

Equitable d -degenerate choosability of graphs

join work with E. Drgas-Burchardt, E. Sidorowicz

Hanna Furmańczyk
Division of Combinatorial Optimization, Institute of Informatics
University of Gdańsk, Poland

31st International Workshop on Combinatorial Algorithms
IWOCA 2020

some earlier papers, eg.

Zhang, X., Equitable list point arboricity of graphs. *Filomat* (2016)

some earlier papers, eg.

Zhang, X., Equitable list point arboricity of graphs. *Filomat* (2016)

+

Drgas-Burchardt, E., Dybizbański, J., Furmańczyk, H., Sidorowicz, E.: Equitable List Vertex Colourability and Arboricity of Grids. *Filomat* (2018)

some earlier papers, eg.

Zhang, X., Equitable list point arboricity of graphs. *Filomat* (2016)

+

Drgas-Burchardt, E., Dybizba«ski, J., Furma«czyk, H., Sidorowicz, E.: Equitable List Vertex Colourability and Arboricity of Grids. *Filomat* (2018)

application given in: Zhang, X., Niu, B., Li, Y., Li, B.: Equitable vertex arboricity of d -degenerate graphs, [arxiv: 1908.05066v1] (2019)

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d -degenerate coloring (H) d for every subgraph H of each color class

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d -degenerate coloring (H) d for every subgraph H of each color class

the sizes of pieces are balanced equitable coloring (it might be helpful in maintaining the network)

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d -degenerate coloring (H) d for every subgraph H of each color class

the sizes of pieces are balanced equitable coloring (it might be helpful in maintaining the network)

some additional requirements on nodes list coloring

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d -degenerate coloring (H) d for every subgraph H of each color class

the sizes of pieces are balanced **equitable coloring** (it might be helpful in maintaining the network)

some additional requirements on nodes **list coloring**

Kostochka, A. V., Pelsmajer, M. J., West, D. B. (2003)

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d-degenerate coloring (H) d for every subgraph H of each color class

the sizes of pieces are balanced **equitable coloring** (it might be helpful in maintaining the network)

some additional requirements on nodes **list coloring**

Zhang, X. (2016)

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d-degenerate coloring (H) Δ for every subgraph H of each color class

the sizes of pieces are balanced **equitable coloring** (it might be helpful in maintaining the network)

some additional requirements on nodes **list coloring**

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d-degenerate coloring (H) d for every subgraph H of each color class

proper coloring 0-degenerate coloring

the sizes of pieces are balanced **equitable coloring** (it might be helpful in maintaining the network)

some additional requirements on nodes **list coloring**

decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)

proper coloring the pieces are independent

arboricity the pieces are acyclic

d-degenerate coloring (H) d for every subgraph H of each color class

proper coloring 0-degenerate coloring

acyclic coloring 1-degenerate coloring

the sizes of pieces are balanced **equitable coloring** (it might be helpful in maintaining the network)

some additional requirements on nodes **list coloring**

G is d -degenerate if $(H) \leq d$ for any subgraph H of G ,

G is **d-degenerate** if $\Delta(H) \leq d$ for any subgraph H of G ,
 D_d - the class of all d -degenerate graphs,

G is **d-degenerate** if $(H) \leq d$ for any subgraph H of G ,

D_d - the class of d -degenerate graphs,

D_d -coloring $c : V(G) \rightarrow \mathbb{N}$ such that for each $2 \leq N$ the set of vertices coloured with $\leq N$ induces a d -degenerate graph

G is **d-degenerate** if $(H) \leq d$ for any subgraph H of G ,

D_d - the class of all d -degenerate graphs,

D_d -coloring $c : V(G) \rightarrow N$ such that for each $i \in N$ the set of vertices coloured with i induces a d -degenerate graph

list assignment L : a mapping that assigns a nonempty subset of N to each vertex $v \in V(G)$; given $k \in N$, a list assignment L is **k -uniform** if $|L(v)| = k$ for every $v \in V(G)$,

$(L; D_d)$ -colourability

Given $d \in \mathbb{N}_0$, a graph G is $(L; D_d)$ -colourable if there exists a colouring $c : V(G) \rightarrow \mathbb{N}$, such that $c(v) \in L(v)$ for each $v \in V(G)$, and for each $i \in \mathbb{N}$ the set of vertices coloured with i induces a d -degenerate graph.

$(L; D_d)$ -colourability

Given $d \in \mathbb{N}_0$, a graph G is $(L; D_d)$ -colourable if there exists a colouring $c : V(G) \rightarrow \mathbb{N}$, such that $c(v) \in L(v)$ for each $v \in V(G)$, and for each $i \in \mathbb{N}$ the set of vertices coloured with i induces a d -degenerate graph.

