Approximating Maximum Edge 2-Coloring in Simple Graphs via Local Improvement

Zhi-Zhong Chen[1] and Ruka Tanahashi[1]

Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan

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 $\frac{24}{29} = 0.827586\ldots$. This improves on the previous best ratio of $\frac{468}{575} = 0.813913\ldots$.

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. [2] 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. [2] 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. [2] 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. [2] 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*: Given a simple graph G, first 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, and then remove one *arbitrary* edge from each odd cycle of H.

In [1], the authors have improved the ratio to $\frac{468}{575}$. Essentially, the algorithm in [1] differs from the simple algorithm only in the handling of 5-cycles where instead of removing one arbitrary edge from each 5-cycle of H, we remove a *random* edge from each 5-cycle of H. The intuition behind the algorithm is as follows: If we delete a random edge from each 5-cycle of H, then for each edge $\{u, v\}$ in the optimal solution such that u and v belong to different 5-cycles, both u and v become of degree 1 in H (after handling the 5-cycles) with a probability of $\frac{4}{25}$ and so can be added into H without losing the edge 2-colorability of H.

In this paper, we further improve the ratio to $\frac{24}{29}$. The basic idea behind our algorithm is as follows: Instead of removing a random edge from each 5-cycle of H and removing an arbitrary edge from each other odd cycle of H, we remove one edge from each odd cycle of H with more care in the hope that after the removal, a lot of edges $\{u, v\}$ (in the optimal solution) with u and v belonging to different odd cycles of H can be added to H. More specifically, we define a number of operations that modify each odd cycle of H together with its neighborhood carefully without decreasing the number of edges in H by two or more; our algorithm just performs these operations on H until none of them is applicable. The nonapplicability of these operations guarantees that H is edge 2-colorable and its number of edges is close to optimal; the analysis is quite challenging.

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 particular, this ratio is smaller than $\frac{24}{29}$ when $\Delta \geq 6$.

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

A graph always means a simple undirected graph. A graph G has a vertex set $V(G)$ and an edge set $E(G)$. For each $v \in V(G)$, $N_G(v)$ denotes the set of all vertices adjacent to v in G and $d_G(v) = |N_G(v)|$ is the *degree* of v in G. If $d_G(v) = 0$, then v is an *isolated vertex* of G. For each $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G induced by U.

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. Each path has two endpoints and zero or more inner vertices. An edge $\{u, v\}$ of a path P is an *inner edge* of P if both u and v are inner vertices of P. The *length* of a cycle or path C is the number of edges in C. A cycle of odd (respectively, even) length is 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). 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 $v \in V(H)$. A *cycle cover* of G is a path-cycle cover of G in which each connected component is a cycle. A path-cycle cover C of G is *triangle-free* if C does not contain a 3-cycle.

G is *edge-2-colorable* if each connected component of G is an isolated vertex, a path, or an even cycle. Note that Max Simple Edge 2-Coloring is the problem of finding a maximum edge-2-colorable subgraph in a given graph.

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:

- $n_0(K)$ is the number of isolated vertices in K.
- $p(K)$ is the number of path components in K.

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 odd cycle in H. In order to improve this ratio, we have to treat 5-cycles (and other short odd cycles) in H more carefully. In more details, when removing edges from odd cycles in H, we also want to add some edges of $E(G) - E(H)$ to H. For this purpose, we will define a number of operations on H that always decrease the number of cycles in H but may decrease the number of edges in H or not. To tighten the analysis of the approximation ratio achieved by our algorithm, we set up a charging scheme that charges the net loss of edges from H (due to the operations) to some edges still remaining in H. Whenever we do this, we will always maintain the following invariants:

I1. Every edge of H is charged a real number smaller than or equal to $\frac{1}{9}$.
I2. The total charge on the edges of H equals the total number of operations performed on H that decrease the number of edges in H.
I3. No cycle component of H contains a charged edge.
I4. If a path component P of H contains a charged edge, then the length of P is at least 6.

Initially, every edge of H is charged nothing. However, as we modify H by performing operations (to be defined below), some edges of H will be charged.

We first define those operations on H that decrease the number of odd cycles in H but do not decrease the number of edges in H. In order to do this, the following three concepts are necessary:

A quadruple (x, y, P, u, v) is a *5-opener* for an odd cycle C of H if the following hold:

- $d_H(x) \leq 1$ and $y \in V(C)$.
- P is a path component of H, both u and v are inner vertices of P, and x is not a vertex of P.
- Both $\{u, x\}$ and $\{v, y\}$ are contained in $E(G) - E(H)$.

