On the Minimum Load Coloring Problem

Nitin Ahuja [a] Andreas Baltz [b,*] Benjamin Doerr [c]
Aleš Přívětivý [d] Anand Srivastav [b]

[a] *Department of Mathematical Optimization, Technical University Braunschweig, Pockelsstrasse 14, D-38106 Braunschweig, Germany*

[b] *Department of Computer Science, Christian-Albrechts-University Kiel, Christian-Albrechts-Platz 4, D-24098 Kiel, Germany*

[c] *Max-Planck-Institute for Computer Science, Stuhlsatzenhausweg 85, D-66123 Saarbrücken, Germany*

[d] *Department of Applied Mathematics, Charles University, Malostranské nám. 25, 11800 Praha, Czech Republic*

Abstract

Given a graph $G = (V, E)$ with n vertices, m edges and maximum vertex degree Δ, the *load distribution* of a coloring $\varphi : V \to \{\text{red, blue}\}$ is a pair $d_\varphi = (r_\varphi, b_\varphi)$, where r_φ is the number of edges with at least one end-vertex colored red and b_φ is the number of edges with at least one end-vertex colored blue. Our aim is to find a coloring φ such that the (maximum) *load*, $l_\varphi := \frac{1}{m} \cdot \max\{r_\varphi, b_\varphi\}$, is minimized. This problems arises in Wavelength Division Multiplexing (WDM), the technology currently in use for building optical communication networks. After proving that the general problem is NP-hard we give a polynomial time algorithm for optimal colorings of trees and show that the optimal load is at most $1/2 + (\Delta/m)\log_2 n$. For graphs with genus $g > 0$, we show that a coloring with load $\mathrm{OPT}(1 + o(1))$ can be computed in $O(n + g \log n)$-time, if the maximum degree satisfies $\Delta = o(\frac{m^2}{ng})$ and an embedding is given. In the general situation we show that a coloring with load at most $\frac{3}{4} + O(\sqrt{\Delta/m})$ can be found by analyzing a random coloring with Chebychev's inequality. This bound describes the "typical" situation: in the random graph model $G(n, m)$ we prove that for almost all graphs, the optimal load is at least $\frac{3}{4} - \sqrt{n/m}$. Finally, we state some conjectures on how our results generalize to k–colorings for $k > 2$.

Key words: graph coloring, graph partitioning

---

* corresponding author

Preprint submitted to Discrete Algorithms *5 October 2006*

1 Introduction

We consider the following problem. Given a graph $G = (V, E)$ on n vertices and m edges, define the load of a k-coloring $\varphi : V \to \{1, \ldots, k\}$ as

$$\frac{1}{m} \cdot \max_{i \in \{1,\ldots,k\}} |\{e \in E \mid \varphi^{-1}(i) \cap e \neq \emptyset\}|,$$

the maximum fraction of edges with at least one end-point in color i, where the maximum is taken over all $i \in \{1, \ldots, k\}$. The question we ask is: How can one minimize the load over all k-colorings?

In this paper the focus is on coloring the vertices of a graph with 2 colors, red and blue. For a *coloring* $\varphi : V \to \{\text{red, blue}\}$ we define the *load distribution* of φ by $d_\varphi := (r_\varphi, b_\varphi)$, where r_φ counts the number of edges incident with at least one red vertex, and b_φ is the number of edges incident with at least one blue vertex. The aim is to find a coloring φ such that the maximum load, $l_\varphi := \frac{1}{m} \cdot \max\{r_\varphi, b_\varphi\}$, is minimized. In the following we shall skip the term "maximum" and refer to l_φ simply as the *load* of the coloring φ. We call the problem of finding a coloring φ that minimizes l_φ *Minimum Load Coloring Problem (MLCP).*

MLCP is a natural *judicious graph partitioning problem* aiming at the optimization of several quantities at the same time. Let (V_r, V_b) be a bipartition of G and let $E(V_r)$ (resp. $E(V_b)$) denote the set of edges with both end-points in V_r (resp. V_b). Then our problem is equivalent to maximizing $\min\{|E(V_r)|, |E(V_b)|\}$ over all bipartitions of G.

Remark 1.1 *Let $G = (V, E)$ be a graph, (V_r, V_b) a bipartition of V, and $\varphi : V \to \{\text{red}, \text{blue}\}$ a coloring with $V_r = \varphi^{-1}(\text{red})$, $V_b = \varphi^{-1}(\text{blue})$. Then, the bipartition (V_r, V_b) maximizes $\min\{|E(V_r)|, |E(V_b)|\}$ if and only if l_φ is minimum.*

Proof. Take any partition (V'_r, V'_b) and its corresponding coloring φ'. We may assume without loss of generality that $|E(V_b)| \leq |E(V_r)|$ and $|E(V'_b)| \leq |E(V'_r)|$. This implies $r_\varphi = |E(V_r)| + |E(V_r, V_b)| \geq |E(V_b)| + |E(V_r, V_b)| = b_\varphi$ – where $E(V_r, V_b)$ denotes the set of edges connecting V_r to V_b – and similarly $r_{\varphi'} \geq b_{\varphi'}$. Now, $|E(V_b)| \geq |E(V'_b)|$ is equivalent to $r_\varphi = |E(V_r)| + |E(V_r, V_b)| = m - |E(V_b)| \leq m - |E(V'_b)| = |E(V'_r)| + |E(V'_r, V'_b)| = r_{\varphi'}$. □

Similar graph partitioning problems were studied in the survey article of Scott [12]. One of the problems mentioned in this survey article is minimize $\max\{|E(V_r)|, |E(V_b)|\}$ over all bipartitions of G. The max-min and the min-max problem are not equivalent as can be seen from the following examples: while for chains (graphs consisting of a single open path) the optima are the same, on bipartite graphs with $4n$ vertices and $m = 2n$ edges covering all vertices, the optimal solutions differ by $m/2$.

Motivation

Wavelength division multiplexing (WDM) enables a simple optical fiber to carry more information per unit time by dividing its carrying capacity into many channels. Each of these channels uses a separate wavelength or frequency. Thus, in a broadcast WDM network with a passive star coupler a packet transmitted by a node on a particular channel can be received by all nodes with receivers tuned to that channel. A broadcast WDM network with a passive star coupler consists of n nodes. Each node or user has a transmitter (resp. receiver) that can be used to transmit (resp. receive) packets using any of the k ($< n$) channels. More precisely, a node can transmit (resp. receive) copies of a packet, albeit not simustaneously, on different channels. It is not allowed to transmit two packets on the same channel at the same time. The reader is referred to the survey of Rouskas and Thaker [11] or [3] for more details.

