A Theorem about the Channel Assignment Problem

Daniel Král'[a] Riste Škrekovski*[a,b]

December 11, 2001

[a] Department of Applied Mathematics, DIMATIA and
Institute for Theoretical Computer Science (ITI)[1]
Charles University,
Malostranské náměstí 25, 118 00, Prague, Czech Republic
{kral,skreko}@kam.mff.cuni.cz

[b] Department of Mathematics,
University of Ljubljana,
Jadranska 19, 1111 Ljubljana, Slovenia
skreko@fmf.uni-lj.si

Abstract

A list channel assignment problem is a triple (G, L, w) where G is a graph, L is a function assigning vertices of G lists of integers (colours) and w is a function assigning edges of G positive integers (weights). A colouring c of the vertices of G is proper if $c(v) \in L(v)$ for each vertex v and $|c(u) - c(v)| \geq w(uv)$ for each edge uv. A weighted degree $\deg_w(v)$ of a vertex v is the sum of the weights of the edges incident with v. If G is connected, $|L(v)| > \deg_w(v)$ for at least one v and $|L(v)| \geq \deg_w(v)$ for all v, then a proper colouring always exists. A list channel assignment problem is balanced if $|L(v)| = \deg_w(v)$ for all v. We characterize all balanced list channel assignment problems (G, L, w) which admits a proper colouring.

MR Subject classification: 05C15
Keywords: graph colouring, list–colouring, channel assignment problem

*Supported in part by the Ministry of Science and Technology of Slovenia, Research Project Z1-3129, and in part by GAČR 201/1999/0242.
[1]Institute for Theoretical Computer Science is supported by Ministry of Education of Czech Republic as project LN00A056.

1 Introduction

We study a common generalization of colouring, list–colouring, and channel assignment problem which is called a list channel assignment problem in this paper. A *list channel assignment problem* is a triple (G, L, w): G is a graph, L is a function assigning vertices of G sets of positive integers, i.e. $L : V(G) \to 2^{\mathbb{N}}$, and w is a function assigning edges of G positive integers, i.e. $w : E(G) \to \mathbb{N}$. An assignment $c : V(G) \to \mathbb{N}$ of colours to the vertices of G is *proper* if $c(v) \in L(v)$ for each $v \in V(G)$ and $|c(u) - c(v)| \geq w(uv)$ for each $uv \in E(G)$. A *weighted degree* $\deg_w(v)$ of a vertex v of G is the sum of the weights of the edges incident with v; the *maximum weighted degree* $\Delta_w(G)$ is the largest $\deg_w(v)$ where $v \in V(G)$. If $w(e) = 1$ for all $e \in E(G)$, then the problem becomes a list–colouring problem [7, 14]. If $L(v) = \mathbb{N}$, then the problem becomes a channel assignment problem [10]. In the latter case, we define $\chi_w(G)$ to be the smallest number for which there is a proper assignment c such that $1 \leq c(v) \leq \chi_w(G)$ for all $v \in V(G)$. If both $w(e) = 1$ for all $e \in E(G)$ and $L(v) = \mathbb{N}$, then the problem becomes just an ordinary colouring problem for a graph G; note that $\chi(G) = \chi_w(G)$ in this case. A list channel assignment problem can be interpreted as follows: The vertices of G are transmitters, $L(v)$ is a set of frequencies which can be assigned to a vertex v and $w(uv)$ correspond to interference between transmitters u and v (the minimal distance between frequencies assigned to u and v).

Some theorems for ordinary colourings, list–colourings or channel assignment problems may be (naturally) extended to list channel assignment problems and others cannot: A (weighted) graph is called *k–degenerated* if each its induced subgraph contains a vertex of (weighted) degree at most k. If $|L(v)| \geq k + 1$ for each $v \in V(G)$ and $w(e) = 1$ for each $e \in E(G)$, then there exists a proper colouring. This can be reformulated using terminology of [7, 14]: Each k–degenerated graph is $(k+1)$–choosable, i.e. it admits a proper assignment for any lists such that $|L(v)| \geq k + 1$ for all $v \in V(G)$. If we remove the condition $w(e) = 1$, the conclusion becomes false — consider the following channel assignment problem: $V(G) = \{v_i | 1 \leq i \leq 5\} \cup \{v_{ij} | 1 \leq i < j \leq 5\}$, $E(G) = \{v_i v_j | 1 \leq i < j \leq 5\} \cup \{v_i v_{ij}, v_j v_{ij} | 1 \leq i < j \leq 5\}$, $L(v) = \{1, 2, 3, 4, 5\}$ for all v, $w(v_i v_j) = 1$ for $1 \leq i < j \leq 5$ and $w(v_i v_{ij}) = w(v_j v_{ij}) = 2$ for $1 \leq i < j \leq 5$. Note that G is weighted 4–degenerated, but (G, L, w) does not admit a proper assignment.

