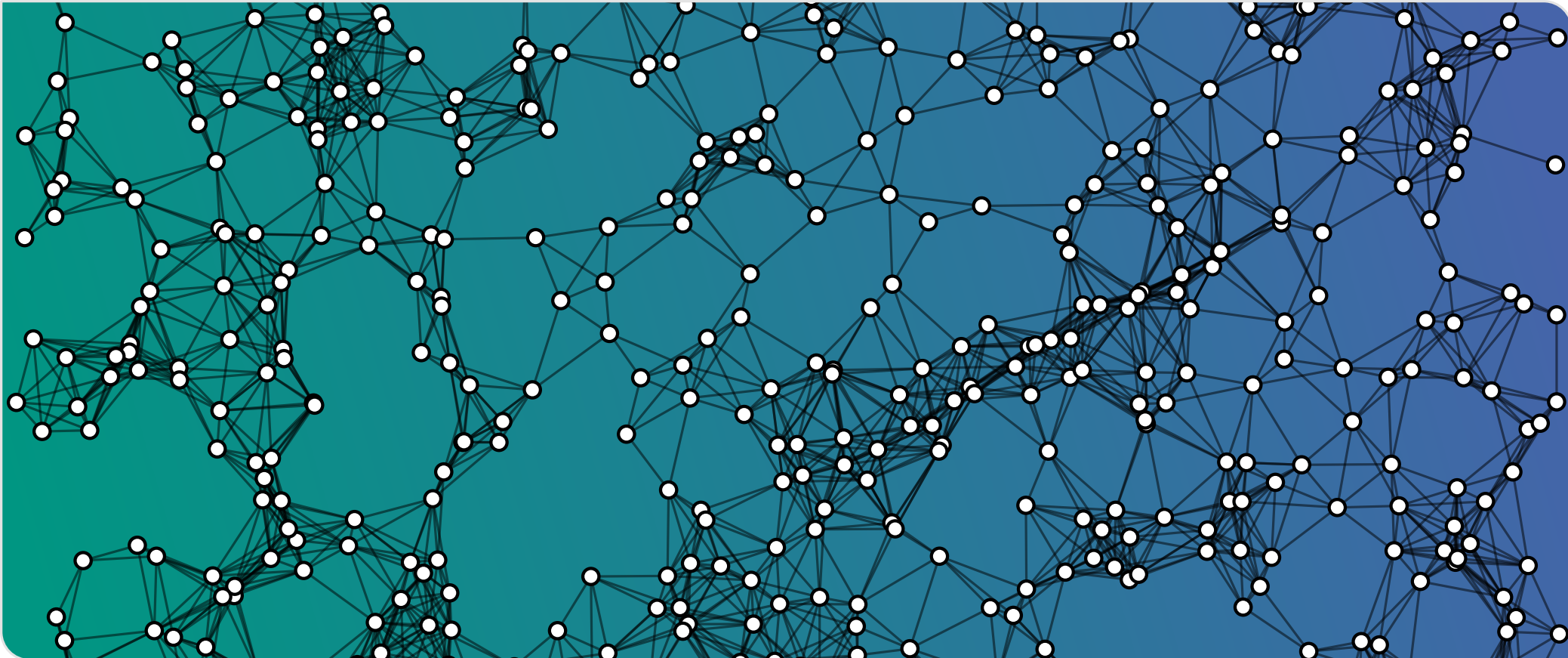


Parametrisierte Algorithmen

Beschränkte Suchbäume: bessere Verzweigung



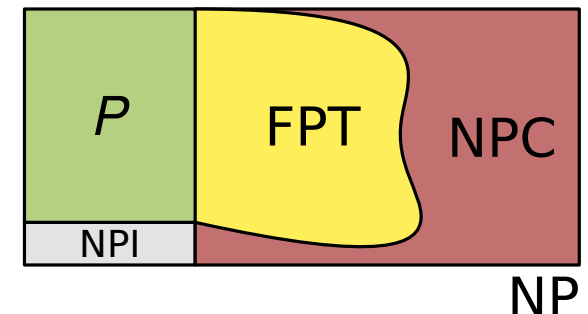
Nachtrag: Wie verhält sich FPT zu NP?

Formal korrekte Antwort: gar nicht

- $NP \cap FPT = \emptyset$ (NP enthält Probleme, FPT enthält parametrisierte Probleme)
- sei L entscheidbar; es gibt Parametrisierungen L' und L'' sodass:
 - $L' \in FPT$ (Laufzeit des Algos als Parameter)
 - $L'' \notin FPT$, außer wenn $L \in P$ (konstanter Parameter)

Typische Situation: Was wollen wir mit FPT erreichen?

- löse Probleme in NPC halbwegs effizient
- wir betrachten hauptsächlich Probleme in NP
- gute Vorstellung: FPT schlägt eine Brücke zwischen P und NPC
(auch wenn man das formal so nicht sagen kann)



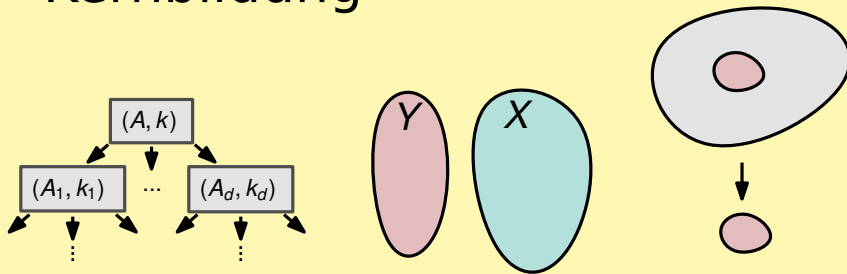
Zusatzhinweis: para-NP

- para-NP ist das Gegenstück zu NP für parametrisierte Probleme
- FPT verhält sich zu para-NP wie P zu NP

