UNIVERSITE PARIS DAUPHINE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
CNRS FRE 3234

CAHIER DU LAMSADE 284

Juillet 2009

Approximating the max-weight edge coloring problem

N. Bourgeois, G. Lucarelli, I. Milis, V. Th. Paschos

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Approximating the max edge-coloring problem☆

Nicolas Bourgeois[a], Giorgio Lucarelli[b,1], Ioannis Milis[b], Vangelis Th. Paschos[a]

[a]*LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine, France*
[b]*Dept. of Informatics, Athens University of Economics and Business, Greece*

Abstract

The max edge-coloring problem is a natural weighted generalization of the classical edge-coloring problem arising in the domain of communication systems. In this problem each color class is assigned the weight of the heaviest edge in this class and the objective is to find a proper edge-coloring of the input graph minimizing the sum of all color classes' weights. We present new approximation results, that improve substantially the known ones, for several variants of the problem with respect to the class of the underlying graph. In particular, we deal with variants which either are known to be NP-hard (general and bipartite graphs) or are proven to be NP-hard in this paper (complete graphs with bi-valued edge weights) or their complexity question still remains open (trees).

Key words: max edge-coloring, approximation algorithms, complexity

1. Introduction

We study a weighted generalization of the classical edge-coloring problem which takes as input a graph $G = (V, E)$ and a positive integer weight $w(e)$, for each edge $e \in E$. For a proper edge-coloring of G, $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$, each color class (matching) $M_i \subseteq E$ is assigned the weight of the heaviest edge in this class, i.e., $w_i = \max\{w(e)|e \in M_i\}$, $1 \leq i \leq k$. The objective of the problem is to find a proper edge-coloring of G, such that the sum of all color classes' weights, $W = \sum_{i=1}^{k} w_i$, is minimized. Clearly, for unit edge weights our problem reduces to the classical edge-coloring problem. We refer to this problem as Max Edge-Coloring (MEC) problem, respectively to the analogous

---

☆Parts of this work were presented in WAOA'08 [21] and IWOCA'09 [4].

Email addresses: nbourgeo@phare.normalesup.org (Nicolas Bourgeois), gluc@aueb.gr (Giorgio Lucarelli), milis@aueb.gr (Ioannis Milis), paschos@lamsade.dauphine.fr (Vangelis Th. Paschos)

[1]This work has been funded by the project PENED 2003. The project is cofinanced 75% of public expenditure through EC–European Social Fund, 25% of public expenditure through Ministry of Development–General Secretariat of Research and Technology of Greece and through private sector, under measure 8.3 of Operational Programme "Competitiveness" in the 3rd Community Support Programme.

weighted generalization of the classical vertex-coloring problem which is known as Max (Vertex-)Coloring (MVC) problem, [25, 24].

The MEC problem arises in switch based communication systems, like `SS/TDMA` [15, 18], where messages are to be transmitted in a single hop from senders to receivers through direct connections established by an underlying network. Any node of such a system can participate in at most one transmission at a time, while the transmission of messages between pairwise disjoint pairs of nodes can take place simultaneously. The scheduler of such a system establishes successive configurations of the underlying network, each one routing a non-conflicting subset of messages from senders to receivers. Given the transmission times of messages, the transmission time of each configuration equals to the longest message transmitted. The aim is to find a sequence of configurations minimizing the transmission time of all messages. It is easy to see that the above situation corresponds directly to the MEC problem: senders and receivers correspond to the vertices of the graph G, (transmission times of) messages correspond to (weights of) edges of G and configurations correspond to color classes (matchings).

The MEC problem can be also viewed as a parallel batch scheduling problem with conflicts between jobs [10, 13]. According to the standard three field notation for scheduling problems, our problem is denoted by $1 \mid p-batch,\ E(G) \mid C_{max}$. In this problem, 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).

In practical applications in this context there is, however, a non negligible delay, say d, to set up each schedule phase (color class). The presence of such a delay, in the instance of the MEC problem, can be easily handled: by adding d to the weight of all edges of G, the weight of each color class will be also increased by d, incorporating its set up delay. Furthermore, a standard idea to decrease the completion time of a schedule is to allow preemption, i.e., interrupt the service of a (set of) scheduled activity(ies) and complete it (them) latter. It is obvious that allowing preemption in the MEC problem will result in increasing the number of the phases in a schedule. In this case, the presence of a set up delay d plays a crucial role in the hardness of the (preemptive) MEC problem [15, 7, 1].

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 [17]; for bipartite graphs the problem becomes polynomial [19]. 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 [8]. On the other hand, the MEC problem is known to be polynomial for a few special cases including bipartite graphs with edge weights $w(e) \in \{1, t\}$ [10], chains [12, 16], stars of chains and bounded degree trees [22]. 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 [18]. For bipartite graphs of maximum degree $\Delta = 3$, an algorithm that attains the 7/6 inapproximability bound has been presented in [8]. For bipartite graphs, algorithms have been also presented improving the 2 approximation ratio for general graphs. In fact, algorithms presented in [12] and [22] achieve better than 2 ratios for bipartite graphs of $\Delta \leq 7$, and $\Delta \leq 12$, respectively. However, for bipartite graphs of greater maximum degrees the ratios of both algorithms become greater than 2 and they are dominated by the 2-approximation algorithm for general graphs.

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 [10, 24]. This bound is tight for general bipartite graphs as an 8/7-approximation algorithm is also known [8, 24]. For the MVC problem on trees a PTAS has been presented in [24, 12]; however, the complexity for this case is an open question, as for the MEC problem. Other results for the MVC problem on several graph classes have been also presented in [10, 8, 25, 24, 12, 11]. Notice that the MEC problem, on a general graph G, is equivalent to the MVC problem on the line graph of G and thus any algorithm for the MVC problem applies also to the MEC problem. However, this is true only for graph classes that are closed under line graph transformation. This is the case of general graphs or chains but not of bipartite graphs or trees.

Our results and organization of the paper. Although a 2-approximation algorithm is known for the MEC problem on general graphs, no algorithm of ratio $2 - \delta$, for any small constant $\delta > 0$, is known for any special graph class. Apart from their theoretical interest, special graph classes, like bipartite graphs and trees, are also motivated by practical applications [18, 24]. Towards this direction we present approximation algorithms for the MEC problem on general and bipartite graphs, trees and graphs with bi-valued edge weights.

The next section starts with our notation and a remark on the known greedy 2-approximation algorithm [18]. By combining this remark with a simple idea, we present a first approximation algorithm for general and bipartite graphs which already beats the best known ratios for these classes. In Section 3, we present a new approximation algorithm for the MEC problem on bipartite graphs of ratio $\frac{2(\Delta+1)^3}{\Delta^3+5\Delta^2+5\Delta+3-2(-1/\Delta)^\Delta}$, which improves furthermore the known ratios for graphs of maximum degree $\Delta \geq 7$. In Section 4, we present a polynomial 3/2-approximation algorithm for trees. This is the first algorithm, for any special graph class, of a ratio strictly less than the known ratio of 2 for general graphs. In Section 5, we propose two moderately exponential approximation algorithms for trees that improve the 3/2 ratio with running time much better than that needed for the computation of an optimal solution. In Section 6, we prove that the MEC problem is NP-complete even for complete graphs with bi-valued edge weights, and we give an asymptotic $\frac{4}{3}$-approximation algorithm for general graphs with bi-valued edge weights and arbitrarily large maximum degree Δ. Finally, we conclude in Section 7.

