

# Riešenie problému maximálnej kliky pomocou neurónových sietí

Bc. Mário Lipovský  
školiťel': Mgr. Vladimír Boža, PhD.

# Úvod

Motivácia, súvisiace výsledky, grafové neurónové siete

# Riešenie NP-opt. úloh pomocou strojového učenia

Motivácia: NP-ťažké úlohy → **široká škála heuristík**

- pracovito navrhnuté
- pre rôzne inštancie vhodné rôzne heuristiky

Prístup strojového učenia: naučíme model

- **detekovať** podštruktúry v množinách, grafoch
  - **rozhodovať**, ktoré štruktúry sú blízko optimálnych
- pomocou
- príkladov (vstup, výstup)
  - odmeny za kvalitné riešenia

**Strojové učenie nemusí nahradiť klasické algoritmy, no môže ich vylepšiť**

- automatický návrh heuristickej funkcie
- heuristika optimalizovaná pre špecifický typ vstupov

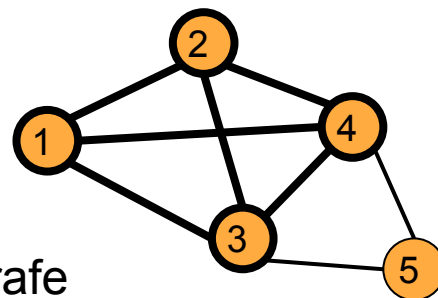
# Doterajšie výsledky

- Khalil et al.: Learning to branch in mixed integer programming, 2016; Learning to run heuristics in tree search, 2017
- Bello et al.: Combinatorial optimization with reinforcement learning, 2016
- Li et al.: Combinatorial optimization with graph convolutional networks and guided tree search, 2018
- A mnoho ďalších: Bengio et al.: Machine learning for combinatorial optimization: a methodological tour d'horizon, 2018

# Maximálna klika

Jedna z najviac študovaných NP-úplných úloh

Hľadáme najväčší kompletný podgraf v jednoduchom grafe



Aktuálne najúspešnejší exaktný prístup: **branch-and-bound (B&B)**

- rekurzívne prehľadávanie stromu s orezávaním
  - heuristika branch - určuje poradie navštevovania vrcholov
  - heuristika bound - určuje, ktoré časti stromu prehľadávania orezať

Náš prístup: **Natrénovať neurónovú sieť pre heuristickú funkciu branch**

Problém: Ako reprezentovať graf ako vstup pre neurónovú sieť?

# Grafové neurónové siete (GNN)

Cieľ: naučiť sa vnorenie  $v \in V \mapsto h_v \in \mathbb{R}^k$

- špecializované pre koncovú úlohu
- použiteľné pre nové grafy

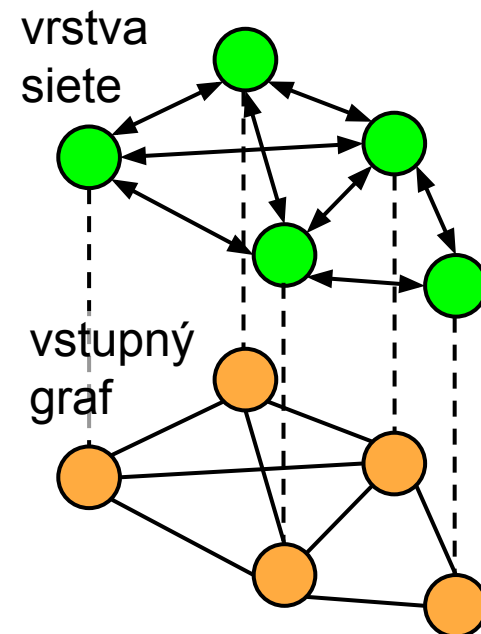
## Topológia siete

- 1 neurón v každom vrchole (zdieľané váhy)
- pospájané navzájom rovnako ako v grafe

## Výpočet

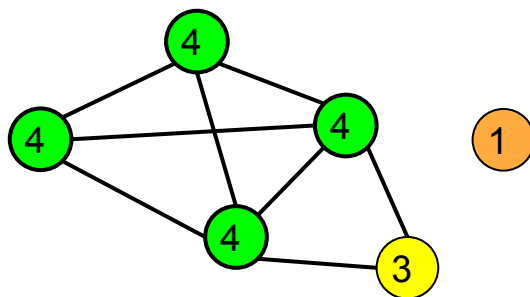
- inicializácia:  $\forall v \in V_G : h_v^{(0)} = \vec{1}$
- niekoľko iterácií (vrstiev) posielania správ medzi vrcholmi

$$h_v^{(i)} := \text{update} \left( h_v^{(i-1)}, \text{aggregate}_{u \in N(v)} \left( \text{message} \left( h_v^{(i-1)}, h_u^{(i-1)} \right) \right) \right)$$

