

Polynomial Time Data Reduction for Dominating Set

Xinran Wang ✉

Goethe Universität Frankfurt am Main

Abstract

Ziel dieser Arbeit ist es, eine effektive Methode zur Datenreduktion für das NP-vollständige Problem DOMINATING SET vorzustellen. Die Reduktion erfolgt im Rahmen einer Vorverarbeitung und zeigt sich sowohl theoretisch als auch praktisch als äußerst wirkungsvoll. Durch zwei einfach gehaltene Regeln lässt sich die Größe von Probleminstanzen in Experimenten deutlich verringern. Da diese Regeln auf lokale Strukturen im Graphen angewendet werden, ist ihre Umsetzung unkompliziert. Im Verlauf dieses Berichts werden wir die beiden Regeln ausführlich beschreiben, ihre Laufzeiteigenschaften betrachten und ihre Korrektheit begründen.

2012 ACM Subject Classification Mathematics of computing → Random number generation

Keywords and phrases Dominating Set, Datenreduktion

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Einleitung

Beim Versuch, das Dominating-Set-Problem effizient zu lösen, stößt man schnell auf eine Schwierigkeit: Es handelt sich um ein NP-schweres Problem und, noch konkreter, es ist sogar W[2]-vollständig. Diese Einordnung ist von fundamentaler Bedeutung, da das Dominating-Set-Problem als das kanonische und prototypische Problem für die Komplexitätsklasse W gilt. Die W-Härte impliziert, dass ein sogenannter FPT-Algorithmus (fixed-parameter tractable) mit einer Laufzeit von $f(k) \cdot n^{O(1)}$ für allgemeine Graphen als höchst unwahrscheinlich gilt, was die Suche nach effizienten Lösungen für eingeschränkte Graphklassen, wie die in dieser Arbeit behandelten planaren Graphen, motiviert. [2] Diese hohe Komplexität bedeutet in der Praxis, dass herkömmliche Algorithmen bei diesem Problem in der Regel versagen. Daher stellt sich eine zentrale Frage: *Wie lässt sich ein derart komplexes Problem überhaupt handhabbar machen?*

Das Dominating-Set-Problem ist wie folgt definiert: Gegeben sei ein ungerichteter Graph G . Eine k -Dominating-Set D ist eine Menge von k Knoten, sodass jeder Knoten im Graphen, der nicht in D enthalten ist, mit mindestens einem Knoten in D benachbart ist. Die sogenannte Dominationszahl $\gamma(G)$ ist die kleinste Anzahl von Knoten, die ein dominierendes Menge bilden. Das Dominating-Set-Problem besteht also darin zu entscheiden, ob $\gamma(G)$ höchstens einen gegebenen Wert k beträgt.

Eine sehr effektive Lösung ist das *Preprocessing*, welches auch im Mittelpunkt dieser Arbeit steht. Dabei handelt es sich um eine Vorverarbeitung, die noch vor dem eigentlichen Lösungsverfahren erfolgt und zum Ziel hat, durch gezielte Datenreduktion die Problemgröße zu verkleinern. In der praktischen Anwendung ist Preprocessing oft sehr nützlich. Die zentrale Idee besteht darin, durch einfache Reduktionsregeln die Eingabedaten so zu transformieren, dass der darauf folgende Algorithmus deutlich effizienter arbeiten kann. Ein praxisrelevantes Beispiel hierfür sind mobile Ad-hoc-Netzwerke (MANETs). In diesen wird oft eine zusammenhängende dominierende Menge (Connected Dominating Set) als „virtuelles Rückgrat“ für die Kommunikation genutzt. „A small connected dominating set is used as a backbone for communications, and nodes that are not in this set communicate by passing messages through neighbors that are in the set“. Die Effizienz solcher Netzwerke hängt entscheidend



© Xinran Wang;
licensed under Creative Commons License CC-BY 4.0

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:10

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

von der Fähigkeit ab, eine möglichst kleine dominierende Menge zu finden, was die Bedeutung von Reduktions- und Lösungsalgorithmen unterstreicht. [2]

Die Grundlage unserer Untersuchung ist die Arbeit „*Polynomial Time Data Reduction for Dominating Set*“ von Alber, Fellows und Niedermeier [3]. Die Autoren schlagen darin ein Verfahren vor, das mithilfe von nur zwei sehr intuitiven Reduktionsregeln die Größe des Problems signifikant verringern kann. Besonders hervorzuheben ist, dass dieser Prozess in polynomieller Zeit erfolgt und dabei die Qualität der Lösung garantiert nicht beeinträchtigt wird.

Im Rahmen dieses Seminars möchten wir diese Methode im Detail analysieren und nachvollziehen. Ziel unserer Untersuchung ist es, die zugrunde liegende Logik der beiden Reduktionsregeln zu analysieren und zu verstehen, wie durch die Analyse lokaler Strukturen eine signifikante globale Reduktion erzielt werden kann.