In the shade of the previous paragraph, it might be surprising that if $|L(v)| = \deg_w(v)$ for each $v \in V(G)$ and $|L(v)| > \deg_w(v)$ for at least one

$v \in V(G)$, then (G, L, w) admits a proper assignment (Theorem 1). This is a counterpart of a well–known inequality $\chi(G) \le \Delta(G) + 1$ where $\chi(G)$ is the chromatic number of G and $\Delta(G)$ is the maximum degree of G; the inequality $\chi_w(G) \le \Delta_w(G)+1$ for the channel assignment problem has been recently proved by McDiarmid in [9, 11, 12]. We state and prove an analogue of Brooks theorem for list channel assignment problem in this paper. Brooks theorem for ordinary colourings is proved in [3, 8], for choosability in [4, 15], for list–colourings in [1, 2, 4] and for list–colourings with separation in [6]. Brooks theorem for channel assignment problem was stated as an open problem in [10].

A list channel assignment problem is *balanced* if $L(v) = \deg_w(v)$. We characterize all balanced list channel assignment problems which admits a proper assignment (Theorem 5). In particular, we prove that a balanced list channel assignment problem (G, L, w) admits a proper assignment if G is 2–connected graph and it is neither a complete graph nor an odd cycle (for missing definitions see Subsection 1.1).

We first describe in Section 1 a greedy algorithm which was previously used for channel assignment problems by McDiarmid [9, 11, 12]. Then we prove Brooks theorem for list channel assignment problems (G, L, w) where G is a 2–connected graph in Section 3: distinct theorems for odd cycles (Theorem 3), complete graphs (Theorem 4) and remaining 2–connected graphs (Theorem 2) are stated. We prove the main theorem, Brooks theorem for list channel assignment problem, in Section 4 (Theorem 5). Brooks theorem for channel assignment problem and previously known Brooks theorems for other problems mentioned earlier can be easily derived from Theorem 5. Our results suggest a polynomial algorithm (Corollary 1) which given a balanced list channel assignment either outputs a proper assignment or states its non–existence. We devote Subsection 4.1 for corollaries of Theorem 5 to the channel assignment problem. We state that if $\chi_w(G) = \Delta_w(G) + 1$ and G is connected, then G is a Gallai tree in Theorem 6; there are really such Gallai trees as stated in Proposition 5.

1.1 Notation

We often deal with sets of integers in the paper; we write $[a, b]$ for an interval of integers between a and b (inclusively). We write $G - v$ for the graph obtained from G by deleting the vertex v together with the edges incident with v. We use standard graph notation throughout the paper and we refer a reader to any book about graph theory if necessary; we briefly recall some less known definitions: A graph with at least $k+1$ vertices is *k–connected* if it remains connected after removal of any $k-1$ or less vertices. A *block* of graph is a maximal (in edge–inclusion) subgraph which is 2–connected. A graph whose blocks are complete graphs and odd cycles is a *Gallai forest*; Gallai forests form an important class of graphs related to colourings of graphs as showed in [5].

2 Greedy Algorithm

The following greedy algorithm has been previously used by McDiarmid [9, 11, 12] to prove an upper bound for the span of channel assignment problems:

Algorithm 1

```
Input:  ordering of the vertices v_1,...,v_n
        edge-weight function w
        lists of colours L[1],...,L[n]
Output: assignment c of the colours to the vertices

colour  := minimum colour in L[i]'s
maxcol := maximum colour in L[i]'s
while colour <= maxcol do
  for i := 1 to n do
    if v_i is not coloured and colour is in L[i] then
      if for all neighbours v_j of v_i which are coloured holds
        | c[v_j] - colour | >= w ( v_i, v_j ) then
          c[v_i] := colour
      fi
    fi
  colour := colour + 1
done
```

We first state two propositions about Algorithm 1 which are straightforward to prove:

Proposition 1 *If Algorithm 1 assigns colours to all the vertices, then the obtained assignment is proper.*

Proposition 2 *The vertex v of G does not get a colour $k \in L(v)$ when applied Algorithm 1 if and only if the colour k is assigned to a neighbour of*

v preceding v in the ordering or a colour $k' < k$ is assigned to a neighbour v' of v such that $k - k' < w(vv')$.

We prove the list channel assignment counterpart of the well–known inequality $\chi(G) \leq \Delta(G) + 1$:

Theorem 1 *Let (G, L, w) be a list channel assignment problem. If $|L(v)| \geq \deg_w(v)$ for each $v \in V(G)$, the inequality is strict for at least one vertex and G is connected, then (G, L, w) admits a proper assignment.*

Proof: Let $v_1, \ldots, v_n$ be an ordering of the vertices of G such that $|L(v_n)| > \deg_w(v_n)$ and for each vertex $v_i, i < n$ there is $j > i$ such that v_i and v_j are adjacent; such an ordering can be obtained as a post–ordering of the vertices produced by a depth–first search algorithm started in v_n. We prove that each vertex gets a colour when we apply Algorithm 1 to this ordering (this is sufficient due to Proposition 1). Let v_i be a fixed vertex of G. Each neighbour u preceding v_i can prevent assigning a colour to v_i at most $w(v_i u)$ times and each neighbour u following v_i can prevent v_i from assigning a colour at most $w(v_i u) - 1$ times by Proposition 2; this together with a choice of the ordering implies that each vertex gets a colour.

