

# Untersuchung von Graphreduktionsregeln beim Knotenüberdeckungsproblem

## Bachelorarbeit

zur Erlangung des akademischen Grades  
Bachelor (B.Sc.)

Universität Trier  
FB IV - Informatikwissenschaften  
Lehrstuhl für theoretische Informatik

Gutachter:	Prof. Dr. Henning Fernau Prof. Dr. Stefan Näher
Betreuer:	Prof. Dr. Henning Fernau

Vorgelegt am xx.xx.xxxx von:

Benedikt Lüken-Winkels  
Bahnhofstraße 32  
54292 Trier  
s4beluek@uni-trier.de  
Matr.-Nr. 1138844

# Zusammenfassung

Hier steht eine Kurzzusammenfassung (Abstract) der Arbeit. Stellen Sie kurz und präzise Ziel und Gegenstand der Arbeit, die angewendeten Methoden, sowie die Ergebnisse der Arbeit dar. Halten Sie dabei die ersten Punkten eher kurz und fokussieren Sie die Ergebnisse. Bewerten Sie auch die Ergebnissen und ordnen Sie diese in den Kontext ein.

Die Kurzzusammenfassung sollte maximal 1 Seite lang sein.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problemstellung . . . . .	1
1.3	Zielsetzung . . . . .	1
1.4	Gliederung/Aufbau der Arbeit . . . . .	2
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Knotenüberdeckung . . . . .	3
2.2	Nemhauser-Trotter Reduktionsregeln . . . . .	3
2.3	Kronenregel . . . . .	3
2.4	Einfache Reduktionsregeln . . . . .	3
<b>3</b>	<b>Analyse</b>	<b>4</b>
3.1	Anforderungen . . . . .	4
3.2	Bewertung der Reduktionsregeln . . . . .	4
3.3	Anwendung der Reduktionsregeln . . . . .	5
3.4	Interpretation der Ergebnisse . . . . .	8
3.5	Zusammenfassung . . . . .	8
<b>4</b>	<b>Entwurf / Konzeption</b>	<b>10</b>
4.1	Abschnitt 1 . . . . .	10
4.2	Abschnitt 2 . . . . .	10
4.3	Zusammenfassung . . . . .	10
<b>5</b>	<b>Implementierung</b>	<b>11</b>
5.1	Kronenregel . . . . .	11
<b>6</b>	<b>Evaluation</b>	<b>13</b>
6.1	Abschnitt 1 . . . . .	13
6.2	Abschnitt 2 . . . . .	13
6.3	Zusammenfassung . . . . .	13
<b>7</b>	<b>Diskussion und Ausblick</b>	<b>14</b>
	<b>Literaturverzeichnis</b>	<b>15</b>

# Abbildungsverzeichnis

3.1	Anwendung der Kronenregel. . . . .	7
-----	------------------------------------	---

# Tabellenverzeichnis

3.1	Bewertung der Reduktionsregeln . . . . .	5
3.2	Anwendung einzelner Reduktionsregeln . . . . .	6
3.3	Besondere Graphen für die Kronenregel . . . . .	8
3.4	Anwendung kombinierter Reduktionsregeln . . . . .	9
5.1	Mindestens ein Knoten mit Einschränkung . . . . .	12
5.2	Beide Knoten mit Einschränkung . . . . .	12

# 1. Einleitung

Die Einleitung besteht aus der Motivation, der Problemstellung, der Zielsetzung und einem ersten Überblick über den Aufbau der Arbeit.

## 1.1 Motivation

- Definition
- Wo findet das Knotenüberdeckungsproblem Anwendung?
- Woher kommt die Komplexität?
- Was macht eine schwere Instanz aus?
- Wie sieht eine schwere Instanz aus?