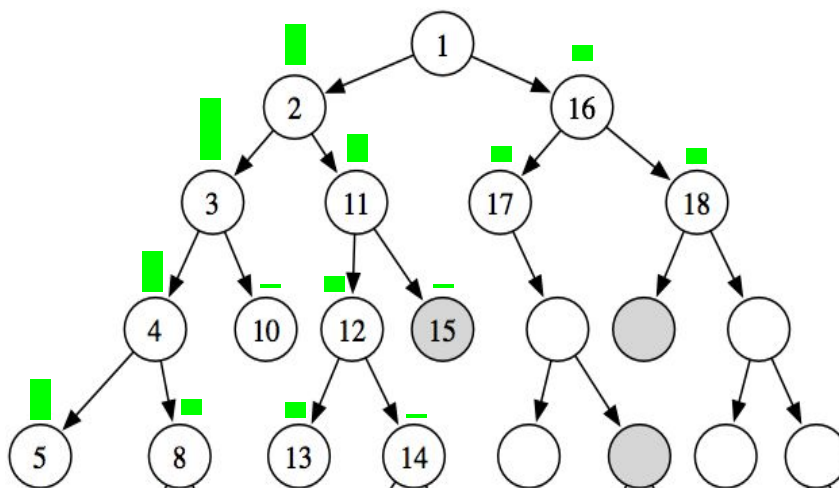


# Naše ciele

Ako dobre dokážu **GNN detekovať kliky**, rozoznávať väčšie od menších?



Vieme **vylepšiť B&B algoritmus** pre hľadanie max. kliky pomocou GNN?



# Náš prístup

- 1) Vygenerujeme množstvo malých grafov, nájdeme v nich kliky
- 2) Natrénujeme na nich grafovú sieť (GNN) učením s učiteľom
- 3) Použijeme predpovede siete vnútri algoritmu B&B pre výber vetvy (branch h.)



# Náš prístup a otvorené otázky

- 1) Vygenerujeme množstvo malých grafov, nájdeme v nich kliky
  - reprezentácia veľkostí klík v tréningových vstupoch
  - typy grafov
  
- 2) Natrénujeme na nich grafovú sieť (GNN) učením s učiteľom
  - architektúra GNN
  - úloha pre NN: chybové funkcie a metriky
  - baseline modely
  
- 3) Použijeme predpovede siete vnútri algoritmu B&B pre výber vetvy (branch h.)
  - metriky
  - baseline heuristiky

# Trénovanie a testovanie GNN

Vstupné grafy, architektúra, výsledky, analýza

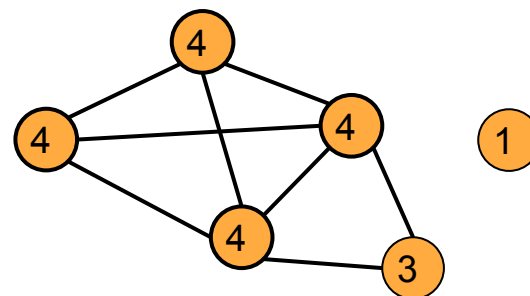
# Trénovacie a testovacie dáta

**Triedy grafov** vybrané z benchmarkov pre kliky (DIMACS, BHOSLIB)

- C.5, C.9 - náhodné grafy
- rb - ťažké inštancie CSP
- dsjc - k-partitné grafy podobné C.5
- brock - náhodné grafy so skrytými klikami
- 100 x 20-vrcholový graf z každej triedy pre tréovanie, 100 pre testovanie

**Reprezentácia veľkostí klík**

$\omega(v)$  = veľkosť najväčšej kliky  
grafu indukovaného  $\{v\} \cup N(v)$



- pomocou B&B algoritmu nájdeme najväčšie kliky v okolí každého vrchola

# Architektúra siete

**Grafová sieť** = 2-5 grafových konvolučných vrstiev

- Structure2Vec<sup>1</sup> - zdieľanie parametrov medzi vrstvami
- ChebNet<sup>2</sup> - každá vrstva má vlastné parametre + normalizácia

$$g_{\theta_1} : G \mapsto (h_{v_1}, \dots, h_{v_{|V|}})$$

Predikcia veľkosti kliky = 2-4-vrstvový **perceptron** (spoločný pre všetky vrcholy)

$$h_{\theta_2} : \mathbb{R}^k \rightarrow \mathbb{R}$$

Výsledná predpoveď = zloženie grafovej siete a perceptronu:

$$f_{\theta}(v_i) = h_{\theta_2}((g_{\theta_1}(G))_i)$$

<sup>1</sup> Dai et al.: Discriminative embeddings of latent variable models for structured data, 2016

<sup>2</sup> Defferrard et al.: Convolutional neural networks on graphs with fast localized spectral filtering, 2016

