On the Maximum Edge Coloring Problem
(Extended Abstract)

Giorgio Lucarelli[1,⋆], Ioannis Milis[1], and Vangelis Th. Paschos[2,⋆⋆]

[1] Dept. of Informatics, Athens University of Economics and Business, Greece
{gluc,milis}@aueb.gr
[2] LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine, France
paschos@lamsade.dauphine.fr

Abstract. We study the following generalization of the classical edge coloring problem: Given a weighted graph, find a partition of its edges into matchings (colors), each one of weight equal to the maximum weight of its edges, so that the total weight of the partition is minimized. We present new approximation algorithms 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).

1 Introduction

In the classical *edge coloring* problem we ask for the minimum number of colors required in order to assign different colors to adjacent edges of a graph $G = (V, E)$. Equivalently, we ask for a partition $S = \{M_1, M_2, \ldots, M_s\}$ of the edge set of G into matchings (color classes) such that s is minimized. This minimum number of matchings (colors) is known as the *chromatic index* of the graph and it is denoted by $\chi'(G)$.

In several applications, the following generalization of the classical edge coloring problem arises: a positive integer weight is associated with each edge of G and we now ask for a partition $S = \{M_1, M_2, \ldots, M_s\}$ of the edges of G into matchings (colors), each one of weight $w_i = \max\{w(e)|e \in M_i\}$, such that their total weight $W = \sum_{i=1}^{s} w_i$ is minimized. As the weight w_i of each matching is defined to be the maximum weight of the edges colored i, we refer to this problem as Maximum Edge Coloring (MEC) problem.

⋆ 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.

⋆⋆ Part of this work has been carried out while the author was with the Department of Informatics of Athens University of Economics and Business.

E. Bampis and M. Skutella (Eds.): WAOA 2008, LNCS 5426, pp. 279–292, 2009.

The most common application for the MEC problem arises in the domain of communication systems and, especially, in single hop systems. In such systems messages are to be transmitted directly from senders to receivers through direct connections established by an underlying switching network. Any node of such a system cannot participate in more than one transmissions at a time, while the transmission of messages between several pairs of nodes can take place simultaneously. The scheduler of such a system 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 the above situation corresponds directly to the MEC problem.

In practical applications there exists a non negligible setup delay to establish each configuration (matching). The presence of such a delay, say d, 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 matching in S will be also increased by d, incorporating its set-up delay. A natural idea to decrease the weight of a solution to such a problem is to allow preemption i.e., interrupt the transmission of a (set of) message(s) in a configuration and complete it later. However, in this preemptive-MEC problem the presence of the set-up delay d plays a crucial role in the problem's complexity [11,4,1].

The analogous to the MEC problem generalization for the classical vertex coloring problem, called Maximum (vertex) Coloring (MVC), has been also studied in the literature during last years [2,9,7,5,8,19,18]. In the MVC problem we ask for a partition of the vertices of G into independent sets (colors), each one of weight equal to the maximum weight of its vertices, so that the total weight of the partition is minimized. Like the classical edge and vertex coloring problems, the MEC problem, on a general graph G, is equivalent to the MVC problem on the line graph, $L(G)$, of G. However, this is not true for every special graph class, since most of them are not closed under line graph transformation (e.g. complete graphs, trees and bipartite graphs).

Related Work. It is known that the MEC problem is strongly NP-hard even for (i) complete balanced bipartite graphs [20], (ii) bipartite graphs of maximum degree three and edge weights $w(e) \in \{1, 2, 3\}$ [11,13], (iii) cubic bipartite graphs [7] and (iv) cubic planar bipartite graphs with edge weights $w(e) \in \{1, 2, 3\}$ [5]. Moreover, in conjunction with the results (iii) and (iv) above, it has been shown that the MEC problem on k-regular bipartite graphs cannot be approximated within a ratio less than $\frac{2^k}{2^k-1}$, which for $k = 3$ becomes 8/7 [7]. This inapproximability result has been improved to 7/6 for cubic planar bipartite graphs [5].

Concerning the approximability of the MEC problem, a natural greedy 2-approximation algorithm has been proposed by Kesselman and Kogan [13] for general graphs. A $\frac{2\Delta-1}{3}$-approximation algorithm, for bipartite graphs of maximum degree Δ, has been presented in [7], which gives an approximation ratio of 5/3 for $\Delta = 3$. Especially for bipartite graphs of $\Delta = 3$, an algorithm that

Table 1. Known approximation ratios for bipartite graphs in [8] and [16] vs. those in this paper

Δ	[8]	[16]	This paper
3	1.42	1.17	1.42
4	1.61	1.32	1.54
5	1.75	1.45	1.62
6	1.86	1.56	1.68
7	1.95	1.65	1.72
8	> 2	1.74	1.76
9	> 2	1.81	1.78
10	> 2	1.87	1.80
11	> 2	1.93	1.82
12	> 2	1.98	1.84
13	> 2	> 2	1.85
20	> 2	> 2	1.90
50	> 2	> 2	1.96

attains the 7/6 inapproximability bound has been presented in [5]. For general bipartite graphs of $\Delta \leq 12$ have been also presented algorithms that achieve approximation ratios $\rho < 2$. In fact, an algorithm presented in [8] achieves such a ratio for $4 \leq \Delta \leq 7$, while another one presented in [16] achieves the best known ratios for maximum degrees between $4 \leq \Delta \leq 12$ (see the 2nd and 3rd columns of Table 1). However, for bipartite graphs of $\Delta > 12$ the best known ratio is achieved by the 2-approximation algorithm in [13] for general graphs.

