Karlsruher Institut für Technologie

Maximum-Weight Independent Set mittels Baumzerlegung

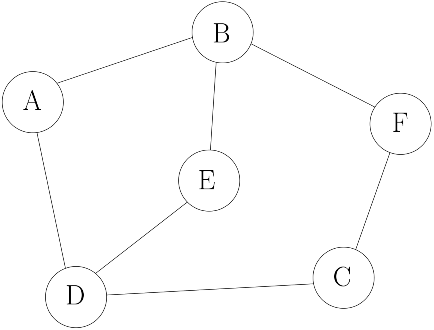
Marc Jenne

Institut für Theoretische Informatik - Proseminar NP-schwere Probleme

Gliederung

1. Das Problem MWIS
2. Dynamische Programmierung auf Bäumen
3. Baumweite und Baumzerlegung
4. Dynamische Programmierung auf Baumzerlegungen

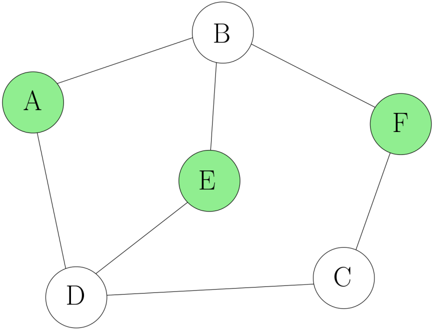
Maximum Independent Set



Definition Maximum Independent Set

- Gegeben: Graph $G = (V, E)$
- Gesucht: $\max_{S \subseteq V}\{|S| : \forall u, v \in S : \nexists \{u, v\} \in E\}$

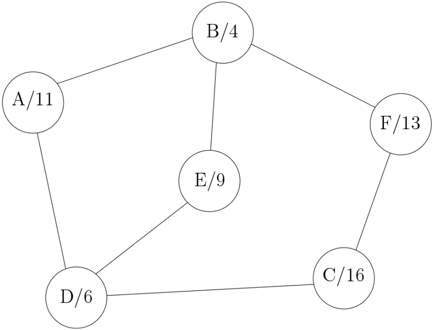
Maximum Independent Set



Definition Maximum Independent Set

- Gegeben: Graph $G = (V, E)$
- Gesucht: $\max_{S \subseteq V}\{|S| : \forall u, v \in S : \nexists\{u, v\} \in E\}$

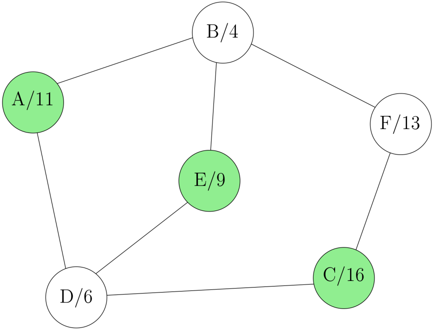
Maximum Weight Independent Set



Definition Maximum Weight Independent Set

- Gegeben: Graph $G = (V, E)$ mit Knotengewichtsfunktion $w : V \to \mathbb{N}_0$
- Gesucht: $\max_{S \subseteq V} \{ \sum_{s \in S} w(s) : \forall u, v \in S : \nexists \{u, v\} \in E \}$

Maximum Weight Independent Set

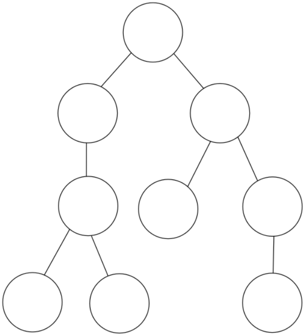


Definition Maximum Weight Independent Set

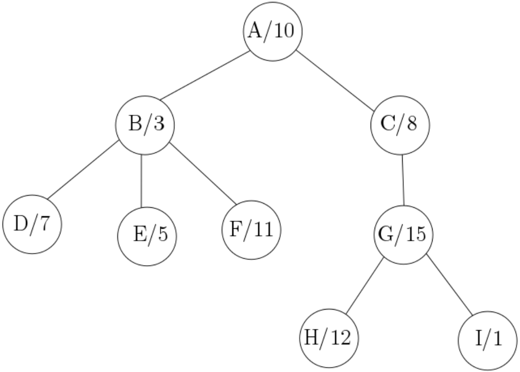
- Gegeben: Graph $G = (V, E)$ mit Knotengewichtsfunktion $w : V \to \mathbb{N}_0$
- Gesucht: $\max_{S \subseteq V}\{\sum_{s \in S} w(s) : \forall u, v \in S : \nexists \{u, v\} \in E\}$

MWIS auf Bäumen: DP

- funktioniert aufgrund der Eigenschaften von Bäumen
- gewurzelter Baum mit Orientierung: jeder Knoten ist Wurzel des Teilbaums darunter
- Problem in Teilprobleme teilen und unabhängig voneinander lösen
- zwei Möglichkeiten für jeden Knoten j:
  - j gehört nicht zum MWIS: Kinder von j können dazugehören
  - j gehört zum MWIS: Kinder von j können nicht dazugehören
- $f^{+}(j)$: |MWIS| in G_j: $j \in MWIS$
  $f^{-}(j)$: |MWIS| in G_j: $j \notin MWIS$