We call those nodes of the network that want to send packets of data to other nodes *active nodes*. We assume that each active node wants to send its packet to exactly r unique other nodes. This enables us to model the WDM network as an r-uniform hypergraphs where vertices correspond to the nodes of the network and hyperedges represent joint destinations of data packets sent by active nodes. As an example consider the WDM network in Figure 1 with $n = 5$ nodes/users all of which are active and $r = 2$ (on the left) and the corresponding graph (on the right). In this WDM network node 1 wants to send a message to nodes 2 und 3, node 2 wants to transmit a message to nodes 1 and 5, and so on. The graph on the right can be interpreted as follows. The

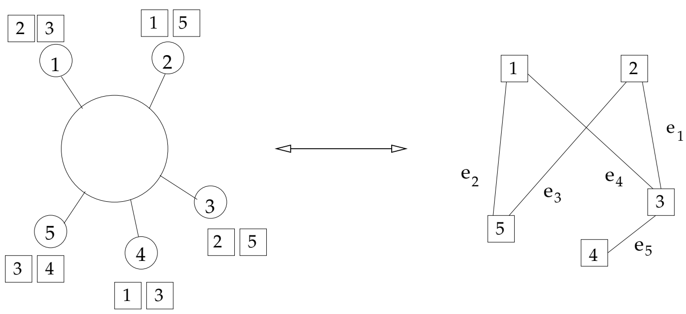

Fig. 1. A broadcast WDM network and the corresponding graph

edges in the graph correspond to the active nodes in the network. The two end-vertices of an edge e_i in the graph represent the two nodes in the network to which node i wants to send a message to. So the end-vertices of edge e_1 in the graph are 2 and 3, the end-vertices of edge e_2 in the graph are 1 and 5, and so on. Taking the number of channels, k, to be the number of colors, we can define an MLCP on this graph.

Since the number of channels is less than the number of nodes, two kinds of optimization problems arise. The first problem is, given the demands of the n nodes, assign channels to receivers of these nodes such that no channel or wavelength is overloaded with packets. The network performs better when the maximum load over all the channels is minimized. Once the assignment part is over, the next problem is to schedule the broadcast of packets so as to minimize the duration or makespan of the broadcast. Baldine and Rouskas [3] use dynamic load balancing while reconfiguring the demands of some nodes. In [1], Ageev et al. consider the scheduling aspects stated above. Bampis and Rouskas [4] give approximation algorithms that take both the assignment part and the scheduling part of the whole problem into account.

In this paper the main focus is on the case $k = 2$ and $r = 2$. We hope that our results can serve as a basis for further research on the general case corresponding to an MLCP on a non-uniform multi-hypergraph.

Our Results

After some preliminaries including the establishment of NP-hardness of the problem in Section 2, we show how to solve the MLCP on trees optimally in $O(n^3)$ time (Section 3). Such an optimal solution is proven to have a load of at most $\frac{1}{2} + (\Delta/m)\log_2 n$. Section 4 is concerned with graphs of genus $g > 0$. With a separator theorem proved with techniques from Djidjev [6] we obtain an $O(n + g)$-time algorithm for constructing a coloring with load bounded by $1/2 + 48\sqrt{g\Delta n}/m$. This is a $(1 + o(1))$-approximation in case $\Delta = o(\frac{m^2}{ng})$. In Section 5 we analyze arbitrary instances of the problem. We show that a random coloring has load $\frac{3}{4} + O(\sqrt{\Delta/m})$ with high probability. This immediately yields a randomized algorithm that can be turned into a deterministic algorithm via the method of conditional probabilities [2]. The bound is quite strong: in the random graph model $G(n, m)$, almost all graphs have no coloring with load less than $\frac{3}{4} - \sqrt{n/m}$. In the last section we consider extensions of our results to $k > 2$ colors.

2 Preliminaries

In this section we state some basic facts. Let

$$l(G) := \min\{l_\varphi \mid \varphi : V \to \{\text{red,blue}\}\}$$

denote the optimal load of a graph $G = (V, E)$. Given a red-blue coloring φ, we shall refer to the set of red vertices as V_r and to the set of blue vertices as V_b. The set of edges connecting V_r and V_b is denoted by $E(V_r, V_b)$.

Remark 2.1 *Let G be a graph, then $l(G) \in [1/2, 1]$.*

Proof. Since every edge of G is counted as red or blue (or both), $l(G) \geq \frac{1}{2}$. Obviously, every red-blue coloring of G has load at most 1. □

In particular, each two-coloring of a graph G is a 2-approximation of $l(G)$.

Let G be a star, then $l(G) = 1$. In fact, stars and the triangle graph are the only graphs with optimum load 1.

Remark 2.2 *Let $G = (V, E)$ be a graph with $l(G) = 1$ then G is either a triangle or a star.*

Proof. If G is neither a triangle nor a star, then E contains two edges e_1, e_2 with $e_1 \cap e_2 = \emptyset$. Coloring the endpoints of e_1 in red and the rest of E in blue yields a load that is strictly less than 1. □

Stars also show that the maximum degree Δ of the input graph yields a lower bound of Δ/m on $l(G)$. It is easy to see that the lower bound of 1/2 is attained only if G consists of monochromatic components.

Remark 2.3 *Let $G = K_n$ be the complete graph on n vertices. A coloring φ with $r_\varphi \geq b_\varphi$ induces minimum load on G if and and only if $|V_r| = \lceil n/2 \rceil$.*

Proof. ¿From $\binom{|V_r|}{2} + |V_r|(n - |V_r|) = r_\varphi \geq b_\varphi = \binom{n-|V_r|}{2} + |V_r|(n - |V_r|)$, we infer that $|V_r| \geq n/2$. Since the number of red edges is increasing with $|V_r|$ the claim follows. □

It is also easy to find an optimal two-coloring of cycles and chains. Here, each of the two classes in an optimal coloring forms a connected component. This is already false for trees (cf. Section 3).

Let us observe that for regular graphs on an even number of vertices, the MLCP is equivalent to the MinBisection-problem of dividing the set of vertices into two equal halves that are connected by a minimum number of edges.

Lemma 2.1 *Let $k \in \mathbb{N}$. Let $G = (V, E)$ be a k-regular graph with $n := |V|$ even, and let $\varphi : V \to \{red, blue\}$ be an optimal coloring, then either $|V_r| = |V_b|$ or an optimal coloring with $|V_r| = |V_b|$ can be obtained by recoloring an arbitrary vertex of the larger color class.*

Proof. Suppose that $|V_r| > |V_b|$. The number of red edges is $r_\varphi = \frac{|V_r|\cdot k}{2} + \frac{|E(V_r,V_b)|}{2}$, and the number of blue edges is $b_\varphi = \frac{|V_b|\cdot k}{2} + \frac{|E(V_r,V_b)|}{2}$, hence

$$r_\varphi - b_\varphi = \frac{k}{2}(|V_r| - |V_b|) \geq k$$

since n is even. If we change the color of an arbitrary red vertex v into blue, the number of red edges decreases by at most k, while the number of blue edges increases by at most k. Consequently, l_φ does not increase and the resulting coloring is still optimal. On the other hand, l_φ must not decrease either. This means that r_φ has to stay the same or b_φ has to increase by at least k. Either of these events can occur only if v has only red neighbors. Since v is an arbitrary red vertex, we conclude that G consists of monochromatic components. If $|V_r| > |V_b| + 2$ we can recolor another red vertex v' without increasing l_φ. But choosing v' as a neighbor of v results in an overall decrease of l_φ contradicting the choice of φ as an optimal coloring. Hence $|V_r| = |V_b| + 2$, and thus recoloring v yields an optimal coloring with $|V_r| = |V_b|$. □