Inhalt

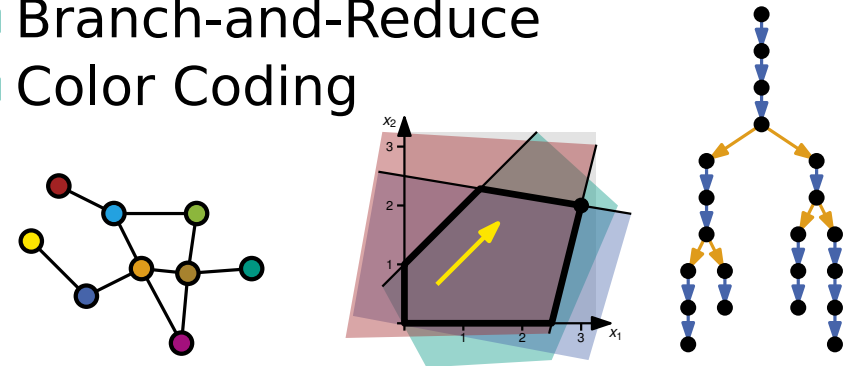
Basic Toolbox

- beschränkte Suchbäume
- iterative Kompression
- Kernbildung



Erweiterte Toolbox

- lineare Programme
- Branch-and-Reduce
- Color Coding



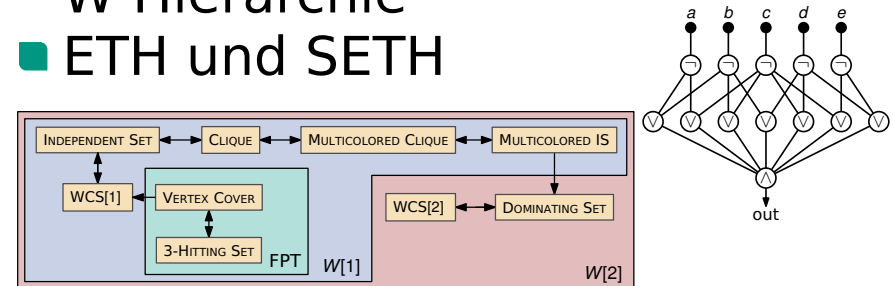
Baumweite

- dynamische Programme
- chordale & planare Graphen
- Courcelles Theorem



Untere Schranken

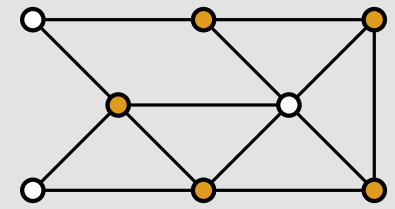
- parametrisierte Reduktionen
- boolesche Schaltkreise und die W-Hierarchie
- ETH und SETH



Wiederholung: Beschränkter Suchbaum

Problem: VERTEX COVER

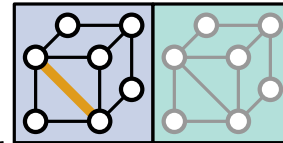
Gegeben sind ein Graph $G = (V, E)$ und ein Parameter k .
 Gibt es ein Vertex Cover der Größe k ?
 (Knotenmenge $V' \subseteq V$ mit $e \cap V' \neq \emptyset$ für alle $e \in E$)



noch zu überdeckender Teilgraph

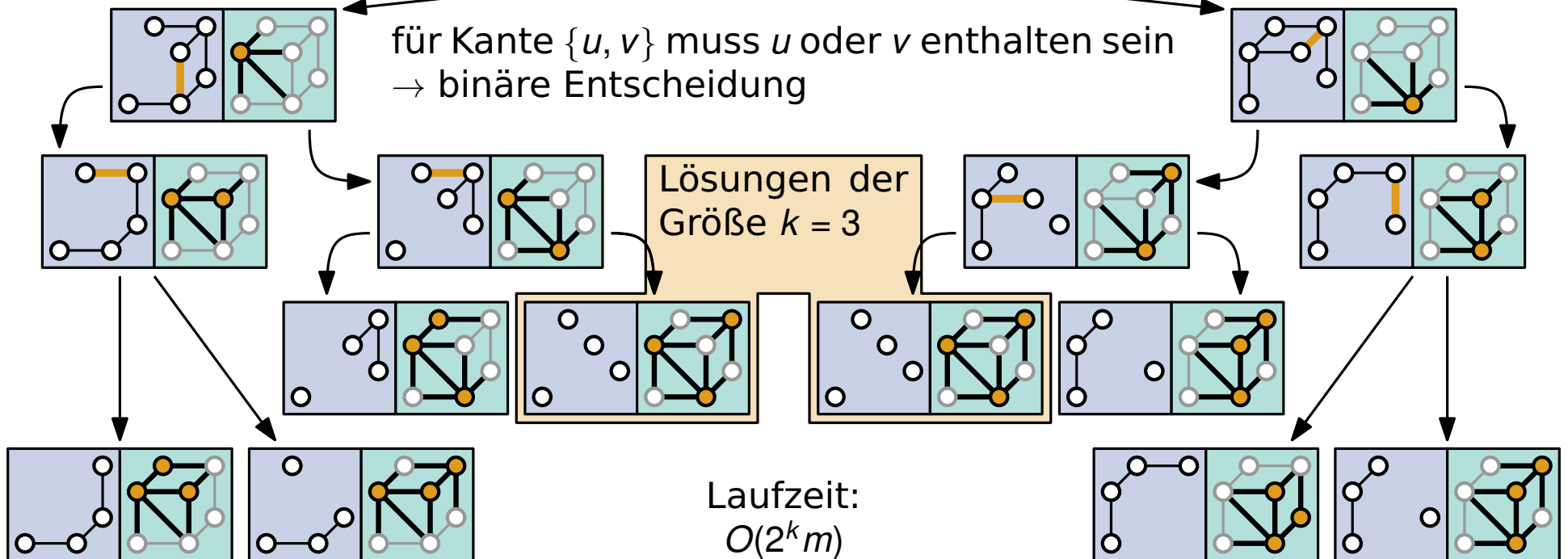
gewählte Knoten & überdeckte Kanten

Jede Kante muss noch überdeckt werden \rightarrow wähle eine beliebige



Gibt es ein Vertex Cover mit maximal $k = 3$ Knoten?