Equitable $(k; D_d)$ -choosability

Given $k \in \mathbb{N}$ and $d \in \mathbb{N}_0$, a graph G is $(k; D_d)$ -equitably choosable if for any k -uniform list assignment L there is an $(L; D_d)$ -colouring of G such that the size of any colour class $|V_i| = k$.

Definition

Let $k, d \in \mathbb{N}$. A partition $S_1 \sqcup \dots \sqcup S_{k+1}$ of $V(G)$ is called a **$(k; d)$ -partition** of G if $|S_j| \leq k$, and $|S_j| = k$ for $j \in \{2, \dots, k+1\}$, and for each j , there is such an ordering $\{x_1^j, \dots, x_k^j\}$ of vertices of S_j that

$$|N_G(x_i^j) \setminus (S_1 \sqcup \dots \sqcup S_{j-1})| \leq d-1; \quad i \in [k]:$$

Definition

Let $k, d \in \mathbb{N}$. A partition $S_1 \sqcup \dots \sqcup S_{r+1}$ of $V(G)$ is called a **$(k; d)$ -partition** of G if $|S_j| = k$, and $|S_j^i| = d$ for $j \in \{1, \dots, r+1\}$, and for each j , there is such an ordering $\{x_1^j, \dots, x_k^j\}$ of vertices of S_j that

$$|N_G(x_i^j) \cap (S_1 \sqcup \dots \sqcup S_{j-1})| = d - 1; \quad i \in [k]:$$

Some observation

each $(k; d)$ -partition of G is also its $(k; d + 1)$ -partition,

Definition

Let $k, d \in \mathbb{N}$. A partition $S_1 \sqcup \dots \sqcup S_{k+1}$ of $V(G)$ is called a **$(k; d)$ -partition** of G if $|S_j| \leq k$, and $|S_j| = k$ for $j \in \{2, \dots, k+1\}$, and for each j , there is such an ordering $\{x_1^j, \dots, x_k^j\}$ of vertices of S_j that

$$|N_G(x_i^j) \setminus (S_1 \sqcup \dots \sqcup S_{j-1})| \leq d-1; \quad i \in [k]:$$

Some observation

each $(k; d)$ -partition of G is also its $(k; d+1)$ -partition, if G has a $(k; d)$ -partition, then G may have neither a $(k+1; d)$ -partition nor a $(k-1; d)$ -partition (cf. Example 1)

Theorem 1

Let $k, d, t \in \mathbb{N}$ and $t \leq k$. If a graph G has a $(k; d)$ -partition, then it is equitably $(t; D_{d-1})$ -choosable.

Theorem 1

Let $k, d, t \in \mathbb{N}$ and $t \geq k$. If a graph G has a $(k; d)$ -partition, then it is equitably $(t; D_{d-1})$ -choosable.

Theorem 2

There is a polynomial-time algorithm that for any graph with a given $(k; d)$ -partition and for any t -uniform list assignment L returns an equitable $(t; D_{d-1})$ -colouring of G .

Proposition (generalization from $(k; D_0)$ - and $(k; D_1)$ -choosability)

Let $k, d \geq 2$ and let S be a set of distinct vertices x_1, \dots, x_k of a graph G . If $G - S$ is equitably $(k; D_{d-1})$ -choosable and

$$|N_G(x_i) \cap S| \leq d - 1$$

holds for every $i \in [k]$, then G is equitably $(k; D_d)$ -choosable.

Proposition (generalization from $(k; D_0)$ - and $(k; D_1)$ -choosability)

Let $k, d \geq 2$ and let S be a set of distinct vertices x_1, \dots, x_k of a graph G . If $G - S$ is equitably $(k; D_{d-1})$ -choosable and

$$|N_G(x_i) \cap S_j| \leq d - 1$$

holds for every $j \in [k]$, then G is equitably $(k; D_{d-1})$ -choosable.

Some observations

if G has a $(k; d)$ -partition, then G is eq. $(k; D_{d-1})$ -choosable,

Proposition (generalization from $(k; D_0)$ - and $(k; D_1)$ -choosability)

Let $k, d \geq 2$ and let S be a set of distinct vertices x_1, \dots, x_k of a graph G . If $G - S$ is equitably $(k; D_{d-1})$ -choosable and

$$|N_G(x_i) \cap S_j| \leq d-1$$

holds for every $j \in [k]$, then G is equitably $(k; D_{d-1})$ -choosable.