Given a k-regular graph with an even number of vertices, we see by Lemma 2.1 that every optimal coloring φ induces a bisection of V (either at once or after recoloring an arbitrary vertex of the larger class) with

$$l_\varphi = \frac{n}{2m} \cdot \frac{k}{2} + \frac{|E(V_r, V_b)|}{2m}.$$

Since φ is optimal, $|E(V_r, V_b)|$, the size of the edge cut separating the classes V_r and V_b, is minimum, so we have a minimum bisection. On the other hand, every minimum bisection V_1, V_2 of V gives rise to a coloring with load

$$\frac{n}{2m} \cdot \frac{k}{2} + \frac{|E(V_1, V_2)|}{2m},$$

where $E(V_1, V_2)$ denotes the set of edges between V_1 and V_2. Obviously, this load is optimal. Hence MLCP and MinBisection are equivalent on regular graphs with n even. For $k \geq 3$, MinBisection on k-regular graphs is as hard as general MinBisection (see [5]). Since the decision version of MinBisection is NP-complete [7], and the load of any proposed solution for the MLCP can be evaluated in polynomial time, we have established NP-completeness also for the MLCP.

Theorem 2.1 *The decision version of the MLCP is NP-complete.*

3 Polynomial Time Algorithms for Trees

In this section, we show how to efficiently compute an optimal solution for the MLCP on trees. We also show that any tree G with n vertices and maximum vertex degree Δ has load at most $\frac{1}{2} + (\Delta/m)\log_2 n$. The key to prove this result is the following more general lemma.

Lemma 3.1 *Let $G = (V, E)$ be a tree on n vertices and let $m_1, m_2 \in \mathbb{N}$ such that $m_1 + m_2 = n - 1$. Then there is a red-blue coloring of V such that at least $m_1 + 1 - \Delta \log_2 n$ edges are monochromatic red and at least $m_2 + 1 - \Delta \log_2 n$ are monochromatic blue.*

Proof. We use induction. Clearly, the lemma holds for $n \leq 3$. Let us assume that the lemma holds for all trees on less than n vertices. Let $v \in V$ be a vertex such that deleting v breaks G into $k \geq 2$ components C_i, $i \in \{1, \ldots, k\}$, where the number of vertices n_i in component C_i is at most $n/2$. (To show the existence of v, assume for the sake of a contradiction that each vertex is the origin of at least one branch with more than $\frac{n}{2}$ nodes. Let v be a vertex whose maximum branch C is minimum, and let v' be the neighbor of v in C. Denote the maximum branch of v' by C'. Then $|C'| \leq \max\{n - |C|, |C| - 1\} \leq \max\{\frac{n}{2} - 1, |C| - 1\} < |C|$ contradicting the minimality of $|C|$.) It is easy to see that there exist $I_1, I_2 \subseteq \{1, \ldots, k\}$ such that:

(i) $I_1 \cap I_2 = \emptyset$,
(ii) $|\{1, \ldots, k\} \setminus (I_1 \cup I_2)| = 1$,
(iii) $\sum_{i \in I_1} n_i \leq m_1$, and
(iv) $\sum_{i \in I_2} n_i \leq m_2$.

Note that either I_1 or I_2 can also be empty, but not both. Color the vertices of components with indices in I_1 (resp. I_2) with red (resp. blue). The central vertex v is arbitrarily colored red or blue. Let C_j be the component that is left uncolored, that is, $\{1, \ldots, k\} \setminus (I_1 \cup I_2) = \{j\}$. Let $m_1' = m_1 - (\sum_{i \in I_1} n_i) - 1$ and $m_2' = m_2 - \sum_{i \in I_2} n_i$. Then, $m_1' + m_2' = n_j - 1$ is a partition of the number of edges of C_j. By induction, there is a red-blue coloring of C_j such that at least $m_1' + 1 - \Delta \log_2 n_j$ of its edges are monochromatic red and at least $m_2' + 1 - \Delta \log_2 n_j$ are monochromatic blue. Now, the total number of monochromatic red edges is at least $\sum_{i \in I_1}(n_i - 1) + m_1' + 1 - \Delta \log_2 n_j \geq m_1 - |I_1| - \Delta \log_2(n/2)$, which is at least $m_1 + 1 - \Delta \log_2 n$. Similarly, the total number of monochromatic blue edges is at least $m_2 + 1 - \Delta \log_2 n$. □

¿From the lemma, we easily deduce the following.

Theorem 3.1 *Let $G = (V, E)$ be a tree on n vertices with maximum vertex degree Δ. Then $l(G) \leq \frac{1}{2} + (\Delta/m)\log_2 n$.*

We did not try to optimize the error term $(\Delta/m)\log_2 n$. It is clear that it has to contain a linear dependence on Δ, this is shown by stars, and a logarithmic dependence on the number of vertices. The latter is shown by a complete ternary tree.

Remark 3.1 *Let G be a complete ternary tree, then $l(G) \geq \frac{1}{2} + \frac{\log_3(n)-1}{2m}$.*

Proof. For $d \in \mathbb{N}$, we denote by $G_d = (V_d, E_d)$ the complete ternary tree with leaves at distance d from the root. We consider the nodes as grouped into levels according to the distance from the root, so the root is at level 0, and the leaves are the level d nodes. Moreover we view the edges as being directed from lower to higher level nodes. Let φ be an optimal coloring of G_d such that $r_d := r_\varphi = m \cdot l(G_d)$ and $b_d := b_\varphi$ is as small as possible. We call a node v *i-unsaturated* ($i \in \{0, \ldots, 3\}$) if exactly i of its direct successors have a different color than v. A 0-unsaturated vertex is called *saturated*, and an i-unsaturated vertex with $i > 0$ is referred to as *unsaturated*. We will show that without loss of generality we may assume that

(i) G_d contains no 3-unsaturated vertices,
(ii) each level contains at most one unsaturated vertex.

Consider the levels in increasing order: moving any unsaturated vertex and all of its successors to the center of its respective level does not change the load distribution of the coloring. We can thus perform these operations while maintaining the values of r_d and b_d. After this procedure G_d has the following structure: it consists of a red G_{d-1}, an optimally colored G_{d-1}, and a blue G_{d-1} that are connected to the (red or blue) root vertex. Since for each $d' \in \mathbb{N}$, $G_{d'}$ has $\frac{3^{d'+1}-3}{2}$ edges, we can recursively estimate

$$\begin{aligned}
2 \cdot r_d &\geq r_d + b_d \geq 2 \cdot |E_{d-1}| + r_{d-1} + b_{d-1} + 4 \\
&\geq 2 \cdot (|E_{d-1}| + |E_{d-2}|) + r_{d-2} + b_{d-2} + 8 \\
&\geq 2 \cdot (|E_{d-1} + \ldots + |E_1|) + 3 + 4(d-1) \\
&= 2\left(\sum_{i=1}^{d-1} \frac{3^{i+1}-3}{2}\right) + 4d - 4 \\
&= \frac{3^{d+1}-1}{2} - 4 - 3(d-1) + 4d - 1 \\
&= \frac{3^{d+1}-3}{2} + d - 1 \\
&= |E_d| + d - 1,
\end{aligned}$$

and hence, $r_d \geq \frac{|E_d|+d-1}{2}$. Since $|E_d| + d - 1$ is odd we infer

$$l(G_d) \geq \frac{|E_d| + d}{2m} \geq \frac{1}{2} + \frac{\log_3(n) - 1}{2m}.$$

To see that (i) holds, note that if G_d does not contain both red and blue leaves, then

$$l(G_d) \geq 3^d/m = \frac{2|E_d|+3}{3m} > \frac{1}{2} + \log_3(n)/m.$$

So we may assume that G_d has red and blue leaves. Since r_d and b_d are minimal, G_d contains triples of red and blue leaves whose predecessors are saturated. Hence G_d does not contain any 3-unsaturated vertex at level $d-1$. Clearly the root of G_d cannot be 3-unsaturated either. Now consider a 3-unsaturated vertex v of arbitrary color, say red. It is easy to check that by coloring v and its predecessor in blue and replacing three blue leaves together with their predecessor by red vertices, the load can only decrease.