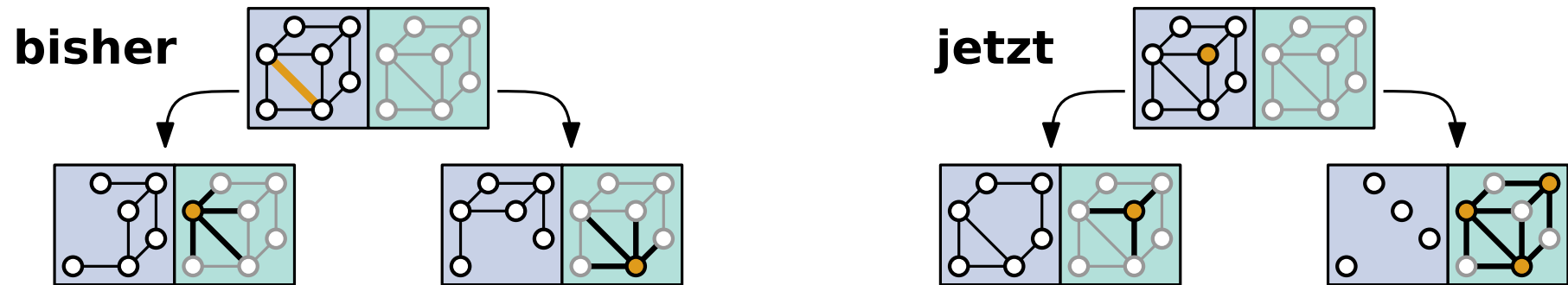
für Kante $\{u, v\}$ muss u oder v enthalten sein \rightarrow binäre Entscheidung



Wiederholung: Beschränkterer Suchbaum

Geht es schneller?

- Ziel: reduziere die Größe des Suchbaums (bisher: 2^k Blätter)
- Verzweigungsregel bisher: für Kante $\{u, v\}$ wähle entweder u oder v
- neue Verzweigungsregel: für Knoten v wähle entweder v oder $N(v)$



Wie viele Blätter hat der resultierende Baum abhängig von k ?

- wähle für v immer einen Knoten mit $\text{Grad} \geq 2$
- obere Schranke für #Blätter:
$$T(k) = \begin{cases} T(k-1) + T(k-2), & \text{für } k \geq 2 \\ 1, & \text{sonst} \end{cases}$$
- wir erhalten: $T(k) \leq 1,6181^k$
- Wie kommt man auf 1,6181? Geht es besser? \longrightarrow ~~nächste Vorlesung~~ ^{heute}

Verzweigungsvektoren

Allgemeine Verzweigung

- für Instanz (G, k) erzeuge ℓ Kind-Instanzen $(G_1, k_1), \dots, (G_\ell, k_\ell)$
- (G, k) lösbar \Leftrightarrow mindestens eine der Instanzen (G_i, k_i) lösbar
- sei $d_i = k - k_i$ (Parameter wird in Kind-Instanz G_i um d_i kleiner)
- Verzweigungsvektor: (d_1, \dots, d_ℓ)

Bisher gesehen

- Verzweigungsvektor $(1, 1) \Rightarrow$ Baumgröße 2^k
- Verzweigungsvektor $(1, 2) \Rightarrow$ Baumgröße $1,6181^k$

Wie beweist man die Baumgröße?

- rate Basis λ
- zeige $T(k) \leq c \cdot \lambda^k$ mittels Induktion (gleich mehr)

Wie rät man die Basis?

- finde Nullstelle eines Polynoms (gleich mehr)
- WolframAlpha hilft beim Lösen

Baumgröße

Beispiel: Verzweigungsvektor (1, 2)

- Rekurrenz:

$$T(k) = \begin{cases} T(k-1) + T(k-2), & \text{für } k \geq 2 \\ 1, & \text{sonst} \end{cases}$$
- Behauptung: $T(k) \leq 1,6181^k$

Beweis

- I.A.: $k < 2 \Rightarrow T(k) = 1 \leq 1,6181^k$
- I.S.:

$$\begin{aligned} T(k) &= T(k-1) + T(k-2) \\ &\leq 1,6181^{k-1} + 1,6181^{k-2} \\ &= 1,6181^{k-2} \cdot \underbrace{(1,6181 + 1)}_{\leq 1,6181^2 \approx 2,6182} \leq 1,6181^k \end{aligned}$$

Basis raten

Beispiel: Verzweigungsvektor (1, 2)

- Rekurrenz:

$$T(k) = \begin{cases} T(k-1) + T(k-2), & \text{für } k \geq 2 \\ 1, & \text{sonst} \end{cases}$$

- finde möglichst kleines $\lambda > 0$, sodass: $T(k) \leq c \cdot \lambda^k$

- wähle λ , sodass: $c \cdot \lambda^k \geq c \cdot \lambda^{k-1} + c \cdot \lambda^{k-2}$

