Vertex colouring edge weightings: A logarithmic upper bound on weight-choosability

Kasper Szabo Lyngsie
Department of Mathematics
Technical University of Denmark
Denmark
kasperszabo@hotmail.com

Liang Zhong[*]
Center for Discrete Mathematics
Fuzhou University
China
liangzh42@gmail.com

Submitted: Mar 9, 2017; Accepted: May 4, 2020; Published: Apr 23, 2021

Abstract

A graph G is said to be (k,m)-choosable if for any assignment of k-element lists $L_v \subset \mathbb{R}$ to the vertices $v \in V(G)$ and any assignment of m-element lists $L_e \subset \mathbb{R}$ to the edges $e \in E(G)$ there exists a total weighting $w : V(G) \cup E(G) \to \mathbb{R}$ of G such that $w(v) \in L_v$ for any vertex $v \in V(G)$ and $w(e) \in L_e$ for any edge $e \in E(G)$ and furthermore, such that for any pair of adjacent vertices u, v, we have $w(u)+\sum_{e\in E(u)} w(e) \neq w(v)+\sum_{e\in E(v)} w(e)$, where $E(u)$ and $E(v)$ denote the edges incident to u and v respectively. In this paper we give an algorithmic proof showing that any graph G without isolated edges is $(1, 2\lceil \log_2(\Delta(G))\rceil + 1)$-choosable, where $\Delta(G)$ denotes the maximum degree in G.

Mathematics Subject Classifications: 05C07, 05C15

1 Introduction

A graph is said to be k-*choosable* if for any assignment of k-element lists $L_e \subset \mathbb{R}$ to the edges $e \in E(G)$ there exists a *weighting* $w : E(G) \to \mathbb{R}$ of G such that $w(e) \in L_e$ for any edge $e \in E(G)$ and furthermore, such that for any pair of adjacent vertices u, v, we have $\sum_{e\in E(u)} w(e) \neq \sum_{e\in E(v)} w(e)$, where $E(u)$ and $E(v)$ denote the edges incident to u and v respectively. This concept of weight-choosability was introduced in [1] by Bartnicki, Grytczuk and Niwczyk. The motivation for this concept was that it generalises the well-known *1-2-3 Conjecture* formulated in [4], which states that the edges of any graph with no isolated edges can be labelled with the numbers 1, 2 and 3 so that any two adjacent vertices have different sums of incident edge-labels. In particular, if a graph is 3-chooseable

[*]Supported by The China Scholarship Council

it satisfies the 1-2-3 Conjecture. Bartnicki et al. [1] proved that trees and complete graphs (which are not K_2) are 3-choosable and conjectured that any graph without an isolated edge is 3-choosable. A more general concept of weight-choosability where there are also weights on the vertices was introduced in [10] by Wong and Zhu and is defined as follows: a graph G is said to be (k,m)*-choosable* if for any assignment of k-element lists $L_v \subset \mathbb{R}$ to the vertices $v \in V(G)$ and any assignment of m-element lists $L_e \subset \mathbb{R}$ to the edges $e \in E(G)$ there exists a *total weighting* $w : V(G) \cup E(G) \to \mathbb{R}$ of G such that $w(v) \in L_v$ for any vertex $v \in V(G)$ and $w(e) \in L_e$ for any edge $e \in E(G)$ and furthermore, such that for any pair of adjacent vertices u, v, we have $w(u) + \sum_{e \in E(u)} w(e) \neq w(v) + \sum_{e \in E(v)} w(e)$. In particular, any graph which is $(1,k)$-chooseable is also k-chooseable. This concept introduced by Wong and Zhu also generalizes the so-called *1-2 Conjecture* formulated in [6] which states that for any graph G there exists a total weighting $w : V(G) \cup E(G) \to \{1,2\}$ such that for any pair of adjacent vertices u, v, we have $w(u) + \sum_{e \in E(u)} w(e) \neq w(v) + \sum_{e \in E(v)} w(e)$. Wong and Zhu [10] proved that any graph is $(2,3)$-choosable. As mentioned above, the case of (k,m)-chooseability where $k = 1$ is particularly interesting since it directly relates to the 1-2-3 Conjecture. However, there is still no constant c known for which any graph without an isolated edge is $(1,c)$-choosable and the known results in this area mostly concern the maximum degree instead: Seamone showed in [7] that any graph G without an isolated edge is $(1, 2\Delta(G)+1)$-choosable and other linear bounds have also been proven in [3], [5] and [8]. The best result so far is the result by Ding et al. [2] mentioned by Wong and Zhu in [9] which says that any graph G without an isolated edge is $(1, \Delta(G)+1)$-choosable. The present paper shows that any graph G without an isolated edge is $(1, 2\lceil \log_2(\Delta(G)) \rceil + 1)$-choosable, replacing the linear term of $\Delta(G)$ by a logarithmic term. This is implied by a slightly stronger statement which is proved in the next section. The proof describes a linear time algorithm for finding appropriate edge weights.

