

Außen 1-gap planare Graphen

Bachelorarbeit
von

Daniel Berreiter

An der Fakultät für Informatik und Mathematik
Lehrstuhl für theoretische Informatik



Gutachter: Prof. Dr. Ignaz Rutter
Betreuer: Peter Stumpf, M.Sc.

Bearbeitungszeit: 18.11.2018 – 18.02.2019

Selbstständigkeitserklärung

Hiermit versichere ich, dass diese Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden.

Passau, 15. Februar 2019

Zusammenfassung

Diese Arbeit behandelt außen 1-gap planare Graphen. Dabei handelt es sich um Graphen, bei denen jede Kante eine Lücke haben kann, durch die eine andere Kante kreuzen kann. Eine Kante kann dabei durch beliebig viele Lücken anderer Kanten führen. Wir kommen zu den Ergebnissen, dass ein solcher Graph mit $n \geq 2$ Knoten maximal $3n - 5$ Kanten hat. Außerdem zeigen wir, dass es für jedes n einen Graphen mit mindestens $3n - 6$ Kanten gibt. Die minimale Dichte eines Graphen, um eine Einbettung zu finden, die nicht um neue Kanten erweitert werden kann, liegt bei $2n$ Kanten. Zudem zeigen wir ein ganzzahliges lineares Programm zum Erkennen und Erweitern von außen 1-gap planaren Graphen.

Inhaltsverzeichnis

1	Einführung	1
1.1	Verwandte Arbeiten	2
1.2	Überblick	3
2	Grundlagen	5
3	Dichte	7
3.1	Maximale Dichte	7
3.2	Minimale Dichte	8
4	Lösung mittels Gurobi	19
4.1	Aufbau des Programms	19
4.2	Auswertung	21
5	Zusammenfassung	27
	Literaturverzeichnis	29

1. Einführung

In dieser Arbeit untersuchen wir außen 1-gap planare Graphen und deren Eigenschaften. Es gibt verschiedene Arten von Graphen, die sich je nach Komplexität besser oder weniger gut zeichnen lassen. Planare Graphen sind dabei von Vorteil, da es bei ihnen möglich ist, den Graphen ohne Kreuzungen zwischen den Kanten darzustellen. Komplexere Graphen mit vielen Kreuzungen erschweren es den Graphen zu lesen, da das Verfolgen einzelner Kanten durch die Kreuzungen schwieriger wird. Grafik 1.1 ist eine Darstellung eines Graphen mit vielen Kreuzungen. Wenn man nun an jeder Kreuzung eine der beiden Kanten aufteilt, entsteht eine Zeichnung wie in 1.2.

Um die Platzierung dieser Lücken genauer zu definieren, ist festgelegt, dass jede Kante nur eine bestimmte Anzahl an Lücken haben darf. Bei solchen k -gap planaren Graphen kann jede Kante maximal k Lücken aufweisen [BBC⁺18]. Für diese Arbeit betrachten wir eine Subklasse dieser Graphen. Bei planaren Graphen gibt es die außenplanaren Graphen, die eine Einbettung besitzen, bei der alle Knoten am Rand einer von Kanten ununterbrochenen Fläche liegen. Wenn man dies auf k -gap planare Graphen anwendet erhält man außen k -gap planare Graphen, wobei wir uns auf außen 1-gap planare Graphen beschränken, bei denen jede Kante nur eine Lücke aufweisen kann. Wir werden verschiedene Eigenschaften zur Dichte solcher Graphen ermitteln, um ein besseres Verständnis für diese spezifische Graphenklasse und ihre Grenzen zu erhalten. Neben der maximalen Dichte betrachten wir

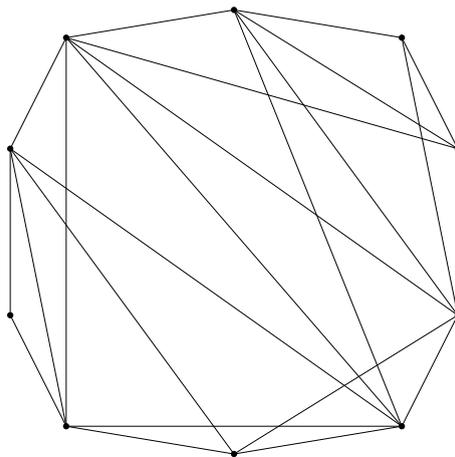


Abbildung 1.1: Zeichnung eines Graphen mit vielen Kreuzungen.