■

One can immediately generalize the usage of Proposition 2 in the previous proof and state the following:

Proposition 3 *Let us apply Algorithm 1 to a list channel assignment problem (G, L, w) with the ordering $v_1, \ldots, v_n$ of its vertices. If v_i has not been assigned a colour, then $L(v_i)$ is a subset of the union of intervals $[c(v_j), c(v_j) + w(v_j v_i) - 1]$ where v_j is a coloured neighbour of v_i preceding v_i (i.e., $j < i$) and intervals $[c(v_j) + 1, c(v_j) + w(v_j v_i) - 1]$ where v_j is a coloured neighbour of v_i following v_i (i.e., $j > i$).*

We formulate Proposition 3 also for special case when (G, L, w) is balanced:

Proposition 4 *Let us apply Algorithm 1 to a balanced list channel assignment problem (G, L, w) with the ordering $v_1, \ldots, v_n$ of its vertices such that for each vertex v_i, $1 \leq i \leq n-1$, there is $j > i$ such that v_i and v_j are* *adjacent. Then the vertices $v_1, \ldots, v_{n-1}$ has been assigned colours. If v_n has not been coloured, then:*

$$L(v_n) = \bigcup_{v_i v_n \in E(G)} [c(v_i), c(v_i) + w(v_i v_n) - 1],$$

where the intervals in the above union are disjoint.

3 2–connected Graphs

Lemma 1 *Let (G, L, w) be a balanced list channel assignment problem. If G is 2–connected and $\min \bigcup_{v \in V(G)} L(v)$ or $\max \bigcup_{v \in V(G)} L(v)$ is not contained in all the lists, then (G, L, w) admits a proper assignment.*

Proof: We first deal with the case that the minimum colour is not contained in all the lists. Let $c_m = \min \bigcup_{v \in V(G)} L(v)$. Since G is connected there exist adjacent vertices v_1 and v_n such that $c_m \in L(v_1)$ and $c_m \notin L(v_n)$. Let $v_1, \ldots, v_n$ be any ordering of the vertices of G such for each vertex $v_i, i < n$ there is $j > i$ such that v_i and v_j are adjacent; such an ordering can be a post–ordering of the vertices produced by a depth–first search algorithm applied to $G - v_1$ started in v_n. Let us apply Algorithm 1. Each vertex (with a possible exception for v_n) has been assigned a colour due to Proposition 4. If v_n has not been assigned a colour, then the fact that the colour of v_1 is c_m and $c_m \notin L(v_n)$ yield contradiction due to Proposition 4.

The case that the maximum colour is not contained in all the lists can be dealt as follows: Let $L'(v) = \{M - k | k \in L(v)\}$ for sufficiently large M; then (G, L', w) has a proper assignment c' since the minimum colour is not contained in all the lists. The mapping $c(v) = M - c'(v)$ is a proper assignment of (G, L, w).

■

The following lemma can be found in [13] (Lemma 1.15):

Lemma 2 *Every 2-connected graph G which is neither a cycle nor a complete graph contains vertices x, y and z such that x and y are neighbours of z, the vertices x and y are non-adjacent, and $G - x - y$ is connected.*

Theorem 2 *Let (G, L, w) be a balanced list channel assignment problem. If G is 2–connected and it is neither an odd cycle nor a complete graph, then (G, L, w) admits a proper assignment.*

Proof: Let $c_1 = \min \bigcup_{v\in V(G)} L(v)$ and $c_2 = \max \bigcup_{v\in V(G)} L(v)$. If c_1 or c_2 is not contained in all the lists, we apply Lemma 1.

Suppose here that G is an even cycle. Let $v_1, \ldots, v_{2n}$ be the vertices of the cycle and let $c(v_i) = c_1$ for odd i and $c(v_i) = c_2$ for even i. The assignment c is proper due to the following:

$$|c(v_i)-c(v_{i+1})| = c_2-c_1 \geq |L(v_i)|-1 = w(v_{i-1}v_i)+w(v_iv_{i+1})-1 \geq w(v_iv_{i+1}).$$

We deal with the case that G is not a cycle in this paragraph. Let x, y and z be three vertices with the properties of the statement of Lemma 2. Let $x, y, v_3, \ldots, v_{n-1}, v_n = z$ be any ordering of the vertices of G such for each vertex $v_i, 3 \leq i < n$ there is $j > i$ such that v_i and v_j are adjacent; such an ordering can be obtained as a post–ordering of the vertices produced by a depth–first search algorithm applied to $G - x - y$ started in $v_n = z$. Let us apply Algorithm 1; this yields a partial assignment c. Each vertex (with a possible exception for v_n) has been assigned a colour due to Proposition 4. If $v_n = z$ has not been assigned a colour, then $L(v_n) = \bigcup_{v_iv_n\in E(G)}[c(v_i), c(v_i) + w(v_iv_n) - 1]$ and the intervals in the union have to be disjoint by Proposition 4. But this is impossible, since $c(x) = c(y) = c_1$. ■

3.1 Colouring Odd Cycles

We assume throughout this subsection that (G, L, w) is a balanced list channel assignment problem such that G is an odd cycle and $\{c_1, c_2\} \subseteq L(v)$ for all $v \in V(G)$ where $c_1 = \min \bigcup_{v\in V(G)} L(v)$ and $c_2 = \max \bigcup_{v\in V(G)} L(v)$; note that $c_2 - c_1 \geq w(e)$ for each edge $e \in E(G)$.

