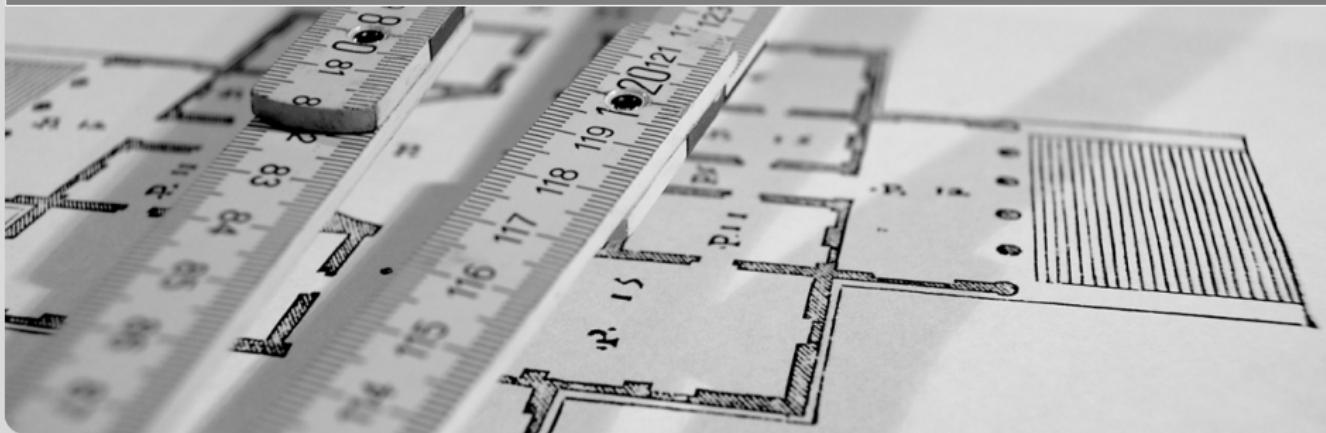


# Engineering FPT-based Edge Editing Algorithms

Sven Zühlsdorf | 13. Oktober 2017

INSTITUT FÜR THEORETISCHE INFORMATIK



- 1 Einführung
- 2 Editeralgorithmen
  - Basic Editing Algorithm
  - Redundant Editing Algorithm
  - Untere Schranken
- 3 Teilgraphauswahl
  - First & Most edited
  - Single edge editing
- 4 Evaluation
  - weiteres
- 5 Zusammenfassung
- 6 Template

- Identifizieren von Gruppen in sozialen Netzwerken
  - Definition „Gruppe“ offenes Problem
  - Ansatz: Hinzufügen/entfernen von Kanten bis bestimmte Eigenschaft gegeben
- Grapheditieren
- Edit: Einfügen/entfernen einer Kante

## Notation

- Anwenden einer Menge von Edits:  $G_L = G \Delta L$
- $E_L = E \Delta L = (E \setminus L) \cup (L \setminus E)$

# Beispiele

- Alle Zusammenhangskomponenten sind
  - vollständige Cliques
  - Quasi-Threshold-Graphen
- Äquivalent: Graph enthält keinen ... als induzierten Teilgraphen
  - $P_3$
  - $P_4$  oder  $C_4$
- Allgemeiner: Graph ist  $\mathcal{F}$ -frei
- TODO: Bild  $P_4$ ,  $C_4$ , nicht  $P/C4$

## $\mathcal{F}$ -free Editing Search Problem

- Graph  $G$ , Menge von verbotenen Teilgraphen  $\mathcal{F}$
- minimale Menge  $L$  von Edits, so dass  $G \triangle L$   $\mathcal{F}$ -frei

## $\mathcal{F}$ -free Editing Decision Problem

- Graph  $G$ , Menge von verbotenen Teilgraphen  $\mathcal{F}$ , Anzahl erlaubter Edits  $k$
- Kann  $G$  mit maximal  $k$  Edits in einen  $\mathcal{F}$ -freien Graph überführt werden?