Zur Beantwortung dieser Frage wird zunächst eine formale Einführung in das Dominating-Set-Problem und dessen Komplexität gegeben. Anschließend wird der Kern der vorgestellten Methode behandelt: eine ausführliche Analyse der beiden Reduktionsregeln. Es wird gezeigt, dass die erste Regel auf Einzelknoten basiert, während die zweite auf Paaren von Knoten operiert. Anschließend wird dargelegt, wie die Autoren der Originalquelle beweisen, dass durch diese Modifikationen die Dominationszahl des Graphen unverändert bleibt, also $\gamma(G) = \gamma(G')$. Abschließend wird der daraus resultierende reduzierte Graph vorgestellt und die Effizienz und Relevanz des Algorithmus im praktischen Kontext kurz diskutiert.

2 Grundlagen

In diesem Kapitel werden die grundlegenden graphentheoretischen Definitionen und Notationen eingeführt, die für das Verständnis der nachfolgenden Kapitel notwendig sind. Darauf aufbauend wird das Dominating-Set-Problem formal definiert.

2.1 Graphentheoretische Begriffe

► **Definition 1** (ungerichteter Graph). Ein **ungerichteter Graph** $G = (V, E)$ besteht aus einer endlichen Menge V , die **Knotenmenge** von G genannt wird, und einer Menge

$$E \subseteq \{\{i, j\} : i \in V, j \in V, i \neq j\},$$

die **Kantenmenge** von G genannt wird. Die Elemente aus V heißen **Knoten** von G (auch: "Ecken"; englisch: **vertices**, singular: *vertex*); die Elemente aus E heißen **Kanten** von G (englisch: **edges**, singular: *edge*).

► **Definition 2** (Nachbarschaft und Grad). Sei $G = (V, E)$ ein Graph. Zwei Knoten $i, j \in V$ heißen **adjacent** oder **benachbart**, wenn eine Kante $\{i, j\} \in E$ existiert. Die **offene Nachbarschaft** eines Knotens $i \in V$ ist die Menge aller seiner Nachbarknoten und wird mit $N(i)$ bezeichnet:

$$N(i) = \{j \in V \mid \{i, j\} \in E\}$$

Die **geschlossene Nachbarschaft** eines Knotens $i \in V$ ist die Menge, die i selbst und alle seine Nachbarn enthält und wird mit $N[i]$ bezeichnet:

$$N[i] = N(i) \cup \{i\}$$

Der **Grad** eines Knotens i , bezeichnet mit $\deg(i)$, ist die Anzahl seiner Nachbarn, also $\deg(i) = |N(i)|$.

► **Definition 3** (Paar-Nachbarschaft). Sei $G = (V, E)$ ein Graph und seien $i, j \in V$ zwei verschiedene Knoten.

Die Menge der kombinierten Nachbarn von i und j ist die Menge aller Knoten, die zu mindestens einem der beiden Knoten i oder j adjazent sind. Gemäß der hier verwendeten Quelle wird sie mit $N(i, j)$ bezeichnet:

$$N(i, j) = N(i) \cup N(j)$$

2.2 Das Dominating-Set-Problem

► **Definition 4** (Dominierende Menge). Sei $G = (V, E)$ ein Graph. Eine Teilmenge von Knoten $D \subseteq V$ wird als dominierende Menge (engl. Dominating Set) bezeichnet, wenn jeder Knoten, der nicht in D ist, zu mindestens einem Knoten in D benachbart ist. Formal ausgedrückt: Für jeden Knoten $v \in V \setminus D$ existiert ein Knoten $u \in D$, sodass $\{u, v\} \in E$.

Eine äquivalente, kompaktere Definition lautet: Eine Menge D ist eine dominierende Menge, wenn die Vereinigung der geschlossenen Nachbarschaften aller Knoten in D die gesamte Knotenmenge V ergibt:

$$\bigcup_{u \in D} N[u] = V$$

► **Definition 5** (Dominationszahl). Eine minimale dominierende Menge ist eine dominierende Menge, bei der kein Knoten entfernt werden kann, ohne dass die Menge aufhört zu dominieren. Eine minimum dominierende Menge ist eine dominierende Menge mit der kleinstmöglichen Kardinalität (Anzahl der Knoten). Die Kardinalität einer minimum dominierenden Menge wird Dominationszahl des Graphen G genannt und mit $\gamma(G)$ bezeichnet. Formal:

$$\gamma(G) = \min\{|D| \mid D \text{ ist eine dominierende Menge von } G\}$$

Das Dominating-Set-Entscheidungsproblem besteht darin, für einen gegebenen Graphen G und eine ganze Zahl k zu entscheiden, ob $\gamma(G) \leq k$ gilt.