2. Notation and Preliminaries

In the following, we consider the MEC problem on a graph $G = (V, E)$, where $|V| = n$, $|E| = m$ and 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 $OPT = \sum_{i=1}^{k^*} w_i^*$. As in the sequel we deal only with edge-coloring of graphs, the terms k-coloring or k-colorable graph always refer to edge-coloring. We also use the terms color class and matching interchangeably.

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 $w(e_1) \geq w(e_2) \geq \cdots \geq w(e_{m'})$.

We call a solution $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$ to the MEC problem *nice* if: (i) $w_1 \geq w_2 \geq \cdots \geq w_k$, and (ii) each matching M_i is maximal in the subgraph $G[\bigcup_{j=i}^{k} M_j]$. Due to the next straightforward proposition (see also [22]), w.l.o.g., we consider any, suboptimal or optimal, solution to the MEC problem to be a nice one.

Proposition 1. *Any solution to the* MEC *problem can be transformed into a nice one, without increasing its total weight. For the number of matchings, k, in such a solution it holds that* $\Delta \leq k \leq 2\Delta - 1$.

The most interesting and general result for the MEC problem is due to Kesselman and Kogan [18] who proposed the following greedy algorithm:

Algorithm KK
1. Let $\langle E \rangle = \langle e_1, e_2, \ldots, e_m \rangle$;
2. For i=1,2,..., m do
3. Insert e_i into the first matching not containing other edges adjacent to e_i;

In [18], it has been shown that Algorithm KK is a 2-approximation one and an example has been presented yielding an approximation ratio of $2 - \frac{1}{\Delta}$. By a slightly tighter analysis we prove here the next lemma.

Lemma 2. *Algorithm KK achieves an approximation ratio of* $\min\{2 - \frac{w_1^*}{OPT}, 2 - \frac{1}{\Delta}\}$ *for the* MEC *problem.*

Proof. The solution, $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$, that Algorithm KK returns is, by its construction, a nice one. Let e be the first edge that the algorithm inserts into matching M_i; then it holds that $w_i = w(e)$. Let E_i be the set of edges preceding e in $\langle E \rangle$ and edge e itself, and Δ_i be the maximum degree of the subgraph $G[E_i]$. The optimal solution for the MEC problem on the graph $G[E_i]$

contains $i^* \geq \Delta_i$ matchings each one of weight at least w_i, that is $w_i \leq w^*_{i^*}$. By Proposition 1, the matchings constructed by Algorithm KK for the graph $G[E_i]$ are $i \leq 2\Delta_i - 1 \leq 2i^* - 1$, that is $i^* \geq \lceil \frac{i+1}{2} \rceil$. Hence, $w_i \leq w^*_{i^*} \leq w^*_{\lceil \frac{i+1}{2} \rceil}$.

Summing up the above bounds for all w_i's, $1 \leq i \leq k \leq 2\Delta - 1$, we obtain $W \leq \sum_{i=1}^{2\Delta-1} w_i \leq w_1^* + 2(\sum_{i=2}^{\Delta} w_i^*) = 2(\sum_{i=1}^{\Delta} w_i^*) - w_1^*$. As $k^* \geq \Delta$, it follows that $\sum_{i=1}^{\Delta} w_i^* \leq OPT$. Therefore, $\frac{W}{OPT} \leq 2 - \frac{w_1^*}{OPT}$ and also $\frac{W}{OPT} \leq$ $\frac{2\sum_{i=1}^{\Delta} w_i^* - w_1^*}{\sum_{i=1}^{\Delta} w_i^*} \leq 2 - \frac{w_1^*}{\sum_{i=1}^{\Delta} w_i^*} \leq 2 - \frac{w_1^*}{\Delta \cdot w_1^*} = 2 - \frac{1}{\Delta}$. □

It is well known that a general graph is $(\Delta+1)$-colorable [26] and a bipartite one is Δ-colorable [19]. Such a coloring can be found in polynomial time and yields a feasible, but in general not optimal, solution for the MEC problem. Intuitively, a solution obtained this way will be close to an optimal one when the edge weights are close to each other, while the Algorithm KK performs better in the opposite case. Next theorem follows by selecting the best among the two solutions found by Algorithm KK and a $(\Delta + 1)-$ or Δ-coloring of the input graph.

Theorem 3. *There is an approximation algorithm for the* MEC *problem of ratio* $2 - \frac{2}{\Delta+1}$ *for bipartite graphs and* $2 - \frac{2}{\Delta+2}$ *for general graphs.*

Proof. By Lemma 2, a solution found by Algorithm KK is of weight $W \leq 2OPT - w_1^*$. Any Δ-coloring of a bipartite graph yields a solution for the MEC problem of weight $W \leq \Delta w_1^*$. Multiplying both sides of the second inequality with $1/\Delta$ and adding this to the first one we obtain: $(1 + \frac{1}{\Delta})W \leq 2OPT$, that is $W \leq (2 - \frac{2}{\Delta+1})OPT$. For general graphs we simply consider a $(\Delta + 1)$-coloring. □

For the tightness of our analysis for bipartite graphs, consider the instance of the MEC problem shown in Figure 1(a); a similar example can be also constructed for general graphs. The weight of an optimal solution to this instance is $2C+\epsilon$ (Figure 1(b)), the weight of the solution of Algorithm KK is $3C$ (Figure 1(c)) and the weight of a solution found by a Δ-coloring (Figure 1(d)) is also $3C$. By selecting either solution a ratio of $\frac{3C}{2C+\epsilon} \simeq \frac{3}{2} = 2 - \frac{2}{\Delta+1}$ is attained.

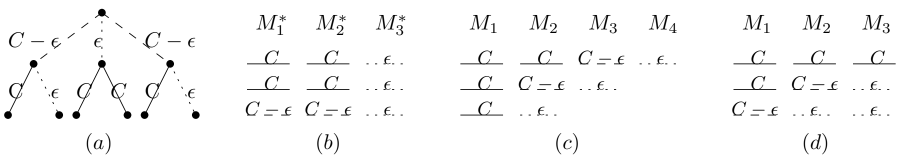

Figure 1: A tight example for the ratio of Theorem 3 for bipartite graphs ($\Delta = 3$, $C >> \epsilon$).

Note that the ratios of Theorem 3 are better than $2-\frac{1}{\Delta}$ for any $\Delta \geq 3$. More interestingly, the ratio for bipartite graphs is better than those of algorithms in [12], for $\Delta \geq 4$, and in [22], for $\Delta \geq 9$.

3. Bipartite graphs

All known approximation algorithms [8, 12, 22] of ratios less than 2 for the MEC problem on a bipartite graph $G = (V, E)$ are based on the following general idea: Consider an ordering $\langle E \rangle = \langle e_1, e_2, \ldots, e_m \rangle$ of the edges of G, and let $E_{p,q} = \{e_p, e_{p+1}, \ldots, e_q\}$. Repeatedly, partition the graph G into three edge induced subgraphs $G[E_{1,p}]$, $G[E_{p+1,q}]$ and $G[E_{q+1,m}]$, find a solution for the whole graph G by considering the MEC problem on these three subgraphs and return the best among the solutions found. Depending on how the problem is handled for each subgraph and the analysis followed, this general idea leads to different algorithms and approximation ratios. Notice that the same approach is employed by the 8/7-approximation algorithm for the MVC problem [8, 24].

In [21] we have also exploited this approach and we have proposed an algorithm for bipartite graphs of ratio $\frac{2\Delta^3}{\Delta^3+\Delta^2+\Delta-1}$. In this section we further explore the limitations of this approach and we present a new algorithm for the MEC problem on bipartite graphs, which improves all the previous ratios for $\Delta \geq 7$.