For the proof of (ii), we can convince ourselves that given two unsaturated vertices at the same level saturate one of them without increasing the load by exchanging two of the unsaturated vertices' branches. □

This example also demonstrates that in an optimal coloring the color classes may induce disconnected subgraphs.

Note that the proof of Lemma 3.1 is constructive. We thus have an efficient algorithm computing colorings with load at most $\frac{1}{2} + (\Delta/2)\log_2 n$. However, it is also possible to compute optimal colorings for trees efficiently.

Theorem 3.2 *On trees with n vertices, the MLCP can be solved in time $O(n^3)$.*

Proof. Let $G = (V, E)$ be a tree on n vertices. Let us consider G as being rooted in some arbitrary vertex a. We assign each $v \in V$ a distance dist_v given by the length of the path from a to v and view each edge $e \in E$ as pointing from lower to higher level nodes. So, we think of G as a directed tree with the root a at level 0, the successors $N(a) := \{v \in V \mid (a, v) \in E\}$ of a at level 1, etc. For each $v \in V$ we denote by T_v the induced subtree of G rooted in v, i.e., T_v is the subgraph of G induced by v and *all* of its (iterated) successors. We define for each *arbitrary* subtree G' of G with root a',

$$D_{G'} := \{(r, b) \mid (r, b) = d_\varphi \text{ for some coloring } \varphi \text{ of } G' \text{ with } \varphi(a') = \text{red}\},$$

the set of possible load distributions for G' (we may assume $\varphi(a') = \text{red}$ without loss of generality). Suppose, we can efficiently compute D_G. Since $|D_G| \leq n^2$, we can also efficiently find the load $l(G)$ of an optimal coloring by searching D_G for the load distribution with smallest maximum component. We will show that D_G can be determined in polynomial time by iteratively computing D_{T_v} for all $v \in V$, in *reverse breadth first order*. The iteration is based on two operations:

(OP1) Consider a subtree G' of G with root $a' \neq a$, $v \in V$ with $(v, a') \in E$, and the tree $v + G' := (V(G') \cup \{v\}, E(G') \cup \{(v, a')\})$ obtained by appending the edge (v, a') to G'. We define

$$v + D_{G'} := \{(r+1, b) \mid (r, b) \in D_{G'}\} \cup \{(b+1, r+1) \mid (r, b) \in D_{G'}\}.$$

(OP2) Consider two subtrees G'_1, G'_2 of G that do not intersect but in their joint root a'. Let $G'_1 + G'_2 := (V(G'_1) \cup V(G'_2), E(G'_1) \cup E(G'_2))$ denote the composite tree and define

$$D_{G'_1} + D_{G'_2} := \{(r_1 + r_2, b_1 + b_2) \mid (r_1, b_1) \in D_{G'_1}, (r_2, b_2) \in D_{G'_2}\}.$$

Since for each tree G' we defined $D_{G'}$ to contain only load distributions of colorings where the root of G' is colored red, it will be necessary to eventually flip colors in the course of our desired iteration. For convenience, let us denote the *inverse coloring* of a given coloring φ (where colors red and blue are exchanged) by $\overline{\varphi}$.

Claim 1. *For all subtrees $G' = (V', E')$ of G with root a' and all $v \in V$ with $(v, a') \in E$, $D_{v+G'} = v + D_{G'}$.*

Proof. Let $(r, b) \in D_{v+G'}$ and let $\varphi : V' \cup \{v\} \to \{\text{red, blue}\}$ be a coloring with $d_\varphi = (r, b)$ and $\varphi(v) = \text{red}$. Let $\varphi' := \varphi|_{V'}$ denote the restriction of φ on V'. Then f' is a coloring of G'. If $\varphi'(a') = \text{red}$, then $(r', b') := d_{\varphi'} = (r-1, b) \in D_{G'}$ and thus $(r, b) = (r'+1, b) \in v + D_{G'}$, whereas if $\varphi'(a')=$ blue, then $d_{\varphi'} = (r-1, b-1)$ and $\overline{\varphi'}$ induces a load distribution $d_{\overline{\varphi'}} = (r', b') := (b-1, r-1) \in D'_G$, so $(r, b) = (b'+1, r'+1) \in v + D_{G'}$.

Let $(r, b) \in v + D_{G'}$. There is a coloring $\varphi : V' \to \{\text{red, blue}\}$ with $\varphi(a') = \text{red}$ and either $d_\varphi = (r-1, b)$ or $d_\varphi = (b-1, r-1)$. In the first case, extending φ to $V' \cup \{v\}$ by coloring v red gives a coloring φ' of $v + G'$ with $d_{\varphi'} = (r, b)$, in the second case we similarly extend $\overline{\varphi'}$. □

Claim 2. *For all subtrees $G'_1 = (V'_1, E'_1), G'_2 = (V'_2, E'_2)$ intersecting only in their joint root a', $D_{G'_1+G'_2} = D_{G'_1} + D_{G'_2}$.*

Proof. Let $(r, b) \in D_{G'_1+G'_2}$ and let $\varphi : V'_1 \cup V'_2 \to \{\text{red, blue}\}$ be a coloring with $d_\varphi = (r, b)$ and $\varphi(a') = \text{red}$. Obviously, $\varphi|_{V'_1}$ and $\varphi|_{V'_2}$ are colorings of G'_1 and G'_2, respectively, with $\varphi|_{V'_1}(a') = \varphi|_{V'_2}(a') = \text{red}$ and $d_{\varphi|_{V'_1}} + d_{\varphi|_{V'_2}} = (r, b)$. Hence $D_{G'_1+G'_2} \subseteq D_{G'_1} + D_{G'_2}$.

On the other hand, if $(r, b) \in D_{G'_1} + D_{G'_2}$, then there are colorings φ_1, φ_2 of G'_1 and G'_2, respectively, with $d_{\varphi_1} = (r_1, b_1)$, $d_{\varphi_2} = (r_2, b_2)$, $(r_1+r_2, b_1+b_2) = (r, b)$, and $\varphi_1(a') = \varphi_2(a') = \text{red}$. Clearly, $\varphi' := \varphi_1 \cup \varphi_2$ is a coloring of $G'_1 + G'_2$ with $\varphi'(a') = \text{red}$ and $d_{\varphi'} = (r, b)$, thus $D_{G'_1} + D_{G'_2} \subseteq D_{G'_1+G'_2}$. □

As an easy consequence we observe the following fact.