2 $(1,\phi)$-choosablity

Let G be a graph, let k be a natural number and let $\phi : E(G) \to \mathbb{N}$ be a mapping. A (k,ϕ)*-list assignment* to G is an assignment of lists $L_e \subset \mathbb{R}$, $e \in E(G)$ to the edges of G such that the size of any list L_e is $\phi(e)$, together with an assignment of k-element lists $L_v \subset \mathbb{R}$, $v \in V(G)$ to the vertices. We say that G is (k,ϕ)*-choosable* if for any (k,ϕ)-list assignment to G there exists a total weighting $w : E(G) \cup V(G) \to \mathbb{R}$ of G such that for any edge $e = uv$ we have that $w(u) + \sum_{e \in E(u)} w(e) \neq w(v) + \sum_{e \in E(v)} w(e)$ and that $w(v) \in L_v$ for any vertex $v \in V(G)$ and $w(e) \in L_e$ for any edge $e \in E(G)$. Given a total weighting $w : E(G) \cup V(G) \to \mathbb{R}$ of a graph G and a vertex u in G the term $w(u) + \sum_{e \in E(u)} w(e)$ is also called the *colour* of u induced by w and is denoted by $C_w(u)$. If for two adjacent vertices u, v we have $C_w(u) = C_w(v)$, then we call this pair of vertices a *conflict*.

In the following we prove that any graph without isolated edges is $(1,\phi)$-choosable when $\phi : E(G) \to \mathbb{N}$ is defined by $\phi(e) = \lceil \log_2(d(u)) \rceil + \lceil \log_2(d(v)) \rceil + 1$ for $e = uv \in E(G)$. The proof describes an algorithm for finding appropriate edge weights and greedily assigns as small edge-weights as possible. This is done stepwise where in each step we choose a special vertex v and assign the smallest possible weights to all edges incident to v while increasing the weight on an edge in $E(u) \setminus E(v)$ for each neighbour u of v in order to avoid the potential conflicts between u and its neighbours. This greedy approach is the main idea of the algorithm, but some additional procedures are needed in order to ensure that we end up with no conflicts.

Theorem 1. *Any graph G without an isolated edge is $(1, \phi)$-choosable when $\phi : E(G) \to \mathbb{N}$ is defined by $\phi(uv) = \lceil \log_2(d(u)) \rceil + \lceil \log_2(d(v)) \rceil + 1$ for $uv \in E(G)$.*

Proof. Let G be a graph with n vertices and without any isolated edges. Let $e_1, \dots, e_m$ denote the edges of G. For any vertex v let s_v denote the prescribed weight (making up the list of size 1) on v and for $j = 1, \dots, m$ let $L_j = \{t_{j,1}, \dots, t_{j,\phi(e_j)}\}$ be a list associated with e_j and assume that the ordering is such that $t_{j,1} < \dots < t_{j,\phi(e_j)}$. We will, through a number of steps, recursively construct a sequence of total weight functions $w_i : V(G) \cup E(G) \to \mathbb{R}$ for $i = 0, \dots, k+1 \leqslant n+1$ where each w_{i+1} will be a modification of w_i and where w_{k+1} will be our final total weight function. All the total weight functions will agree with the lists assigned to the edges, that is, $w_i(e_j) \in L_j$ and $w_i(v) = s_v$ for all $i = 0, \dots, k+1$ and $j = 1, \dots, m$ and all vertices $v \in V(G)$. A "step" in the algorithm is when we move from considering w_i to considering w_{i+1}, so the algorithm will consist of $k+1$ steps and in each step we define a set of edges whose weights will never be changed again. This defines a sequence of edge sets $\emptyset = E_0 \subset E_1 \subset \dots \subset E_{k+1} = E(G)$. For each edge $e_j = uv$ we define three values $f_u(e_j) \in [0, \lceil \log_2(d(u)) \rceil]$, $f_v(e_j) \in [0, \lceil \log_2(d(v)) \rceil]$ and $f(e_j) = f_u(e_j) + f_v(e_j) + 1$. These values might be modified through the $k+1$ steps of the algorithm so for each edge e_j we let $f_{u,i}(e_j)$, $f_{v,i}(e_j)$ and $f_i(e_j) = f_{u,i}(e_j) + f_{v,i}(e_j) + 1$ denote the values within and after the i'th step. If nothing else is explicitly stated it will always be the case that $f_{u,i}(e_j) = f_{u,i-1}(e_j)$, $f_{v,i}(e_j) = f_{v,i-1}(e_j)$ and $f_i(e_j) = f_{u,i}(e_j) + f_{v,i}(e_j) + 1$.