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 call *critical matching* a matching $M \subseteq E_{p+1,q}$ which saturates all the vertices of $G[E_{1,q}]$ of degree $\Delta_{1,q}$. The proposed algorithm relies on the existence of such a critical matching M: a solution for the subgraph $G[E_{1,q}]$ is found by concatenating a $(\Delta_{1,q}-1)$-coloring solution for the subgraph $G[E_{1,q}\backslash M]$ and the matching M, if 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.

Algorithm BIPARTITE

1. Find a Δ-coloring solution for G;
2. Let $\langle E \rangle = \langle e_1, e_2, \ldots, e_m \rangle$
3. For $p = 0, 1, 2, \ldots, m-1$ do
4.   For $q = p + 1$ to m do
5.   Find, if any, a critical matching M in $G[E_{p+1,q}]$;
6.   If M exists
         then find a $(\Delta_{1,q}-1)$-coloring solution for $G[E_{1,q} \setminus M]$
         else find a $\Delta_{1,q}$-coloring solution for $G[E_{1,q}]$;
7.   Find a Δ-coloring solution for $G[E_{q+1,m}]$;
8.   Find a solution for G by concatenating the solutions
       found in Lines 6 and 7 and matching M, if exists;
9. Return the best among the solutions found in Lines 1 and 8;

The next proposition deals with finding, if any, a critical matching M in Line 5 of the algorithm.

Proposition 4. *For a partition (p, q) of a graph $G = (V, E)$, a critical matching M, if any, can be found in $O(n^{2.5})$ time.*

Proof. Let U be the set of vertices of $G[E_{1,q}]$ of degree $\Delta_{1,q}$ to be saturated by a critical matching $M \subseteq E_{p+1,q}$. Consider the graph $G' = (V', E')$ where V' consists of V and an additional vertex, if $|V|$ is odd, and E' consists of $E_{p+1,q}$ and all the edges between the vertices $V' \setminus U$ (i.e., the vertices $V' \setminus U$ induce a clique in G'). If there exists a perfect matching in G', then there exists a critical matching M, since no edges adjacent to U have been added in G'. Conversely, if there exists a critical matching M, then there exists a perfect matching in G', consisting of the edges of M plus the edges of a perfect matching in the complete subgraph of G' induced by its vertices not saturated by M. Therefore, a critical matching M, if any, can be found by looking for a perfect matching, if any, in G'. It is well known that this can be done in $O(n^{2.5})$ time [23]. □

Theorem 5. *Algorithm BIPARTITE achieves an approximation ratio of* $\dfrac{2(\Delta+1)^3}{\Delta^3 + 5\Delta^2 + 5\Delta + 3 - 2(-1/\Delta)^{\Delta}}$ *for the* MEC *problem on bipartite graphs.*

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 $w(e_{p+1}) = w_{i-1}^*$ 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 $E_{1,q}$ belong to $i - 1 \geq \Delta_{1,q}$ matchings of an optimal solution $\mathcal{M}^*$.

If $\Delta_{1,q} < i - 1$, then an $(i-2)$-coloring of $G[E_{1,q}]$ yields a solution of weight at most $(i-2) \cdot w_1^*$ for this subgraph.

If $\Delta_{1,q} = i - 1$ then a critical matching M exists. Indeed, in this case the $(i-1)$-th matching of $\mathcal{M}^*$ always contains some edges from $E_{p+1,q}$, for otherwise all the edges in $E_{1,q}$ belong to $i-2$ matchings of $\mathcal{M}^*$, a contradiction; these edges of $E_{p+1,q}$ could be a critical matching M for the partition (p, q). Thus, a $(i-2)$-coloring solution of $G[E_{1,q} \setminus M]$ and critical matching M yield a solution for the subgraph $G[E_{1,q}]$ of weight at most $(i-2) \cdot w_1^* + w_{i-1}^*$. Finally, a Δ-coloring solution for $G[E_{q+1,m}]$ is of cost 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

$$W_i \leq (i-2) \cdot w_1^* + w_{i-1}^* + \Delta \cdot w_i^*, \ 2 \leq i \leq \Delta.$$

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

$$\begin{aligned} &W \leq \Delta \cdot w_1^*, \text{ if } i = 1, \text{ and} \\ &W \leq (i-2) \cdot w_1^* + w_{i-1}^* + \Delta \cdot w_i^*, \text{ if } \ 2 \leq i \leq \Delta. \end{aligned}$$

To derive our ratio we denote by $c_{ji}, 1 \leq i, j \leq \Delta$, 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 $\mathbf{C} \cdot \mathbf{x}^T = \mathbf{1}^T$, that is

$$x_i = \begin{cases} \frac{1}{\Delta}, & \text{if} \quad i = \Delta \\ \frac{1}{\Delta+1}\left(1 - \left(\frac{-1}{\Delta}\right)^{\Delta-i+1}\right), & \text{if} \quad \Delta - 1 \geq i \geq 2 \\ \frac{1}{\Delta} - \sum_{j=0}^{\Delta-3}\left(\frac{\Delta-(j+2)}{\Delta} x_{\Delta-j}\right) - \frac{1}{\Delta}x_2, & \text{if} \quad i = 1. \end{cases}$$

By multiplying both sides of the i-th bound on W by x_i and adding all of them we have $\sum_{i=1}^{\Delta} x_i \cdot W \leq w_1^* + w_2^* + \cdots + w_\Delta^* \leq OPT$.

Hence, $\frac{W}{OPT} \leq \frac{1}{\sum_{i=1}^{\Delta} x_i}$, which after some algebra becomes

$$\frac{W}{OPT} \leq \frac{(\Delta+1)}{\frac{\Delta^3+3\Delta^2+\Delta-3}{2(\Delta^2-1)} - \frac{\left(\Delta^{2+(\Delta \bmod 2)}+(-1)^{\Delta}(\Delta-1)\right)}{(\Delta^2-1)\Delta^{\Delta}} - (\Delta-1)\sum_{i=1}^{\lfloor \Delta/2 \rfloor} \frac{2i}{\Delta^{2i}}}.$$

By differentiating both sides of the formula $\sum_{i=0}^{\lfloor \Delta/2 \rfloor} \left(\frac{1}{x^2}\right)^i = \frac{1-(x^{-2})^{\lfloor \Delta/2 \rfloor+1}}{1-x^{-2}}$ for the sum of geometric series we get

$$-\frac{1}{x}\sum_{i=1}^{\lfloor \Delta/2 \rfloor}\left(\frac{2i}{x^{2i}}\right) = \frac{-2x + 2x^{-2\lfloor \Delta/2 \rfloor+1} + 2\lfloor \Delta/2 \rfloor (x^2-1)x^{-2\lfloor \Delta/2 \rfloor-1}}{(x^2-1)^2}$$

and by using this last expression for $x = \Delta$ we finally get

$$\frac{W}{OPT} \leq \frac{2(\Delta+1)^3}{\Delta^3+5\Delta^2+5\Delta+3-2(-1/\Delta)^{\Delta}}. \qquad \square$$

Lines 5-8 of the algorithm are repeated $O(m^2)$ times. Finding a critical matching in Line 5, takes, by Proposition 4, $O(n^{2.5})$ time, while finding the colorings of the bipartite subgraphs of G in Lines 6 and 7, takes $O(m \log \Delta)$ time [6].

In Table 1 we compare the approximation ratios achieved by Algorithm BIPARTITE, as Δ increases, with the best known ones. Note that our algorithm is of the same complexity with that in [8], while the complexity of the algorithm in [22] is greater by a factor of $O(m^2)$.