Corollary 3.1 *For all* $v \in V$,

$$D_{T_v} = \sum_{v' \in N(v)} D_{v+T_{v'}} = \sum_{v' \in N(v)} v + D_{T_{v'}},$$

where the sums are defined inductively by (OP1) and (OP2).

Now the algorithm for computing $l(G)$ is straightforward:

1. Let level $:= \max\{\text{dist}_v \mid v \in V\} - 1$, $D_{T_{v'}} := \{(0,0)\}$ for all $v' \in V$ with $\text{dist}_{v'} = \text{level} + 1$.
2. For all $v \in V$ with $\text{dist}_v = \text{level}$:
   let $D_{T_v} := \begin{cases} \{(0,0)\}, & \text{if } v \text{ is a leaf,} \\ \sum_{v' \in N(v)} (v + D_{T_{v'}}) & \text{otherwise.} \end{cases}$
3. Set level $:=$ level $- 1$.
4. If level ≥ 0 then go to 2.
5. Output $\min\{\max\{r,b\} \mid (r,b) \in D_{T_a}\}$.

Note that the time required for operation (OP1) is bounded by $2|D_{G'}| = O(n^2)$, since we have to consider each $(r,b) \in D_{G'}$ twice, and (r,b) takes at most n^2 values. Operation (OP2) consists of $|D_{G'_1}| \cdot |D_{G'_2}| = O(n^4)$ steps. The running time of the algorithm is dominated by the iterated calls of line 2, i.e., by the computations of D_{T_v}. Computing D_{T_v} involves $\deg(v)$ operations of type (OP2), where each summand is computed via a type (OP1) operation. Hence, the overall running time is bounded by $\sum_{v \in V} \deg(v) \cdot O(n^4 + n^2) = O(n^5)$. However, we can reduce the running time to $O(n^3)$ by neglecting "irrelevant" colorings. Note that, if (r, b_1) and $(r, b_2) \in D_{T_v}$ are possible load distributions for a tree T_v imposed by colorings φ_1 and φ_2, then the load distribution with larger second component, say (r, b_2), will be irrelevant for computing $l(G)$ (suppose, φ is an optimal coloring of G with $\varphi_{|T_v} = \varphi_2$, then replacing φ on T_v by φ_1 will not increase the load). Thus, for each r we have to store only $b := \min\{b' \mid (r,b') \in D_{T_v}\}$. Defining the set of *relevant load distributions*

$$\hat{D}_{G'} := \{(r,b) \mid (r,b) \in D_{G'}, b = \min\{b' \mid (r,b') \in D_{G'}\}\}$$

for each subtree G' of G, we have that $|\hat{D}_{G'}| = O(n)$. Obviously, $\hat{D}_G$ can be computed iteratively via operations similar to (OP1) and (OP2) that are performed on $\hat{D}_{G'}$ instead of $D_{G'}$ and thus require only $O(n)$ and $O(n^2)$ steps, respectively. This yields the desired $O(n^3)$ bound. The iterative procedure for computing D_G (or $\hat{D}_G$) can be easily modified such that it gives not only the optimal load, but also an optimal coloring. To this end we record for each $v \in V$ and each $(r,b) \in D_{T_v}$, the summands from the distributions of the neighboring subtrees that add up to (r,b) together with the information whether or not a swap of colors occurred. This information is stored as a set

$p_v(r,b)$ of 4-tuples (r',b',i,v'), where v' is a successor of v and i is 0 or 1 depending on whether $(r',b') \in D_{T_{v'}}$ or $(b',r') \in D_{T_{v'}}$. More precisely, we alter step 2 of the previous algorithm in the following way.

2. For all $v \in V$ with $\mathrm{dist}_v = $ level do begin
2.1 if v is a leaf then set $D_{T_v} := \{(0,0)\}$
2.2 else begin
  2.2.1 Let $v_1, \ldots, v_\nu$ be the neighbors of v.
  2.2.2 For all $(r,b) \in D_{T_{v_1}}$ define $p_v(r+1,b) := \{(r,b,0,v_1)\}$ and $p_v(b+1,r+1) := \{(r,b,1,v_1)\}$. Let $D_1 := \{(r+1,b),(b+1,r+1) \mid (r,b) \in D_{T_{v_1}}\}$.
  2.2.3 For $i := 2$ to ν do begin
    2.2.3.1 $D_i := \emptyset$.
    2.2.3.2 For all $(r,b) \in D_{i-1}$ and all $(r',b') \in D_{T_{v_i}}$ do begin
      2.2.3.2.1 if $(r+r'+1, b+b') \notin D_i$ then include it into D_i and define $p_v(r+r'+1, b+b') := p_v(r,b) \cup \{(r',b',0,v_i)\}$.
      2.2.3.2.2 If $(r+b'+1, b+r'+1) \notin D_i$ then include it into D_i and define $p_v(r+b'+1, b+r'+1) := p_v(r,b) \cup \{(r',b',1,v_i)\}$.
    2.2.3.3 end
  2.2.4 Let $D_{T_v} := D_\nu$.
2.3 end

Starting from an optimal load distribution $d = (r_0, b_0)$ we trace back the load computations via p and determine for each node an optimal color with the following algorithm.

1. Define $\varphi(a) :=$ red, $v := a$, $d := (r_0, b_0)$, $M := \emptyset$.
2. Set $M := M \cup p_v(d)$.
3. If $M = \emptyset$ then output φ and stop.
4. Let $(r',b',i,v') \in M$, set $M := M \setminus \{(r',b',i,v')\}$.
5. Define $\varphi(v') := \begin{cases} \varphi(v) & \text{if } i = 0 \\ \{\text{red,blue}\} \setminus \varphi(v) & \text{otherwise.} \end{cases}$
6. Set $v := v'$, $d := (r',b')$, and go to 2.

We assume that $p_v(0,0) = \emptyset$ for all $v \in V$. Note that $\varphi(v')$ does not depend on the order in which we select elements from M. The above algorithm can be implemented to run in $O(n)$ time. Thus the time required to solve the MLCP on trees with n vertices is $O(n^3)$. This ends the proof of Theorem 3.2. □

4 An Approximation Algorithm for Graphs with Genus g

In this section, we show how a $(1 + o(1))$-approximate solution for the MLCP for graphs of genus $g > 0$ can be computed if $\Delta = o(\frac{m^2}{ng})$. Recall that the genus of a graph is the smallest integer g such that the graph can be drawn without crossing itself on a sphere with g "handles". The problem of determining the genus of a graph is NP-hard [14]. A trivial upper bound on the genus g of a graph with m edges and n vertices is $m - 1$ since each crossing of two edges can be eliminated by introducing a handle. A lower bound of $g \geq \frac{m-3n}{6} + 1$ can be obtained by generalizing Euler's formula for planar graphs (see [15]). The main idea of our algorithm is to partition V into two sets A and B such that

- the number of edges having both endpoints in A is at most $m/2$,
- the same holds for B,
- there are only $O(\sqrt{g\Delta n})$ edges between the sets A and B.