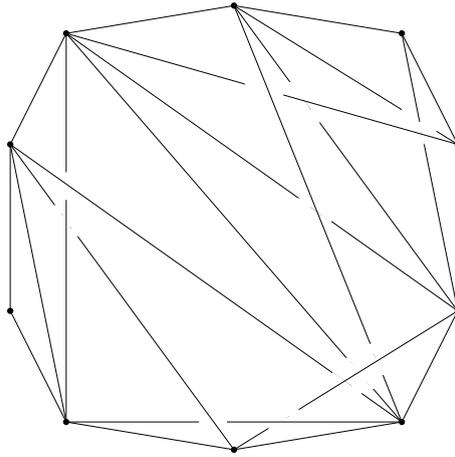


Abbildung 1.2: Zeichnung eines Graphen mit Lücken anstelle von Kreuzungen.

auch die minimale Dichte, bei der eine Einbettung maximal sein kann, also keine weiteren Kanten mehr hinzugefügt werden können. Außerdem erstellen wir ein ganzzahliges lineares Programm, mit Hilfe dessen wir für einen gegebenen Graphen ermitteln können, ob es sich um einen außen k -gap planaren Graphen handelt, und wie er erweitert werden kann, um eine maximale Kantenanzahl zu erreichen. Dieses Programm hat uns außerdem dabei geholfen, unsere theoretischen Ergebnisse zu verifizieren.

1.1 Verwandte Arbeiten

Die wichtigste Arbeit, die für uns relevant ist, wurde verfasst von Bae et al. [BBC⁺18]. In dieser Arbeit werden die Graphen mit Lücken an Kreuzungen zuerst eingeführt. Diese Betrachtung wird durch eine erleichterte Art der Betrachtung von Graphen begründet, da das Verfolgen einzelner Kanten nicht durch Kreuzungen verkompliziert wird. Es wurden Graphen betrachtet, bei denen die Anzahl der Lücken einer Kante durch die Variable k begrenzt wurden. Es handelt sich also um k -gap planare Graphen. Die Autoren finden für 1-gap planare Graphen eine obere Grenze von $5n - 10$ Kanten, bei n Knoten. Diese Grenze ist gültig für alle Graphen mit mindestens 20 Knoten. Es wurde ermittelt, dass ein vollständiger Graph K_n genau dann 1-gap planar ist, wenn $n \leq 8$ gilt. Die Arbeit vergleicht k -gap planare Graphen auch zu anderen beyond-planaren Graphklassen, bei denen nicht planare Graphen gezeichnet werden, indem bestimmte Kreuzungskonfigurationen vermieden werden, oder bestimmte Eigenschaften für die Kreuzungen garantiert werden. Beispiele für solche Klassen sind k -planare Graphen, bei denen jede Kante maximal k Kreuzungen hat, oder k -quasiplanare Graphen, bei denen ein Graph maximal k paarweise gekreuzte Kanten haben darf. Weitere Ergebnisse der Arbeit beschreiben die Beziehungen zwischen k -gap planaren Graphen und solchen beyond-planaren Graphen. Unter anderem wird gezeigt, dass für $k \geq 1$ die Klasse von $2k$ -planaren Graphen in k -gap planaren Graphen enthalten ist, welche wiederum in den $(2k + 2)$ -quasiplanaren Graphen enthalten ist. Die Arbeit von Bae et al. zeigt außerdem, dass das Problem der Erkennung von 1-gap planaren Graphen NP-schwer ist.

