Binární paint shop problém

Jiří Kalvoda
Vedoucí: doc. Mgr. Robert Šámal, Ph.D.
Oponent: Mgr. Michal Opler, Ph.D.

IÚUK MFF

V Hnojníku dne 2024-09-04

Definice problému

Problém (Binární paint shop problém)

- $2n$ *aut v řadě*
- n *typů, od každého dvě auta*

Definice problému

Problém (Binární paint shop problém)

- $2n$ *aut v řadě*
- n *typů, od každého dvě auta*
- *Barvení aut v pořadí dle vstupu*
- *Chceme: každý typ* $\rightarrow$ *jedno červeně a jedno zeleně*
- *Co nejméně změn barev v posloupnosti*

Definice problému

Problém (Binární paint shop problém)

- *$2n$ aut v řadě*
- *n typů, od každého dvě auta*
- *Barvení aut v pořadí dle vstupu*
- *Chceme: každý typ $\rightarrow$ jedno červeně a jedno zeleně*
- *Co nejméně změn barev v posloupnosti*

Vstup:	Typy aut v řadě
Výstup:	Přiřazení barev autům
Hodnota výstupu:	Počet změn barev

Doposud známé výsledky

Problém (Binární paint shop problém)

- $2n$ *aut v řadě*
- n *typů, od každého dvě auta*
- *Barvení aut v pořadí dle vstupu*
- *Chceme: každý typ* $\rightarrow$ *jedno červeně a jedno zeleně*
- *Co nejméně změn barev v posloupnosti*

- Rozhodovací problém (Existuje řešení s nejvýše k změnami?) je NP-těžký
- Optimalizační problém je APX-těžký (Bonsma et al., 2006)

$\Rightarrow$ ($\mathrm{P} \neq \mathrm{NP} \rightarrow$ Neexistuje aproximační schéma)

- Unique games conjecture $\Rightarrow$ neexistuje c-aproximace (Gupta et al., 2013)

Dává smysl zkoumat chování v průměrném případě.

Notace

$$\gamma(\alpha) = \textit{Hodnota (počet změn barev) optimálního řešení vstupu } \alpha$$

$$\gamma(n) = \frac{\sum_{\alpha:\ \textit{vstup velikosti } n} \gamma(\alpha)}{(2n)!/2^n} = \mathbb{E}_\alpha[\gamma(\alpha)]$$

$\gamma_{\text{alg}}(n)$, $\gamma_{\text{alg}}(\alpha)$ … *totéž pro hodnotu výstupu algoritmu*

Dává smysl zkoumat chování v průměrném případě.

Notace

$\gamma(\alpha)$ = *Hodnota (počet změn barev) optimálního řešení vstupu* α

$$\gamma(n) = \frac{\sum_{\alpha:\text{ vstup velikosti } n} \gamma(\alpha)}{(2n)!/2^n} = \mathbb{E}_\alpha[\gamma(\alpha)]$$

$\gamma_{\text{alg}}(n)$, $\gamma_{\text{alg}}(\alpha)$ … *totéž pro hodnotu výstupu algoritmu*

- Hladový algoritmus (Andres et al., 2011):

$$\gamma(n) \leq \gamma_{\text{alg}}(n) = \sum_{0 \leq k < n} \frac{2k^2 - 1}{4k^2 - 1} \sim \frac{1}{2} \cdot n$$

- Hladový algoritmus (Andres et al., 2011):

$$\gamma(n) \leq \gamma_{\text{alg}}(n) = \sum_{0 \leq k < n} \frac{2k^2 - 1}{4k^2 - 1} \sim \frac{1}{2} \cdot n$$

- Rekurzivní hladový algoritmus (Andres et al., 2011):

$$\frac{2}{5}n - \frac{8}{15} \leq \gamma_{\text{alg}}(n) \leq \frac{2}{5}n + \frac{7}{10} \;\Rightarrow\; \gamma(n) \leq \gamma_{\text{alg}}(n) \sim 0.4n$$

- Hladový algoritmus (Andres et al., 2011):

$$\gamma(n) \leq \gamma_{\text{alg}}(n) = \sum_{0 \leq k < n} \frac{2k^2 - 1}{4k^2 - 1} \sim \frac{1}{2} \cdot n$$

