Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision

CNRS FRE 3234

CAHIER DU LAMSADE

301

Septembre 2010

On the approximability of the Max Edge-Coloring problem

GIORGIO LUCARELLI, IOANNIS MILIS

On the approximability of the Max Edge-Coloring problem

Giorgio Lucarelli[1] and Ioannis Milis[2]

[1] LAMSADE, Université Paris-Dauphine and CNRS FRE 3234
lucarelli@lamsade.dauphine.fr
[2] Department of Informatics, Athens University of Economics and Business
milis@aueb.gr

Abstract. The max edge-coloring problem asks for a proper edge-coloring of an edge-weighted graph minimizing the sum of the weights of the heaviest edges in each color class. This problem arises in optical communication systems and has been well studied during last years. However, no algorithm of approximation ratio $2 - \delta$, for any constant $\delta > 0$, is known for general or bipartite graphs, while the complexity of the problem on trees remains an open question. In this paper we present new approximation results towards these questions. In fact, we present a PTAS for trees and an 1.74-approximation algorithm for bipartite graphs; we also adapt the last algorithm to one for general graphs of the same, asymptotically, approximation ratio.

1 Introduction

In several communication systems messages are to be transmitted directly from senders (input ports) to receivers (output ports) through direct connections established by an underlying switching network (e.g., SS/TDMA [10], IQ switch architectures [14]). Any node of such a system cannot participate in more than one transmissions at the same time, while messages between different pairs of senders and receivers can be transmitted simultaneously. A scheduler establishes successive configurations of the switching network, each one routing a non-conflicting subset of the messages from senders to receivers. Given the transmission time of each message, the transmission time of each configuration equals to the longest message transmitted. The aim is to find a sequence of configurations such that all the messages are transmitted and the total transmission time is minimized.

It is easy to see that this situation corresponds directly to the following generalized coloring problem: Given a graph $G = (V, E)$ and a positive integer weight $w(e)$, for each edge $e \in E$, we seek for a proper edge-coloring of G, $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$, where each color class (matching) $M_i \subseteq E$ is assigned the weight of the heaviest edge in this class, i.e., $w_i = \max\{w(e)|e \in M_i\}$, $1 \leq i \leq k$, and the sum of all color classes' weights, $W = \sum_{i=1}^{k} w_i$, is minimized. In fact, senders and/or receivers correspond to the vertices of the graph G, (transmission times of) messages correspond to (weights of) edges of G and configurations correspond to matchings. Although the graph G obtained is originally a weighted directed multi-graph it can be considered as an undirected one, since the directions of its edges do not play any role in the objective function.

The above coloring problem is known as the *Max Edge-Coloring* (MEC) problem; clearly, for unit edge weights it reduces to the classical edge-coloring problem. The analogous weighted generalization of the classical vertex-coloring problem has been also addressed in the literature as *Max (Vertex-)Coloring* (MVC) problem [19, 18].

Remark that the MEC problem on a general graph, G, is equivalent to the MVC problem on the line graph, $L(G)$, of G. Thus, the results for the MVC problem on a graph G apply also to the MEC problem on the graph $L(G)$ and vice versa, if both G and $L(G)$ are in the same graph class. Note, however, that this is true for general graphs and chains, but not for the most other special graph classes, including bipartite graphs and trees, since they are not closed under line graph transformation (e.g., the line graph of a bipartite graph is not anymore a bipartite one).

The MEC problem can be also viewed as a parallel batch scheduling problem with conflicts between jobs [5, 8]. According to the standard three field notation for scheduling problems, the MEC problem is equivalent to $1 \mid p-batch,\ E(G) \mid C_{max}$: Jobs correspond to the edges $E(G)$ of a weighted graph G and edge weights to processing times of jobs. The graph G describes incompatibilities between jobs, i.e., jobs corresponding to adjacent edges cannot be scheduled (resp., colored) in the same batch (resp., by the same color).

Related work. It is well known that for general graphs it is NP-hard to approximate the classical edge-coloring problem within a factor less than 4/3 [12]; for bipartite graphs the problem becomes polynomial [15]. The MEC problem is known to be non approximable within a factor less than 7/6 even for cubic planar bipartite graphs with edge weights $w(e) \in \{1, 2, 3\}$, unless P=NP [3]. It is also NP-complete for complete graphs with bi-valued edge weights [2]. On the other hand, the MEC problem is known to be polynomial for a few special cases including bipartite graphs with edge weights $w(e) \in \{1, t\}$ [5], chains [7, 11, 13], stars of chains [17] and bounded degree trees [2]. It is interesting that the complexity of the MEC problem on trees remains open.

