Kolloquium zur Bachelorarbeit

Referent:	Benedikt Lüken-Winkels
Prüfer:	Prof. Dr. Henning Fernau
	Prof. Dr. Stefan Näher

07. März 2018

Universität Trier

Knotenüberdeckungsproblem

Knotenüberdeckung

EINGABE: *Graph* $G = (V, E)$, *positive Integer* $k \leq |V|$

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

Graphreduktion

Einfache Reduktionsregeln

Einfache Regeln

Graph $G = (V, E)$*, Integer* k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad$_0$-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$ (Grad$_1$-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Einfache Regeln

Graph $G = (V, E)$, *Integer* k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad$_0$-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$ (Grad$_1$-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Einfache Regeln

Graph $G = (V, E)$, Integer k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad_0-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$ (Grad_1-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Einfache Regeln

Graph $G = (V, E)$, *Integer* k

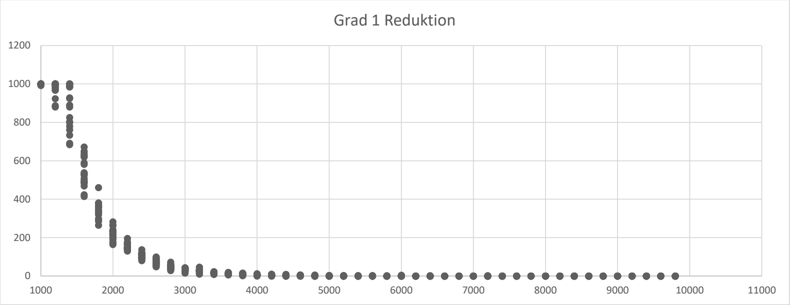
1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad_0-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$ (Grad_1-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

Einfache Regeln

Graph $G = (V, E)$, Integer k

1. $v \in V$ hat keine Kanten $\Rightarrow V = V \setminus v$ (Grad$_0$-Regel)
2. $v \in V$ hat genau eine Kante $\Rightarrow V = V \setminus (v \cup N(v)); k = k - 1$ (Grad$_1$-Regel)
3. $v \in V$ hat mehr als k Kanten $\Rightarrow V = V \setminus (v); k = k - 1$ (Buss-Regel)

$Grad_1$ - Ergebnisse



Kronenregel

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
    $M_1 := \emptyset$
    $\forall e \in$ E:
        $M_1 = M_1 \cup e$
        Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
    $I' := I$
    $H := N(I)$
    $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
    $M_1 := \emptyset$
    $\forall e \in$ E:
        $M_1 = M_1 \cup e$
        Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
    $I' := I$
    $H := N(I)$
    $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

```
G = (V, E)
M_1 := Maximal Matching von G
  M_1 := ∅
  ∀e ∈ E:
    M_1 = M_1 ∪ e
    Entferne e und N(e) aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von B = G[O ∪ N(O)]
I := nicht gepaarte Knoten aus O in M_2
I' := ∅
while I' ≠ I
  I' := I
  H := N(I)
  I := I ∪ {∀u ∈ O|∃v ∈ H(uv ∈ M_2)}
Entferne N(I) aus G
```

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H (uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H (uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H (uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
    $M_1 := \emptyset$
    $\forall e \in$ E:
        $M_1 = M_1 \cup e$
        Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
    $I' := I$
    $H := N(I)$
    $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

$G = (V, E)$

M_1 := Maximal Matching von G

$M_1 := \emptyset$

$\forall e \in$ E:

$M_1 = M_1 \cup e$

Entferne e und $N(e)$ aus der weiteren Betrachtung

O := nicht gepaarte Knoten in M_1

M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$

I := nicht gepaarte Knoten aus O in M_2

$I' := \emptyset$

while $I' \neq I$

$I' := I$

$H := N(I)$

$I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$

Entferne $N(I)$ aus G

Kronenregel - Algorithmus

```
G = (V, E)
M_1 := Maximal Matching von G
    M_1 := ∅
    ∀e ∈ E:
        M_1 = M_1 ∪ e
        Entferne e und N(e) aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von B = G[O ∪ N(O)]
I := nicht gepaarte Knoten aus O in M_2
I' := ∅
while I' ≠ I
    I' := I
    H := N(I)
    I := I ∪ {∀u ∈ O|∃v ∈ H(uv ∈ M_2)}
Entferne N(I) aus G
```