- Rekurzivní hladový algoritmus (Andres et al., 2011):

$$\frac{2}{5}n - \frac{8}{15} \leq \gamma_{\text{alg}}(n) \leq \frac{2}{5}n + \frac{7}{10} \;\Rightarrow\; \gamma(n) \leq \gamma_{\text{alg}}(n) \sim 0.4n$$

- Hvězdičkový rekurzivní algoritmus (Hančl et al., 2024):

Hypotéza: $\gamma_{\text{alg}}(n) \sim \left(\frac{1}{13} \cdot \sqrt{61} - \frac{3}{13}\right) n \doteq 0.370n$

- Hladový algoritmus (Andres et al., 2011):

$$\gamma(n) \le \gamma_{\mathrm{alg}}(n) = \sum_{0 \le k < n} \frac{2k^2 - 1}{4k^2 - 1} \sim \frac{1}{2} \cdot n$$

- Rekurzivní hladový algoritmus (Andres et al., 2011):

$$\frac{2}{5}n - \frac{8}{15} \le \gamma_{\mathrm{alg}}(n) \le \frac{2}{5}n + \frac{7}{10} \;\Rightarrow\; \gamma(n) \le \gamma_{\mathrm{alg}}(n) \sim 0.4n$$

- Hvězdičkový rekurzivní algoritmus (Hančl et al., 2024):

Hypotéza: $\gamma_{\mathrm{alg}}(n) \sim \left(\frac{1}{13} \cdot \sqrt{61} - \frac{3}{13}\right) n \doteq 0.370n$

- Dolní odhad pomocí pravděpodobnosti (Hančl et al., 2024):

$$\gamma_{\mathrm{alg}}(n) \ge 0.214n$$

Nové řešení založené na semidefinitním programování

Semidefinitní programování

- *Optimalizační technika podobna lineárnímu programování*
- *Jedna z forem: Rozmísťování n bodů na $n-1$ dimenzionální sféru, podmínky a optimalizační funkce pracující se skalárními součiny bodů*

- Založeno na Goemans-Williamsově algoritmu na maximální řez (Gärtner et al., 2012)
- Pro každé auto bod, podmínky na protilehlost dvojic stejného typu, minimalizujeme vzdálenost sousedních aut
- Zaokrouhlení pomocí řezu náhodnou rovinou

Nové řešení založené na semidefinitním programování

- Pro každé auto bod, podmínky na protilehlost dvojic stejného typu, minimalizujeme vzdálenost sousedních aut
- Zaokrouhlení pomocí řezu náhodnou rovinou

Hypotéza

Pro dostatečně velká n platí:

$$\gamma_{\mathrm{alg}}(n) \leq 0.34n$$

Nové řešení založené na semidefinitním programování

- Pro každé auto bod, podmínky na protilehlost dvojic stejného typu, minimalizujeme vzdálenost sousedních aut
- Zaokrouhlení pomocí řezu náhodnou rovinou

Hypotéza

Pro dostatečně velká n platí:

$$\gamma_{\mathrm{alg}}(n) \leq 0.34n$$

Dosavadní výsledky:

- Dolní odhad: $\gamma_{\mathrm{alg}}(n) \geq 0.214n$
- Rekurzivní hladový algoritmus: $\gamma_{\mathrm{alg}}(n) \sim 0.4n$
- Hvězdičkový rekurzivní algoritmus: Hypotéza: $\gamma_{\mathrm{alg}}(n) \sim 0.370n$

Nové řešení založené na semidefinitním programování

- Pro každé auto bod, podmínky na protilehlost dvojic stejného typu, minimalizujeme vzdálenost sousedních aut
- Zaokrouhlení pomocí řezu náhodnou rovinou

Hypotéza

Pro dostatečně velká n platí:

$$\gamma_{\text{alg}}(n) \leq 0.34n$$

Věta

Pro každý vstup α platí:

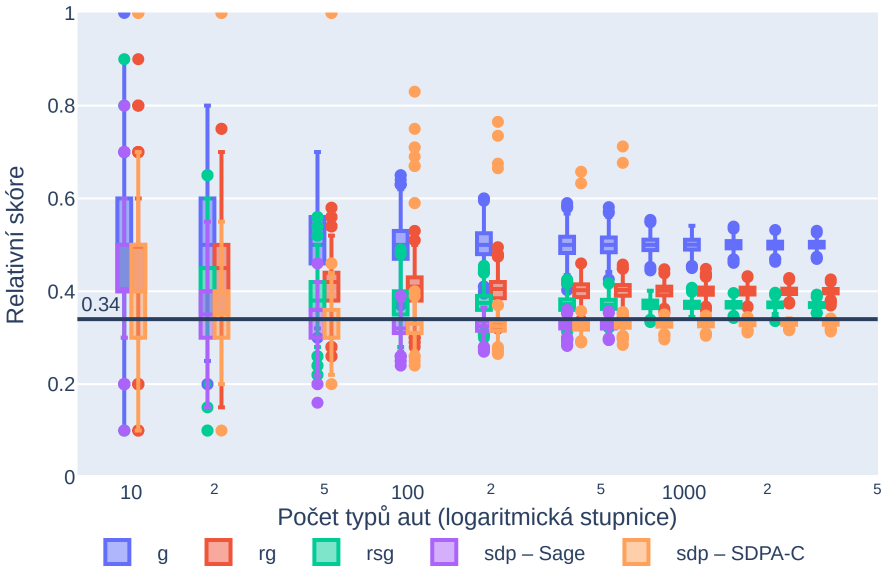
$$\mathbb{E}_{\textit{Náhodné bity}}[\gamma_{\text{alg}}(\alpha)] \ \leq \ \gamma(\alpha) + 0.212n$$

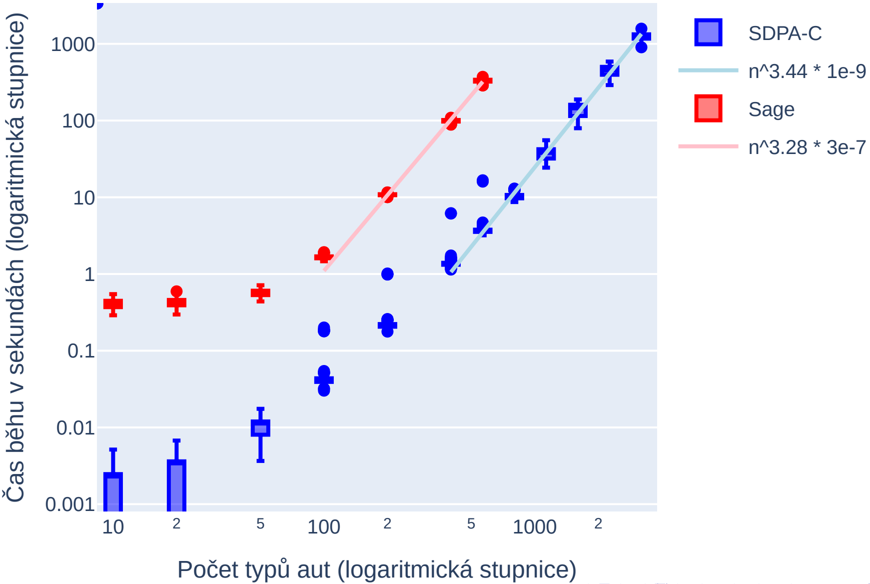
Implementace a testování

- Implementace původních i nového algoritmu
- Různé implementace semidefinitního programování

Implementace a testování

- Implementace původních i nového algoritmu
- Různé implementace semidefinitního programování
- Různé velikosti vstupů (n od 10 do $3\,200$)
- Pokaždé $1\,000$ nezávisle náhodně vybraných vstupů
- Několik dní na 4 strojích (moderní workstations)


Relativní skóre
1
0.8
0.6
0.4
0.34
0.2
0
10
2
5
100
2
5
1000
2
5
Počet typů aut (logaritmická stupnice)
g
rg
rsg
sdp – Sage
sdp – SDPA-C

