Sum Edge Coloring on Multitrees

Stephen Andrews
Chatham High School

Hongyi Hu
Carnegie Mellon University

Atefeh Mohajeri (Mentor)
Nokia Bell Labs

July 25, 2020

Abstract

We study edge coloring for multigraphs, where each edge e has multiplicity $\mu(e)$. In the edge coloring of multigraph, we assign $\mu(e)$ distinct colors to e such that no two adjacent edge share a color in common.

In this paper, we compare the Minimum Edge Chromatic Sum (MECS) problem and the Optimum Cost Chromatic Partition (OCCP) problem. The former aims to find an edge coloring that minimizes the sum of the maximum color on each edge while the latter the sum of all colors used. Despite similarity of the two objectives, they are quite different in complexity. This difference becomes clear when restricted to trees; Marx showed that MECS is NP-hard, while OCCP polynomially solvable by linear programming. We contribute to these results by patching Marx's NP-hardness reduction proof using an additional gadget and making explicit the run-time for solving OCCP.

1 Introduction

Our project is about edge coloring multigraphs. In a multigraph, edges may have parallel copies. We define a *multigraph* as the triplet $G = (V, E, \mu)$, where V and E are the conventional vertex and edge sets of a simple graph equipped with a *multiplicity* function $\mu : E \to \mathbb{N}$ such that $\mu(uv)$ indicates the number of copies of that edge between vertices u and v. Let $\mu(G) := \max_e \mu(e)$ be the maximum multiplicity. Denote *degree* by $\deg(u) := \sum_{v \in N(u)} \mu(uv)$, the number of edges incident to vertex u, and $\Delta(G) := \max_u \deg(u)$ the maximum degree.

When edge coloring a multigraph, we assign $\mu(e)$ distinct colors to edge e. If two edges share a common endpoint, then the two edges cannot not share any color in common. Formally, the notion of an edge coloring on a multigraph goes as follows: a $\psi : E \to 2^{\mathbb{N}}$ must assign each

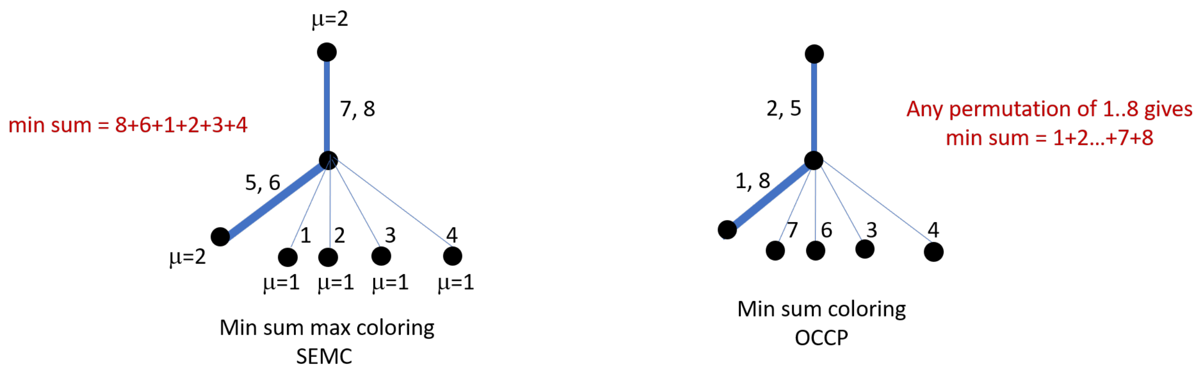

Figure 1: (Left) MECS, minimizing sum of max color. (Right) OCCP, minimizing sum color.

edge e with a *color*, the $\mu(e)$-element subsets of $\mathbb{N}$, such that no two edges have overlapping colors. We call such ψ a *proper* edge coloring.

We consider two objectives in edge coloring. The first objective is to minimize the sum of the max color used on each edge. In the literature, the problem is called *Minimum Edge Chromatic Sum (MECS)*. We study a paper [Mar03] authored by Daniel Marx. The second objective aims to minimize the sum of *all* colors used. In the literature the problem is called *Optimal Cost Chromatic Partition (OCCP)*.