(dann gilt: $T(k) = T(k-1) + T(k-2) \leq c \cdot \lambda^{k-1} + c \cdot \lambda^{k-2} \leq c \cdot \lambda^k$)

- löse: $\lambda^2 - \lambda - 1 = 0$

- „geratene“ Lösung: $\lambda = \frac{1+\sqrt{5}}{2} \leq 1,6181$

Allgemein

- Verzweigungsvektor: (d_1, \dots, d_ℓ)

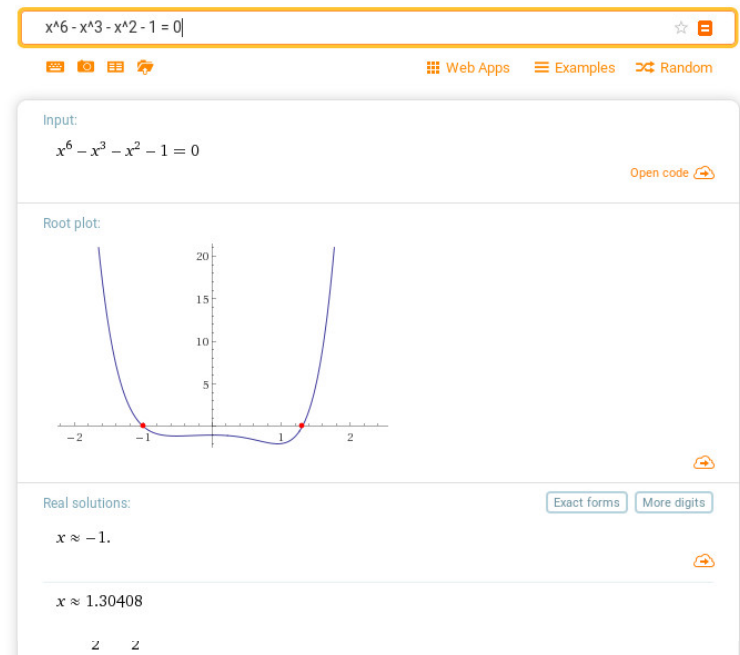
- $d = \text{Maximum der } d_i$

- löse: $0 = \lambda^d - \lambda^{d-d_1} - \dots - \lambda^{d-d_\ell}$

- Beispiel: Verzweigungsvektor (3, 4, 6)

- löse: $0 = \lambda^6 - \lambda^3 - \lambda^2 - 1$

 **WolframAlpha** computational knowledge engine.



Zurück zu VERTEX COVER

Verzweigungsvektor	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 3)	(3, 3, 6)	(3, 4, 6)
Basis b	2	1,618	1,466	1,381	1,325	1,325	1,26	1,342	1,305

Bisher gesehen

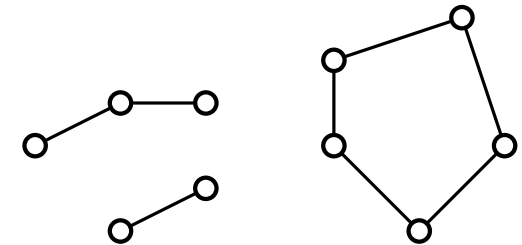
- für eine Kante uv , wähle u oder $v \rightarrow$ Verzweigungsvektor (1, 1)
- für Knoten v mit $\deg(v) \geq 2$, wähle v oder $N(v) \rightarrow$ Vektor (1, 2)

Warum gibt es immer einen Knoten v mit $\deg(v) \geq 2$?

Antwort: $\deg(v) = 1$ für alle $v \in V \Rightarrow G$ ist Matching und trivial lösbar

Geht es besser?

- gibt es auch immer ein $v \in V$ mit $\deg(v) \geq 3$?
- falls nicht: G besteht aus Pfaden und Kreisen
- $\deg(v) \leq 2$ für alle $v \in V \Rightarrow$ in poly-Zeit lösbar
- also: für Knoten v mit $\deg(v) \geq 3$, wähle v oder $N(v) \rightarrow$ Vektor (1, 3)

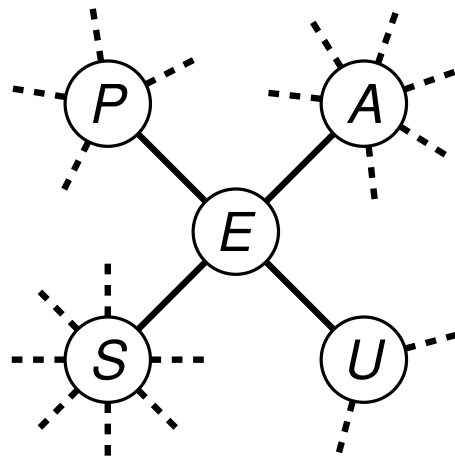


Geht es noch besser?

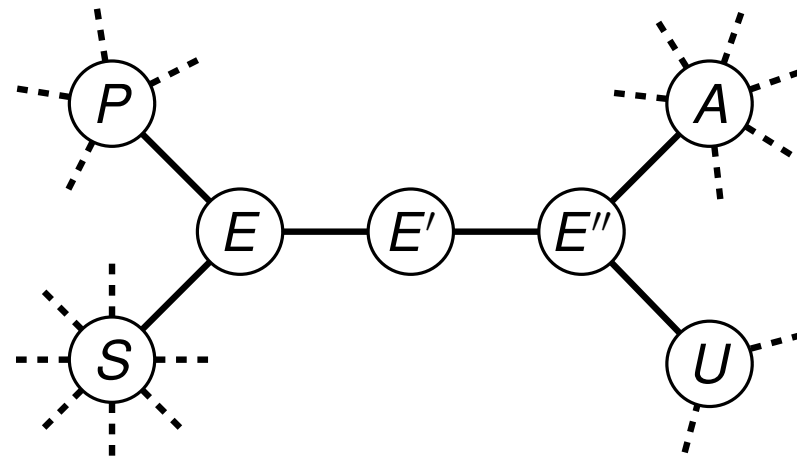
- VERTEX COVER ist NP-schwer für Graphen mit Maximalgrad 3
- Verzweigungsvektor (1, 4) erhält man nicht genauso leicht

Was passiert hier?

