List edge multicoloring in bounded cyclicity graphs

Dániel Marx*

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
H-1521 Budapest, Hungary
dmarx@cs.bme.hu

Abstract: The list edge multicoloring problem is a version of edge coloring where every edge e has a list of available colors $L(e)$ and an integer demand $x(e)$. For each e, we have to select $x(e)$ colors from $L(e)$ such that adjacent edges receive disjoint sets of colors. Marcotte and Seymour proved a good characterization theorem for list edge multicoloring in trees, which can be turned into a polynomial time algorithm. We present a slightly more general algorithm that works also on odd cycles. It is further extended to a randomized polynomial time algorithm handling even cycles as well.

Keywords: list coloring, edge coloring, multicoloring

1 Introduction

In the *list multicoloring* problem a graph $G(V, E)$ is given, together with a nonempty list of available colors $L(v) \subset \mathbb{N}$ and a demand $0 < x(v) \leq |L(v)|$ for every vertex $v \in V$. A *multicoloring* is an assignment of a set $\Psi(v) \subseteq L(v)$ of size $x(v)$ to every vertex $v \in V$, such that $\Psi(u) \cap \Psi(v) = \emptyset$ when u and v are adjacent vertices in G.

Clearly, the problem is a generalization of graph vertex coloring (coloring with k colors corresponds to $L(v) = \{1, 2, \ldots, k\}$ and $x(v) = 1$ for each v), therefore it is **NP**-complete in general. The problem remains **NP**-complete even when there is no restriction on the sets $L(v)$ but $x(v) = 1$ for every vertex v and the graph is either a planar bipartite graph [4] or the line graph of a complete bipartite graph [1]. Moreover, if there is no restriction on the demand $x(v)$, then the problem is **NP**-complete even in binary trees [7]. On the other hand, the problem can be solved in polynomial time for paths [3]. A good overview of list coloring and related problems can be found in [9].

*Research is supported by grant OTKA 30122 of the Hungarian National Science Fund

In this paper we consider the edge coloring version of list multicoloring:

List edge multicoloring

Input: A graph $G(V, E)$, a *demand* function $x : E \to \mathbb{N}$ and a *color list* L: $E \to 2^{\mathbb{N}}$ for each edge

Question: Is there a *multicoloring* Ψ: $E \to 2^{\mathbb{N}}$ such that

- $\Psi(e) \subseteq L(e)$ for all $e \in E$,
- $\Psi(e_1) \cap \Psi(e_2) = \emptyset$ if e_1 and e_2 are incident to the same vertex in G and
- $|\Psi(e)| = x(e)$ for all $e \in E$?

Hereinafter "coloring" will always mean list edge multicoloring. Marcotte and Seymour gave a good characterization for this problem in the special case when G is a tree. Denote by $E_c \subseteq E$ the set of those edges whose lists contain the color c, and for all $X \subseteq E$, let $\nu_c(X) = \nu(X \cap E_c)$ be the maximum number of independent edges in X whose lists contain c.

Theorem 1 (Marcotte and Seymour, 1990, [6]) *Let G be a tree. The list edge multicoloring problem has a solution if and only if for every $X \subseteq E$ we have*

$$\sum_{c \in \mathbb{N}} \nu_c(X) \geq \sum_{e \in X} x(e). \tag{1}$$

The necessity of the condition is obvious for any graph, since color c can be used at most $\nu_c(X)$ times in X, thus at most $\sum_{c \in \mathbb{N}} \nu_c(X)$ colors can be assigned to the edges in X.

This theorem, in general, does not remain valid on cycles. Figure 1 shows two uncolorable instances of the problem. The reader can easily convince himself that Inequality 1 holds for every subset X of the edges, but the graphs are not colorable.

