

# 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., Dybizbański, J., Furmańczyk, 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 of large-scale networks

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

# Decomposition of large-scale networks

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent

# Decomposition of large-scale networks

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic

# Decomposition of large-scale networks

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic
  - $d$ -degenerate coloring  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class



# Decomposition of large-scale networks

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic
  - $d$ -degenerate coloring  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class
- the sizes of pieces are balanced  $\equiv$  equitable coloring (it might be helpful in maintaining the network)

# Decomposition of large-scale networks

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic
  - $d$ -degenerate coloring  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class
- the sizes of pieces are balanced  $\equiv$  equitable coloring (it might be helpful in maintaining the network)
- some additional requirements on nodes  $\equiv$  list coloring

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - **proper coloring**  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic
  - $d$ -degenerate coloring  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class
- the sizes of pieces are balanced  $\equiv$  **equitable coloring** (it might be helpful in maintaining the network)
- some additional requirements on nodes  $\equiv$  **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  $\equiv$  the pieces are independent
  - **arboricity**  $\equiv$  the pieces are acyclic
  - $d$ -degenerate coloring  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class
- the sizes of pieces are balanced  $\equiv$  **equitable coloring** (it might be helpful in maintaining the network)
- some additional requirements on nodes  $\equiv$  **list coloring**

Zhang, X. (2016)

# What we consider?

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic
  - $d$ -degenerate coloring  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class
- the sizes of pieces are balanced  $\equiv$  equitable coloring (it might be helpful in maintaining the network)
- some additional requirements on nodes  $\equiv$  list coloring

# What we consider?

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic
  - **$d$ -degenerate coloring**  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class
    - proper coloring  $\equiv$  0-degenerate coloring
- the sizes of pieces are balanced  $\equiv$  **equitable coloring** (it might be helpful in maintaining the network)
- some additional requirements on nodes  $\equiv$  **list coloring**

# What we consider?

- decomposition into smaller pieces fulfilling some structural properties (might be helpful in identifying node failure)
  - proper coloring  $\equiv$  the pieces are independent
  - arboricity  $\equiv$  the pieces are acyclic
  - **$d$ -degenerate coloring**  $\equiv \delta(H) \leq d$  for every subgraph  $H$  of each color class
    - proper coloring  $\equiv$  0-degenerate coloring
    - acyclic coloring  $\equiv$  1-degenerate coloring
- the sizes of pieces are balanced  $\equiv$  **equitable coloring** (it might be helpful in maintaining the network)
- some additional requirements on nodes  $\equiv$  **list coloring**

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



- $G$  is  *$d$ -degenerate* if  $\delta(H) \leq d$  for any subgraph  $H$  of  $G$ ,
- $\mathcal{D}_d$  - the class of all  $d$ -degenerate graphs,

- $G$  is  *$d$ -degenerate* if  $\delta(H) \leq d$  for any subgraph  $H$  of  $G$ ,
- $\mathcal{D}_d$  - the class of all  $d$ -degenerate graphs,
- *$\mathcal{D}_d$ -coloring*:  $c : V(G) \rightarrow \mathbb{N}$  such that for each  $i \in \mathbb{N}$  the set of vertices coloured with  $i$  induces a  $d$ -degenerate graph

- $G$  is  *$d$ -degenerate* if  $\delta(H) \leq d$  for any subgraph  $H$  of  $G$ ,
- $\mathcal{D}_d$  - the class of all  $d$ -degenerate graphs,
- *$\mathcal{D}_d$ -coloring*:  $c : V(G) \rightarrow \mathbb{N}$  such that for each  $i \in \mathbb{N}$  the set of vertices coloured with  $i$  induces a  $d$ -degenerate graph
- *list assignment*  $L$ : a mapping that assigns a nonempty subset of  $\mathbb{N}$  to each vertex  $v \in V(G)$ ; given  $k \in \mathbb{N}$ , a list assignment  $L$  is  *$k$ -uniform* if  $|L(v)| = k$  for every  $v \in V(G)$ ,

## $(L, \mathcal{D}_d)$ -colourability

Given  $d \in \mathbb{N}_0$ , a graph  $G$  is  $(L, \mathcal{D}_d)$ -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.