## 1.2 Problemstellung

- Woher kommt die Komplexität?
- Was macht eine schwere Instanz aus?
- Wie sieht eine schwere Instanz aus?
- Effekt von Graphreduktionsalgorithmen auf die Problemkomplexität

## 1.3 Zielsetzung

- Kategorisierung der Regeln?
- Bewertungskriterien für einen GRalgorithmus
  - Laufzeit (Parametrisierung)
  - Erwartete Reduktion/Wie oft wird die Regel angewandt
  - Ressourcenverbrauch
  - Wie gut ist das Ergebnis im Vergleich zu anderen Algorithmen?
- Wie funktionieren die GRA in Kombination?
- Wie sehen Graphen aus, auf die keine Regel anwendbar ist?
- Wie sehen Graphen aus, auf die genau eine Regel anwendbar ist?
- Welche Regeln werden untersucht?

## 1.4 Gliederung/Aufbau der Arbeit

Was enthalten die weiteren Kapitel? Wie ist die Arbeit aufgebaut? Welche Methodik wird verfolgt?

## 2. Grundlagen

Beschreibung der verschiedenen Reduktionsregeln und wie funktionieren. Mit graphischen Beispielen und Pseudocode?

### 2.1 Knotenüberdeckung

...

### 2.2 Nemhauser-Trotter Reduktionsregeln

...

### 2.3 Kronenregel

...

### 2.4 Einfache Reduktionsregeln

...



## 3. Analyse

### 3.1 Anforderungen

Um einen Graphen, der zur Diskussion steht auf einen Problemkern zu reduzieren können verschiedene Reduktionsregeln verwendet werden.

Zunächst werden die Regeln anhand der folgenden Kriterien beurteilt:

- Laufzeit
- Erwartete Reduktion
- Ressourcenverbrauch

Dann wird die Anwendung und der Effekt der Regeln auf das eigens erstellte Testset untersucht, um folgende Fragen zu beantworten:

- Wie effektiv sind die Reduktionsregeln in der Anwendung?
  - Wie oft sind die Regeln anwendbar?
  - Wie viel wird reduziert?
- Wie (gut) funktionieren die Reduktionsregeln in Kombination?
- Wie sehen Graphen aus, auf die keine Regel anwendbar ist?
- Wie sehen Graphen aus, auf die genau eine Regel anwendbar ist?

### 3.2 Bewertung der Reduktionsregeln

Um die verwendeten Reduktionsregeln zu bewerten, beziehungsweise zu vergleichen, betrachten wir auf der einen Seite die theoretisch mögliche Reduktion und auf der anderen, wie sich der implementierte Algorithmus verhält.

In Tabelle 3.1 stehen die Laufzeiten in der  $O$ -Notation und erwartete Größe der reduzierten Problemkerns der Nemhauser Trotter Regel [Nied06, S67], der Kronenregel [Nied06, S71] [Uhri17], die der Buss Regel [Nied06, S67] und von einfachen Reduktionsregeln (Grad 0 und Grad 1) gegenüber. Die Laufzeiten der einfachen Reduktionsregeln ergeben sich folgt:

Tabelle 3.1: Bewertung der Reduktionsregeln

Reduktionsregel	Laufzeit	Reduktion
Nemhauser Trotter	$O(k^3)$	$\leq 2k$
Kronenregel	$O(\sqrt{ V } E )$	$\leq 3k$
Buss	$O(k V )$	$\leq k^2$
Grad 0	$O(m V )$ mit $m \leq 1$	*
Grad 1	$O(m V )$ mit $m < 1$	*

\* Die Reduktion dieser Regeln hängt stark von der Struktur des Graphen ab. Eine formale Bestimmung der zu erwartenden Reduktion steht noch aus.