Figure 1 is an example of coloring a star graph. This star has two thick edges with multiplicity 2 and four thin edges with multiplicity 1. So a total of 8 colors are needed. To minimize the sum of max color per edge, we pair up the largest colors 7 and 8 for one edge of multiplicity 2 and pair up 5 and 6 for the other edge of multiplicity 2. This is because for this objective, only the largest color on each edge counts towards the sum. So 8 and 6 count toward the sum, but not 7 and 5. The other 4 edges get 1 through 4. The sum is $8 + 6 + 1 + 2 + 3 + 4 = 24$. To minimize the sum of all colors used, any assignment of 1–8 on these 6 edges gets the minimum sum of $1 + 2 + \cdots + 8 = 36$.

Both objectives are NP-hard to optimize on general graphs. When we restrict to simple graphs, where edge multiplicity is 1, the two objectives become the same. However, it is still NP-hard on general simple graphs since the standard graph coloring problem is NP-complete. When we further restrict to simple trees, this problem can be solved in polynomial time using dynamic programming in [Sal03]. Unfortunately, this algorithm does not generalize to multitrees for either objective.

Our Results. We will see, coloring multitrees lies on the boundary of being polynomially solvable and NP-hard.

MECS, minimization of the sum of the maximum color per edge, is NP-hard. In [Mar03], the author constructs an elaborate reduction from the 3-occurrence 3SAT problem. However, we found missing links in his proof, and we fix this by introducing in section 2 a new gadget.

On the contrary, OCCP, the minimization of all colors used, is not NP-hard. In section **??**, we employ a linear program relaxation to solve OCCP in $O(n^5)$ time.

2 NP-Hardness of MECS

The NP-hardness reduction of MECS is from 3-occurrence 3SAT.

3-occurrence 3SAT. To decide if a 3-CNF boolean formula ϕ has a satisfiable assignment is a well known NP-complete problem. Regardless of whether ϕ has at most 3 or exactly 3 literals per clause, 3SAT is NP-complete. The following is an example of a 3SAT formula that has at most 3 literals per clause:

$$\phi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_3 \vee x_4) \wedge (\bar{x}_3 \vee \bar{x}_4).$$

The term 3-occurrence means each variable appears at most 3 times in ϕ. In the example above, variable x_1 appears twice, x_2 appears once, x_3 appears three times and x_4 appears twice, so this is a 3-occurrence formula.

If there are exactly 3 literals per clause, somewhat surprisingly, *every* 3-occurrence 3SAT formula ϕ is satisfiable. Therefore, this version of 3-occurrence 3SAT is trivial, not NP-hard. On the other hand, if each clause has at most 3 literals, for example 2 or 3 literals, this version of 3-occurrence 3SAT problem is NP-hard [Tov84].

Marx's construction assumes exactly 3 literals per clause, and as we just saw, this is not an NP hard problem. In order to present how we fixed this issue, we first give a high-level summary of Marx's construction.

Overview of Marx's Proof. From any 3-occurrence 3SAT formula ϕ, Marx constructs a tree $T(\phi)$. He created a gadget for each variable (Left, Figure 2) and each clause (Right, Figure 2) of ϕ. Each gadget is a tree and these gadgets share a common root. Therefore, the coloring of the root edge of these gadget trees must be coordinated, and that's the essence of Marx's proof.

Marx proves that ϕ is satisfiable if and only if the tree $T(\phi)$ can be colored with a *minimal* coloring. Informally, minimal coloring around a vertex v means using the smallest colors around v and getting the smallest sum around v. (In the paper, minimal coloring is called *A-good*.) The variable gadget is relatively simple. For variable x_i, the gadget is a star tree of $4i$ edges, 2 of which are of multiplicity 2 and rest of multiplicity 1. The root edge of this tree has multiplicity 2. The root edge has edge multiplicity 2. In a minimal coloring, there are two ways to color the root edge, either with colors $\{4i+2, 4i+1\}$ or with colors $\{4i, 4i-1\}$. These two assignments correspond to the binary assignment of the variable x_i.