We will also define a sequence of subsets of $V(G) \times E(G)$: $\emptyset = T_0 \subset T_1 \subset \dots \subset T_k$ during the first k steps of the algorithm. Each element (v', uv) of T_k will represent a triangle $v'uv$ in the graph where the only possible conflicts are between v' and u or v' and v. These potential conflicts will be the only possible conflicts after the first k steps of the algorithm and they will be disposed of in the last part of the algorithm.
In the algorithm we will in each of the first k steps choose at most four vertices and extend a vertex set V_i, which is initialized as $V_0 = \emptyset$, by adding those vertices. This will define a sequence of vertex sets $\emptyset = V_0 \subset V_1 \subset \dots \subset V_k$.

The algorithm consists of two parts: Procedure 1 followed by Procedure 2 described below. The first part, Procedure 1, is a greedy way to assign the edge-weights and allows us to keep track of potential conflicts. These conflicts will then be disposed of in Procedure 2.

Procedure 1 Greedy weight-choosing

Define $i = 1$, $E_0 = \emptyset$, $V_0 = \emptyset$, $T_0 = \emptyset$, $f_{u,0}(e_j) = f_{v,0}(e_j) = 0$ and $w_0(e_j) = t_{j,f_0(e_j)}$ for all $j = 1, \ldots, m$ and $w_0(v) = s_v$ for all vertices $v \in V(G)$.
while $E_i \neq E(G)$ **do**
  Choose a vertex v_i in the set $V(G) - V_{i-1}$ minimizing $C_{w_{i-1}}(v_i)$ and subject to that, incident to the fewest number of edges in $E(G) - E_{i-1}$.
  if $G - (E_{i-1} \cup E(v_i))$ contains no isolated edge uv where $C_{w_{i-1}}(u) = C_{w_{i-1}}(v)$ **then**
    Define $V_i = V_{i-1} \cup \{v_i\}$ and $E_i = E_{i-1} \cup E(v_i)$ and $T_i = T_{i-1}$.
    for each edge $v_i v$ in $E(v_i) - E_{i-1}$ **do**
      if $E(v) - E_i \neq \emptyset$ **then**
        Choose an edge $e = vw$ in $E(v) - E_i$ minimizing $f_{v,i-1}(e)$ and define
        $f_{v,i}(e) = f_{v,i-1}(e) + 1$.
    for any edge $e_j \in E(G)$ **do**
      Define $w_i(e_j) = t_{j,f_i(e_j)}$.
  if $G - (E_{i-1} \cup E(v_i))$ contains an isolated edge uv where $C_{w_{i-1}}(u) = C_{w_{i-1}}(v)$ **then**
    if u is adjacent to v_i and v is not adjacent to v_i as in Figure 1 **then**
      Define $V_i = V_{i-1} \cup \{v\}$ and $E_i = E_{i-1} \cup E(v)$ and $T_i = T_{i-1}$.
      Define $f_{u,i}(v_i u) = f_{u,i-1}(v_i u) + 1$.
      for any edge $e_j \in E(G)$ **do**
        Define $w_i(e_j) = t_{j,f_i(e_j)}$.
      if $C_{w_i}(v_i) = C_{w_i}(u)$ and uv_i is an isolated edge in $G - E_i$ **then**
        Define $f_{u,i}(uv) = f_{u,i-1}(uv) + 1$.
    if both u and v are adjacent to v_i **then**
      if v_i is not incident to an isolated edge $v_i v'$ in $G - (E_{i-1} \cup \{uv, v_i u, v_i v\})$ **then**
        $V_i = V_{i-1} \cup \{u, v\}$, $E_i = E_{i-1} \cup \{uv, v_i u, v_i v\}$, $T_i = T_{i-1} \cup \{(v_i, uv)\}$.
        Define $f_{u,i}(v_i u) = f_{u,i-1}(v_i u) + 1$.
        for any edge $e_j \in E(G)$ **do**
          Define $w_i(e_j) = t_{j,f_i(e_j)}$.
      if v_i is incident to an isolated edge $v_i v'$ in $G - (E_{i-1} \cup \{uv, v_i u, v_i v\})$ **then**
        $V_i = V_{i-1} \cup \{u, v, v_i, v'\}$, $E_i = E_{i-1} \cup \{uv, v_i u, v_i v, v_i v'\}$,
        $T_i = T_{i-1} \cup \{(v_i, uv)\}$.
        Define $f_{u,i}(v_i u) = f_{u,i-1}(v_i u) + 1$.
        if now $C_{w_i}(v_i) = C_{w_i}(v')$ **then**
          Redefine $f_{u,i}(v_i u) = f_{u,i-1}(v_i u) + 2$.
        for any edge $e_j \in E(G)$ **do**
          Define $w_i(e_j) = t_{j,f_i(e_j)}$.
  Replace i with $i + 1$.