$k$  beschreibt jeweils den Eingabeparameter der Knotenüberdeckung für einen Graphen  $G = (V, E)$

*Beweis.* Sei Graph  $G = (V, E)$

*Grad 0 – Reduktion*

1.Fall :  $|E| = 0$

*jeder Knoten aus  $V$  muss betrachtet werden.*

$\Rightarrow$  Laufzeit :  $O(|V|)$

2.Fall :  $|E| > 0$

*Sobald bei der Iteration ein Knoten mit  $\text{Grad}(v) > 0$  ( $v \in V$ ) ausgewählt wird, können alle Nachbarn ignoriert werden.*

$\Rightarrow$  Laufzeit :  $O(m|V|)$  mit ( $m < 1$ )

□

*Beweis.* Sei Graph  $G = (V, E)$

*Grad 1 – Reduktion*

*Sobald bei der Iteration ein Knoten mit  $\text{Grad}(v) = 1$  ( $v \in V$ ) ausgewählt wird, wird der Nachbar in die Knotenberdeckung aufgenommen.*

$\Rightarrow$  Laufzeit :  $O(m|V|)$  mit ( $m < 1$ )

□

TODO was bringt die Bewertung?

### 3.3 Anwendung der Reduktionsregeln

Das Testset an dem die Algorithmen angewandt wurden besteht aus ungerichteten Graphen, die jeweils eine Knotenmenge von 1000 Knoten und eine Kantenanzahl

von 1000 bis 10000 (aufsteigend in 200er Schritten) umfassen. Von jeder *Graphklasse*, die sich durch die Kantenmenge auszeichnet gibt es 20 Exemplare. Dadurch entsteht ein Testset von 900 Graphen. Zum Erstellen der Graphen wurde die LEDA-Funktion `random_simple_undirected_graph(|V|,|E|)` [Uhri17] verwendet. Da Zufallsgraphen verwendet werden, gibt es keine Vorgabe für den Parameter  $k$ . Die obere Beschränkung der Kantenmenge (10000) hat sich bei den Tests ergeben, da ab einer bestimmten Dichte, beziehungsweise Knotenanzahl keine Reduktionsregel mehr erfolgreich ist. Dies kann mit der erwarteten Reduktion (siehe Tabelle 3.1) erklärt werden: Je dichter der Graph wird, desto größer wird  $k$  und sobald  $k > \frac{|V|}{2}$  ist, ist bei der Nemhauser Trotter-Regel und somit auch bei den anderen die theoretische Größe des reduzierten Problemkerns größer oder gleich der Menge der Knoten. Daher eignen sich viele Benchmarktests <sup>1 2</sup> nicht, was sich bei Testdurchläufen herausstellte. Die Knotenmenge 1000 erwies sich in den Tests als ausreichend hoher Wert, um das Einsetzen der Reduktionsregeln zu beobachten.

Die Reduktionsregeln wurden jeweils solange auf einen Graphen angewandt, bis sich keine Änderungen mehr ergeben haben. Erzielte eine der Regeln eine Reduktion, wurde der gesamte Vorgang wiederholt. Ausgeführt wurden die Experimente auf einem 2.6 GHz, Zweikern, Intel Core i5-3320M mit 8 GB Speicher. In der Tabelle 3.2 werden die durchschnittliche Anzahl der Anwendungen pro Graph, die durchschnittliche Reduktion (Anzahl der Knoten, die aus dem Graphen entfernt wurden), und die durchschnittliche CPU-Zeit, die pro Graph aufgebracht wurde. Die Buss Regel wurde von der Untersuchung ausgeschlossen, da kein Wert für  $k$  vorliegt, welcher für die Anwendung essentiell ist, während die restlichen Regeln auch ohne diesen Parameter verwendbar sind.

Tabelle 3.2 spiegelt die erwarteten Werte (3.1) in soweit wieder, als das die Nemhauser Trotter Regel eine deutlich bessere Reduktion als die Kronenregel erreicht, zumindest, wenn sie isoliert angewandt wird. Dabei werden weniger Durchläufe, allerdings ein höherer Rechenaufwand benötigt.