On the other hand, the MEC problem is known to be polynomial for a few very special cases including complete balanced bipartite graphs and edge weights $w(e) \in \{1, 2\}$ [20], general bipartite graphs and edge weights $w(e) \in \{1, 2\}$ [7], chains [8] (in fact, this algorithm can be also applied for graphs of $\Delta = 2$), stars of chains and bounded degree trees [16]. It is interesting that the complexity of the MEC problem on trees remains open.

Our results and organization of the paper. In this paper we further explore the complexity and approximabilty of the MEC problem with respect to the class of the underlying graph. Especially, we present new approximation results for several variants of the problem exploiting the general idea of producing more than one solutions for the problem and choosing the best of them.

The next section starts with our notation and a remark on the known greedy 2-approximation algorithm [13]. Then, combining this remark with a simple idea, we present a first algorithm for general and bipartite graphs. For bipartite graphs, this algorithm achieves better approximation ratios than the algorithms in [8] (for $\Delta \geq 4$) and [16] (for $\Delta \geq 9$) which ratios, in addition, tend asymptotically to 2 as Δ increases.

In Section 2 we present a new algorithm for the MEC problem on bipartite graphs which, like algorithms in [8] and [16], produces Δ different solutions and chooses the best of them. Our algorithm derives the best known ratios for bipartite graphs for any $\Delta \geq 9$, that remain always strictly smaller than 2 (see the 4th column of Table 1).

Section 4 deals with the MEC problems on trees. An exact algorithm for this case of complexity $O(|E|^{2\Delta+O(1)})$ has been proposed in [16]. In this section we present a generic algorithm for trees depending on a parameter k, which determines both the complexity of the algorithm and the quality of the solution found. In fact, the complexity of the algorithm is $O(|E|^{k+O(1)})$ and it produces an optimal solution, if $k = 2\Delta - 1$, an $e/(e-1)$-approximate solution, if $k = \Delta$, and a ρ-approximate solution, with $\rho < 2$, if $2 \leq k \leq \Delta$.

Finally, in Section 5 we prove that the MEC problem is NP-complete even in complete graphs with bi-valued edge weights, and we give an asymptotic $\frac{4}{3}$-approximation algorithm for general graphs of arbitrarily large Δ and bi-valued edge weights.

2 Notation and Preliminaries

We consider the MEC problem on a weighted graph $G = (V, E)$. By $d_G(v), v \in V$ (or simply $d(v)$), we denote the degree of vertex v and by $\Delta(G)$ (or simply Δ) the maximum degree of G. We consider, also, the edges of G sorted in non-increasing order of their weights with e_1 denoting the heaviest edge of G, that is $w(e_1) \geq w(e_2) \geq \ldots \geq w(e_m)$. By $S^* = \{M_1^*, M_2^*, \ldots, M_{s^*}^*\}$ we denote an optimal solution to the MEC problem of weight $OPT = w_1^* + w_2^* + \ldots + w_{s^*}^*$. We call a solution $S = \{M_1, M_2, \ldots, M_s\}$ to the MEC problem *nice* if $w_1 \geq w_2 \geq \ldots \geq w_s$ and each matching M_i is maximal in the subgraph induced by the edges $E \setminus \bigcup_{j=1}^{i-1} M_j$. In the following we consider any (suboptimal or optimal) solution to the MEC problem to be nice. This is due to the next proposition (see also [16]).

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, s, in such a solution it holds that* $\Delta \leq s \leq 2\Delta - 1$.

The most interesting and general result for the MEC problem is due to Kesselman and Kogan [13] who proposed the following greedy algorithm:

Algorithm 1
```
1. Sort the edges of G in non-increasing order of their weights;
2. Using this order:
  - Insert each edge into the first matching that fits;
  - If such a matching does not exist then compute a new matching;
```

It is proved in [13] that Algorithm 1 obtains a solution, S, of total weight $W \leq 2OPT$ and they also presented a $2 - \frac{1}{\Delta}$ tightness example. We prove here that the approximation ratio of this algorithm matches exactly its lower bound on the given tightness example. In fact, the solution S is, by its construction, a nice one. Using the bound on s in Proposition 1, the bound on W can be slightly improved as in next lemma.

Lemma 1. *The total weight of the solution obtained by Algorithm 1 is* $W \leq 2\sum_{i=1}^{\Delta} w_i^* - w_1^* \leq 2OPT - w_1^*$.

Proof. (Sketch) Let e be the first edge inserted into matching M_i, i.e. $w_i = w(e)$. Let E_i be the set of edges preceding e in the order of the algorithm plus edge e itself, G_i be the graph induced by those edges and Δ_i be the maximum degree of G_i. The optimal solution for the MEC problem on the graph G_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 1 for the graph G_i are $i \leq 2\Delta_i - 1 \leq 2i^* - 1$, that is $\Rightarrow i^* \geq \lceil \frac{i+1}{2} \rceil$. Hence, $w_i \leq w_{i^*}^* \leq w_{\lceil \frac{i+1}{2} \rceil}^*$.

Summing up the above bounds for all w_i's, $1 \leq i \leq s \leq 2\Delta - 1$, we obtain $W \leq \sum_{i=1}^{2\Delta-1} w_i = w_1^* + 2(\sum_{i=2}^{\Delta} w_i^*) = 2OPT - w_i^*$.

