# Algorithms and Experiments for Parameterized Approaches to Hard Graph Problems

Falk Hüffner

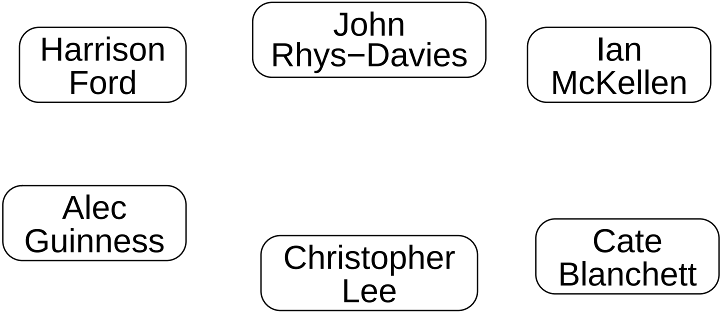
Friedrich-Schiller-Universität Jena
Institut für Informatik

19. Dezember 2007

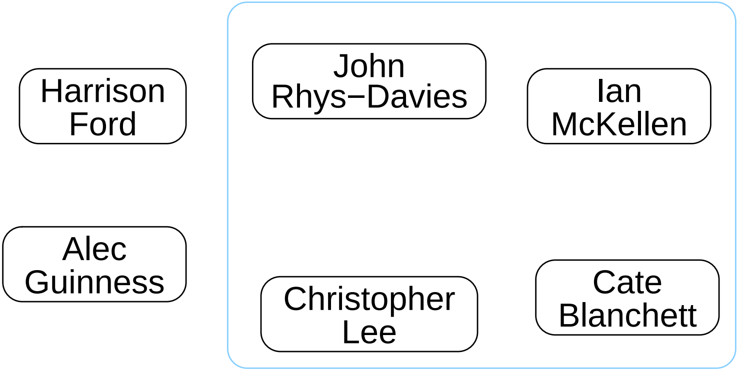
# Übersicht

- Datenreduktion
  - CLIQUE COVER
- Iterative Kompression
  - CLUSTER VERTEX DELETION
  - BALANCED SUBGRAPH
- Farbkodierung
  - MINIMUM-WEIGHT PATH

