

Kolloquium zur Bachelorarbeit

Untersuchung von Knotenreduktionsregeln beim
Knotenüberdeckungsproblem

Referent: Benedikt Lüken-Winkels

Prüfer: Prof. Dr. Henning Fernau
Prof. Dr. Stefan Näher

07. März 2018

Universität Trier

Knotenüberdeckungsproblem

Knotenüberdeckungsproblem - Definition

Knotenüberdeckung

EINGABE: Graph $G = (V, E)$, natürliche Zahl $k \leq |V|$

AUSGABE: $S \subseteq V$ mit $|S| \leq k$, sodass jede Kante aus E einen Endpunkt in S hat.

- *NP – vollständig*
- Naive Algorithmen haben eine Laufzeit von $O(n^k)$
- Suchbaumalgorithmen laufen in $O(2^k \cdot (m + n))$

Knotenüberdeckung

EINGABE: Graph $G = (V, E)$, natürliche Zahl $k \leq |V|$

AUSGABE: $S \subseteq V$ mit $|S| \leq k$, sodass jede Kante aus E einen Endpunkt in S hat.

- *NP – vollständig*
- Naive Algorithmen haben eine Laufzeit von $O(n^k)$
- Suchbaumalgorithmen laufen in $O(2^k \cdot (m + n))$

Knotenüberdeckungsproblem - Definition

Knotenüberdeckung

EINGABE: Graph $G = (V, E)$, natürliche Zahl $k \leq |V|$

AUSGABE: $S \subseteq V$ mit $|S| \leq k$, sodass jede Kante aus E einen Endpunkt in S hat.

- *NP – vollständig*
- Naive Algorithmen haben eine Laufzeit von $O(n^k)$
- Suchbaumalgorithmen laufen in $O(2^k \cdot (m + n))$

Knotenüberdeckungsproblem - Definition

Knotenüberdeckung

EINGABE: Graph $G = (V, E)$, natürliche Zahl $k \leq |V|$

AUSGABE: $S \subseteq V$ mit $|S| \leq k$, sodass jede Kante aus E einen Endpunkt in S hat.

- *NP – vollständig*
- Naive Algorithmen haben eine Laufzeit von $O(n^k)$
- Suchbaumalgorithmen laufen in $O(2^k \cdot (m + n))$

Graphreduktion

Graphreduktion für das Knotenüberdeckungsproblem

Graph $G = (V, E)$, natürliche Zahl k ; $VC(G, k)$

- Entfernen von Knoten und Kanten aus G
- Verkleinerung von k
- Problemkern $G' = (V', E')$:

$$VC(G, k) = VC(G', k') \cup VC(G \setminus G', k - k')$$

Graphreduktion für das Knotenüberdeckungsproblem

Graph $G = (V, E)$, natürliche Zahl k ; $VC(G, k)$

- Entfernen von Knoten und Kanten aus G
- Verkleinerung von k
- Problemkern $G' = (V', E')$:

$$VC(G, k) = VC(G', k') \cup VC(G \setminus G', k - k')$$

Graphreduktion für das Knotenüberdeckungsproblem

Graph $G = (V, E)$, natürliche Zahl k ; $VC(G, k)$

- Entfernen von Knoten und Kanten aus G
- Verkleinerung von k
- Problemkern $G' = (V', E')$:

$$VC(G, k) = VC(G', k') \cup VC(G \setminus G', k - k')$$

Graphreduktion für das Knotenüberdeckungsproblem

Graph $G = (V, E)$, natürliche Zahl k ; $VC(G, k)$

- Entfernen von Knoten und Kanten aus G
- Verkleinerung von k
- Problemkern $G' = (V', E')$:

$$VC(G, k) = VC(G', k') \cup VC(G \setminus G', k - k')$$

Graphreduktion für das Knotenüberdeckungsproblem

Graph $G = (V, E)$, natürliche Zahl k ; $VC(G, k)$

- Entfernen von Knoten und Kanten aus G
- Verkleinerung von k
- Problemkern $G' = (V', E')$:
$$VC(G, k) = VC(G', k') \cup VC(G \setminus G', k - k')$$