From the first inequality of Lemma 1 we have $\frac{W}{OPT} = \frac{2\sum_{i=1}^{\Delta} w_i^* - w_1^*}{\sum_{i=1}^{s^*} w_i^*} \leq \frac{2\sum_{i=1}^{\Delta} w_i^* - w_1^*}{\sum_{i=1}^{\Delta} w_i^*} \leq 2 - \frac{w_1^*}{\sum_{i=1}^{\Delta} w_i^*} \leq 2 - \frac{w_1^*}{\Delta \cdot w_1^*} = 2 - \frac{1}{\Delta}$, and hence the approximation ratio of Algorithm 1 is $2 - \frac{1}{\Delta}$.

It is well known that the chromatic index of any graph is either Δ or $\Delta + 1$ [21], but deciding between these two values is NP-hard even for cubic graphs [12]. On the other hand, the chromatic index of a bipartite graph is Δ [14]. As in the following we deal only with edge colorings of graphs, the terms k-coloring or k-colorable graph always refer to an edge coloring. It is well known that a $(\Delta + 1)$-coloring of a general graph or a Δ coloring of a bipartite graph can be found in polynomial time. Obviously, such an edge coloring algorithm applied to a weighted graph, leads to a solution for the MEC problem that is feasible but not necessarily optimal. If, in addition, the edge weights in an instance of the MEC problem are very close to each other, then such an algorithm will obtain a solution very close to optimal. In fact, this is the case of tightness example presented in [13] for Algorithm 1. Thus, a natural idea is to combine such an edge coloring algorithm and Algorithm 1 as following (for the case of bipartite graphs).

Algorithm 2

```
1. Run Algorithm 1;
2. Find a solution by a Δ-coloring of the input graph;
3. Select the best solution found;
```

Theorem 1. *Algorithm 2 is a tight* $(2 - \frac{2}{\Delta+1})$*-approximation one for the* MEC *problem on bipartite graphs.*

Proof. (Sketch) By Lemma 1, the solution computed in Line 1 of the algorithm has weight $W \leq 2OPT - w_1^*$. The solution built in Line 2 consists of Δ matchings

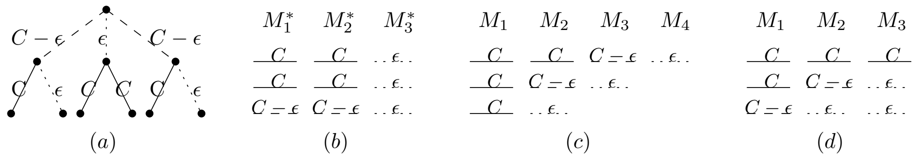

Fig. 1. (a) A instance of the MEC problem where $\Delta = 3$ and $C >> \epsilon$. (b) An optimal solution of weight $2C + \epsilon$. (c) The solution built by Algorithm 1 of weight $3C$. (d) A solution obtained by a Δ-coloring of weight $3C$.

each one of weight at most $w_1^* = w(e_1)$ and it is, therefore, of total weight $W \leq \Delta w_1^*$. Multiplying the second inequality with $1/\Delta$ and adding them we obtain: $(1 + \frac{1}{\Delta})W \leq 2OPT$, that is $W \leq \frac{2\Delta}{\Delta+1}OPT = (2 - \frac{2}{\Delta+1})OPT$.

For the tightness of this ratio let the instance of the MEC problem shown in Figure 1, where an optimal solution as well as the two solutions computed by the algorithm are also shown. The ratio achieved by the algorithm for this instance is $\frac{3C}{2C+\epsilon} \simeq \frac{3}{2} = 2 - \frac{2}{\Delta+1}$.

Note that $2 - \frac{2}{\Delta+1} < 2 - \frac{1}{\Delta}$ for any $\Delta \geq 2$, and thus Algorithm 2 outperforms Algorithm 1. More interestingly, Algorithm 2 outperforms the algorithm proposed in [8] for bipartite graphs of any $\Delta \geq 4$ as well as the algorithm proposed in [16] for bipartite graphs of any $\Delta \geq 9$.

Algorithm 2 can be also extended for general graphs, by creating in Line 2 a $(\Delta + 1)$-coloring of the input graph. The approximation ratio achieved in this case becomes $2 - \frac{2}{\Delta+2}$, which is better than $2 - \frac{1}{\Delta}$, for any $\Delta \geq 3$. By modifying the counterexample for bipartite graphs, we can prove that this ratio is also tight. Thus, the next theorem follows.

Theorem 2. *Algorithm 2 achieves a tight* $2 - \frac{2}{\Delta+2}$ *approximation ratio for general graphs.*

3 Bipartite Graphs

A general idea towards an approximation algorithm for the MEC problem with ratio less than two, is to produce more than one solutions for the problem and to choose the best of them. Algorithm 2 above produces two solutions, while for the case of bipartite graphs with $\Delta = 3$ [5] three solutions were enough to derive a $\frac{7}{6}$ ratio. Algorithms proposed in [8] and [16] are generalizations of this idea, which produce Δ different solutions. In this section we present a new algorithm for the MEC problem on bipartite graphs. It also produces Δ different solutions and chooses the best of them, beats the best known ratios for bipartite graphs for any $\Delta \geq 9$ and it is the first one of this kind yielding approximation ratios that tends asymptotically to 2 as Δ increases.