Some observations

if G has a $(k; d)$ -partition, then G is eq. $(k; D_{d-1})$ -choosable,
the equitable $(k; D_{d-1})$ -choosability of G does not imply the
equitable $(t; D_{d-1})$ -choosability of G for $t \leq k$

Theorem 1

Let $k, d, t \in \mathbb{N}$ and $t \geq k$. If a graph G has a $(k; d)$ -partition, then it is equitably $(t; D_{d-1})$ -choosable.

Theorem 2

There is a polynomial-time algorithm that for any graph with a given $(k; d)$ -partition and for any t -uniform list assignment L returns an equitable $(t; D_{d-1})$ -colouring of G .

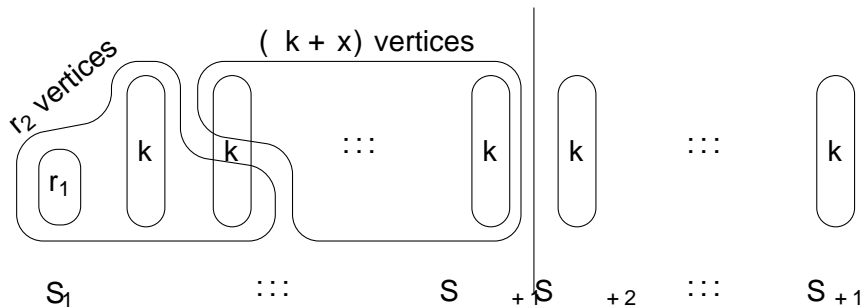
Algorithm 1: Equitable($L; D_{d-1}$)-colouring(G)

Input : Graph G on n vertices; L - t -uniform list assignment;
a $(k; d)$ -partition $S_1 \sqcup \dots \sqcup S_{k+1}$ of G , given by lists
 $S_1 = (x_1^1; \dots; x_{r_1}^1)$ and $S_j = (x_1^j; \dots; x_k^j)$ for $j = 2; \dots$
Output: Equitable($L; D_{d-1}$)-colouring of G

```

1 for j := 1 to t + 1 do
2   | add reverse (S_j) to S;
3 end
4
5 t := t - k; r := t (mod k); r := r + k; x := r (mod k);
6 take and delete r_2 elements from the beginning of S, and add them to L_R;
7 colour_List (L_R; r_2);
8 take and delete x elements from the beginning of S, and add them to L_X ;
9 colour_List (L_X; x);
10 S_col := L_X;
11 for j = 1 to t do
12   | take and delete k elements from the beginning of S, and add to S^0;
13   | colour_List (S^0; k);
14   | S_col := S_col + S^0;
15 end
16 Reorder (S_col);
17 S_bar := S; Modify_colourLists (S_col; S_bar);
18 colour_List (S; k);

```



$$|V(G)| = k + r_1$$

$$|V(G)| = t + r_2; \quad 0 < r_2 \leq t$$

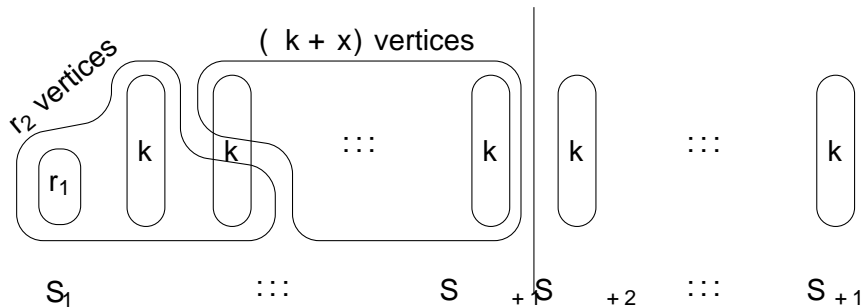
$$|V(G)| = k + (k+x) + r_2$$

we create the list S , consisting of the elements from $M(G)$, whose order corresponds to the order in which the colouring is expanded to successive vertices in S (Prop. 1)

we create the list S , consisting of the elements from $V(G)$, whose order corresponds to the order in which the colouring is expanded to successive vertices in S (Prop. 1)

we colour v_2 vertices from the beginning of S taking into account the lists of available colours; we delete the colour assigned to v from the lists of available colours for vertices from $N_G^{\text{col}}(d; v)$

(f $w \in N_G(v)$: w has neighbors coloured with $(v)g$)
 (in consequence we have rainbow colored vertices)