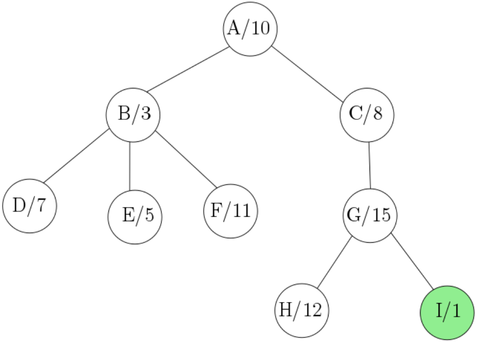


Beispiel



-	A	B	C	D	E	F	G	H	I
$f^+(v)$	-	-	-	-	-	-	-	-	-
$f^-(v)$	-	-	-	-	-	-	-	-	-

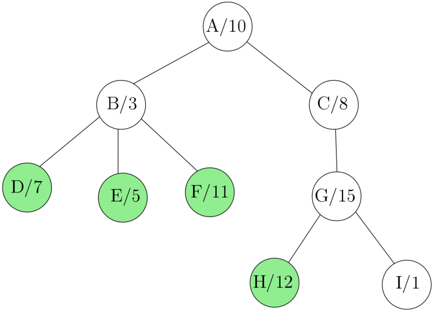
Beispiel



$f^+(I) = 1$
$f^-(I) = 0$

-	A	B	C	D	E	F	G	H	I
$f^+(v)$	-	-	-	-	-	-	-	-	1
$f^-(v)$	-	-	-	-	-	-	-	-	0

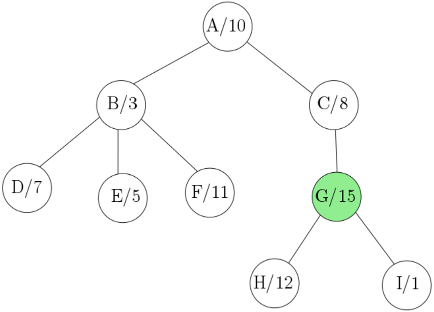
Beispiel



$$f^+(H) = 12$$
$$f^-(H) = 0$$
$$f^+(F) = 11$$
$$f^-(F) = 0$$
$$f^+(E) = 5$$
$$f^-(E) = 0$$
$$f^+(D) = 7$$
$$f^-(D) = 0$$

-	A	B	C	D	E	F	G	H	I
$f^+(v)$	-	-	-	7	5	11	-	12	1
$f^-(v)$	-	-	-	0	0	0	-	0	0

Beispiel

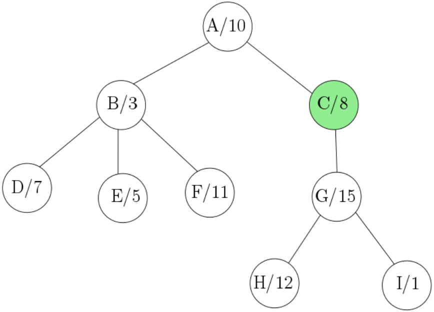


$f^+(G) = w(F) + f^-(H) + f^-(I) = 15 + 0 + 0 = 15$

$f^-(G) = \max\{f^+(H), f^-(H)\} + \max\{f^+(I), f^-(I)\} = 12 + 1 = 13$

-	A	B	C	D	E	F	G	H	I
$f^+(v)$	-	-	-	7	5	11	15	12	1
$f^-(v)$	-	-	-	0	0	0	13	0	0

Beispiel

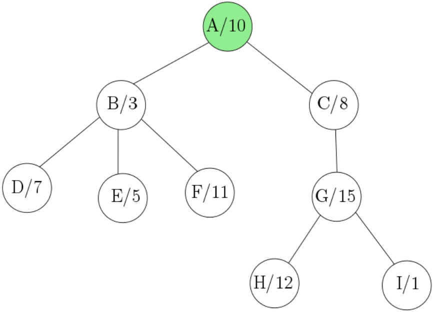


$f^{+}(C) = w(C) + f^{-}(G) =$
$8 + 13 = 21$

$f^{-}(C) = \max\{f^{+}(G), f^{-}(G)\} = 15$

-	A	B	C	D	E	F	G	H	I
$f^{+}(v)$	-	-	21	7	5	11	15	12	1
$f^{-}(v)$	-	-	15	0	0	0	13	0	0

Beispiel



$f^+(A) = w(A) + f^-(B) + f^-(C) = 10 + 23 + 15 = 48$

$f^-(A) = \max\{f^+(B), f^-(B)\} + \max\{f^+(C), f^-(C)\} = 23 + 21 = 44$

-	A	B	C	D	E	F	G	H	I
$f^+(v)$	48	3	21	7	5	11	15	12	1
$f^-(v)$	44	23	15	0	0	0	13	0	0

Laufzeit

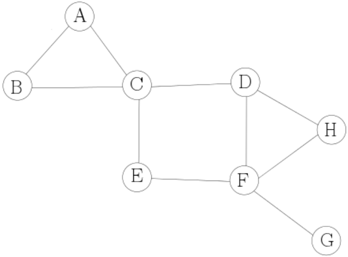
- jeder Knoten wird einmal betrachtet
- pro Betrachtung eine Abfrage und 2 Wertezuweisungen
- Laufzeit ist also $O(|V|)$

for $v \in V$ **do**
  if *v is a leaf* **then**
    $f^-(v) = 0$
    $f^+(v) = w(v)$
  else
    $f^+(v) = w(v) + \sum_{c \in C} f^-(c)$
    $f^-(v) = \sum_{c \in C} \max\{f^+(c), f^-(c)\}$
  end
end

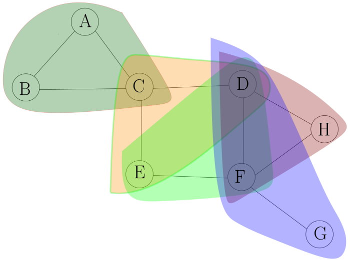
DP auch für weitere Graphen möglich?