In our algorithm we repeatedly split a given bipartite graph G, of maximum degree Δ, first into two and then into three edge induced subgraphs. To describe this partition as well as our algorithm, let us introduce some additional notation. Recall that we consider the edges of G sorted in non-increasing order with respect to their weights, i.e., $w(e_1) \geq w(e_2) \geq \ldots \geq w(e_m)$. For this order of edges we denote by $G_{j,k}$, $j \leq k$, the subgraph of G induced by the edges $e_j, e_{j+1}, \ldots, e_k$. We denote by $\Delta_{j,k}$ the maximum degree of graph $G_{j,k}$. By convention, we define $G_{j+1,j}$ to be an empty graph. We denote by j_q the maximum index such that $\Delta_{1,j_q} = q$. It is clear that $j_1 < j_2 < \ldots < j_\Delta = m$.

In general, for each $j = 1, 2, \ldots, j_2$ our algorithm examines a partition of graph G into two edge induced subgraphs: the graph $G_{1,j}$ of $\Delta_{1,j} \leq 2$, induced by the j heaviest edges of G, and the graph $G_{j+1,m}$, induced by the $m-j$ lightest edges of G. For each one of these partitions, the algorithm computes a solution to the MEC problem on graph G.

Moreover, for each pair (j,k), $j = 1, 2, \ldots, j_2$, $k = j+1, \ldots, m$, of indices, our algorithm examines a partition of graph G into three edge induced subgraphs: the graph $G_{1,j}$ of $\Delta_{1,j} \leq 2$, induced by the j heaviest edges of G, the graph $G_{j+1,k}$, induced by the next $k-j$ edges of G, and the graph $G_{k+1,m}$, induced by the $m-k$ lightest edges of G. We shall call such a partition of G a partition (j,k). For each one of these partitions, the algorithm checks the existence of a set of edges in graph $G_{j+1,k}$ and if there exists it computes a solution to the MEC problem on graph G.

The algorithm computes one more solution by finding a Δ-coloring of the original graph G and returns the best among all the solutions found.

Algorithm 3

1. Find a solution $S^0_{1,m}$ by a Δ-coloring of G;
2. For $j = 1, 2, \ldots, j_2$ do
3.   Find an optimal solution $S^1_{1,j}$ for $G_{1,j}$;
4.   Find a solution $S^1_{j+1,m}$ by a Δ-coloring of $G_{j+1,m}$;
5.   Concatenate $S^1_{1,j}$ and $S^1_{j+1,m}$;
6.   For $k = j+1$ to m do
7.   Find an optimal solution $S^2_{1,j}$ for $G_{1,j}$;
8.   If there is a set of edges E' in $G_{j+1,k}$ saturating any vertex of $G_{j+1,k}$ with degree $\Delta_{1,k}$ and E' fits in $S^2_{1,j}$ then
9.   Find a solution $S^2_{j+1,k}$ by a $(\Delta_{1,k}-1)$-coloring of $G_{j+1,k}-E'$;
10.   Find a solution $S^2_{k+1,m}$ by a Δ coloring of $G_{k+1,m}$;
11.   Concatenate $S^2_{1,j}$, $S^2_{j+1,k}$ and $S^2_{k+1,m}$;
12. Return the best solution found in Lines 1, 5 and 11;

The following lemma shows that the check in Line 8 of Algorithm 3 can be done in polynomial time.

Lemma 2. *It is polynomial to determine if there exists a set of edges E' in $G_{j+1,k}$ saturating all vertices of degree $\Delta_{1,k}$ in $G_{j+1,k}$ that fits the solution $S^2_{1,j}$.*

Proof. (Sketch) For a partition (j,k) of G let $d_{1,j}(u)$ and $d_{j+1,k}(u)$ be the degrees of vertex u in subgraphs $G_{1,j}$ and $G_{j+1,k}$, respectively. Consider the subgraph H of $G_{j+1,k}$ induced by its vertices of degree $d_{1,j}(u) \leq \Delta_{1,j} - 1$. Note that, by construction, each edge in H fits in a matching of the solution $S^2_{1,j}$. Let A be the subset of vertices of H of degree $d_{j+1,k}(u) = \Delta_{1,k}$, i.e. the set of vertices which we want to saturate, and B the subset of vertices in A of degree $d_H(u) = 1$. For each vertex $u \in B$ we can clearly insert the single edge (u,v) in E'. Let H' be the subgraph of H induced by its vertices but those in B and $A' \subseteq A$ be the subset of vertices of A that are not saturated by the edges already in E'. It is now enough to find a matching on H' that saturates each vertex in A'. Adding the edges of this matching in E' we get a set that saturates each vertex in A.

Determining if such a matching exists can be done in polynomial time as follows. Consider the graph $Q = (X,F)$ constructed by adding into H' an additional vertex, if the number of vertices in H' is odd, and all the missing edges between the vertices $X - A'$ (i.e., the vertices $X - A'$ induce a clique in Q). If there exists a perfect matching in Q, then there exists a matching in H' saturating all vertices in A', since no edges adjacent to A' have been added in Q. Conversely, if there exists a matching M in H' saturating all vertices in A', then there exists a perfect matching in Q, consisting of the edges of M plus the edges of a perfect matching in the complete subgraph of Q induced by its vertices that are not saturated by M. Therefore, in order to determine if there exists a matching M in H' it is enough to check if there exists a perfect matching in Q. It is well known that this can be done in polynomial time (see for example [17]).