Δ	Best known		Our ratio
3	1.17	[8]	1.42
4	1.32	[22]	1.50
5	1.45	[22]	1.55
6	1.56	[22]	1.60
7	1.65	[22]	1.64
8	1.74	[22]	1.67
9	1.81	[22]	1.69
10	1.87	[22]	1.71
11	1.93	[22]	1.73
12	1.98	[22]	1.75
13	2	[18]	1.76
20	2	[18]	1.83
50	2	[18]	1.93

Table 1: Approximation ratios for bipartite graphs

4. A 3/2 approximation algorithm for trees

In this section, we first present an $(1 + \frac{w_1^* - w_\Delta^*}{OPT})$-approximation algorithm for the MEC problem on trees. Then, combining this algorithm with Algorithm KK we derive a 3/2 approximation ratio.

For our first algorithm we consider the tree rooted in an arbitrary vertex and we denote by E^u the edges of the tree adjacent to a vertex u. The algorithm traverses the vertices of the tree in pre-order and for each vertex u assigns the edges in E^u to matchings as follows.

Algorithm TREES

1. Root the tree in an arbitrary vertex r;
2. For each vertex u in a pre-order traversal of the tree do
3. Let $\langle E^u \rangle = \langle e_1^u, e_2^u, \dots, e_{d(u)}^u \rangle$, and e_j^u, $1 \leq j \leq d(u)$, be the edge between $u, u \neq r$, and its parent;
4. For $i = 1, 2, \dots, d(u)$, $i \neq j$, do
5. Insert edge e_i^u into the first matching not containing other edge in E^u;

To analyze our algorithm we define y_i, $1 \leq i \leq \Delta$, to be the weight of the heaviest edge between those ranked i in each ordering $\langle E^u \rangle$, $u \in V$, i.e., $y_i = \max_{u \in V}\{w(e_i^u)\}$. It is clear that $y_1 \geq y_2 \geq \cdots \geq y_\Delta$. Next two propositions use these values for bounding the weights of the matchings of both an optimal solution and a solution found by Algorithm TREES. Recall that an optimal solution to the MEC problem consists of at least Δ matchings.

Proposition 6. *For all $1 \leq i \leq \Delta$, it holds that $w_i^* \geq y_i$.*

Proof. Let $e = (u, v)$ be the heaviest edge with rank equal to i, i.e., $y_i = w(e)$. W.l.o.g., assume that e is ranked i in E^u. Then, there exist i edges in E^u of weight at least y_i and as they belong into i different matchings in an optimal solution, it follows that $w_i^* \geq y_i$. □

Proposition 7. *Algorithm TREES constructs a solution of exactly Δ matchings. For the weight, w_i, of the i-th, $2 \leq i \leq \Delta$, matching it holds that $w_i \leq y_{i-1}$.*

Proof. For a vertex $u \neq r$ of the tree let e be the edge between u and its parent and j be its rank in E^u, i.e., $e = e_j^u$. In the iteration processing the vertex u the edge e has already been inserted by the algorithm into a matching, say M_p.

The algorithm inserts the edges in E^r into $d(r) \leq \Delta$ matchings. For any other vertex u, the algorithm inserts the edges in $E^u \setminus \{e\}$ into $d(u)-1 \leq \Delta - 1$ matchings different than M_p. Therefore, the algorithm finds a solution $\mathcal{M} = \{M_1, M_2, \ldots, M_\Delta\}$ of exactly Δ matchings.

We prove the bounds on the matching's weights by induction on the vertices in the order they are processed by the algorithm. We consider all matchings in $\mathcal{M}$ of an initial weight $w_i = 0$, $1 \leq i \leq \Delta$.

For the root vertex r, the algorithm inserts each edge e_i^r into matching M_i, $1 \leq i \leq d(r)$. Clearly, $w_i = w(e_i^r) \leq y_i \leq y_{i-1}$, $2 \leq i \leq \Delta$.

Assume that before the iteration processing a vertex $u \neq r$, it holds that $w_i \leq y_{i-1}$, $2 \leq i \leq \Delta$, and let w_i' be the weight of the matching M_i, $2 \leq i \leq \Delta$, after processing the vertex u. We prove that $w_i' \leq y_{i-1}$, $2 \leq i \leq \Delta$, by distinguishing among three cases depending on the values of p and j:

(i) $p = j$: Each edge e_i^u belongs to matching M_i, $1 \leq i \leq d(u)$. Since $w_i \leq y_{i-1}$ and $w(e_i^u) \leq y_i$, it follows that $w_i' = \max\{w_i, w(e_i^u)\} \leq \max\{y_{i-1}, y_i\} = y_{i-1}$, $2 \leq i \leq \Delta$.

(ii) $p > j$: For $1 \leq i \leq j-1$ and $p+1 \leq i \leq d(u)$ each edge e_i^u belongs to matching M_i and we conclude as in Case (i). For $j+1 \leq i \leq p$ each edge e_i^u belongs to matching M_{i-1}, that is $w_i' = \max\{w_i, w(e_{i+1}^u)\} \leq \max\{y_{i-1}, y_{i+1}\} = y_{i-1}$.

(iii) $p < j$: For $1 \leq i \leq p-1$ and $j+1 \leq i \leq d(u)$ each edge e_i^u belongs to matching M_i and we conclude as in Case (i). For $p \leq i \leq j-1$ each edge e_i^v belongs to matching M_{i+1}, that is $w_i' = \max\{w_i, w(e_{i-1}^u)\} \leq \max\{y_{i-1}, y_{i-1}\} = y_{i-1}$. □

Using the bounds established in Propositions 6 and 7 we obtain the next lemma.

Lemma 8. *Algorithm TREES achieves an approximation ratio of $1 + \frac{w_1^* - w_\Delta^*}{OPT} < 2$ for the* MEC *problem on trees.*

Proof. For the weight of the first matching obtained by Algorithm TREES it holds that $w_1 \leq y_1 = w_1^*$, since both y_1 and w_1^* are equal to the weight of the heaviest edge of the tree. By Proposition 7 it holds that $w_i \leq y_{i-1}$, $2 \leq i \leq \Delta$ and by Proposition 6 it holds that $y_i \leq w_i^*$, $1 \leq i \leq \Delta$. Therefore, the weight of the solution obtained by Algorithm TREES is $W = \sum_{i=1}^{\Delta} w_i \leq y_1 + \sum_{i=2}^{\Delta} y_{i-1} = y_1 + \sum_{i=1}^{\Delta-1} y_i \leq w_1^* + \sum_{i=1}^{\Delta-1} w_i^* \leq w_1^* + OPT - w_\Delta^*$, that is $\frac{W}{OPT} \leq 1 + \frac{w_1^* - w_\Delta^*}{OPT} < 2$. □

The example illustrated in Figure 2(a) shows that the ratio of our algorithm can be arbitrarily close to 2. For this instance $OPT = C + 2\epsilon$ (Figure 2(b)), the weight of the solution found by Algorithm TREES is $W = 2C + \epsilon$ (Figure 2(c)) and the approximation ratio becomes $\frac{2C+\epsilon}{C+2\epsilon}$.

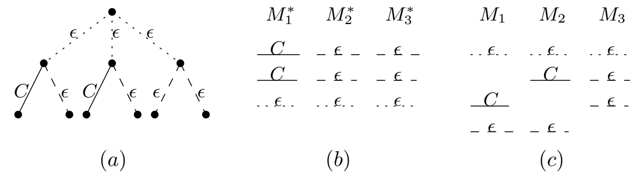

Figure 2: A tight example for the 2 approximation ratio of Algorithm TREES.

To derive the 3/2 approximation ratio we simply select the best among the solutions found by Algorithm KK and Algorithm TREES.