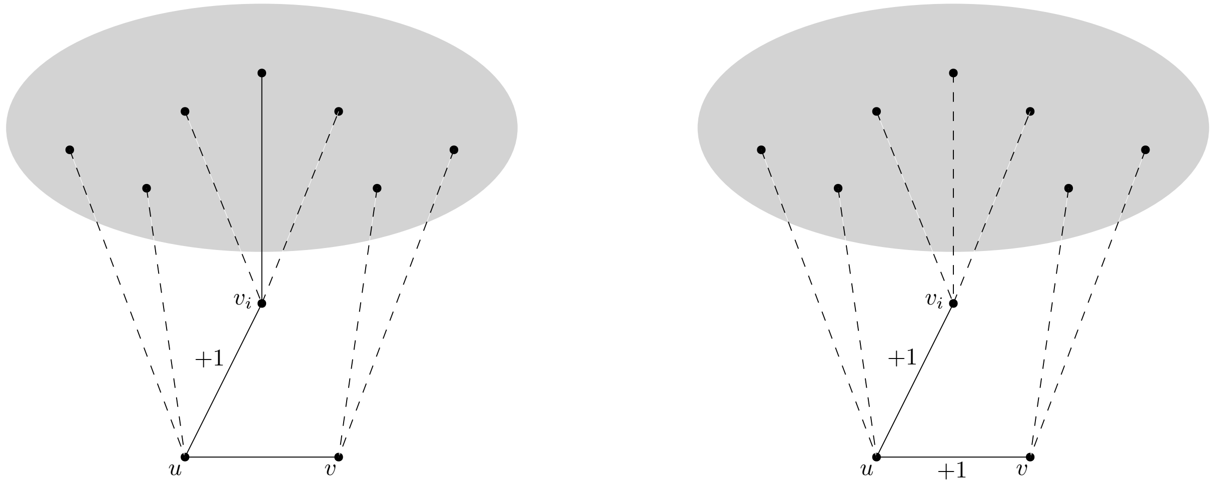

Figure 1: An illustration of the situation in line 13 in Procedure 1 (left) and of the situation in line 28 (right). Dashed edges indicate edges in E_{i-1}.

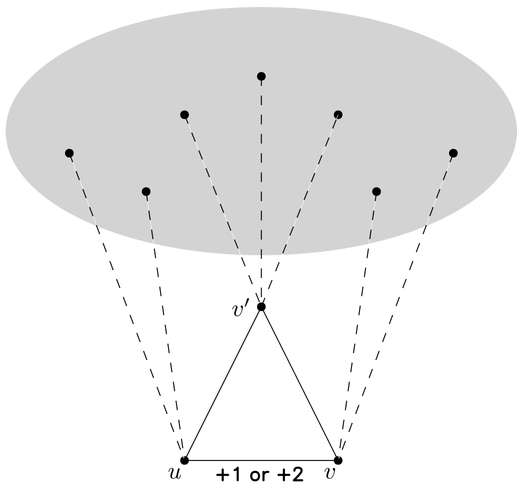

Figure 2: An illustration of Procedure 2.

When Procedure 1 terminates we have a well-defined weight function $w_k : E(G) \to \mathbb{R}$ and a set $T_k \subset V(G) \times E(G)$ representing some triangles in G. Let $(u_1, e_1), \ldots, (u_{|T_k|}, e_{|T_k|})$ denote the elements of T_k enumerated in the order they appear in Procedure 1. Note that when we repair conflicts in Procedure 2 below, we consider the triangles in T_k in reverse order starting with $(u_{|T_k|}, e_{|T_k|})$. When Procedure 2 terminates we have a weight function w_{k+1} and it remains to show that for any pair of adjacent vertices u, v we have $C_{w_{k+1}}(u) \neq C_{w_{k+1}}(v)$ and that $f_{k+1}(e) \leqslant \phi(e)$ holds for any edge $e \in E(G)$.

Procedure 2 Finalisation (Defining w_{k+1} repairing conflicts in triangles in T_k, see Figure 2).

for $i = |T_k| \dots 1$ **do**
  Define $(v', uv) = (u_i, e_i)$.
  if one of u, v, say, v has the same colour as v' **then**
    Define $f_{v,k+1}(uv) = f_{v,k}(uv) + 1$.
  if now u has the same colour as v' **then**
    Define $f_{v,k+1}(uv) = f_{v,k}(uv) + 2$.
  for any edge $e_j \in E(G)$ **do**
    Define $w_{k+1}(e_j) = t_{j,f_{k+1}(e_j)}$.

First we prove that for any edge uv we have $C_{w_{k+1}}(u) \neq C_{w_{k+1}}(v)$. To do this we look at three different cases:

1. $(v', uv) \notin T_k$ for all $v' \in V(G)$ and $(u, e') \notin T_k$ and $(v, e') \notin T_k$ for all $e' \in E(u) \cup E(v)$.

2. $(v', uv) \in T_k$ for some $v' \in V(G)$.

3. $(u, e') \in T_k$ or $(v, e') \in T_k$ for some $e' \in E(u) \cup E(v)$.

Case 1:
We look at two separate subcases.

Subcase 1.1: For some $i \leqslant k$ the edge uv is isolated in $G - E_i$.