Lemma 3 *Suppose that (G, L, w) does not admit a proper list assignment. Then, for each its vertex v incident with edges e_1 and e_2 holds one of the following:*

$$L(v) = \begin{cases} [c_1, c_1 + w(e_1) - 1] \cup [c_2 - w(e_1) + 1, c_2], & \text{if } w(e_1) = w(e_2) \\ [c_1, c_2], & \text{otherwise.} \end{cases}$$

Proof: Suppose that the claim is false. Let v be a vertex adjacent to the edges e_1 and e_2 which does not satisfy none of the above cases, in particular $L(v)$ is not an interval. Let us assume $w(e_1) \leq w(e_2)$. If $c_2 - w(e_2) < c_1 + w(e_1)$, then we get that $w(e_1) + w(e_2) = c_2 - c_1 - 1$ and $L(v)$

is an interval. We prove that there is $k \in L(v)$ such that $c_1 + w(e_1) \leq k \leq c_2 - w(e_2)$ or $c_1 + w(e_2) \leq k \leq c_2 - w(e_1)$. If there is no such k, then $L(v) \subseteq [c_1, c_1 + w(e_1) - 1] \cup [c_2 - w(e_2) + 1, c_2]$ and $L(v) \subseteq [c_1, c_1 + w(e_2) - 1] \cup [c_2 - w(e_1) + 1, c_2]$. This implies the following inclusion:

$$L(v) \subseteq [c_1, c_1 + w(e_1) - 1] \cup [c_2 - w(e_2) + 1, c_1 + w(e_2) - 1] \cup [c_2 - w(e_1) + 1, c_2].$$

The middle interval in the union above might be empty. From the above inclusion one gets easily that $|L(v)| \leq w(e_1) + w(e_2) - 1$ which contradicts that (G, L, w) is balanced.

Let k be such that $c_1 + w(e_1) \leq k \leq c_2 - w(e_2)$ or $c_1 + w(e_2) \leq k \leq c_2 - w(e_1)$. Then we can assign to the vertices of G except for v alternatively colours c_1 and c_2 and to the vertex v the colour k. The above inequalities assure that one of the two possible alternating assignments is proper.

■

Theorem 3 *A list channel assignment problem (G, L, w) does not have a proper assignment if and only if there exist integers $1 \leq a \leq b$ and $1 \leq k$ such that one of the following holds for each vertex v:*

(a) The vertex v is adjacent to two edges with weights a and $L(v) = [k, k + a - 1] \cup [k + b, k + a + b - 1]$.

(b) The vertex v is adjacent to two edges of different weights a and b (thus $a < b$ in this case) and $L(v) = [k, k + a + b - 1]$.

Proof: If (G, L, w) does not admit a proper assignment, then it is of the form described in the statement due to Lemma 3: Let a be the weight of the lightest edge, k minimum colour contained in all the lists (cf. Lemma 1) and k' the maximum one; let $b = k' - k - a + 1$. If a vertex is incident with two edges of weights a, its list is a disjoint union of two intervals of length a by Lemma 3 and due to choice of a and b of the form described in (a). On the other hand, if a vertex is incident with an edge of weight a and an edge of another weight, its list have to be an interval $[k, k'] = [k, k + a + b - 1]$ and thus the weight of its other edge is b by Lemma 3. The other end–vertex of the edge of weight b cannot be incident with another edge of weight b, thus its list has to be an interval $[k, k']$ and the weight of the other edge (distinct from that with weight b) is a. In this fashion, one can prove that each vertex of the cycle is incident with an edge of weight a and the lists of the vertices are of the form described in the statement of the theorem.

Next, we prove the list channel assignment problems of the form described in the statement of the theorem do not admit proper assignments. We say a vertex has been assigned a *low* colour if its colour is in $[k, k+a-1]$ and it has been assigned a *high* colour if its colour is in $[k+b, k+a+b-1]$. Each vertex must be assigned either a low or a high colour (a vertex incident with an edge with weight b cannot be assigned colour in $[k+a, k+b-1]$ since this would disable colouring the other end of such edge because of $a \leq b$) and no two adjacent vertices can be assigned both either low or high colours (the weight of each edge is at least a). This together with the fact that G is an odd cycle proves the theorem.

■

3.2 Colouring Complete Graphs

We assume throughout this subsection that (G, L, w) is a balanced list channel assignment problem such that G is a complete graph.

Lemma 4 *Suppose that (G, L, w) does not admit a proper assignment. Let $v_1, \ldots, v_n$ be an ordering of the vertices of G and c the assignment obtained by Algorithm 1 run to the sequence $v_1, \ldots, v_n$ (this assigns all vertices except for v_n colours due to Proposition 3). The following holds:*

(a) $c(v_1) < \cdots < c(v_{n-1})$;

(b) $\bigcup_{j=1}^{i-1}[c(v_j), c(v_j) + w(v_j v_i) - 1] = \{k | k \in L(v_i) \wedge k < c(v_i)\}$ *for all* $2 \leq i \leq n-1$;

(c) $L(v_n) = \bigcup_{1 \leq i \leq n-1}[c(v_i), c(v_i) + w(v_i v_n) - 1]$.

