Coloring Graphs to Minimize Load

- Extended Abstract -

Nitin Ahuja[*] Andreas Baltz[**] Benjamin Doerr [‡]
Aleš Přívětivý [§] Anand Srivastav[**]

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*, $l_\varphi := \max\{r_\varphi, b_\varphi\}$, is minimized. The problem has applications in broadcast WDM communication networks (Ageev et al., 2004). 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 $m/2 + \Delta \log_2 n$. For graphs with genus $g > 0$, we show that a coloring with load $\mathrm{OPT}(1 + o(1))$ can be computed in $O(n + g)$-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 $\frac{3}{4}m + O(\sqrt{\Delta m})$ can be found in deterministic polynomial time using a derandomized version of Azuma's martingale inequality. This bound describes the "typical" situation: in the random multi-graph model we prove that for almost all graphs, the optimal load is at least $\frac{3}{4}m - \sqrt{3mn}$. Finally, we generalize our results to k–colorings for $k > 2$.

1 Introduction

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

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

the maximum number of edges with at least one end-point in color i, where the maximum is taken over all $i \in \{1, \ldots, k\}$. The problem of minimizing

[*]Department of Mathematical Optimization, Technical University Braunschweig, Pockelsstrasse 14, D-38106 Braunschweig, Germany, `n.ahuja@tu-bs.de`

[**]Department of Computer Science, Christian-Albrechts-University Kiel, Christian-Albrechts-Platz 4, D-24098 Kiel, Germany, `{aba,asr}@numerik.uni-kiel.de`

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

[§]Department of Applied Mathematics, Charles University, Malostranské nám. 25, 11800 Praha, Czech Republic, `privetivy@kam.mff.cuni.cz`

this load arises naturally in wavelength division multiplexing (WDM) networks with broadcast traffic: here, the nodes represent senders/receivers each of which wants to send messages to every other node via one of k available wavelength channels. The objective is to assign to each node a channel, such that the maximum traffic taken over all channels is minimized. Ageev et al. [1] consider scheduling aspects of the capacitated weighted version of this problem. Closely related is the *k–balanced graph partitioning problem* [6], where the aim is to find a set of edges of minimum capacity such that removing these edges partitions the graph into at most k roughly equally weighted and connected subgraphs. 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, $l_\varphi := \max\{r_\varphi, b_\varphi\}$, 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).*

Our Results

After some preliminaries including the establishment of NP-hardness of the problem in Section 2, we show how to solve MLCP on trees optimally in $O(n^3)$ time (Section 3). Such an optimal solution is proven to have a load of at most $\frac{1}{2}m+\Delta \log_2 n$. Section 4 is concerned with graphs of genus $g > 0$. With a separator theorem proved with techniques from Djidjev [5] we obtain an $O(n+g)$-time algorithm for constructing a coloring with load bounded by $m/2 + 48\sqrt{g\Delta n}$. 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 $\frac{3}{4}m + O(\sqrt{\Delta m})$ with high probability. This immediately yields a randomized algorithm. Furthermore, using an algorithmic version of the Azuma-inequality, we derive a deterministic $O(n^3)$–time algorithm for computing colorings with the same load-bound. This is quite strong: in the random multi-graph model (and similarly in other random models), almost all graphs have no coloring with load less than $\frac{3}{4}m - \sqrt{3mn}$. In the last section we extend 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 denote the number of "cut edges" that connect a red vertex with a blue vertex by c_φ. We will refer to the set of red vertices as V_r and to the set of blue vertices as V_b.

Since every edge of G is counted as red or blue (or both), $l(G) \geq \frac{m}{2}$. Obviously,

every red-blue coloring of G has load at most m, so we see that each two-coloring of a graph G is a 2-approximation of $l(G)$.
Let G be a star with $d+1$ vertices, then $l(G) = d$. In fact, the maximum degree Δ of the input graph is another lower bound on $l(G)$. It is also easy to find an optimal two-coloring of cycles and chains (graphs consisting of a single open path). 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, MLCP is equivalent to MinBisection.

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 $r_\varphi = \frac{|V_r|\cdot k}{2} + \frac{c_\varphi}{2}$, and the number of blue edges is $b_\varphi = \frac{|V_b|\cdot k}{2} + \frac{c_\varphi}{2}$, 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, 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

$$l_\varphi = \frac{n}{2} \cdot \frac{k}{2} + \frac{c_\varphi}{2}.$$

Since φ is optimal, c_φ, 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

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

which is obviously optimal. Hence MLCP and MinBisection are equivalent on regular graphs. For $k \geq 3$, MinBisction on k-regular graphs is as hard as general MinBisection (see [3]). Since the decision version of MinBisection is NP-complete [7] and the load of any proposed solution for MLCP can be evaluated in polynomial time, we have established NP-completeness also for MLCP.