A sextuple (x, y, Q, P, u, v) is a *6-opener* for an odd cycle C of H if the following hold:

– $x \in V(C)$ and $y \in V(C)$. Moreover, if $x = y$, then Q is a cycle cover of $G[V(C) - \{x\}]$ in which each connected component is an even cycle; otherwise, Q is a path-cycle cover of $G[V(C)]$ in which one connected component is a path from x to y and each other connected component is an even cycle.
– P is a path component of H and both u and v are inner vertices of P.
– Both $\{u, x\}$ and $\{v, y\}$ are contained in $E(G) - E(H)$.

An operation (to be performed) on H is *robust* if the following holds:

– If G has no edge $\{u, v\}$ before the operation such that u is an isolated vertex in H and either v is an isolated vertex in H or v appears in a cycle component of H, then neither does it after the operation.

Based on the above concepts, we are now ready to define six robust operations on H that decrease the number of odd cycles in H but do not decrease the number of edges in H.

Type 1: Suppose that $\{u, v\}$ is an edge in $E(G) - E(H)$ such that $d_H(u) \leq 1$ and v is a vertex of some cycle C of H. Then, a *Type-1 operation* on H using $\{u, v\}$ modifies H by deleting one (arbitrary) edge of C incident to v and adding edge $\{u, v\}$. Obviously, this operation is robust and does not change $|E(H)|$. (*Comment:* If $d_H(u) = 0$ before a Type-1 operation, then $n_0(H)$ decreases by 1 and $p(H)$ increases by 1 after the operation. Similarly, if $d_H(u) = 1$ before a Type-1 operation, then neither $n_0(H)$ nor $p(H)$ changes after the operation.)

Type 2: Suppose that some odd cycle C of H has a 5-opener (x, y, P, u, v) with $\{u, v\} \in E(H)$ (see Figure 1). Then, a *Type-2 operation* on H using (x, y, P, u, v) modifies H by deleting edge $\{u, v\}$, deleting one (arbitrary) edge of C incident to y, and adding edges $\{u, x\}$ and $\{v, y\}$. Obviously, this operation is robust and does not change the number of edges in H. However, edge $\{u, v\}$ may have been charged before this operation. If that is the case, we move its charge to $\{u, x\}$. Moreover, if the path component Q of H containing edge $\{u, x\}$ after this operation is of length at most 5, then we move the charges on the edges of Q to edge $\{v, y\}$ and the edges of C still remaining in H. (*Comment:* A Type-2 operation on H maintains Invariants I1 through I4. Moreover, if $d_H(x) = 0$ before a Type-2 operation, then $n_0(H)$ decreases by 1 and $p(H)$ increases by 1 after the operation. Similarly, if $d_H(x) = 1$ before a Type-2 operation, then neither $n_0(H)$ nor $p(H)$ changes after the operation.)

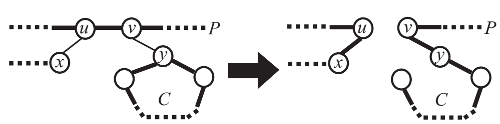

Fig. 1. A Type-2 operation, where bold edges are in H.

Type 3: Suppose that some odd cycle C of H has a 5-opener (x, y, P, u, v) such that $E(G) - E(H)$ contains the edge $\{w, s\}$, where w is the neighbor of v in the subpath of P between u and v and s is the endpoint of P with $dist_P(s, u) < dist_P(s, v)$ (see Figure 2 in the appendix). Then, a *Type-3 operation* on H using (x, y, P, u, v) modifies H by deleting edge $\{v, w\}$, deleting one (arbitrary) edge

e_u of P incident to u, deleting one (arbitrary) edge of C incident to y, and adding edges $\{u, x\}$, $\{v, y\}$, and $\{s, w\}$. Obviously, this operation is robust and does not change the number of edges in H. Note that $\{v, w\}$ or e_u may have been charged before this operation. If that is the case, we move their charges to edges $\{u, x\}$ and $\{s, w\}$, respectively. Moreover, if the path component Q of H containing edge $\{u, x\}$ after this operation is of length at most 5, then we move the charges on the edges of Q to edge $\{v, y\}$ and the edges of C still remaining in H. (*Comment:* A Type-3 operation on H maintains Invariants I1 through I4. Moreover, if $d_H(x) = 0$ before a Type-3 operation, then $n_0(H)$ decreases by 1 and $p(H)$ increases by 1 after the operation. Similarly, if $d_H(x) = 1$ before a Type-3 operation, then neither $n_0(H)$ nor $p(H)$ changes after the operation.)