The proof of Theorem 1 is based on the total unimodularity of a network matrix, thus, using standard techniques, the proof can be turned into a polynomial time algorithm by reducing the problem to a flow problem. Here we present another polynomial time algorithm, which solves the problem in a slightly more general class of graphs, including trees and odd cycles. Moreover, with some further modifications, it can be turned into a randomized polynomial time algorithm working on an even more general class of graphs, which also includes even cycles.

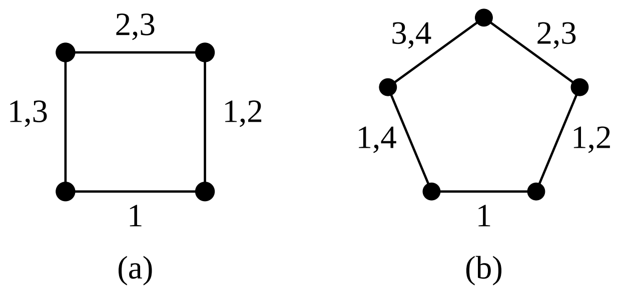

Figure 1: Theorem 1 does not hold for (a) even cycles and (b) odd cycles. Every edge has demand 1, the numbers on the edges are the colors contained in the lists of the edges.

In Section 2, a polynomial time solvable variant of the list edge multicoloring problem is introduced. This gives us a polynomial time solution of the original list edge multicoloring problem in some special cases (e.g. trees, odd cycles). Section 3 presents a modified randomized algorithm for list edge multicoloring arbitrary graphs having at most $|V| + O(1)$ edges.

2 A polynomial case

We introduce a new variant of list edge multicoloring. The requirement that edge e has to receive $x(e)$ colors is replaced by the requirement that the edges incident to v have to receive $y(v)$ colors in total. It turns out that in certain cases list edge multicoloring can be reduced to this new problem. Moreover, this problem can be solved in polynomial time for *any* graph (Theorem 4).

List edge multicoloring with demand on the vertices

Input: A graph $G(V, E)$, a demand function $y\colon V \to \mathbb{N}$ and a color list $L\colon E \to 2^{\mathbb{N}}$ for each edge

Question: Is there a *multicoloring* $\Psi\colon E \to 2^{\mathbb{N}}$ such that

- $\Psi(e) \subseteq L(e)$ for all $e \in E$,
- $\Psi(e_1) \cap \Psi(e_2) = \emptyset$ if e_1 and e_2 are incident to the same vertex in G, and
- $\sum_{e \ni v} |\Psi(e)| = y(v)$ for all $v \in V$?

The incidence matrix $\mathbf{B}$ of an undirected simple graph $G(V, E)$ has $|V|$ rows and $|E|$ columns, and for every $v \in V$ and $e \in E$, the element in row v and column e is 1 if e is incident to v and 0 otherwise. It will be convenient to think of the demand function $x\colon E \to \mathbb{N}$ in the list edge multicoloring problem as a vector $\mathbf{x}$ with $|E|$ (integer) components. Similarly, the demand function $y\colon V \to \mathbb{N}$ corresponds to a vector $\mathbf{y}$ with $|V|$ components. From now on, the demand function and its vector will be used interchangeably. A coloring Ψ is *valid for the edges* if $|\Psi(e)| = x(e)$ for every edge e. It is *valid for the vertices* if $\sum_{e \ni v} |\Psi(e)| = y(v)$ for every vertex v. When using these terms, the demand functions $x(e)$ and $y(v)$ will be clear from the context.

Let $\mathbf{x}$ be an arbitrary demand function on the edges of G, and define $\mathbf{y} = \mathbf{B}\mathbf{x}$. Let L be an arbitrary list assignment. If list edge multicoloring with demand $\mathbf{x}$ has a solution, then list edge multicoloring with demand $\mathbf{y}$ on the vertices has a solution as well. To see this, observe that any coloring Ψ valid for the edges is also valid for the vertices: $\sum_{e \ni v} |\Psi(e)| = \sum_{e \ni v} x(e)$ equals the component of $\mathbf{B}\mathbf{x} = \mathbf{y}$ corresponding to v, as required. The converse is not necessarily true: a coloring Ψ valid for the vertices is not always valid for the edges. In fact, as shown on Figure 2, it is possible that that there is a coloring satisfying the demand $\mathbf{y}$ on the vertices, but there is no coloring valid for the edges.