# Úloha 1 - predikcia $\omega(v)$

$\omega(v)$  = veľkosť najväčšej kliky grafu indukovaného  $\{v\} \cup N(v)$

**Úloha:**  $\forall v \in V_G : f_\theta(v) \approx \omega(v)$  alebo  $\omega(v)/(\deg(v) + 1)$

**Trénonvanie:** minimalizácie kvadratickej chyby na trénovacej množine grafov  $\mathcal{G}_T$

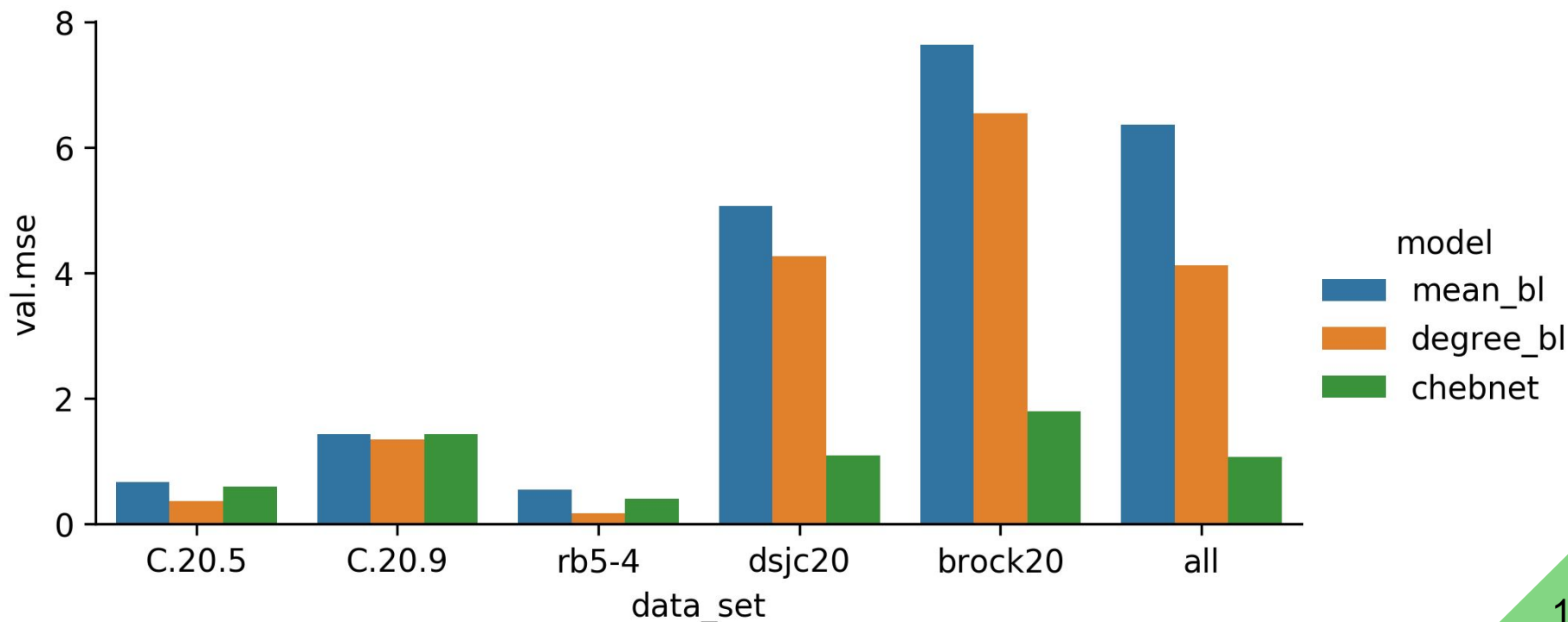
$$l_{MSE}(\theta, v) = (f_\theta(v) - \omega(v))^2$$

