-cycle of H after Step 5, then C would have been killed in Step 3, a contradiction. So, the lemma holds. $\square$

We now state the basic ideas behind the rest of our algorithm. By Lemma 3.1 and Statement (2) in Lemma 3.2, H still has at least $\frac{6}{7}m + \frac{5}{7}n_{5c}$ edges immediately after Step 5. Suppose that we delete one arbitrary edge from each 5-cycle still remaining in H (so that it becomes edge-2-colorable). After this deletion process, H has at least $\frac{6}{7}m - \frac{2}{7}n_{5c}$ edges, which is significantly larger than $\frac{4}{5}m$ ($\geq \frac{4}{5}|E(Opt)|$) if n_{5c} is small.

Unfortunately, n_{5c} may be very large. Fortunately, after the deletion process (which removes one edge from each 5-cycle still remaining in H), G may have some edges e such that both endpoints of e are of degree at most 1 in H and appear in different connected components of H . For convenience, let us call such an edge of G an *available bridge*. Clearly, an available bridge can be added to H without invalidating the edge-2-colorability of H . Of course, we do not want to add only one available bridge to H but want to add many available bridges to H simultaneously. Unfortunately, two available bridges may conflict each other, i.e., adding them to H simultaneously invalidates the edge-2-colorability of H . In order to find a large set of nonconflicting available bridges, we may first

construct an auxiliary graph B whose vertices are the endpoints of paths of H (after the deletion process) and whose edges are the available bridges, and then find a maximum b -matching N in B where $b(v) = 2 - d_H(v)$ for each vertex v of B . We now add the edges of N to H simultaneously, obtaining a graph H' whose connected components are paths, even cycles, or odd cycles. Each odd cycle of H' must contain at least two edges of N and we can delete exactly one edge of N from each odd cycle of H' . The resulting H' is now edge-2-colorable and contains at least $\frac{6}{7}m - \frac{2}{7}n_{5c} + \frac{1}{2}|N|$ edges.

Unfortunately, $|N|$ may be small and hence the above basic ideas do not work. So, we modify the basic ideas as follows. First, we construct an auxiliary graph A (instead of constructing B as above), where $V(A)$ consists of those vertices v in H such that v is of degree at most 1 in H or appears on a 5-cycle in H and $E(A)$ consists of those edges $\{u, v\}$ of G such that u and v belong to different connected components of H . We further construct a maximum b -matching M in A (instead of constructing N as above), where $b(v) = 1$ if v is a vertex of a 5-cycle in H while $b(v) = 2 - d_H(v)$ otherwise. Obviously, $|M|$ may be much larger than $|N|$, especially when there are a lot of 5-cycles in H . In particular, we can easily prove that $|M| \geq \frac{1}{2}|E(\text{Opt}) \cap E(A)|$ (see Lemma 3.8 below). Now, it comes the *modified* deletion process: For each 5-cycle C still remaining in H , instead of deleting one arbitrary edge from C , we select one edge from C uniformly at random and delete it from H . As before, we call an edge e of A an *available bridge* if both endpoints of e become of degree at most 1 in H after the modified deletion process. Obviously, each edge of M becomes an available bridge with probability at least $\frac{4}{25}$. Thus, we can expect at least $\frac{2}{25}|E(\text{Opt}) \cap E(A)|$ available bridges in M , which can then be added to H as before to obtain H' .

By the discussion in the last paragraph, if $|E(\text{Opt}) \cap E(A)|$ is significantly large, then the above modified ideas lead to a randomized algorithm whose output can be expected to contain significantly more than $\frac{6}{7}m - \frac{2}{7}n_{5c}$ edges. So, the problematic case happens when $|E(\text{Opt}) \cap E(A)|$ is not significantly large. A large portion of the analysis of our algorithm is devoted to this case. Intuitively speaking, we can prove that in this problematic case, $\frac{6}{7}m - \frac{2}{7}n_{5c}$ should be significantly better than $\frac{4}{5}|E(\text{Opt})|$.

We next give a formal description of the rest of our algorithm.