Theorem 3. *Algorithm 3 is a* $(\frac{2\Delta^3}{\Delta^3+\Delta^2+\Delta-1})$*-approximation one for the* MEC *problem on bipartite graphs.*

Proof. (Sketch) The solution obtained by a Δ-coloring of the input graph computed in Line 1 of the algorithm is of weight $W \leq S^0_{1,m} \leq \Delta \cdot w^*_1$, since w^*_1 equals to the heaviest edge of the graph.

In Lines 3–5, consider the solutions obtained in the iterations where $w(e_{j+1}) = w^*_z$, for $z = 2, 3$. In both cases it holds that $\Delta_{1,j} \leq 2$. An optimal solution is computed for $G_{1,j}$ of weight $S^1_{1,j} \leq \sum_{i=1}^{z-1} w^*_i$, since the edges of $G_{1,j}$ are a subset of the edges that appear in the $z-1$ heaviest matchings of the optimal solution. Moreover, a Δ-coloring is built for $G_{j+1,m}$ of weight $S^1_{j+1,m} \leq \Delta \cdot w^*_z$, since e_{j+1} is the heaviest edge of this subgraph. Therefore, $W \leq \sum_{i=1}^{z-1} w^*_i + \Delta \cdot w^*_z$, for $z = 2, 3$.

In Lines 7–11, consider the solutions obtained in the iterations (j,k) where $w(e_{j+1}) = w^*_3$ and $w(e_{k+1}) = w^*_z$, for $4 \leq z \leq \Delta$. In these iterations the set of edges E' exists, since in the optimal solution the edges of $G_{1,k}$ belong in at most $\Delta_{1,k} \leq z-1$ matchings. The edges of E' are lighter than the edges of $G_{1,k}$, and thus it is possible to add them in $S^2_{1,j}$ without increasing its weight. Thus, using the same arguments as for the weight of $S^1_{1,j}$, it holds that $S^2_{1,j} \leq w^*_1 + w^*_2$. The heaviest edges in $G_{j+1,k} - E'$ and $G_{k+1,m}$ are equal to w^*_3 and w^*_z, respectively. Hence, we have that $S^2_{j+1,k} \leq (\Delta_{1,k} - 1) \cdot w^*_3 \leq (z-2) \cdot w^*_3$ and $S^2_{k+1,m} \leq \Delta \cdot w^*_z$. Therefore, $W \leq w^*_1 + w^*_2 + (z-2) \cdot w^*_3 + \Delta \cdot w^*_z$, for $4 \leq z \leq \Delta$.

In this way we have Δ different bounds on W. Multiplying each one of these inequalities with an appropriate factor and adding them we get $\frac{W}{OPT} \leq \frac{2\Delta^3}{\Delta^3+\Delta^2+\Delta-1}$.

The complexity of Algorithm 3 is dominated by the check in Line 8, which by Lemma 2 can be done in polynomial time. This check runs for $\binom{|E|}{2} = O(|E|^2)$ different combinations of weights.

The approximation ratios achieved by Algorithm 3, as Δ increases, are given in the 4th column of Table 1.

4 Trees

The complexity of the MEC problem on trees still remains open, while an exact algorithm of complexity $O(|E|^{2\Delta+O(1)})$ is known [16]. In this section we present a generic algorithm which for a given number k searches exhaustively for the weights of k matchings of an optimal solution. The complexity of our algorithm is $O(|E|^{k+O(1)})$ and, within this time it produces an optimal solution, if $k = 2\Delta-1$, an $(e/(e-1))$-approximate solution, if $k = \Delta$, and a ρ-approximate solution, with $\rho < 2$, if $2 \leq k < \Delta$.

Our algorithm is based upon the fact that the following List Edge-Coloring problem can be solved in polynomial time in trees [6], while it is NP-complete for bipartite graphs even for $\Delta = 3$ [15].

List Edge-Coloring:
Instance: A graph $G = (V, E)$, a set of colors $C = \{C_1, C_2, \ldots, C_k\}$ and for each $e \in E$ a list of authorized colors $L(e)$.
Question: Is there a feasible edge coloring of G, that is a coloring such that each edge e is assigned a color from its list $L(e)$ and adjacent edges are assigned different colors?

The first part of our algorithm searches exhaustively for the weights of the z, $1 \leq z \leq k-1$, heaviest matchings of the optimal solution, $w_1^* \geq w_2^* \geq \ldots \geq w_z^*$. Then, for each z, the graph is partitioned into two subgraphs induced by the edges of weights $w(e) > w_z^*$ and $w(e) \leq w_z^*$, respectively. By a transformation to the List Edge-Coloring problem we obtain a solution for the whole tree consisting of an optimal solution for the first subgraph and a Δ-coloring solution for the second one.

In the second part the algorithm searches exhaustively for the weight of the z-th, $k \leq z \leq \Delta$, matching of the optimal solution w_z^*. Then, the graph G is partitioned into three subgraphs induced by the edges of weights $w(e) > w_{k-1}^*$, $w_{k-1}^* \geq w(e) > w_z^*$ and $w_z^* \geq w(e)$, respectively. By a transformation to the List Edge-Coloring problem we obtain a solution for the whole tree consisting of an optimal solution for the first subgraph, a $(z-k+1)$-coloring solution for the second and a Δ-coloring for the third one.