By coloring A and B with different colors, we obtain a coloring φ with $l_\varphi(G) \leq 1/2 + c\sqrt{g\Delta n}/m$. Since $l(G) \geq 1/2$, for $\Delta = o(\frac{m^2}{gn})$ we have a $(1 + o(1))$-approximate solution. A polynomial time algorithm finding a partition with $O(\sqrt{n})$-vertex separator for planar graphs ($g = 0$) was described in [10] and then extended for graphs of genus $g > 0$ in [6]. Let $E(A)$, $E(B)$, and $E(A, B)$ denote the sets of monochromatic edges in A, B, and the set of bichromatic edges connecting A and B, respectively. Let us define a weighted graph (G, w) as a graph $G = (V, E)$ equipped with a mapping $w : V \to \mathbb{R}_{\geq 0}$ assigning each vertex a nonnegative weight. We extend the notion of weight to arbitrary subsets V' of V in an obvious way as $w(V') = \sum_{v \in V'} w(v)$. For our purpose we use the following theorem, given in [13].

Theorem 4.1 *[13] Let G be a graph of genus $g > 0$, having nonnegative vertex weights summing to one such that no weight exceeds $2/3$. There is a partition of V into sets A and B, such that weight(A) $\leq 2/3$, weight(B) $\leq 2/3$, and $|E(A, B)| \leq 5\sqrt{3g\Delta n}$. Provided that we are given an embedding of G into its genus surface, there is an $O(n+g)$-time algorithm which finds such a partition.*

By assigning to each vertex of G a uniform weight of $1/n$, we obtain the following corollary.

Corollary 4.1 *Let G be a graph of genus $g > 0$. Then the vertices of G can be partitioned into two sets A and B, such that neither A nor B contains more than $2n/3$ vertices and $|E(A, B)| \leq 5\sqrt{3g\Delta n}$. Provided that we are given an embedding of G into its genus surface, there is an algorithm which finds such a partition in time $O(n + g)$.*

Theorem 4.1 can be applied in the following way: for any graph of bounded genus $g > 0$, we assign to each vertex $v \in V$ a weight $w(v) = \frac{\deg(v)}{2m}$. The theorem yields a partition of V into A and B, such that $|E(A)| \leq \frac{2}{3}m$, $|E(B)| \leq \frac{2}{3}m$, and there are at most $5\sqrt{3g\Delta n}$ edges between A and B. The factor $\frac{2}{3}$ can be reduced to $\frac{1}{2}$ by iterating the algorithm. We show this in the proof of the following Theorem 4.2. Let us remark that although it is tempting to assign the weights $w(v) = \frac{\deg(v)}{2m}$ to vertices $v \in V$, the separation of the graph is done as in Corollary 4.1 with respect to uniform weights.

Theorem 4.2 *Let G be a graph of genus $g > 0$. There is a partition of V into sets A, B, such that $|E(A)| \leq |E(B)| \leq \frac{1}{2}m + 48\sqrt{g\Delta n}$, and $|E(A,B)| \leq 48\sqrt{g\Delta n}$. Provided that we are given an embedding of G into its genus surface, there is an algorithm which finds such a partition in time $O(n + g \log n)$.*

Proof. Let $G = (V, E)$ be an n-vertex graph with m edges of genus $g > 0$. Let us define sequences of sets (A_i), (B_i) and (C_i) and a sequence of sets of edges (W_i) such that:

(i) A_i, B_i and C_i partition V,
(ii) $|E(A_i)| \leq |E(B_i)|$ and $|E(B_i) \setminus W_i| \leq \frac{1}{2}|E|$,
(iii) in the i-th step at most $5\sqrt{3g\Delta|C_{i-1}|}$ edges are added to W_i,
(iv) $E(A_i, B_i) \subseteq W_i$,
(v) $|C_i| \leq 2/3|C_{i-1}|$.

Let $A_0 = B_0 = \emptyset$, $C_0 = V$, $W_0 = \emptyset$. Then (i), (ii), and (iv) trivially hold. If $A_{i-1}, B_{i-1}, C_{i-1}$, and W_{i-1} have been defined, and $C_{i-1} \neq \emptyset$, let G^* be the subgraph of G induced by the vertex set C_{i-1}. Let A^*, B^* be a partition of G^* according to Corollary 4.1. The new "cut" edges $E(A^*, B^*)$ together with W_{i-1} form the new set W_i, i.e. $W_i := W_{i-1} \cup E(A^*, B^*)$. Without loss of generality we shall assume that $|E(A^*)| \leq |E(B^*)|$. If $|E(A_{i-1} \cup A^*)| \leq |E(B_{i-1}|$, we put $A_i := A_{i-1} \cup A^*$ and $B_i := B_{i-1}$, otherwise we put $A_i := B_{i-1}$ and $B_i := A_{i-1} \cup A^*$. It is easy to check that (i)–(v) hold for our constructed sequences $(A_i), (B_i), (C_i)$, and (W_i).
Let k be the smallest index for which $C_k = \emptyset$. Let $A = A_k$, $B = B_k$. By (i), A, B is a partition of V. By (iv), the number $|E(A, B)|$ is bounded by $|W_k|$, and by (iii) and (v), this is bounded by

$$\sum_{i=0}^{\infty} 5\sqrt{3g\Delta n}\left(\frac{2}{3}\right)^{i/2} \leq 48\sqrt{g\Delta n}.$$

By (ii),

$$|E(A)| \leq |E(B)| \leq \frac{1}{2}|E| + |W_k| \leq \frac{1}{2}m + 48\sqrt{g\Delta n}.$$

Since the algorithm performs at most $k = O(\log n)$ iterations, the running time is bounded by $O(|C_{i-1}| + g) \leq c_1 \left(\left(\frac{2}{3}\right)^i n + g\right)$, the total running time is bounded by

$$\sum_{i=0}^{c\log n} c_1 \left(\left(\frac{2}{3}\right)^i n + g \right) = O(n + g\log n).$$

□

Corollary 4.2 *Let G be any graph of genus $g > 0$. Given an embedding of G into its genus surface, a coloring φ with $l_\varphi(G) \leq 1/2 + 96\sqrt{g\Delta n}/m$ can be constructed in time $O(n + g\log n)$.*

For a planar graph G, we can similarly use the following separator theorem from [8] to show that a coloring φ with $l_\varphi(G) \leq \frac{1}{2} + 1.58\sqrt{d_1^2 + \ldots + d_n^2}/m$ can be constructed in time $O(n^2 \cdot \alpha(n,n))$, where $d_1, \ldots, d_n$ is the degree sequence of G and $\alpha(n,n)$ is the inverse of Ackerman's function.