6. For $i \in \{0, 1\}$, let T_i be the set of vertices v of H with $d_H(v) = i$.
7. Let V_{5c} be the set of vertices on 5-cycles of H .
8. 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 .
9. 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 $v \in V_{5c}$.
10. Choose one edge from each 5-cycle of H uniformly and independently at random and remove it from H .
11. Let M' be the set of all edges $\{u, v\}$ in M such that $d_H(u) + d_M(u) \leq 2$ and $d_H(v) + d_M(v) \leq 2$.
12. Add the edges in M' to H . (*Comment:* Each connected component of H is an isolated vertex, a path, or a cycle.)

13. 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 .
14. Output H .

3.1 The First Analysis

For each $i \in \{3, 5, 10, 12, 13\}$, let H_i be the content of graph H immediately after Step i of our algorithm. Note that H_{13} is the output of our algorithm. Related to Opt , we define three sets of edges as follows:

- E_{opt}^A is the set of edges $\{u, v\}$ in Opt such that both u and v are vertices of A .
- $E_{opt}^{A,5c}$ is the set of edges $\{u, v\} \in E_{opt}^A$ such that some 5-cycle of H_5 contains both u and v .
- $E_{opt}^{A,pc}$ is the set of edges $\{u, v\} \in E_{opt}^A$ such that some path component of H_5 contains both u and v .

Lemma 3.4 $|E_{opt}^A| \geq |E(Opt)| - 2m + 2n_{pc} + 4(n_{gp} + n_{7c+}) + 10n_{5c}$.

PROOF. Since each vertex can be adjacent to at most two edges in Opt , $|E(Opt) - E_{opt}^A| \leq 2(n - |V(A)|) = 2(n - |T_0| - |T_1| - 5n_{5c})$. By Statement (1) in Lemma 3.2, $n - |T_0| - |T_1| = m - n_{pc} - 2n_{gp} - 2n_{7c+}$. So, the lemma holds. $\square$

Corollary 3.5 $|E(Opt) \cap E(A)| \geq |E(Opt)| - 2m + 2n_{pc} + 4(n_{gp} + n_{7c+}) + 10n_{5c} - |E_{opt}^{A,5c}| - |E_{opt}^{A,pc}|$.

PROOF. Obviously, $|E(Opt) \cap E(A)| = |E_{opt}^A| - (|E_{opt}^{A,5c}| + |E_{opt}^{A,pc}|)$. So, the corollary follows from Lemma 3.4 immediately. $\square$

The following key lemma shows that each edge of M is included in the output with a significantly high probability.

Lemma 3.6 For each edge $e \in M$, $\Pr[e \in E(H_{13})] \geq \frac{8}{75}$.

PROOF. Let us first recall that each connected component of H_{12} is an isolated vertex, a path, or a cycle. 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 $\Pr[d_{H_{10}}(u) = 1] = \frac{2}{5}$, $\Pr[d_{H_{10}}(v) = 1] = \frac{2}{5}$, $d_M(u) \leq 1$, and $d_M(v) \leq 1$, we have $\Pr[e \in M'] = \frac{4}{25}$. Thus, it remains to show that $\Pr[e \in E(H_{13}) \mid e \in M'] \geq \frac{2}{3}$. Assume that $e \in M'$. If no odd cycle of H_{12} contains e , then we are done. So, further assume that some odd cycle C of H_{12} 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_{10} . 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 10 by deleting one edge from each 5-cycle of H_5 . 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 claim, we have $\Pr[e \in E(H_{13}) \mid e \in M'] \geq \frac{2}{3}$ 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, $\Pr[d_{H_{10}}(u) = 1] = \frac{2}{5}$ and $d_M(u) \leq 1$. Moreover, $v \in T_0$ or $v \in T_1$. In the former case, $d_{H_{10}}(v) = 0$ and $d_M(v) \leq 2$. In the latter case, $d_{H_{10}}(v) = 1$ and $d_M(v) \leq 1$. So, in both cases, $d_{H_{10}}(v) + d_M(v) \leq 2$. Consequently, $\Pr[e \in M'] \geq \frac{2}{5} \cdot 1 = \frac{2}{5}$. Thus, it suffices to show that $\Pr[e \in E(H_{13}) \mid e \in M'] \geq \frac{1}{2}$. Assume that $e \in M'$. If no odd cycle of H_{12} contains e , then we are done. On the other hand, if some odd cycle C of H_{12} contains e , then the assumption $u \in V_{5c}$ guarantees that C contains at least two edges of M' and hence $\Pr[e \in E(H_{13}) \mid e \in M'] \geq \frac{1}{2}$.