Concerning the approximability of the MEC problem, a natural greedy 2-approximation algorithm for general graphs has been proposed in [14]. For bipartite graphs of maximum degree $\Delta = 3$, an algorithm that attains the 7/6 inapproximability bound has been presented in [3]. For bipartite graphs of small maximum degrees, algorithms which improve the 2 approximation ratio have been also presented. However, the ratios of these algorithms either exceed 2 [7] or tend asymptotically to 2 [17, 2] as the maximum degree of the input graph increases. In [2] has been also presented a 3/2-approximation algorithm for trees, and an asymptotic 4/3-approximation algorithm for general graphs with bi-valued edge weights and arbitrarily large maximum degree Δ.

The MVC problem has been also studied extensively during last years. It is known to be non approximable within a factor less than 8/7 even for planar bipartite graphs, unless P=NP [5, 18]. This bound is tight for general bipartite graphs as an 8/7-approximation algorithm is also known [3, 18]. For the MVC problem on trees a PTAS has been presented in [18, 7], while the complexity of this case is open. Other results for the MVC problem on several graph classes have been also presented in [5, 3, 19, 18, 7, 6, 13].

Our results and organization of the paper. The two most interesting open questions about the MEC problem concern the existence of an approximation algorithm of ratio $2 - \delta$, for any constant $\delta > 0$, for general or bipartite graphs, and the complexity of the problem on trees. In this paper we present substantial improvements towards these questions. In the next section we present a PTAS for the MEC problem on trees; recall that the situation for the MVC problem on trees is the same: a PTAS is known while its complexity remains unknown. In Section 3, we succeed to beat the longstanding 2 approximation ratio for the MEC problem in bipartite graphs by presenting an 1.74-approximation algorithm. In addition, in Section 4, we adapt our algorithm for bipartite graphs to general graphs yielding an approximation ratio which also tends asymptotically to 1.74 as the maximum degree of the input graph increases. Finally, we conclude in Section 5.

Notation. In the following, we consider the MEC problem on an edge-weighted graph $G = (V, E)$, $|V| = n$, $|E| = m$, where a positive integer weight $w(e)$ is associated with each edge $e \in E$. We denote by $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$ a proper k-edge-coloring of G of weight $W = \sum_{i=1}^{k} w_i$, where $w_i = \max\{w(e)|e \in M_i\}$, $1 \leq i \leq k$. By $\mathcal{M}^* = \{M_1^*, M_2^*, \ldots, M_{k^*}^*\}$ we denote an optimal solution to the MEC problem on the graph G of weight $OPT = \sum_{i=1}^{k^*} w_i^*$.

By $d_G(u)$ (or simply $d(u)$) we denote the degree of vertex $u \in V$ and by $\Delta(G)$ (or simply Δ) the maximum degree of the graph G. For a subset of edges of G, $E' \subseteq E$, $|E'| = m'$, we denote by $G[E']$ the subgraph of G induced by the edges in E' and by $\langle E' \rangle = \langle e_1, e_2, \ldots, e_{m'} \rangle$ an ordering of the edges in E' such that $w(e_1) \geq w(e_2) \geq \cdots \geq w(e_{m'})$.

2 A PTAS for trees

In [2] a 2-approximation algorithm (Algorithm Trees) for the MEC problem on trees has been presented. This algorithm in conjunction with the 2-approximation algorithm for general graphs [14] has led to a $\frac{3}{2}$ ratio for trees. In this section we also exploit Algorithm Trees to derive a PTAS.

Next proposition is proven in [2].

Proposition 1. *Algorithm* Trees *constructs a solution of exactly Δ matchings in $O(|V| \cdot \Delta \cdot \log \Delta)$ time. For the weights of the matchings in this solution it holds that $w_1 = w_1^*$ and $w_i \leq w_{i-1}^*$, $2 \leq i \leq \Delta$.*

To obtain our scheme we shall use a transformation of our problem to the following list edge-coloring problem.

*List Edge-Coloring (*LEC*) problem*
Instance: A graph $G = (V, E)$, a set of k colors and a list of colors $\phi(e) \subseteq \{1, 2, \dots, k\}$ for each $e \in E$.
Question: Is there a k-edge-coloring of G such that each edge e is assigned a color in its list $\phi(e)$?