Was erreicht man mit der folgenden Konstruktion?



G



G'

Antwort

- G hat VC der Größe $k \Leftrightarrow G'$ hat VC der Größe $k + 1$
- Grad von E wurde von 4 auf 3 reduziert
- lässt sich auf höhere Grade verallgemeinern
- man kann so also zeigen, dass VERTEX COVER für Graphen mit Maximalgrad 3 NP-schwer ist

Lieber eliminieren als ignorieren

Knoten mit kleinem Grad

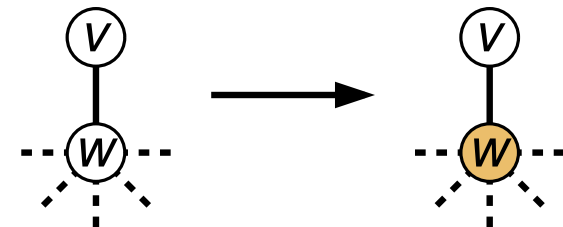
- eben: ignoriere Knoten v wenn $\deg(v) = 1$ bzw. $\deg(v) = 2$
- jetzt: eliminiere solche Knoten

Warum sollte das besser sein?

- Instanz verkleinern ist immer gut
- schränkt im weiteren Verlauf die Klasse der möglichen Instanzen ein
- hilft bei der Argumentation für spätere Fälle (für $v \in V$ gilt $\deg(v) \geq 3$)

Regel 1: es gibt $v \in V$ mit $\deg(v) = 1$

- es ist nie sinnvoll v zu wählen
- wähle also Nachbarn w von v



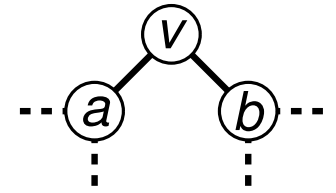
Beachte

- das ist nicht wirklich eine Verzweigungsregel, sondern eine Reduktionsregel (bzw. Verzweigung mit Vektor (1))
- verschlechtert die Basis in der Laufzeit aber sicher nicht

Knoten mit Grad 2

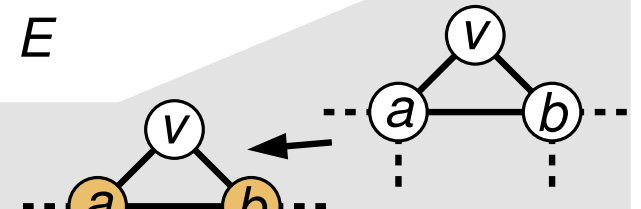
Regel 2: es gibt $v \in V$ mit $\deg(v) = 2$ (sei $N(v) = \{a, b\}$)

- falls a gewählt wird, macht es keinen Sinn v zu wählen
- wähle entweder a und b oder keinen der beiden
- keinen der beiden ist problematisch, wenn $ab \in E$



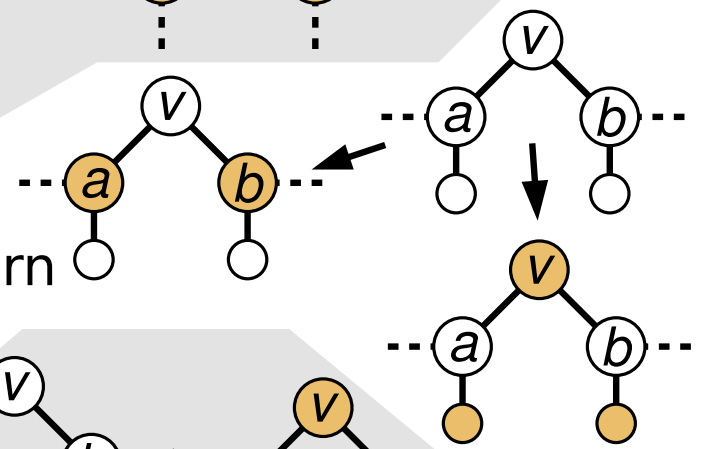
Fall 2.1: $ab \in E$

- zwei der drei Knoten müssen gewählt werden
- es ist immer besser a und b zu wählen
- keine Verzweigung nötig



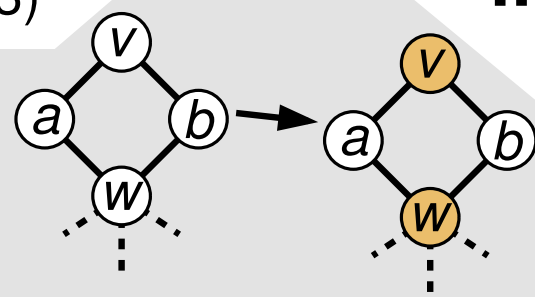
Fall 2.2: $|N(a) \cup N(b)| \geq 3$

- Bsp: a und b haben je einen anderen Nachbarn
- wähle $\{a, b\}$ oder $N(a) \cup N(b) \rightarrow$ Vektor $(2, 3)$