Tabelle 3.2: Anwendung einzelner Reduktionsregeln

Reduktionsregel	Anwendungen	Reduktion	CPU-Zeit
Nemhauser Trotter	0.27	50.3	0.014
Kronenregel	0.46	19.77	0.005
Grad 1	1.32	99.06	0.001

Während sich die Reduktion durch die Nemhauser Trotter und die Grad 1 Regel bei der Anwendung an dichteren Graphen, also Graphen mit einer höheren Kantenanzahl [Ummy00], erwartungsgemäß stetig verringerte, zeigten sich bei der Kronenregel einige Ausnahmen. Wie in Abbildung 3.1 zu sehen ist, setzten sich diese deutlich vom Durchschnitt ab. In Tabelle 3.3 werden die drei Graphen mit einer Kantenmenge über (beziehungsweise gleich) 3000 und einer durch die Kronenregel erreichten Reduktion  $> 480$  Knoten betrachtet. Keiner dieser Graphen ist bipartit. Bei allen drei Graphen führte die Anwendung der Nemhauser Trotter Regel zu keinerlei nennenswertem Effekt, weder isoliert, noch in Kombination mit den anderen Re-

<sup>1</sup><http://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/>

<sup>2</sup><http://sites.nlsde.buaa.edu.cn/kexu/benchmarks/graph-benchmarks.htm>

geln. Ähnlich verhielt sich zunächst die  $\text{Grad}_1$  Regel, wohingegen die Anwendung in Kombination einen großen Einfluss auf die Reduktion hatte. Kombiniert man die Regeln, scheint die Reihenfolge, in der sie eingesetzt werden schwer ins Gewicht zu fallen. Besonders fiel das bei  $\text{Graph}_1$  (3200 Kanten) auf. Wurde zuerst die  $\text{Grad}_1$  und dann die Kronenregel verwendet, war jeweils lediglich ein Durchlauf (Anwendung) zu beobachten. Die  $\text{Grad}_1$  Regel scheint den Graphen derart zu verändern, als dass kaum Reduktionsmöglichkeiten bei der Kronenregeln übrig bleiben. Betrachtet man dagegen das Ergebnis des  $\text{Graphen}_2$ , erzeugte die Anwendung in gerade dieser Reihenfolge eine Lösung des Problems: 1000 von 1000 Knoten reduziert, wovon 619 eine Knotenüberdeckung für den  $\text{Graphen}_2$  bilden. Auch hier war die Reihenfolge wieder wichtig, wie man bei der Reduktionsmenge beim entgegengesetzten Experiment (erst Kronenregel, dann  $\text{Grad}_1$  Regel) sieht, wo 858 Knoten entfernt wurden. Bei  $\text{Graph}_1$  und  $\text{Graph}_3$  zeichnet sich der übrig gebliebene Problemerkern nach der jeweils größten Reduktion dadurch aus, dass der Großteil der Knoten vom Grad 2 ist. Der Problemerkern  $G'_3$  von  $\text{Graph}_3$  lässt sich in die Knotenmengen  $V_1$  und  $V_2$  mit  $|V_1| = |V_2| = 5$  aufteilen. Bis auf die Knoten  $v_1 \in V_1$  und  $v_2 \in V_2$ , welche vom Grad 3 sind, hat jeder andere Knoten in  $G'_3$  (durch die vorherige Reduktion) den Grad 2. Für  $v_1$  und  $v_2$  existiert eine Kante  $(v_1, v_2)$  in  $G'_3$ , welche die Knotenmengen verbindet. Innerhalb der Knotenmengen existiert einen Zyklus (Kreis) [Ummy00], mit ungerader Knotenzahl, woraus sich folgern lässt, dass  $G'_3$  nicht bipartit ist. Hier ist keine der Reduktionsregeln mehr erfolgreich.