The MEC and LEC problems are strongly related: Given an edge-weighted graph $G = (V, E)$, consider a combination of k edges' weights $w_1 \geq w_2 \geq \cdots \geq w_k$ and answer to the following LEC problem on G: is there a k-edge-coloring of G such that each edge $e \in E$ is assigned a color in $\phi(e) = \{i : w(e) \leq w_i,\ 1 \leq i \leq k\}$? Then, a "yes" answer to this question corresponds to a feasible solution for the MEC problem of weight $W = \sum_{i=1}^{k} w_i$. There are $O(|E|^k)$ combinations of weights to be considered and an optimal solution to the MEC problem corresponds to the combination where W is minimized.

It is known that the LEC problem can be solved in $O(|E| \cdot \Delta^{3.5})$ time for trees [4], while it becomes NP-complete for bipartite graphs even for three colors ($k = 3$) [16].

Therefore, the next proposition follows.

Proposition 2. *For a fixed number of matchings k the* MEC *problem on trees is polynomial.*

Our scheme splits a tree $G = (V, E)$, into subgraphs $G[E_{1,j}]$ and $G[E_{j+1,m}]$ induced by the j heaviest and the $n - j$ lightest edges of G, respectively (by convention, we consider $G[E_{1,0}]$ as an empty subgraph). Our scheme depends on a parameter p such that all the edges of G of weights $w_1^*, w_2^*, \dots, w_{p-1}^*$ are in a subgraph $G[E_{1,j}]$. We obtain a solution for the whole graph by concatenating an optimal solution of at most $p-1$ colors for $G_{1,j}$, if there is one, and the solution obtained by Algorithm Trees for $G[E_{j+1,m}]$.

```
Algorithm Scheme($p$)
1: Let $\langle E \rangle = \langle e_1, e_2, \dots e_m \rangle$;
2: for $j = 0$ to $m$ do
3:    Split the graph into two edge induced subgraphs:
          - $G[E_{1,j}]$ induced by edges $e_1, e_2, \dots, e_j$
          - $G[E_{j+1,n}]$ induced by edges $e_{j+1}, e_{j+2}, \dots, e_m$
4:    if there is a solution for $G[E_{1,j}]$ with at most $p-1$ matchings then
5:       Find an optimal solution for $G[E_{1,j}]$ with at most $p-1$ matchings;
6:       Run Algorithm Trees for $G[E_{j+1,m}]$;
7:       Concatenate the two solutions found in Lines 5 and 6;
8: Return the best solution found;
```

Theorem 1. *Algorithm* Scheme(p) *is a PTAS for the* MEC *problem on trees.*

Proof. Consider the iteration j, $j \leq m$, of the algorithm where the weight of the heaviest edge in $G[E_{j+1,m}]$ equals to the weight of the i-th matching of an optimal solution, i.e. $w(e_{j+1}) = w_i^*$, $1 \leq i \leq p$.

The edges of $G[E_{1,j}]$ are a subset of those appeared in the $i-1$ heaviest matchings of the optimal solution. Thus, an optimal solution for $G[E_{1,j}]$ is of weight

$$OPT_{1,j} \leq w_1^* + w_2^* + \ldots + w_{i-1}^*.$$

The edges of $G[E_{j+1,m}]$ are a superset of those appeared in the $k^* - (i-1)$ lightest matchings of the optimal solution. The extra edges of $G[E_{j+1,m}]$ are of weight at most w_i^* and appear in an optimal solution into at most $i-1$ matchings. Thus, an optimal solution for $G[E_{j+1,m}]$ is of weight

$$OPT_{j+1,m} \leq w_i^* + w_{i+1}^* + \ldots + w_{k^*}^* + (i-1) \cdot w_i^* = i \cdot w_i^* + w_{i+1}^* + \ldots + w_{k^*}^*.$$

By Proposition 1, Algorithm TREES returns a solution for $G[E_{j+1,m}]$ of weight

$$\begin{aligned} W_{j+1,m} &\leq OPT_{j+1,m} + w_i^* - w_\Delta^* \\ &\leq i \cdot w_i^* + w_{i+1}^* + \ldots + w_{k^*}^* + w_i^* \\ &\leq (i+1) \cdot w_i^* + w_{i+1}^* + \ldots + w_{k^*}^*. \end{aligned}$$