### $(L, \mathcal{D}_d)$ -colourability

Given  $d \in \mathbb{N}_0$ , a graph  $G$  is  $(L, \mathcal{D}_d)$ -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 $(k, \mathcal{D}_d)$ -choosability

Given  $k \in \mathbb{N}$  and  $d \in \mathbb{N}_0$ , a graph  $G$  is *equitably  $(k, \mathcal{D}_d)$ -choosable* if for any  $k$ -uniform list assignment  $L$  there is an  $(L, \mathcal{D}_d)$ -colouring of  $G$  such that the size of any colour class  $\leq \lceil |V(G)|/k \rceil$ .

# The concept of $(k, d)$ -partition

## Definition

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

$$|N_G(x_i^j) \cap (S_1 \cup \dots \cup S_{j-1})| \leq di - 1, \quad i \in [k].$$

# The concept of $(k, d)$ -partition

## Definition

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

$$|N_G(x_i^j) \cap (S_1 \cup \dots \cup S_{j-1})| \leq di - 1, \quad i \in [k].$$

## Some observation

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

# The concept of $(k, d)$ -partition

## Definition

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

$$|N_G(x_i^j) \cap (S_1 \cup \dots \cup S_{j-1})| \leq di - 1, \quad i \in [k].$$

## 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  $t \geq k$ . If a graph  $G$  has a  $(k, d)$ -partition, then it is equitably  $(t, \mathcal{D}_{d-1})$ -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  $(t, \mathcal{D}_{d-1})$ -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  $(L, \mathcal{D}_{d-1})$ -colouring of  $G$ .

Proposition (generalization from  $(k, \mathcal{D}_0)$ - and  $(k, \mathcal{D}_1)$ -choosability)

Let  $k, d \in \mathbb{N}$  and let  $S$  be a set of distinct vertices  $x_1, \dots, x_k$  of a graph  $G$ . If  $G - S$  is equitably  $(k, \mathcal{D}_{d-1})$ -choosable and

$$|N_G(x_i) \setminus S| \leq di - 1$$

holds for every  $i \in [k]$ , then  $G$  is equitably  $(k, \mathcal{D}_{d-1})$ -choosable.

## Proposition (generalization from $(k, \mathcal{D}_0)$ - and $(k, \mathcal{D}_1)$ -choosability)

Let  $k, d \in \mathbb{N}$  and let  $S$  be a set of distinct vertices  $x_1, \dots, x_k$  of a graph  $G$ . If  $G - S$  is equitably  $(k, \mathcal{D}_{d-1})$ -choosable and

$$|N_G(x_i) \setminus S| \leq di - 1$$

holds for every  $i \in [k]$ , then  $G$  is equitably  $(k, \mathcal{D}_{d-1})$ -choosable.

## Some observations

- if  $G$  has a  $(k, d)$ -partition, then  $G$  is eq.  $(k, \mathcal{D}_{d-1})$ -choosable,

## Proposition (generalization from $(k, \mathcal{D}_0)$ - and $(k, \mathcal{D}_1)$ -choosability)

Let  $k, d \in \mathbb{N}$  and let  $S$  be a set of distinct vertices  $x_1, \dots, x_k$  of a graph  $G$ . If  $G - S$  is equitably  $(k, \mathcal{D}_{d-1})$ -choosable and

$$|N_G(x_i) \setminus S| \leq di - 1$$

holds for every  $i \in [k]$ , then  $G$  is equitably  $(k, \mathcal{D}_{d-1})$ -choosable.

## Some observations

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

## 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  $(t, \mathcal{D}_{d-1})$ -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  $(L, \mathcal{D}_{d-1})$ -colouring of  $G$ .