Auer et al. beschäftigen sich ebenfalls mit ähnlichen Graphen [ABB⁺16]. Sie betrachten außen 1-planare Graphen, also Graphen, bei denen alle Knoten entlang der äußeren Fläche liegen, und jede Kante genau eine Kreuzung haben kann. Diese Graphen haben eine maximale Dichte von $5/2n - 4$ Kanten bei einem Graphen mit n Knoten. Für jedes gerade $n \geq 2$ gibt es außen 1-planare Graphen mit $5/2n - 4$ Kanten. Die Arbeit beinhaltet eine Konstruktionsvorschrift um einen solchen Graphen zu erzeugen. Die Autoren haben sich außerdem mit der Erkennung solcher Graphen beschäftigt und einen Algorithmus gefunden,

der außen 1-gap planare Graphen in linearer Zeit erkennen kann, und diese auf maximale Graphen erweitern kann.

Da die Dichte von außen 1-gap planaren Graphen einen großen Teil dieser Arbeit ausmacht, sind auch die Ergebnisse von Brandenburg et al. relevant, die sich auf die Dichte in 1-planaren Graphen beziehen [BEG⁺13]. Diese Arbeit zeigt, dass 1-planare Graphen eine Obergrenze von $4n - 8$ Kanten haben. Weitere Ergebnisse zeigen, dass für bestimmte Einbettungen von 1-gap planaren Graphen die maximale Dichte bis auf $2.1n - O(1)$ gesenkt werden kann.

1.2 Überblick

Zuerst erläutern wir in Kapitel 2 die Grundlagen, die wir im weiteren Verlauf benötigen. Daraufhin beginnen wir unsere Untersuchungen zur Dichte der Graphen. In Kapitel 3.1 Betrachten wir die maximale Dichte in außen 1-gap planaren Graphen. Anschließend zeigen wir die minimale Dichte in einem kanten-maximalen Graphen in Kapitel 3.2. Zudem beschreiben wir ein ganzzahliges lineares Programm zur Erkennung und Erweiterung von außen 1-gap planaren Graphen in Kapitel 4 und werten die Ergebnisse dieses Programmes aus. Diese Auswertung findet in Kapitel 4.2 statt.

2. Grundlagen

In diesem Kapitel werden die für diese Arbeit notwendigen Begriffe und Konzepte eingeführt. Wir arbeiten hier mit Graphen und auch Zeichnungen dieser Graphen. Als Zeichnung bezeichnen wir die Abbildung einer festen Einbettung eines Graphen. Die Zeichnung eines Graphen besteht aus Punkten, die die Knoten darstellen, und Linien, die den Kanten entsprechen. Dabei ist zu beachten, dass zwei Kanten sich zwar Kreuzen können, aber nicht übereinander verlaufen dürfen. Außerdem darf eine Kante nicht durch einen Knoten verlaufen.

Wir verwenden den Begriff 'vollständig' im Verlauf der Arbeit sowohl für Graphen als auch für Zeichnungen. Im Falle von Graphen bedeutet vollständig, dass in den Graphen keine weiteren Kanten eingefügt werden können, ohne die Graphklasse zu verlassen. Einige wichtige vollständige Graphen in dieser Arbeit sind K_4 und K_5 . Im Bezug auf Zeichnungen bedeutet der Begriff vollständig, dass in diese Zeichnung keine weiteren Kanten eingezeichnet werden können, ohne die entsprechenden Voraussetzungen für die Zeichnung zu verletzen. Dabei ist es durchaus möglich, dass eine Zeichnung vollständig ist, der zugehörige Graph aber nicht vollständig ist, und weitere Einbettungen mit nicht vollständigen Zeichnungen besitzt.

Jeder außen 1-gap planare Graph, besteht aus einem Ring an Knoten, der entlang der äußeren Fläche verläuft. Wir bezeichnen hierbei die beiden Knoten, die entlang dieser äußeren Fläche direkt mit einem Knoten v_1 verbunde

3. Dichte

In diesem Kapitel betrachten wir die Dichte von außen 1-gap planaren Graphen. Dabei ermitteln wir Eigenschaften bezüglich der Anzahl möglicher Kanten in einem Graph, basierend auf der Anzahl der Knoten des Graphen.

3.1 Maximale Dichte

Zuerst betrachten wir die obere Grenze an möglichen Kanten in außen 1-gap planaren Graphen.