Let $i \leqslant k$ be the smallest index such that uv is an isolated edge in $G - E_i$. In a later loop of Procedure 1 one of u, v, say u, is chosen as the vertex with minimum potential. That is, for some smallest $i' > i$ we have $u = v_{i'}$, $v \notin V_{i'}$ and $u \notin V_{i'-1}$. Since uv is an isolated edge in $G - E_i$ and hence also in $G - E_{i'-1}$ it follows from lines 4-11 in Procedure 1 that in the i''th loop of Procedure 1 no edge-weights changed and $E_{i'} = E_{i'-1} \cup \{uv\}$. Also the weight of uv does not change during Procedure 2. Thus, $C_{w_i}(u) = C_{w_k}(u) = C_{w_{k+1}}(u)$ and $C_{w_i}(v) = C_{w_k}(v) = C_{w_{k+1}}(v)$, so it suffices to show that $C_{w_i}(u) \neq C_{w_i}(v)$. If the if-statement in line 4 of Procedure 1 was satisfied in the i'th loop $C_{w_i}(u) \neq C_{w_i}(v)$ follows immediately, so we can assume that the if-statement in line 12 was satisfied in the i'th loop of Procedure 1. Furthermore, if the if-statement in line 20 was satisfied, then it follows from the lines 20-33, that any isolated edge in $G - E_i$ is also an isolated edge in $G - E_{i-1}$ and this contradicts the choice of i. Thus, we can assume that the if-statement in line 13 was satisfied in the i'th loop of Procedure 1. Now it follows from lines 13-19 in Procedure 1 that $C_{w_i}(u) \neq C_{w_i}(v)$.

Subcase 1.2: For all $i \leqslant k$ the edge uv is not isolated in $G - E_i$.

Let $i \leqslant k$ be the smallest index such that $uv \in E_i$. Without loss of generality we can assume that $v \notin V_{i-1}$, $v \in V_i$ and $u \notin V_{i-1}$. If also $u \in V_i$, then since $(v', uv) \notin T_k$ for all $v' \in V(G)$, it follows from Procedure 1 that the if-statements in lines 12, 20 and 26 were satisfied in the i'th loop of Procedure 1 and that uv is a pendant edge in a component of $G - E_{i-1}$ which is isomorphic to a triangle with a pendant edge added. In this case it follows from lines 26-33 in Procedure 1 that $C_{w_i}(u) \neq C_{w_i}(v)$ and since $E(u) \cup E(v) \subset E_i$ this implies that $C_{w_k}(u) \neq C_{w_k}(v)$. Furthermore, since $(v', uv) \notin T_k$ for all $v' \in V(G)$ and $(u, e') \notin T_k$ and $(v, e') \notin T_k$ for all $e' \in E(u) \cup E(v)$, the weight of u or v does not change in Procedure 2 and hence $C_{w_{k+1}}(u) \neq C_{w_{k+1}}(v)$. Thus we can assume $u \notin V_i$ and since $(v', uv) \notin T_k$ for all $v' \in V(G)$ and $(u, e') \notin T_k$ and $(v, e') \notin T_k$ for all $e' \in E(u) \cup E(v)$ we can assume that either the if-statement in line 4 or both the if-statements in lines 12 and 13 in Procedure 1 were satisfied in the i'th loop of Procedure 1. If the if-statement in line 4 was satisfied then $C_{w_i}(v) < C_{w_i}(u)$ follows from lines 4-11 in Procedure 1 since uv is not an isolated edge in $G - E_{i-1}$. Also if the if-statements in lines 12 and 13 were satisfied $C_{w_i}(v) < C_{w_i}(u)$ follows from lines 12-17 in Procedure 1. Thus we have that $C_{w_i}(v) < C_{w_i}(u)$. More over in both cases, $C_{w_{k+1}}(v) = C_{w_i}(v)$ and $(x, yv) \notin T_k$ for all $x, y \in V(G)$, and hence $C_{w_{k+1}}(v) = C_{w_i}(v) < C_{w_i}(u) \leqslant C_{w_{k+1}}(u)$.

Case 2: Let i be the smallest index such that $(v', uv) \in T_i$ for some $v' \in V(G)$. Since we put (v', uv) into T_i we have $C_{w_{i-1}}(u) = C_{w_{i-1}}(v)$. By lines 20-33 in Procedure 1, we increased the value of $C_{w_{i-1}}(u)$ to make sure that $C_{w_i}(u) \neq C_{w_i}(v)$ and never changed these two values before Procedure 2. Also, it follows from the lines 2-6 in Procedure 2 that we can only change the value of $w_k(uv)$, but not $w_k(uv')$ or $w_k(vv')$ in the finalisation. Thus we have that

$$C_{w_{k+1}}(u) = C_{w_i}(u) - w_i(uv) + w_{k+1}(uv) \neq C_{w_i}(v) - w_i(uv) + w_{k+1}(uv) = C_{w_{k+1}}(v).$$