**Metrika:** priemerná kvadratická chyba na vrchol

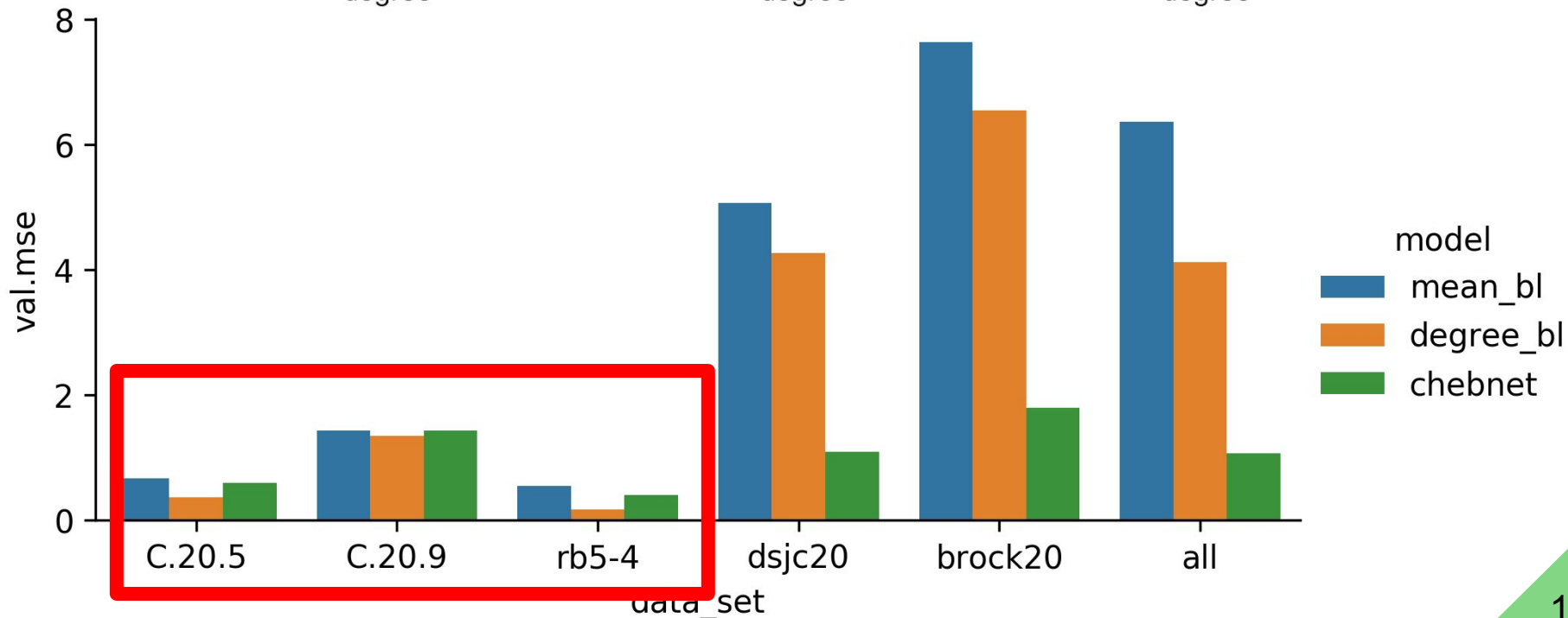
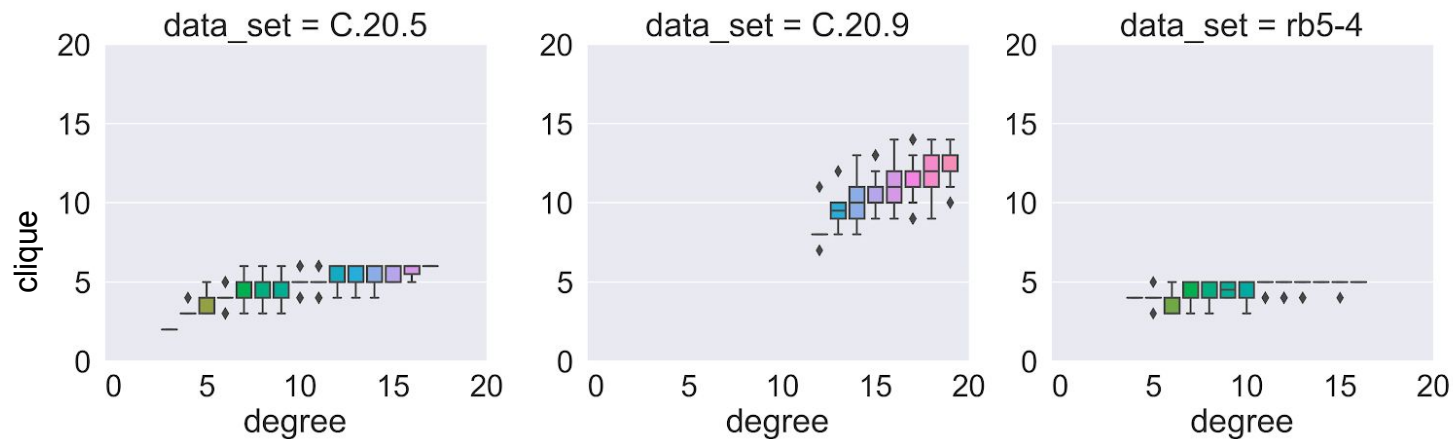
**Baseline:** priemer  $\omega(v)$ , priemer  $\omega(v)$  podľa stupňa vrchola