Theorem 2.1 *The decision version of 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 $l(G) \leq \frac{n-1}{2} + \Delta \log_2 n$. 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 $G - v$ has $k \geq 2$ components C_i, $i \in \{1, \ldots, k\}$, where the number of vertices n_i in component C_i is at most $n/2$. 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 $\sum_{i \in I_1} (n_i - 1) + m_1' - \Delta \log n_j \geq m_1 - |I_1| - \Delta \log_2 (n/2)$, which is at least $m_1 + 1 - \Delta \log_2 n$. Similarly, the total number of monochromatic blue edges is at least $m_2 + 1 - \Delta \log_2 n$. $\square$

We did not try to optimize the error term $\Delta \log_2 n$. 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 (proof omitted). This example also demonstrates that in an optimal coloring the color classes may induce disconnected subgraphs. 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 $l(G) \leq \frac{n-1}{2} + \Delta \log_2 n$.*

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

Theorem 3.2 *On trees with n vertices, 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' (the assumption $\varphi(a') = \text{red}$ will cause no 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:

(i) 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

$$v + D_{G'} := \{(r+1, b) \mid (r, b) \in D_{G'}\} \cup \{(b+1, r+1) \mid (r, b) \in D_{G'}\} \quad (3.1)$$

(ii) 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

$$D_{G'_1} + D_{G'_2} := \{(r_1 + r_2, b_1 + b_2) \mid (r_1, b_1) \in D_{G'_1}, (r_2, b_2) \in D_{G'_2}\}. \quad (3.2)$$

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 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 $\varphi(v) = \text{red}$. Then $\varphi' := \varphi|_{V'}$ 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 $v \in V$,*

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

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

1. Let level $:= \max\{\text{dist}_v \mid v \in V\} - 1$, $D_{T_{v'}} := \{(1, 0)\}$ for all $v' \in V$ with $\text{dist}_{v'} = \text{level} + 1$.
2. For all $v \in V$ with $\text{dist}_v = \text{level}$: compute $D_{T_v} = \sum_{v' \in N(v)} v + D_{T_{v'}}$.
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 (3.1) 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 (3.2) 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 (3.2), where each summand is computed via a type (3.1) 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 (3.1) and (3.2) 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. All we have to do is store, for each $(r, b) \in \hat{D}_{T_v}$ and each $v' \in N(v)$ a pair $((r', b'), i) =: p_{v'}(r, b)$, where $(r', b') \in \hat{D}_{T_{v'}}$ was used in the computation of (r, b) and $i \in \{1, 0\}$ indicates whether of not in computing (r, b) from (r', b') we swapped the colors of $T_{v'}$. 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 $M := M \cup \{(v, v', p_{v'}(d)) \mid v' \in N(v)\}$.
3. If $M = \emptyset$ then output φ and stop.
4. Let $(v, v', ((r', b'), i)) \in M$, set $M := M \setminus (v, v', ((r', b'), i))$.
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.

This algorithm can be implemented to run in $O(n)$ time. Thus the time required to solve MLCP on trees with n vertices is $O(n^3)$ in total. 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 [12]. 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 [13]). 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 $l_\varphi(G) \leq m/2 + c\sqrt{g\Delta n}$. Since $l(G) \geq m/2$, for $\Delta = o(\frac{m^2}{gn})$ we have a $(1+o(1))$-approximate solution. A polynomial time algorithm finding a partition with small vertex separator for planar graphs ($g = 0$) was described in [8, 4] and then extended for graphs of genus $g > 0$ in [5]. 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. For our purpose we use the following theorem, given in [11].

Theorem 4.1 *[11] Let G be a graph of genus $g > 0$, having nonnegative vertex weights summing to no more than one. 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.*

We can use this theorem in 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. This $\frac{2}{3}$ factor can be reduced to $\frac{1}{2}$ by iterating the algorithm on the bigger of the sets resulting from the partitioning. Both, the size of the edge separator and the running time, increase only by a constant factor. We summarize this in the following theorem. The proof is similar to the proof of Corollary 3 in [8], and thus will be given only in the full version of the paper.

Theorem 4.2 *Let G be a graph of genus $g > 0$. There is a partition of V into sets A, B, such that $|E(A)| \leq \frac{1}{2}m$, $|E(B)| \leq \frac{1}{2}m$, 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 $O(n+g)$.*

Corollary 4.1 *Let G be any graph of genus $g > 0$. Given an embedding of G into its genus surface, a coloring φ with $l_\varphi(G) \leq m/2 + 48\sqrt{g\Delta n}$ can be constructed in time $O(n+g)$.*

For a planar graph G, we can similarly use the separator theorem from [4] to show that a coloring φ with $l_\varphi(G) \leq \frac{m}{2} + (6\sqrt{2}+4\sqrt{3})\sqrt{\Delta n}$ can be constructed in time $O(n)$, provided that an embedding is given.