However, there is an important special case where every coloring valid for the vertices is also valid for the edges. We say that a graph $G(V, E)$ has *full edge rank* if the rank of $\mathbf{B}$ is $|E|$, that is, the characteristic vectors of the edges of G are linearly independent (over $\mathbb{Q}$).

Lemma 2 *Let $\mathbf{x}$ be an arbitrary demand function on the edges of G, and let $\mathbf{y} = \mathbf{B}\mathbf{x}$, where $\mathbf{B}$ is the incidence matrix of G. If G has full edge rank, then for every list assignment L, any coloring valid for the vertices is also valid for the edges.*

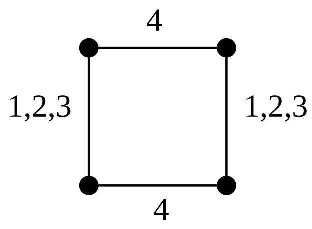

Figure 2: The list edge multicoloring with $x(e) \equiv 2$ has no solution, but with $y(v) \equiv 4$ there is a coloring valid for the vertices.

PROOF: Let Ψ be a coloring valid for the vertices. Define $x'(e) = |\Psi(e)|$, and let $\mathbf{x}'$ be the corresponding vector with $|E|$ components. Since $\sum_{e \ni v} |\Psi(e)| = y(v)$ holds, vector $\mathbf{x}'$ satisfies $\mathbf{B}\mathbf{x}' = \mathbf{y}$. However, the columns of $\mathbf{B}$ are linearly independent, thus $\mathbf{x}$ is the unique vector satisfying $\mathbf{B}\mathbf{x} = \mathbf{y}$. Hence $\mathbf{x} = \mathbf{x}'$, and $|\Psi(e)| = x(e)$ follows. $\square$

It is well known that every tree and odd cycle has full edge rank. From the definition it is clear that a graph has full edge rank if and only if all of its connected components have full edge rank. The following lemma characterizes all connected graphs having full edge rank.

Lemma 3 *A connected simple graph $G(V, E)$ has full edge rank if and only if it does not contain any even cycle and it has at most one odd cycle.*

PROOF: We prove the lemma by induction on the number of vertices. Assume first that G has a degree 1 vertex v, let e be the edge incident to v. In the incidence matrix $\mathbf{B}$ of G, there is only one non-zero element in row v, thus deleting row v and column e decreases the rank by exactly one. The resulting matrix is the incidence matrix of $G - v$, thus G has full edge rank if and only if $G - v$ has full edge rank. Since deleting a degree 1 vertex does not change any of the cycles, thus the lemma follows from the induction hypothesis.

Next assume that every vertex has degree at least 2. If G has full edge rank, then it has at most $|V|$ edges, thus the degree of every vertex is exactly 2 and G is a cycle. A cycle has full edge rank if and only if it is odd, thus the lemma holds in this case as well. $\square$

We show that list edge multicoloring with demand on the vertices can be solved in polynomial time. Together with Lemma 2, this implies that the list edge multicoloring problem also can be solved in polynomial time if the graph has full edge rank.

Theorem 4 *List edge multicoloring with demand on the vertices can be solved in polynomial time.*

PROOF: Let $C = |\bigcup_{e \in E} L(e)|$ be the total number of colors appearing in the lists, it can be assumed that $L(e) \subseteq \{1, 2, \ldots, C\}$ for every $e \in E$. We construct a graph $G'(U, F)$ as follows. For every $v \in V$, there are $2C - y(v)$ vertices $v_1, v_2, \ldots, v_C, v'_1, v'_2, \ldots, v'_{C-y(v)}$ corresponding to v in G'. If $uv \in E$ and $c \in L(uv)$, then there is an edge $u_c v_c$ in G'. Furthermore, for every $v \in V$, the vertices $v'_1, v'_2, \ldots, v'_{C-y(v)}$ are connected to every vertex $v_1, v_2, \ldots, v_C$. This completes the description of the graph G'.

