Technische Universität Braunschweig

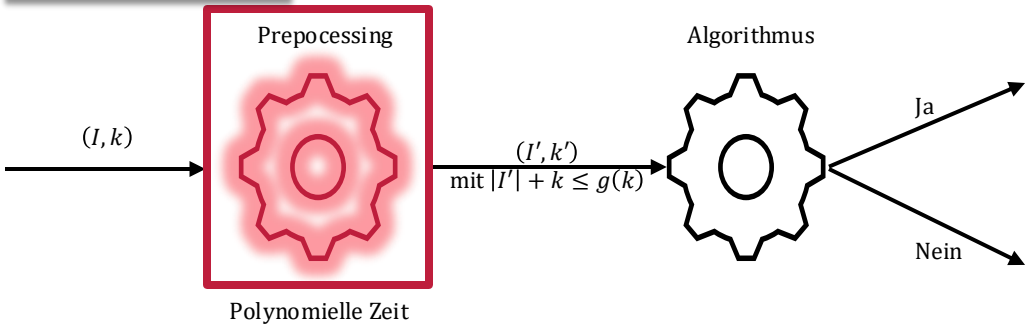
Parametrisierte Algorithmen

Arne Schmidt

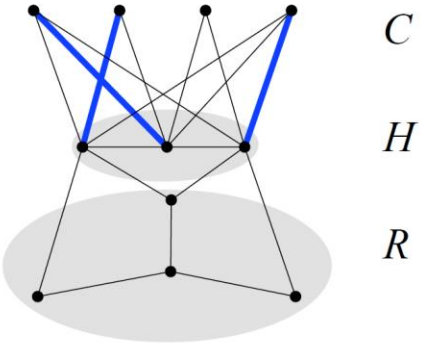
Wiederholung

Techniken

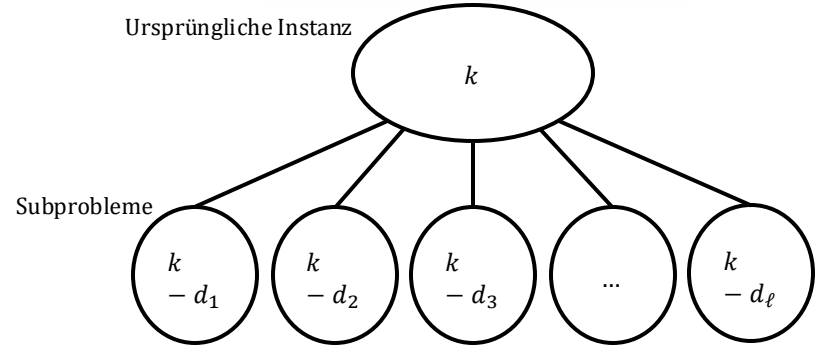
Kernelization



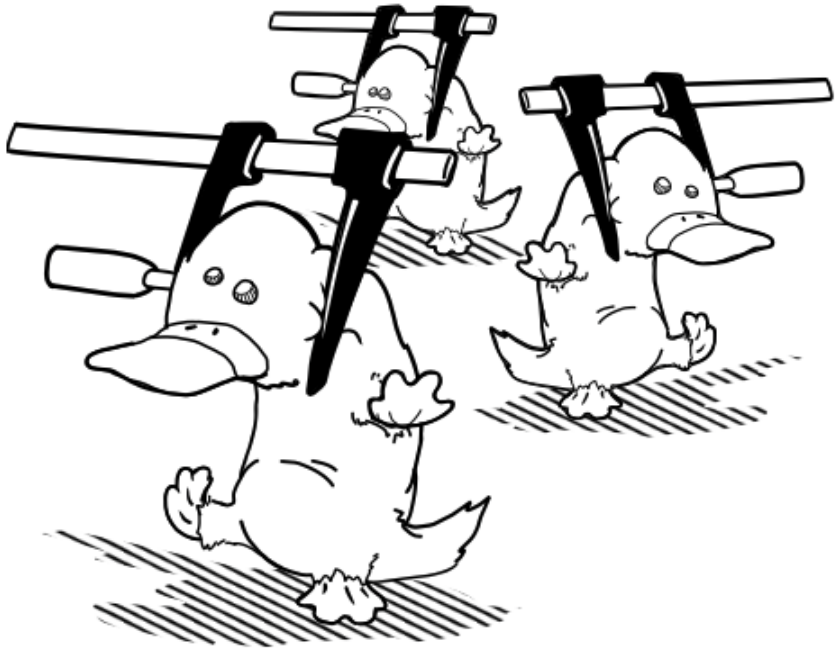
Crown Decomposition



Bounded Search Tree



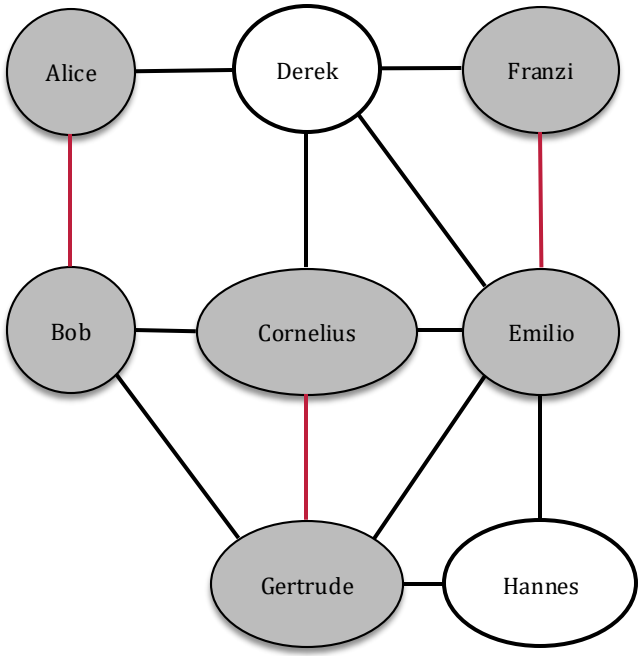
Heute



(Keine Sorge, wir werden keine Tiere verletzen!)

Kapitel 4 – Iterative Compression

Vertex Cover



Bestimme ein Vertex Cover VC folgendermaßen:

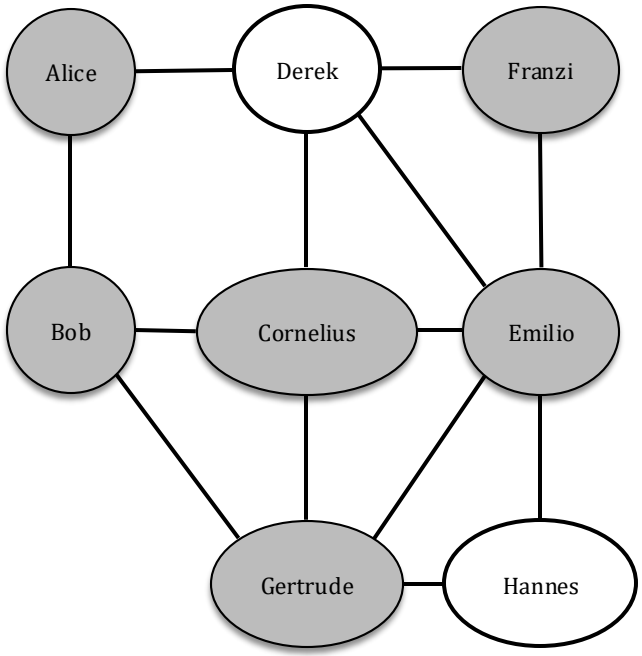
1. Sei $M \subseteq E$ ein inkl. maximales Matching
2. Setze $VC = \bigcup_{\{u,v\} \in M} \{u, v\}$

Lemma 4.1:
Sei $G = (V, E)$ ein Graph und VC^* ein kleinstes Vertex Cover. Für das über obigen Algorithmus erhaltene Vertex Cover VC gilt:

$$|VC| \leq 2|VC^*|$$

Korollar 4.2:
Ist $|VC| > 2k$, dann existiert kein Vertex Cover der Größe k.

Vertex Cover



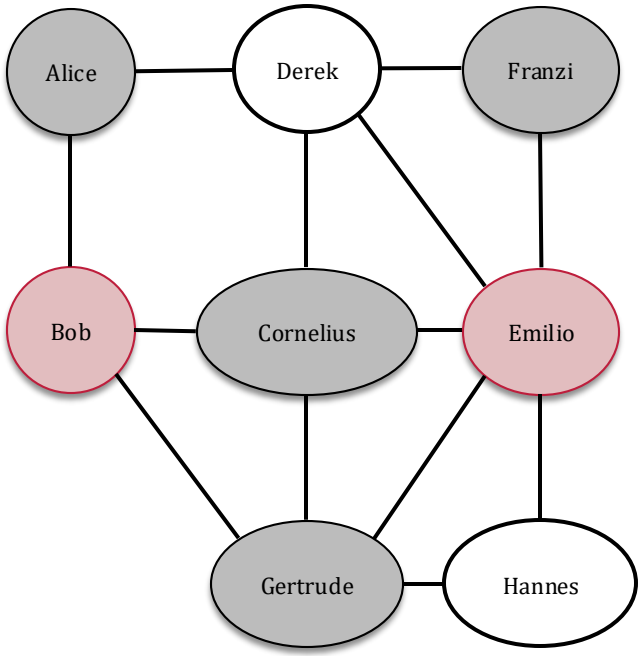
Gibt es ein Vertex Cover der Größe 5?
Falls ja:

- Sei VC das aktuelle Vertex Cover,
- Angenommen, $W \subset VC$ darf nicht im Vertex Cover liegen. Dann:
  1. Wie muss der Graph $G[W]$ aussehen?
  2. Wie sieht der Graph $G[V \setminus VC]$ aus?
  3. Wie sieht der Graph $G[W \cup (V \setminus VC)]$ aus?

Definition 4.3
$G[U]$ für eine Menge $U \subseteq V$ beschreibt den von U **induzierten Teilgraphen** von G, d.h. $G[U] = (U, E_U)$ mit

$$E_U = \{\{u, v\} \in E \mid u, v \in U\}$$

Vertex Cover

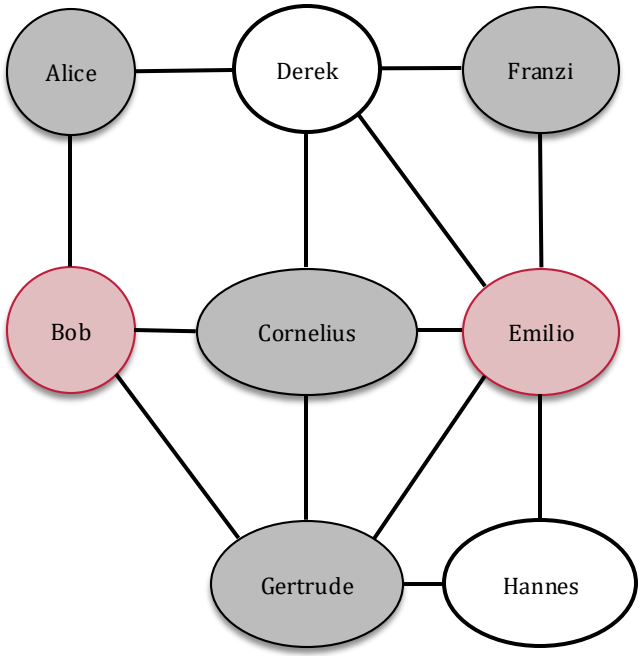


Lemma 4.4
Sei $G = (V, E)$, $VC \subseteq V$ ein Vertex Cover in G, und $W \subset VC$. Sucht man ein Vertex Cover in G ohne Knoten aus W zu benutzen, dann muss $G[W \cup (V \setminus VC)]$ ein bipartiter Graph sein.

Problem 4.5 (Disjoint Vertex Cover Problem)
Gegeben: $G = (V, E)$,Vertex Cover $VC \subseteq V, W \subset VC, k \in \mathbb{N}$
Frage: Existiert in $G[W \cup (V \setminus VC)]$ ein Vertex Cover der Größe $k - |VC \setminus W|$ ohne Knoten aus W zu benutzen?

Das Problem ist effizient entscheidbar (und wir können sogar das Vertex Cover zurückgeben, falls existent).

Vertex Cover



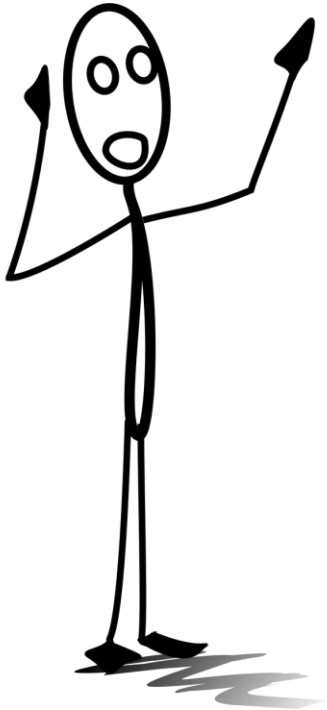
Wie wählt man die Menge W?

Lemma 4.1 und Korollar 4.2 garantieren uns:
Das gegebene Vertex Cover enthält maximal $2k$ Knoten

- Für jedes $W \subset VC$, löse das Disjoint-VC-Problem
- Falls eine Lösung existiert, sind wir fertig!
- Andernfalls probiere nächstes W.