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 $\frac{3}{4}m + O(\sqrt{\Delta m})$. This shows existence of such colorings, and also yields a randomized algorithm. Using an algorithmic version of the Azuma-inequality, we derive a deterministic algorithm for computing such colorings. Since $\frac{1}{2}m$ is a trivial lower bound for l_φ, these results yield a $(1.5 + o(1))$–approximation algorithm if $\Delta = o(m)$.

To analyze random colorings, we use the following martingale inequality[1] that can be found in McDiarmid [9]. It is an application of the well known inequality of Azuma [2]:

Lemma 5.1 *Let $X_1, \dots, X_n$ be independent random variables taking values in some sets $A_1, \dots, A_n$. Let $f : \prod_{i=1}^n A_i \to \mathbb{R}$ such that $|f(x) - f(y)| \le c_i$ whenever x and y differ only in the ith coordinate. Let $X = (X_1, \dots, X_n)$ and $\mu = \mathbb{E}(f(X))$. Then for any $\lambda \ge 0$,*

$$\mathbb{P}(f(X) - \mu \ge \lambda) \le \exp\Big(- 2\lambda^2 / \sum_{i=1}^n c_i^2 \Big). \tag{5.1}$$

Theorem 5.1 *There is a coloring φ such that $l_\varphi \le \frac{3}{4}m + \sqrt{(\ln 2)\Delta m}$. For all $q \ge 0$, a random coloring satisfies $\mathbb{P}\Big(l_\varphi \ge \frac{3}{4}m + q\sqrt{(\ln 2)\Delta m}\Big) \le 2^{-q^2+1}$.*

Proof. We analyze the behavior of a random coloring. Let $\varphi : V \to \{\text{red}, \text{blue}\}$ such that $\mathbb{P}(\varphi(v) = \text{red}) = \frac{1}{2} = \mathbb{P}(\varphi(v) = \text{blue})$ independently for all $v \in V$. Clearly, if two colorings φ_1, φ_2 differ only in the color of some vertex $v \in V$, then $|r_{\varphi_1} - r_{\varphi_2}| \le \deg(v)$. We compute $\mathbb{E}(r_\varphi) = \sum_{e \in E} \mathbb{P}(\exists v \in e : \varphi(v) = \text{red}) = \frac{3}{4}m$. Since $\sum_{v \in V} \deg(v)^2 \le \sum_{v \in V} \deg(v)\Delta = 2\Delta m$, for $\lambda = \sqrt{(\ln 2)\Delta m}$, we have $\mathbb{P}(r_\varphi > \frac{3}{4}m + \lambda) < \frac{1}{2}$. Thus with positive probability, both r_φ and b_φ are at most $\frac{3}{4}m + \lambda$. In particular, a coloring with $l_\varphi \le \frac{3}{4}m + \lambda$ exists. The second statement follows in a similar way. $\square$

The algorithm behind Theorem 5.1 can be efficiently derandomized.

Theorem 5.2 *A coloring φ such that $l_\varphi \le \frac{3}{4}m + \sqrt{(\ln 4)\Delta m}$ can be constructed in $O(n^3)$ time.*

For the proof we invoke an algorithmic version of Azuma's martingale inequality proved by Srivastav and Stangier [10]. Let $\Omega = \{0,1\}^n$ be a probability space with probability measure $\mathbb{P}$ and let $\varphi : \Omega \to \mathbb{R}$ be a quadratic form. Let $X = (X_1, \dots, X_n)$ be a vector of independent random variables with $X_k \in \{0,1\}$, for all $k \in [n]$. Further, let $\mathbb{P}(X_k = 1) = p$ and $\mathbb{P}(X_k = 0) = 1 - p$ for all k and $p \in (0,1)$. We wish to bound the large deviation probability $\mathbb{P}(|\varphi(X) - \mathbb{E}(\varphi(X))| \ge \lambda)$, for $\lambda > 0$. If f satisfies a Lipschitz condition: $|\varphi(X) - \varphi(X')| \le c_k$ if $X, X' \in \Omega$ differ only in the k-th component, then we can use the bounded difference inequality (5.1).

[1] One advantage of this version is that it can be formulated without introducing the martingale machinery used in its proof.