- funktioniert wegen Baumeigenschaften
- lässt sich dieses Prinzip auf beliebige Graphen übertragen?
- es gibt Graphen, die "baumähnlich" sind

Baumzerlegung

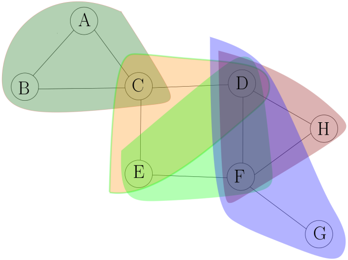


Baumzerlegung



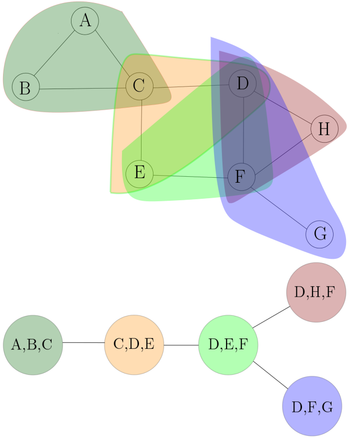
Baumzerlegung

- Node Coverage: $\bigcup_{t \in T} V_t = V$
- Edge Coverage:
  $\forall (u, v) \in E : \exists t \in T : u, v \in V_t$



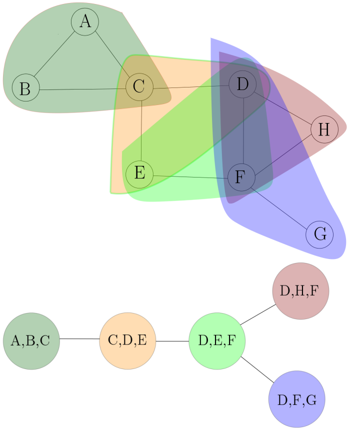
Baumzerlegung

- Node Coverage: $\bigcup_{t \in T} V_t = V$
- Edge Coverage: $\forall (u, v) \in E : \exists t \in T : u, v \in V_t$
- Coherence: $\forall t_1, t_2, t_3 \in T$, t_2 liegt auf Pfad zwischen t_1 und t_3 : $u \in V_{t_1}, V_{t_3} \Rightarrow u \in V_{t_2}$



Baumzerlegung

- Node Coverage: $\bigcup_{t \in T} V_t = V$
- Edge Coverage:
  $\forall (u,v) \in E : \exists t \in T : u,v \in V_t$
- Coherence: $\forall t_1, t_2, t_3 \in T$, t_2 liegt auf Pfad zwischen t_1 und t_3 :
  $u \in V_{t_1}, V_{t_3} \Rightarrow u \in V_{t_2}$



Definition Baumzerlegung

Eine Baumzerlegung ist ein geordnetes Paar $(T, \{V_t : t \in T\})$.

Eigenschaften einer Baumzerlegung

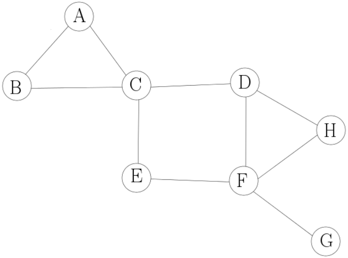
Karlsruher Institut für Technologie

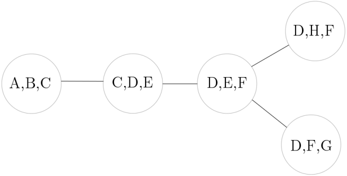
- Node Coverage und Edge Coverage: widerspiegeln der Zusammenhänge der Knoten in den Bags V_t
- Coherence: Spaltungen in T spiegeln sich in G wider
- Spalteigenschaften von Bäumen:
  - wenn eine Kante entfernt wird, zerfällt der Baum in 2 Teilbäume
  - wenn ein Knoten v gelöscht wird, zerfällt der Baum in $deg(v)$ Teilbäume

Spalteigenschaften durch Coherence

Löschen eines Knotens t aus T

Annahme: $T - t$ besteht aus den Komponenten $T_1, ..., T_d$. Dann haben die Teilgraphen $G_{T_1} - V_t, G_{T_2} - V_t, G_{T_d} - V_t$ keine gemeinsamen Knoten und sind durch keine Kante verbunden.

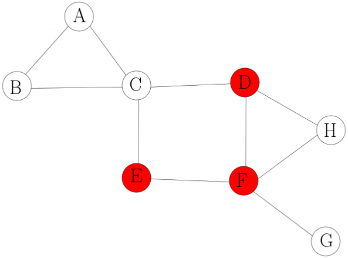


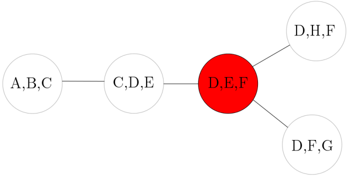


Spalteigenschaften durch Coherence

Löschen eines Knotens t aus T

Annahme: $T - t$ besteht aus den Komponenten $T_1, ..., T_d$. Dann haben die Teilgraphen $G_{T_1} - V_t, G_{T_2} - V_t, G_{T_d} - V_t$ keine gemeinsamen Knoten und sind durch keine Kante verbunden.