Case 3: Both $u \notin V_{5c}$ and $v \notin V_{5c}$. Then, as discussed in Case 2 about v , we have $d_{H_{10}}(u) + d_M(u) \leq 2$ and $d_{H_{10}}(v) + d_M(v) \leq 2$. So, $\Pr[e \in M'] = 1$. Thus, it suffices to show that $\Pr[e \in E(H_{13}) \mid e \in M'] \geq \frac{1}{2}$. Assume that $e \in M'$. If no odd cycle of H_{12} contains e , then we are done. So, further assume that some odd cycle C of H_{12} 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_5 . However, this contradicts the construction of graph A in Step 8. Thus, the claim holds. Consequently, $\Pr[e \in E(H_{13}) \mid e \in M'] \geq \frac{1}{2}$. $\square$

By the algorithm, Statement (2) in Lemma 3.2, and Lemma 3.6, we have the following corollary immediately:

Corollary 3.7 $\mathcal{E}[|E(H_{13})|] \geq m - n_{gp} - n_{7c^+} - n_{5c} + \frac{8}{75}|M|$.

The following lemma proves a simple lower bound on $|M|$.

Lemma 3.8 $|M| \geq \frac{1}{2}|E(Opt) \cap E(A)|$.

PROOF. Let M'' be a maximum matching in graph A . Since Opt has no odd cycle, $E(Opt) \cap E(A)$ can be partitioned into two matchings of A . So, $|M''| \geq \frac{1}{2}|E(Opt) \cap E(A)|$. 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. $\square$

Theorem 3.9 $\mathcal{E}[|E(H_{13})|] \geq \frac{146}{175}|E(Opt)| + \frac{2}{21}n_{5c} + \frac{8}{75}n_{pc} - \frac{4}{75}|E_{opt}^{A,5c}| - \frac{4}{75}|E_{opt}^{A,pc}|$.

PROOF. Combining Corollary 3.7 and Lemma 3.8, we have

$$\mathcal{E}[|E(H_{13})|] \geq m - n_{gp} - n_{7c^+} - n_{5c} + \frac{4}{75}|E(Opt) \cap E(A)|.$$

So, by Corollary 3.5 and a simple calculation, we have

$$\mathcal{E}[|E(H_{13})|] \geq \frac{4}{75}|E(Opt)| + \frac{67}{75}m - \frac{7}{15}n_{5c} - \frac{59}{75}(n_{gp} + n_{7c^+}) + \frac{8}{75}n_{pc} - \frac{4}{75}|E_{opt}^{A,5c}| - \frac{4}{75}|E_{opt}^{A,pc}|.$$

Consequently, by Lemma 3.1, we have

$$\mathcal{E}[|E(H_{13})|] \geq \frac{4}{75}|E(Opt)| + \frac{82}{105}m + \frac{2}{21}n_{5c} + \frac{8}{75}n_{pc} - \frac{4}{75}|E_{opt}^{A,5c}| - \frac{4}{75}|E_{opt}^{A,pc}|.$$

Now, since m is larger than or equal to $|E(Opt)|$, the theorem follows. $\square$

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

Corollary 3.10 $\mathcal{E}[|E(H_{13})|] \geq \frac{4}{75}|E(Opt)| + \frac{58}{75}m - \frac{2}{25}n_{5c} \geq \frac{304}{375}|E(Opt)|$.