Fall 2.3: $N(a) \cup N(b) = \{v, w\}$

- es ist immer besser v und w zu wählen
- keine Verzweigung nötig

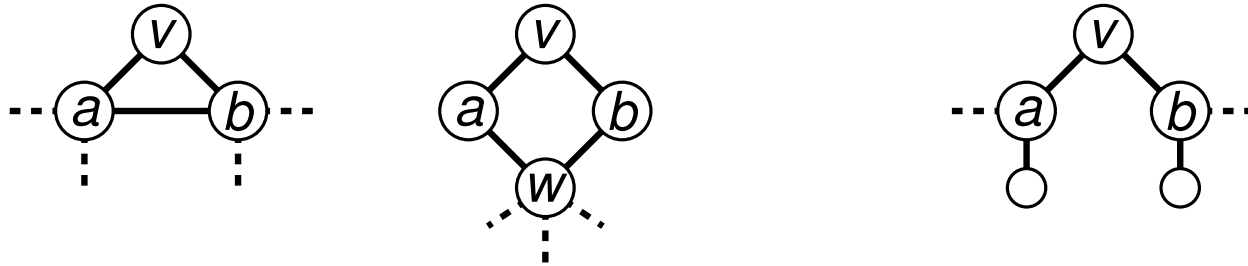


Verzweigungsvektor	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 3)	(3, 3, 6)	(3, 4, 6)
Basis b	2	1,618	1,466	1,381	1,325	1,325	1,26	1,342	1,305

Knoten mit Grad 2

Grobe Strategie bei Knoten mit Grad 2

- Fallunterscheidung, wie „unabhängig“ die Nachbarschaft von v ist
- Fall 2.1: Nachbarn sind direkt verbunden
- Fall 2.3: Nachbarn teilen sich Nachbarn
- Fall 2.2: halbwegs unabhängige Nachbarschaften



- unabhängige Nachbarschaften \Rightarrow große Vereinigungen von Nachbarschaften \Rightarrow gute Verzweigungsvektoren
- abhängige Nachbarschaften \Rightarrow wegdiskutierbar (mittels Reduktion)

Ziel im Folgenden

- ähnliches Vorgehen für $\deg(v) = 3$
- nutze, dass $\deg(u) \geq 3$ für alle $u \in V$

Verzweigungsvektor	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 3)	(3, 3, 6)	(3, 4, 6)
Basis b	2	1,618	1,466	1,381	1,325	1,325	1,26	1,342	1,305

Knoten mit Grad 3

Regel 3: es gibt $v \in V$ mit $\deg(v) = 3$ (sei $N(v) = \{a, b, c\}$)

Fall 3.1: paarweise disjunkte Nachbarschaften $N(a), N(b), N(c), N(v)$

- Möglichkeit 1: wähle a nicht

⇒ $N(a)$ muss gewählt werden

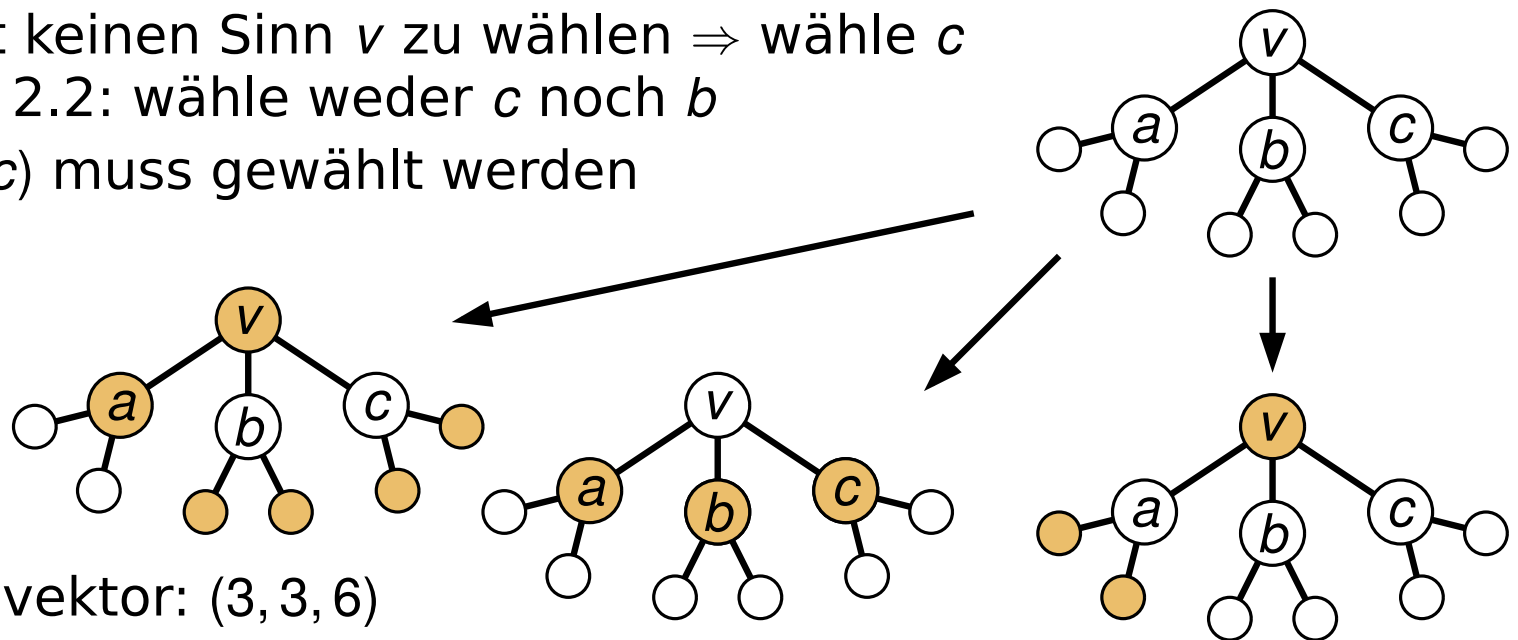
- Möglichkeit 2: wähle a

- Möglichkeit 2.1: wähle zusätzlich b

⇒ es macht keinen Sinn v zu wählen ⇒ wähle c

- Möglichkeit 2.2: wähle weder c noch b

⇒ $N(b) \cup N(c)$ muss gewählt werden



- Verzweigungsvektor: $(3, 3, 6)$

Verzweigungsvektor	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 3)	(3, 3, 6)	(3, 4, 6)
Basis b	2	1, 618	1, 466	1, 381	1, 325	1, 325	1, 26	1, 342	1, 305