Therefore, the solution found in this iteration j for the whole graph G is of weight

$$W_i = OPT_{1,j} + W_{j+1,m} \leq w_1^* + w_2^* + \ldots + w_{i-1}^* + (i+1) \cdot w_i^* + w_{i+1}^* + \ldots + w_{k^*}^*.$$

As the algorithm returns the best among the solutions found, we have p bounds on the weight W of this best solution, i.e.,

$$W_i \leq w_1^* + w_2^* + \ldots + w_{i-1}^* + (i+1) \cdot w_i^* + w_{i+1}^* + \ldots + w_{k^*}^*, 1 \leq i \leq p.$$

To derive our ratio we denote by $c_{ji}, 1 \leq i, j \leq p$, the coefficient of the weight w_j^* in the i-th bound on W and we find the solution of the system of linear equations $\mathbf{C} \cdot \mathbf{x}^T = \mathbf{1}^T$. Using the standard Gaussian elimination method, we get the following solution:

$$x_i = \frac{1}{i \cdot (H_p + 1)}, 1 \leq i \leq p.$$

By multiplying both sides of the i-th, $1 \leq i \leq p$, inequality by x_i and adding up all of them we have $\left(\sum_{i=1}^{p} \frac{1}{i \cdot (H_p + 1)} \right) \cdot W \leq OPT$, that is $\frac{W}{OPT} \leq \frac{H_p + 1}{H_p} = 1 + \frac{1}{H_p}$.

Algorithm SCHEME(p) iterates $|E|$ times. In each iteration: (i) an optimal solution, if any, with at most $p-1$ matchings for $G[E_{1,j}]$ is found by Proposition 2 in

$O(|E|^{p-1} \cdot |E| \cdot \Delta^{3.5})$ time and (ii) Algorithm Trees of complexity $O(|V| \cdot \Delta \cdot \log \Delta)$ is called for $G[E_{j+1,m}]$. Choosing p such that $\epsilon = \frac{1}{H_p}$ we get $p = O(2^{\frac{1}{\epsilon}})$. Consequently, we have a PTAS for the MEC problem on trees, that is an approximation ratio of $1 + \frac{1}{H_p} = 1 + \epsilon$ within time $O\left(|E|\left(|V| \cdot \Delta \cdot \log \Delta + |E|^p \cdot \Delta^{3.5}\right)\right)$. □

3 Beating the 2-approximation ratio for bipartite graphs

A promising idea in order to create an approximation algorithm for the MEC problem on bipartite graphs is to repeatedly partition the input graph into a number of edge induced subgraphs and then to find a solution for each of them independently. In fact, this is the idea behind the known approximation algorithms of ratios less than 2 for the MEC problem on bipartite graph [3, 7, 2], as well as the 8/7-approximation algorithm for the MVC problem on bipartite graphs [3, 18]. However all known algorithms for the MEC problem that follow this idea achieve ratios which either exceed 2 or tend asymptotically to 2 as the maximum degree of the input graph increases. In this section we exploit the same idea and we are able to show an 1.74-approximation ratio for the MEC problem on bipartite graphs.

Consider an ordering $\langle E \rangle = \langle e_1, e_2, \ldots, e_m \rangle$ of the edges of G. Let us denote by (p, q), $0 \leq p < q \leq m$, a partition of G into subgraphs $G[E_{1,p}]$, $G[E_{p+1,q}]$ and $G[E_{q+1,m}]$; by convention, we define $E_{1,0} = \emptyset$ and $E_{0,q} = E_{1,q}$. By $\Delta_{1,q}$ we denote the maximum degree of the subgraph $G[E_{1,q}]$. For a partition (p, q) of G, we define a *critical set of edges* $A \subseteq E_{p+1,q}$, such that each vertex $u \in V$ of degree $d_{1,q}(u) > \Delta_{1,p}$ has degree $d_{1,q}(u) - \Delta_{1,p} \leq d_A(u) \leq \Delta_{1,q} - \Delta_{1,p}$. The proposed algorithm relies on the existence of such a critical set of edges A: a solution for the subgraph $G[E_{1,q}]$ is found by concatenating a $\Delta_{1,p}$-coloring solution for the subgraph $G[E_{1,q} \setminus A]$ and a $(\Delta_{1,q} - \Delta_{1,p})$-coloring solution for the subgraph $G[A]$, if A exists, and by a $\Delta_{1,q}$-coloring of the subgraph $G[E_{1,q}]$, otherwise. For each partition (p, q), the algorithm computes a solution for the input graph G by concatenating a solution for $G[E_{1,q}]$ and a Δ-coloring solution for $G[E_{q+1,m}]$. The algorithm computes also a Δ-coloring solution for the input graph and returns the best among them.