We show that G' has a perfect matching if and only if there is a coloring of G valid for the edges. This implies the theorem, since there are polynomial time algoritms for finding perfect matchings in arbitrary graphs (cf. [5]). First assume that Ψ is coloring valid for the vertices. If $c \in \Psi(uv) \subseteq L(uv)$, then select the edge $u_c v_c$ into the set M'. Since $\Psi(uv)$ is a proper coloring, every vertex is covered at most once by the edges in M'. Furthermore, from the C vertices $v_1, v_2, \dots, v_C$, exactly $y(v)$ is covered by M'. The remaining $C - y(v)$ vertices can be matched with the $C - y(v)$ vertices $v'_1, v'_2, \dots, v'_{C-y(v)}$, thus we can extend M' to a perfect matching M of G'.

On the other hand, assume that $M \subseteq F$ is a perfect matching of G'. Let $c \in \Psi(uv)$ if and only if $u_c v_c \in M$, then clearly $\Psi(uv) \subseteq L(uv)$, since $u_c v_c \in M \subseteq F$ implies $c \in L(uv)$. Furthermore, $\Psi(uv) \cap \Psi(uw) = \emptyset$, since $c \in \Psi(uv)$ and $c \in \Psi(uw)$ would imply $u_c v_c \in M$ and $u_c w_c \in M$, which is impossible. What remains to be shown is that $\sum_{e \ni v} |\Psi(e)| = y(v)$. From the C vertices $v_1, v_2, \dots, v_C$ there are exactly $C - y(v)$ that are matched with the vertices $v'_1, v'_2, \dots, v'_{C-y(v)}$, thus the total size of the sets $\Psi(e)$ on the edges incident with e is exactly $y(v)$. $\square$

Corollary 5 *If G has full edge rank, then the list edge multicoloring problem can be solved in polynomial time.*

Corollary 6 *The list edge multicoloring problem can be solved in polynomial time for odd cycles.*

When note that if G is a tree, then the constructed graph G' is bipartite. Thus the more efficient bipartite matching algorithms can be used [2].

3 Bounded cyclicity graphs

In this section we try to extend the results of Section 2 to graphs that are "almost trees": to graphs that have only a small number of cycles. However, Lemma 2 is best possible:

Proposition 7 *If G does not have full edge rank, then there is a list assignment L and demand function $\mathbf{x}$ such that there is no coloring valid for the edges, but there is a coloring valid for the vertices (with $\mathbf{y} = \mathbf{B}\mathbf{x}$).*

Proof: If G does not have full edge rank, then there is a nonzero integer vector $\mathbf{z}$ with $\mathbf{B}\mathbf{z} = 0$. Since the columns of $\mathbf{B}$ are nonnegative vectors, at least one component of $\mathbf{z}$ is negative, suppose that $z(e^*) < 0$. Let $d = \min_{e \in E} z(e) < 0$ and let $L(e)$ be a set of $z(e) - d \geq 0$ colors such that every color appears in only one list. By setting $x(e) \equiv -d$, it is clear that there is no coloring valid for the edges, since $|L(e^*)| < -d = x(e^*)$. One the other hand, the coloring $\Psi(e) = L(e)$ is valid for the vertices. Coloring Ψ assigns $z(e) - d$ colors to edge e. Therefore the number of colors appearing at the vertices is given by the vector $\mathbf{B}(\mathbf{z} - d \cdot \mathbf{1}) = \mathbf{B}(-d \cdot \mathbf{1}) = \mathbf{B}\mathbf{x}$, as required. $\square$

On the other hand, we show that if a coloring Ψ is valid for the vertices and it satisfies some additional constraints, then it is also valid for the edges.