3 Reduktionsregeln

Im Folgenden werden zwei Reduktionsregeln für das Problem vorgestellt. Beide basieren auf demselben Prinzip: Anstatt den gesamten Graphen zu analysieren, konzentrieren sie sich auf kleine, lokale Strukturen. Diese lokalen Teilbereiche werden anschließend durch einfachere, äquivalente Strukturen ersetzt, indem für die Problemlösung unnötige Elemente entfernt werden.

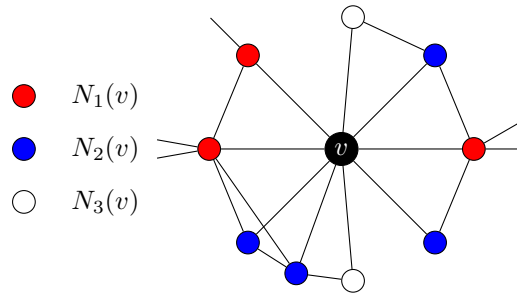
3.1 Die Nachbarschaft eines einzelnen Knotens

Sei $v \in V$ ein Knoten des Graphen $G = (V, E)$. Dessen Nachbarschaft $N(v)$ wird in drei disjunkte Mengen $N_1(v)$, $N_2(v)$ und $N_3(v)$ partitioniert. Die Zuordnung eines Knotens zu einer dieser Mengen hängt von seiner eigenen Nachbarschaftsstruktur ab. Zur genauen Definition wird die geschlossene Nachbarschaft $N[v] := N(v) \cup \{v\}$ herangezogen. Die Partitionierung erfolgt wie folgt:

$$\begin{aligned} N_1(v) &:= \{u \in N(v) : N(u) \setminus N[v] \neq \emptyset\}, \\ N_2(v) &:= \{u \in N(v) \setminus N_1(v) : N(u) \cap N_1(v) \neq \emptyset\}, \\ N_3(v) &:= N(v) \setminus (N_1(v) \cup N_2(v)). \end{aligned}$$

23:4 Polynomial Time Data Reduction for Dominating Set

Folgendes Beispiel visualisiert die Partitionierung:

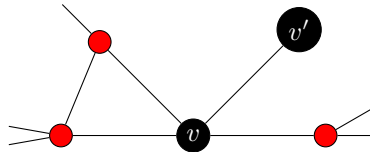


Gemäß der Definition trennt $N_2(v)$ die Mengen $N_1(v)$ und $N_3(v)$, sodass die Knoten in $N_3(v)$ nicht von den Knoten in $N_1(v)$ dominiert werden können. Das bedeutet, dass $N_3(v)$ und $N_1(v)$ nicht verbunden sind. Daher ist eine gute Wahl, dass der Knoten v auszuwählen, um die alle Knoten von $N_3(v)$ zu dominieren. Außerdem, da $N_3(v)$ nicht leer ist, können wir hier folgende Reduktionsregel anwenden:

Reduktionsregel 1:

Die Reduktionsregel wird auf einen Knoten $v \in V$ angewendet, sofern dessen Partition $N_3(v)$ nicht leer ist. In diesem Fall wird der Graph $G = (V, E)$ zu einem neuen Graphen G' transformiert. Die Transformation erfolgt, indem die Knoten der Mengen $N_2(v)$ und $N_3(v)$ aus V entfernt und ein neuer Knoten v' sowie die Kante $\{v, v'\}$ hinzugefügt werden.

Hier ist der aus dem obigen Beispielgraph durch Reduktion erhaltene Graph:



Der Grund, warum ein neuer Knoten v' hinzugefügt werden muss, um zu erzwingen, dass dass v in eine optimale dominante Menge im reduzierten Graphen aufgenommen wird. Wir nennen diesen Knoten „gadget Knoten“.

► **Lemma 6.** Gegeben sei ein Graph $G(V, E)$, und Graph $G'(V', E')$ der Graph, der durch Anwendung der Reduktionsregel 1 entsteht, dann gilt: $\gamma(G) = \gamma(G')$.

Proof. Die Knoten aus $N_3(v)$ können nur von v oder von Knoten aus $N_2(v) \cup N_3(v)$ dominiert werden. Für jedes $w \in N_2(v) \cup N_3(v)$ gilt jedoch $N(w) \subseteq N(v)$. Dies ist offensichtlich, da die Nachbarn der Knoten aus $N_2(v) \cup N_3(v)$ ebenfalls Nachbarn von v sind. Somit werden alle Knoten, die von $N_2(v) \cup N_3(v)$ dominiert werden, auch automatisch von v dominiert. Diese Eigenschaft rechtfertigt die Auswahl von v durch ein **Austauschargument**: Für jede dominierende Menge, die Knoten aus $N_3(v)$ zur Domination von $N_3(v)$ verwendet, können wir eine mindestens genauso gute oder bessere Menge konstruieren, indem wir diese Knoten durch v ersetzen.