The following lemma shows that the check in Line 4 of Algorithm Bipartite can be done in polynomial time.

Lemma 1. *For a partition (p, q) of a graph $G = (V, E)$, a critical set of edges A, if any, can be found in $O(|V|^3)$ time.*

Proof. A (g, f)-factor of a graph G is a spanning subgraph F such that $g(u) \leq d_F(u) \leq f(u)$, for all $u \in V$. Recall that $A \subseteq E_{p+1,q}$ and consider the subgraph $G[E_{p+1,q}]$. For each vertex u of $G[E_{p+1,q}]$ we define $g(u) = \max\{0, d_{1,q}(u) - \Delta_{1,p}\}$ and $f(u) = \Delta_{1,q} - \Delta_{1,p}$. Then, there exists a critical set of edges $A \subseteq E_{p+1,q}$ if and only if there exists a (g, f)-factor in $G[E_{p+1,q}]$. It is known that such a factor, if any, can be found in $O(|V|^3)$ time [1]. □

Theorem 2. *Algorithm* Bipartite *achieves an* 1.74*-approximation ratio for the* MEC *problem on bipartite graphs.*

Algorithm Bipartite

```
 1: Find a $\Delta$-coloring solution for $G$;
 2: for $p = 0$ to $m - 1$ do
 3:    for $q = p + 1$ to $m$ do
 4:       Find, if any, a critical set of edges $A$ in $G[E_{p+1,q}]$;
 5:       if $A$ exists then
 6:          Find a $\Delta_{1,p}$-coloring solution for $G[E_{1,q} \setminus A]$;
 7:          Find a $(\Delta_{1,q} - \Delta_{1,p})$-coloring solution for $G[A]$;
 8:       else
 9:          Find a $\Delta_{1,q}$-coloring solution for $G[E_{1,q}]$;
10:       Find a $\Delta$-coloring solution for $G[E_{q+1,m}]$;
11:       Find a solution for $G$ by concatenating the solutions found either in Lines 6,7
          or in Line 9 with the one found in Line 10;
12: Return the best among the solutions found in Lines 1 and 11;
```

Proof. The solution obtained by a Δ-coloring of the input graph computed in Line 1 of the algorithm is of weight $W_1 \leq \Delta \cdot w_1^*$.

Consider the partition (p, q) of G where $w(e_{p+1}) = w^*_{\lceil \frac{i}{2} \rceil}$ and $w(e_{q+1}) = w_i^*$, for $2 \leq i \leq \Delta$ (recall that $w_1^* \geq w_2^* \geq \cdots \geq w_{k^*}^*$ and $k^* \geq \Delta$). In such an iteration, all the edges in $E_{1,p}$ belong to $\lceil \frac{i}{2} \rceil - 1 \geq \Delta_{1,p}$ matchings of an optimal solution $\mathcal{M}^*$, and all the edges in $E_{1,q}$ belong to $i - 1 \geq \Delta_{1,q}$ colors of an optimal solution $\mathcal{M}^*$.

If $\Delta_{1,q} = \Delta_{1,p}$ then the set A does not exist. Hence, a $\Delta_{1,q}$-coloring of $G[E_{1,q}]$ yields a solution of weight at most $\left(\lceil \frac{i}{2} \rceil - 1\right) \cdot w_1^*$ for this subgraph.

If $\Delta_{1,q} > \Delta_{1,p}$ then a critical set of edges A exists. Indeed, in this case the matchings $M^*_{\lceil \frac{i}{2} \rceil}, M^*_{\lceil \frac{i}{2} \rceil + 1}, \ldots, M^*_{i-1}$ of $\mathcal{M}^*$ always contain some edges from $E_{p+1,q}$, for otherwise all the edges in $E_{1,q}$ belong to $\lceil \frac{i}{2} \rceil - 1$ matchings of $\mathcal{M}^*$, a contradiction; these edges of $E_{p+1,q}$ could be a critical set of edges A for the partition (p, q). Thus, a $\Delta_{1,p}$-coloring solution of $G[E_{1,q} \setminus A]$ and a $(\Delta_{1,q} - \Delta_{1,p})$-coloring solution for $G[A]$ yield a solution for the subgraph $G[E_{1,q}]$ of weight at most $\Delta_{1,p} \cdot w_1^* + (\Delta_{1,q} - \Delta_{1,p}) \cdot w^*_{\lceil \frac{i}{2} \rceil} \leq \left(\lceil \frac{i}{2} \rceil - 1\right) \cdot w_1^* + \lfloor \frac{i}{2} \rfloor \cdot w^*_{\lceil \frac{i}{2} \rceil}$, since $\Delta_{1,p} \leq \lceil \frac{i}{2} \rceil - 1$, $\Delta_{1,q} \leq i - 1$ and $w_1^* \geq w^*_{\lceil \frac{i}{2} \rceil}$.