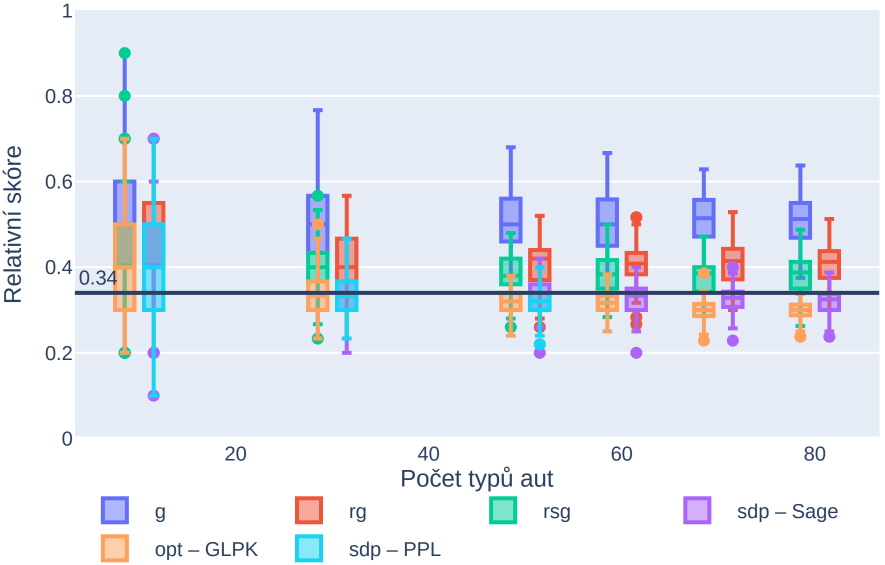

SDPA-C
n^3.44 * 1e-9
Sage
n^3.28 * 3e-7
Čas běhu v sekundách (logaritmická stupnice)
1000
100
10
1
0.1
0.01
0.001
10
2
5
100
2
5
1000
2
Počet typů aut (logaritmická stupnice)

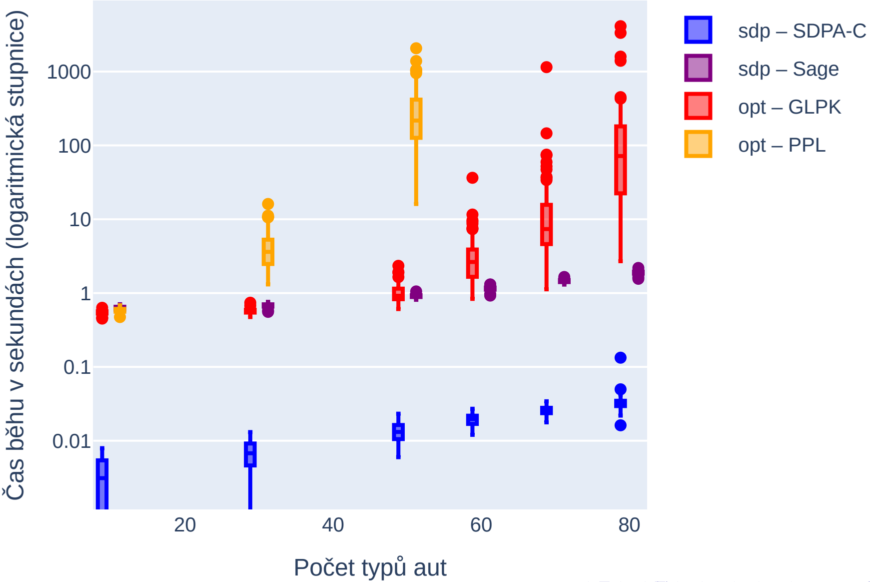
Poděkování

Děkuji vedoucímu práce za oporu při zkoumání a psaní, oponentovi za napsání posudku a všem posluchačům za pozornost.

Otázka od oponenta

Zkoušel jste (na menších vstupech) porovnat výsledek Vašeho algoritmu s optimálním obarvením? Pokud ano, jak daleko se typicky pohyboval?


1
0.8
0.6
0.4
0.34
0.2
0
Relativní skóre
20
40
60
80
Počet typů aut
g
rg
rsg
sdp – Sage
opt – GLPK
sdp – PPL


Čas běhu v sekundách (logaritmická stupnice)
1000
100
10
1
0.1
0.01
20
40
60
80
Počet typů aut
sdp – SDPA-C
sdp – Sage
opt – GLPK
opt – PPL

Otázky?

Prostor pro další vaše otázky.