# Úloha 1 - predikcia $\omega(v)$ - rôzne typy grafov

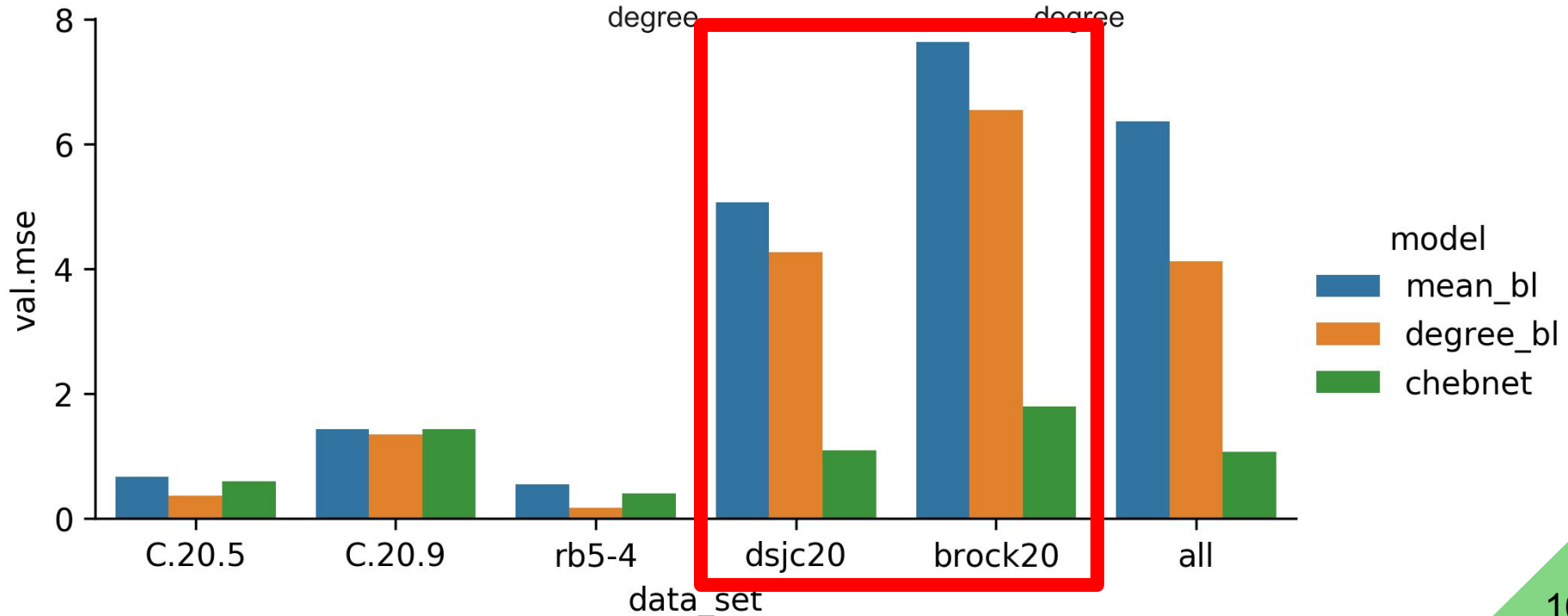
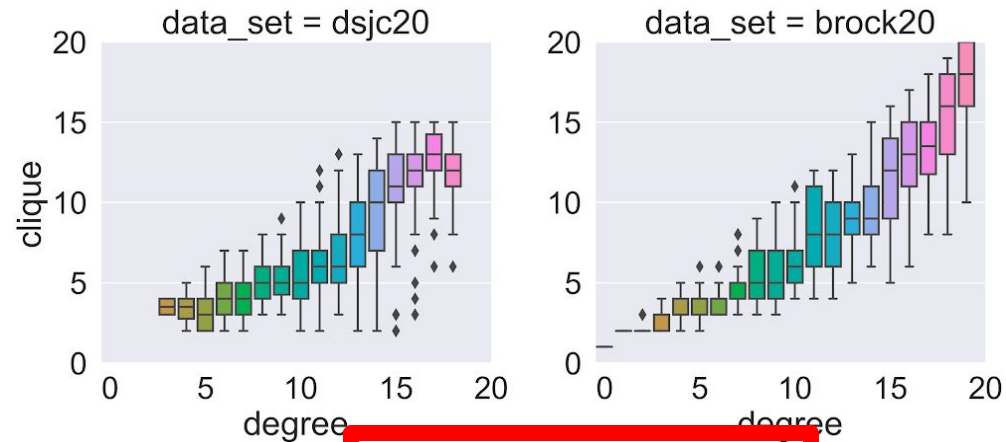
Najlepšie výsledky: ChebNet + predpovede relatívne k stupňu, 5 konv. vrstiev



# Malý rozptyl veľkostí klík → model sa naučí priemer



Veľký rozptyl veľkostí klík → model prekoná baseline, naučí sa zložitejšie štruktúry ako len stupeň





## Úloha 2 - usporiadanie vrcholov podľa $\omega(v)$

**Úloha:**

$$\omega(v_1) \geq \omega(v_2) \geq \dots \geq \omega(v_{|V|})$$
$$f_\theta(v_1) \geq f_\theta(v_2) \geq \dots \geq f_\theta(v_{|V|})$$

**Trénonvanie:** MSE, alebo margin loss - penalizovanie inverzií

$$l_{RANK}(\theta, v_i, v_j) = [\omega(v_i) > \omega(v_j)] \cdot \max(0, f_\theta(v_j) - f_\theta(v_i) + \text{margin}(v_i, v_j))$$