Theorem 9. *There is a $\frac{3}{2}$-approximation algorithm for the* MEC *problem on trees.*

Proof. Let W be the weight of the best among the solutions found by Algorithm KK and Algorithm TREES. By Lemma 2 it holds that $\frac{W}{OPT} \leq 2 - \frac{w_1^*}{OPT}$ and by Lemma 8 that $\frac{W}{OPT} \leq 1 + \frac{w_1^* - w_\Delta^*}{OPT}$. As the first bound is increasing and the second one is decreasing with respect to OPT, it follows that the ratio $\frac{W}{OPT}$ is maximized when $2 - \frac{w_1^*}{OPT} = 1 + \frac{w_1^* - w_\Delta^*}{OPT}$, that is $OPT = 2 \cdot w_1^* - w_\Delta^*$. Therefore, $\frac{W}{OPT} \leq 2 - \frac{w_1^*}{OPT} = 2 - \frac{w_1^*}{2 \cdot w_1^* - w_\Delta^*} \leq 2 - \frac{w_1^*}{2 \cdot w_1^*} = \frac{3}{2}$. □

For the tightness of the analysis in Theorem 9 consider the instance given in Figure 3(a). For this instance $OPT = 2C + 2\epsilon$ (Figure 3(b)) and the weights of the solutions found by Algorithm TREES and Algorithm KK are $3C$ (Figure 3(c)) and $3C - \epsilon$ (Figure 3(d)), respectively. Our algorithm selects the solution found by Algorithm KK and the approximation ratio becomes $\frac{3C-\epsilon}{2C+2\epsilon}$.

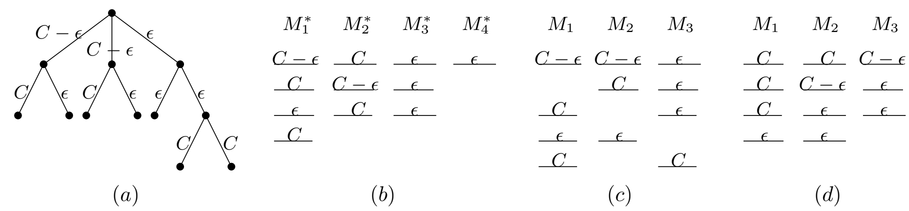

Figure 3: A tight example for the 3/2-approximation algorithm for trees.

5. Moderately exponential approximation algorithms for trees

In this section, we present two approximation algorithms for trees that improve the 3/2 ratio of Theorem 9 within exponential running time much better than that needed for the computation of an optimal solution.

The idea employed by the algorithms is to find an approximate solution to the MEC problem on a tree $T = (V, E)$ by searching exhaustively for the weights of a number of matchings of an optimal solution $\mathcal{M}^*$. A parameter z, given as input to the algorithms, determines the (maximum) number of matchings of $\mathcal{M}^*$ that we search exhaustively and, hence, the complexity and the approximation ratio of the algorithms.

In such an exhaustive search, each step of the proposed algorithms has to answer to the following decision problem:

Feasible-MEC (F-MEC)
Instance: A weighted graph $G = (V, E)$ and a sequence of k weights, $w_1 \geq w_2 \geq \cdots \geq w_k$.
Question: Is there a feasible solution $M = \{M_1, M_2, \ldots, M_k\}$ to the MEC problem on G such that $\max_{e \in M_i} w(e) \leq w_i$, $1 \leq i \leq k$?

The F-MEC problem is equivalent to the next well known variant of the edge-coloring problem:

List Edge-Coloring (LEC)
Instance: A graph $G = (V, E)$, a set of colors $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ and a list of colors $\phi(e) \subseteq \mathcal{C}$ for each $e \in E$.
Question: Is there a k-coloring of G such that each edge e is assigned a color in its list $\phi(e)$?

Indeed, an instance of the F-MEC problem on a graph G, and given weights $w_1 \geq w_2 \geq \cdots \geq w_k$, can be easily transformed to the next equivalent instance of the LEC problem: is there a k-coloring of G where each edge $e \in E$ is assigned a color in $\phi(e) = \{C_i : w_i \geq w(e),\ 1 \leq i \leq k\}$? A "yes" answer to this instance of the LEC problem corresponds to the existence of a feasible solution $M = \{M_1, M_2, \ldots, M_k\}$ for the MEC problem of weight $W = \sum_{i=1}^{k} w_i$.

It is known that the LEC problem can be answered in $O(m \cdot \Delta^{3.5})$ time for trees [9], but it becomes NP-complete for bipartite graphs [20]. Therefore, this approach can be used for trees but cannot be extended to bipartite graphs.

The first algorithm proposed in Section 5.1 is exponential to the maximum degree, Δ, of the input tree and achieves a ρ approximation ratio in $O^*(m^{f(\rho)\cdot\Delta})$ time, where $f(\rho) = \frac{9-\rho}{4\rho}$. The second algorithm presented in Section 5.2 is exponential to the number of edges, m, of the input tree and achieves a ratio of ρ in $O^*(g(\rho)^m)$ time, where $g(\rho) = \frac{(2\rho-1)^2+1}{(2\rho-1)^{2(2\rho-1)^2/((2\rho-1)^2+1)}}$. Some values of $\rho \leq 3/2$, $f(\rho)$ and $g(\rho)$ are summarized in Table 2.

Complexity	ρ	OPT	1.1	1.2	1.3	1.4	1.5
$O^*(m^{f(\rho)\cdot\Delta})$	$f(\rho)$	2	1.795	1.625	1.481	1.357	1.250
$O^*(g(\rho)^m)$	$g(\rho)$	2	1.968	1.896	1.811	1.727	1.649

Table 2: Approximation ratios vs. complexities for trees

5.1. An exponential to Δ algorithm

This algorithm depends on a parameter z taking integer values in $[1, 2\Delta-1]$ and iterates z times, for $j = 1, 2, \ldots z$. In each iteration the algorithm considers all the combinations of j edge weights as the weights of the j heaviest matchings of an optimal solution. For each combination of weights, $w_1 \geq w_2 \geq \cdots \geq w_j$, the algorithm has to answer to an instance of the F-MEC problem on the input tree T. In order a "yes" answer to this F-MEC problem to be probable for all values of j we extend the combination of weights $w_1 \geq w_2 \geq \cdots \geq w_j$ to a sequence $w_1 \geq w_2 \geq \cdots \geq w_j = w_{j+1} = w_{j+2} = \cdots = w_k$ by adding $k-j$ new weights all equal to w_j. In fact, this extended sequence consists of $k = j-1+\Delta$ weights, if $j \leq \Delta$ (this way the T's edges of weights $w(e) \leq w_j$ can be assigned into the Δ matchings of weight w_j) and $k = 2\Delta-1$, otherwise (since by Proposition 1 any solution to the MEC problem consists of at most $2\Delta-1$ matchings). Hence, $k = \min\{j-1+\Delta, 2\Delta-1\}$. This instance of the F-MEC problem has answer "yes" if and only if the edges of weight $w(e) > w_j$ can be assigned to (matchings of) weights greater than w_j (see the proof of Theorem 10). In this case the algorithm finds a feasible solution for the MEC problem and it returns the best among all feasible solutions found.