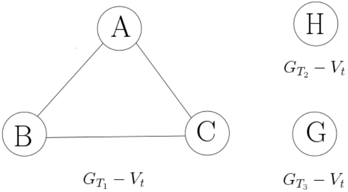


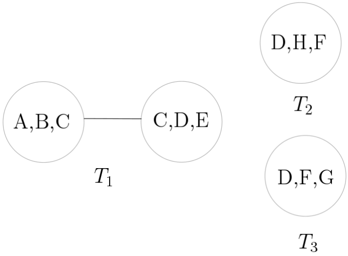


Spalteigenschaften durch Coherence

Löschen eines Knotens t aus T

Annahme: $T - t$ besteht aus den Komponenten $T_1, \ldots, T_d$. Dann haben die Teilgraphen $G_{T_1} - V_t, G_{T_2} - V_t, G_{T_d} - V_t$ keine gemeinsamen Knoten und sind durch keine Kante verbunden.

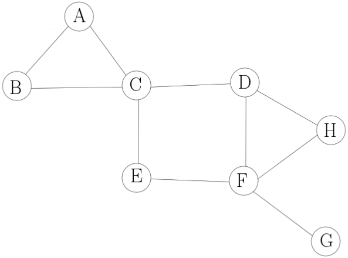


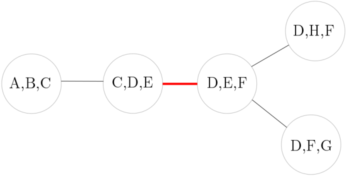


Spalteigenschaften durch Coherence

Löschen einer Kante {x,y} aus T

Seien X und Y die Komponenten von T nach Löschen der Kante (x, y). Dann teilt das Löschen der Menge $V_x \cap V_y$ aus V den Graphen G in zwei Teilgraphen $G_X - (V_x \cap V_y)$ und $G_Y - (V_x \cap V_y)$. Diese Teilgraphen haben keinen gemeinsamen Knoten und sind durch keine Kante verbunden.

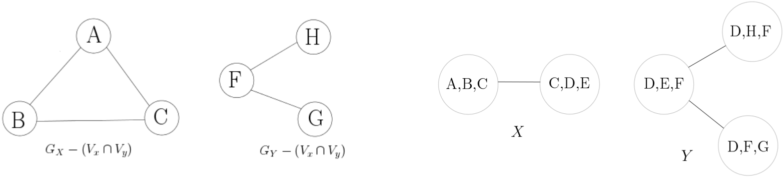




Spalteigenschaften durch Coherence

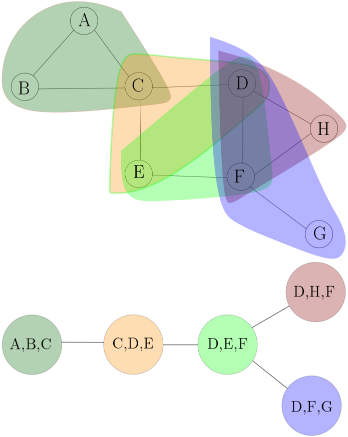
Löschen einer Kante {x,y} aus T

Seien X und Y die Komponenten von T nach Löschen der Kante (x, y). Dann teilt das Löschen der Menge $V_x \cap V_y$ aus V den Graphen G in zwei Teilgraphen $G_X - (V_x \cap V_y)$ und $G_Y - (V_x \cap V_y)$. Diese Teilgraphen haben keinen gemeinsamen Knoten und sind durch keine Kante verbunden.



Baumweite

- Motivation: "baumähnlich" formalisieren
- Baumweite einer Zerlegung: $tw(T, \{V_t\}) = \max_t |V_t| - 1$
- Baumweite eines Graphen: kleinste Baumweite unter allen möglichen Zerlegungen



Zurück zum Problem MWIS

- Ziel: DP auf beliebige Graphen anwenden
- mithilfe von Baumzerlegungen: Anwendung von DP in leicht abgewandelter Form
- Problem MWIS ist fixed parameter tractable
  - Laufzeit exponentiell in der Baumweite

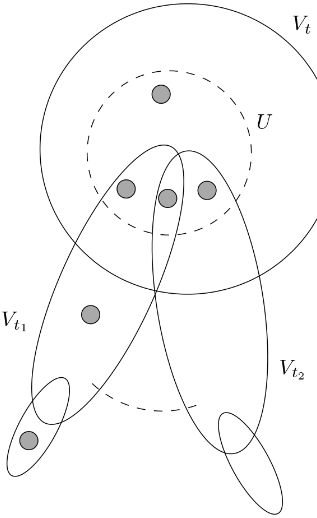
DP auf Baumzerlegung

- im orientierten Baum: man betrachtet bottom-up alle Bags V_t
- bei jeder Bag gilt: das optimale MWIS schneidet die Menge V_t in einer Teilmenge U
- man betrachtet alle möglichen Teilmengen: Brute-Force
- Baumweite w: es gibt also 2^{w+1} Teilmengen