**Metrika:** FMC - fraction of max. clique

relevantné vrcholy = sú v klikách max. veľkosti, ich počet  $R(G)$

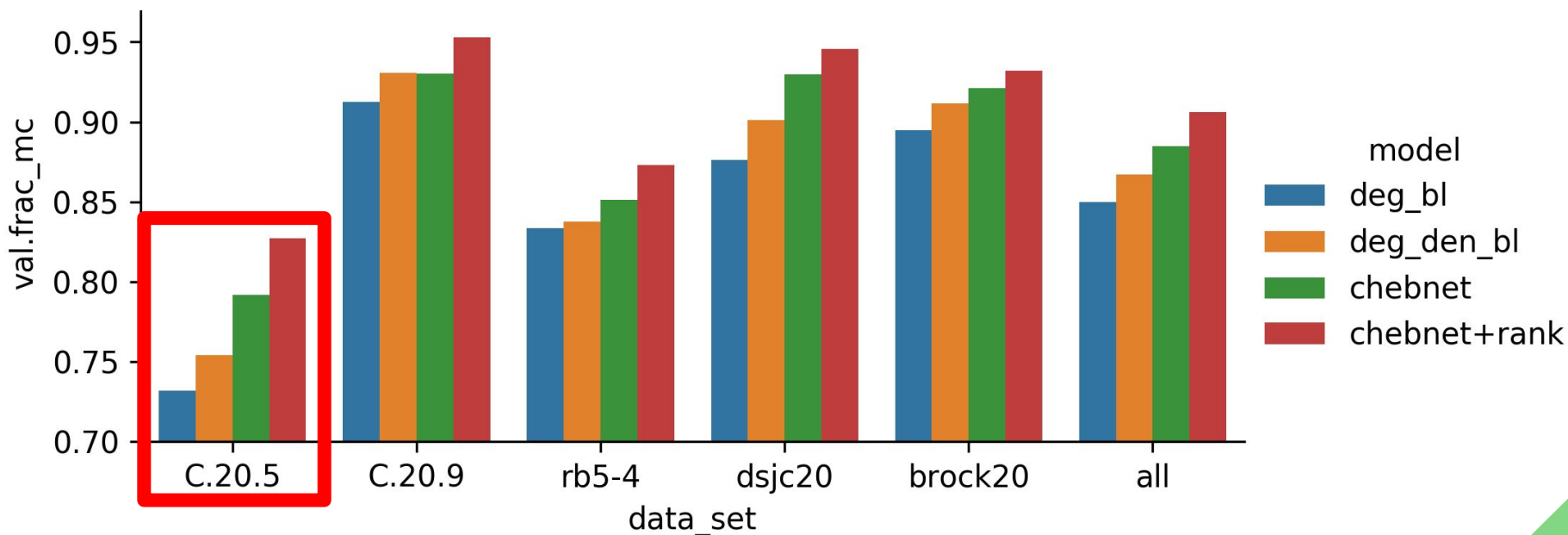
zlomok relevantných vrcholov v  $R(G)$  najvyššie hodnotených vrcholoch

**Baseline:** usporiadaj podľa stupňa, usporiadaj podľa stupňa a súčtu stupňov sus.

## Úloha 2 - usporiadanie vrcholov podľa $\omega(v)$

metrika = aká časť z prvých  $R(G)$  najvyššie hodnotených vrcholov je z max. Kličky?

- rankovací model je konzistentne lepší ako modely trénované s MSE
- najväčšie zlepšenie - malý rozptyl veľkostí klíky, malá korelácia so stupňami
- rozdiely výsledkov sú pomerne malé



# Testovanie generalizácie

## Generalizácia na väčšie grafy (50 vrcholov)

- regresná úloha
  - veľké chyby, pri najlepšom podobné mean baseline
- úloha usporiadavania vrcholov
  - GNN si zachovávajú malú konzistentnú výhodu oproti baseline modelom