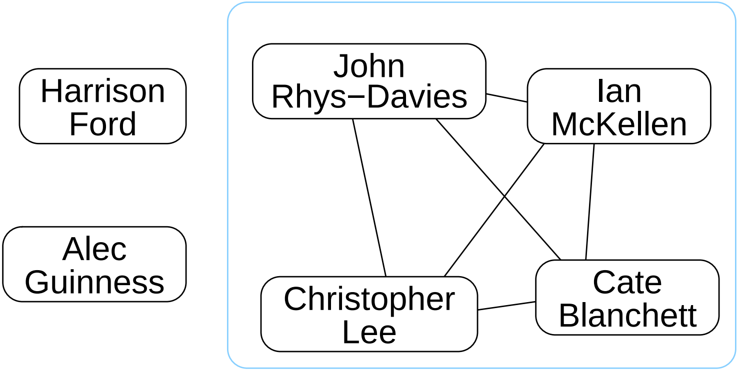
# Schauspieler-Graph



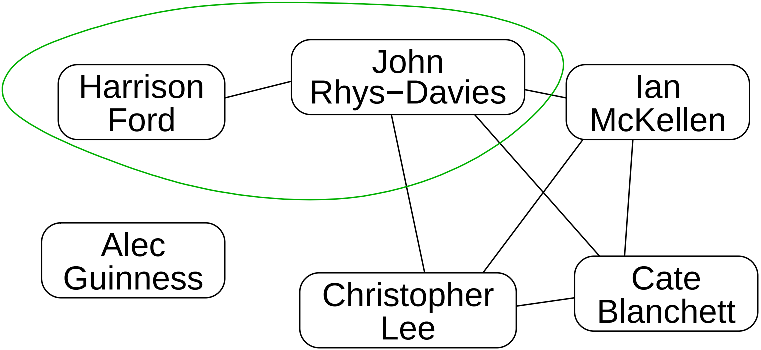
# Schauspieler-Graph



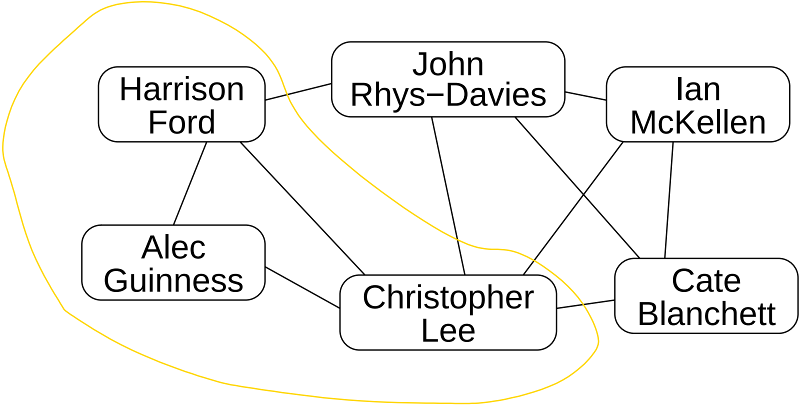
# Schauspieler-Graph



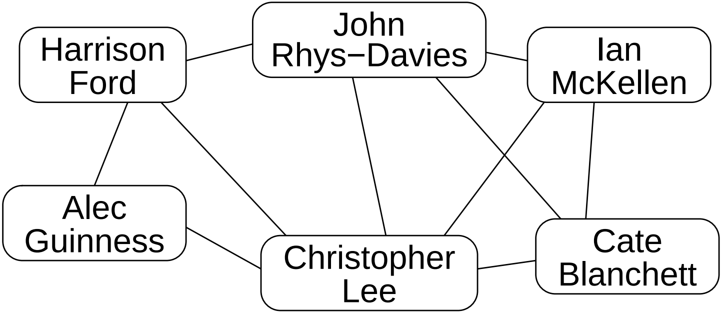
# Schauspieler-Graph



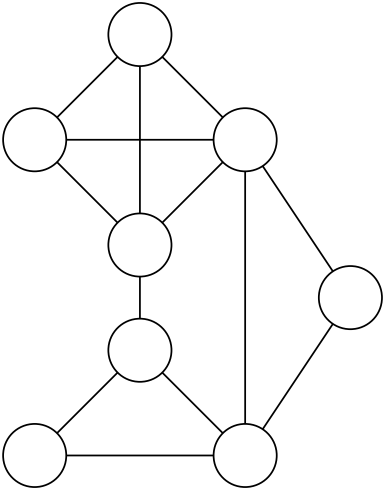
# Schauspieler-Graph



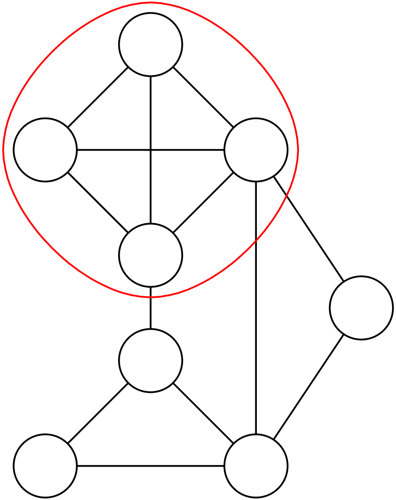
# Schauspieler-Graph



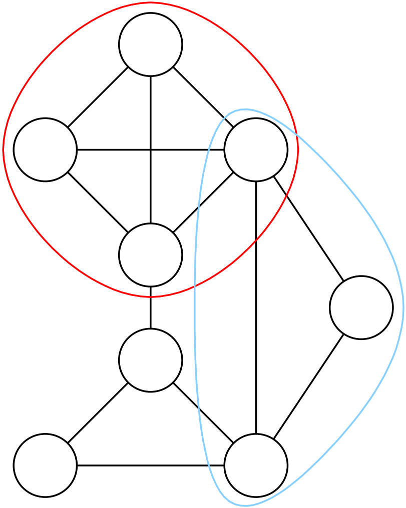
# Clique Cover



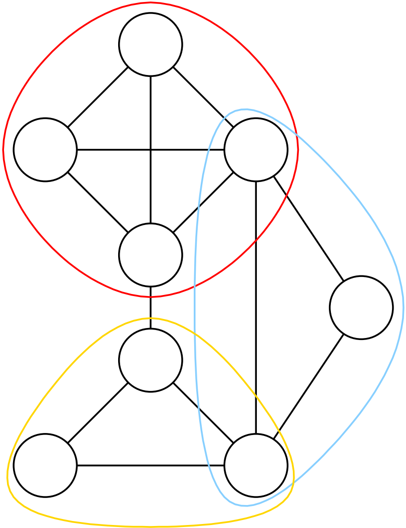
# Clique Cover



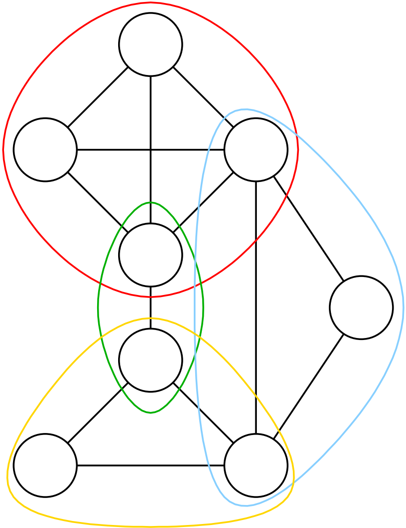
# Clique Cover



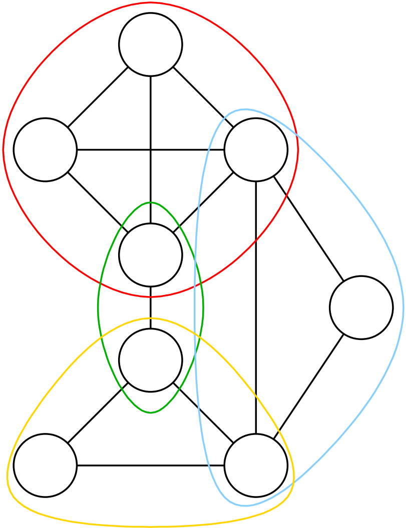
# Clique Cover



# Clique Cover



# Clique Cover



**Clique Cover**

**Eingabe:** Ein ungerichteter Graph $G = (V, E)$.
**Aufgabe:** Finde eine kleinstmögliche Zahl $k$ von Cliquen, sodass jede Kante in mindestens einer Clique enthalten ist.

# Clique Cover

## Anwendungen

- Compiler-Optimierung,
- geometrische Berechnungsprobleme,
- Statistik-Visualisierung, ...

# Clique Cover

## Anwendungen

- Compiler-Optimierung,
- geometrische Berechnungsprobleme,
- Statistik-Visualisierung, ...

## Theorem ([KELLERMAN, IBM Tech. Discl. Bull., 1973])

CLIQUE COVER *ist NP-schwer.*

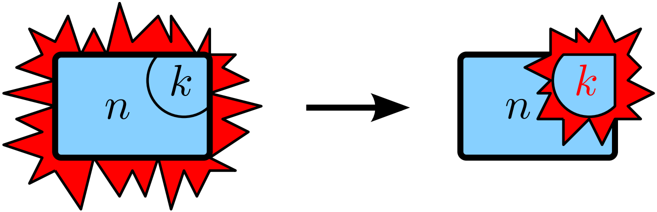
# NP-schwere Probleme

- In der theoretischen Informatik wird „effizient lösbar“ mit „in polynomiell beschränkter Zeit lösbar“ gleichgesetzt.
- Für keines der tausenden bekannten NP-schweren Probleme ist ein effizienter Lösungsalgorithmus bekannt.
- Vermutung: es gibt keine effizienten Lösungsalgorithmen!

# Parametrisierte Komplexität

## Idee: Zweidimensionale Sichtweise

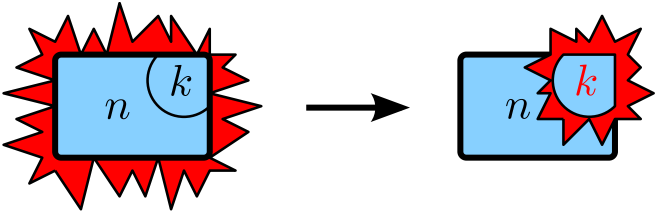
Zusätzlich zur Eingabegröße $n$ betrachte einen Parameter $k$.