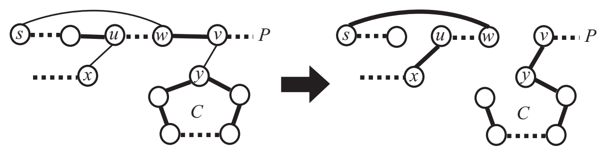

Fig. 2. A Type-3 operation, where bold edges are in H.

Type 4: Suppose that there is a quadruple (x, P, u, v) satisfying the following conditions (see Figure 3 in the appendix):

- x is a vertex of a cycle component C of H.
- P is a path component of H and $\{u, v\}$ is an inner edge of P.
- $E(G) - E(H)$ contains both $\{u, x\}$ and $\{s, v\}$, where s is the endpoint of P with $dist_P(s, u) < dist_P(s, v)$.

Then, a *Type-4 operation* on H using (x, P, u, v) modifies H by deleting edge $\{u, v\}$, deleting one (arbitrary) edge of C incident to x, and adding edges $\{u, x\}$ and $\{s, v\}$. Obviously, this operation is robust and does not change the number of edges in H. However, $\{u, v\}$ may have been charged before this operation. If that is the case, we move its charge to $\{u, x\}$. (*Comment:* A Type-4 operation on H maintains Invariants I1 through I4, and changes neither $n_0(H)$ nor $p(H)$.)

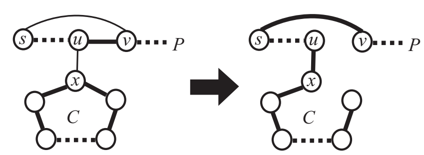

Fig. 3. A Type-4 operation, where bold edges are in H.

Type 5: Suppose that there is a quintuple (x, P, u, v, w) satisfying the following conditions (see Figure 4 in the appendix):

- x is a vertex of a cycle component C of H.
- P is a path component of H, u is an inner vertex of P, $\{v, w\}$ is an inner edge of P, and $dist_P(u, v) < dist_P(u, w)$.
- $E(G) - E(H)$ contains $\{u, x\}$, $\{s, w\}$, and $\{t, v\}$, where s is the endpoint of P with $dist_P(s, u) < dist_P(s, v)$ and t is the other endpoint of P.

Then, a *Type-5 operation* on H using (x, P, u, v, w) modifies H by deleting edge $\{v, w\}$, deleting one (arbitrary) edge e_u of P incident to u, deleting one (arbitrary) edge of C incident to x, and adding $\{s, w\}$, $\{t, v\}$, and $\{u, x\}$. Obviously, this operation is robust and does not change the number of edges in H. However, $\{v, w\}$ and e_u may have been charged before this operation. If that is the case, we move their charges to $\{u, x\}$ and $\{s, w\}$, respectively. (*Comment:* A Type-5 operation on H maintains Invariants I1 through I4, and changes neither $n_0(H)$ nor $p(H)$.)

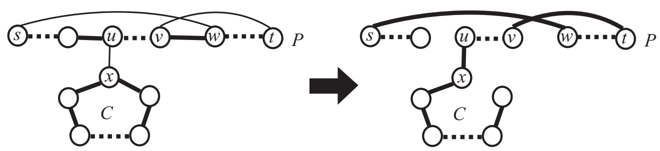

Fig. 4. A Type-5 operation, where bold edges are in H.

Type 6: Suppose that some odd cycle C of H with length at most 9 has a 6-opener (x, y, Q, P, u, v) such that $\{u, v\} \in E(H)$ (see Figure 5 in the appendix). Then, a *Type-6 operation* on H using (x, y, Q, P, u, v) modifies H by deleting edge $\{u, v\}$, deleting all edges of C, adding edges $\{u, x\}$ and $\{v, y\}$, and adding all edges of Q. Obviously, this operation does not change the number of edges in H, and is robust because (1) it does not create a new isolated vertex in H and (2) if it creates one or more new cycles in H then $V(C') \subseteq V(C)$ for each new cycle C'. However, $\{u, v\}$ may have been charged before this operation. If that is the case, we move its charge to $\{u, x\}$. (*Comment:* A Type-6 operation on H maintains Invariants I1 through I4, and changes neither $n_0(H)$ nor $p(H)$.)