Case 3: Assume that $(u, e') \in T_k$ and $e' = vv'$. At some point in Procedure 2 the triangle (u, e') is considered. Note that there might exist a vertex u' and an edge e'' incident to u such that $(u', e'') \in T_k$. If this is the case then that triangle (u', e'') appeared later than (u, e') in Procedure 1 and is therefore considered earlier than (u, e') in Procedure 2 (see Figure 3). This implies that at the time Procedure 2 reaches (u, e') and throughout the rest of Procedure 2 the colour of u does not change. By lines 2-6 in Procedure 2 we change the value of $w_k(e')$ ensuring $C_{w_{k+1}}(u) \neq C_{w_{k+1}}(v)$ as well as $C_{w_{k+1}}(u) \neq C_{w_{k+1}}(v')$. So $C_{w_{k+1}}(u) \neq C_{w_{k+1}}(v)$.

It remains to show that $f_{k+1}(e) \leqslant \phi(e) = \lceil \log_2(d(u)) \rceil + \lceil \log_2(d(v)) \rceil + 1$ holds for any edge $e = uv$ in G. This time we also look at the three different cases mentioned above:

Case 1: Let ℓ be the smallest index such that $uv \in E_\ell$. We may without loss of generality assume $v \notin V_{\ell-1}$, $v \in V_\ell$ and $u \notin V_{\ell-1}$. We start by looking at how large $f_{u,\ell-1}(e)$ can possibly be. This is the number of times $f_{u,i}(e)$ (for $i = 0, \ldots, \ell - 1$) has increased during Procedure 1 before the step where uv was added to E_ℓ. Suppose we increase $f_{u,i-1}(e)$

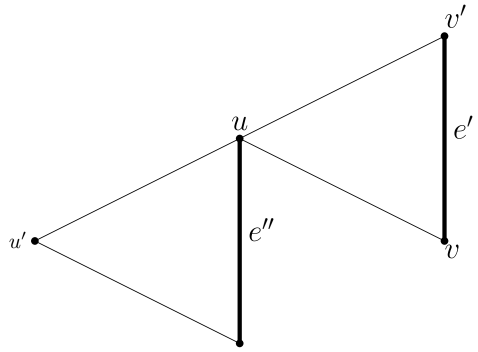

Figure 3: An illustration of how two triangles (u', e'') and (u, e') in T_k can appear in G. In this case (u', e'') will be considered before (u, e') in Procedure 2.

in the steps $i = i_1, i_2, \ldots, i_{f_{u,\ell-1}(e)}$. Since we are interested in an upper bound for $f_{u,i}(e)$ we may assume that in any step j' where Procedure 1 chose a vertex in $N(u)$ as $v_{j'}$ and e minimized $f_{u,j'-1}(x)$ for $x \in E(u) - E_{j'}$, the edge e was chosen (even if there where multiple minimizers) in line 8 in Procedure 1. Note that this implies that in each of the steps i_j for $j \in \{1, \ldots, f_{u,\ell-1}(e)\}$ the term $f_{u,i_j-1}(x)$ is constant for $x \in E(u) - E_{i_j}$.
In step i_1 a vertex in $N(u)$ was picked as v_{i_1} and put into V_{i_1} and $f_{u,i_1-1}(e)$ was increased by 1. Note that by the above we can assume that $V_{i_1} \cap N(u) = \{v_{i_1}\}$. In step i_2 another vertex in $N(u)$ was picked as v_{i_2} and $f_{u,i_2-1}(e)$ was increased because $f_{u,i_2-1}(x)$ was constant for $x \in E(u) - E_{i_2}$. Since $f_{u,i_2-1}(e) = 1$ it follows that at least $\left\lfloor \frac{d(u)}{2} \right\rfloor$ of the edges incident to u were in E_{i_2-1}, see Figure 4. Similarly, for step i_3 we have $|(E(u) - E_{i_2}) \cap E_{i_3-1}| \geqslant \left\lfloor \frac{|E(u)-E_{i_2}|-1}{2} \right\rfloor$. Hence

$$\begin{aligned}
|E(u) \cap E_{i_3-1}| &= |E(u) \cap E_{i_2}| + |(E(u) - E_{i_2}) \cap E_{i_3-1}| \\
&\geqslant |E(u) \cap E_{i_2}| + \left\lfloor \frac{|E(u) - E_{i_2}| - 1}{2} \right\rfloor \\
&= |E(u) \cap E_{i_2-1}| + 1 + \left\lfloor \frac{|E(u) - E_{i_2-1}| - 2}{2} \right\rfloor \\
&= |E(u) \cap E_{i_2-1}| + \left\lfloor \frac{|E(u) - E_{i_2-1}|}{2} \right\rfloor
\end{aligned}$$