Einfache Reduktionsregeln

Reduktionsregeln

Graph $G = (V, E)$, natürliche Zahl k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad₀-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$
(Grad₁-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Reduktionsregeln

Graph $G = (V, E)$, natürliche Zahl k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad₀-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$
(Grad₁-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Reduktionsregeln

Graph $G = (V, E)$, natürliche Zahl k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad₀-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$
(Grad₁-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Reduktionsregeln

Graph $G = (V, E)$, natürliche Zahl k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad₀-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$
(Grad₁-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Reduktionsregeln

Graph $G = (V, E)$, natürliche Zahl k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad₀-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$
(Grad₁-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Testset

- 900 Graphen mit 1000 Knoten und bis zu 10000 Kanten
- LEDA:random_simple_undirected_graph
- Anwendung bis sich keine Änderung mehr ergibt

Testset

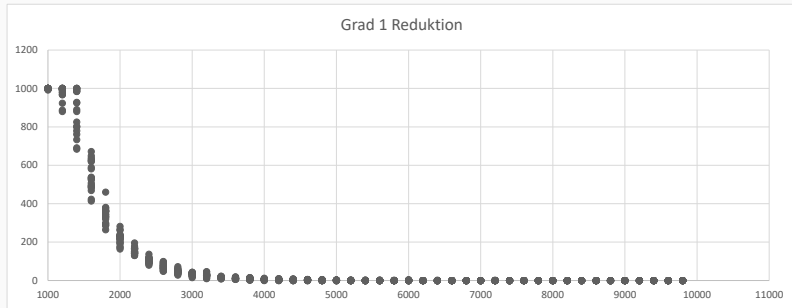
- 900 Graphen mit 1000 Knoten und bis zu 10000 Kanten
- LEDA:random_simple_undirected_graph
- Anwendung bis sich keine Änderung mehr ergibt

Testset

- 900 Graphen mit 1000 Knoten und bis zu 10000 Kanten
- LEDA:random_simple_undirected_graph
- Anwendung bis sich keine Änderung mehr ergibt

Testset

- 900 Graphen mit 1000 Knoten und bis zu 10000 Kanten
- LEDA:random_simple_undirected_graph
- Anwendung bis sich keine Änderung mehr ergibt



Kronenregel

Krone

Für einen Graphen $G = (V, E)$ besteht eine Krone aus $H \subseteq V$ und $I \subseteq V$ mit $H \cap I = \emptyset$, sodass

1. $H = N(I)$,
2. $\forall v, w \in I$ gilt $(vw) \notin E$ und
3. die Kanten zwischen H und I enthalten ein Matching indem alle Knoten aus H enthalten sind.

Krone

Für einen Graphen $G = (V, E)$ besteht eine Krone aus $H \subseteq V$ und $I \subseteq V$ mit $H \cap I = \emptyset$, sodass

1. $H = N(I)$,
2. $\forall v, w \in I$ gilt $(vw) \notin E$ und
3. die Kanten zwischen H und I enthalten ein Matching indem alle Knoten aus H enthalten sind.

Krone

Für einen Graphen $G = (V, E)$ besteht eine Krone aus $H \subseteq V$ und $I \subseteq V$ mit $H \cap I = \emptyset$, sodass

1. $H = N(I)$,
2. $\forall v, w \in I$ gilt $(vw) \notin E$ und
3. die Kanten zwischen H und I enthalten ein Matching indem alle Knoten aus H enthalten sind.

Krone

Für einen Graphen $G = (V, E)$ besteht eine Krone aus $H \subseteq V$ und $I \subseteq V$ mit $H \cap I = \emptyset$, sodass

1. $H = N(I)$,
2. $\forall v, w \in I$ gilt $(vw) \notin E$ und
3. die Kanten zwischen H und I enthalten ein Matching indem alle Knoten aus H enthalten sind.

Krone

Für einen Graphen $G = (V, E)$ besteht eine Krone aus $H \subseteq V$ und $I \subseteq V$ mit $H \cap I = \emptyset$, sodass

1. $H = N(I)$,
2. $\forall v, w \in I$ gilt $(vw) \notin E$ und
3. die Kanten zwischen H und I enthalten ein Matching indem alle Knoten aus H enthalten sind.