# Parametrisierte Komplexität

## Idee: Zweidimensionale Sichtweise

Zusätzlich zur Eingabegröße $n$ betrachte einen Parameter $k$.




## Definition

Ein Problem ist *festparameter-handhabbar*, wenn eine Instanz der Größe $n$ in Zeit $f(k) \cdot n^{O(1)}$ gelöst werden kann.

# Einfache Datenreduktionsregeln für Clique Cover

**Regel 1**

Lösche isolierte Knoten.

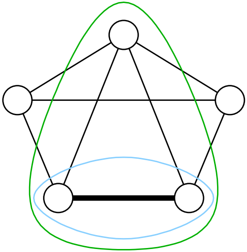
# Einfache Datenreduktionsregeln für Clique Cover

**Regel 1**

Lösche isolierte Knoten.

**Regel 2**

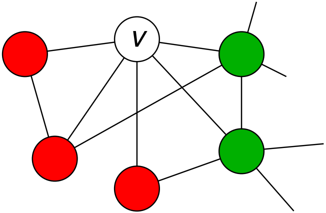
Wenn eine Kante $\{u, v\}$ nur in genau einer maximalen Clique $C$ enthalten ist, dann füge $C$ zur Lösung hinzu.



# Gefangene/Ausgänge-Reduktionsregel für Clique Cover

Partitioniere die Nachbarschaft eines Knoten $v$ in:

- Gefangene $p$ mit $N(p) \subseteq N(v)$ und
- Ausgänge $x$ mit $N(x) \setminus N(v) \neq \emptyset$.




**Regel 3**

Wenn alle Ausgänge mindestens einen Gefangenen als Nachbarn haben, dann lösche $v$.

# Festparameter-Handhabbarkeit von Clique Cover

**Theorem**

*Nach Anwendung aller Reduktionsregeln hat die* CLIQUE COVER*-Instanz höchstens* $2^k$ *Knoten, d. h.* CLIQUE COVER *hat einen Problemkern.*

# Festparameter-Handhabbarkeit von Clique Cover

## Theorem

*Nach Anwendung aller Reduktionsregeln hat die* CLIQUE COVER*-Instanz höchstens* $2^k$ *Knoten, d. h.* CLIQUE COVER *hat einen Problemkern.*

## Korroloar

CLIQUE COVER ist festparameter-handhabbar bezüglich des Parameters $k$.

# Clique Cover-Experimente

## Getreideertragsdaten

Alle Testinstanzen mit bis zu 124 Knoten und 4 847 Kanten wurden in $< 1\,\mathrm{s}$ gelöst.

# Clique Cover-Experimente

## Getreideertragsdaten

Alle Testinstanzen mit bis zu 124 Knoten und 4 847 Kanten wurden in $< 1\,\mathrm{s}$ gelöst.

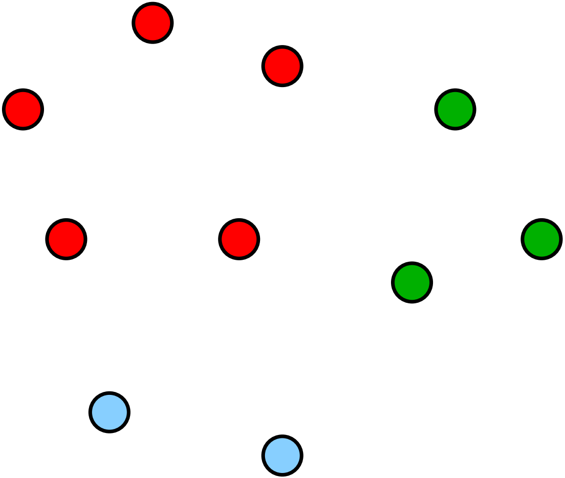
## Simulierte Getreideertragsdaten

Unter 20 Sekunden für bis zu 300 Knoten.

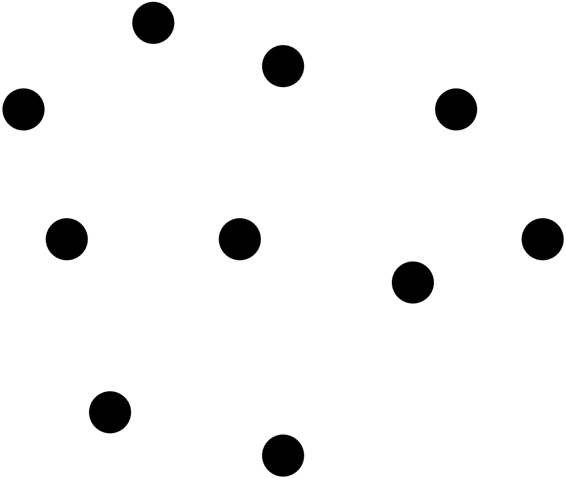
## Zufallsgraphen

Für Graphen mit wenigen Kanten ($m \approx n \ln n$): 10 000 Knoten in 7 min gelöst, die meisten allein durch die Datenreduktion.