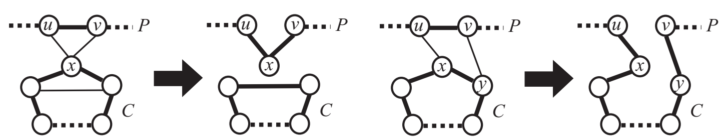

Fig. 5. A Type-6 operation, where bold edges are in H.

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

2. Repeat performing a Type-i operation on H with $1 \leq i \leq 6$, until none is applicable.

Obviously, H remains a triangle-free path-cycle cover of G. Moreover, the following fact holds:

Lemma 1. *After Step 2, G has no edge $\{u, v\}$ such that u is an isolated vertex in H and either v is an isolated vertex in H or v appears in a cycle component of H.*

Unfortunately, H may still have odd cycles after Step 2. So, we need to perform new types of operations on H that always decrease the number of odd cycles in H but may also decrease the number of edges in H. Before defining the new operations on H, we define two concepts as follows. Two cycles C_1 and C_2 of H are *pairable* if at least one of them is odd and their total length is at least 10. A quintuple (x, y, P, u, v) is an *opener* for two pairable cycles C_1 and C_2 of H if the following hold:

- x is a vertex of C_1 and y is a vertex of C_2.
- P is either a path component of H or a 4-cycle of H other than C_1 and C_2.
- u and v are distinct vertices of P with $d_H(u) = d_H(v) = 2$.
- Both $\{u, x\}$ and $\{v, y\}$ are in $E(G) - E(H)$.

Now, we are ready to define the new types of robust operations on H as follows:

Type 7: Suppose that C is an odd cycle of H with length at least 11. Then, a *Type-7 operation* on H using C modifies H by deleting one (arbitrary) edge from C. Clearly, the net loss in the number of edges in H is 1. We charge this loss evenly to the edges of C still remaining in H. In more details, if C was a k-cycle before the operation, then a charge of $\frac{1}{k-1}$ is charged to each edge of C still remaining in H after the operation. Since $k \geq 11$, the charge assigned to one edge here is at most $\frac{1}{10}$. Obviously, this operation is robust. (*Comment:* A Type-7 operation on H maintains Invariants I1 through I4, does not change $n_0(H)$, and increases $p(H)$ by 1.)

Type 8: Suppose that C_1 and C_2 are two pairable cycles of H such that there is an edge $\{u, v\} \in E(G)$ with $u \in V(C_1)$ and $v \in V(C_2)$. Then, a *Type-8 operation* on H using $\{u, v\}$ modifies H by deleting one (arbitrary) edge of C_1 incident to u, deleting one (arbitrary) edge of C_2 incident to v, and adding edge $\{u, v\}$. Note that this operation decreases the number of edges in H by 1. So, the net loss in the number of edges in H is 1. We charge this loss evenly to edge $\{u, v\}$ and the edges of C_1 and C_2 still remaining in H. In more details, if C_1 was a k-cycle and C_2 was an ℓ-cycle in H before the operation, then a charge of $\frac{1}{k+\ell-1}$ is assigned to $\{u, v\}$ and each edge of C_1 and C_2 still remaining in H after the operation. Since $k \geq 5$ and $\ell \geq 5$, the charge assigned to one edge here is at most $\frac{1}{9}$. Obviously, this operation is robust. (*Comment:* A Type-8 operation on H maintains Invariants I1 through I4, does not change $n_0(H)$, and increases $p(H)$ by 1.)

Type 9: Suppose that two odd cycles C_1 and C_2 of H have an opener (x, y, P, u, v) with $\{u, v\} \in E(H)$ (see Figure 6 in the appendix). Then, a *Type-9 operation* on H using (x, y, P, u, v) modifies H by deleting edge $\{u, v\}$, deleting one (arbitrary) edge of C_1 incident to x, deleting one (arbitrary) edge of C_2 incident to y, and adding edges $\{u, x\}$ and $\{v, y\}$. Note that edge $\{u, v\}$ may have been charged before this operation. If that is the case, we move its charge to edge $\{u, x\}$. Moreover, the operation decreases the number of edges in H by 1. So, the net loss in the number of edges in H is 1. We charge this loss evenly to edge $\{v, y\}$ and the edges of C_1 and C_2 still remaining in H. Obviously, the charge assigned to one edge here is at most $\frac{1}{9}$. It is also clear that this operation is robust. (*Comment:* A Type-9 operation on H maintains Invariants I1 through I4, does not change $n_0(H)$, and increases $p(H)$ by 1.)