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2   $M_1 := \emptyset$ 
3   $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11      $I' := I$ 
12      $H := N(I)$ 
13      $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```


Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2   $M_1 := \emptyset$ 
3   $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11      $I' := I$ 
12      $H := N(I)$ 
13      $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2   $M_1 := \emptyset$ 
3   $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11      $I' := I$ 
12      $H := N(I)$ 
13      $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2   $M_1 := \emptyset$ 
3   $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11      $I' := I$ 
12      $H := N(I)$ 
13      $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```


Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Algorithmus

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Kronenregel - Laufzeit

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Zeilen 1-5: $m \cdot d$

Kronenregel - Laufzeit

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Zeile 6: n

Kronenregel - Laufzeit

```
0   $G = (V, E)$ ,  $n := |V|$ ,  $m := |E|$ ,  $d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 :=$  Maximal Matching von  $G$ 
2   $M_1 := \emptyset$ 
3   $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O :=$  nicht gepaarte Knoten in  $M_1$ 
7   $M_2 :=$  Maximum Matching von  $B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I :=$  nicht gepaarte Knoten aus  $O$  in  $M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11      $I' := I$ 
12      $H := N(I)$ 
13      $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Zeile 7 (1): $nd/2$

Kronenregel - Laufzeit

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 :=$  Maximal Matching von  $G$ 
2   $M_1 := \emptyset$ 
3   $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O :=$  nicht gepaarte Knoten in  $M_1$ 
7   $M_2 :=$  Maximum Matching von  $B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I :=$  nicht gepaarte Knoten aus  $O$  in  $M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11      $I' := I$ 
12      $H := N(I)$ 
13      $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Zeile 7 (2): $\sqrt{n} \cdot m$ (LEDA:mcb_matching, Hopcroft und Karp)

Kronenregel - Laufzeit

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

Zeile 8: n

Kronenregel - Laufzeit

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1   $M_1 := \text{Maximal Matching von } G$ 
2     $M_1 := \emptyset$ 
3     $\forall e \in E:$ 
4       $M_1 = M_1 \cup e$ 
5      Entferne  $e$  und  $N(e)$  aus der weiteren Betrachtung
6   $O := \text{nicht gepaarte Knoten in } M_1$ 
7   $M_2 := \text{Maximum Matching von } B = (O, N(O), \{uv | u \in O \wedge v \in N(O)\})$ 
8   $I := \text{nicht gepaarte Knoten aus } O \text{ in } M_2$ 
9   $I' := \emptyset$ 
10 while  $I' \neq I$ 
11    $I' := I$ 
12    $H := N(I)$ 
13    $I := I \cup \{u \in O | \exists v \in H (uv \in M_2)\}$ 
14 Entferne  $N(I)$  aus  $G$ 
```

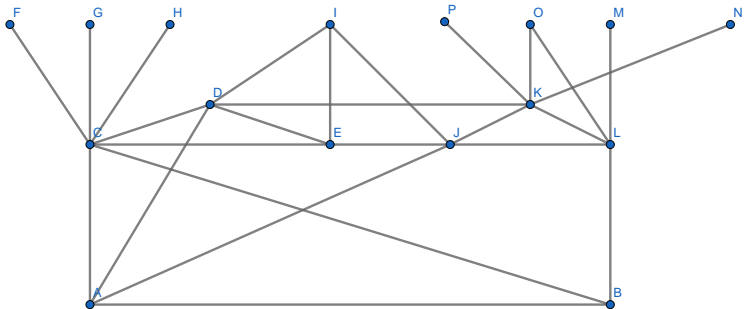
Zeilen 10-13: $n \cdot d$

$$\begin{aligned} & m \cdot d + n + nd/2 + \sqrt{n} \cdot m + n + n \cdot d \\ &= d(m + n) + 2n + nd/2 + \sqrt{n} \cdot m \\ &\Rightarrow O(\sqrt{n} \cdot m) \end{aligned}$$

$$\begin{aligned} & m \cdot d + n + nd/2 + \sqrt{n} \cdot m + n + n \cdot d \\ &= d(m + n) + 2n + nd/2 + \sqrt{n} \cdot m \\ &\Rightarrow O(\sqrt{n} \cdot m) \end{aligned}$$

$$\begin{aligned} & m \cdot d + n + nd/2 + \sqrt{n} \cdot m + n + n \cdot d \\ &= d(m + n) + 2n + nd/2 + \sqrt{n} \cdot m \\ &\Rightarrow O(\sqrt{n} \cdot m) \end{aligned}$$