Folglich ist es für eine *minimum* dominierende Menge immer eine optimale Entscheidung, den Knoten v auszuwählen, anstatt einer Kombination aus Knoten aus $N_3(v)$. Deshalb kann ohne Beschränkung der Allgemeinheit angenommen werden, dass v in einer *minimum* dominierenden Menge enthalten ist. Die Hinzunahme des *Gadget-Knotens* v' mit der Kante $\{v, v'\}$ erzwingt diese Auswahl im neuen Graphen. Da v die dominierende Rolle für die entfernten Knoten übernimmt, ist das Löschen von $N_2(v)$ und $N_3(v)$ eine korrekte Reduktion. Daraus folgt $\gamma(G') = \gamma(G)$. ◀

► **Lemma 7.** *Die Anwendung von Regel 1 auf einen Graphen $G = (V, E)$ mit n Knoten erfordert eine Laufzeit von $O(n)$, falls G planar ist, und eine Laufzeit von $O(n^3)$ im allgemeinen Fall.*

Proof. Wir analysieren die Laufzeit, indem wir zunächst die Kosten für die Anwendung der Regel auf einen einzelnen Knoten $v \in V$ betrachten und anschließend die Gesamtkosten durch Summation über alle Knoten bestimmen.

Fall 1: Planare Graphen

Für einen gegebenen Knoten $v \in V$ müssen zur Anwendung von Regel 1 die Mengen $N_1(v)$, $N_2(v)$ und $N_3(v)$ konstruiert werden. Dies erfordert die Analyse der Nachbarschaftsbeziehungen innerhalb der geschlossenen Nachbarschaft $N[v] = N(v) \cup \{v\}$. Die relevante Struktur ist also der von $N[v]$ induzierte Teilgraph, $G[N[v]]$.

Die Anzahl der Knoten in diesem Teilgraphen ist $k = |N[v]| = \deg(v) + 1$. Da der ursprüngliche Graph G planar ist, ist auch jeder seiner Teilgraphen, einschließlich $G[N[v]]$, planar. Nach dem Euler-Satz für planare Graphen besitzt ein einfacher planarer Graph mit $k \geq 3$ Knoten höchstens $3k - 6$ Kanten. Die Anzahl der Kanten in $G[N[v]]$ ist somit durch $3(\deg(v) + 1) - 6 = 3\deg(v) - 3$ beschränkt, liegt also in $O(\deg(v))$.

Die Bestimmung der Partition $N_1(v)$, $N_2(v)$, $N_3(v)$ erfordert die Untersuchung aller Knoten und Kanten in $G[N[v]]$. Da sowohl die Anzahl der Knoten als auch die Anzahl der Kanten in diesem Teilgraphen in $O(\deg(v))$ liegt, beträgt die Laufzeit für die Verarbeitung eines einzelnen Knotens v ebenfalls $O(\deg(v))$.

Um die Gesamtlaufzeit zu ermitteln, summieren wir die Kosten über alle Knoten des Graphen:

$$\sum_{v \in V} O(\deg(v)) = O\left(\sum_{v \in V} \deg(v)\right)$$

Nach dem Handschlaglemma gilt $\sum_{v \in V} \deg(v) = 2|E|$. Für einen planaren Graphen mit n Knoten ist die Kantenzahl durch $|E| \leq 3n - 6$ beschränkt, also $|E| \in O(n)$. Daraus folgt, dass die Gesamtlaufzeit in $O(2|E|) = O(|E|) = O(n)$ liegt.

Fall 2: Allgemeine Graphen

Im allgemeinen Fall ist der von $N[v]$ induzierte Teilgraph $G[N[v]]$ nicht notwendigerweise planar und kann dicht sein. Die Anzahl der Knoten ist weiterhin $k = \deg(v) + 1$. Im schlimmsten Fall (wenn $N[v]$ ein Klick ist) kann die Anzahl der Kanten in diesem Teilgraphen in der Größenordnung von $O(k^2) = O(\deg(v)^2)$ liegen. Die Analyse der Nachbarschaftsbeziehungen für einen Knoten v erfordert daher eine Laufzeit von $O(\deg(v)^2)$.

Die Gesamtlaufzeit ergibt sich somit zu:

$$\sum_{v \in V} O((\deg(v))^2).$$

Für jeden Knoten $v \in V$ gilt: $\deg(v) < n$, also

$$(\deg(v))^2 < n^2.$$

Daher ergibt sich für einen einzelnen Knoten die obere Schranke:

$$(\deg(v))^2 = O(n^2).$$

Wenn wir nun in der Gesamtsumme jeden Summanden durch diese obere Schranke ersetzen, erhalten wir:

$$\sum_{v \in V} O((\deg(v))^2) \leq \sum_{v \in V} O(n^2) = O(n^3).$$

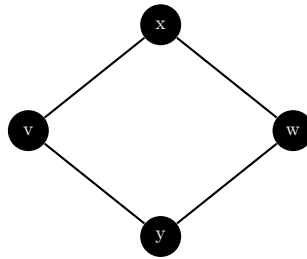
Somit ist die Gesamtlaufzeit für allgemeine Graphen durch $O(n^3)$ beschränkt. ◀

3.2 Die Nachbarschaft eines Knotenpaares

Die Anwendbarkeit von Reduktionsregel 1 ist auf lokale Strukturen beschränkt, die sich um einen einzelnen Knoten zentrieren. In Graphen existieren jedoch häufig komplexere Konfigurationen, deren Vereinfachung die simultane Betrachtung von zwei Knoten erfordert.