Note this is a non-decreasing function of $|E(u) \cap E_{i_2-1}|$, we have

$$\begin{aligned}
|E(u) \cap E_{i_3-1}| &= |E(u) \cap E_{i_2-1}| + \left\lfloor \frac{|E(u) - E_{i_2-1}|}{2} \right\rfloor \\
&\geqslant \left\lfloor \frac{d(u)}{2} \right\rfloor + \left\lfloor \frac{\frac{d(u)}{2}}{2} \right\rfloor \\
&= \sum_{r=1}^{2} \left\lfloor \frac{d(u)}{2^r} \right\rfloor .
\end{aligned}$$

We continue counting in this way and we get the following for all $j = 1, \dots, f_{u,\ell-1}(e)$:

$$|E(u) \cap E_{i_j-1}| \geqslant \sum_{r=1}^{j-1} \left\lfloor \frac{d(u)}{2^r} \right\rfloor \quad \text{and} \quad \left\lfloor \frac{d(u)}{2^{j-1}} \right\rfloor > 0.$$

Furthermore, note that for all $j \in \{1, \dots, f_{u,\ell-1}(e)\}$ we have $|E(u) \cap E_{i_j-1}| < d(u) - 1$ since $uv \notin E_{i_j-1}$ and $uw \notin E_{i_j-1}$ for some $w \in N(u) \setminus \{v\}$ (where $w \in N(u)$ is the vertex we choose to put into V_{i_j} in step i_j). Thus we have

$$\sum_{r=1}^{f_{u,\ell-1}(e)-1} \left\lfloor \frac{d(u)}{2^r} \right\rfloor < d(u) - 1,$$

which together with $\left\lfloor \frac{d(u)}{2^{f_{u,\ell-1}(e)-1}} \right\rfloor > 0$ implies $f_{u,\ell-1}(e) \leqslant \lceil \log_2(d(u)) \rceil$. We can repeat the above analysis for $f_{u,\ell-1}(e)$ and get $f_{v,\ell-1}(e) \leqslant \lceil \log_2(d(v)) \rceil$. If none of $f_{u,\ell-1}(e)$, $f_{v,\ell-1}(e)$ increases in step ℓ of Procedure 1 we now get

$$f_{k+1}(e) = f_{\ell-1}(e) = f_{u,\ell-1}(e) + f_{v,\ell-1}(e) + 1 \leqslant \lceil \log_2(d(u)) \rceil + \lceil \log_2(d(v)) \rceil + 1 = \phi(e).$$

Thus, we may assume that one of $f_{u,\ell-1}(e)$, $f_{v,\ell-1}(e)$, say $f_{u,\ell-1}(e)$ increases in step ℓ of Procedure 1. Since $(u, e') \notin T_k$ and $(v, e') \notin T_k$ for all $e' \in E(u) \cup E(v)$ it must be that the if-statement in lines 12, 13 and 18 were satisfied in the ℓ'th loop of Procedure 1 and u is a vertex of degree 2 in $G - E_{\ell-1}$ and v is a vertex of degree 1 in $G - E_{\ell-1}$. In this case we have $|E(u) \cap E_{i_j-1}| < d(u) - 2$ for all $j \in \{1, \dots, f_{u,\ell-1}(e)\}$ and so we get:

$$\sum_{r=1}^{f_{u,\ell-1}(e)-1} \left\lfloor \frac{d(u)}{2^r} \right\rfloor < d(u) - 2,$$

which together with $\left\lfloor \frac{d(u)}{2^{f_{u,\ell-1}(e)-1}} \right\rfloor > 0$ implies $f_{u,\ell-1}(e) \leqslant \lceil \log_2(d(u)) \rceil - 1$. Hence

$$\begin{aligned}
f_{k+1}(e) = f_\ell(e) &= f_{u,\ell}(e) + f_{v,\ell}(e) + 1 \\
&= f_{u,\ell-1}(e) + 1 + f_{v,\ell-1}(e) + 1 \\
&\leqslant \lceil \log_2(d(u)) \rceil - 1 + 1 + \lceil \log_2(d(v)) \rceil + 1 \\
&= \phi(e)
\end{aligned}$$

Case 2: Let i be the smallest index such that $(v', uv) \in T_i$ for some $v' \in V(G)$. As in Case 1, since $|E(u) - E_{i-1}| = 2$ we have $f_{u,i-1}(e) \leqslant \lceil \log_2(d(u)) \rceil - 1$. Similarly

$$f_{v,k}(e) = f_{v,i-1}(e) \leqslant \lceil \log_2(d(v)) \rceil - 1,$$

thus $f_k(e) \leqslant \lceil \log_2(d(u)) \rceil + \lceil \log_2(d(v)) \rceil - 2 + 1$. Within Procedure 2, we increase $f_k(uv)$ at most twice, so $f_{k+1}(e) \leqslant \lceil \log_2(d(u)) \rceil + \lceil \log_2(d(v)) \rceil + 1 \leqslant \phi(e)$.