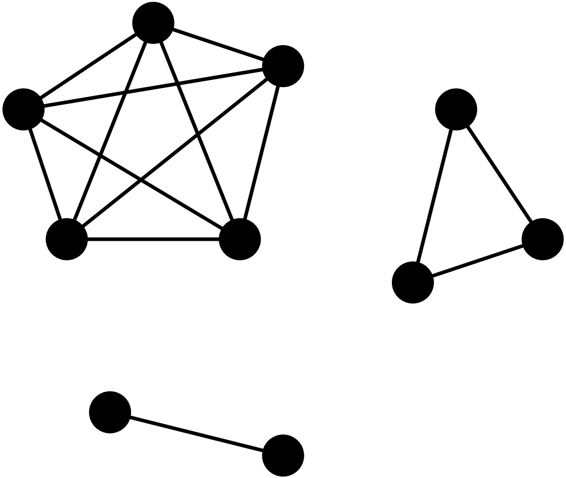
# Cluster Vertex Deletion



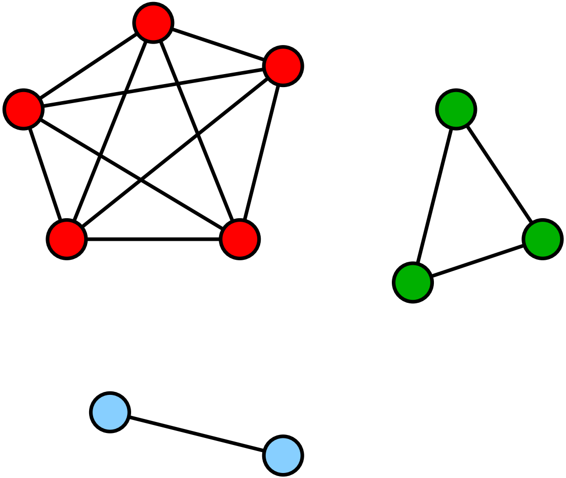
# Cluster Vertex Deletion



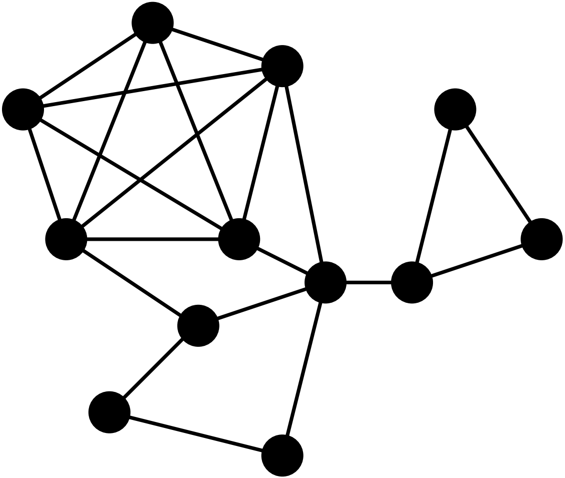
# Cluster Vertex Deletion



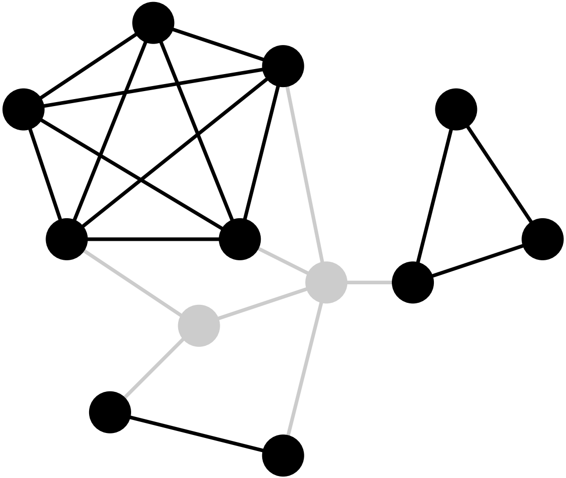
# Cluster Vertex Deletion



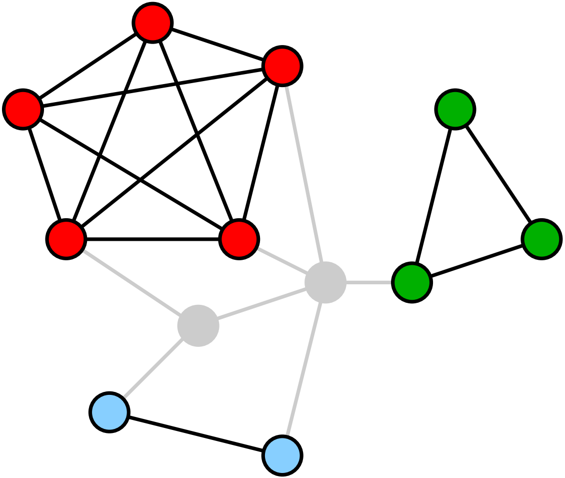
# Cluster Vertex Deletion



# Cluster Vertex Deletion



# Cluster Vertex Deletion



# Cluster Vertex Deletion

## Cluster Vertex Deletion

**Eingabe:** Ein ungerichteter Graph $G = (V, E)$.
**Aufgabe:** Lösche eine kleinstmögliche Zahl $k$ von Knoten, sodass ein Clustergraph übrig bleibt.

# Cluster Vertex Deletion

## Cluster Vertex Deletion

**Eingabe:** Ein ungerichteter Graph $G = (V, E)$.
**Aufgabe:** Lösche eine kleinstmögliche Zahl $k$ von Knoten, sodass ein Clustergraph übrig bleibt.

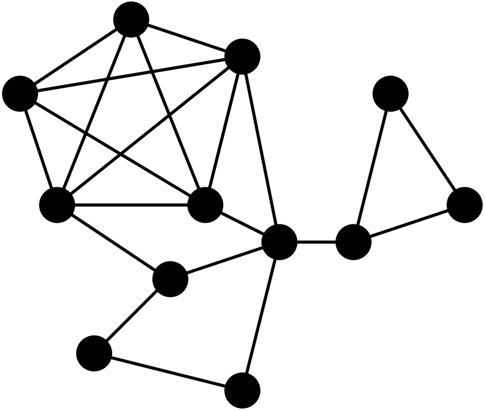
## Lemma

*$G$ ist Clustergraph $\iff$ $G$ enthält keinen induzierten $P_3$ ()*

# Iterative Kompression

## Idee: Induktion

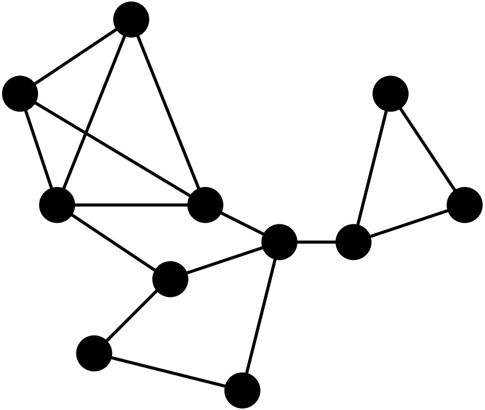
Nimm an, dass wir bereits die Lösung für die Instanz ohne einen Knoten haben.