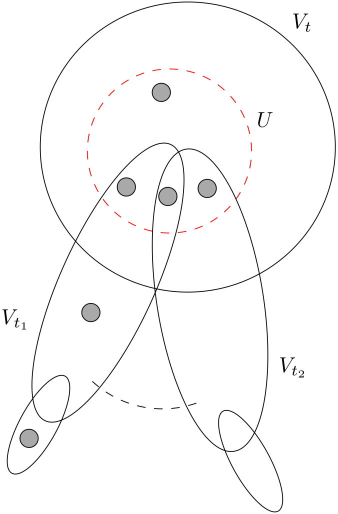
Algorithmus - Die Teilprobleme

- zu jedem Teilbaum T_t definieren wir eine Menge von Teilproblemen: eines für jede Teilmenge $U \subseteq V$
- $f_t(U)$ bezeichnet das maximale Gewicht eines IS S in G_t, unter der Voraussetzung dass $S \cap V_t = U$
- wenn U kein IS ist, ist $f_t(U)$ nicht definiert
- zu jeder Bag gehören maximal 2^{w+1} Teilprobleme

Zusammenhang U mit Kindern

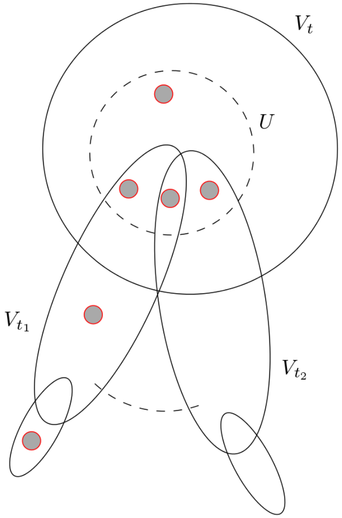


Zusammenhang U mit Kindern



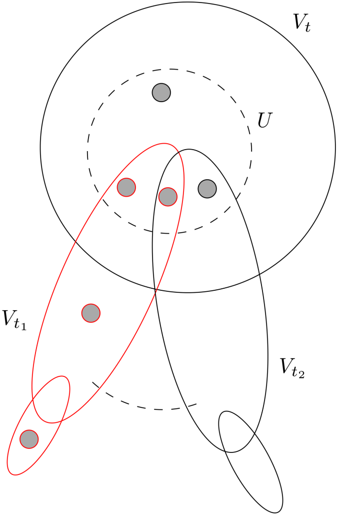
- Betrachtung einer Teilmenge $U \subseteq V_t$

Zusammenhang U mit Kindern



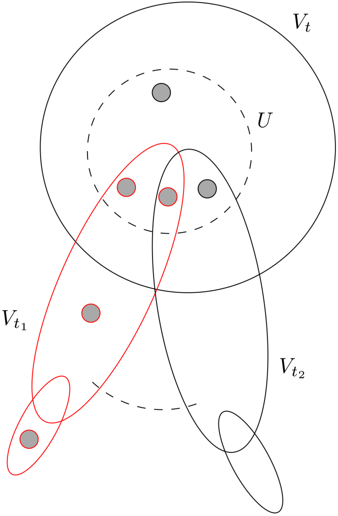
- Betrachtung einer Teilmenge $U \subseteq V_t$
- S sei MWIS von G_t unter der Bedingung: $S \cap V_t = U$

Zusammenhang U mit Kindern



- Betrachtung einer Teilmenge $U \subseteq V_t$
- S sei MWIS von G_t unter der Bedingung: $S \cap V_t = U$
- $S_i = S \cap G_{t_i}$
- S_i ist ein MWIS von G_{t_i} unter der Bedingung: $S_i \cap V_t = U \cap V_{t_i}$

Zusammenhang U mit Kindern



- Betrachtung einer Teilmenge $U \subseteq V_t$
- S sei MWIS von G_t unter der Bedingung: $S \cap V_t = U$
- $S_i = S \cap G_{t_i}$
- S_i ist ein MWIS von G_{t_i} unter der Bedingung: $S_i \cap V_t = U \cap V_{t_i}$

Lösung der Teilprobleme

Satz

S sei ein MWIS von G_t unter der Bedingung: $S \cap V_t = U$. Dann gilt: S_i ist ein MWIS von G_{t_i} unter der Bedingung: $S_i \cap V_t = U \cap V_{t_i}$

- Annahme: $\exists$ IS S_i' mit $S_i' \cap V_t = U \cap V_{t_i}$ und $w(S_i') > w(S_i)$
- betrachte Set $S' = (S - S_i) \cup S_i'$
- Fall 1: S' ist IS $\rightarrow$ widerspricht Wahl von S als MWIS
- Fall 2: S' ist kein IS
  - $e = \{u, v\}$ mit beiden Enden in S': u und v können weder beide in S noch beide in S_i' sein
  - $u \in S - S_i'$ und $v \in S_i' - S \Rightarrow u \notin G_{t_i}, v \in G_{t_i} - (V_t \cap V_{t_i})$
  - Schnitteigenschaft: es kann keine Kante $\{u, v\}$ geben

Rekurrenz zum Berechnen der Teilprobleme

- Fall 1: t ist ein Blatt
  - $f_t(U) = w(U)$ für jedes IS $U \subseteq V_t$
- Fall 2: t ist ein innerer Knoten
  - t hat Kinder $t_1, ..., t_d$
  - von diesen Kindern ist $f_{t_i}(U_i)$ für jedes IS $U_i \subseteq V_{t_i}$ bereits berechnet worden
  - $f_t(U) = w(U) + \sum_{i=1}^{d} \max\{f_{t_i}(U_i) - w(U_i \cap U) :$ $U_i \cap V_t = U \cap V_{t_i}$ und $U_i \subseteq V_{t_i}$ ist unabhängig$\}$