Algorithm 4

1. Exhaustively search for the weights of the $k-1$ heaviest matchings of the optimal solution, $w_1^* \geq w_2^* \geq \ldots \geq w_{k-1}^*$;
2. For $z = 1, 2, \ldots, k-1$
3. Build the input for the List Edge-Coloring algorithm:
   - Set of colors $\{C_1, C_2, \ldots, C_z, \ldots, C_{z+\Delta-1}\}$;
   - If $w(e) > w_z^*$ then $L(e) = \{C_i : w(e) \leq w_i^*, 1 \leq i \leq z-1\}$;
   - If $w(e) \leq w_z^*$ then $L(e) = \{C_1, C_2, \ldots, C_{z+\Delta-1}\}$;
4. Run the algorithm for the List Edge-Coloring problem;
5. For $z = k, k+1, \ldots, \Delta$
6. Exhaustively search for the weight of the z-th matching of the optimal solution, w_z^*;
7. Build the input for the List Edge-Coloring algorithm:
   - Set of colors $\{C_1, C_2, \ldots, C_{k-1}, \ldots, C_z, \ldots, C_{z+\Delta-1}\}$;
   - If $w(e) > w_{k-1}^*$ then $L(e) = \{C_i : w(e) \leq w_i^*, 1 \leq i \leq k-2\}$;
   - If $w_z^* < w(e) \leq w_{k-1}^*$ then $L(e) = \{C_1, C_2, \ldots, C_{z-1}\}$;
   - If $w(e) \leq w_z^*$ then $L(e) = \{C_1, C_2, \ldots, C_{z+\Delta-1}\}$;
8. Run the algorithm for the List Edge-Coloring problem;
9. Return the best solution found;

Lemma 3. *Algorithm 4 computes a solution for the* MEC *problem of weight*

$$W \leq \begin{cases} w_1^* + w_2^* + \ldots + w_{z-1}^* + \Delta \cdot w_z^*, & \text{if } 1 \leq z \leq k-1 \\ w_1^* + w_2^* + \ldots + w_{k-2}^* + (z-k+1) \cdot w_{k-1}^* + \Delta \cdot w_z^*, & \text{if } k \leq z \leq \Delta \end{cases}$$

Proof. (Sketch) For the first part of the bound, consider the solution computed at the z-th iteration of Line 4 of the algorithm. By the construction of the instance of the List Edge-Coloring problem in Line 3, its solution is also a solution for the MEC problem with $z+\Delta-1$ matchings and matching weights $w_1^*, w_2^*, \ldots, w_{z-1}^*$ plus Δ matchings of weight w_z^*. Observe that, for each z, $1 \leq z \leq k-1$, the List Edge-Coloring algorithm always finds a feasible solution, because (i) the optimal solution for the MEC problem contains a feasible coloring for the edges with weights greater than w_z^* and (ii) there exists a Δ-coloring for the remaining edges, since the graph is a tree. Therefore, in the z-th iteration of Line 4, the algorithm returns a solution of weight $W \leq w_1^* + w_2^* + \ldots + w_{z-1}^* + \Delta \cdot w_z^*$.

For the second part of the bound similar arguments apply for the z-th iteration of Line 8 of the algorithm.

The complexity of Algorithm 4 is exponential in k. In Line 1, the exhaustive search for the weights of the $k-1$ heaviest matchings of the optimal solution examines $\binom{|E|}{k-1} = O(|E|^{k-1})$ combinations of weights. Furthermore, in Line 6, the weight of one more matching is exhaustively chosen in $O(|E|)$ time. For each one of these combinations, an algorithm of complexity $O(|E| \cdot \Delta^{3.5})$ for the List Edge-Coloring problem is called. Thus, the complexity of Algorithm 4 is $O(|E|^{k+1} \cdot \Delta^{3.5})$, that is $O(|E|^{k+O(1)})$, since Δ is $O(|E|)$.

Theorem 4. *Algorithm 4 achieves a* $\frac{e}{e-1} \simeq 1.582$ *approximation ratio for the* MEC *problem within* $O(|E|^{\Delta+O(1)})$ *time.*

Proof. (Sketch) For $k = \Delta$ the second part of the algorithm (Lines 5–8) runs exactly once for $z = \Delta$. Thus, $\Delta - 1$ inequalities are obtained by the first part of Lemma 3 and one inequality by its second part. Multiplying the z-th inequality by $\frac{\Delta^{z-1}\cdot(\Delta-1)^{\Delta-z}}{\Delta^{\Delta}}$, $1 \leq z \leq \Delta$, and adding them we get:

$$\frac{W}{OPT} \leq \frac{\Delta^{\Delta}}{\sum_{k=1}^{\Delta} \Delta^{k-1}\cdot(\Delta-1)^{\Delta-k}} = \frac{\Delta^{\Delta}}{\sum_{k=1}^{\Delta}(\frac{\Delta}{\Delta-1})^{k}\cdot\frac{(\Delta-1)^{\Delta}}{\Delta}} = \frac{\Delta^{\Delta+1}}{(\Delta-1)^{\Delta}\cdot\sum_{k=1}^{\Delta}(\frac{\Delta}{\Delta-1})^{k}}.$$

Using the formulæ $\sum_{k=1}^{\Delta} x^k = \frac{x^{\Delta+1}-x}{x-1}$ and $e = (\frac{x}{x-1})^{x-1}$ it follows that

$$\frac{W}{OPT} \leq \frac{e\cdot(\frac{\Delta}{\Delta-1})}{e\cdot(\frac{\Delta}{\Delta-1})-1} < \frac{e}{e-1}.$$

In a same way, we can prove that, for any fixed $2 \leq k < \Delta$, the approximation ratio achieved by Algorithm 4 for the MEC problem in trees is equal to

$$\frac{2}{(1+\frac{k^2+\Delta-3k+1}{(\Delta-1)^2})(2-\frac{k}{\Delta}-(\frac{\Delta-1}{\Delta})^{k})}.$$

This ratio becomes $\frac{e}{e-1}$ for $k = \Delta$ and its is strictly less than two, for any $k \geq 2$.