## Generalizácia na iné typy grafov (tréning a testovanie na iných typoch)

- modely tréňované na vstupoch ťažkých pre baseline: schopné generalizovať

# Použitie v branch and bound

B&B algoritmus, tradičné heuristiky, porovnanie s GNN

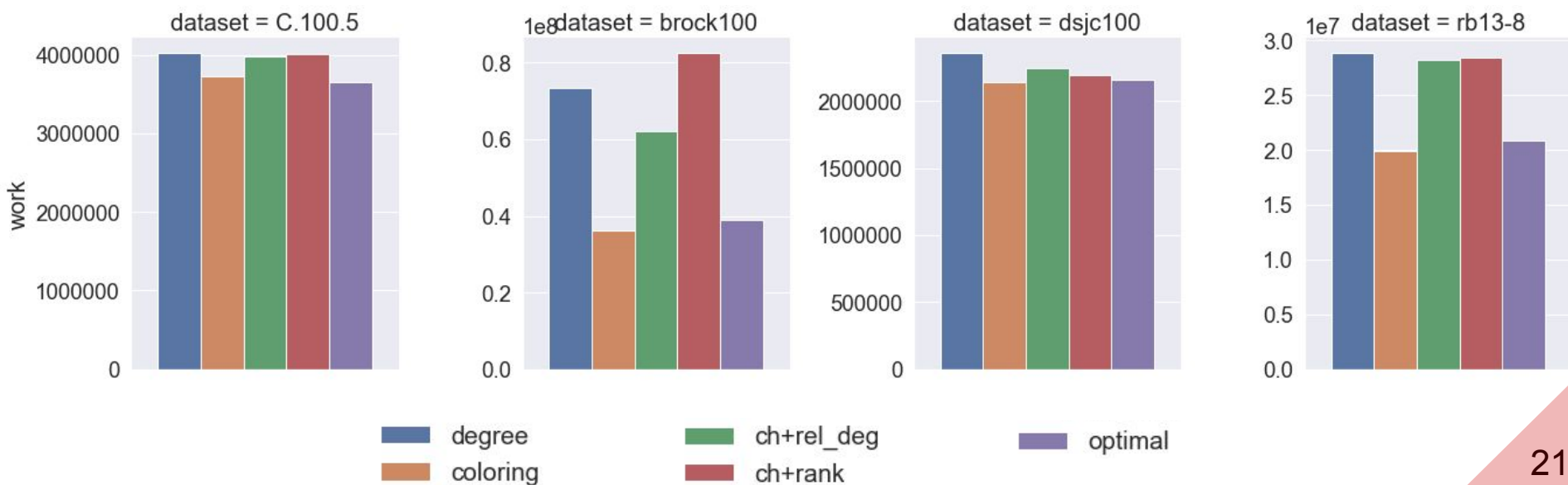
# Použitie GNN v branch heuristike

**Metriky:** počet rekurzívnych volaní B&B, množstvo práce ~ časová zložitost' volaní

$$\text{work}(h, G) = \sum_{P \in \mathcal{P}_h} |E_{G_P}|$$

**Baseline:** najskôr do vrcholov s najvyšším stupňom, do vrchola s max. farbou

**Výsledky:** GNN mierne lepšie než degree baseline, neprekoná farbiace heuristiky



# Záver

zhrnutie, budúcnosť GNN & ML for CO

# Zhrnutie práce

## Štúdium literatúry

- Hlboké učenie, ML for CO, GNN, klikové algoritmy

## Implementácia

- Generátory grafov, B&B, modely a trénovacia infraštruktúra, analýza
- <https://github.com/maario/mcp-gnns> (kód + notebooky pre reprodukciu)

## Návrh a trénovanie sietí, experimenty

- Typy grafov, architektúra, transformácie vstupov, chybové funkcie, metriky
- Generalizácia, výkon sietí v B&B

## Výsledky

- GNN dokážu využiť zložitejšie štruktúry v grafoch pre hľadanie klík
- GNN-branch zatiaľ nedosahuje lepšiu kvalitu než používané h.f.

# Lessons learned

## **Systematický výber dát**

- grafy s potenciálom pre zlepšenie (práca B&B >> práca OPT)
- triedy s rôznymi charakteristikami - rozptily veľkosti klík, korelácie so stupňami
- väčšie grafy -> zaujímavejšie štruktúry

## **Trénovanie a vyhodnocovanie modelu pre koncovú úlohu**

- dáta zbierané počas behu algoritmu
- metriky podľa úloh (napr. čas výpočtu, počet volaní)

## **Lepší potenciál využita GNN**

- v inom algoritme - využiť prehľadávanie viac do šírky
- na heuristické rýchle hľadanie veľkých (nie zaručene max.) klík

Silnejšie baseline modely, analýza outlierov

→ vizualizácia / objektívne hodnotenie schopností GNN



# Smerovanie pre budúci výskum

## **Teória: Čo dokážu GNN?**

- Teoretické hranice, ako dobre dokážu GNN detegovať kliky
- Návrh architektúr, ktoré dokážu viac - napr. spracúvať regulárne grafy

## **Interpretácia: Čo sa siete v skutočnosti naučia?**

- Zobrazovanie štruktúr, na ktoré sú siete citlivé
- Generovanie outlierov - štruktúr na ktorých robia chyby

## **Ako vyberať tréningové a testovacie dáta pre ML for CO?**

- Lacné (automatizovateľné) riešenie malých inštancií
- Automatizované aktívne učenie / adversarial training