Proof: Let k_1 be the least colour contained in any of the lists; by Lemma 1, k_1 is contained in all the lists. If there is $i > 1$ such that $[k_1, k_1 + w(v_1 v_i) - 1] \not\subseteq L(v_i)$, then Algorithm 1 applied to the sequence $v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n, v_i$ yields a proper assignment due to Proposition 4 (partitioning described in it cannot exists since $[k_1, k_1 + w(v_1 v_i) - 1] \not\subseteq L(v_i)$ and $c(v_1) = k_1$).

Let k_2 be the least colour contained in the following set:

$$\bigcup_{i=2}^{n} (L(v_i) \setminus [k_1, k_1 + w(v_1 v_i) - 1])$$

The colour k_2 is the second (according to the time) colour assigned to a vertex of G by Algorithm 1 run to the sequence $v_1, \ldots, v_n$. We prove that $k_2 \in L(v_2) \setminus [k_1, k_1 + w(v_1v_2) - 1]$ and $[k_2, k_2 + w(v_2v_i) - 1] \subseteq L(v_i) \setminus [k_1, k_1 + w(v_1v_i) - 1]$ for $i \geq 3$: If $k_2 \notin L(v_2) \setminus [k_1, k_1 + w(v_1v_2) - 1]$, we apply Algorithm 1 to the sequence $v_1, v_3, \ldots, v_n, v_2$ and we get a colouring of (G, L, w) since the partitioning described in Proposition 4 cannot exist because there is a vertex with a colour k_2 and $k_2 \notin L(v_2) \setminus [k_1, k_1 + w(v_1v_2) - 1]$. Thus Algorithm 1 assigns colour k_2 to v_2. If there is $i \geq 3$ such that $[k_2, k_2 + w(v_2v_i) - 1] \not\subseteq L(v_i) \setminus [k_1, k_1 + w(v_1v_i) - 1]$, then we apply Algorithm 1 to the sequence $v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n, v_i$, we get a colouring of (G, L, w) by Proposition 4 (partitioning described in it cannot exists since $[k_2, k_2 + w(v_2v_i) - 1] \not\subseteq L(v_i) \setminus [k_1, k_1 + w(v_1v_i) - 1]$, $c(v_1) = k_1$ and $c(v_2) = k_2$). Hence, $[k_2, k_2 + w(v_2v_i) - 1] \subseteq L(v_i)$ for all $i \geq 3$.

Let k_3 be the third (according to the time) colour assigned to a vertex of G by Algorithm 1 run to the sequence $v_1, \ldots, v_n$. Then the following holds (for $i \geq 4$):

$$k_3 \in L(v_3) \setminus ([k_1, k_1 + w(v_1v_3) - 1] \cup [k_2, k_2 + w(v_2v_3) - 1])$$

$$[k_3, k_3 + w(v_3v_i) - 1] \in L(v_i) \setminus ([k_1, k_1 + w(v_1v_i) - 1] \cup [k_2, k_2 + w(v_2v_i) - 1])$$

The argumentation is essentially the same as in the previous paragraph: If the above is not the case, then we apply Algorithm 1 to the sequence v_1, v_2, $v_4, \ldots, v_n$, v_3 (if the first is false) or $v_1, \ldots, v_{i-1}, v_{i+1}, v_n, v_i$ (if the latter is false) and we get a proper assignment. We conclude that Algorithm 1 assigns colour k_3 to v_3. We can continue in this fashion and we find $k_4, \ldots, k_{n-1}$ the colours assigned to $v_4, \ldots, v_{n-1}$.

Finally, the claim (a) follows from $k_1 < k_2 < \cdots < k_{n-1}$. The claim (b) is established by inclusions $[k_1, k_1 + w(v_1v_2)] \subseteq L(v_2)$, $[k_1, k_1 + w(v_1v_3)] \cup [k_2, k_2 + w(v_2v_3)] \subseteq L(v_3)$, etc.; the claim (c) follows also from these inclusions (or from Proposition 4, too).

■

Lemma 5 *Suppose that (G, L, w) does not admit a proper assignment. If a vertex v is adjacent to at least two edges of different weights, then $L(v)$ is an interval.*

Proof: Let v be a fixed vertex of G and k the smallest number of $L(v)$. Let a be the minimum weight of an edge adjacent to v and b the maximum weight of an edge adjacent to v. Consider an ordering $O = v_1, \ldots, v_n$ of the vertices of G such that $v_n = v$, $w(v_{n-2}v_n) = a$ and $w(v_{n-1}v_n) = b$; let $w_i = w(v_i v)$ for $1 \leq i \leq n-1$. We prove by induction on i that $[k, k + w_1 + \cdots + w_i - 1] \subseteq L(v)$ and then since (G, L, w) is balanced, we get that $L(v)$ is an interval.

If $i = 1$, it is enough to apply Lemma 4 to the above sequence of the vertices of G. Let us suppose $i > 1$. Suppose first that $w_{i-1} < b$. If we apply Lemma 4 to the sequence $v_1, \ldots, v_{i-2}, v_{n-1}, v_{i-1}, v_i, \ldots, v_{n-2}, v_n$, we get that $k + w_1 + \cdots + w_{i-1} \in L(v)$ since $[k, k + w_1 + \cdots + w_{i-1} - 1] \subseteq L(v)$ and $w_{i-1} < b$; this is because $[k, k + w_1 + \cdots + w_{i-1} - 1] \subseteq L(v)$ and $L(v)$ can be covered by intervals of length $w_1, w_2, \ldots, w_{i-2}, w_{n-1} = b, w_{i-1}, w_i, \ldots, w_{n-2}$ which follow one after another. If we apply Lemma 4 to the ordering O, then we get $[k, k + w_1 + \cdots + w_i - 1] \subseteq L(v)$ since Algorithm 1 applied to the ordering O colours v_i by $k + w_1 + \cdots + w_{i-1}$.