# Iterative Kompression

## Idee: Induktion

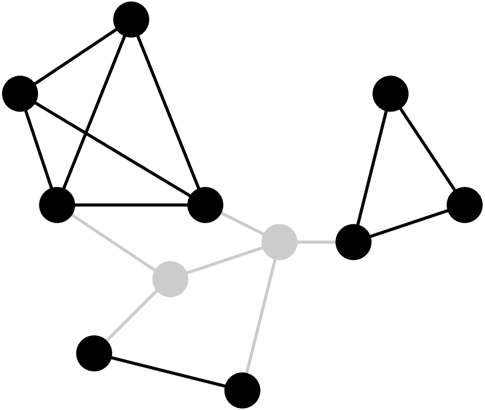
Nimm an, dass wir bereits die Lösung für die Instanz ohne einen Knoten haben.



# Iterative Kompression

## Idee: Induktion

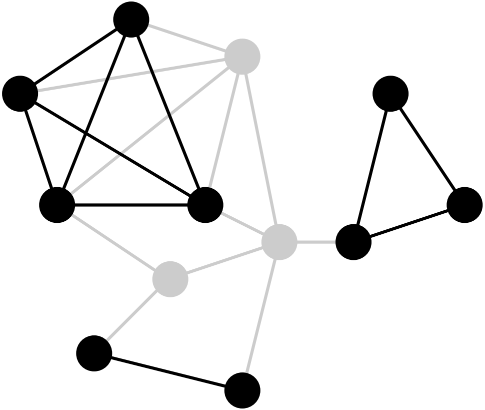
Nimm an, dass wir bereits die Lösung für die Instanz ohne einen Knoten haben.



# Iterative Kompression

## Idee: Induktion

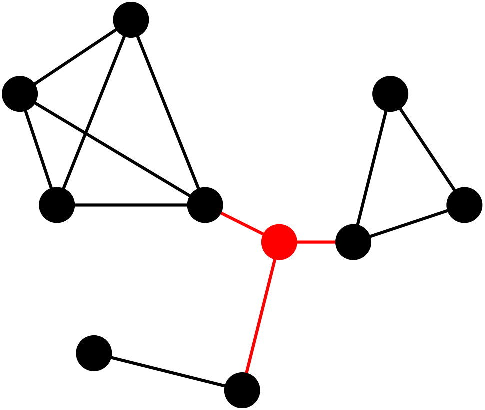
Nimm an, dass wir bereits die Lösung für die Instanz ohne einen Knoten haben.



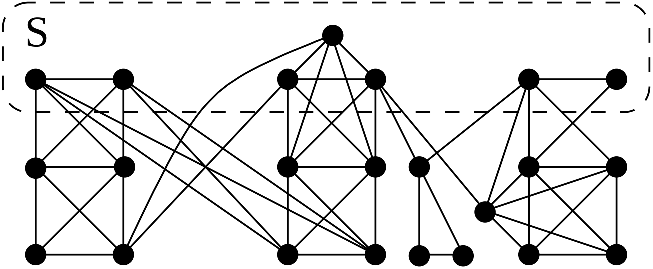
# Iterative Kompression

## Idee: Induktion

Nimm an, dass wir bereits die Lösung für die Instanz ohne einen Knoten haben.