Kronenregel - Besonderheiten



Kronenregel - Besonderheiten

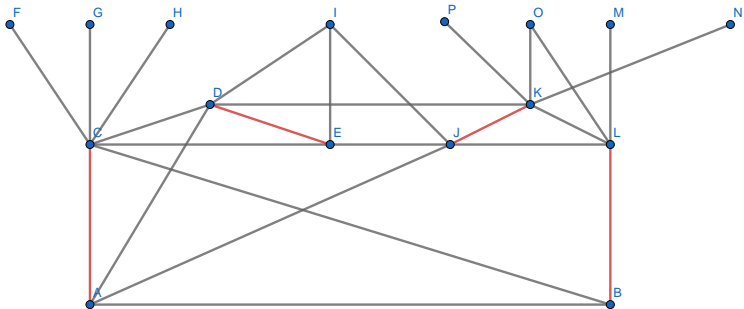


Tabelle 1: Mindestens ein Knoten mit Einschränkung

Grad der Knoten	Anwendungen	Reduktion
keine Einschränkung	0.29	13.04
>1	0.29	13.04
>2	0.29	13.22
>3	0.27	12.92
>4	0.3	13.71
>5	0.31	13.38
Größte Anzahl	0.32	13.44
Durchschnittliche Anzahl	0.29	12.98

Tabelle 1: Mindestens ein Knoten mit Einschränkung

Grad der Knoten	Anwendungen	Reduktion
keine Einschränkung	0.29	13.04
>1	0.29	13.04
>2	0.29	13.22
>3	0.27	12.92
>4	0.3	13.71
>5	0.31	13.38
Größte Anzahl	0.32	13.44
Durchschnittliche Anzahl	0.29	12.98

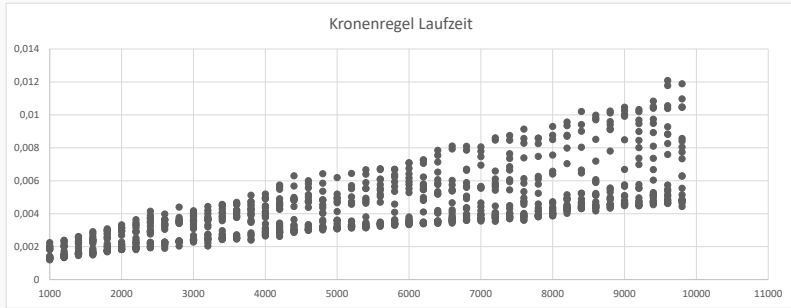
Tabelle 2: Beide Knoten mit Einschränkung

Grad der Knoten	Anwendungen	Reduktion
keine Einschränkung	0.29	13.04
>1	0.36	15.34
>2	0.41	16.96
>3	0.39	16.52
>4	0.4	15.78
>5	0.4	15.72
Größte Anzahl	0.29	13.06
Durchschnittliche Anzahl	0.46	19.77

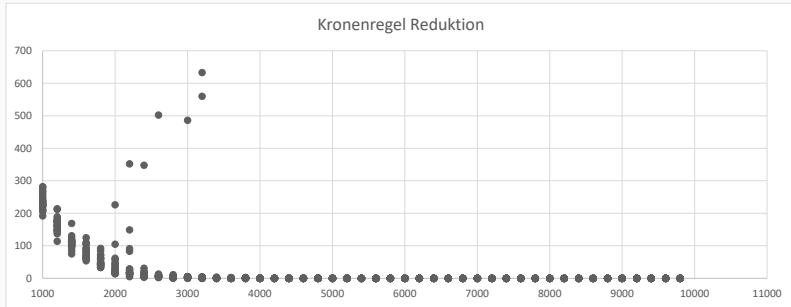
Tabelle 2: Beide Knoten mit Einschränkung