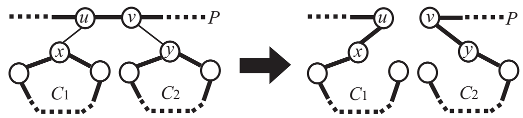

Fig. 6. A Type-9 operation, where bold edges are in H.

Type 10: Suppose that two odd cycles C_1 and C_2 of H have an opener (x, y, P, u, v) such that $E(G) - E(H)$ contains the edge $\{w, s\}$, where w is the neighbor of v in the subpath of P between u and v and s is the endpoint of P with $dist_P(s, u) < dist_P(s, v)$ (see Figure 7 in the appendix). Then, a *Type-10 operation* on H using (x, y, P, u, v) modifies H by deleting edge $\{v, w\}$, deleting one (arbitrary) edge e_u of P incident to u, deleting one (arbitrary) edge of C_1 incident to x, deleting one (arbitrary) edge of C_2 incident to y, and adding edges $\{u, x\}$, $\{v, y\}$, and $\{s, w\}$. Note that $\{v, w\}$ or e_u may have been charged before this operation. If that is the case, we move their charges to edges $\{u, x\}$ and $\{v, y\}$, respectively. Moreover, the operation decreases the number of edges in H by 1. So, the net loss in the number of edges in H is 1. We charge this loss evenly to edge $\{s, w\}$ and the edges of C_1 and C_2 still remaining in H. Obviously, the charge assigned to one edge here is at most $\frac{1}{9}$. It is also clear that this operation is robust. (*Comment:* A Type-10 operation on H maintains Invariants I1 through I4, does not change $n_0(H)$, and increases $p(H)$ by 1.)

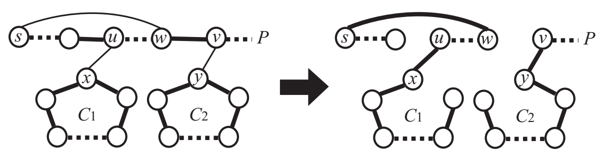

Fig. 7. A Type-10 operation, where bold edges are in H.

After Step 2, no matter how many times we perform Type-i operations on H with $1 \le i \le 10$, G cannot have an edge $\{u, v\}$ such that u is an isolated vertex in H and either v is an isolated vertex in H or v appears in a cycle component of H. This follows from Lemma 1 and the fact that every Type-i operation on H with $1 \le i \le 10$ is robust. However, after performing a Type-i operation on H with $7 \le i \le 10$, the following new type of robust operations on H may be applicable:

Type 11: Suppose that $\{u, v\}$ is an edge in $E(G) - E(H)$ such that $d_H(u) = 1$, $d_H(v) \le 1$, and no connected component of H contains both u and v. Then, a *Type-11 operation* on H using $\{u, v\}$ modifies H by adding edge $\{u, v\}$. Obviously, this operation is robust and increases the number of edges in H by 1. (*Comment:* If $d_H(v) = 0$ before a Type-11 operation, then $p(H)$ does not change and $n_0(H)$ decreases by 1 after the operation. Similarly, if $d_H(v) = 1$ before a Type-11 operation, then $n_0(H)$ does not change and $p(H)$ decreases by 1 after the operation.)

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

3. Repeat using a Type-i operation to modify H with $1 \le i \le 11$, until none is applicable.
4. For each odd cycle C of H, remove one (arbitrary) edge from C. (*Comment:* Each odd cycle modified in this step is a 5-, 7-, or 9-cycle.)
5. Output H.