PROOF. Obviously, $|E_{opt}^{A,pc}|$ does not exceed the number of path components in H_5 . So, by Statement (4) in Lemma 3.2, $|E_{opt}^{A,pc}| \leq n_{7c+} + n_{gp} + n_{pc} \leq \frac{1}{7}(m - 5n_{5c}) + n_{pc}$. 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_5 . Consequently, $|E_{opt}^{A,5c}| \leq 4n_{5c}$. Also recall that $m \geq |E(Opt)|$. Now, by the last inequality in the proof of Theorem 3.9, the first inequality in the corollary follows. The second inequality follows from the first because $n_{5c} \leq \frac{1}{5}m$ and $m \geq |E(Opt)|$. $\square$

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 $E_{opt}^A - (E_{opt}^{A,5c} \cup E_{opt}^{A,pc})$.

Lemma 3.11 *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 set of edges $\{u, v\} \in E_{opt}^{A,5c}$ with $\{u, v\} \subseteq V(C)$. Note that $|F| \leq 4$ because Opt cannot contain a 5-cycle. We claim that there are at least $\frac{5}{4}|F|$ vertices $v \in V(C)$ with $d_K(v) < 2$. The claim implies the lemma immediately because summing up $\frac{5}{4}|F|$ over all 5-cycles C of H_1 yields the bound $\frac{5}{4}|E_{opt}^{A,5c}|$. To prove the claim, we distinguish two cases as follows:

Case 1: The edges of F do not form a cycle. Then, $|W| \geq |F| + 1$ where W is the set of the endpoints of the edges in F . Thus, $|W| \geq \frac{5}{4}|F|$ because $|F| \leq 4$. Obviously, for each $v \in W$, $d_K(v) < 2$. Hence, the claim holds in this case.

Case 2: The edges of F form a cycle C' . Then, C' must be a 4-cycle because Opt cannot contain an odd cycle. Let u be the (unique) vertex of C that is not in C' . Since C' is a 4-cycle, it is a Hamiltonian cycle of $G[V(C) - \{u\}]$. So, G has no edge $\{u, v\}$ with $v \in (T_0 \cup T_1) - V(C)$ or else (u, v) would have been a killer for the superb 5-cycle C in H_5 . Similarly, G has no edge $\{u, v\}$ with $v \in V_{5c} - V(C)$ or else (u, v) would have been a killer for the good pair (C, C') in H_3 , where C' is the 5-cycle in H_3 containing v . Moreover, there is no edge $\{u, v\} \in E(Opt)$ with $v \in V(C)$ because C' is a Hamiltonian cycle of $G[V(C) - \{u\}]$. Thus, $d_K(u) = 0$. Of course, $d_K(v) = 0$ for each $v \in V(C) - \{u\}$, because C' is a Hamiltonian cycle of $G[V(C) - \{u\}]$. Therefore, $d_K(v) = 0$ for all vertices $v \in V(C)$. Hence, the claim holds in this case, too. $\square$

Besides Lemma 3.8, we have another (less obvious) lower bound on $|M|$.

Lemma 3.12 $|M| \geq |E_{opt}^A| - 5n_{5c} - 2(n_{gp} + n_{7c+}) - 2n_{pc} + \frac{1}{4}|E_{opt}^{A,5c}| + |E_{opt}^{A,pc}|$.

PROOF. Let h be the number of vertices $v \in V_{5c}$ with $d_K(v) = 2$. By Lemma 3.11, $h \leq 5n_{5c} - \frac{5}{4}|E_{opt}^{A,5c}|$.

Let ℓ be the number of vertices $v \in T_1$ with $d_K(v) = 2$. We claim that $\ell \leq 2(n_{gp} + n_{7c+} + n_{pc}) - 2|E_{opt}^{A,pc}|$. To see this, first observe that $|T_1| = 2(n_{gp} + n_{7c+} + n_{pc})$ by Statement (4) in Lemma 3.2. Moreover, if $\{u, v\} \in E_{opt}^{A,pc}$, then both $d_K(u) \leq 1$ and $d_K(v) \leq 1$. Now, since no two edges of $E_{opt}^{A,pc}$ 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

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

Thus, by the aforementioned bounds on h and ℓ , the lemma holds. $\square$

Theorem 3.13 $\mathcal{E}[|E(H_{13})|] \geq \frac{82}{105}|E(Opt)| + \frac{2}{21}n_{5c} + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,pc}|.$