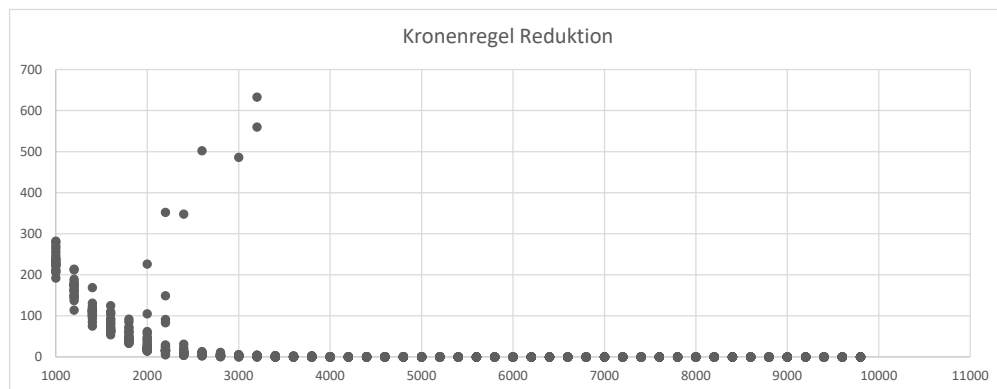


Abbildung 3.1: Anwendung der Kronenregel.

Tabelle 3.3: Besondere Graphen für die Kronenregel

Graph	Reduktionsregeln	Anwendungen <sub>1</sub>	Anwendungen <sub>2</sub>	Reduktion
Graph <sub>1</sub>	Kronenregel	11	-	560
	Nemhauser Trotter	1	-	4
	Grad <sub>1</sub>	1	-	18
	Grad <sub>1</sub> -Kronenregel	1	1	22
	Kronenregel - Grad <sub>1</sub>	14	11	946
Graph <sub>2</sub>	Kronenregel	13	-	486
	Nemhauser Trotter	1	-	6
	Grad <sub>1</sub>	2	-	40
	Grad <sub>1</sub> -Kronenregel	12	13	1000
	Kronenregel - Grad <sub>1</sub>	18	12	858
Graph <sub>3</sub>	Kronenregel	15	-	633
	Nemhauser Trotter	1	-	4
	Grad <sub>1</sub>	2	-	46
	Grad <sub>1</sub> -Kronenregel	18	11	990
	Kronenregel - Grad <sub>1</sub>	15	9	971

EINFÜGEN

Werden die Reduktionsregeln in Kombination angewandt, zeigen sich sowohl durch verschiedene Kombinationen, als auch durch unterschiedliche Reihenfolge im Ergebnis große Unterschiede.

## 3.4 Interpretation der Ergebnisse

## 3.5 Zusammenfassung

Am Ende sollten ggf. die wichtigsten Ergebnisse nochmal in *einem* kurzen Absatz zusammengefasst werden.

Tabelle 3.4: Anwendung kombinierter Reduktionsregeln

Kombination	Anwendungen <sub>1</sub>	Anwendungen <sub>2</sub>	Anwendungen <sub>3</sub>	Reduktion
K - G <sub>1</sub>	3.63	4.3	-	331.8
G <sub>1</sub> - K	4.37	3.22	-	331.17
K - NT	0.8	0.38	-	68.28
NT - K	0.45	0.56	-	68.6
G <sub>1</sub> - NT	1.33	0.017	-	99.87
NT - G <sub>1</sub>	0.28	1.13	-	99.87
K - G <sub>1</sub> - NT	3.61	4.29	0.11	334.67
K - NT - G <sub>1</sub>	3.6	0.87	3.39	334.83
G <sub>1</sub> - NT - K	4.36	0.12	3.2	334.17
G <sub>1</sub> - K - NT	3.61	3.2	0.65	334.16
NT - K - G <sub>1</sub>	0.39	3.44	4.03	335.2
NT - G <sub>1</sub> - K	0.91	3.42	3.2	334.16

## 4. Entwurf / Konzeption

In diesem Kapitel erfolgt die ausführliche Beschreibung des eigenen Lösungsansatzes. Dabei sollten Lösungsalternativen diskutiert und Entwurfsentscheidungen dargelegt werden.

### 4.1 Abschnitt 1

### 4.2 Abschnitt 2

### 4.3 Zusammenfassung

Am Ende sollten ggf. die wichtigsten Ergebnisse nochmal in *einem* kurzen Absatz zusammengefasst werden.

## 5. Implementierung

Die Regeln wurden in der Programmiersprache *C++* unter Verwendung der Bibliothek *LEDA* [Uhri17] implementiert. Sie wurden bei der Anwendung jeweils solange wiederholt, bis sich am Graphen keine Änderung mehr ergab.