Grad der Knoten	Anwendungen	Reduktion
keine Einschränkung	0.29	13.04
>1	0.36	15.34
>2	0.41	16.96
>3	0.39	16.52
>4	0.4	15.78
>5	0.4	15.72
Größte Anzahl	0.29	13.06
Durchschnittliche Anzahl	0.46	19.77

Kronenregel - Laufzeit



Kronenregel - Ergebnisse



Nemhauser-Trotter-Regel

Nemhauser-Trotter-Theorem

Für einen Graphen $G = (V, E)$ können zwei disjunkte Mengen C_0 und V_0 gefunden werden, sodass

- 1. C_0 in einer minimalen Knotenüberdeckung von G enthalten ist,*
- 2. der Teilgraph $G[V_0]$ eine Knotenüberdeckung der Größe $\leq |V_0|/2$ hat,*
- 3. und $VC(G) = VC(G[V_0]) \cup C_0$ gilt.*

Nemhauser-Trotter-Theorem

Für einen Graphen $G = (V, E)$ können zwei disjunkte Mengen C_0 und V_0 gefunden werden, sodass

1. C_0 in einer minimalen Knotenüberdeckung von G enthalten ist,
2. der Teilgraph $G[V_0]$ eine Knotenüberdeckung der Größe $\leq |V_0|/2$ hat,
3. und $VC(G) = VC(G[V_0]) \cup C_0$ gilt.

Nemhauser-Trotter-Theorem

Für einen Graphen $G = (V, E)$ können zwei disjunkte Mengen C_0 und V_0 gefunden werden, sodass

1. C_0 in einer minimalen Knotenüberdeckung von G enthalten ist,
2. der Teilgraph $G[V_0]$ eine Knotenüberdeckung der Größe $\leq |V_0|/2$ hat,
3. und $VC(G) = VC(G[V_0]) \cup C_0$ gilt.

Nemhauser-Trotter-Theorem

Für einen Graphen $G = (V, E)$ können zwei disjunkte Mengen C_0 und V_0 gefunden werden, sodass

- 1. C_0 in einer minimalen Knotenüberdeckung von G enthalten ist,*
- 2. der Teilgraph $G[V_0]$ eine Knotenüberdeckung der Größe $\leq |V_0|/2$ hat,*
- 3. und $VC(G) = VC(G[V_0]) \cup C_0$ gilt.*

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1  Bipartiten Graphen erstellen  $B = (V, V', E')$ 
2    mit  $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$ 
3  Maximum Matching  $M$  von  $B$  bestimmen
4   $C_B := VC(B)$ 
5   $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$ 
6   $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$ 
```

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1  Bipartiten Graphen erstellen  $B = (V, V', E')$ 
2    mit  $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$ 
3  Maximum Matching  $M$  von  $B$  bestimmen
4   $C_B := VC(B)$ 
5   $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$ 
6   $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$ 
```

- 0 $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$
- 1 Bipartiten Graphen erstellen $B = (V, V', E')$
- 2 mit $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$
- 3 Maximum Matching M von B bestimmen
- 4 $C_B := VC(B)$
- 5 $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$
- 6 $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$

- 0 $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$
- 1 Bipartiten Graphen erstellen $B = (V, V', E')$
- 2 mit $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$
- 3 Maximum Matching M von B bestimmen
- 4 $C_B := VC(B)$
- 5 $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$
- 6 $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$

- 0 $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$
- 1 Bipartiten Graphen erstellen $B = (V, V', E')$
- 2 mit $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$
- 3 Maximum Matching M von B bestimmen
- 4 $C_B := VC(B)$
- 5 $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$
- 6 $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$

- 0 $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$
- 1 Bipartiten Graphen erstellen $B = (V, V', E')$
- 2 mit $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$
- 3 Maximum Matching M von B bestimmen
- 4 $C_B := VC(B)$
- 5 $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$
- 6 $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$


```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1  Bipartiten Graphen erstellen  $B = (V, V', E')$ 
2    mit  $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$ 
3  Maximum Matching  $M$  von  $B$  bestimmen
4   $C_B := VC(B)$ 
5   $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$ 
6   $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$ 
```