Finally, a Δ-coloring solution for $G[E_{q+1,m}]$ is of weight at most $\Delta \cdot w_i^*$.

Hence, for such a partition (p, q) the algorithm finds a solution for the whole input graph of weight

$$W_i \leq \left(\left\lceil \frac{i}{2} \right\rceil - 1 \right) \cdot w_1^* + \left\lfloor \frac{i}{2} \right\rfloor \cdot w^*_{\lceil \frac{i}{2} \rceil} + \Delta \cdot w_i^*, \; 2 \leq i \leq \Delta.$$

As in the case of trees, the algorithm returns the best among the solutions found. Hence, we have Δ bounds on the weight W of this best solution, i.e.,

$$\begin{aligned} &W_1 \leq \Delta \cdot w_1^*, \text{ if } i = 1, \text{ and} \\ &W_i \leq \left(\left\lceil \frac{i}{2} \right\rceil - 1 \right) \cdot w_1^* + \left\lfloor \frac{i}{2} \right\rfloor \cdot w^*_{\lceil \frac{i}{2} \rceil} + \Delta \cdot w_i^*, \text{ if } 2 \leq i \leq \Delta. \end{aligned}$$

Solving again the system of linear equations $\mathbf{C}\cdot\mathbf{x}^T = \mathbf{1}^T$, where $c_{ji}, 1 \leq i, j \leq \Delta$, is the coefficient of the weight w_j^* in the i-th bound on W, we get the following solution for the case where the maximum degree of the graph is a power of 2:

$$x_i = \begin{cases} \sum_{j=0}^{\left\lfloor \log \frac{\Delta}{i} \right\rfloor} \left(-\left(\frac{-1}{\Delta}\right)^{j+1} \sum_{y=1}^{2^j} \left(\prod_{z=1}^{j} \left(2^{z-1}(i-1) + \left\lceil \frac{y}{2^{j-z+1}} - \frac{1}{2} \right\rceil \right) \right) \right), & \text{if } \Delta \geq i \geq 2 \\ \frac{1}{\Delta} \left(1 - x_2 - \sum_{j=3}^{\Delta} \left(\left\lceil \frac{j}{2} \right\rceil - 1 \right) x_j \right), & \text{if } i = 1. \end{cases}$$

For the case where the maximum degree of the input graph is not a power of 2 the solution of the system becomes more complicated:

$$x_i = \sum_{j=0}^{\left\lfloor \log \frac{\Delta}{i} \right\rfloor} \left(-\left(\frac{-1}{\Delta}\right)^{j+1} \sum_{y=1}^{2^j} \left(\prod_{z=1}^{j} \left(2^{z-1}(i-1) + \left\lceil \frac{y}{2^{j-z+1}} - \frac{1}{2} \right\rceil \right) \right) \right) -$$

$$\left(\frac{-1}{\Delta}\right)^{\left\lfloor \log \frac{\Delta}{i} \right\rfloor + 2} \sum_{y=1}^{\left(\Delta - i + 1 - \sum_{r=0}^{\left\lfloor \log \frac{\Delta}{i} \right\rfloor} ((i-1)2^r)\right)} \left(\prod_{z=1}^{\left\lfloor \log \frac{\Delta}{i} \right\rfloor + 1} \left(2^{z-1}(i-1) + \left\lceil \frac{y}{2^{\left\lfloor \log \frac{\Delta}{i} \right\rfloor + 2 - z}} - \frac{1}{2} \right\rceil \right) \right)$$

while x_1 is the same as in the previous case.

For both cases, it holds that $\frac{W}{OPT} \leq \frac{1}{\sum_{i=1}^{\Delta} x_i}$. Using `Mathematica`, we computed the above ratio for quite large values of Δ, and it is found to tend to 1.74 (see Table 1 in our concluding section). □

It is an interesting question is whether a close formula for this ratio can be computed.