Theorem 5.3 ([10]) *Let* $\delta \in (0,1)$ *such that* $1-\delta \geq 2\exp\Big(-2\lambda^2/\sum_{i=1}^n c_i^2\Big)$. *Then a vector* $X \in \Omega$ *which satisfies* $|\varphi(X) - \mathbb{E}(\varphi(X))| \leq \lambda$ *can be constructed in* $O\left(n^3 \log(\delta^{-1})\right)$ *time.*

Proof of Theorem 5.2. First we write the objective function l_φ, the load, as the maximum of two quadratic forms describing r_φ and b_φ respectively. We model a two coloring of the vertex set V as a vector $X = (X_1, \ldots, X_n) \in \Omega = \{0,1\}^n$, where for $i \in [n]$, $X_i = 1$ if the vertex i is colored red and $X_i = 0$ if it is colored blue. Let (a_{ij}) be the adjacency matrix of the graph $G = (V, E)$ under consideration. We may identify a two-coloring $\varphi : V \to$ {red,blue} by $X \in \{0,1\}^n$, so for $X \in \{0,1\}^n$ let

$$r(X) = \sum_{i=1}^n \sum_{j=1}^n \frac{a_{ij}\, X_i\, X_j}{2} + \sum_{i=1}^n \sum_{j=1}^n a_{ij}\, X_i\, (1 - X_j)\ ,$$

and

$$b(X) = \sum_{i=1}^n \sum_{j=1}^n \frac{a_{ij}\, (1 - X_i)\, (1 - X_j)}{2} + \sum_{i=1}^n \sum_{j=1}^n a_{ij}\, X_i\, (1 - X_j)\ .$$

Note that $\varphi(i) = X_i$ for all $i \in [n]$. So, $r(X) = r_\varphi$, $b(X) = b_\varphi$ and $l_\varphi = l(X) := \max\{r(X), b(X)\}$. Theorem 5.3 can be extended to cover also the maximum of two quadratic forms, $r(X)$ and $b(X)$, with minor modifications in the proof (the important thing is to be able to compute conditional expectations of the form $\mathbb{E}(f \mid X_1 = a_1, \ldots, X_k = a_k)$). Thus, applying Theorem 5.3 to $l(X)$ with $c_k = deg(v_k)$, $\lambda = \sqrt{(\ln 4)\Delta m}$ and $\delta = 0.5$, we can construct a two-coloring $X \in \{0,1\}^n$ in $O(n^3)$ time that satisfies $l(X) \leq \frac{3}{4}m + \sqrt{(\ln 4)\Delta m}$. □

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

5.2 Random Multi-Graphs

In fact, in some sense almost all graphs have a load of $(\frac{3}{4} - o(1))m$. Without proof, we state the following.

Theorem 5.4 *Let* $m \geq 12n$. *For a random multi-graph* $G = (V, E)$, $|V| = n$ *obtained by choosing* m *edges from* $\binom{V}{2}$ *independently with repetition, we have* $l(G) \geq \frac{3}{4}m - \sqrt{3mn}$ *with probability* $1 - 2^{-n}$.

In other words, all but a fraction of less than 2^{-n} of the multi-graphs having n vertices and m edges have a load of at least $\frac{3}{4}m - \sqrt{3mn}$. If $n = o(m)$, this shows that almost all multi-graphs have a load of $(\frac{3}{4} - o(1))m$. The use of multi-graphs has mainly technical reasons. Unless m is close to $\binom{n}{2}$, most multi-graphs as above have only few multiple edges. Hence the random multi-graph model is close to the standard random graph model $G(n, p(n))$.

6 MLCP with More than Two Colors

Most of our results have a natural extension to MLCP with more than two colors. For reasons of brevity and readability we omit the proofs, which are mostly similar (though more technical) to the ones for two colors.

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

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 $\frac{1}{2}|E(G)|$, 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 $(2\sqrt{3} - 3 + o(1))n^2 \approx 0.4641|E(G)|$. Compared to the smallest possible value of $\frac{1}{3}|E(G)|$, 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 $\frac{1}{3}|E(G)|$, but the 2–color load is approximately $\frac{7}{12}|E(G)|$.

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

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] K. Azuma, *Weighted sums of certain dependent variables.* Tohoku Math. Journal 3(1967), 357 - 367.

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

[4] K. Diks, H.N. Djidjev, O. Sýkora and I. Vrťo, *Edge Separators of Planar and Outerplanar Graphs with Applications*, Journal of Algorithms 14(1993), 258 - 279.

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

[6] G. Even, J. Naor, S. Rao and B. Schieber, *Fast Approximate Graph Partitioning Algorithms*. SIAM J. Comput. 28(6)(1999), 2187 - 2214.

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

[9] C. McDiarmid, *Concentration*. In Probabilistic Methods for Algorithmic Discrete Mathematics, Volume 16 of Algorithms Combin.(1998), Springer, Berlin, 195 - 248.

[10] A. Srivastav and P. Stangier, *On quadratic lattice approximations*. In Proc. of the 4th Internat. Symp. on Algorithms and Computation(1993), LNCS 762, Springer, 176 - 184.

[11] 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.

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

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