Zeilen 1-2: $n \cdot 2d$

```
0   $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$ 
1  Bipartiten Graphen erstellen  $B = (V, V', E')$ 
2    mit  $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$ 
3  Maximum Matching  $M$  von  $B$  bestimmen
4   $C_B := VC(B)$ 
5   $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$ 
6   $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$ 
```

Zeilen 3-4: $\sqrt{n} \cdot m$ (LEDA:mcb_matching, Hopcroft und Karp)

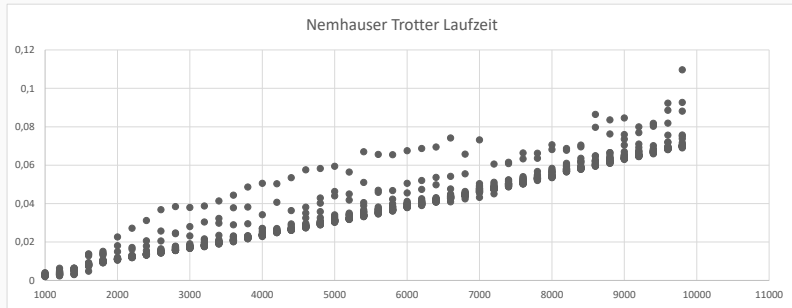
- 0 $G = (V, E), n := |V|, m := |E|, d := \text{maximaler Grad eines Knoten aus } G$
- 1 Bipartiten Graphen erstellen $B = (V, V', E')$
- 2 mit $E' := \{\{x, y'\}, \{x', y\} \mid \{x, y\} \in E\}$
- 3 Maximum Matching M von B bestimmen
- 4 $C_B := VC(B)$
- 5 $C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$
- 6 $V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$

Zeilen 5-6: $2n + k \cdot d$

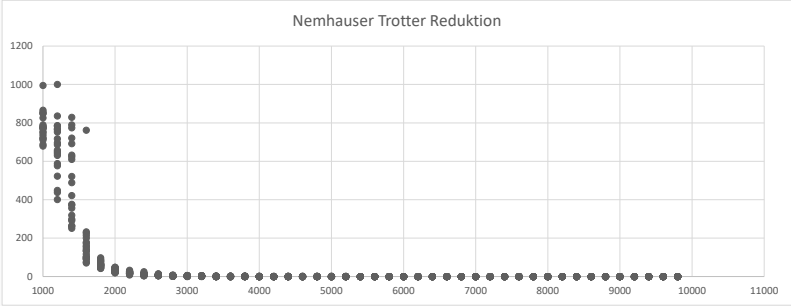
$$n \cdot 2d + \sqrt{n} \cdot m + 2n + k \cdot d$$
$$\Rightarrow O(n + \sqrt{n} \cdot m)$$

$$\begin{aligned} & n \cdot 2d + \sqrt{n} \cdot m + 2n + k \cdot d \\ \Rightarrow & O(n + \sqrt{n} \cdot m) \end{aligned}$$

NT-Regel - Laufzeit



NT-Regel - Ergebnisse



Vergleich

Tabelle 3: Anwendung einzelner Reduktionsregeln

Reduktionsregel	Anwendungen	Reduktion	CPU-Zeit
Nemhauser-Trotter	0.27	50.3	0.041s
Kronenregel	0.46	19.77	0.004s
Grad 1	1.32	99.06	0.0006s

Anwendung

Tabelle 4: Anwendung kombinierter Reduktionsregeln

Kombination	Anwendungen ₁	Anwendungen ₂	Anwendungen ₃	Reduktion
K - G ₁	3.63	4.3	-	331.8
G ₁ - K	4.37	3.22	-	331.17
K - NT	0.8	0.38	-	68.28
NT - K	0.45	0.56	-	68.6
G ₁ - NT	1.33	0.017	-	99.87
NT - G ₁	0.28	1.13	-	99.87

Tabelle 5: Anwendung kombinierter Reduktionsregeln

Kombination	Anwendungen ₁	Anwendungen ₂	Anwendungen ₃	Reduktion
K - G ₁ - NT	3.61	4.29	0.11	334.67
K - NT - G ₁	3.6	0.87	3.39	334.83
G ₁ - NT - K	4.36	0.12	3.2	334.17
G ₁ - K - NT	3.61	3.2	0.65	334.16
NT - K - G ₁	0.39	3.44	4.03	335.2
NT - G ₁ - K	0.91	3.42	3.2	334.16