4 An adaptation for general graphs

The idea of splitting the input graph into three edge induced subgraphs and creating a Δ-coloring solution for each of them can be also exploited for general graphs. However, in this case, it is NP-complete to find, if any, a Δ-coloring solution of the input graph [12]. Instead of this, a $(\Delta + 1)$-coloring solution can be found in polynomial time [9]. Note that Lemma 1 holds for general graphs, and hence a critical set of edges A, if any, can be found in polynomial time.

Theorem 3. *There is an asymptotic* 1.74*-approximation ratio for the* MEC *problem on general graphs.*

Proof. The analysis is almost the same as in the bipartite case. Considering the partition (p, q) where $w(e_{p+1}) = w^*_{\lceil \frac{i}{2} \rceil}$ and $w(e_{q+1}) = w_i^*$; the difference is that if the set A exists then at most (i) a $\lceil \frac{i}{2} \rceil$-coloring solution is created for $G[E_{1,q} \setminus A]$, (ii) a $\left(\lfloor \frac{i}{2} \rfloor + 1\right)$-coloring solution is created for $G[A]$, and (iii) a $(\Delta + 1)$-coloring solution is created for $G[E_{q+1,m}]$.

Therefore, as in the previous case we have $\varDelta$ bounds on the weight W of this best solution, i.e.,

$$\begin{aligned}
&W_1 \leq (\varDelta+1)\cdot w_1^*, \text{ if } i=1,\\
&W_2 \leq w_1^* + (\varDelta+1)\cdot w_2^*, \text{ if } i=2,\\
&W_3 \leq w_1^* + w_2^* + (\varDelta+1)\cdot w_3^*, \text{ if } i=3,\\
&W_4 \leq w_1^* + 3w_2^* + (\varDelta+1)\cdot w_4^*, \text{ if } i=4, \text{ and}\\
&W_i \leq \left\lceil \frac{i}{2} \right\rceil \cdot w_1^* + \left(\left\lfloor \frac{i}{2} \right\rfloor + 1 \right) \cdot w^*_{\lceil \frac{i}{2} \rceil} + (\varDelta+1)\cdot w_i^*, \text{ if } 5 \leq i \leq \varDelta.
\end{aligned}$$

Note that, for $i = 2, 3$ or 4, the subgraph $G[E_{1,q} \setminus A]$ is of maximum degree at most one, and hence an optimal solution of one matching is created in this case. Analogous remark can be done for the subgraph $G[A]$ for $i = 3$.

Solving the adapted system of linear equations as in the proof of Theorem 2, we find a solution x_i, $1 \leq i \leq \varDelta$, such that $\frac{W}{OPT} \leq \frac{1}{\sum_{i=1}^{\varDelta} x_i}$. For the case where the maximum degree of the graph is a power of 2, this solution is:

$$x_i = \begin{cases} \sum_{j=0}^{\left\lfloor \log \frac{\varDelta}{i} \right\rfloor} \left(- \left(\frac{-1}{\varDelta+1} \right)^{j+1} \sum_{y=1}^{2^j} \left(\prod_{z=1}^{j} \left(2^{z-1}(i-1) + \left\lceil \frac{y}{2^{j-z+1}} + \frac{1}{2} \right\rceil \right) \right) \right), & \text{if } \varDelta \geq i \geq 5\\ \frac{1-4x_7-5x_8}{\varDelta+1}, & \text{if } i=4\\ \frac{1-3x_5-4x_6}{\varDelta+1}, & \text{if } i=3\\ \frac{1-x_3-3x_4}{\varDelta+1}, & \text{if } i=2\\ \frac{1}{\varDelta+1}\left(1-x_2-x_3-x_4-\sum_{j=5}^{\varDelta} \left\lceil \frac{j}{2} \right\rceil x_j \right), & \text{if } i=1. \end{cases}$$

Using again `Mathematica` we computed this ratio which tends asymptotically to 1.74 as $\varDelta$ increases (see Table 1 below). □

5 Conclusions

We presented new results towards two open questions for the MEC problem; its complexity on trees, and the existence of an approximation algorithm for general and bipartite graphs of ratio $2 - \delta$, for any constant δ. We decrease the approximability gaps for both questions by presenting a PTAS for trees (improving the known 3/2 approximation ratio), and an 1.74-approximation algorithm for bipartite and general graphs (see Table 1).