Lemma 8 *Let $G(V, E)$ be an arbitrary graph, and let $E' \subseteq E$ be a subset of edges such that $G'(V, E \setminus E')$ has full edge rank. For arbitrary demand function $\mathbf{x}$ and list assignment L, if coloring Ψ is valid for the vertices and it satisfies $|\Psi(e)| = x(e)$ for every $e \in E'$, then Ψ is also valid for the edges.*

PROOF: It can be assumed that the edges in E' correspond to the first $|E'|$ columns of $\mathbf{B}$. Therefore $\mathbf{B}$ can be written as $\mathbf{B} = (\mathbf{B}_1\ \mathbf{B}_2)$, where $\mathbf{B}_1$ has $|E'|$ columns. Similarly $\mathbf{x} = \binom{\mathbf{x}_1}{\mathbf{x}_2}$, $\mathbf{x}_1$ has $|E'|$ components. Clearly, $\mathbf{y} = \mathbf{Bx} = \mathbf{B}_1\mathbf{x}_1 + \mathbf{B}_2\mathbf{x}_2$.

Let $x'(e) = |\Psi(e)|$, and let $\mathbf{x}' = \binom{\mathbf{x}'_1}{\mathbf{x}'_2}$ be the corresponding vector. Since Ψ is valid for the vertices, it follows that $\mathbf{Bx}' = \mathbf{Bx} = \mathbf{y}$, that is

$$\mathbf{B}_1\mathbf{x}'_1 + \mathbf{B}_2\mathbf{x}'_2 = \mathbf{B}_1\mathbf{x}_1 + \mathbf{B}_2\mathbf{x}_2 = \mathbf{y}.$$

Moreover, since $|\Psi(e)| = x(e)$ for every $e \in E'$, we have that $\mathbf{x}'_1 = \mathbf{x}_1$, $\mathbf{B}_1\mathbf{x}'_1 = \mathbf{B}_1\mathbf{x}_1$, and $\mathbf{B}_2\mathbf{x}'_2 = \mathbf{B}_2\mathbf{x}_2$ follows. Since $G'(V, E \setminus E')$ has full edge rank, the columns of the matrix $\mathbf{B}_2$ are linearly independent, hence $\mathbf{x}'_2 = \mathbf{x}_2$. Therefore $\mathbf{x}' = \mathbf{x}$, and Ψ is valid for the edges. □

Since every coloring that is valid for the edges is also valid for the vertices, the list edge multicoloring problem has a solution if and only if there is a coloring valid for the vertices that satisfies the requirements in Lemma 8. We reduce the problem of finding such a coloring to a variant of the exact matching problem, which can be solved in randomized polynomial time. Before presenting the reduction (which is essentially the same as in the proof of Lemma 4), we briefly overview exact matching and some related problems.

There are strongly polynomial time algorithms for finding a maximum weight perfect matching in a graph (cf. [5]), it can be decided in polynomial time whether a perfect matching with weight *at least* K exists. In the *exact matching* problem we have to find a perfect matching whose weight *equals* K. The exact matching problem can be solved in randomized polynomial time if every weight is a polynomially bounded integer [8]. That is, there is a polynomial time algorithm that answers 'no' if there is no matching with weight exactly K, and answers 'yes' with probability at least $\frac{1}{2}$ if there is such a matching. It is an open question whether there exists a deterministic polynomial time algorithm for this problem.