Fakultät für Informatik
Institut für Theoretische Informatik

Algorithmus - Pseudocode

Root T at an arbitrary node r

for $t \in T$ *in post-order* **do**
 if t *is a leaf* **then**
  for *each IS* $U \subseteq V_t$ **do**
   $f_t(U) = w(U)$
  end
 else
  for *each IS* $U \subseteq V_t$ **do**
   $f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
   $U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent
  end
 end
end

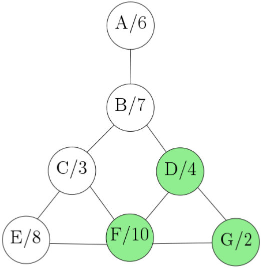
return $\max\{f_r(U) : U \subseteq V_r$ is independent$\}$

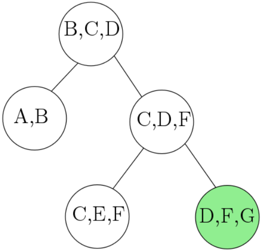
Marc Jenne:
Maximum-Weight Independent Set mittels Baumzerlegung

Fakultät für Informatik
Institut für Theoretische Informatik

Beispiel

Knoten ist ein Blatt



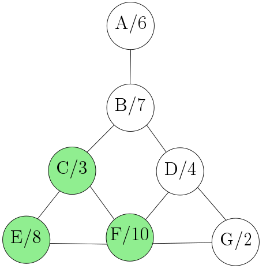


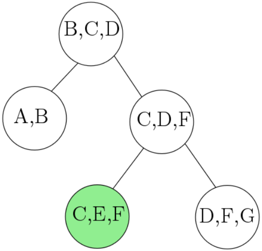
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

U	$\emptyset$	D	F	G
$f_t(U)$	0	4	10	2

Beispiel

Knoten ist ein Blatt



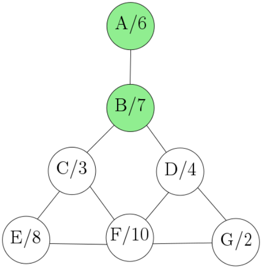


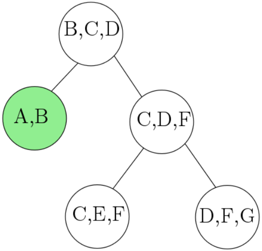
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

U	$\emptyset$	D	F	G	$\emptyset$	C	E	F
$f_t(U)$	0	4	10	2	0	3	8	10

Beispiel

Knoten ist ein Blatt



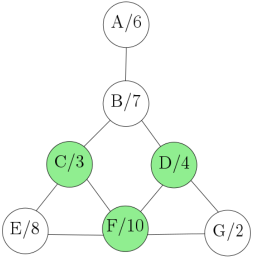


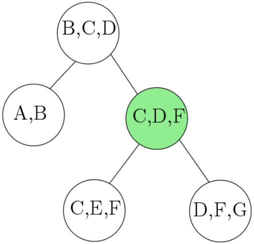
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

Beispiel

Innerer Knoten





$$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U) :$$
$$U_i \cap V_t = U \cap V_{t_i} \text{ and } U_i \subseteq V_{t_i} \text{ is independent}$$

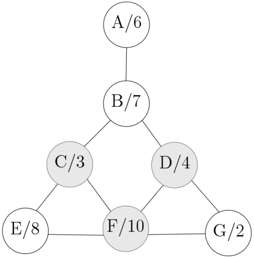
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

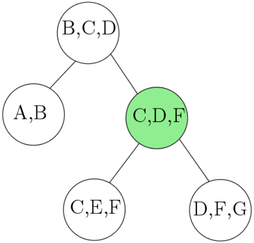
U	$\emptyset$	C	D	F	C,D
$f_t(U)$					

 Marc Jenne:

Beispiel

$U = \{\emptyset\}$
$U_i \subseteq \{D, F, G\} : \{\emptyset\}, \{G\}$
$U_i \subseteq \{C, E, F\} : \{\emptyset\}, \{E\}$
$f_t(U) = 0 + 2 + 8 = 10$





$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

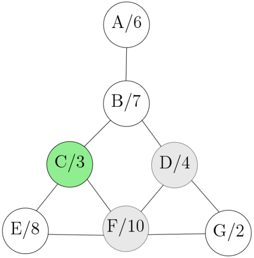
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

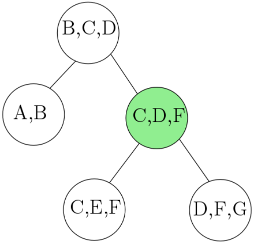
U	$\emptyset$	C	D	F	C,D
$f_t(U)$	10				

Beispiel

KIT Karlsruher Institut für Technologie

$U = \{C\}$
$U_i \subseteq \{D, F, G\} : \{\emptyset\}, \{G\}$
$U_i \subseteq \{C, E, F\} : \{C\}$
$f_t(U) = 3 + 2 + 0 = 5$