4 Performance Analysis

For $1 \le i \le 4$, let H_i be the triangle-free path-cycle cover H of G immediately after Step i of our algorithm. In order to analyze the approximation ratio achieved by our algorithm, we need to define several notations as follows:

- Let n, m, n_{is}, and n_{pc} be the numbers of vertices, edges, isolated vertices, and path components in H_2, respectively. (*Comment:* $m \ge |E(Opt)|$.)
- Let m_- be the number of Type-i operations with $7 \le i \le 10$ performed in Step 3.
- Let $m_{+,-1}$ be the number of Type-11 operations performed Step 3 that decrease the number of isolated vertices in H by 1.
- Let $m_{+,0}$ be the number of Type-11 operations performed Step 3 that do not change the number of isolated vertices in H.
- Let $n_{0,-1}$ be the number of Type-i operations with $1 \le i \le 3$ performed in Step 3 that decrease the number of isolated vertices in H by 1.
- For each $i \in \{5, 7, 9\}$, let c_i be the number of i-cycles in H_3.
- Let m_c and m_{uc} be the numbers of charged edges and uncharged edges in H_3, respectively.)

Lemma 2. *The following statements hold:*

1. $m_- \le \frac{1}{10}(m + m_{+,0} + m_{+,-1} - m_{uc})$.
2. $|E(H_4)| = m - m_- + m_{+,0} + m_{+,-1} - c_5 - c_7 - c_9$.
3. $|E(H_4)| \ge \frac{9}{10}(m + m_{+,0} + m_{+,-1}) - (\frac{1}{2}c_5 + \frac{3}{10}c_7 + \frac{1}{10}c_9)$.

Proof. By the algorithm, $|E(H_3)| = m - m_- + m_{+,0} + m_{+,-1}$. On the other hand, $|E(H_3)| = m_c + m_{uc}$ by definition. So, $m_c = m - m_- + m_{+,0} + m_{+,-1} - m_{uc}$. We also have $m_- \le \frac{1}{9}m_c$ by Invariant I2. Thus, $m_- \le \frac{1}{10}(m + m_{+,0} + m_{+,-1} - m_{uc})$.

By Step 3, $|E(H_4)| = |E(H_3)| - c_5 - c_7 - c_9$. So, by the first equality in the last paragraph, $|E(H_4)| = m - m_- + m_{+,0} + m_{+,-1} - c_5 - c_7 - c_9$.

By Statements 1 and 2, $|E(H_4)| \ge \frac{9}{10}(m + m_{+,0} + m_{+,-1}) + \frac{1}{10}m_{uc} - c_5 - c_7 - c_9$. We also have $m_{uc} \ge 5c_5 + 7c_7 + 9c_9$, because each edge in a cycle component of H_3 is uncharged according to Invariant I3. Combining these two inequalities, we have Statement 3. □

Lemma 3. *The following statements hold:*

1. $n - n_0(H_3) - 2p(H_3) = m - n_{pc} - 2m_- + 2m_{+,0} + m_{+,-1} - n_{0,-1}$.
2. $p(H_3) = n_{pc} + m_- - m_{+,0} + n_{0,-1}$.

Proof. Immediately before Step 3, $n - n_0(H) - 2p(H) = m - n_{pc}$ because $p(H) = n_{pc}$ and the number of vertices on a path is 1 plus the number of edges on the path. Now, to prove the lemma, it suffices to see how the values of $n - n_0(H) - 2p(H)$ and $p(H)$ change when performing an operation in Step 3. The comment on the definition of each type of operations helps. □

In order to analyze the algorithm, we need more definitions:

- For $i \in \{0, 1\}$, let T_i be the set of all vertices v in H_3 with $d_{H_3}(v) = i$.
- Let T_2 be the set of all vertices v in H_3 such that v appears in an odd cycle of H_3.
- Let $T = T_0 \cup T_1 \cup T_2$.
- For $i \in \{0, 1, 2\}$, let $\overline{T_i}$ be the set of vertices $u \in V(G) - T$ such that the number of edges $\{u, v\} \in E(Opt)$ with $v \in T$ is exactly i. (*Comment:* $V(G) - T = \overline{T_0} \cup \overline{T_1} \cup \overline{T_2}$.)
- Let E^T_{opt} be the set of all edges $\{u, v\}$ in Opt such that both u and v are vertices of T.
- Let $\mathcal{C}_{-2}$ be the set of all odd cycles in H_3 such that Opt contains at most $|V(C)| - 2$ edges $\{u, v\}$ with $\{u, v\} \subseteq V(C)$.

Lemma 4. $|E^T_{opt}| \le p(H_3) + 4c_5 + 6c_7 + 8c_9 - |\mathcal{C}_{-2}| \le n_{pc} + m_- - m_{+,0} + n_{0,-1} + 4c_5 + 6c_7 + 8c_9 - |\mathcal{C}_{-2}|$.