We deal with the remaining case $w_{i-1} = b$ in this paragraph. We first prove that $k + w_1 + \cdots + w_{i-1} \in L(v)$: Let us apply Lemma 4 to the sequence $v_1, \ldots, v_{i-2}, v_{n-2}, v_{i-1}, v_i, \ldots, v_{n-3}$, v_{n-1}, v_n and then realize that $[k, k + w_1 + \cdots + w_{i-1} - 1] \subseteq L(v)$, $w_{n-2} < b = w_{i-1}$ and $L(v)$ can be covered by intervals of length $w_1, w_2, \ldots, w_{i-2}, w_{n-2} = a, w_{i-1}, w_i, \ldots, w_{n-3}, w_{n-1}$ which follow one after another. Next, we apply Lemma 4 to the ordering O, and we conclude that $[k, k + w_1 + \cdots + w_i - 1] \subseteq L(v)$; the argumentation is like in the previous case.

■

Theorem 4 *A list channel assignment problem (G, L, w) with the vertex set $V(G)$ equal to $\{v_1, \ldots, v_n\}$ does not admit a proper assignment if and only if one of the following holds:*

(a) There exist integers $1 \leq a$ and $1 \leq k_1 < \cdots < k_{n-1}$ such that $k_i + a \leq k_{i+1}$ for $1 \leq i \leq n-2$, $w(e) = a$ for all $e \in E(G)$ and $L(v_i) = \bigcup_{1 \leq j \leq n-1} [k_j, k_j + a - 1]$ for all $1 \leq i \leq n$.

(b) There exist integers $1 \leq a < b$ and $1 \leq k$ such that $w(v_i v_j) = b$ for $1 \leq i, j \leq n-1$, $w(v_i v_n) = a$ for $1 \leq i \leq n-1$, $L(v_i) = [k, k + b(n-2) + a - 1]$ for $1 \leq i \leq n-1$ and $L(v_n) = \bigcup_{0 \leq j \leq n-2} [k + bj, k + bj + a - 1]$.

Proof: None of list channel assignment problems described in the statement admit a proper assignment. We prove that they are the only ones. We distinguish several cases:

- **The weights of all the edges are the same.** Let a be the common weight of all the edges. It is enough to prove that $L(v_i) = L(v_j)$ for all $1 \leq i, j \leq n$ by Lemma 4. Let us assume the opposite; we may suppose that $L(v_{n-1}) \neq L(v_n)$. Let κ be the colour assigned to v_{n-1} by Algorithm 1 applied to the sequence $v_1, \ldots, v_n$. Then by Lemma 4:

  $\{i | i \in L(v_{n-1}) \wedge i < \kappa\} = \{i | i \in L(v_n) \wedge i < \kappa\}$ and $[\kappa, \kappa + a - 1] \subseteq L(v_n)$.

  If we apply Lemma 4 to the sequence $v_1, \ldots, v_{n-2}, v_n, v_{n-1}$, we get $[\kappa, \kappa + a - 1] \subseteq L(v_{n-1})$. This implies that $L(v_{n-1}) = L(v_n)$.

- **All the lists are intervals.** We claim that the weights of the edges are the same (which was dealt in the first case). Otherwise, there exists a vertex which is adjacent to two edges of different weights. We may assume that v_1 is such a vertex and the edge $v_1 v_2$ is the edge with the largest weight incident with v_1 and $v_1 v_3$ is the edge with the smallest weight incident with v_1. If we apply Algorithm 1 to the sequence $v_1, \ldots, v_n$, we get an assignment such that $c(v_3) < c(v_2)$ which is contradicted by Lemma 4.

- **There exist edges of different weights and a vertex whose list is not an interval.** Let k be the smallest colour in the lists; then k is contained in all the lists by Lemma 1. Let v_n be a vertex such that $L(v_n)$ is not an interval. The edges incident with v_n have the same weight by Lemma 5; let a be their common weight. We first prove that the weight of any other edge is at least a. Suppose the opposite and assume that $w(v_1 v_2) < a$ (note that then $L(v_2)$ is an interval by Lemma 5). If we apply Algorithm 1 to the sequence $v_1, v_n, v_2, \ldots, v_{n-1}$, we get an assignment such that $c(v_2) < c(v_n)$ which is impossible due to Lemma 4. Thus the weight of any edge in the graph is at least a. Let b be further the largest weight of an edge; we may assume that $w(v_1 v_2) = b$. Note that $a < b$ since there are edges of different weights and thus both $L(v_1)$ and $L(v_2)$ are intervals.