Theorem 4.3 (Theorem 1.2 of [8]) *A planar embedded graph G has a weighted-simple-cycle separator of size $\leq 1.58\sqrt{d_1^2 + \ldots + d_n^2}$. The separator is computable in $O(n\alpha(n,n))$ sequential time.*

5 Randomized Approximation

5.1 Approximation for General Graphs

In this section, we study the MLCP on arbitrary graphs. Since the problem is NP–hard, approximate solutions are the best one can expect to find efficiently. We first analyze the load of random colorings. With high probability, their load is less than $\frac{3}{4} + O(\sqrt{\Delta/m})$. This shows existence of such colorings, and also yields a randomized algorithm which can be derandomized via the standard method of conditional probabilities [2]. Since $\frac{1}{2}$ is a trivial lower bound for l_φ, these results yield a $(1.5 + o(1))$–approximation algorithm if $\Delta = o(m)$.

We analyze random colorings with Chebychev's inequality.

Lemma 5.1 (Chebychev's inequality) *Let $(\Omega, \mathbb{P})$ be a discrete probability space and $X : \Omega \to \mathbb{R}$ a random variable with finite variance. Then for every $\varepsilon > 0$,*

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq \varepsilon) \leq \frac{\mathrm{Var}(X)}{\varepsilon^2}.$$

Theorem 5.1 *There is a coloring φ such that $l_\varphi \leq \frac{3}{4} + \sqrt{\left(\frac{\Delta}{4} + \frac{3}{8}\right)/m}$.*

Proof. Let $\varphi : V \to \{\text{red,blue}\}$ be a random coloring such that $P(\varphi(v) = \text{red}) = \frac{1}{2} = \mathbb{P}(\varphi(v) = \text{blue})$ independently for all $v \in V$. For each $v \in V$, let

$$X_v := \begin{cases} 1, & \text{if } \varphi(v) = \text{red} \\ 0 & \text{otherwise} \end{cases}$$

be a random variable indicating if v is colored red. Then, $\{X_v \mid v \in V\}$ is a set of independent random variables satisfying $r_\varphi = \sum_{\{v,w\}\in E}(X_v + X_w - x_v X_w)$, and we can calculate the expectation

$$\mathbb{E}(r_f) = \sum_{\{v,w\}\in E} (\mathbb{E}(X_v) + \mathbb{E}(X_w) - \mathbb{E}(X_v) \cdot \mathbb{E}(X_w)) = \frac{3}{4}m.$$

For $e = \{v, w\} \in E$, let

$$Y_e := (X_v + X_w - X_v X_w),$$

then

$$\mathbb{E}(r_\varphi^2) = \mathbb{E}((\sum_{e\in E} Y_e)^2) = \mathbb{E}(\sum_{e\in E} Y_e^2) + 2\mathbb{E}(\sum_{\substack{e,e'\in E \\ |e\cap e'|=1}} Y_e \cdot Y_{e'}) + 2\mathbb{E}(\sum_{\substack{e,e'\in E \\ |e\cap e'|=0}} Y_e \cdot Y_{e'}).$$

We consider the three expectation terms in the above sum separately.

$$\begin{aligned}
\mathbb{E}(\sum_{e\in E} Y_e^2) &= \sum_{e\in E} E(Y_e^2) \\
&= \sum_{\{v,w\}\in E} \mathbb{E}(X_v^2 + X_w^2 + X_v^2 X_w^2 + 2X_v X_w - 2X_v^2 X_w - 2X_v X_w^2) \\
&= \sum_{\{v,w\}\in E} \mathbb{E}(X_v + X_w + X_v X_w + 2X_v X_w - 2X_v X_w - 2X_v X_w) \\
&= \sum_{\{v,w\}\in E} \mathbb{E}(X_v + X_w - X_v X_w) = \mathbb{E}(r_\varphi) = \frac{3}{4}m.
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}(\sum_{\substack{e,e'\in E \\ |e\cap e'|=1}} Y_e Y_{e'}) &= \sum_{\{u,v\},\{v,w\}\in E} \mathbb{E}((X_u + X_v - X_u X_v)(X_v + X_w - X_v X_w)) \\
&= \sum_{\{u,v\},\{v,w\}\in E} \mathbb{E}(X_u X_v + X_u X_w - X_u X_v X_w + X_v + X_v X_w \\
&\qquad - X_v X_w - X_u X_v - X_u X_v X_w + X_u X_v X_w) \\
&= \sum_{\{u,v\},\{v,w\}\in E} \mathbb{E}(X_v) + \mathbb{E}(X_u X_w) - \mathbb{E}(X_u X_v X_w) = \lambda \cdot \frac{5}{8},
\end{aligned}$$

where λ denotes the number of two element sets $\{e, e'\}$ such that $|e \cap e'| = 1$. Each fixed edge e is incident with at most 2Δ edges e'. Summing over E, we count each two element set $\{e, e'\}$ twice, hence $\lambda \leq m\Delta$. Finally,

$$
\begin{aligned}
\mathbb{E}\Big(\sum_{\substack{e,e' \in E \\ |e \cap e'|=0}} Y_e Y_{e'}\Big) &= \sum_{\{t,u\},\{v,w\} \in E} \mathbb{E}(X_t + X_u - X_t X_u)(X_v + X_w - X_v X_w) \\
&= \sum \mathbb{E}(X_t X_v + X_t X_w - X_t X_v X_w + X_u X_v + X_u X_w \\
&\quad - X_u X_v X_w - X_t X_u X_v - X_t X_u X_w + X_t X_u X_v X_w) \\
&= \left(\binom{m}{2} - \lambda\right)(1 - 1/2 + 1/16) = \frac{9}{16}\left(\binom{m}{2} - \lambda\right).
\end{aligned}
$$

This yields the estimate

$$
\mathbb{E}(r_\varphi^2) = \frac{9}{16}m^2 + \frac{3}{16}m + \frac{1}{8}\lambda \leq \frac{9}{16}m^2 + \frac{3}{16}m + \frac{m\Delta}{8},
$$

giving

$$
\operatorname{Var}(r_\varphi) = \mathbb{E}(r_\varphi^2) - \mathbb{E}(r_\varphi)^2 \leq m\left(\frac{\Delta}{8} + \frac{3}{16}\right).
$$

By Chebychev's inequality,

$$
\mathbb{P}\left(r_\varphi - \frac{3}{4}m > \varepsilon\right) < \mathbb{P}\left(|r_\varphi - \frac{3}{4}m| > \varepsilon\right) \leq \frac{m(\frac{\Delta}{8} + \frac{3}{16})}{\varepsilon^2} = \frac{1}{2}
$$

for $\varepsilon = \sqrt{m(\frac{\Delta}{4} + \frac{3}{8})}$.

Similarly, $\mathbb{P}(b_\varphi - \frac{3}{4}m > \varepsilon) < \frac{1}{2}$, thus with positive probability both r_φ and b_φ are at most $\frac{3}{4} + \sqrt{m(\frac{\Delta}{4} + \frac{3}{8})}$. In particular, a coloring with $l_\varphi \leq \frac{3}{4} + \sqrt{(\frac{\Delta}{4} + \frac{3}{8})/m}$ exists. □

Note that the dependence on Δ cannot be avoided. This is shown by star graphs. Moreover, if $\Delta = o(m)$, then the bound of $\frac{3}{4} + o(1)$ cannot be improved in general. The complete graph $K_n = (\{1, \ldots, n\}, \binom{\{1,\ldots,n\}}{2})$ satisfies $l_\varphi \geq (\frac{3}{8}n^2 - \frac{1}{4}n)/m = \frac{3}{4} + o(1)$ for all colorings φ.