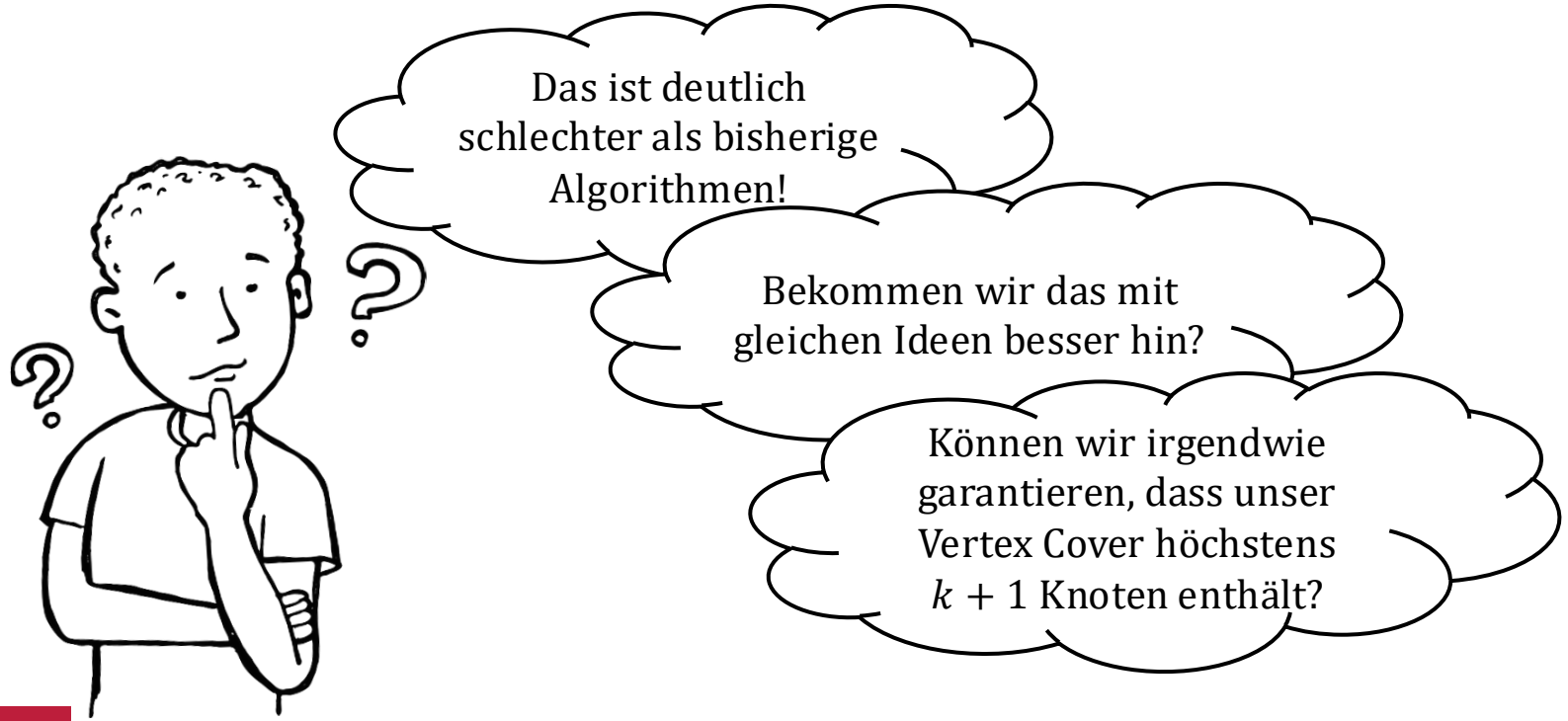
Laufzeit:
$O\left(2^{2k}\right)$ Teilmengen, je poly$(|G|) = O\left(n^{O(1)}\right)$ Zeit für Disjoint-VC.

Compression



Dieser Vorgang wird auch Kompression (engl. compression) genannt!

Verbesserungsideen?



Iterative Compression

Algorithmus 4.6 (Iterative Compression für VC)
Sei G_i der von den ersten i Knoten $V_i = \{v_1, \dots, v_i\}$ induzierte Graph.
Für G_k nehmen wir einfach alle Knoten in unser Vertex Cover.