Kronenregel - Algorithmus

$G = (V, E)$
M_1 := Maximal Matching von G
  $M_1 := \emptyset$
  $\forall e \in$ E:
    $M_1 = M_1 \cup e$
    Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
  $I' := I$
  $H := N(I)$
  $I := I \cup \{\forall u \in O | \exists v \in H (uv \in M_2)\}$
Entferne $N(I)$ aus G

Kronenregel - Algorithmus

```
G = (V, E)
M_1 := Maximal Matching von G
   M_1 := \emptyset
   \forall e \in E:
      M_1 = M_1 \cup e
      Entferne e und N(e) aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von B = G[O \cup N(O)]
I := nicht gepaarte Knoten aus O in M_2
I' := \emptyset
while I' \neq I
   I' := I
   H := N(I)
   I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}
Entferne N(I) aus G
```

Kronenregel - Laufzeit

0 $G = (V, E)$
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 := Maxmimum Matching von $B = G[O \cup N(O)]$
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     $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
14 Entferne $N(I)$ aus G

Zeilen 1-5: $m \cdot d$

Kronenregel - Laufzeit

0 $G = (V, E)$
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 := Maxmimum Matching von $B = G[O \cup N(O)]$
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    $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
14 Entferne $N(I)$ aus G

Zeile 6: n

Kronenregel - Laufzeit

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

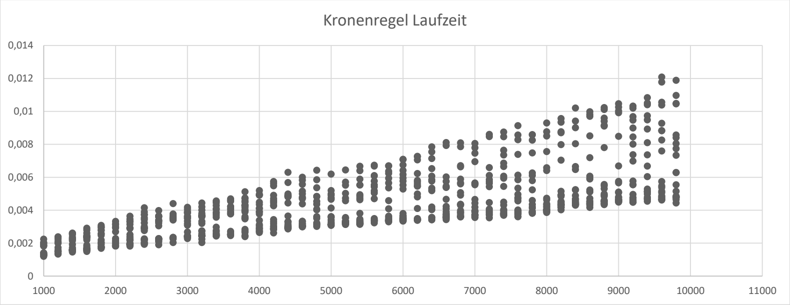
Zeile 7: $\frac{2k^2}{2} + \sqrt{2k} \cdot k$ (LEDA:mcb_matching, Hopcroft and Karp)

Kronenregel - Laufzeit

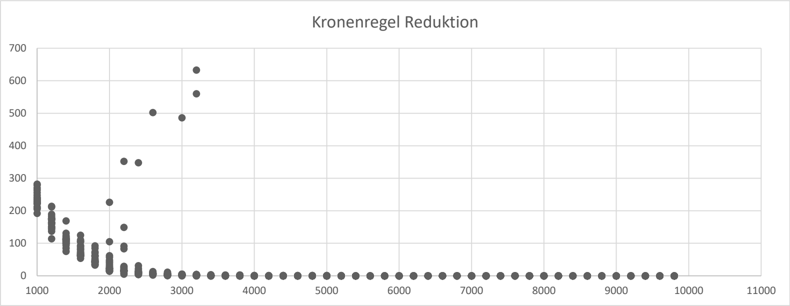
$G = (V, E)$
M_1 := Maximal Matching von G
    $M_1 := \emptyset$
    $\forall e \in$ E:
        $M_1 = M_1 \cup e$
        Entferne e und $N(e)$ aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von $B = G[O \cup N(O)]$
I := nicht gepaarte Knoten aus O in M_2
$I' := \emptyset$
while $I' \neq I$
    $I' := I$
    $H := N(I)$
    $I := I \cup \{\forall u \in O | \exists v \in H(uv \in M_2)\}$
Entferne $N(I)$ aus G

```
G = (V, E)
M_1 := Maximal Matching von G
  M_1 := ∅
  ∀e ∈ E:
    M_1 = M_1 ∪ e
    Entferne e und N(e) aus der weiteren Betrachtung
O := nicht gepaarte Knoten in M_1
M_2 := Maxmimum Matching von B = G[O ∪ N(O)]
I := nicht gepaarte Knoten aus O in M_2
I' := ∅
while I' ≠ I
  I' := I
  H := N(I)
  I := I ∪ {∀u ∈ O|∃v ∈ H(uv ∈ M_2)}
Entferne N(I) aus G
```