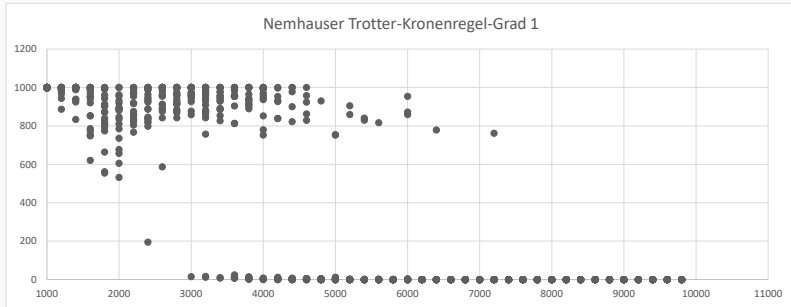













Tabelle 6: Besondere Graphen für die Dreierkombinationen von Regeln



Graph	Reduktionsregeln	Anwend. ₁	Anwend. ₂	Anwend. ₃	Reduktion
Graph ₁	NT - K - G ₁	1	5	6	195
	G ₁ - NT - K	6	0	5	195
	Nemhauser-Trotter	1	-	-	11
	Kronenregel	1	-	-	11
	Grad ₁	2	-	-	91
	Kronenregel - Grad ₁	6	5	-	195
Graph ₂	NT - K - G ₁	1	9	2	762
	Nemhauser-Trotter	0	-	-	0
	Kronenregel	0	-	-	0
	Grad ₁	1	-	-	2
	Kronenregel - Grad ₁	9	3	-	762

Fazit und Ausblick

- Form von randomgraphen untersuchen
 - Welche Form hinterlassen die Reduktionsregeln
 - Welche Form wäre für die Nemhauser Trotter am besten
- Warum ist die Nemhauser-Trotter-Regel in der Praxis so schlecht?
- Wie ergibt sich in bei den Reduktionen (Abbildung ??) der Große Unterschied zwischen den Graphen im Bereich der Kantenmenge zwischen 3000 und 4600?
- Wieso ist das Matching M1 so wichtig für die Kronenregel?

-  E. C. Sewell, S. H. Jacobson, and H. Kaul, “Reductions for the stable set problem,” *Algorithmic Operations Research Vol.6* 40–55, 2011.
-  F. N. Abu-Khzam, M. A. Langston, P. Shanbhag, and C. T. Symons, “Scalable parallel algorithms for fpt problems,” *Algorithmica, Volume 45, Number 3*, 269-284, 2006.
-  F. Abu-Khzam, R. Collins, M. Fellows, M. Langston, and W. S. C. Symons, “Kernelization algorithms for the vertex cover problem: Theory and experiments,” *Proceedings of the 6th Workshop on Algorithm Engineering and Experiments (ALENEX), ACM/SIAM, Proc. Applied Mathematics 115*, 2004.
-  J. Chen, I. A. Kanj, and W. Jia, “Vertex cover: further observations and further improvements,” *Journal of Algorithms* 41, 280 - 301, 2001.

-  S. R. aniel Mölle and P. Rossmanith, “Enumerate and expand: New runtime bounds for vertex cover variants,” *Computing and Combinatorics. Lecture Notes in Computer Science*, vol 4112, 265-273, 2006.
-  R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
-  K. Mehlhorn and S. Näher, *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 2006.
-  M. Cygan, *Parameterized algorithms*. Springer Cham, 2015.
-  F. Gurski, *Exakte Algorithmen für schwere Graphenprobleme*. Springer Heidelberg, 2010.
-  R. G. Downey and M. R. Fellows, *Fundamentals of parameterized complexity*. Springer London, 2013.
-  ———, *Parametrized Complexity*. Springer New York, 1997.

-  M. R. Garey and D. S. Johnson, *Computers and Intractability : a guide to the theory of NP-completeness*. Freeman New York, 2007.
-  G. Valiente, *Algorithms on Trees and Graphs*. Springer, Berlin, Heidelberg, 2002.