  Lemma 4 applied to the sequence $v_1, v_2, \ldots, v_n$ gives (since the algorithm assigns the colour k to v_1 and $k + w(v_1 v_2) = k + b$ to v_2) that $L(v_n) \cap [k, k + b] = [k, k + a - 1] \cup \{k + b\}$. If there is an edge $e = xy$ such that $a \leq w(e) < b$ where neither x nor y is v_n, then Algorithm 1 applied to the sequence $x, y, \ldots, v_n$ assigns x the colour k and y the colour $k + w(e)$ (note that $L(y)$ has to be an interval by Lemma 5) due to Lemma 4; but then due to the same lemma,

$k+w(e) \in L(v_n)$ which is false (note that $k+a \leq k+w(e) < k+b$). Thus the weights of all the edges which are not incident with v_n are b. Lemma 4 applied to the sequence $v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{n-1}, v_i, v_n$ gives together with Lemma 5 that $L(v_i) = [k, k+(n-2)b+a-1]$ and $L(v_n) = \bigcup_{0 \leq i \leq n-2} [k+bi, k+bi+a-1]$.

■

4 Brooks Theorem for List Channel Assignment Problem

We prove that the only bad list channel assignment problems can be obtained by pasting bad odd cycles and bad complete graphs with disjoint lists:

Theorem 5 *A balanced list channel assignment problem (G, L, w) does not admit a proper assignment if and only if G is a Gallai forest whose blocks are $G_1, \ldots, G_m$ and there exist $L_i : V(G_i) \to 2^{\mathbb{N}}$ for $1 \leq i \leq m$ with the following properties:*

- *$L(v)$ is a union of $L_i(v)$ for i such that $v \in G_i$ and $L_i(v) \cap L_j(v) = \emptyset$ for $i \neq j$.*
- *Let w_i be the weight function w restricted to the edges of G_i. Then (G_i, L_i, w_i) is balanced and it is one of the following three types:*
  - *(a) G_i is an odd cycle and there exists integers $1 \leq a \leq b$ and k such that each vertex v of G_i is* **either** *incident with two edges with the weights a in (G_i, L_i, w_i) and $L_i(v) = [k, k+a-1] \cup [k+b, k+a+b-1]$* **or** *$v$ is incident with an edge with weight a and an edge with weight b and $L_i(v) = [k, k+a+b-1]$.*
  - *(b) G_i is a complete graph with n vertices. There exist integers $1 \leq a$ and $1 \leq k_1 < \ldots < k_{n-1}$ such that $k_j + a \leq k_{j+1}$ for $1 \leq j \leq n-2$, $w_i(e) = a$ for all $e \in E(G_i)$ and $L_i(v) = \bigcup_{1 \leq j \leq n-1} [k_j, k_j + a-1]$ for any $v \in V(G)$.*
  - *(c) G_i is a complete graph with n vertices. There exist integers $1 \leq a < b$, $1 \leq k$ and a vertex $v \in V(G_i)$ such that $w_i(e) = b$ for any edge e of G_i which is not incident with v, $w_i(e) = a$ for any edge*

e of G_i incident with v, the list $L_i(u) = [k, k+b(n-2)+a-1]$ for any $u \in V(G), u \neq v$, and $L(v) = \bigcup_{0 \leq i \leq n-2}[k+bi, k+bi+a-1]$.

Proof: It is enough to prove the theorem for connected graphs. The proof proceeds by induction on the number of blocks. If G has just one block, then the proof immediately follows from Theorems 2–4.

Suppose G has at least two blocks. Let G_1 be one of its end–blocks, v the cut vertex separating G_1 from the rest of G and G' be the rest of G including v. Let L_1 (resp. L') be the function L restricted to G (resp. G') except for v and w_1 (resp. w') be the function w restricted to the edges of G (resp. G'). Let U_1 (resp. U') be the largest set of colours such that the list channel assignment problem (G_1, L_1, w_1) (resp. (G', L', w')) with $L_1(v) = U$ (resp. $L'(v) = U'$) does not admit a proper assignment. Note that $|U_1| \leq \deg_{w_1}(v)$ and $|U'| \leq \deg_{w'}(v)$ due to Theorem 1 and the sets U_1 and U' are uniquely determined (they are simply the sets of those colours such that when assigned to v, there is no proper extension to the rest of the graph). Thus for any $k \in L(v) \backslash U_1$ (resp. $k \in L(v) \backslash U'$) there is a proper assignment of (G_1, L_1, w_1) (resp. (G', L', w')) such that the colour of v is k.

If there is k such that $k \in L(v) \setminus (U_1 \cup U')$, we can assign to v the colour k and extend this to a proper assignment of G_1 and G' and thus also of G. If (G, L, w) does not admit a proper assignment, then $|U_1| = \deg_{w_1}(v)$, $|U'| = \deg_{w'}(v)$ and $L(v) = U_1 \cup U'$ (equality is because (G, L, w) is balanced). In such case, (G_1, L_1, w_1) has to be either an odd cycle described in (a) due to Theorem 3 or a complete graph described in (b) or (c) due to Theorem 4. The channel assignment problem (G', L', w') is of the desired form due to the induction hypothesis.

On the other hand, if (G_1, L_1, w_1) is a "bad" cycle or a "bad" complete graph and (G', L', w') is the union of "bad" cycles and complete graphs described in the statement of the theorem, then (G, L, w) does not admit a proper assignment.