Kronenegel - Laufzeit



Kronenregel - Ergebnisse



Nemhauser-Trotter-Regel

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-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-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.*

NT-Regel - Algorithmus

```
G = (V, E)
Bipartiden Graphen erstellen B = (V, V', E')
  mit E' := {{x, y'}, {x', y}|{x, y} ∈ E}
Maximum Matching M von B bestimmen
C_B := VC(B)
C_0 := {x ∈ V | x ∈ C_B und x' ∈ C_B}
V_0 := {x ∈ V | entweder x ∈ C_B oder x' ∈ C_B}
```

NT-Regel - Algorithmus

$G = (V, E)$
`Bipartiden Graphen erstellen` $B = (V, V', E')$
   `mit` $E' := \{\{x, y'\}, \{x', y\} | \{x, y\} \in E\}$
`Maximum Matching` M `von` B `bestimmen`
$C_B := VC(B)$
$C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$
$V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$

NT-Regel - Algorithmus

```
G = (V, E)
Bipartiden Graphen erstellen B = (V, V', E')
  mit E' := {{x, y'}, {x', y} | {x, y} ∈ E}
Maximum Matching M von B bestimmen
C_B := VC(B)
C_0 := {x ∈ V | x ∈ C_B und x' ∈ C_B}
V_0 := {x ∈ V | entweder x ∈ C_B oder x' ∈ C_B}
```

$G = (V, E)$
Bipartiden Graphen erstellen $B = (V, V', E')$
  mit $E' := \{\{x, y'\}, \{x', y\} | \{x, y\} \in E\}$
Maximum Matching M von B bestimmen
$C_B := VC(B)$
$C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$
$V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$

NT-Regel - Algorithmus

$G = (V, E)$
`Bipartiden Graphen erstellen` $B = (V, V', E')$
  `mit` $E' := \{\{x, y'\}, \{x', y\} | \{x, y\} \in E\}$
`Maximum Matching` M `von` B `bestimmen`
$C_B := VC(B)$
$C_0 := \{x \in V \mid x \in C_B \textit{ und } x' \in C_B\}$
$V_0 := \{x \in V \mid \textit{entweder } x \in C_B \textit{ oder } x' \in C_B\}$

NT-Regel - Algorithmus

$G = (V, E)$
`Bipartiden Graphen erstellen` $B = (V, V', E')$
`mit` $E' := \{\{x, y'\}, \{x', y\} | \{x, y\} \in E\}$
`Maximum Matching` M `von` B `bestimmen`
$C_B := VC(B)$
$C_0 := \{x \in V \mid x \in C_B \; und \; x' \in C_B\}$
$V_0 := \{x \in V \mid entweder \; x \in C_B \; oder \; x' \in C_B\}$

NT-Regel - Laufzeit

```
0  G = (V, E)
1  Bipartiden Graphen erstellen B = (V, V', E')
2     mit E' := {{x, y'}, {x', y} | {x, y} ∈ E}
3  Maximum Matching M von B bestimmen
4  C_B := VC(B)
5  C_0 := {x ∈ V | x ∈ C_B und x' ∈ C_B}
6  V_0 := {x ∈ V | entweder x ∈ C_B oder x' ∈ C_B}
```