### 5.1 Kronenregel

Beim Austesten der Kronenregel hat sich gezeigt, dass die Auswahl des Matchings  $M_1$  im in Kapitel 2.3 dargestellten Algorithmus das Ergebnis der Reduktion in großem Maße beeinflusst. Wenn also beim Finden von  $M_1$  zunächst Kanten mit Knoten höheren Grades betrachtet werden, können bessere Ergebnisse in der darauf folgenden Reduktion erzielt werden. Daraufhin wurde untersucht, wie die höhergradigen Knoten, beziehungsweise die dazugehörigen Kanten ausgewählt werden müssen. Zum einen (Tabelle 5.1) wurden Kanten betrachtet, bei denen das Auswahlkriterium auf mindestens einen der Knoten zutrifft, auf der anderen Seite Kanten, bei denen beide Knoten die Bedingung erfüllen (Tabelle 5.2). *Größte Anzahl* und *Durchschnittliche Anzahl* ergeben sich jeweils aus der Menge an Knoten eines bestimmten Grades. Bei ersterem werden Knoten, deren Grad im aktuellen Graphen am häufigsten vorkommt bevorzugt. Bei letzterem dementsprechend Knoten, deren Grad im aktuellen Graphen dem Durchschnitt entspricht. Diese Werte werden bei jeder Iteration des Algorithmus neu berechnet und passen sich dadurch während der Laufzeit an den Graphen an.

Generell wird eine bessere (größere) Reduktion mit der Kronenregel erreicht, wenn beim Matching  $M_1$  zunächst Kanten betrachtet werden, bei denen die Einschränkung auf beide Knoten zutrifft. Die Reduktionsmenge bei Knoten mit  $Grad > 2$  erzeugt im Vergleich mit anderen statischen Werten das beste Ergebnis. Dies könnte damit zusammenhängen, dass bei den Graphen, bei denen diese Regel sehr effektiv ist, der durchschnittliche Grad der Knoten 2 ist. Vermutlich erzielt die Bevorzugung des durchschnittlichen Grades das beste Ergebnis, da sich dieser Wert mit jedem Durchlauf verändert.



Tabelle 5.1: 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

*Grad der Knoten* bezieht sich auf die Bedingung für die bevorzugte Auswahl der Kanten für  $M_1$ , bzw. dessen Knoten. *Anwendung* und *Reduktion* stellen jeweils den Durchschnittswert beim gesamten Testset dar.

Tabelle 5.2: 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

*Grad der Knoten* bezieht sich auf die Bedingung für die bevorzugte Auswahl der Kanten für  $M_1$ , bzw. dessen Knoten. *Anwendung* und *Reduktion* stellen jeweils den Durchschnittswert beim gesamten Testset dar.

# 6. Evaluation

Hier erfolgt der Nachweis, dass das in Kapitel 4 entworfene Konzept funktioniert. Leistungsmessungen einer Implementierung werden immer gerne gesehen.

## 6.1 Abschnitt 1

...

## 6.2 Abschnitt 2

...

## 6.3 Zusammenfassung

Am Ende sollten ggf. die wichtigsten Ergebnisse nochmal in *einem* kurzen Absatz zusammengefasst werden.

## 7. Diskussion und Ausblick

(Keine Untergliederung mehr)

# Literaturverzeichnis

- [Nied06] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press. 2006.
- [SeJK11] E. C. Sewell, S. H. Jacobson und H. Kaul. Reductions for the Stable Set Problem. *Algorithmic Operations Research Vol.6 40–55*, 2011.
- [Uhri17] C. Uhrig. The LEDA User Manual, Version 6.5. [http://www.algorithmic-solutions.info/leda\\_manual/MANUAL.html](http://www.algorithmic-solutions.info/leda_manual/MANUAL.html), 2017. Zugriff: 16.02.2018.
- [Ummy00] D. Ummy. *Citation for Dummies*. Oxford University Press. 0000.

# Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich diese Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe. Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher auch nicht veröffentlicht.

Trier, den 2. März 2018