Case 3: In this case without loss of generality we may assume there is a vertex v' and an edge $e' = vv'$ such that $(u, e') \in T_k$. Let i be the index in Procedure 1 where we put v

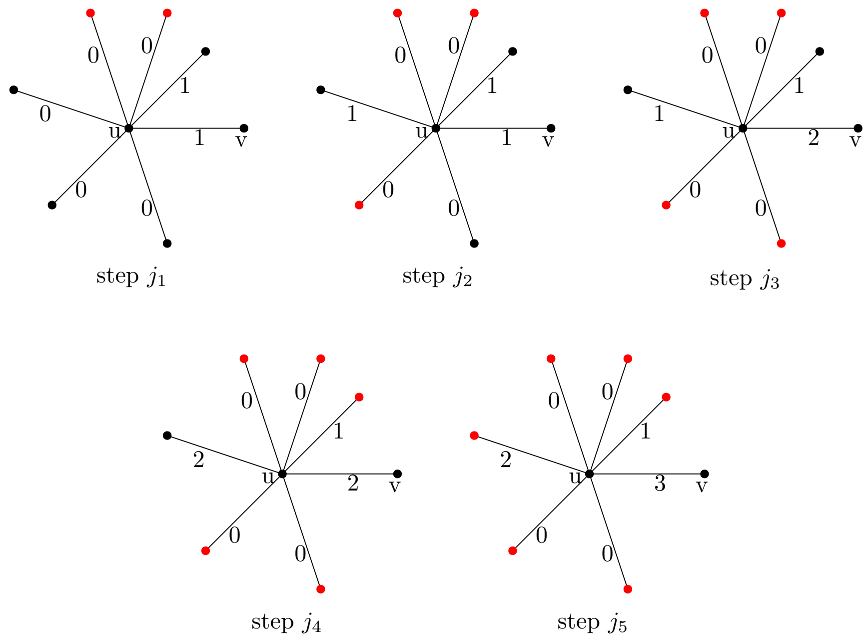

Figure 4: An illustration of how edge weights can increase during Procedure 1. The five graphs illustrate the same vertices in five different steps $j_1, \ldots, j_5$ in the algorithm. A number on an edge e indicates how many times $f_u(e)$ has been increased and the red colour indicates vertices belonging to $V_{j_1}, \ldots, V_{j_5}$. The five shown steps illustrate how the neighbours of u are, one by one, added into $V_{j_1}, \ldots, V_{j_5}$ in such a way that $f_u(uv)$ is increased as many times as possible. This can be thought of as a worst case scenario for $f_u(uv)$.

and v' together into V_i. At this step in Procedure 1 it follows from the same arguments as in Case 1 that $f_{u,i-1}(e) \leqslant \lceil \log_2(d(u)) \rceil - 1$ as well as $f_{v,i-1}(e) \leqslant \lceil \log_2(d(v)) \rceil - 1$, which means $f_{i-1}(e) \leqslant \lceil \log_2(d(u)) \rceil + \lceil \log_2(d(v)) \rceil - 2 + 1$. Furthermore, in step i we increase $f_{i-1}(e)$ at most twice and never change its value afterwards, thus $f_{k+1}(e) \leqslant \phi(e)$. $\square$

Acknowledgements

The authors would like to thank John Gimbel and referees for careful reading of the manuscript.

References

[1] T. Bartnicki, J. Grytczuk, S. Niwczyk. Weight Choosability of Graphs, Journal of Graph Theory. 60 (3) (2009), 242-256.

[2] L. Ding, A. Duh, G. Wang, T. Wong, J. Wu, X. Yu, X. Zhu. Graphs are $(1, \Delta + 1)$-choosable, manuscript, (2015).

[3] J. Fu, G. Wang, J. Wu, J. Xu. A note on edge weight choosability of graphs, Discrete Mathematics Algorithms and Applications 6 (1) (2014) 1450010.

[4] M. Karoński, T. Łuczak, A. Thomason. Edge weights and vertex colours, Journal of Combinatorial Theory Series B. 91 (2004), 151-157.

[5] H. Pan, D. Yang. On total weight choosability of graphs. Journal of Combinatorial Optimization 25 (2013) 766-783.

[6] J. Przybyło, M. Woźniak. On a 1,2 conjecture, Discrete Mathematics and Theoretical Computer Science. 12 (1) (2010), 101-108.

[7] B. Seamone. Bounding the monomial index and (1,l)-weight choosability of a graph, Discrete Mathematics and Theoretical Computer Science. 16 (3) (2014), 173-188.

[8] G. Wang, G. Yan. An Improved Upper Bound on Edge Weight Choosability of Graphs, Graphs and Combinatorics 31 (5) (2014), 1789–1793.

[9] T. Wong, X. Zhu. Total weight choosability of d-degenerate graphs, arXiv:1510.00809.

[10] T. Wong, X. Zhu. Every graph is (2,3)-choosable, Combinatorica 36 (1) (2016), 121-127.