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

NT-Regel - Laufzeit

0 $G = (V, E)$
1 `Bipartiden Graphen erstellen` $B = (V, V', E')$
2 `mit` $E' := \{\{x, y'\}, \{x', y\} | \{x, y\} \in E\}$
3 `Maximum Matching` M `von` B `bestimmen`
4 $C_B := VC(B)$
5 $C_0 := \{x \in V \mid x \in C_B \; und \; x' \in C_B\}$
6 $V_0 := \{x \in V \mid entweder \; x \in C_B \; oder \; x' \in C_B\}$

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

NT-Regel - Laufzeit

```
0 G = (V, E)
1 Bipartiden Graphen erstellen B = (V, V', E')
2    mit E' := {{x, y'}, {x', y} | {x, y} ∈ E}
3 Maximum Matching M von B bestimmen
4 C_B := VC(B)
5 C_0 := {x ∈ V | x ∈ C_B und x' ∈ C_B}
6 V_0 := {x ∈ V | entweder x ∈ C_B oder x' ∈ C_B}
```

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

NT-Regel - Laufzeit

$G = (V, E)$

`Bipartiden Graphen erstellen` $B = (V, V', E')$

`mit` $E' := \{\{x, y'\}, \{x', y\} | \{x, y\} \in E\}$

`Maximum Matching` M `von` B `bestimmen`

$C_B := VC(B)$

$C_0 := \{x \in V \mid x \in C_B \text{ und } x' \in C_B\}$

$V_0 := \{x \in V \mid \text{entweder } x \in C_B \text{ oder } x' \in C_B\}$

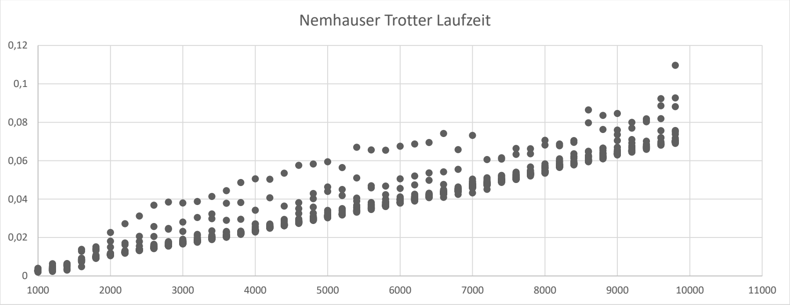
$n \cdot 2d + \sqrt{n} \cdot m + 2n + k \cdot d$

NT-Regel - Laufzeit

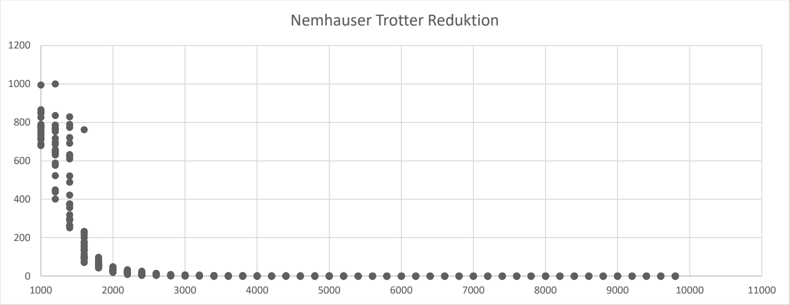
$G = (V, E)$
`Bipartiden Graphen erstellen` $B = (V, V', E')$
`  mit` $E' := \{\{x, y'\}, \{x', y\} | \{x, y\} \in E\}$
`Maximum Matching` M `von` B `bestimmen`
$C_B := VC(B)$
$C_0 := \{x \in V \mid x \in C_B \;\; \mathit{und} \;\; x' \in C_B\}$
$V_0 := \{x \in V \mid \mathit{entweder} \;\; x \in C_B \;\; \mathit{oder} \;\; x' \in C_B\}$

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

NT-Regel - Laufzeit



NT-Regel - Ergebnisse



Vergleich

Anwendung

Tabelle 1: 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
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

Implementierung

Fazit