---

**Algorithm 1:** Equitable  $(L, \mathcal{D}_{d-1})$ -colouring( $G$ )

---

**Input** : Graph  $G$  on  $n$  vertices;  $L$  -  $t$ -uniform list assignment;  
a  $(k, d)$ -partition  $S_1 \cup \dots \cup S_{\eta+1}$  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  $(L, \mathcal{D}_{d-1})$ -colouring of  $G$

---

---

---

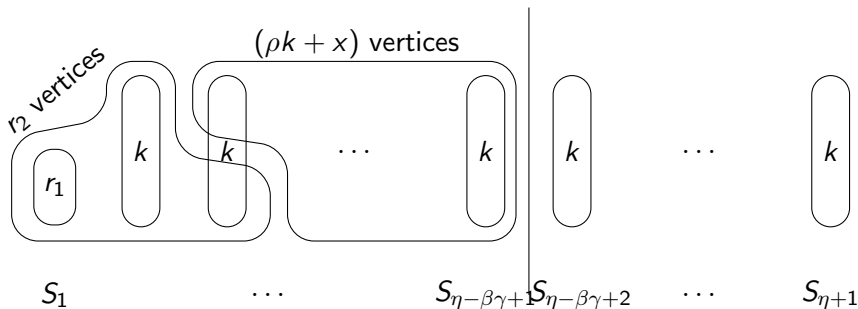
```

1 for  $j := 1$  to  $\eta + 1$  do
2   |   add REVERSE( $S_j$ ) to  $S$ ;
3 end
4 ...
5  $\gamma := t \div k$ ;  $r := t \pmod k$ ;  $\rho := \beta r \div k$ ;  $x := \beta r \pmod k$ ;
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  $\rho$  do
12   |   take and delete  $k$  elements from the beginning of  $S$ , and add to  $S'$ ;
13   |   COLOUR_LIST( $S', k$ );
14   |    $S_{col} := S_{col} + S'$ ;
15 end
16 REORDER( $S_{col}$ );
17  $\bar{S} := S$ ; MODIFY_COLOURLISTS( $S_{col}, \bar{S}$ );
18 COLOUR_LIST( $S, \gamma k$ );

```

---





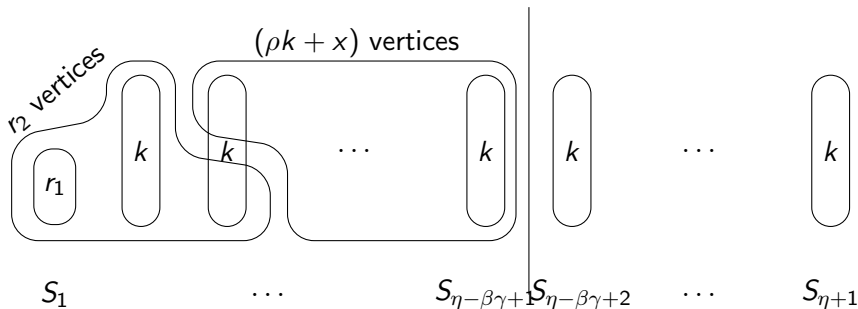
$$|V(G)| = \eta k + r_1$$

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

$$|V(G)| = \beta \gamma k + (\rho k + x) + r_2$$

- 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 each  $S_j$  (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 each  $S_j$  (Prop. 1)
- we colour  $r_2$  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  $N_G^{col}(d, v)$   
 $(\{w \in N_G(v) : w \text{ has } d \text{ neighbors coloured with } c(v)\})$   
(in consequence we have  $r_2$  rainbow colored vertices)