Diese Lücke schließt die zweite Reduktionsregel. Sie erweitert den Fokus von einzelnen Knoten auf Knotenpaare (v, w) und deren gemeinsame Nachbarschaftsstruktur. Ihr Ziel ist die Identifikation und Reduktion bestimmter Knotenmengen – hier als $N_3(v, w)$ bezeichnet, die ausschließlich durch die kombinierte Dominanz des Paares (v, w) beeinflusst und von Regel 1 nicht erfasst werden können.

Die Funktionsweise dieser Regel wird am folgenden Beispielgraphen veranschaulicht:



Dieses einfache Rautendiagramm zeigt, warum die zweite Regel notwendig ist, wenn die erste Regel nicht funktioniert. Aus der Perspektive eines einzelnen Knotens (zum Beispiel v) sind seine Nachbarn (x und y) stets durch den Knoten w mit dem äußeren Umfeld verbunden. Daher sind die Mengen N_3 aller Knoten im Graphen leer, was die Anwendung der ersten Regel unmöglich macht. Betrachtet man jedoch eine Knotenpaar (v, w) gemeinsam, so sind deren gemeinsame Nachbarn x und y vollständig im gemeinsamen Nachbarschaftsbereich von v und w eingeschlossen, ohne jegliche Verbindung nach außen. Dadurch bilden x und y eine nicht-leere Menge $N_3(v, w)$, die genau die Voraussetzungen der zweiten Regel für eine Reduktion erfüllt. Dieses Beispiel zeigt, dass die zweite Regel in der Lage ist, komplexere lokale Redundanzen zu erkennen und zu bearbeiten, die aus der Sicht eines einzelnen Knotens nicht entdeckt werden können.

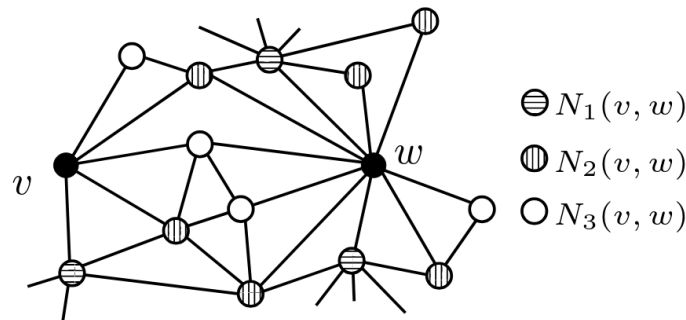
In Analogie zu Regel 1 wird für ein Knotenpaar $(v, w) \in V$ die kombinierte Nachbarschaftsmenge $N(v, w) := N(v) \cup N(w)$ untersucht. Diese Menge wird anschließend in die drei disjunkten Teilmengen $N_1(v, w)$, $N_2(v, w)$ und $N_3(v, w)$ partitioniert. Die Kriterien für die Einteilung basieren auf der erweiterten Nachbarschaft $N[v, w] := N[v] \cup N[w]$, woraus sich folgende Definitionen ergeben:

$$N_1(v, w) := \{u \in N(v, w) : N(u) \setminus N[v, w] \neq \emptyset\},$$

$$N_2(v, w) := \{u \in N(v, w) \setminus N_1(v, w) : N(u) \cap N_1(v, w) \neq \emptyset\},$$

$$N_3(v, w) := N(v, w) \setminus (N_1(v, w) \cup N_2(v, w)).$$

Folgendes Beispiel visualisiert die Partitionierung:



Reduktionsregel 2:

Es gibt vor der Anwendung der zweiten Regel eine wichtige Vorprüfung: Gibt es eine einfache Lösung? Existiert ein Knoten in $N_2(v, w)$ oder $N_3(v, w)$, der allein in der Lage ist, ganz $N_3(v, w)$ zu dominieren ($N_3(v, w) \neq \emptyset$)? Falls eine solche „Einzelpunktlösung“ nicht existiert, wird die zweite Regel aktiviert und ausgeführt.