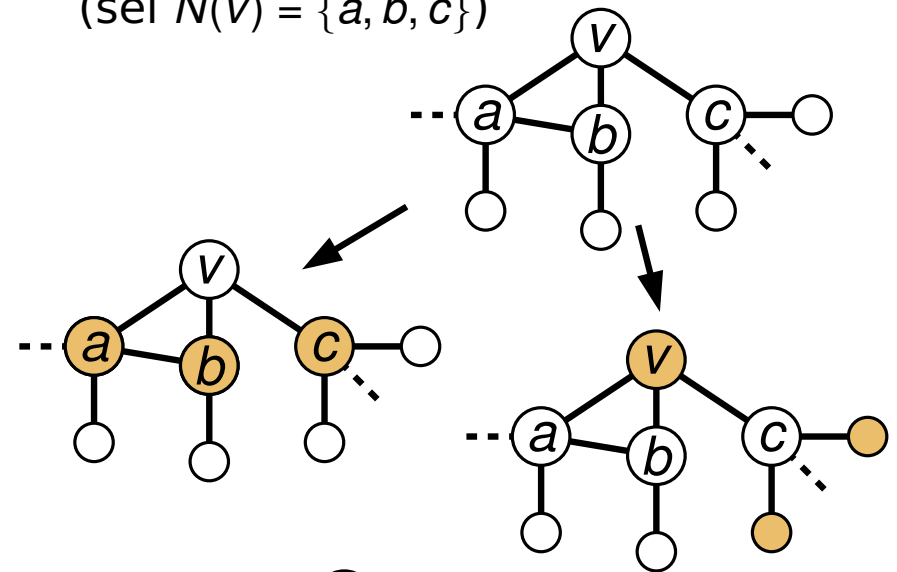
Knoten mit Grad 3

Regel 3: es gibt $v \in V$ mit $\deg(v) = 3$

Fall 3.2: $ab \in E$

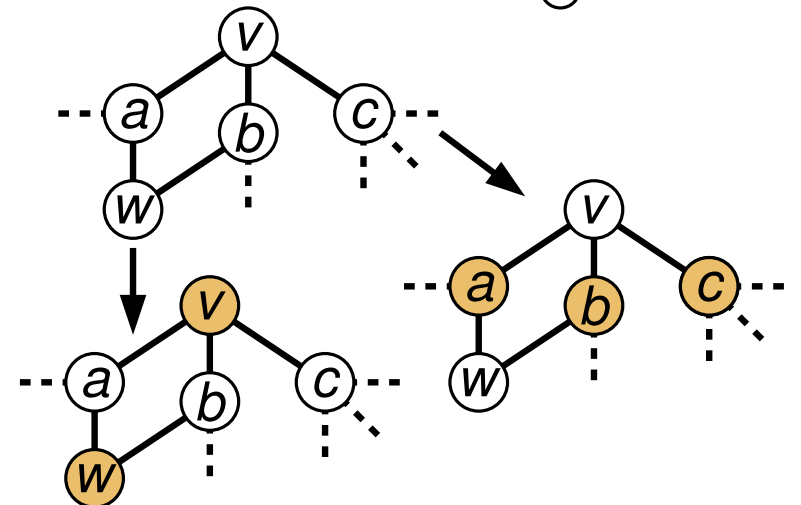
- Möglichkeit 1: wähle c nicht
 $\Rightarrow N(c)$ muss gewählt werden
- Möglichkeit 2: wähle c
 \Rightarrow es macht keinen Sinn v zu wählen
 \Rightarrow wähle zusätzlich $\{a, b\}$
- Verzweigungsvektor: $(3, 3)$

(sei $N(v) = \{a, b, c\}$)



Fall 3.3: a und b haben gem. Nachbarn w

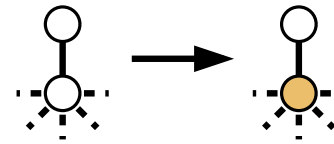
- Möglichkeit 1: wähle $\{v, w\}$
- Möglichkeit 2: wähle $\{a, b\}$
 \Rightarrow es macht keinen Sinn v zu wählen
 \Rightarrow wähle zusätzlich c
- Verzweigungsvektor: $(2, 3)$



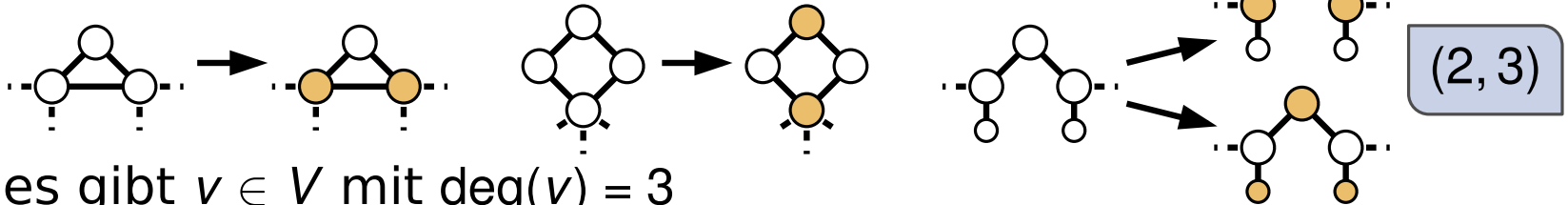
Verzweigungsvektor	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 3)	(3, 3, 6)	(3, 4, 6)
Basis b	2	1, 618	1, 466	1, 381	1, 325	1, 325	1, 26	1, 342	1, 305