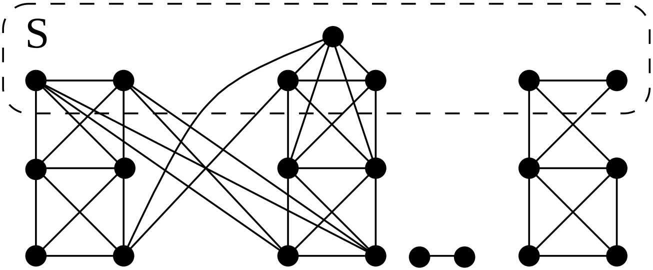


# Cluster Vertex Deletion-Kompression



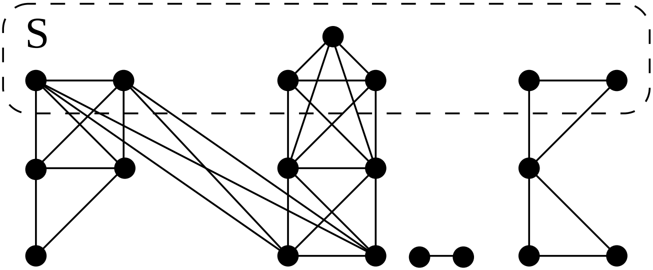

Lösche Knoten, die mit mehreren Clustern in $S$ verbunden sind

# Cluster Vertex Deletion-Kompression



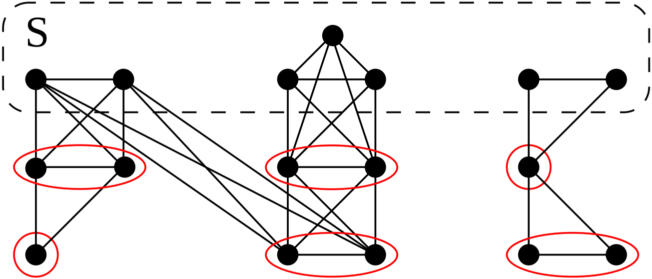

Lösche Knoten, die mit nur einigen Knoten eines Clusters in $S$ verbunden sind

# Cluster Vertex Deletion-Kompression



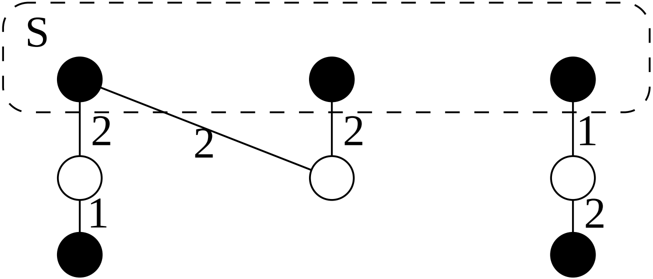

Lösche Clique-Komponenten

# Cluster Vertex Deletion-Kompression



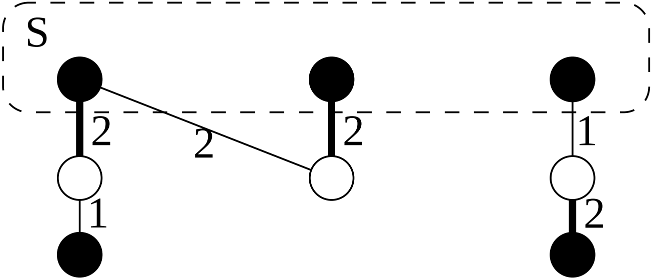

Klassifikation nach Nachbarschaft

# Cluster Vertex Deletion-Kompression



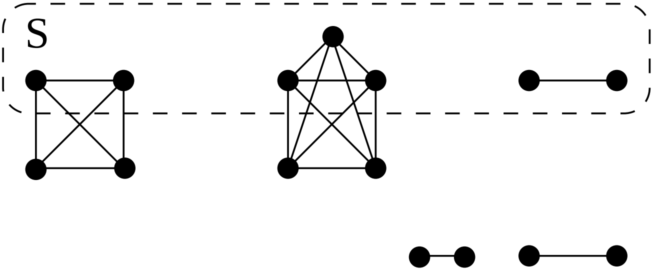

Matching-Instanz

# Cluster Vertex Deletion-Kompression




Matching-Lösung

# Cluster Vertex Deletion-Kompression



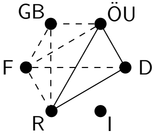

Gefundener Clustergraph

# Cluster Vertex Deletion

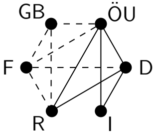
### Theorem

CLUSTER VERTEX DELETION *kann in Zeit* $O(2^k \cdot km\sqrt{n} \log n)$ *gelöst werden.*

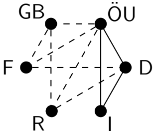
# Staatsbeziehungen



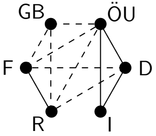

Dreikaiserbund 1872–81



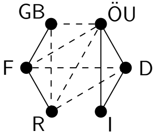

Dreibund 1882



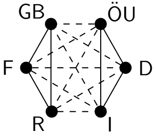

Deutsch–Russischer Streit 1890




Frz.–Russisches Bündnis 1891–94




Entente cordiale 1904




Britisch–Russisches Bündnis 1907

——— positive Beziehungen
- - - - negative Beziehungen

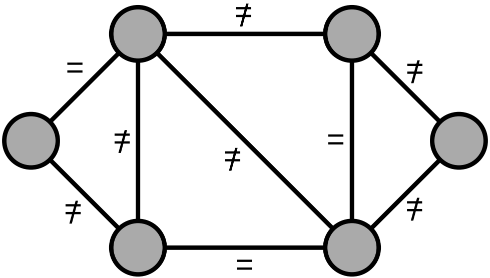
GB = Großbritannien
F = Frankreich
R = Russland

ÖU = Österreich–Ungarn
D = Deutschland
I = Italien

# Balancierte Graphen

## Definition

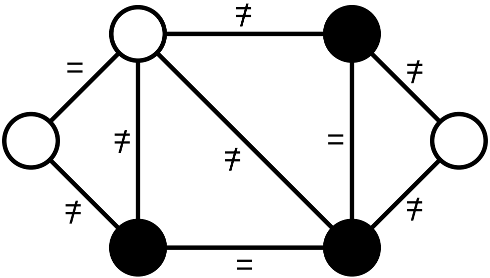
Ein Graph mit Kantenbeschriftungen $=$ oder $\neq$ ist balanciert, wenn seine Knoten mit zwei Farben so eingefärbt werden können, dass die Relation entlang jeder Kante eingehalten wird.



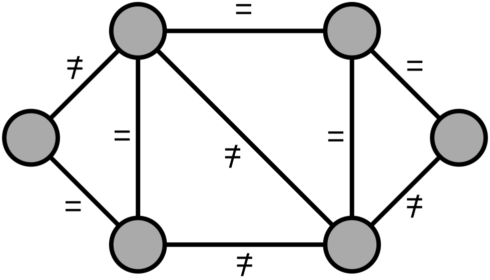
# Balancierte Graphen

## Definition

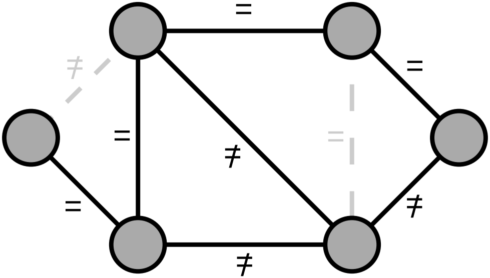
Ein Graph mit Kantenbeschriftungen $=$ oder $\neq$ ist balanciert, wenn seine Knoten mit zwei Farben so eingefärbt werden können, dass die Relation entlang jeder Kante eingehalten wird.



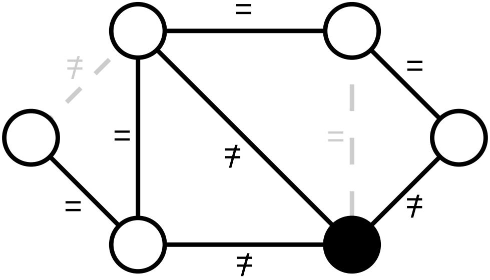
# Balanced Subgraph



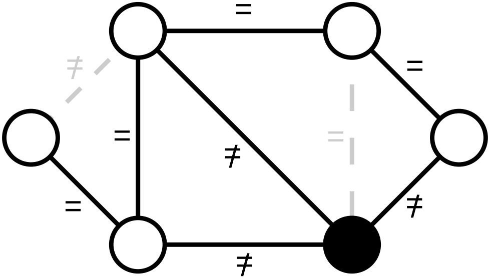
# Balanced Subgraph



# Balanced Subgraph



# Balanced Subgraph




**Definition** (Balanced Subgraph)

**Eingabe:** Ein Graph mit Kantenbeschriftungen $=$ oder $\neq$.
**Aufgabe:** Lösche möglichst wenig Kanten, sodass der Graph balanciert wird.