5.2 Random Graphs

In fact, in some sense almost all graphs have a load of $\frac{3}{4} - o(1)$. We prove the following result.

Theorem 5.2 *For a random graph $G = (V, E)$, $|V| = n$ obtained by choosing a random set E of m edges (without repetition), we have $l(G) \geq \frac{3}{4} - \sqrt{n/m}$ with probability greater than $1 - 2^{-n}$.*

In other words, all but a fraction of less than 2^{-n} of the graphs having n vertices and m edges have a load of at least $\frac{3}{4} - \sqrt{n/m}$. If $n = o(m)$, this shows that almost all graphs have a load of $\frac{3}{4} - o(1)$.

Proof of Theorem 5.2. Let $G = (V, E)$ be the random graph described in the theorem, that is, E is a random element from $\{E \subseteq \binom{V}{2} \mid |E| = m\}$. Fix any two-coloring χ of V. W.l.o.g. we may assume that at least $n/2$ vertices are colored red. Let $B = \{e \in \binom{V}{2} \mid \chi(e) = \{\text{blue}\}\}$, the set of all possible edges that are monochromatic blue. Clearly, $|B| \leq \frac{1}{4}\binom{n}{2}$. Note that the hypergeometric distribution admits the usual Chernoff bounds for independent random variables (cf. e.g. Theorem 2.10 in Janson, Łuczak and Rucinski [9]). Hence with $\lambda = \sqrt{mn}$, the number $N = |B \cap E|$ of monochromatic blue edges satisfies

$$\begin{aligned}\mathbb{P}(N \geq \tfrac{m}{4} + \lambda) &\leq \mathbb{P}\left(N \geq m \cdot |B| / \binom{n}{2} + \lambda\right) \\ &= \mathbb{P}(N - \mathbb{E}[N] \geq \lambda) \\ &\leq \exp(-2\lambda^2/m) = \exp(-2n).\end{aligned}$$

Hence for any fixed coloring, the probability that our random graph has load at most $\frac{3}{4} - \lambda/m$ is less than $\exp(-2n)$. We conclude that the probability that there is a 2–coloring achieving a load of at most $\frac{3}{4} - \lambda/m$, is less than 2^n (the number of colorings) times this value, that is, $2^n \exp(-2n) = (2/e^2)^n \leq 2^{-n}$. This proves the claim. □

6 MLCP with More than Two Colors

Most of our results have a natural extension to the MLCP with more than two colors. We believe that the following conjectures can be proven in a way that similar to (but technically more involved than) the two-color case.

- For any fixed number of colors, the MLCP is NP–complete.
- For any fixed number of colors, there is a polynomial time algorithm computing a minimal load coloring for trees.
- A tree G with m edges can be colored in k colors with load bounded by $\frac{1}{k} + O((\Delta/m) \log m)$.

- For all graphs $G = (V, E)$ there is a k-coloring with load at most $\frac{2k-1}{k^2} + O\left(\sqrt{\Delta/m}\right)$.
- For graphs on n vertices with genus $g > 0$ we can find a k-coloring with load bounded by $1/k + O(\sqrt{g\Delta n}/m)$.

There are graphs having small load in some numbers of colors and large one in others. We give three examples.

(i) Let G be a graph consisting of two disjoint cliques on n vertices. Then the load in two colors is $\frac{1}{2}$, shown by coloring both cliques monochromatic in a different color. This is smallest possible for any graph. Let $\gamma = \sqrt{3} - 1$. In three colors, an optimal coloring will contain $(\gamma + o(1))n$ red vertices in the first clique, $(\gamma + o(1))n$ blue vertices in the second and $(1 - \gamma + o(1))n$ green vertices in each clique. This yields a load of $(2\sqrt{3} - 3 + o(1))n^2/m \approx 0.4641$. Compared to the smallest possible value of $\frac{1}{3}$, this is quite large.

(ii) If G consists of three disjoint cliques of n vertices each, then the 3–color load is smallest possible with $\frac{1}{3}$, but the 2–color load is approximately $\frac{7}{12}$.

(iii) The same behavior is also displayed by trees. A complete 3–ary tree T has a 3–color load of $\frac{1}{3} + 2/m$. However, we proved it to have a 2–color load of $\frac{1}{2} + \Omega(\log(n)/m)$, which is (up to the implicit constant) maximum possible for trees as shown in Theorem 3.1.

7 Acknowledgments

We thank the anonymous referees for their very helpful detailed comments on a preliminary version of this paper.

The second author was supported by DFG grant SR 7/9 – 3.

References

[1] A.A. Ageev, A.V. Fishkin, A.V. Kononov and S.V. Sevastianov, *Open Block Scheduling in Optical Communication Networks*. Springer LNCS 2909 (2004), 13 - 26.

[2] N. Alon and J. Spencer, The Probabilistic Method, Wiley Interscience (1992).

[3] I. Baldine and G. N. Rouskas, *Reconfiguration and Dynamic Load Balancing in Broadcast WDM Networks*. Photonic Network Communications Journal 1(1) (1999), 49 - 64.

[4] E. Bampis and G. N. Rouskas, *The Scheduling and Wavelength Assignment Problem in Optical WDM Networks.* IEEE/OSA Journal of Lightwave Technology 20(5) (2002), 782 - 789.

[5] P. Berman and M. Karpinski, *Approximation Hardness of Bounded Degree MIN-CSP and MIN-BISECTION*, Electronic Colloquium on Computational Complexity, Report No. 26 (2001).

[6] H.N. Djidjev, *A separator theorem.* Comptes Rendus de l'Academie Bulgare des Sciences 34 (1981), 643 - 645.

[7] M.R. Garey, D.S. Johnson and L. Stockmeyer, *Some simplified NP-complete graph problems*, Theoret. Comput. Sci. 1(33) (1976), 237 - 267.

[8] H. Gazit and G.M. Miller, *Planar Separators and the Euclidean Norm*, In Proc. of the SIGAL Internat. Symp. on Algorithms (1990), LNCS 450, Springer, 338 - 347.

[9] S. Janson, T. Łuczak, and A. Rucinski, *Random Graphs*, Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York (2000).

[10] R. J. Lipton and R.E. Tarjan, *A separator theorem for planar graphs*, SIAM Journal on Applied Mathematics 36 (1979), 177 - 189.

[11] G. N: Rouskas and D. Thaker, *Multi-destination communication in broadcast WDM networks: a survey.* Optical Networks, 3(1) (2002), 34 - 44.

[12] A.D. Scott, *Judicious partitions and related problems*, Surveys in Combinatorics (2005), 95 -117.

[13] O. Sýkora and I. Vrťo, *Edge Separators for Graphs of Bounded Genus with Applications*, In Proc. of the 17th International Workshop on Graph Theoretic Concepts in Computer Science (1992), 159 - 168.

[14] C. Thomassen, *The graph genus problem is NP-complete*, Journal of Algorithms 10(4) (1989), 568 - 576.

[15] D.B. West, Introduction to Graph Theory, Prentice Hall (1996).