Note that if $k = 2\Delta - 1$ the second part of the algorithm is not executed. In this case the algorithm coincides with the exact algorithm in [16].

Furthermore, if $\Delta < k < 2\Delta - 1$, we can modify Algorithm 4 in order to obtain an approximation ratio equal to

$$\frac{1}{1-\frac{2\Delta-1-k}{\Delta}\cdot(\frac{\Delta-1}{\Delta})^{\Delta-1}}.$$

Observe that for such a value of k, Algorithm 4 can create in some iterations not nice solutions i.e., solutions consisting of more than $2\Delta - 1$ matchings. By decreasing the number of colors used in Lines 3 and 7 of the algorithm from $z + \Delta - 1$ to $\min\{z + \Delta - 1, 2\Delta - 1\}$, all the solutions created will be nice, and the above ratio follows. This ratio is between 1, if $k = 2\Delta-1$, and $\frac{e}{e-1}$, if $k = \Delta$.

Table 2 summarizes the ratios achieved by Algorithm 4 for different values of k and Δ, taking into account all the above discussion. The values of $k \geq 5$ have been selected such that $k = 2\Delta - 1$, as for these values the algorithm returns an optimal solution.

5 Complete Graphs

In this section we show that the MEC problem is NP-complete for complete graphs even with bi-valued edge weights. We give a reduction from the classical edge coloring problem, which is known to be NP-complete even for cubic graphs [12]. In this problem we are given a graph $G = (V, E)$ with $d(v) = 3$, for each $v \in V$, and we ask if there exists a 3-coloring of the edges of G, that is a partition of the set of edges E into three matchings (colors).

Theorem 5. *The* MEC *problem is NP-complete even in complete graphs with edge weights* $w(e) \in \{1, 2\}$.

Table 2. Approximation ratios for trees

Δ	k											
	2	3	4	5	7	9	19	29	39	59	79	99
3	1.50	1.42	1.17	*OPT*								
4	1.60	1.55	1.46	1.27	*OPT*							
5	1.67	1.64	1.56	1.49	1.20	*OPT*						
10	1.82	1.81	1.79	1.75	1.64	1.56	*OPT*					
15	1.88	1.87	1.86	1.84	1.78	1.71	1.34	*OPT*				
20	1.90	1.90	1.90	1.89	1.85	1.80	1.56	1.23	*OPT*			
30	1.94	1.94	1.93	1.93	1.91	1.89	1.70	1.57	1.33	*OPT*		
40	1.95	1.95	1.95	1.95	1.94	1.92	1.80	1.65	1.57	1.23	*OPT*	
50	1.96	1.96	1.96	1.96	1.95	1.94	1.86	1.73	1.63	1.42	1.17	*OPT*

Proof. (Sketch) Given an instance of the edge coloring on a cubic graph $G = (V, E)$, $|V| = n$, we construct the complete weighted graph K_n with edge weights $w(e) = 2$, for each $e \in E$, and $w(e) = 1$, for each $e \notin E$. We will show that there is a 3-coloring of G iff there is a solution for the MEC problem on K_n of weight at most $n + 2$, if n is even, or $n + 3$, if n is odd.

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 remaining edges of K_n (those of weight 1). $K_n - G$ is a $(n-4)$-regular graph. If n is even, then $K_n - G$ is $(n-4)$-colorable [3]. Therefore, there is a solution for the MEC problem on K_n of weight at most $3 \cdot 2 + (n - 4) \cdot 1 = n + 2$. If n is odd, then $K_n - G$ is $(n - 3)$-colorable as an overfull graph[1]. Therefore, there is a solution for the MEC problem on K_n of weight at most $3 \cdot 2 + (n - 3) \cdot 1 = n + 3$.

Conversely, consider, first, that n is even and we have a solution for the MEC problem for K_n of weight at most $n + 2$. This solution contains $s \geq n - 1$ matchings, since a complete graph of even order has chromatic index equal to $n-1$ [10]. By the construction of K_n, any solution for the MEC problem contains at least three matchings of weight equal to 2, since there are exactly three edges of weight 2 adjacent to each vertex. Assume that there was a forth matching of weight equal to 2. In this case we get a solution of weight at least $4 \cdot 2 + (s-4) \cdot 1 \geq n + 3$, a contradiction. Thus, in a solution to the MEC problem on K_n there exist exactly 3 matchings of weight equal to 2, which imply a 3-coloring for G.

Using the same arguments, we can prove that if n is odd and there is a solution for the MEC problem for K_n with weight at most $n + 3$ then we can get a 3-coloring of G. The only difference, here, is that a complete graph of odd order has chromatic index equal to n [10].

Theorem 5 implies that the MEC problem is NP-complete in all superclasses of complete graphs, including split and interval graphs. On the other hand, the

[1] A graph is called overfull if $|E| > \Delta \cdot \lfloor \frac{|V|}{2} \rfloor$. It is easy to see that an overfull graph is $(\Delta + 1) - colorable$.

MEC problem on bipartite graphs with edge weights $w(e) \in \{1, t\}$ is polynomially solvable [7].