Algorithm TREES-$\Delta(z)$

```
1. For j = 1, 2, ..., z do
2.   For each combination of j edge weights, w_1 ≥ w_2 ≥ ··· ≥ w_j, do
3.     Answer to the F-MEC problem with input T and
       k = min{j − 1 + Δ, 2Δ − 1} weights:
       w_1 ≥ w_2 ≥ ··· ≥ w_j = w_{j+1} = w_{j+2} = ··· = w_k;
4.     If the answer is "yes"
       then a feasible solution to the MEC problem is found;
5. Return the best among the feasible solutions found;
```

Theorem 10. *For any $\rho \geq 1$, Algorithm TREES-$\Delta(z)$ achieves a ρ approximation ratio for the* MEC *problem on trees, in polynomial space and running time $O^*(m^{f(\rho)\Delta})$, where $f(\rho) = \frac{9-\rho}{4\rho}$.*

Proof. Consider the j-th iteration of the algorithm and in this iteration the combination of j edge weights which coincide with the weights, $w_1^* \geq w_2^* \geq \cdots \geq w_j^*$, of the j heaviest matchings of an optimal solution $\mathcal{M}^*$. In this step the algorithm answers to the instance of the F-MEC problem with input T and weights $w_i \geq w_i^*$, $1 \leq i \leq k$. We claim that this F-MEC problem has always a "yes" answer. Indeed, if $k = 2\Delta + 1$, then the claim follows since $k^* \leq 2\Delta - 1$ and $w_i \geq w_i^*$, $1 \leq i \leq k^*$. If $k = j - 1 + \Delta < 2\Delta - 1$, then the edges of weights $w(e) > w_j^*$ can be assigned (belong) to the $j - 1$ heaviest weights (matchings of $\mathcal{M}^*$). Moreover, there are Δ weights equal to w_j^* and the edges of weights $w(e) \leq w_j^*$ can be assigned to them. Hence, a feasible solution for the MEC problem on T is found of weight

$$W_j = w_1^* + w_2^* + \cdots + w_{j-1}^* + (k - j + 1) \cdot w_j^*.$$

The algorithm finds such a feasible solution in each iteration j and as it returns the best among them we obtain Δ bounds on the weight of this best solution, that is $W \leq w_1^* + w_2^* + \cdots + w_{j-1}^* + (k - j + 1) \cdot w_j^*$, $1 \leq j \leq z$. Proceeding as in the proof of Theorem 5 we find z multipliers

$$x_j = \begin{cases} \dfrac{2\Delta - 1 - z}{\Delta^2} \left(\dfrac{\Delta - 1}{\Delta}\right)^{\Delta - 1 - j}, & \text{if} \quad 1 \leq j \leq \Delta \\ \dfrac{2\Delta - 1 - z}{(2\Delta - j)(2\Delta - 1 - j)}, & \text{if} \quad \Delta + 1 \leq j \leq z \end{cases}$$

such that $\dfrac{W}{OPT} \leq \dfrac{1}{\sum_{i=1}^{z} x_i} = \dfrac{1}{1 - \frac{2\Delta - 1 - z}{\Delta} \cdot (\frac{\Delta - 1}{\Delta})^{\Delta - 1}}$.

The MEC problem is polynomial for graphs of $\Delta = 2$ and as for $\Delta \geq 3$ it holds that $(\frac{\Delta-1}{\Delta})^{\Delta-1} > \frac{4}{9}$ we get $\dfrac{W}{OPT} \leq \dfrac{1}{1 - \frac{4}{9} \cdot \frac{2\Delta - 1 - z}{\Delta}} = \rho$. Hence, an approximation ratio ρ is derived for $z = \frac{9-\rho}{4\rho}\Delta - 1 = f(\rho)\Delta - 1$, where $f(\rho) = \frac{9-\rho}{4\rho}$.

The complexity of Algorithm TREES-$\Delta(z)$ is exponential in z. In Line 2 the algorithm examines $\binom{m}{j}$ combinations of weights. Thus, for all iterations $\sum_{j=1}^{z} \binom{m}{j} = O(z \cdot m^z)$ combinations of weights are examined. For each one of these combinations, it takes $O(m \cdot \Delta^{3.5})$ time to answer to the instance of the F-MEC in Line 3. Since z and Δ are $O(m)$, the complexity of Algorithm TREES-$\Delta(z)$ is $O^*(m^z)$, that is $O^*(m^{f(\rho)\Delta})$. Moreover, the algorithm needs polynomial space, since Line 3 is executed independently for each combination of weights. □

Notice that for $z = 2\Delta - 1$ the Algorithm TREES-$\Delta(z)$ finds an optimal solution within $O^*(m^{2\Delta})$ time.

5.2. An exponential to m algorithm

This algorithm depends on a parameter z taking integer values in $[1, \lfloor \frac{m}{2} \rfloor]$ and iterates $2z$ times, for $k = 1, 2, \ldots, z, m-z, \ldots, m$. In each iteration, the algorithm exhaustively considers k edge weights, $w_1, w_2, \ldots, w_k$, as the weights of the k heaviest matchings of an optimal solution $\mathcal{M}^*$, and answers to the instance of the F-MEC problem, with input T and $w_1 \geq w_2 \geq \cdots \geq w_k$. This way an optimal solution is found when $k^* \leq z$ or $k^* \geq m - z$. In order to derive an approximate solution when $z < k^* < m - z$, the algorithm, in the iteration where $k = z$, answers also to instances of the F-MEC problem with input T and weights $w_1 \geq w_2 \geq \cdots \geq w_z = w_{z+1} = \cdots = w_{k'}$, for $k' = z+1, z+2, \ldots, m-z-1$. The algorithm returns the best among the feasible solutions found.

Algorithm TREES-$E(z)$

1. For $k = 1, 2, \ldots, z, m-z, \ldots, m$ do
2.   For each combination of k edge weights, $w_1 \geq w_2 \geq \cdots \geq w_k$, do
3.   Answer to the F-MEC with input T and $w_1 \geq w_2 \geq \cdots \geq w_k$;
4.   If the answer is "yes"
    then a feasible solution to the MEC problem is found;
5.   If $k = z$ then
6.   For $k' = z+1, z+2, \ldots, m-z-1$ do
7.   Answer to the F-MEC with input T and k' weights:
        $w_1 \geq w_2 \geq \cdots \geq w_z = w_{z+1} = w_{z+2} = \cdots = w_{k'}$;
8.   If the answer is "yes"
        then a feasible solution to the MEC problem is found;
9. Return the best among the feasible solutions found;

Theorem 11. *For any $\rho \geq 1$, Algorithm TREES-E(z) achieves a ρ approximation ratio for the* MEC *problem on trees, in polynomial space and running time $O^*(g(\rho)^m)$, where* $g(\rho) = \dfrac{(2\rho-1)^2+1}{(2\rho-1)^{2(2\rho-1)^2/((2\rho-1)^2+1)}}$.

Proof. If $k^* \leq z$ or $k^* \geq m - z$ then the algorithm in an iteration of Lines 2-4 finds an optimal solution.

If $z < k^* < m - z$ then we consider the following two solutions found by the algorithm:

(i) In the iteration where $k = m - z$, for a combination $w_1 \geq w_2 \geq \cdots \geq w_k$ of weights, it holds that $w_i = w_i^*$, $1 \leq i \leq k^*$. Hence, for this combination there is a feasible solution of weight at most $w_1^* + w_2^* + \cdots + w_{k^*}^* + (m - z - k^*)w_{k^*}^* = OPT + (m - z - k^*)w_{k^*}^*$.

(ii) In the iteration where $k = z$ and $k' = k^*$, for a combination $w_1 \geq w_2 \geq \cdots \geq w_k$ of weights, it holds that $w_i = w_i^*$, $1 \leq i \leq z$. Hence, for this combination there is a feasible solution of weight at most $w_1^* + w_2^* + \cdots + w_z^* + (k^* - z)w_z^* = OPT - \sum_{i=z+1}^{k^*} w_i^* + (k^* - z)w_z^*$.