# Anwendungen von Balanced Subgraph

- Balance in Sozialen Netzwerken
  [Harary, Mich. Math. J. 1953]
- Minimalenergiezustände in magnetischen Materialien (Spin-Gläser)
  [Kasteleyn, J. Math. Phys. 1963]
- „Monotone Subsysteme" in biologischen Netzwerken
  [DasGupta et al., WEA 2006]
- Portfolio-Risikoanalyse
  [Harary et al., IMA J. Manag. Math. 2002]
- Stabilität von Fullerenen
  [Došlić&Vikičević, Discr. Appl. Math. 2007]
- Entwurf integrierter Schaltungen
  [Chiang et al., IEEE Trans. CAD of IC&Sys. 2007]

# Anwendungen von Balanced Subgraph

- Balance in Sozialen Netzwerken

  [Harary, Mich. Math. J. 1953]
- Minimalenergiezustände in magnetischen Materialien (Spin-Gläser)

  [Kasteleyn, J. Math. Phys. 1963]
- „Monotone Subsysteme“ in biologischen Netzwerken

  [DasGupta et al., WEA 2006]
- Portfolio-Risikoanalyse

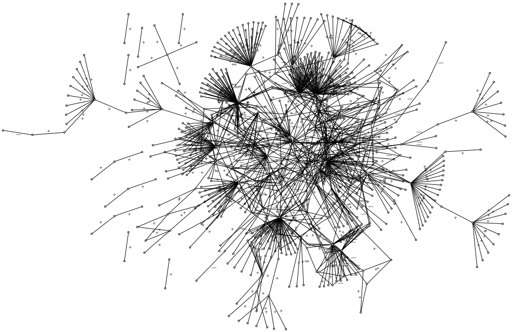
  [Harary et al., IMA J. Manag. Math. 2002]
- Stabilität von Fullerenen

  [Došlić&Vikičević, Discr. Appl. Math. 2007]
- Entwurf integrierter Schaltungen

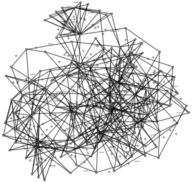
  [Chiang et al., IEEE Trans. CAD of IC&Sys. 2007]

Balanced Subgraph ist NP-schwer.

# Datenreduktion



$n = 690,\ m = 1082$



$n = 144,\ m = 405$

# Festparameter-Handhabbarkeit von Balanced Subgraph

**Theorem**

*Mittels iterativer Kompression kann* BALANCED SUBGRAPH *in Zeit* $O(2^k \cdot m^2)$ *gelöst werden.*

# Balanced Subgraph-Experimente

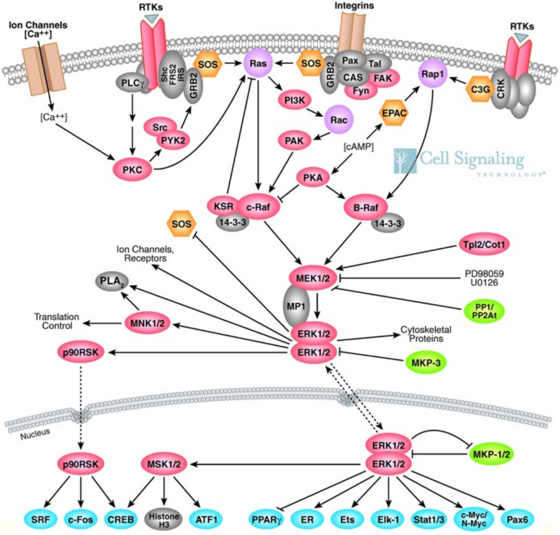
| | | | Approximation | | | Neuer Alg. | |
|---|---|---|---|---|---|---|---|
| Netzwerk | $n$ | $m$ | $k \geqslant$ | $k \leqslant$ | $t$ [min] | $k$ | $t$ [min] |
| EGFR | 330 | 855 | 196 | 219 | 7 | 210 | 108 |
| Hefe | 690 | 1082 | 0 | 43 | 77 | 41 | 1 |
| Macr. | 678 | 1582 | 218 | 383 | 44 | 374 | 1 |

# Balanced Subgraph-Experimente

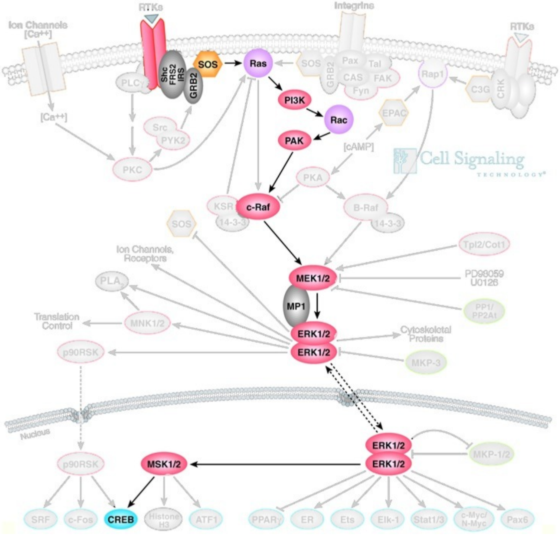
| | | | Approximation | | | Neuer Alg. | |
|---|---|---|---|---|---|---|---|
| Netzwerk | $n$ | $m$ | $k \geqslant$ | $k \leqslant$ | $t$ [min] | $k$ | $t$ [min] |
| EGFR | 330 | 855 | 196 | 219 | 7 | 210 | 108 |
| Hefe | 690 | 1082 | 0 | 43 | 77 | 41 | 1 |
| Macr. | 678 | 1582 | 218 | 383 | 44 | 374 | 1 |

- Ein Netzwerk mit 688 Knoten und 2 208 Kanten konnte nicht gelöst werden.