The gadget for each clause is a tree with a complex construction. For a clause $x_{i_1} \vee x_{i_2} \vee x_{i_3}$, Marx first maps the indices i_1, i_2 and i_3 to a, b and c. In a minimal coloring of a clause

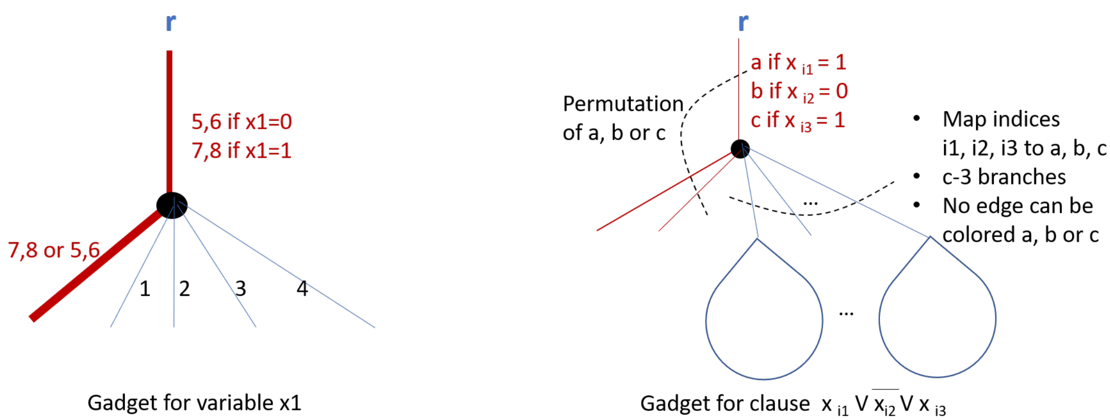

Figure 2: (Left) Variable gadget. (Right) Gadget for a clause with 3 literals.

gadget tree, the root edge has to have color a or b or c. Whichever variable makes this clause tree, the corresponding color shows up on the root edge. (If multiple variables can make the clause true, anyone can show up on the root edge.) By computing the mapping from the indices to a, b and c carefully, Marx makes sure that ϕ is satisfiable if and only if there is a way to coordinate the coloring of the root edges under a minimal coloring.

If we had a poly-time solution for MECS, we can then decide whether $T(\phi)$ has a minimum coloring and whether ϕ has a satisfiable assignment. This proves MECS is NP-hard. Formally,

Theorem 1 (Theorem 3.1 [Mar03]). *MECS is NP-hard even if the multiplicity is either 1 or 2 on each edge.*

Missing Link in Marx's Proof. However, Marx's proof assumes exactly three literals per clause. Recall that for 3-occurrence 3SAT to be NP-hard, some clauses must have 2 literals. We therefore add a gadget for 2-literal clauses and patched Marx's proof. Again, the gadget has many intricate details, but we are only showing an illustrative drawing in Figure 3.

Marx's NP Construction Not Generalizable to OCCP. Here we give a brief reason as to why Marx's construction cannot be tweaked to prove the hardness of OCCP, which minimizes all colors used.

Marx's variable gadget is a star with 2 edges of multiplicity of 2. The key to this construction is that there are 2 ways to color this star minimally, which correspond to the binary setting of the variable in ϕ. In this drawing, the two ways are assigning $\{7, 8\}$ or $\{5, 6\}$ to the top red edge, or flipping the assignment. This restriction no longer exists for OCCP because any permutation of the coloring would work. Therefore, if we use Marx's construction of $T(\phi)$, we can show that for any 3-occurrence 3-CNF formula ϕ, if ϕ is satisfiable, then $T(\phi)$ has

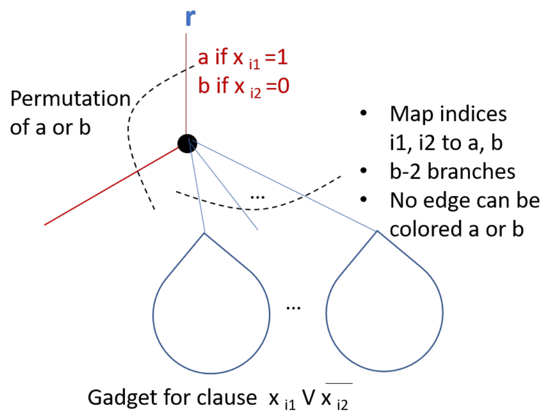

Figure 3: Gadget for a 2-literal clause.

a minimal coloring. On the other hand, even if ϕ is not satisfiable, we can still color $T(\phi)$ minimally since minimal coloring in OCCP is much less restrictive than SEMC. Of course, the fundamental reason that no NP-hard proof would work for OCCP is because it has a polynomial solution.