$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

U	∅	D	F	G	∅	C	E	F	∅	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

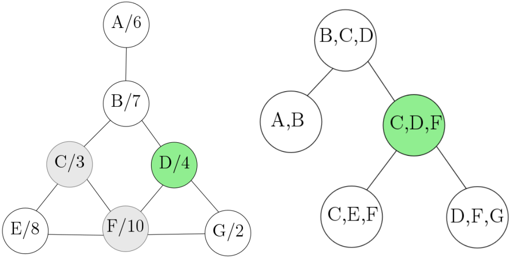
U	∅	C	D	F	C,D
$f_t(U)$	10	5			

Marc Jenne:
Maximum-Weight Independent Set mittels Baumzerlegung

Fakultät für Informatik
Institut für Theoretische Informatik

Beispiel

$U = \{D\}$
$U_i \subseteq \{D, F, G\} : \{D\}$
$U_i \subseteq \{C, E, F\} : \{\emptyset\}, \{E\}$
$f_t(U) = 4 + 0 + 8 = 12$



$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

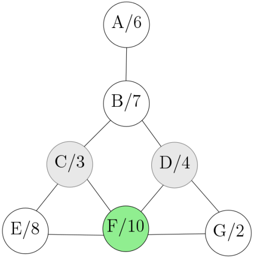
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

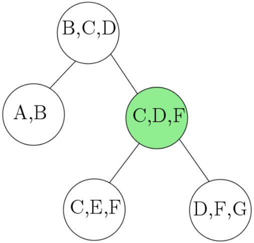
U	$\emptyset$	C	D	F	C,D
$f_t(U)$	10	5	12		

Fakultät für Informatik
Institut für Theoretische Informatik

Beispiel

$U = \{F\}$
$U_i \subseteq \{D, F, G\} : \{F\}$
$U_i \subseteq \{C, E, F\} : \{F\}$
$f_t(U) = 10 + 0 + 0 = 10$





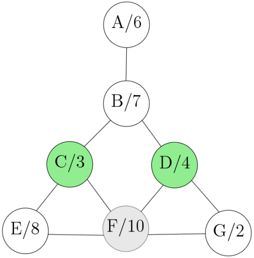
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

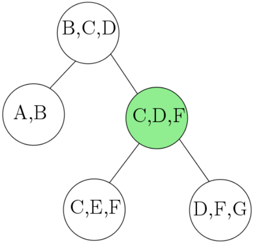
U	∅	D	F	G	∅	C	E	F	∅	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	∅	C	D	F	C,D
$f_t(U)$	10	5	12	10	

Beispiel

$U = \{C, D\}$
$U_i \subseteq \{D, F, G\} : \{D\}$
$U_i \subseteq \{C, E, F\} : \{C\}$
$f_t(U) = 7 + 0 + 0 = 7$





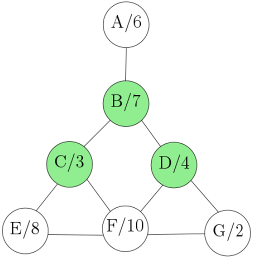
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

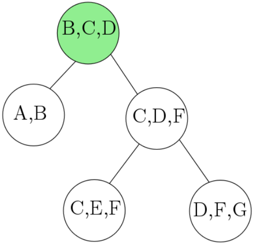
U	∅	D	F	G	∅	C	E	F	∅	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	∅	C	D	F	C,D
$f_t(U)$	10	5	12	10	7

Beispiel

innerer Knoten





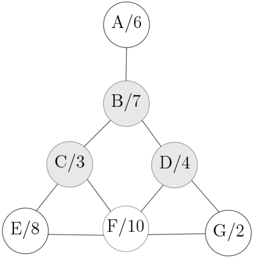
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

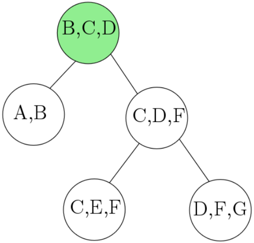
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7					

Beispiel

$U = \{\emptyset\}$
$U_i \in \{A, B\} : \{\emptyset\}, \{A\}$
$U_i \in \{C, D, F\} : \{\emptyset\}, \{F\}$
$f_t(U) = 0 + 6 + 10 = 16$





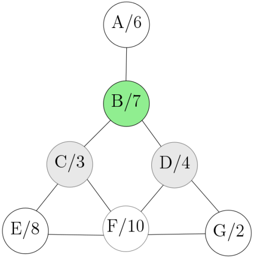
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

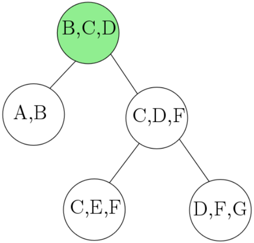
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16				

Beispiel

$U = \{B\}$
$U_i \in \{A, B\} : \{B\}$
$U_i \in \{C, D, F\} : \{\emptyset\}, \{F\}$
$f_t(U) = 7 + 0 + 10 = 17$





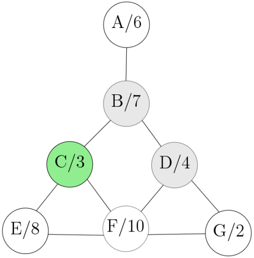
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U) :$
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

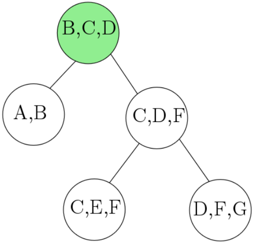
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16	17			

Beispiel

$U = \{C\}$
$U_i \in \{A, B\} : \{\emptyset\}, \{A\}$
$U_i \in \{C, D, F\} : \{C\}$
$f_t(U) = 3 + 6 + 2 = 11$





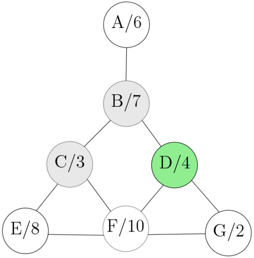
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