Im Allgemeinen NP-vollständig, für bestimmte  $\mathcal{F}$  einfacher

Vorgestellt von [Cai96]

## Basic Editing Algorithm

- ① Wähle verbotenen Teilgraphen
- ② Für jedes Knotenpaar des Teilgraphens:
  - ① Editiere Knotenpaar
  - ② Rekursion
  - ③ Rekursion erfolgreich? return true
  - ④ Edit rückgängig machen
- ③ return false

# Basic Editing Algorithm

- Laufzeit:  $O(s^k \text{ poly}(n))$ ,  $s := \max\left\{\frac{|V_F|(|V_F|-1)}{2} \mid F \in \mathcal{F}\right\}$   
→ Fixed Parameter Tracktable in  $k$
- Optimierungen:
  - Doppeltes Editieren von Knotenpaaren nicht sinnvoll
  - Für  $\mathcal{F} = \{P_I, C_I\}$ ,  $I \geq 4$  kann immer ein Knotenpaar ignoriert werden  
→  $s := \frac{I(I-1)}{2} - 1$

# Beispiel

TODO

# Redundant Editing Algorithm

- Beobachtung: Basic testet gleiche Situation mehrfach
    - Selbe Menge an Edits, aber in anderer Reihenfolge ausgeführt
  - Ziel: Redundante Tests vermeiden
  - Beobachtung: Wenn Algorithmus aus Rekursion zurückkehrt, kann das editierte Knotenpaar nicht zu einer Lösung führen
  - Idee: Verhindern, dass die folgenden Rekursionen das Knotenpaar editieren
- Einführung von markierten Knotenpaaren
- sind nicht editiert, dürfen aber nicht mehr editiert werden

## Redundant Editing Algorithm

- ① Wähle verbotenen Teilgraphen
- ② Für jedes Knotenpaar des Teilgraphens:
  - ① Editiere Knotenpaar
  - ② Rekursion
  - ③ Rekursion erfolgreich? return true
  - ④ Edit rückgängig machen, Knotenpaar markieren
- ③ Markierungen entfernen
- ④ return false

- Greedy knotendisjunktes Packing von verbotenen Teilgraphen
  - maximal  $\lfloor \frac{n}{x} \rfloor$ ,  $x :=$  Anzahl Knoten im kleinsten verbotenen Teilgraphen
- Besser: Knotenpaardisjunkt
  - maximal  $\lfloor \frac{n(n-1)}{x(x-1)} \rfloor$
  - Optimierung: Überlappung von bereits editierten/markierten Kanten erlaubt
- Noch besser: Hitting-Set Ansätze
  - non-greedy Variante zum knotenpaardisjunkten Packing
  - Aber: NP-vollständig → Heuristiken

- Finden eines verbotenen Teilgraphen
- Laufzeit:  $O(n^x \cdot \text{Isomorphietest})$ ,  $x := \text{Anzahl Knoten im größten verbotenen Teilgraph}$
- Bessere Laufzeiten für bestimmte  $\mathcal{F}$ :
  - $\mathcal{F} = \{P_5\}$ :  $O(m^2\Delta)$  [Boh15]
  - $\mathcal{F} = \{P_5, C_5\}$ :  $O(n^2m)$  [Sch15]
  - $\mathcal{F} = \{P_4, C_4\}$ : linear [BHSW15]

Idee: Zusätzlicher polynomieller Aufwand um exponentiellen Aufwand zu verhindern

- Benutzen des meisteditiersten verbotenen Teilgraphen
  - Idee: weniger Rekursionen → bessere Laufzeit
  - Erfordert Aufzählen aller verbotenen Teilgraphen

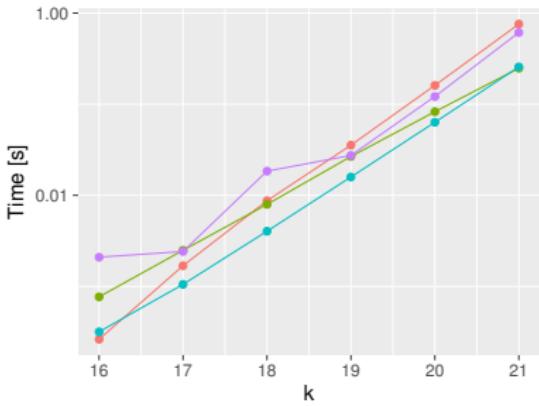
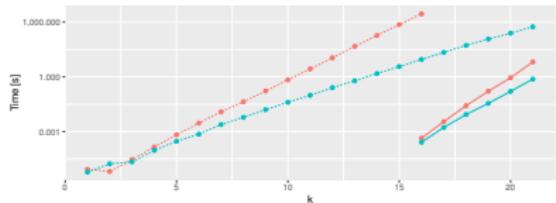
- Beobachtung:
  - Beweis Unlösbarkeit mit  $k - 1$  wichtiger als Lösung für  $k$  finden
  - Strengere Schranken brechen schlechte Zweige schneller ab
- Neuer Ansatz: Gezieltes Verschärfen der unteren Schranke
- Idee: Wähle Knotenpaar das die untere Schranke erhöht, egal ob editiert oder markiert
- Immer passendes Knotenpaar vorhanden?
  - Laufzeit Editing Problem  $O(4^k \text{ poly}(n))$
  - Existenz von passendem Knotenpaar nicht garantiert
  - Ggf. Rückfall auf meist editierten Teilgraphen
  - Experimente: in max 2% aller Fälle
- TODO: Beispiel