Bisheriger Stand

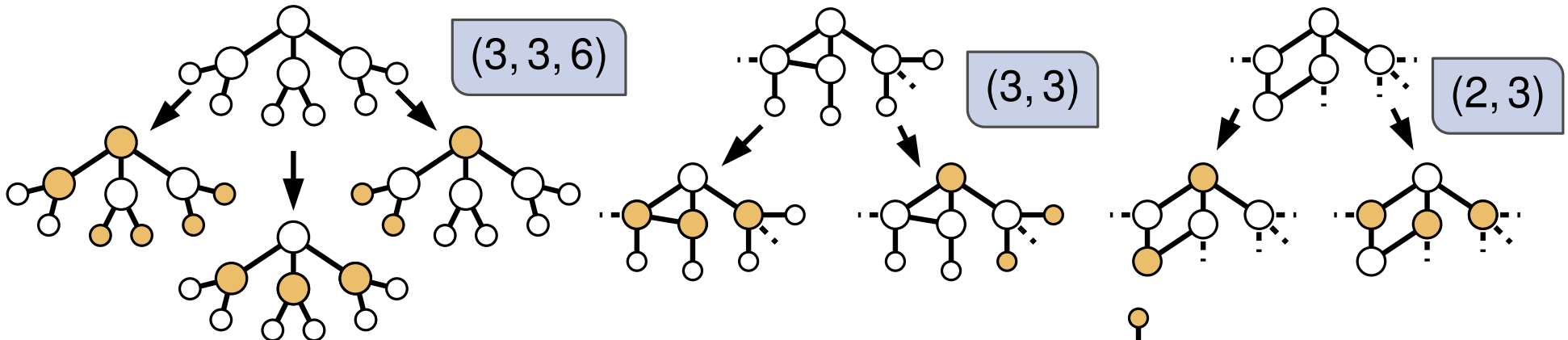
Regel 1: es gibt $v \in V$ mit $\deg(v) = 1$



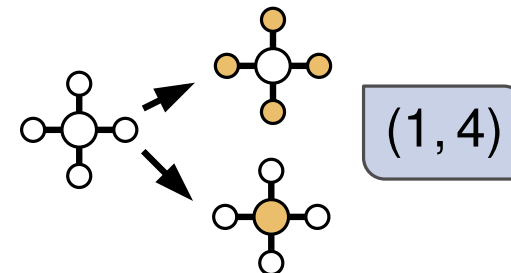
Regel 2: es gibt $v \in V$ mit $\deg(v) = 2$



Regel 3: es gibt $v \in V$ mit $\deg(v) = 3$



Regel 4: es gibt $v \in V$ mit $\deg(v) = 4$



⇒ **Gesamtlaufzeit:** $1,381^k \cdot n^{O(1)}$

Verzweigungsvektor	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 3)	(3, 3, 6)	(3, 4, 6)
Basis b	2	1,618	1,466	1,381	1,325	1,325	1,26	1,342	1,305

Geht es noch besser?

Regel 1: es gibt $v \in V$ mit $\deg(v) = 1$

Regel 2: es gibt $v \in V$ mit $\deg(v) = 2$

(2, 3)

Regel 3: es gibt $v \in V$ mit $\deg(v) = 3$

(3, 3, 6)

(3, 3)

(2, 3)

Regel 4: es gibt $v \in V$ mit $\deg(v) \geq 5$

(1, $\deg(v)$)

Regel 5: alle Knoten haben Grad 4

(1, 4)

Behauptung: die Gesamtlaufzeit ist $1,342^k \cdot n^{O(1)}$

Intuitive Begründung: Regel 5 kann in jedem Teilbaum nur ein Mal angewendet werden \Rightarrow Verzweigungsvektor (1, 4) kann man ignorieren

Verzweigungsvektor	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 3)	(3, 3, 6)	(3, 4, 6)
Basis b	2	1,618	1,466	1,381	1,325	1,325	1,26	1,342	1,305

Zusammenfassung & Literaturhinweis

Theorem

VERTEX COVER mit der Lösungsgröße k als Parameter kann in $1.342^k \cdot n^{O(1)}$ Zeit gelöst werden.

Beschränkte Suchbäume

- mit besseren Verzweigungsregeln kann man die Basis verkleinern
- Verzweigungsvektoren helfen bei der Analyse
- Nutzung des „schlechtesten“ aller benutzter Verzweigungsvektoren liefert potentiell ungenaue obere Schranke
(Beispiel: verzweige meistens mit (5, 5) und nur selten mit (1, 1))

Kleinere Basis

- Basis kann auf 1, 2738 gedrückt werden
- siehe auch: <http://fpt.wikidot.com/fpt-races>

Problem	$f(k)$	vertices in kernel	Reference/Comments
Vertex Cover	1.2738^k	$2k$	1
Connected Vertex Cover	2^k	no $k^{O(1)}$	26, randomized algorithm
Multiway Cut	2^k	not known	21, 38: $O(k^{s+1})$ -vertex kernel with s terminals
Directed Multiway Cut	$2^{O(k^3)}$	no $k^{O(1)}$	34
Almost-2-SAT (VC-PM)	2.3146^k	$O(k^6)$	37, 38, randomized kernel
Multicut	$2^{O(k^3)}$	no $k^{O(1)}$	22, 35
Pathwidth One Deletion Set	4.65^k	$O(k^2)$	28