# Signalpfade in Proteininteraktionsnetzwerken



# Signalpfade in Proteininteraktionsnetzwerken



# Signalpfade

## Signalpfade

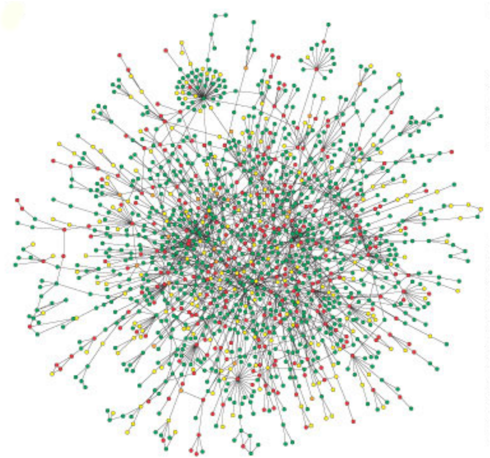
Ein Signalpfad ist eine Folge unterschiedlicher Proteine, sodass jedes mit dem vorigen stark wechselwirkt.

## Minimum-Weight Path

**Eingabe:** Graph $G = (V, E)$, Gewicht $w : E \to [0, 1]$, ganze Zahl $k > 0$

**Aufgabe:** Finde einen einfachen Pfad $v_1, \ldots, v_k$ der Länge $k$ in $G$, der $w(v_1, v_2) + \cdots + w(v_{k-1}, v_k)$ minimiert.

# Hefenetzwerk



4 400 Proteine, 14 300 Interaktionen, gesucht Pfade der Länge 5–15

# Minimum-Weight Path

**Theorem**

MINIMUM-WEIGHT PATH *ist NP-schwer* [GAREY&JOHNSON 1979].

**Idee**

Nutze die Tatsache, dass die gesuchten Pfade relativ kurz sind ($\approx$ 5–15): Beschränkung der kombinatorischen Explosion auf den Parameter „Pfadlänge“.

# Farbkodierung

Farbkodierung ist ein parametrisiertes Verfahren zur Lösung von MINIMUM-WEIGHT PATH

## Color-coding [ALON, YUSTER & ZWICK, J. ACM 1995]

- Färbe jeden Knoten des Graphen zufällig mit einer von $k$ Farben
- Hoffe, dass die Knoten im optimalen Pfad alle unterschiedliche Farben erhalten (bunt)
- Löse MINIMUM-WEIGHT PATH unter dieser Annahme (was erheblich schneller geht)
- Wiederhole diese Versuche, bis mit hoher Wahrscheinlichkeit mindestens einmal der Pfad tatsächlich bunt war.

# Mehr Farben

### Idee

Benutze $k + x$ anstelle von $k$ Farben.

# Mehr Farben

## Idee

Benutze $k + x$ anstelle von $k$ Farben.

## Theorem

MINIMUM-WEIGHT PATH *kann in Zeit* $O(-\ln \varepsilon \cdot 4.32^k |G|)$ *gelöst werden, indem man* $x = 0.3k$ *setzt.*

# Mehr Farben

## Idee

Benutze $k + x$ anstelle von $k$ Farben.

## Theorem

*Minimum-Weight Path kann in Zeit $O(-\ln \varepsilon \cdot 4.32^k |G|)$ gelöst werden, indem man $x = 0.3k$ setzt.*

Aber: Höherer Speicherbedarf

# Zustandsraumsuche

**Idee**

Fasse das dynamische Programmieren als Zustandsraumsuche auf (Kürzeste-Wege-Problem in einem implizit definierten Graphen).

# Zustandsraumsuche

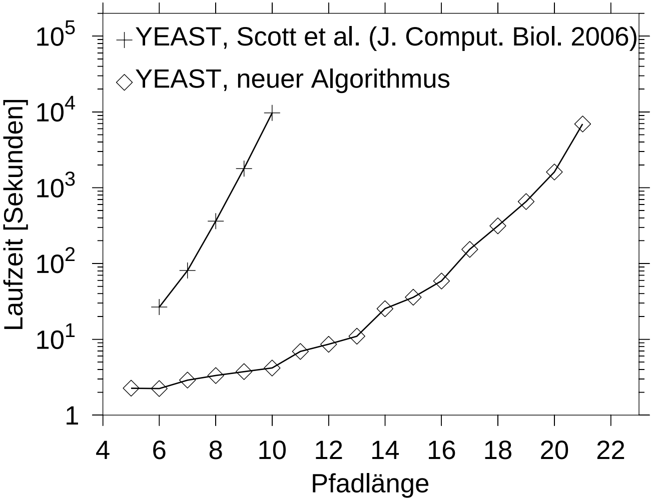
## Idee

Fasse das dynamische Programmieren als Zustandsraumsuche auf (Kürzeste-Wege-Problem in einem implizit definierten Graphen).

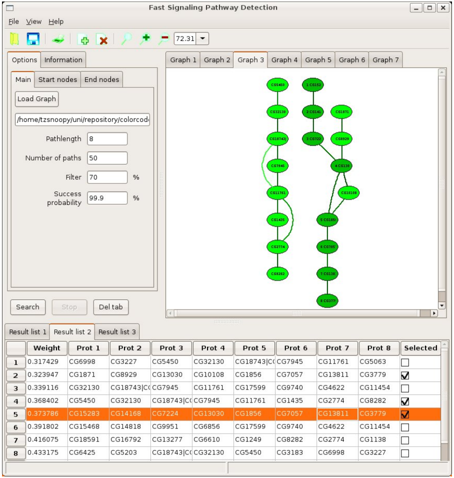
## Heuristische Evaluationsfunktionen

- können Zustände wegschneiden
- können die Suche leiten (A*)

# Hefenetzwerk



# Graphical user interface: FASPAD




Freie software, erhältlich bei

http://theinf1.informatik.uni-jena.de/faspad/

# Zusammenfassung

- Neue Festparameter-Algorithmen für eine Reihe von Problemen
- Implementierungen für
  - Clique Cover
  - Balanced Subgraph
  - Minimum-Weight Path
  - Vertex Bipartization

# Zusammenfassung

- Neue Festparameter-Algorithmen für eine Reihe von Problemen
- Implementierungen für
  - Clique Cover
  - Balanced Subgraph
  - Minimum-Weight Path
  - Vertex Bipartization

**Schlussfolgerung**

Der parametrisierte Ansatz ist ein nützlicher Leitfaden zum Entwurf praktisch einsetzbarer Algorithmen für NP-schwere Probleme.