In what follows, we present an approximation algorithm for general graphs with two different edge weights. Assume that the edges of the graph $K_n = (V, E)$ have weights either 1 or t, where $t \geq 2$. Let $G_1 = (V, E_1)$, of maximum degree Δ_1, and $G_t = (V, E_t)$, of maximum degree Δ_t, be the graphs induced by the edges of K_n with weights 1 and t, respectively.

Algorithm 5

```
1. Find a solution by a (Δ + 1)-coloring of K_n;
2. Find a solution by a (Δ_1 + 1)-coloring of G_1,
   a solution by a (Δ_t + 1)-coloring of G_t and concatenate them;
3. Return the best of the two solutions found;
```

Theorem 6. *Algorithm 5 achieves an asymptotic $\frac{4}{3}$-approximation ratio for the* MEC *problem on general graphs of arbitrarily large Δ and edge weights $w(e) \in \{1, t\}$.*

Proof. (Sketch) An optimal solution contains at least $\Delta(K_n) = n - 1$ matchings and at least Δ_t of them are of weight equal to t. Therefore, a lower bound to the total weight of an optimal solution is $OPT \geq \Delta_t \cdot t + (\Delta - \Delta_t)$.

By Vizing's theorem any graph has a $(\Delta + 1)$-coloring. Using such colorings the algorithm computes in Line 1 a solution of total weight $W \leq (\Delta+1) \cdot t$, and in Line 2 a solution of total weight $W \leq (\Delta_t+1) \cdot t + (\Delta_1+1) \cdot 1 \leq (\Delta_t+1) \cdot t + (\Delta+1)$. Multiplying the first inequality with $\frac{\Delta_t^2 + 2\Delta_t - \Delta}{(\Delta+1)^2}$, 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 $\frac{W}{OPT} \leq \frac{4(\Delta+1)}{(\Delta+1) + 2(\Delta-1)} = \frac{4\Delta+4}{3\Delta-1} = \frac{4}{3} + \frac{16}{9\Delta-3}$.

References

1. Afrati, F.N., Aslanidis, T., Bampis, E., Milis, I.: Scheduling in switching networks with set-up delays. Journal of Combinatorial Optimization 9, 49–57 (2005)
2. Brucker, P., Gladky, A., Hoogeveen, H., Koyalyov, M., Potts, C., Tautenham, T., van de Velde, S.: Scheduling a batching machine. Journal of Scheduling 1, 31–54 (1998)
3. Chetwynd, A.G., Hilton, A.J.W.: Regular graphs of high degree are 1-factorizable. In: Proceedings of the London Mathematical Society, vol. 50, pp. 193–206 (1985)
4. Crescenzi, P., Deng, X., Papadimitriou, C.H.: On approximating a scheduling problem. Journal of Combinatorial Optimization 5, 287–297 (2001)
5. de Werra, D., Demange, M., Escoffier, B., Monnot, J., Paschos, V.T.: Weighted coloring on planar, bipartite and split graphs: Complexity and improved approximation. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 896–907. Springer, Heidelberg (2004)
6. de Werra, D., Hoffman, A.J., Mahadev, N.V.R., Peled, U.N.: Restrictions and preassignments in preemptive open shop scheduling. Discrete Applied Mathematics 68, 169–188 (1996)

7. Demange, M., de Werra, D., Monnot, J., Paschos, V.T.: Weighted node coloring: When stable sets are expensive. In: Kučera, L. (ed.) WG 2002. LNCS, vol. 2573, pp. 114–125. Springer, Heidelberg (2002)
8. Escoffier, B., Monnot, J., Paschos, V.T.: Weighted coloring: further complexity and approximability results. Information Processing Letters 97, 98–103 (2006)
9. Finke, G., Jost, V., Queyranne, M., Sebő, A.: Batch processing with interval graph compatibilities between tasks. Technical report, Cahiers du laboratoire Leibniz (2004), `http://www-leibniz.imag.fr/NEWLEIBNIZ/LesCahiers/index.xhtml`
10. Fiorini, S., Wilson, R.J.: Edge-Colourings of Graphs. Pitman, London (1977)
11. Gopal, I.S., Wong, C.: Minimizing the number of switchings in a SS/TDMA system. IEEE Transactions On Communications 33, 497–501 (1985)
12. Holyer, I.: The NP-completeness of edge-coloring. SIAM Journal on Computing 10, 718–720 (1981)
13. Kesselman, A., Kogan, K.: Nonpreemptive scheduling of optical switches. IEEE Transactions on Communications 55, 1212–1219 (2007)
14. König, D.: Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. Mathematische Annalen 77, 453–465 (1916)
15. Kubale, M.: Some results concerning the complexity of restricted colorings of graphs. Discrete Applied Mathematics 36, 35–46 (1992)
16. Lucarelli, G., Milis, I., Paschos, V.T.: On a generalized graph coloring/batch scheduling problem. In: 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA), pp. 353–360 (2007)
17. Micali, S., Vazirani, V.V.: An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In: 21st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 17–27 (1980)
18. Pemmaraju, S.V., Raman, R.: Approximation algorithms for the max-coloring problem. In: 32nd International Colloquium on Automata, Languages and Programming (ICALP), pp. 1064–1075 (2005)
19. Pemmaraju, S.V., Raman, R., Varadarajan, K.R.: Buffer minimization using max-coloring. In: 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 562–571 (2004)
20. Rendl, F.: On the complexity of decomposing matrices arising in satellite communication. Operations Research Letters 4, 5–8 (1985)
21. Vizing, V.G.: On an estimate of the chromatic class of a p-graph. Diskret. Analiz. 3, 25–30 (1964)