3 Polynomial Time Solution for OCCP

Linear Programming. In general, an instance of the linear programming (LP) problem is of the form

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax \geq b \\ & x \in \mathbb{Q}^n, \end{aligned}$$

where A is the coefficient matrix, b and c arbitrary constraint vectors. In [Vai89], vaidya presents an $O(n^{2.5})$ solution to LP problems. However, if variable x is restricted to entries of 0 or 1 and coefficient matrix A integral, then the problem becomes the well known NP-complete Binary Integer Programming (0-1 ILP) problem.

Linear Programming Formulation of OCCP. A result of [CRV08] says that solutions to OCCP for multi-bipartites require exactly Δ colors. Then, for each edge e and color

$i \in [\Delta]$, let $x_{e,i}$ be the indicator for $i \in \psi(e)$. Consider the following 0-1 ILP problem:

$$\begin{aligned} \min \quad & \sum_e \sum_i x_{e,i} \cdot i && \text{(1a)} \\ \text{s.t.} \quad & \sum_{e \in E_u} x_{e,i} \leq 1 \quad \forall u \in V, \forall i \in [\Delta] && \text{(1b)} \\ & \sum_i x_{e,i} \geq \mu(e) \quad \forall e \in E && \text{(1c)} \end{aligned}$$

The objective function (1a) is the sum of colors over all edges. Constraint (1b) requires that no adjacent edges have distinct colors. Constraint (1c) requires each edge to be colored by at least its multiplicity number of colors. Since this is a minimization problem, the solution x achieves equality in (1c). A solution to the above system would be a solution to OCCP for any bipartite graph input, although it would be exponential-time in general. However, a nice property about trees allow for a fast solution.

Theorem 2. *The OCCP problem for n-vertex multitrees admits an $O(n^5)$ solution.*

Proof. Consider a relaxation of the 0-1 ILP problem (1) to a LP, where x lies in $\mathbb{Q}^{\Delta(n-1)}$ and coefficients are also over $\mathbb{Q}$. By Lemma 4.2 of [Mar03], because the inputs are restricted to trees, the coefficient matrix of the system in (1) is totally unimodular. This implies all solutions to the relaxation of (1) are integral. Hence, running Vaidya's algorithm on (1), which has $\Delta(n-1) \leq n^2$ variables, gives a solution in $O(n^5)$. □

4 Conclusion

The area of graph coloring is rich with interesting problems, and the contrast between MECS and OCCP is just one of these. We will continue to explore related coloring problems on other classes of graphs, as well as providing a solution faster than the one in Theorem 3. We believe this is possible, since a similar formulation in [Kro+96] for vertex coloring has a linear-time solution.

Acknowledgment

We would like to thank the National Science Foundation grant CCF-1852215 for supporting our research. We would also like to thank Dr. Atefeh Mohajeri for her guidance, the DIMACS REU, and Dr. Lazaros Gallos for making all of this possible.

References

[CRV08] Jean Cardinal, Vlady Ravelomanana, and Mario Valencia-Pabon. "Chromatic edge strength of some multigraphs". In: *Electronic Notes in Discrete Mathematics* 30 (2008), pp. 39–44.

[Kro+96] Leo G Kroon et al. “The optimal cost chromatic partition problem for trees and interval graphs”. In: *International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer. 1996, pp. 279–292.

[Mar03] Dániel Marx. “Minimum sum multicoloring on the edges of trees”. In: *International Workshop on Approximation and Online Algorithms*. Springer. 2003, pp. 214–226.

[Sal03] Mohammad R Salavatipour. “On sum coloring of graphs”. In: *Discrete Applied Mathematics* 127.3 (2003), pp. 477–488.

[Tov84] Craig A. Tovey. “A Simplified NP-Complete Satisfiability Problem”. In: *Discrete Applied Mathematics* 8 (1984), pp. 85–89.

[Vai89] Pravin M Vaidya. “Speeding-up linear programming using fast matrix multiplication”. In: *30th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society. 1989, pp. 332–337.