∎

Note that the proof of Theorem 5 suggests an algorithm for recognizing balanced channel assignment problems which admit proper assignments: We take an end–block of the given graph and we consider one of its vertices which is not a cut vertex; this vertex together with the weights of the edges of that block determines type of a bad graph (if it is bad) and the corresponding lists of colours at each vertex. We remove this block and continue until either we find an end–block which is not bad or we get an empty graph. If we find a block which is not bad, we use the way suggested by proofs of Theorem 2, Theorem 3 or Theorem 4 to colour it. Hence, we may conclude:

Corollary 1 *There exists a polynomial–time algorithm which for given a balanced list channel assignment problem either finds a proper assignment or decides that a proper assignment does not exist.*

4.1 Channel Assignment Problem

Theorem 5 provides Brooks theorem also for the channel assignment problem when applied to the lists which all are equal to $[1, \Delta_w(G)]$ where G is a given graph and w is a weight function on the edges of G; note that if there is v in G such that $\deg_w(v) < \Delta_w(G)$ and G is connected, then $\chi_w(G) \leq \Delta_w(G)$ as proved in [9, 11, 12] (this also follows from Theorem 1). Thus we immediately get from Theorem 5 the following:

Theorem 6 *Let G be a connected graph and let w be a function which assigns to the edges of G positive weights. If $\chi_w(G) = \Delta_w(G) + 1$, then the weighted degree of each vertex of G is equal to $\Delta_w(G)$ and one of the following holds:*

- *G is an odd cycle and all its edges have the same weights.*
- *G is a complete graph and all its edges have the same weights.*
- *G is a Gallai tree with at least two blocks.*

There are really Gallai trees such that $\chi_w(G) = \Delta_w(G) + 1$ as shown stated in the following:

Proposition 5 *There exists a connected graph G and a function w which assigns to the edges of G positive weights such that $\chi_w(G) = \Delta_w(G) + 1$ and G is not 2–connected.*

Proof: Let $1 \leq a < b$ and $2 \leq n$ be fixed integers. Let G_i be a complete graph on $n + 1$ vertices and let v_i be one of its vertices for $1 \leq i \leq n$. We assign the edges of G_i which are not adjacent to v_i weights b and the edges which are adjacent to v_i weights a. We further form a complete graph on the vertices $v_1, \ldots, v_n$ and we assign the edges of this graph the weights equal to $b - a$. It is easy to check that the weighted degree of each of the vertices is equal to $a + (n-1)b$ and the minimal span of a channel assignment of it is $a + (n-1)b + 1$ by Theorem 5 (we use it for the set $[1, a + (n-1)b]$ assigned to all the vertices).

$\blacksquare$

We remark that the construction of Proposition 5 can be extended to Gallai trees whose some blocks are odd cycles and whose structure is more complex.

References

[1] O. V. Borodin: *Criterion of chromaticity of a degree prescription*, in: Abstracts of IV All–Union Conf. on Theoretical Cybernetics, Novosibirsk, 1977, 127–128 (in Russian).

[2] O. V. Borodin: *Problems of colouring and of covering the vertex set of a graph by induced subgraphs*, Ph. D. Thesis, Novosibirsk State University, Novosibirsk, 1979 (in Russian).

[3] R. L. Brooks: *On colouring the nodes of a network*, Proc. Cambridge Phil. Soc. 37, 1941, 194–197.

[4] P. Erdös, A. L. Rubin, H. Taylor: *Choosability in graphs*, Proc. West Coast Conf. on Combinatorics, Graph Theory and Computing, Congressus Numerantium XXVI, 1979, 125–157.

[5] T. Gallai: *Kritische Graphen I*, Publ. Math. Inst. Hung. Acad. Sci. 8, 1963, 373–395.

[6] J. Kratochvíl, Z. Tuza, M. Voigt: *Brooks-type theorems for choosability with separation*, Journal of Graph Theory, 27 (1998), 43-49.

[7] J. Kratochvíl, Z. Tuza, M. Voigt: *New trends in the theory of graph colorings: Choosability and list coloring*, in: Contemporary Trends in Discrete Mathematics (from DIMACS and DIMATIA to the future), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 49, American Mathematical Society, Providence, RI, 1999, 183-197.

[8] L. Lovasz: *Three short proofs in graph theory*, J. Combin. Theory Ser. B 19 (1975), 269-271.

[9] C. McDiarmid: *Bounds for the span in channel assignment problems*, talk at the 18th Biritish Combinatorial Conference, July 2001.

[10] C. McDiarmid: *Discrete mathematics and radio channel assignment*, in: Recent advances in theoretical and applied discrete mathematics, Eds. C. Linhares–Salas and B. Reed, Springer, 2001.

[11] C. McDiarmid: *On the span in channel assignment problems*, talk at the Midsummer Combinatorial Workshop, Prague, August 2001.

[12] C. McDiarmid: *On the span in channel assignment problems: bounds, computing and counting*, submitted.

[13] M. Molloy, B. Reed: *Graph colouring and the Probabilistic Method*, Algorithms and Combinatorics, Vol. 23, Springer, 2001.

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

[15] V. G. Vizing: *Colouring the vertices of a graph with prescribed colours*, Metody Diskretnogo Analiza Teorii Kodov i Skhem No 29, 1976, 3–10 (in Russian).