A related problem is the following: given a graph $G(V, E)$, a set of edges $F \subseteq E$, and an integer k, find a perfect matching M of G with $|M \cap F| = k$. This can be reduced to the exact matching problem: let every edge in F have weight 1 and every other edge weight 0, then a matching M has $|M \cap F| = k$ if and only if its weight equals k. More generally, we can consider more than one subset of the edges:

Proposition 9 *Assume we are given a graph $G(V, E)$, pairwise disjoint subsets $F_0, F_1, \dots, F_\ell \subseteq E$, and integers $k_0, k_1, \dots, k_\ell$. If ℓ is a fixed constant, then it can be decided in randomized polynomial time whether there is a perfect matching M with $|M \cap F_i| = k_i$ for every $0 \le i \le \ell$.*

PROOF: This problem can be reduced to exact matching: let $e \in F_i$ have weight $(|E| + 1)^i$, edges in $E \setminus (F_0 \cup \dots \cup F_\ell)$ have weight 0, and set $K = \sum_{i=0}^{\ell} k_i(|E| + 1)^i$. It is easy to see that a matching satisfies the requirements if and only if it has weight K. If ℓ is a fixed constant, then the weight of every edge is polynomially bounded in the size of the graph, thus the problem can be solved in randomized polynomial time. □

Theorem 10 *For every fixed ℓ, there is a randomized polynomial time algorithm for list edge multicoloring in connected graphs having at most $|V| + \ell$ edges.*

PROOF: Let T be a spanning tree of $G(V, E)$, and let $E' = \{e_0, e_1, \dots, e_\ell\}$ be the $|E| - (|V| - 1) = \ell + 1$ edges not in T. Notice that $E \setminus E'$ has full edge rank (it is a tree). Construct the graph G', as in the proof of Lemma 4. For every $0 \leq i \leq \ell$, let F_i contain the $|L(e_i)|$ edges in G' that corresponds to edge e_i. That is, if $e_i = uv$ and $L(e_i) = \{c_1, c_2, \dots, c_r\}$, then $F_i = \{u_{c_1}v_{c_1}, u_{c_2}v_{c_2}, \dots, u_{c_r}v_{c_r}\}$. Set k_i to $x(e_i)$.

We show that the list edge multicoloring problem has a solution if and only if G' has a perfect matching M with $|F_i \cap M| = k_i$ for every $0 \leq i \leq \ell$. By Prop. 9, the latter problem can be solved in randomized polynomial time, hence the theorem follows.

If there is a perfect matching M in G such that $|F_i \cap M| = k_i$ for every $0 \leq i \leq \ell$, then construct a coloring Ψ valid for the vertices, as in the proof of Lemma 4. Clearly, $|\Psi(e_i)| = k_i = x(e_i)$, thus by Lemma 8, Ψ is valid for the edges.

The other direction also follows easily: since any coloring Ψ valid for the edges is also valid for the vertices, one can find a perfect matching M of G' based on Ψ. It is clear from the construction that M has exactly k_i edges from F_i, since $|\Psi(e_i)| = x(e_i) = k_i$. □

Corollary 11 *List edge multicoloring can be solved in randomized polynomial time for even cycles.*

Acknowledgments

I would like to thank Katalin Friedl for useful discussions and for her contribution to the final shape of the article.

References

[1] C. J. Colbourn. The complexity of completing partial Latin squares. *Discrete Appl. Math.*, 8(1):25–30, 1984.

[2] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2:225–231, 1973.

[3] W. F. Klostermeyer and J. L. Goldwasser. List multi-coloring problems. *Bull. Inst. Combin. Appl.*, 34:71–76, 2002.

[4] J. Kratochvíl. Precoloring extension with fixed color bound. *Acta Mathematica Universitatis Comenianae*, 62(2):139–153, 1993.

[5] L. Lovász and M. D. Plummer. *Matching Theory.* North-Holland Publishing Co., Amsterdam, 1986. Annals of Discrete Mathematics, 29.

[6] O. Marcotte and P. D. Seymour. Extending an edge-coloring. *J. Graph Theory*, 14(5):565–573, 1990.

[7] D. Marx. The complexity of tree multicolorings. In *Mathematical Foundations of Computer Science 2002 (Warsaw-Otwock)*, pages 532–542. Springer, Berlin, 2002.

[8] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[9] Zs. Tuza. Graph colorings with local constraints—a survey. *Discuss. Math. Graph Theory*, 17(2):161–228, 1997.