Proof. First, we claim that each vertex $u \in T_0$ is an isolated vertex in $G[T]$. To see this, consider an arbitrary $u \in T_0$. Because of Lemma 1 and the fact that all Type-i operations with $1 \le i \le 11$ are robust, there is no vertex $v \in T_0 \cup T_2$ with $\{u, v\} \in E(G)$. Moreover, since no Type-11 operation can be applied to H_3, there is no vertex $v \in T_1$ with $\{u, v\} \in E(G)$. So, the claim holds.

Next, we claim that there is no edge $\{u, v\} \in E(G)$ with $u \in T_1$ and $v \in T_2$. This follows from the fact that no Type-1 operation can be applied to H_3.

By the above two claims, each edge in E^T_{opt} is either in $G[T_1]$ or in $G[T_2]$. Since no Type-11 operation can be applied to H_3, there is no edge $\{u, v\} \in E(G)$ with $\{u, v\} \subseteq T_1$ such that u and v belong to different connected components of H_3. So, there are at most $p(H_3)$ edges in $G[T_1]$. Consequently, to show the first inequality in the lemma, it remains to show that E^T_{opt} contains at most $4c_5 + 6c_7 + 8c_9 - |\mathcal{C}_{-2}|$ edges $\{u, v\}$ with $\{u, v\} \subseteq T_2$.

Suppose that $\{u, v\}$ is an edge in E^T_{opt} with $\{u, v\} \subseteq T_2$. Since no Type-8 operation can be applied to H_3, x and y belong to the same cycle component of $H_3[T_2]$. On the other hand, since each cycle C in $H_3[T_2]$ is an odd cycle, E^T_{opt} can contain at most $|E(C)| - 1$ edges $\{u, v\}$ with $\{u, v\} \subseteq V(C)$. In particular, for each cycle $C \in \mathcal{C}_{-2}$, E^T_{opt} can contain at most $|E(C)| - 2$ edges $\{u, v\}$ with $\{u, v\} \subseteq V(C)$. Hence, E^T_{opt} contains at most $4c_5 + 6c_7 + 8c_9 - |\mathcal{C}_{-2}|$ edges $\{u, v\}$ with $\{u, v\} \subseteq T_2$. This completes the proof of the first inequality in the lemma. The second follows from the first and Statement 2 in Lemma 3. □

Lemma 5. $|E^T_{opt}| \ge |E(Opt)| - 2m + 2n_{pc} + 4m_- - 4m_{+,0} - 2m_{+,-1} + 2n_{0,-1} + 10c_5 + 14c_7 + 18c_9 + |\overline{T_0}| + \frac{1}{2}|\overline{T_1}|$.

Proof. The idea behind the proof is to obtain an upper bound on $|E(Opt)-E^T_{opt}|$. A trivial upper bound is $2(n-|T|)$, because each edge in $E(Opt)-E^T_{opt}$ must be incident to a vertex in $V(G)-T$ and each vertex in $V(G)-T$ can be adjacent to at most two edges in Opt. This bound is not good enough because each edge $\{u,v\}\in E(Opt)$ with $\{u,v\}\subseteq V(G)-T$ is counted twice.

To get a better bound, we set up a savings account for each vertex in $V(G)-T$. Initially, we deposit two credits to the account of each vertex. The total credits amount to $2(n-|T|)$ (namely, the trivial upper bound). Next, for each edge $\{u,v\}\in E(Opt)$ with $\{u,v\}\subseteq V(G)-T$, we pay a half credit from the account of u and another half credit from the account of v. After this, for each edge $\{u,v\}\in E(Opt)$ with $u\in V(G)-T$ and $v\in T$, we pay one credit from the account of u. Obviously, we paid a total of $|E(Opt)-E^T_{opt}|$ credits. We want to estimate the number of credits that are still left in the accounts of the vertices in $V(G)-T$. First, for each vertex $u\in\overline{T_0}$, we paid at most one credit, because u is incident to at most two edges in $E(Opt)-E^T_{opt}$ and each of them has both of its endpoints in $V(G)-T$. So, the total number of credits still left in the accounts of the vertices in $\overline{T_0}$ is at least $|\overline{T_0}|$. Second, for each vertex $u\in\overline{T_1}$, we paid at most one and a half credits, because u is incident to at most two edges in $E(Opt)-E^T_{opt}$ and one of them has an endpoint in T. Thus, the total number of credits still left in the accounts of the vertices in $\overline{T_1}$ is at least $\frac{1}{2}|\overline{T_1}|$. In summary, we have shown that $|E(Opt)-E^T_{opt}|\le 2(n-|T|)-|\overline{T_0}|-\frac{1}{2}|\overline{T_1}|$.