Theorem 3.1. *Ein außen 1-gap planarer Graph mit $n \geq 2$ Knoten hat maximal $3n - 5$ Kanten.*

Beweis. Sei G ein kanten-maximaler außen 1-gap planarer Graph mit n Knoten und m Kanten. Betrachte nun den Graphen G' , den wir erhalten, indem wir die Kanten, die nicht entlang der äußeren Facette des Graphen liegen, nach außen kopieren (siehe Abbildung 3.1). Hierbei entstehen Multikanten, bei denen zwischen einem Kantenpaar zwischen zwei Knoten immer ein anderer Knoten liegt. Dadurch gilt laut Bae et al. [BBC⁺18], dass G' ein 1-gap planarer Graph mit maximal $5n - 10$ Kanten ist. Der Graph G' besteht nun aus $m + (m - n)$ Kanten, da nur die n Kanten entlang des äußeren Rings des Graphen G nicht kopiert werden. Diese Grenze für 1-gap planare Graphen wird ab einem Wert von $m \geq 3n - 4$ überschritten, da $M + (m - n) \leq 5n - 10$ äquivalent ist zu $m \leq 3n - 5$. Bei einer Kantenzahl von $m = 3n - 5$ wäre die Grenze hingegen noch nicht überschritten, da G' dann $3n - 5 + (3n - 5 - n) = 5n - 10$ Kanten hat und damit ein 1-gap planarer Graph sein kann. \square

Theorem 3.2. *Es gibt für jedes n einen außen 1-gap planaren Graphen mit $3n - 6$ Kanten. Wenn gilt $n \equiv 2 \pmod{3}$, kann diese Grenze auf $3n - 5$ Kanten verbessert werden.*

Beweis. Dies ergibt sich durch die folgende Konstruktion: Man beginnt die Zeichnung mit zwei Knoten. Man fügt weitere Knoten in Gruppen von drei Knoten ein. Wähle zwei vorher bereits existierende Knoten, die entlang der äußeren Fläche des Graphen benachbart sind und nenne sie v_1 und v_2 . Die neuen Knoten, v_3 , v_4 und v_5 , werden in die äußere Fläche des Graphen gezeichnet. Verbinde diese Knoten wie in Abbildung 3.2 zu sehen. Wenn nur noch zwei Knoten v_3 und v_4 einzuzichnen sind, fügt man diese in die äußere Fläche ein,

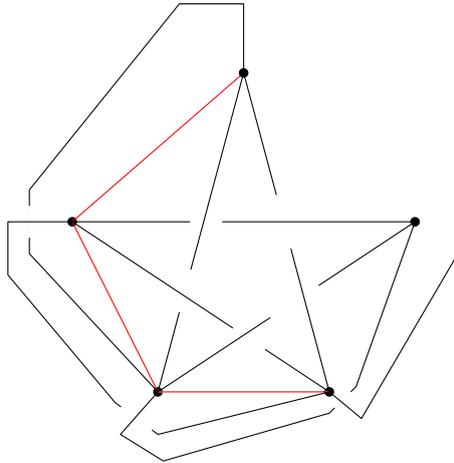


Abbildung 3.1: Graph, bei dem die inneren Kanten nach außen kopiert wurden. Der äußere Kreis des bisherigen Graphen ist in rot dargestellt.

und verbindet sie zu zwei vorher existierenden benachbarten Knoten v_1 und v_2 zu einer K_4 Struktur, bei der die Kante v_1v_3 eine Lücke aufweist um die Kante v_2v_4 hindurchzulassen. Wenn nur noch ein Knoten einzuzeichnen ist, fügt man diesen in die äußere Fläche des Graphen ein und verbindet ihn zu zwei vorher benachbarten Knoten.

Demzufolge lässt sich die Anzahl der Kanten abhängig von der Knotenanzahl berechnen.

Fall 1: Es gilt $n \equiv 0 \pmod{3}$. In diesem Fall errechnet sich die Anzahl an Kanten wie folgt:
 $m = 1 + (n - 3) \cdot 3 + 2 = 3n - 6$.

Fall 2: Es gilt $n \equiv 1 \pmod{3}$. In diesem Fall errechnet sich die Anzahl an Kanten wie folgt:
 $m = 1 + (n - 4) \cdot 3 + 5 = 3n - 6$.