Fall 1 Wenn $N_3(v, w)$ von einem einzelnen Knoten aus $\{v, w\}$ dominiert werden kann:

(1.1) Wenn $N_3(v, w) \subseteq N(v)$ und zugleich $N_3(v, w) \subseteq N(w)$ gilt:

- Entferne $N_3(v, w)$ und $N_2(v, w) \cap N(v) \cap N(w)$ aus G .
- Füge zwei neue Knoten z, z' und die Kanten $\{v, z\}, \{w, z\}, \{v, z'\}, \{w, z'\}$ zu G hinzu.

(1.2) Wenn $N_3(v, w) \subseteq N(v)$, aber nicht $N_3(v, w) \subseteq N(w)$ gilt:

- Entferne $N_3(v, w)$ und $N_2(v, w) \cap N(v)$ aus G .
- Füge einen neuen Knoten v' und die Kante $\{v, v'\}$ zu G hinzu.

(1.3) Wenn $N_3(v, w) \subseteq N(w)$, aber nicht $N_3(v, w) \subseteq N(v)$ gilt:

- Entferne $N_3(v, w)$ und $N_2(v, w) \cap N(w)$ aus G .
- Füge einen neuen Knoten w' und die Kante $\{w, w'\}$ zu G hinzu.

Fall 2 Wenn $N_3(v, w)$ nicht von einem einzelnen Knoten aus $\{v, w\}$ dominiert werden kann:

- Entferne $N_3(v, w)$ und $N_2(v, w)$ aus G .
- Füge zwei neue Knoten v', w' und die Kanten $\{v, v'\}, \{w, w'\}$ zu G hinzu.

► **Lemma 8.** Sei $G = (V, E)$ ein Graph und sei $G' = (V', E')$ der Graph, der nach der Anwendung von Regel 2 auf G resultiert. Dann gilt $\gamma(G) = \gamma(G')$.

Proof. Die zentrale Aufgabe besteht hier darin, dass man $N_3(v, w)$ dominieren muss, da diese keine Verbindungen nach außen hat. Das bedeutet, dass man zur Dominierung von $N_3(v, w)$ die dominierenden Knoten nur aus einer sehr begrenzten lokalen Menge auswählen kann, nämlich aus:

$$M := \{v, w\} \cup N_2(v, w) \cup N_3(v, w).$$

Nun analysieren wir verschiedene Fälle:

Fall (1.1): Alle Knoten aus der Menge $N_3(v, w)$ sind sowohl Nachbarn von v als auch von w . Dies bedeutet, dass die Wahl von v oder w ausreicht, um die gesamte Menge zu dominieren, weshalb sie eine strategisch zentrale Position einnehmen. Es ist immer optimal, v oder w zu nehmen, da für alle Knotenpaare $x, y \in M$ gilt: $N(x, y) \subseteq N(v, w)$. Das Problem in dieser

Situation ist jedoch die unklare Entscheidung, welcher der beiden Knoten gewählt werden soll.

Der Lösungsansatz hierfür ist eine spezielle **Gadget-Konstruktion**. Durch das Hinzufügen der Knoten z und z' und der entsprechenden Kanten wird eine **logische Disjunktion (OR)** modelliert. Die Konstruktion erzwingt, dass mindestens einer der Knoten aus $\{v, w\}$ in die dominierende Menge aufgenommen werden muss, um z und z' zu dominieren.

Die Sicherheit der Entfernung von $N_3(v, w)$ und den mit v oder w verbundenen N_2 -Knoten ist dabei gewährleistet. Durch die erzwungene Wahl von v oder w werden die Knoten in $N_3(v, w)$ sicher dominiert und die mit v oder w verbundenen Knoten aus $N_2(v, w)$ ebenfalls. Die Größe der dominierenden Menge ändert sich dabei nicht.

Fall (1.2): Falls v die Menge $N_3(v, w)$ dominiert, aber w dies nicht tut, wird v in die Dominating Set aufgenommen. v ist immer optimal, da die Auswahl von v (möglicherweise zusammen mit w) mindestens so viele Knoten dominiert wie jede beliebige andere Kombination von zwei Knoten x, y aus der lokalen Menge $M \setminus \{v\}$. Der neu eingefügte Knoten v' und die Kante $\{v, v'\}$ stellt sicher, dass in der optimalen Lösung v enthalten sein muss. Gleichzeitig behalten wir die Möglichkeit bei, den Knoten w später noch auszuwählen, da dieser eventuell notwendig ist, um eine minimale dominierende Menge zu erreichen. Wie im Fall (1.1) ist es auch hier sicher, die Knoten aus $N_3(v, w)$ sowie aus $N_2(v, w) \cap N(v)$ aus dem Graphen zu entfernen. Denn alle diese Knoten sind mit v verbunden und somit bereits dominiert.

Fall (1.3): Symmetrische Analyse von Fall (1.2).