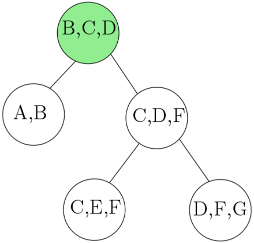
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16	17	11		

Beispiel

$U = \{D\}$
$U_i \in \{A, B\} : \{\emptyset\}, \{A\}$
$U_i \in \{C, D, F\} : \{D\}$
$f_t(U) = 4 + 6 + 8 = 18$





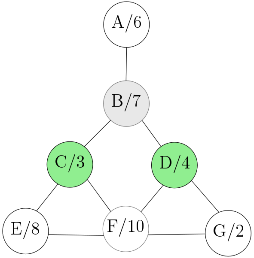
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

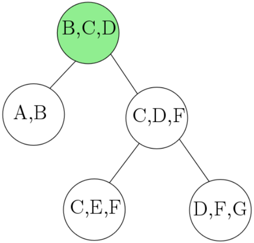
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16	17	11	18	

Beispiel

$U = \{C, D\}$
$U_i \in \{A, B\} : \{\emptyset\}, \{A\}$
$U_i \in \{C, D, F\} : \{C, D\}$
$f_t(U) = 7 + 6 + 0 = 13$





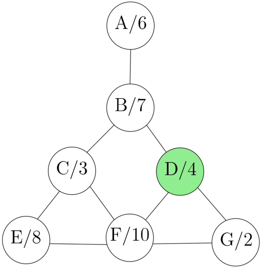
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U) :$
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

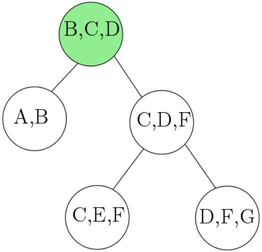
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16	17	11	18	13

Beispiel - Backtracking

Algorithmus liefert nur das Gewicht des MWIS. Mit Backtracking sucht man nun das MWIS





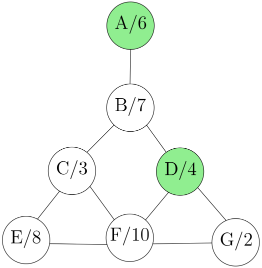
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

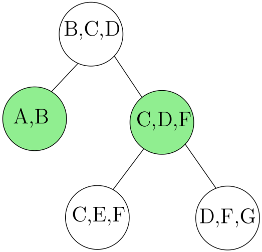
U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16	17	11	18	13

Beispiel - Backtracking

Algorithmus liefert nur das Gewicht des MWIS. Mit Backtracking sucht man nun das MWIS





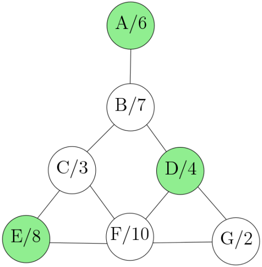
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

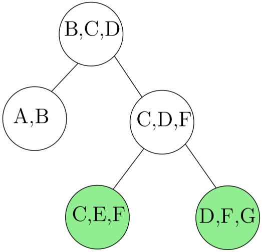
U	∅	D	F	G	∅	C	E	F	∅	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	∅	C	D	F	C,D	∅	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16	17	11	18	13

Beispiel - Backtracking

Algorithmus liefert nur das Gewicht des MWIS. Mit Backtracking sucht man nun das MWIS





$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent

U	$\emptyset$	D	F	G	$\emptyset$	C	E	F	$\emptyset$	A	B
$f_t(U)$	0	4	10	2	0	3	8	10	0	6	7

U	$\emptyset$	C	D	F	C,D	$\emptyset$	B	C	D	C,D
$f_t(U)$	10	5	12	10	7	16	17	11	18	13

Algorithmus - Laufzeitanalyse

- pro Knoten t : 2^{w+1} Teilmengen U
- pro Kind von t : 2^{w+1} Teilmengen U_i
- Prüfen, ob $U_i \cap V_t = U \cap V_{t_i}$: $O(w)$
- insgesamt: $O(2^{w+1} \cdot 2^{w+1} \cdot w \cdot d) = O(4^{w+1} \cdot w \cdot d)$ für Knoten t
- $\sum d = n$, da jeder Knoten genau einmal Kind ist

for *each IS* $U \subseteq V_t$ **do**
$f_t(U) = w(U) + \sum_{i=1}^{d} \max f_{t_i}(U_i) - w(U_i \cap U)$:
$U_i \cap V_t = U \cap V_{t_i}$ and $U_i \subseteq V_{t_i}$ is independent
end

Algorithmus - Nachteile?

- Laufzeit immer noch exponentiell - wo liegt der Gewinn?
  - Auslagerung der Laufzeit auf einen Parameter, die Baumweite
  - Exponentielles Wachstum nur noch davon abhängig: fixed parameter tractable
  - Baumweite kann sehr gering sein trotz vieler Knoten
- außerdem: Baumweite rausfinden ebenfalls NP-schwer
- Approximation schnell möglich

Zusammenfassung

- Lösen von MWIS auf Bäumen in linearer Zeit möglich
- Ausnutzung der Baumeigenschaften und dynamischer Programmierung
- mithilfe von Baumzerlegung auch anwendbar auf andere Graphen
- Lösen von MWIS auf Graphen mit beschränkter Baumweite effizient möglich