Thus, it holds that

$$
\begin{aligned}
\frac{W}{OPT} &\leq \min\left\{\frac{OPT+(m-z-k^*)w^*_{k^*}}{OPT}, \frac{OPT-\sum_{i=z+1}^{k^*} w^*_i+(k^*-z)w^*_z}{OPT}\right\} \\
&\leq \min\left\{1+\frac{(m-z-k^*)w^*_{k^*}}{zw^*_z+(k^*-z)w^*_{k^*}}, 1+\frac{(k^*-z)(w^*_z-w^*_{k^*})}{zw^*_z+(k^*-z)w^*_{k^*}}\right\}.
\end{aligned}
$$

As the first value is increasing with $w^*_{k^*}$ and the second one is decreasing, this quantity is maximized when $(k^*-z)w^*_z=(m-2z)w^*_{k^*}$. Therefore, we have

$$
\frac{W}{OPT} \leq 1+\frac{(m-z-k^*)w^*_{k^*}}{\frac{z(m-2z)}{k^*-z}w^*_{k^*}+(k^*-z)w^*_{k^*}} = \frac{k^*(m-2z)}{z(m-2z)+(k^*-z)^2},
$$

which is maximized for $k^*=\sqrt{z(m-z)}$. Hence,

$$
\frac{W}{OPT} \leq \frac{\sqrt{z(m-z)}(m-2z)}{z(m-2z)+(\sqrt{z(m-z)}-z)^2} = \frac{m-2z}{2\sqrt{z(m-z)}-2z}.
$$

By setting $z=\lambda m$, where $0<\lambda\leq\frac{1}{2}$, we get

$$
\frac{W}{OPT} \leq \frac{m-2\lambda m}{2\sqrt{\lambda m(m-\lambda m)}-2\lambda m} = \frac{1-2\lambda}{2\sqrt{\lambda(1-\lambda)}-2\lambda} = \rho.
$$

Therefore, in order to achieve a ρ approximation ratio we choose $\lambda=\frac{1}{(2\rho-1)^2+1}$, that is $z=\frac{m}{(2\rho-1)^2+1}$.

The algorithm needs polynomial space, since Lines 3-8 are executed independently for each combination of weights. As the F-MEC problem is polynomial for trees, the complexity of the algorithm is, within a polynomial factor, $O(T(m))$, where $T(m)$ is the number of combinations generated. For this number it holds that

$$
T(m) \leq \sum_{i=1}^{z}\binom{m}{i}+\sum_{i=m-z}^{m}\binom{m}{i} = 2\sum_{i=1}^{z}\binom{m}{i} \leq 2z\binom{m}{z} \leq m\binom{m}{\lambda m}
$$

$$
\leq m\left(\left(\frac{1}{\lambda}\right)^{\lambda}\left(\frac{1}{1-\lambda}\right)^{1-\lambda}\right)^m = m\left(\frac{(2\rho-1)^2+1}{(2\rho-1)^{2(2\rho-1)^2/((2\rho-1)^2+1)}}\right)^m = m\cdot g(\rho)^m.
$$

Hence, the complexity of Algorithm TREES-$E(z)$ becomes $O^*(g(\rho)^m)$, where $g(\rho)=\frac{(2\rho-1)^2+1}{(2\rho-1)^{2(2\rho-1)^2/((2\rho-1)^2+1)}}$. □

Note that for $z=\lfloor\frac{m}{2}\rfloor$ Algorithm TREES-$E(z)$ computes an optimal solution for the MEC problem on trees in $O^*(2^m)$ time and polynomial space.

In [2], an algorithm has been presented with running time and space $O^*(2^n)$, which, for any k, computes the number of all proper k-vertex-colorings of a graph, and moreover enumerates these colorings. This algorithm can be used to find an optimal solution for the MVC problem on a general graph, by running it for $1 \leq k \leq n$. Considering the line graph $L(G)$ of the input graph G of the MEC problem, we derive that the MEC problem on general graphs can be optimally solved with running time and space $O^*(2^m)$.

Next proposition shows that if $\Delta = o(m)$ then the running time of Algorithm TREES-$E(z)$ for computing an optimal solution is improved.

Proposition 12. *If $\Delta = o(m)$, then Algorithm TREES-E(z) requires subexponential running time $2^{o(m)}$ in order to compute an exact solution for trees.*

Proof. By Proposition 1, the number k^* of matchings in any optimal solution to the MEC problem is at most $2\Delta - 1$. Thus, the number of combinations of weights needed to be generated by the algorithm becomes

$$\begin{aligned} T(m) &\leq \binom{m}{2\Delta} \leq \frac{m^m}{(2\Delta)^{2\Delta}(m-2\Delta)^{m-2\Delta}} \\ &\leq 2^{m \log m - 2\Delta \log(2\Delta) - (m-2\Delta)\log(m-2\Delta)} \\ &\leq 2^{m \log(1+2\Delta/(m-2\Delta)) + 2\Delta \log(m/2\Delta - 1)} \end{aligned}$$

Notice first that $2\Delta/(m - 2\Delta)$ tends to 0 for $m \to \infty$, since $\Delta = o(m)$, and thus $m \log\left(1 + \frac{2\Delta}{(m-2\Delta)}\right) \to 0$. Moreover, note that $2\Delta \log\left(\frac{m}{2\Delta} - 1\right) = o(m)$, since $\frac{2\Delta \log\left(\frac{m}{2\Delta} - 1\right)}{m}$ tends to 0 as m increases. Combining the two observations above, we get that $T(m) = 2^{o(m)}$ and, hence, the running time of Algorithm TREES-$E(\frac{m}{2})$ is $O^*(2^{o(m)})$. □

Notice that Algorithm TREES-$E(\lfloor \frac{m}{2} \rfloor)$ and Algorithm TREES-$\Delta(2\Delta - 1)$ coincide and both return an optimal solution to the MEC problem on trees. Thus the last proposition holds for both algorithms.

6. Bi-valued graphs

In this section we show first that the MEC problem is NP-complete for complete graphs with bi-valued edge weights. Recall that the MEC problem is polynomial for bi-valued bipartite graphs [10], while for general bi-valued graphs it generalizes the classical edge-coloring problem, which is known to be NP-complete even for cubic graphs [17]. In the next theorem we give a reduction from this latter problem.

Theorem 13. *The* MEC *problem is NP-complete for complete graphs even with edge weights $w(e) \in \{1, 2\}$.*

Proof. The edge-coloring problem for cubic graphs takes as input a graph $G = (V, E)$, $|V| = n$, with $d(u) = 3$, for each $u \in V$, and asks for the existence of a proper 3-coloring of G. Notice that any cubic graph has an even number, n, of vertices.

From such an instance we construct a complete weighted graph K_n with edge weights $w(e) = 2$, for each $e \in E$, and $w(e) = 1$, otherwise, and we show that there is a 3-coloring of G iff there is a solution $\mathcal{M}$ for the MEC problem on K_n of weight at most $n + 2$.

Assume, first, that there is a 3-coloring of G. Then, there are three matchings of K_n each one of weight equal to 2, which include all the edges of K_n of weight 2. Let $K_n - G$ be the graph induced by the remaining edges of K_n (those of weight 1). The graph $K_n - G$ is $(n-4)$-colorable as a $(n-4)$-regular graph of even order [5]. Therefore, there is a solution $\mathcal{M}$ for the MEC problem on K_n of weight at most $3 \cdot 2 + (n-4) \cdot 1 = n + 2$.