Obviously, $|T|=n_0(H_3)+2p(H_3)+5c_5+7c_7+9c_9$. So, by Statement 1 in Lemma 3, $|T|=n-(m-n_{pc}-2m_-+2m_{+,0}+m_{+,-1}-n_{0,-1})+5c_5+7c_7+9c_9$. In other words, $n-|T|=m-n_{pc}-2m_-+2m_{+,0}+m_{+,-1}-n_{0,-1}-5c_5-7c_7-9c_9$. Hence, by the last inequality in the last paragraph, $|E(Opt)-E^T_{opt}|\le 2m-2n_{pc}-4m_-+4m_{+,0}+2m_{+,-1}-2n_{0,-1}-10c_5-14c_7-18c_9-|\overline{T_0}|-\frac{1}{2}|\overline{T_1}|$. So, the lemma holds. □

Lemma 6. $|E(H_4)|\ge\frac{37}{45}|E(Opt)|$.

Proof. Combining Lemmas 4 and 5, we have $|E(Opt)|\le 2m-n_{pc}-3m_-+3m_{+,0}+2m_{+,-1}-n_{0,-1}-6c_5-8c_7-10c_9$. So, by Statement 2 in Lemma 2, $3|E(H_4)|-|E(Opt)|\ge m+m_{+,-1}+n_{pc}+n_{0,-1}+3c_5+5c_7+7c_9$. Thus, $3c_5+5c_7+7c_9\le 3|E(H_4)|-|E(Opt)|-m$. Hence, $\frac{1}{2}c_5+\frac{3}{10}c_7+\frac{1}{10}c_9\le\frac{1}{6}(3c_5+5c_7+7c_9)\le\frac{1}{2}|E(H_4)|-\frac{1}{6}|E(Opt)|-\frac{1}{6}m$. Therefore, by Statement 3 in Lemma 2, we have $|E(H_4)|\ge\frac{9}{10}m-(\frac{1}{2}|E(H_4)|-\frac{1}{6}|E(Opt)|-\frac{1}{6}m)$. Rearranging this inequality and using the fact that $m\ge|E(Opt)|$, we finally obtain $|E(H_4)|\ge\frac{37}{45}|E(Opt)|$. □

Note that $\frac{37}{45}=0.8222\ldots$ which is better than the previously best ratio. To show that our algorithm indeed achieves an even better ratio, our idea is to show that $|\mathcal{C}_{-2}|+|\overline{T_0}|+\frac{1}{2}|\overline{T_1}|$ is large. This holds basically because no Type-i operation with $i\in\{1,\ldots,11\}$ can be applied to H_3. The proof is very complicated and will appear in the full version of this paper. In summary, we have our main result:

Theorem 1. *There is an $O(n^2m^2)$-time approximation algorithm for* Max Simple Edge 2-Coloring *achieving a ratio of $\frac{24}{29}$, where n (respectively, m) is the number of vertices (respectively, edges) in the input graph.*

5 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,Δ]-factor k-packing $\{F_1, \ldots, F_k\}$ of G is $|F_1| + \cdots + |F_k|$. The problem of deciding whether a given graph has a [1,Δ]-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,Δ]-factor k-packing of G of minimum size or decide that G has no [1,Δ]-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]). Min-2-FP is NP-hard [7].

Lemma 7. [7] *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 $O(T(n))$.*

Theorem 2. *There is an $O(n^2m^2)$-time approximation algorithm for* Min-2-FP *achieving a ratio of $\frac{34}{29}$, where n (respectively, m) is the number of vertices (respectively, edges) in the input graph.*

Theorem 2 follows from Theorem 1 and Lemma 7 immediately. Previously, the best ratio achieved by a polynomial-time approximation algorithm for Min-2-FP was $\frac{682}{575}$ [1], 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].

References

1. Z.-Z. Chen, R. Tanahashi, and L. Wang. An Improved Approximation Algorithm for Maximum Edge 2-Coloring in Simple Graphs. To appear in *Journal of Discrete Algorithms*.
2. 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.
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.