Fall (2): Dieser Fall tritt ein, wenn die Menge $N_3(v, w)$ nicht von v oder w individuell dominiert werden kann. Hier benötigen mindestens 2 Knoten, um diese Menge zu dominieren. Für alle Knotenpaare $x, y \in M$ gilt: $N(x, y) \subseteq N(v, w)$. Das Gadget mit den Kanten $\{v, v'\}$ und $\{w, w'\}$ erzwingt die Aufnahme beider Knoten v und w in die dominierende Menge, da nur v und w die Knoten v' bzw. w' dominieren können. Anschließend erfolgt die Entfernung der Menge $N_3(v, w)$ und $N_2(v, w)$. Die Entfernung ist sicher, da alle Knoten in diesen Mengen durch die festgelegte Auswahl von v und w bereits dominiert sind. Die Größe von der dominierenden Menge ändert dabei nicht. ◀

► **Lemma 9.** Die Anwendung von Regel 2 auf einen Graphen $G = (V, E)$ mit n Knoten erfordert eine Laufzeit von $O(n^2)$, falls G planar ist, und eine Laufzeit von $O(n^4)$ im allgemeinen Fall.

Proof. Wir analysieren die Laufzeit, indem wir zunächst die Kosten für die Anwendung der Regel auf ein einzelnes Knotenpaar $(v, w) \in V \times V$ betrachten und anschließend die Gesamtkosten durch Summation über alle Paare bestimmen.

Fall 1: Planare Graphen

Für ein gegebenes Knotenpaar (v, w) müssen zur Anwendung von Regel 2 die Mengen $N_1(v, w)$, $N_2(v, w)$ und $N_3(v, w)$ konstruiert werden. Dies erfordert die Analyse der Nachbarschaftsbeziehungen innerhalb der kombinierten geschlossenen Nachbarschaft $N[v, w] = N[v] \cup N[w]$. Die relevante Struktur ist somit der von $N[v, w]$ induzierte Teilgraph, $G[N[v, w]]$.

Die Anzahl der Knoten in diesem Teilgraphen ist $k = |N[v, w]| \leq |N[v]| + |N[w]| = (\deg(v) + 1) + (\deg(w) + 1)$, liegt also in $O(\deg(v) + \deg(w))$. Da der ursprüngliche Graph G planar ist, ist auch $G[N[v, w]]$ planar. Die Anzahl der Kanten in diesem Teilgraphen ist nach dem Euler-Satz linear zur Anzahl seiner Knoten, also ebenfalls in $O(\deg(v) + \deg(w))$.

Die Bestimmung der Partition erfordert die Untersuchung aller Knoten und Kanten in $G[N[v, w]]$. Da sowohl die Knoten- als auch die Kantenzahl in $O(\deg(v) + \deg(w))$ liegt,

beträgt die Laufzeit für die Verarbeitung eines einzelnen Paares (v, w) ebenfalls $O(\deg(v) + \deg(w))$.

Um die Gesamtlaufzeit zu ermitteln, summieren wir die Kosten über alle $O(n^2)$ Knotenpaare des Graphen:

$$\sum_{v, w \in V} O(\deg(v) + \deg(w)) = O\left(\sum_{v \in V} \left(\sum_{w \in V} \deg(v) + \sum_{w \in V} \deg(w)\right)\right)$$

Die innere Summe $\sum_{w \in V} \deg(w)$ ist nach dem Handschlaglemma $2|E|$, was für planare Graphen in $O(n)$ liegt. Die Summe $\sum_{w \in V} \deg(v)$ entspricht $n \cdot \deg(v)$. Daraus ergibt sich:

$$O\left(\sum_{v \in V} (n \cdot \deg(v) + O(n))\right) = O\left(n \sum_{v \in V} \deg(v) + n \cdot O(n)\right) = O(n \cdot O(n) + O(n^2)) = O(n^2)$$

Somit liegt die Gesamtlaufzeit für planare Graphen in $O(n^2)$.

Fall 2: Allgemeine Graphen

Im allgemeinen Fall ist der von $N[v, w]$ induzierte Teilgraph nicht notwendigerweise planar und kann dicht sein. Die Anzahl der Knoten ist weiterhin $k \in O(\deg(v) + \deg(w))$. Im schlimmsten Fall kann die Anzahl der Kanten in diesem Teilgraphen in der Größenordnung von $O(k^2) = O((\deg(v) + \deg(w))^2)$ liegen. Die Analyse für ein Paar (v, w) erfordert daher eine Laufzeit von $O((\deg(v) + \deg(w))^2)$.