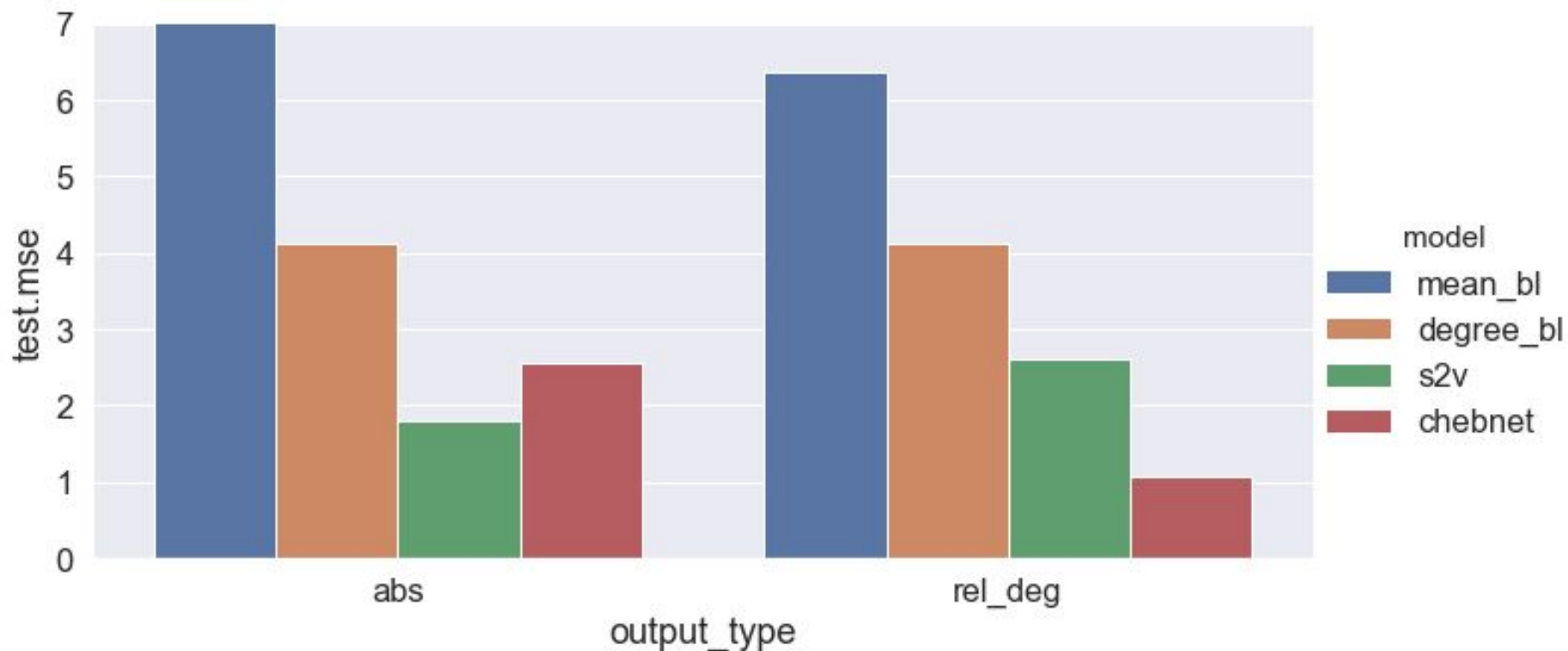
# Ďakujem za pozornosť

Hľadanie maximálnej kliky  
pomocou neurónových sietí

Bc. Mário Lipovský  
školiťel': Mgr. Vladimír Boža, PhD.

# Úloha 1 - predikcia $\omega(v)$ - porovnanie modelov

Najlepšie výsledky: ChebNet + predpovede relatívne k stupňu, 5 konv. vrstiev



# Algoritmus branch and bound

**C** - aktuálne čiastočné riešenie (klika)

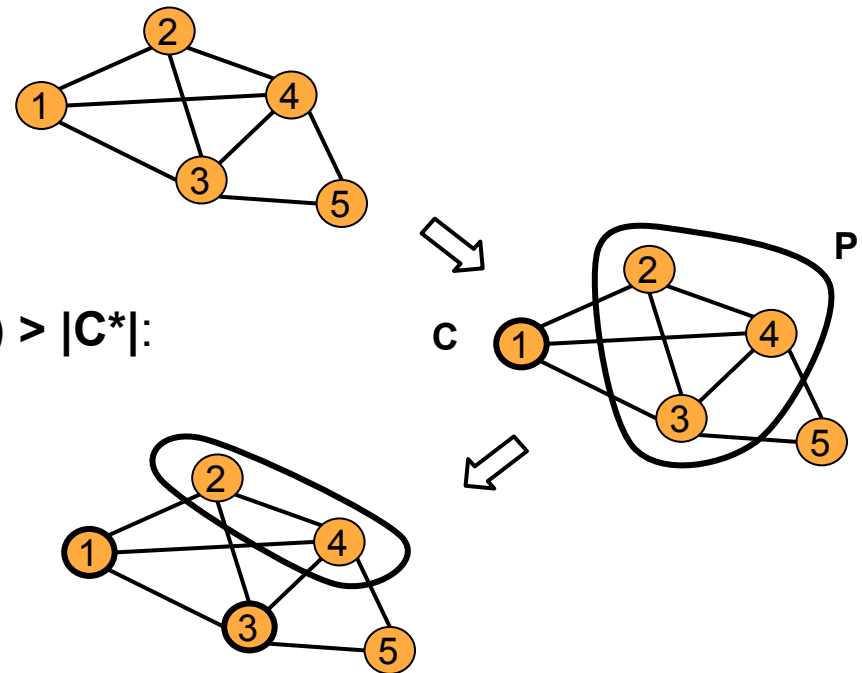
**P** - množina kandidátskych vrcholov

**C\*** - najväčšia nájdená klika

Začíname s  $C = \{\}$ ,  $C^* = \{\}$ ,  $P = V$

$\text{max\_clique}(C, P)$ :

- **zoraď kandidátov v podľa  $h_1(v)$**
- pre každého kandidáta **v**, kde  $|C| + h_2(v) > |C^*|$ :
  - presuň **v** z **P** do **C**
  - ak  $|C| > |C^*|$ , tak  $C^* := C$
  - vytvor novú množinu kandidátov **P'**
  - $\text{max\_clique}(C, P')$
  - odstráň **v** z **P**



$h_1(v)$  - branch heuristika

$h_2(v)$  - bound - horný odhad veľkosti max. kliky v  $P \cap \{\{v\} \cup N(v)\}$