Für $i = 1$ bis $n - k$ führe folgende Schritte durch:

1. Betrachte $G_{k+i} = G[V_{k+i-1} \cup \{v_{k+i}\}]$
2. Sind alle Kanten an v_{k+i} abgedeckt, führe nächste Iteration durch
3. Andernfalls, nehme v_{k+i} in das Vertex Cover auf
4. Führe Kompression durch:
    1. Falls möglich, führe nächste Iteration durch
    2. Anderfalls: Vertex-Cover-Instanz nicht lösbar.

n Iterationen

$O(2^k)$ Möglichkeiten für W
$\rightarrow O\left(2^k \operatorname{poly}(n)\right)$ Zeit

Theorem 4.7
Algorithmus 4.6 löst Vertex Cover in Zeit $O\left(2^k \operatorname{poly}(n)\right)$.

Kapitel 4.1 – Das Schema

$(\star)$-Kompressionsproblem

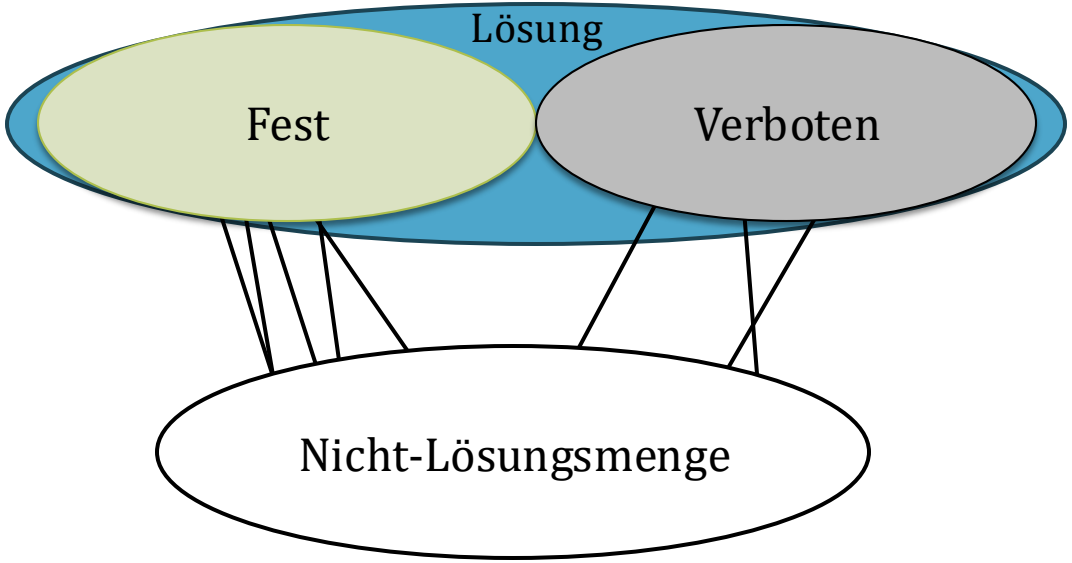
Problem 4.8
Gegeben: Instanz (I, k) eines Problems $(\star)$, $k \in \mathbb{N}$ und eine Lösung L_I der Größe $k + 1$.
Gesucht: Eine Lösung L'_I der Größe k oder die Ausgabe, das keine Lösung existiert.

Korollar 4.9:
Existiert für ein Problem $(\star)$ ein Algorithmus für Problem 4.8 in Zeit $O(f(k)n^c)$ für eine Konstante c, dann kann $(\star)$ in Zeit $O(f(k)n^{c+1})$ gelöst werden.

Algorithmus-Idee

Für eine gegebene Lösung:

- Halte einen Teil fest.
  → Fixer Teil der Lösung
- Prüfe, ob der restliche Teil der Lösung durch eine bessere Lösung der Nicht-Lösungsmenge ersetzt werden kann.
- Probiere alle Möglichkeiten für den fixen Teil.



$(\star)$-Disjoint-Problem

Problem 4.10
Gegeben: Instanz $(I = W \dot{\cup} X, k)$ eines Problems $(\star)$, mit W ist eine Lösung zu I.
Gesucht: Eine Lösung $L \subseteq X$ der Größe k oder die Ausgabe, das keine Lösung existiert.

Korollar 4.11:
Existiert für ein Problem $(\star)$ ein Algorithmus für Problem 4.10 in Zeit $O\left(g(k)n^{O(1)}\right)$, dann existiert ein Algorithmus für Problem 4.8 mit Laufzeit

$$O\left(\sum_{i=0}^{k} \binom{k+1}{i} g(k-i)n^{c+1}\right)$$

gelöst werden.

Speziell für $g(k) = \alpha^k$ für ein α, dann besitzt der $(\star)$-Kompressionsalgorithmus Laufzeit $O((1+\alpha)^k n^{O(1)}$

Kapitel 4.2 – Feedback Vertex Sets in Tournament Graphen