

Verbesserungen

Levin Nemesch, Joshua Sangmeister

03. Februar 2021

Algorithm Engineering - Projekt

Ansätze:

- Alternative ILP-Formulierung
- Alternativer exakter Algorithmus: Reduzierung auf MaxClique
- ILP mit neuer Separationsheuristik
- ILP-Formulierung mit stärkeren Constraints

TODO

→ deutlich langsamer

- Beobachtung: Optimale Lösung des Labeling-Problems bildet Clique maximaler Größe im dualen Konfliktgraphen, da alle Labels sich gegenseitig nicht überlappen dürfen

- Beobachtung: Optimale Lösung des Labeling-Problems bildet Clique maximaler Größe im dualen Konfliktgraphen, da alle Labels sich gegenseitig nicht überlappen dürfen
- Aufbauen des dualen Konfliktgraphen, lösen mittels MCQD, Rücktransformation der Lösung
 - MCQD: unter GNU General Public License veröffentlichter Algorithmus zur Lösung des MaxClique-Problems

- Beobachtung: Optimale Lösung des Labeling-Problems bildet Clique maximaler Größe im dualen Konfliktgraphen, da alle Labels sich gegenseitig nicht überlappen dürfen
- Aufbauen des dualen Konfliktgraphen, lösen mittels MCQD, Rücktransformation der Lösung
 - MCQD: unter GNU General Public License veröffentlichter Algorithmus zur Lösung des MaxClique-Problems

→ deutlich langsamer

- Zu Beginn nur Punkt-Constraints einfügen, keine Konflikt-Constraints
- Separationsheuristik:

```
sort candidates descending by number of conflicts
```

```
foreach candidate with  $\geq 1$  conflict:
```

```
    find max clique in this candidates conflicts
```

```
    add cut for all members of this clique
```

```
    remove conflicts of clique from graph
```

```
    break if "enough" cuts were generated
```

- Zu Beginn nur Punkt-Constraints einfügen, keine Konflikt-Constraints
- Separationsheuristik:

```
sort candidates descending by number of conflicts
```

```
foreach candidate with  $\geq 1$  conflict:
```

```
    find max clique in this candidates conflicts
```

```
    add cut for all members of this clique
```

```
    remove conflicts of clique from graph
```

```
    break if "enough" cuts were generated
```

→ deutlich langsamer

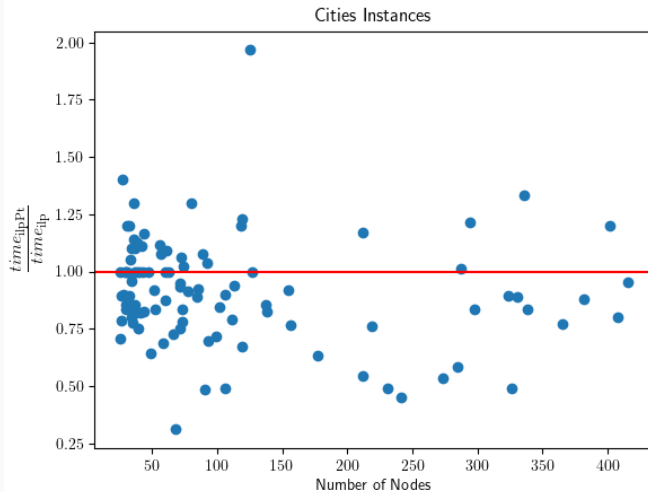
- Beobachtung 1: einfach zu ermitteln, ob Punkt in einem anderen Label liegt
- Beobachtung 2: wenn Punkt in Label liegt, hat dieses Konflikte mit allen Kandidaten des Punktes

- Beobachtung 1: einfach zu ermitteln, ob Punkt in einem anderen Label liegt
- Beobachtung 2: wenn Punkt in Label liegt, hat dieses Konflikte mit allen Kandidaten des Punktes

→ Simple Verbesserung: Bei Generierung der Punkt-Constraints direkt alle Kandidaten mit einbeziehen, die den Punkt umschließen

⇒ stärkere Ungleichungen

ILP mit stärkeren Constraints



Yay! Eine Verbesserung! (zumindest meistens...)