PROOF. Combining Corollary 3.7 and Lemma 3.12, we have

$$\mathcal{E}[|E(H_{13})|] \geq m + \frac{8}{75}|E_{opt}^A| - \frac{23}{15}n_{5c} - \frac{91}{75}(n_{gp} + n_{7c+}) - \frac{16}{75}n_{pc} + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,pc}|.$$

So, by Lemma 3.4 and a simple calculation, we have

$$\mathcal{E}[|E(H_{13})|] \geq \frac{59}{75}m + \frac{8}{75}|E(Opt)| - \frac{59}{75}(n_{gp} + n_{7c+}) - \frac{7}{15}n_{5c} + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,pc}|.$$

Consequently, by Lemma 3.1, we have

$$\mathcal{E}[|E(H_{13})|] \geq \frac{118}{175}m + \frac{8}{75}|E(Opt)| + \frac{2}{21}n_{5c} + \frac{2}{75}|E_{opt}^{A,5c}| + \frac{8}{75}|E_{opt}^{A,pc}|.$$

Now, since $m \geq |E(Opt)|$, the theorem follows. $\square$

Corollary 3.14 $\mathcal{E}[|E(H_{13})|] \geq \frac{1258}{1575}|E(Opt)| + \frac{2}{21}n_{5c}.$

PROOF. By Theorem 3.9, we have

$$\frac{1}{3}\mathcal{E}[|E(H_{13})|] \geq \frac{146}{525}|E(Opt)| + \frac{2}{63}n_{5c} - \frac{4}{225}|E_{opt}^{A,5c}| - \frac{4}{225}|E_{opt}^{A,pc}|.$$

On the other hand, by Theorem 3.13, we have

$$\frac{2}{3}\mathcal{E}[|E(H_{13})|] \geq \frac{164}{315}|E(Opt)| + \frac{4}{63}n_{5c} + \frac{4}{225}|E_{opt}^{A,5c}| + \frac{16}{225}|E_{opt}^{A,pc}|.$$

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

$$\mathcal{E}[|E(H_{13})|] \geq \frac{1258}{1575}|E(Opt)| + \frac{2}{21}n_{5c}.$$

$\square$

Theorem 3.15 $\mathcal{E}[|E(H_{13})|] \geq \frac{468}{575}|E(Opt)|.$

PROOF. By Corollary 3.10, we have

$$\frac{25}{46}\mathcal{E}[|E(H_{13})|] \geq \frac{31}{69}|E(Opt)| - \frac{1}{23}n_{5c}.$$

By Corollary 3.14 and the fact that $m \geq |E(Opt)|$, we have

$$\frac{21}{46}\mathcal{E}[|E(H_{13})|] \geq \frac{629}{1725}|E(Opt)| + \frac{1}{23}n_{5c}.$$

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

$$\mathcal{E}[|E(H_{13})|] \geq \frac{468}{575}|E(Opt)|.$$

$\square$

3.3 Derandomization

Our algorithm makes random choices only in Steps 10 and 13. To derandomize Step 13, 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 13. So, it should be clear that the above modification of Step 13 does not affect the approximation ratio achieved by the algorithm.

In Step 10, we make a random choice for each 5-cycle. In our above analysis of the algorithm, only the proof of Lemma 3.6 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 10 are only pairwise independent. So, we can derandomize it via conventional approaches. Therefore, we have the following theorem:

Theorem 3.16 *There is an $O(n^2m^2)$ -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 derandomized algorithm as follows. Step 1 can be done in $O(n^2m^2)$ 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 $b(v) \leq 2$ for each vertex v , Step 6 can be done in $O(\sqrt{nm})$ 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 $O(n^3)$ 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 $H(s_1, s_2)$. $\square$

3.4 Limitation of Our Approach

In this subsection, we construct an example graph to show that our analysis of the algorithm is almost tight, and also construct an example graph to show that our approach can not be revised to obtain a significantly better ratio.