$\varDelta$	2^3	2^6	2^9	2^{12}	2^{15}	2^{18}
Bipartite graphs	1.60188	1.71809	1.73409	1.73612	1.73637	1.73640
General graphs	1.99605	1.78855	1.74345	1.73730	1.73652	1.73642

Table 1. Approximation ratios for the MEC problem on general and bipartite graphs for different values of $\varDelta$.

To explain the behavior of our approximation ratios it is worth to observe that the ratio for bipartite graphs increases with $\varDelta$, while for general graphs decreases with Δ. This is because we use $(\Delta + 1)$-colorings for general graphs, instead of Δ-colorings for bipartite graphs. Recall that the standard $\frac{\Delta+1}{\Delta}$-approximation ratio for the classical edge-coloring problem (implied by Vizing's Theorem) also decreases with Δ.

Acknowledgement. We would like to thank Professor Paris Vassalos for his help on linear algebra methods used in this paper.

References

1. R. P. Anstee. An algorithmic proof of Tutte's f-factor theorem. *Journal of Algorithms*, 6:112–131, 1985.
2. N. Bourgeois, G. Lucarelli, I. Milis, and V. Th. Paschos. Approximating the max-edge-coloring problem. *Theoretical Computer Science*, 411:3055–3067, 2010. (also in Proc. IWOCA'09).
3. D. de Werra, M. Demange, B. Escoffier, J. Monnot, and V. Th. Paschos. Weighted coloring on planar, bipartite and split graphs: Complexity and approximation. *Discrete Applied Mathematics*, 157:819–832, 2009. (also in Proc. ISAAC'04).
4. D. de Werra, A. J. Hoffman, N. V. R. Mahadev, and U. N. Peled. Restrictions and preassignments in preemptive open shop scheduling. *Discrete Applied Mathematics*, 68:169–188, 1996.
5. M. Demange, D. de Werra, J. Monnot, and V. Th. Paschos. Time slot scheduling of compatible jobs. *Journal of Scheduling*, 10:111–127, 2007. (also in Proc. WG'02).
6. L. Epstein and A. Levin. On the max coloring problem. In *5th Workshop on Approximation and Online Algorithms (WAOA'07)*, volume 4927 of *LNCS*, pages 142–155. Springer, 2008.
7. B. Escoffier, J. Monnot, and V. Th. Paschos. Weighted coloring: further complexity and approximability results. *Information Processing Letters*, 97:98–103, 2006. (also in Proc. ICTCS'05).
8. G. Finke, V. Jost, M. Queyranne, and A. Sebő. Batch processing with interval graph compatibilities between tasks. *Discrete Applied Mathematics*, 156:556–568, 2008.
9. H. N. Gabow, T. Nishizeki, O. Kariv, D. Leven, and O. Terada. Algorithms for edge-coloring graphs. Technical Report TRECIS-8501, Tohoku University, 1985.
10. I. S. Gopal and C. Wong. Minimizing the number of switchings in a SS/TDMA system. *IEEE Transactions On Communications*, 33:497–501, 1985.
11. M. M. Halldorsson and H. Shachnai. Batch coloring flat graphs and thin. In *11th Scandinavian Workshop on Algorithm Theory (SWAT'08)*, volume 5124 of *LNCS*, pages 198–209. Springer, 2008.
12. I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10:718–720, 1981.
13. T. Kavitha and J. Mestre. Max-coloring paths: Tight bounds and extensions. In *20th International Symposium on Algorithms and Computation (ISAAC'09)*, volume 5878 of *LNCS*, pages 87–96. Springer, 2009.
14. A. Kesselman and K. Kogan. Nonpreemptive scheduling of optical switches. *IEEE Transactions on Communications*, 55:1212–1219, 2007. (also in Proc. GLOBECOM'04).
15. D. König. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen*, 77:453–465, 1916.

16. M. Kubale. Some results concerning the complexity of restricted colorings of graphs. *Discrete Applied Mathematics*, 36:35–46, 1992.
17. G. Lucarelli, I. Milis, and V. Th. Paschos. On the max-weight edge coloring problem. *Journal of Combinatorial Optimization*, In Press. (also in Proc. MISTA'07).
18. S. V. Pemmaraju and R. Raman. Approximation algorithms for the max-coloring problem. In *32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 1064–1075. Springer, 2005.
19. S. V. Pemmaraju, R. Raman, and K. R. Varadarajan. Buffer minimization using max-coloring. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'04)*, pages 562–571, 2004.