Conversely, consider that there is a solution $\mathcal{M}$ to the MEC problem on K_n of weight at most $n + 2$. This solution contains $k \geq n - 1$ matchings, since a complete graph of even order can be colored with at least $n - 1$ colors [14]. Moreover, $\mathcal{M}$ contains at least three matchings of weight equal to 2, since, by its construction, K_n has exactly three edges of weight 2 adjacent to each vertex. Assume that there is a fourth matching in $\mathcal{M}$ of weight equal to 2. Then, $\mathcal{M}$ will be of weight at least $4 \cdot 2 + (k-4) \cdot 1 \geq n + 3$, a contradiction. Therefore, $\mathcal{M}$ contains exactly 3 matchings of weight equal to 2, which imply a 3-coloring for G. □

Theorem 13 implies that the MEC problem is NP-complete in all superclasses of complete graphs, including split and interval graphs. Note also that the complexity of the classical edge-coloring problem on interval graphs of even maximum degree remains an open question [3].

In what follows, we present an approximation algorithm for general graphs with two different edge weights. Assume that the edges of the graph $G = (V, E)$ have weights either 1 or $t \geq 2$. Let $G[E_1]$, of maximum degree Δ_1, and $G[E_t]$, of maximum degree Δ_t, be the subgraphs of G induced by its edges of weights 1 and t, respectively.

Algorithm BI-VALUED
1. `Find a` $(\Delta + 1)$`-coloring solution for` G`;`
2. `Find a` $(\Delta_1 + 1)$`-coloring solution for` G_1`,`
   `a` $(\Delta_t + 1)$`-coloring solution for` G_t `and concatenate them;`
3. `Return the best among the two solutions found;`

Theorem 14. *Algorithm BI-VALUED achieves a* $\frac{4}{3}$*-approximation ratio for the* MEC *problem on general graphs of arbitrarily large* Δ *and edge weights* $w(e) \in \{1, t\}$.

Proof. An optimal solution contains at least Δ matchings and at least Δ_t of them are of weight equal to t. Therefore, a lower bound to the weight of an optimal solution is $OPT \geq \Delta_t \cdot t + (\Delta - \Delta_t)$.

A $(\Delta + 1)$-coloring of G in Line 1 of the algorithm yields a solution for the MEC problem of weight $W \leq (\Delta + 1) \cdot t$, while a $(\Delta_1 + 1)$-coloring of $G[E_1]$ and a $(\Delta_t + 1)$-coloring of $G[E_t]$ in Line 2 yield another solution of weight

$W \leq (\Delta_t + 1) \cdot t + (\Delta_1 + 1) \cdot 1 \leq (\Delta_t + 1) \cdot t + (\Delta + 1)$. By multiplying both sides of the first inequality with $\frac{\Delta_t^2 + 2\Delta_t - \Delta}{(\Delta+1)^2}$, both sides of the second one with $\frac{\Delta - \Delta_t}{\Delta+1}$ and adding them, we get $\frac{\Delta^2 + \Delta_t^2 - \Delta \cdot \Delta_t + \Delta_t}{(\Delta+1)^2} \cdot W \leq \Delta_t \cdot t + (\Delta - \Delta_t) \leq OPT$, that is $\frac{W}{OPT} \leq \frac{(\Delta+1)^2}{(\Delta-\Delta_t)^2 + \Delta_t(\Delta+1)}$. This ratio is maximized when $\Delta_t = \frac{\Delta-1}{2}$, and therefore $\frac{W}{OPT} \leq \frac{4(\Delta+1)}{(\Delta+1)+2(\Delta-1)} = \frac{4\Delta+4}{3\Delta-1} = \frac{4}{3} + \frac{16}{9\Delta-3}$. □

7. Concluding remarks

We presented approximation algorithms for the MEC problem on several classes of the underlying graph including general and bipartite graphs, trees and bi-valued graphs. Recall that the MEC problem is known to be approximable within a factor of 2 (for any class of graphs) and inapproximable within a factor less than 7/6 (even for bipartite graphs), while its complexity for trees remains open. The ratios achieved by our algorithms narrow the gaps in the approximability of the problem.

For bipartite graphs we derived an approximation ratio less than 2 which, however, tends asymptotically to 2 as the maximum degree of the graph increases. In recompense, this ratio increases much slower compared to ratios achieved by former algorithms. For trees we presented a 3/2-polynomial approximation algorithm that is the first below-to-ratio 2 algorithm for the MEC problem for a natural class of graphs. Moreover, we have devised moderately exponential algorithms for trees that further improve ratio 3/2. Finally, for general bi-valued graphs, we presented an asymptotic 4/3-approximation algorithm.

However, the gaps in the approximability of the MEC problem remain still wide and their further narrowing is a subject of further research.

References

[1] F. N. Afrati, T. Aslanidis, E. Bampis, and I. Milis. Scheduling in switching networks with set-up delays. *Journal of Combinatorial Optimization*, 9:49–57, 2005.

[2] A. Björklund and Th. Husfeldt. Inclusion–exclusion algorithms for counting set partitions. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 575–582. IEEE Computer Society, 2006.

[3] V. A. Bojarshinov. Edge and total coloring of interval graphs. *Discrete Applied Mathematics*, 114:23–28, 2001.

[4] N. Bourgeois, G. Lucarelli, I. Milis, and V. Th. Paschos. Approximating the max edge-coloring problem. In *20th International Workshop on Combinatorial Algorithms (IWOCA'09)*, to appear.

[5] A. G. Chetwynd and A. J. W. Hilton. Regular graphs of high degree are 1-factorizable. In *Proceedings of the London Mathematical Society*, volume 50, pages 193–206, 1985.

[6] R. Cole, K. Ost, and S. Schirra. Edge-coloring bipartite multigraphs in $O(E \log D)$ time. *Combinatorica*, 21:5–12, 2001.

[7] P. Crescenzi, X. Deng, and Ch. H. Papadimitriou. On approximating a scheduling problem. *Journal of Combinatorial Optimization*, 5:287–297, 2001.

[8] 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).

[9] 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.

[10] 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).

[11] 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.

[12] 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).

[13] G. Finke, V. Jost, M. Queyranne, and A. Sebő. Batch processing with interval graph compatibilities between tasks. *Discrete Applied Mathematics*, 156:556–568, 2008.

[14] S. Fiorini and R. J. Wilson. *Edge-Colourings of Graphs*. Pitman, London, 1977.

[15] I. S. Gopal and C. Wong. Minimizing the number of switchings in a SS/TDMA system. *IEEE Transactions On Communications*, 33:497–501, 1985.

[16] 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.

[17] I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10:718–720, 1981.

[18] A. Kesselman and K. Kogan. Nonpreemptive scheduling of optical switches. *IEEE Transactions on Communications*, 55:1212–1219, 2007. (also in Proc. GLOBECOM'04).

[19] D. König. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen*, 77:453–465, 1916.

[20] M. Kubale. Some results concerning the complexity of restricted colorings of graphs. *Discrete Applied Mathematics*, 36:35–46, 1992.

[21] G. Lucarelli, I. Milis, and V. Th. Paschos. On the maximum edge coloring problem. In *6th Workshop on Approximation and Online Algorithms (WAOA'08)*, volume 5426 of *LNCS*, pages 279–292. Springer, 2009.

[22] G. Lucarelli, I. Milis, and V. Th. Paschos. On the max-weight edge coloring problem. *Journal of Combinatorial Optimization*, In Press.

[23] S. Micali and V. V. Vazirani. An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In *21st Annual IEEE Symposium on Foundations of Computer Science (FOCS'80)*, pages 17–27. IEEE Computer Society, 1980.

[24] 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.

[25] 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.

[26] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz.*, 3:25–30, 1964.