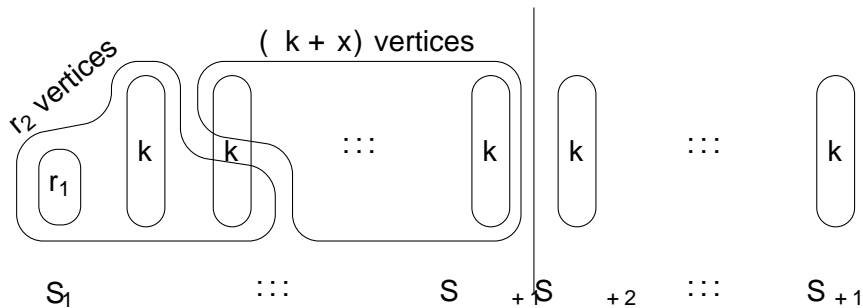
$$|V(G)| = k + r_1$$

$$|V(G)| = t + r_2; \quad 0 < r_2 < t$$

$$r := t \bmod k$$

$$|V(G)| = k + (k+x) + r_2$$

we colour $k + x$ vertices taking into account the lists of available colours in such a way that every sublist of length k formed by vertices coloured differently (consequently, every sublist of length r is coloured differently); we divide the vertices colored here into sets each one of cardinality x ; we delete $c(v)$ from the lists of vertices from $N_G^{\text{col}}(d; v)$;



$$|V(G)| = k + r_1$$

$$|V(G)| = t + r_2; \quad 0 < r_2 \leq t$$

$$|V(G)| = k + (k + x) + r_2$$

we extend the list colouring into the uncoloured k vertices by colouring groups of k vertices; first, we associate each group of k vertices with a set of vertices coloured in the previous step (for different groups these sets are disjoint); next, we color the vertices of each of the group using different colors that are also different from the colors of vertices of the set associated with this group;

we extend the list colouring into the uncoloured k vertices by colouring groups of k vertices; next, we associate each group of k vertices with a set of vertices coloured in the previous step (for different groups these sets are disjoint); next, we color the vertices of each of the group using different colors that are also different from the colors of vertices of the set associated with this group;

our final equitable list colouring is the consequence of a partition of $V(G)$ into $t + 1$ coloured sets, each one of size at most t and each one formed by vertices coloured differently.

Theorem 3

For a given graph G on n vertices, a d -uniform list assignment L , a $(k; d)$ -partition of G the Equitable $(L; D_{d-1})$ -colouring (G) algorithm returns $(L; D_{d-1})$ -colouring of G in $O(n^2(G))$ time.

How to check whether G has $(k; d)$ -partition?

Grid $P_5 \times P_3 \times P_2$.

Theorem 4

For a given 3-dimensional grid G the Partition3d (G) algorithm returns a $(3; 2)$ -partition of G in polynomial-time.

Theorem 4

For a given 3-dimensional grid G the Partition3d(G) algorithm returns a $(3; 2)$ -partition of G in polynomial-time.

Algorithm 3: Partition3d(G)

Input : 3-dimensional grid $G = P_{n_1} \cup P_{n_2} \cup P_{n_3}$.

Output: A $(3; 2)$ -partition $S_1 \cup \dots \cup S_{s+1}$ of G .

Theorem 4

For a given 3-dimensional grid G the Partition3d(G) algorithm returns a (3; 2)-partition of G in polynomial-time.

Algorithm 4: Partition3d(G)

Input : 3-dimensional grid $G = P_{n_1} \cup P_{n_2} \cup P_{n_3}$.

Output: A (3; 2)-partition $S_1 \cup \dots \cup S_{+1}$ of G .

(3; 2)-partition of G :

$S_1 \cup \dots \cup S_{+1}$, $|S_j| \leq 3$, and $|S_j| = 3$ for $j \in \{2, \dots, +1\}$, and for each j , there is such an ordering $\{x_1^j, x_2^j, x_3^j\}$ of vertices of S_j that

$$|N_G(x_i^j) \setminus (S_1 \cup \dots \cup S_{j-1})| \leq 2 \quad \forall i \in \{1, 2, 3\};$$

Theorem 5

Let $t \geq 3$ be an integer. Every 3-dimensional grid is equitably $(t; D_1)$ -choosable. Moreover, there is a polynomial-time algorithm that for every t -uniform list assignment L of the 3-dimensional grid G returns an equitable $(L; D_1)$ -colouring of G .

What is the complexity of recognition of graphs having a $(k; d)$ -partition?

What is the complexity of recognition of graphs having a $(k; d)$ -partition?

What is the parameterized complexity of the problem of Equitable $(L; D_d)$ -Choosability in respect to different parameters?