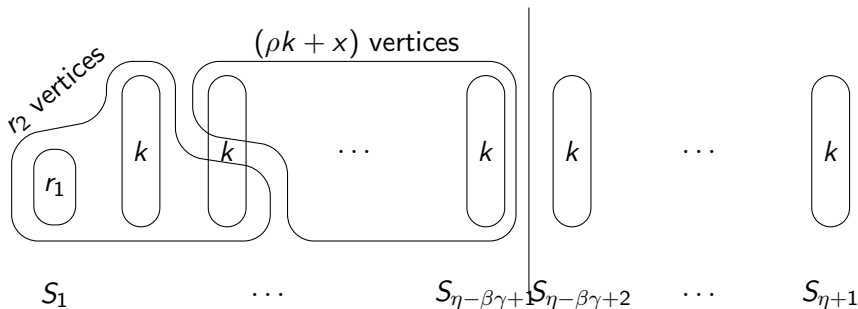
$$|V(G)| = \eta k + r_1$$

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

$$r := t \bmod k$$

$$|V(G)| = \beta \gamma k + (\rho k + x) + r_2$$

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



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

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

$$|V(G)| = \beta \gamma k + (\rho k + x) + r_2$$

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

- we extend the list colouring into the uncoloured  $\beta\gamma k$  vertices by colouring  $\beta$  groups of  $\gamma k$  vertices; first, we associate each group of  $\gamma k$  vertices with a set of  $r$  vertices coloured in the previous step (for different groups these sets are disjoint); next, we color the vertices of each of the group using  $\gamma k$  different colors that are also different from the colors of  $r$  vertices of the set associated with this group;
- our final equitable list colouring is the consequence of a partition of  $V(G)$  into  $\beta + 1$  coloured sets, each one of size at most  $t$  and each one formed by vertices coloured differently.

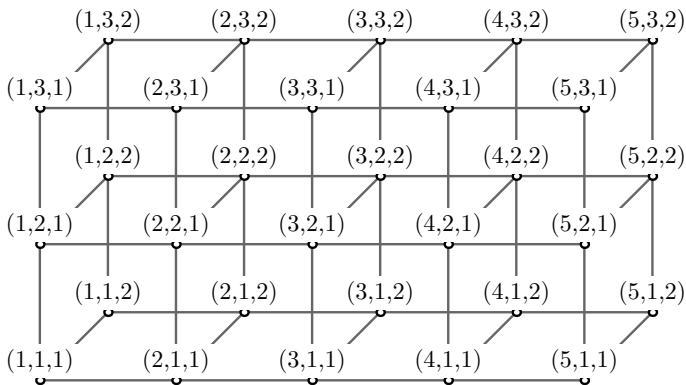


### Theorem 3

For a given graph  $G$  on  $n$  vertices, a  $t$ -uniform list assignment  $L$ , a  $(k, d)$ -partition of  $G$  the  $\text{EQUITABLE } (L, \mathcal{D}_{d-1})\text{-COLOURING}(G)$  algorithm returns  $(L, \mathcal{D}_{d-1})$ -colouring of  $G$  in  $O(n\Delta^2(G))$  time.

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

# Application to 3-dimensional grids



Grid  $P_5 \square P_3 \square P_2$ .

## Theorem 4

For a given 3-dimensional grid  $G$  the  $\text{PARTITION3D}(G)$  algorithm returns a  $(3, 2)$ -partition of  $G$  in polynomial-time.

## Theorem 4

For a given 3-dimensional grid  $G$  the  $\text{PARTITION3D}(G)$  algorithm returns a  $(3, 2)$ -partition of  $G$  in polynomial-time.

---

### Algorithm 3: Partition3d( $G$ )

---

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

**Output**: A  $(3, 2)$ -partition  $S_1 \cup \dots \cup S_{\alpha+1}$  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  $G = P_{n_1} \square P_{n_2} \square P_{n_3}$ .

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

---

### $(3, 2)$ -partition of $G$ :

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

$$|N_G(x_i^j) \cap (S_1 \cup \dots \cup S_{j-1})| \leq 2i - 1, \quad i \in [3].$$

## Theorem 5

Let  $t \geq 3$  be an integer. Every 3-dimensional grid is equitably  $(t, \mathcal{D}_1)$ -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  $(L, \mathcal{D}_1)$ -colouring of  $G$ .

# Some open questions

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



# Some open questions

- What is the complexity of recognition of graphs having a  $(k, d)$ -partition?
- What is the parameterized complexity of the problem of  $\text{EQUITABLE } (L, \mathcal{D}_d)\text{-CHOOSABILITY}$  in respect to different parameters?



Thank you for your attention  
[hanna.furmanczyk@ug.edu.pl](mailto:hanna.furmanczyk@ug.edu.pl)