Consider the graph G_1 in Figure 1, where the bold edges show a maximum triangle-free path-cycle cover of G_1 (that could be found in Step 1 in our algorithm). Clearly, G_1 has an edge-2-colorable subgraph which is made up of eleven 4-cycles (including all the fine lines) and hence has 44 edges in total. However, if our algorithm deletes the edge $\{v_1, v_7\}$ from the 7-cycle C_1 and deletes the edge $\{u_1, u_2\}$ from the 7-cycle C_2 in Step 4, then the edge-2-colorable subgraph output by our algorithm contains only the remaining 36 bold edges. So, the ratio is $\frac{9}{11} \approx 0.818$.

The above example shows that in order to achieve a much better ratio, we cannot afford to delete an arbitrary edge from each odd 7^+ -cycle. However, no matter how we deal with odd 7^+ -cycles, we

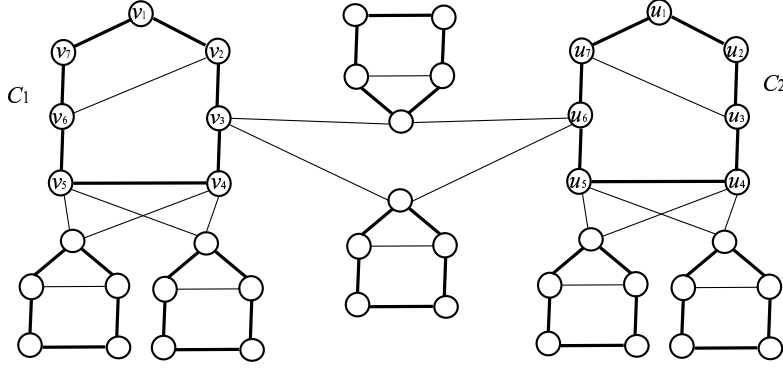


Figure 1: An example graph G_1 .

cannot get an approximation algorithm that achieves a ratio better than $\frac{5}{6} \approx 0.833$. To see this, consider the graph G_2 in Figure 2, where the bold edges show a maximum triangle-free path-cycle cover H of G_2 (that could be found in Step 1 in our algorithm). Clearly, G_2 has an edge-2-colorable subgraph which is made up of four 4-cycles and one 8-cycle (including all the fine lines) and hence has 24 edges in total. However, if we do not modify even cycles in H when transforming it to an edge-2-colorable graph of G_2 , then what we can do is just to delete one edge from each 5-cycle in H , yielding an edge-2-colorable subgraph of G_2 with only 20 edges. So, the ratio achieved is only $\frac{5}{6}$.

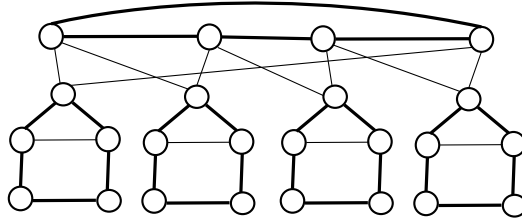


Figure 2: An example graph G_2 .

By the example graph G_2 , in order to achieve a much better approximation ratio, we need to figure out how to modify even cycles in the maximum triangle-free path-cycle cover computed in Step 1. However, this seems to be a very difficult task.

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 $[1, \Delta]$ -factor k -packing of G is a collection of k disjoint edge covers of G . The *size* of a $[1, \Delta]$ -factor k -packing $\{F_1, \dots, F_k\}$ of G is $|F_1| + \dots + |F_k|$. 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 $[1, \Delta]$ -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 achieving 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 3.16, we have the following immediately:

Theorem 4.2 *There is an $O(n^2m^2)$ -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 of $\frac{42\Delta-30}{35\Delta-21}$, where Δ is the maximum degree of a vertex in the input graph [7].

Acknowledgement

The authors would like to thank the referees for very helpful comments. Especially, one of the referees pointed out an error in the previous version.

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 Combinatorial Optimization (IPCO)*, Lecture Notes in Computer Science, **2462** (2002) 108-121.
- [2] H.N. Gabow. An Efficient Reduction Technique for Degree-Constrained Subgraph and Bidi-rected 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.