## Redundant Editing Algorithm mit single edge editing

- ① Wähle passendes Knotenpaar
- ② Kein passendes Knotenpaar? „normale“ Teilgraphauswahl
- ③ Für jedes Knotenpaar des Teilgraphens:
- ④ Editiere Knotenpaar
- ⑤ Rekursion mit  $k - 1$
- ⑥ Rekursion erfolgreich? return true
- ⑦ Edit rückgängig machen, Knotenpaar markieren
- ⑧ Rekursion ohne  $k$  zu verringern
- ⑨ Rekursion erfolgreich? return true
- ⑩ Markierung entfernen
- ⑪ return false

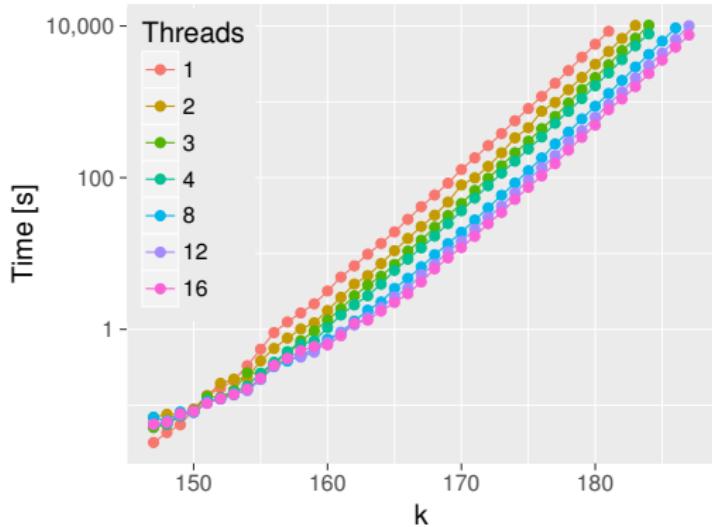
# Evaluation

- selbe Graphen wie [NG13]
- compute11
  - 2 Intel Xeon E5-2670 (16 Kerne @ 2.6 GHz)
  - 64 GiB RAM
- $\mathcal{F} = \{P_4, C_4\}$
- $k$  erhöhen bis Graph gelöst
- Abbruch nach 3 Stunden



- Einige Edits Teil von allen Lösungen
- Anzahl unterschiedlicher Edits  $< 2 * \text{min Edits}$

lineare Beschleunigung, aber exponentielles Problem  
→ einige  $k$  mehr lösbar



## Redundant Editing Algorithm

- Mehrere Größenordnungen schneller

## Single edge editing

- Für  $\mathcal{F} = \{P_4, C_4\}$  sinnvoll

Dadurch: Exakte Lösungen für bisher ungelöste Graphen

Graph	Gelöst	Edits	Zeit [s]	Threads
karate	ja	21	0.25	1
lesmis	ja	60	61.28	1
grassweb	ja	34	980.03	1
dolphins	nein	$64 < x \leq 72$	8632.32	16
football	nein	$187 < x \leq 251$	7594.89	16

Obere Schranken aus [BHSW15]

# Example slide A

- PCM, Citation: [BHSW15]
- Bullet point 2

## Block 1

- Bullet point 1
- Bullet point 2

## Example 1

- Bullet point 1
- Bullet point 2

## Alert 1

- Bullet point 1
- Bullet point 2

-  Ulrik Brandes, Michael Hamann, Ben Strasser und Dorothea Wagner: *Fast quasi-threshold editing.*  
In: *Algorithms-ESA 2015*, Seiten 251–262. Springer, 2015.
-  Bohlmann, Felix: *Graphclustern durch Zerstören langer induzierter Pfade.*  
Bachelor Thesis, Technische Universität Berlin, 2015.
-  Cai, Leizhen: *Fixed-parameter tractability of graph modification problems for hereditary properties.*  
Information Processing Letters, 58(4):171–176, 1996.
-  James Nastos und Yong Gao: *Familial groups in social networks.*  
Social Networks, 35(3):439–450, 2013.

-  Schoch, Philipp: *Editing to ( $P_5$ ,  $C_5$ )-free Graphs - a Model for Community Detection?*  
Bachelor Thesis, Karlsruhe Institute of Technology, October 2015.