Die Gesamtlaufzeit ergibt sich somit aus der Summe über alle Paare:

$$\sum_{v, w \in V} O((\deg(v) + \deg(w))^2)$$

Da für jeden Knoten $\deg(v) < n$ gilt, ist die obere Schranke für einen einzelnen Summanden $O((n + n)^2) = O(n^2)$. Wenn wir nun in der Gesamtsumme, die über $O(n^2)$ Paare läuft, jeden Summanden durch diese Schranke ersetzen, erhalten wir:

$$\sum_{v, w \in V} O(n^2) = O(n^2) \cdot O(n^2) = O(n^4)$$

Daraus folgt, dass die Gesamtlaufzeit im allgemeinen Fall in $O(n^4)$ liegt. ◀

3.3 Reduzierte Graphen und ihre Eigenschaften

Der Prozess der Datenreduktion wird fortgesetzt, indem die Reduktionsregeln 1 und 2 wiederholt auf den Graphen angewendet werden. Dieser Prozess endet, wenn ein Zustand erreicht ist, in dem keine der beiden Regeln mehr eine Veränderung am Graphen bewirkt. Ein solcher Graph wird als „reduziert“ bezeichnet.

► **Definition 10** (Reduzierter Graph). *Ein Graph $G = (V, E)$ wird als **reduziert** bezüglich der Regeln 1 und 2 bezeichnet, wenn die Anwendung beider Regeln den Graphen unverändert lässt.*

Eigenschaften eines reduzierten Graphen

Ein Graph, der diesen Zustand erreicht hat, weist spezifische strukturelle Eigenschaften auf, da die Bedingungen für die Anwendung der Regeln nicht mehr erfüllt sind:

- Für jeden Knoten $v \in V$ ist die Menge $N_3(v)$ leer. Die einzige Ausnahme ist das mögliche Vorhandensein eines einzelnen „Gadget“-Knotens mit Grad eins, der durch eine frühere Regelanwendung entstanden ist.
- Für jedes Knotenpaar (v, w) existiert ein einzelner Knoten in der Menge $N_2(v, w) \cup N_3(v, w)$, der die gesamte Menge $N_3(v, w)$ dominiert. Andernfalls wäre Regel 2 anwendbar gewesen.

Der Reduktionsprozess und seine Komplexität

Jede erfolgreiche Anwendung einer Reduktionsregel verkleinert den Graphen, indem sie Knoten entfernt. Da die Anzahl der Knoten endlich ist, kann es nur eine lineare Anzahl von $O(n)$ erfolgreichen Regelanwendungen geben, was garantiert, dass der Prozess terminiert. Die Gesamtlaufzeit, um einen Graphen in einen reduzierten Zustand zu überführen, wird im folgenden Theorem zusammengefasst.

► **Theorem 11** (Komplexität der Transformation). *Ein gegebener Graph G kann in einen reduzierten Graphen G' mit identischer Dominationszahl ($\gamma(G) = \gamma(G')$) transformiert werden. Die Laufzeit hierfür beträgt $O(n^3)$ für planare Graphen und $O(n^5)$ für allgemeine Graphen.*

Anmerkungen zur Implementierung

Die in den Beweisen der Lemmata vorgestellten Algorithmen sind einfach umzusetzen und zeigen in der Praxis eine gute Leistung. Aus rein theoretischer Sicht gäbe es über das Konzept der Baumzerlegung (Tree Decompositions) potenziell asymptotisch schnellere Algorithmen, um eine Reduktion zu erreichen [1]. Die grundlegende Idee einer Baumzerlegung ist es, einen allgemeinen, potenziell dicht vernetzten Graphen in eine übergeordnete Baumstruktur zu transformieren. [5] Viele Probleme, die auf allgemeinen Graphen NP-schwer sind, können auf Bäumen sehr effizient gelöst werden. Die Baumzerlegung ermöglicht es, diese Effizienz auf Graphen mit kleiner Baumweite zu übertragen. Mithilfe von dynamischer Programmierung arbeitet sich ein Algorithmus entlang der Baumstruktur von den Blättern zur Wurzel vor und löst das Problem schrittweise. Die Bags fungieren dabei als kleine „Schnittstellen“ fester Größe, die die Komplexität der Berechnung eingrenzen. Für einen Graphen mit n Knoten und kleiner Baumweite k resultiert dies oft in einer Laufzeit von der Form $f(k) \cdot \text{poly}(n)$. Das bedeutet, der Algorithmus ist linear in der Größe des Graphen und somit für große Graphen mit kleiner Baumweite sehr schnell. [4] Diese Ansätze sind jedoch aufgrund der hohen involvierten Konstanten und des Implementierungsaufwands für die Praxis weniger relevant. Die hier vorgestellten Regeln sind hingegen praxistauglicher, da sie den Graphen nur lokal verändern (in einer Umgebung von Knoten, deren Abstand höchstens fünf beträgt).

References

- 1 H. L. Bodlaender. Personal communication. 2002.
- 2 Serge Gaspers. 8. parameterized intractability: the w-hierarchy comp6741: Parameterized and exact computation. 2015.
- 3 Rolf Niedermeier Jochen Alber, Michael R. Fellows. Polynomial time data reduction for dominating set. 2002.
- 4 Frank Kammer. Treelike and chordal graphs: Algorithms and generalizations. 2010.
- 5 Philip Zucker. Treeifying graphs: Tree decomposition | hey there buddo! 2025.