Fall 3: Es gilt $n \equiv 2 \pmod{3}$. In diesem Fall errechnet sich die Anzahl an Kanten mittels
 $m = 1 + (n - 2) \cdot 3 = 3n - 5$.

Der so entstandene Graph hat also nun mindestens $3n - 6$ Kanten. Wenn bei dem Graphen $n \equiv 2 \pmod{3}$ gilt, dann hat der konstruierte Graph das Maximum von $3n - 5$ Kanten. Dies lässt sich berechnen, da beim Einfügen einer Gruppe von drei Knoten neun neue Kanten eingefügt werden, beim Einfügen von zwei Knoten werden fünf neue Kanten eingefügt, und beim Einfügen von einem Knoten werden zwei neue Kanten eingezeichnet. Außerdem gibt es eine Kante, die bereits mit den ersten beiden Knoten existiert. Wenn nun $n \equiv 2 \pmod{3}$ gilt, gibt es also $1 + (n - 2)/3 \cdot 9 = 3n - 5$ Kanten. Für $n \equiv 1 \pmod{3}$ gibt es $1 + (n - 4)/3 \cdot 9 + 5 = 3n - 6$ Kanten und für $n \equiv 0 \pmod{3}$ gibt es $1 + (n - 3)/3 \cdot 9 + 2 = 3n - 6$ Kanten. \square

3.2 Minimale Dichte

In diesem Kapitel betrachten wir Graphen mit einer vollständigen Einbettung, also eine Zeichnung, bei der keine weiteren Kanten eingezeichnet werden können. Wir wollen beweisen, dass ein Graph mindestens $2n$ Kanten haben muss, um eine solche Einbettung zu besitzen.

Der Durchschnittsgrad des Graphen lässt sich aus der Anzahl der Knoten n und Anzahl der Kanten m berechnen als $2m/n$. Um weniger als $2n$ Kanten zu erreichen, muss der Durchschnittsgrad also unter 4 liegen. Jeder maximale außen 1-gap plane Graph besteht mindestens aus den n Kanten, die die Knoten entlang der äußeren Fläche verbinden. Auf diesem Kreis hat somit jeder Knoten zwei Nachbarn. Da auf diesem äußeren Kreis keine Kante außen an einem Knoten vorbei führen kann, sind alle Kanten darauf unterbrechungsfrei.

Um diesen Beweis zu führen, nehmen wir an, dass es mindestens einen Graphen mit weniger als $2n$ Kanten gibt, der eine vollständige Einbettung besitzt. Demzufolge gibt es einen

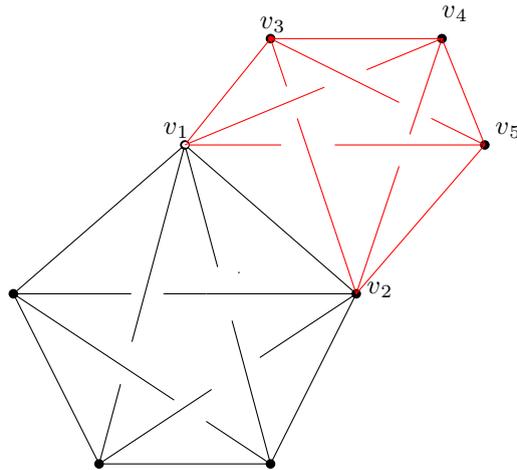


Abbildung 3.2: Graph bei dem drei weitere Knoten eingefügt wurden. Die roten Kanten sind die neuen Verbindungen.

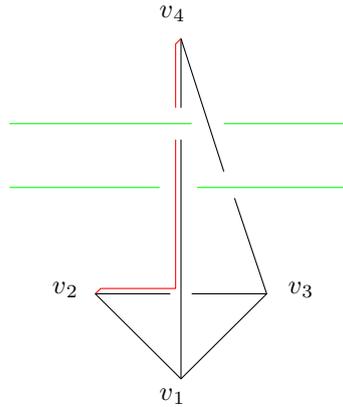
minimalen Graphen, für den es eine solche Einbettung gibt. Im Verlauf des Beweises zeigen wir, dass dieser Graph entweder nicht minimal ist, also dass ein kleinerer Graph mit denselben Eigenschaften existiert, oder dass der Graph unsere Voraussetzungen nicht erfüllt. Um den Beweis durchzuführen zeigen wir zuerst einige Lemmas auf die wir dann zurückgreifen können. Der minimale Graph wird im Folgenden mit G bezeichnet. Im Verlauf dieses Beweises werden wir, wenn wir neue Kanten in den Graphen einfügen um die Maximalität zu widerlegen, keine Lücken in den bestehenden Graphen einfügen oder bestehende Lücken verändern, sondern nur die Lücken der neuen Kanten verwenden. Der Beweis ist somit unabhängig davon, ob existierende Lücken verändert werden müssen, da unser Ergebnis stärker ist, als wenn wir andere Lücken ändern würden.

Lemma 3.3. *Jeder Knoten in G muss mindestens Grad 3 aufweisen.*

Beweis. Angenommen, es gibt in G einen Knoten v mit Grad 0 oder 1. Dies widerspricht der Bedingung auf Maximalität von G , da jeder Knoten mindestens zwei Nachbarn besitzt. Angenommen, es gibt in G einen Knoten v mit Grad 2. Der Knoten v hat dann nur Kanten zu seinen Nachbarknoten v_1 und v_2 . Aufgrund der Maximalität von G muss es dann die Kante v_1v_2 geben, da diese eingefügt werden kann, indem man von v_1 parallel zur Kante vv_1 in einem minimalen Abstand ε eine neue Kante einfügt. Wenn man v erreicht, führt man die Kante knapp an v vorbei, und folgt dann der Kante vv_2 . Da sowohl vv_1 als auch vv_2 unterbrechungsfrei sind, und an v keine weiteren Kanten anliegen, ist auch v_1v_2 unterbrechungsfrei und kann immer eingefügt werden. Wenn man den Knoten v und die Kanten vv_1 und vv_2 aus G entfernt, erhält man den Graphen G' . Dieser Graph hat nun einen Knoten und zwei Kanten weniger als G . Der Graph G' besteht aus n' Knoten und m' Kanten. Wenn $m < 2n$ für G gilt, dann gilt für G' auch $m' < 2n'$, wie die folgende Rechnung zeigt.

$$\begin{aligned} n' &= n - 1 \\ m' &= m - 2 \\ m < 2n &\Rightarrow m' + 2 < 2(n' + 1) \\ &\Rightarrow m' < 2n' \end{aligned}$$

Der neue Graph G' ist ein maximaler Graph, da jede Kante die in G' eingefügt werden kann, auch in G eingefügt werden kann, G aber nach Voraussetzung ein maximaler Graph ist. Der Graph G' hat wie auch G mehr als vier Knoten, da das Entfernen eines Knotens nur zu einem Graphen mit vier oder weniger Knoten führen kann, wenn G exakt fünf

Abbildung 3.3: Umleiten von v_1v_4 zu v_2v_4 .

Knoten besitzt. Wenn G fünf Knoten besitzt, entspricht er exakt dem Graphen K_5 , da jeder außen 1-gap plane Graph zu K_5 ergänzt werden kann. K_5 ist ein außen 1-gap planarer Graph mit zehn Kanten. Da G aber weniger als $2n$ Kanten hat, kann G nicht aus fünf Knoten bestehen. G' ist also ein maximaler 1-gap planer Graph mit weniger als $2n$ Kanten und mehr als vier Knoten. Dies widerspricht der Minimalität von G . Daraus folgt, dass es in G keine Knoten mit einem Grad von 2 geben kann. \square

Da wir Graphen suchen, deren Durchschnittsgrad unter 4 liegt, muss es also Knoten vom Grad 3 in G geben. Für diese Knoten führen wir außerdem eine Substruktur ein, die wir in unseren Zeichnungen antreffen werden. Diese Substruktur besteht aus vier Knoten und entspricht der Form des Graphen K_4 .

Lemma 3.4. *Für jeden Knoten mit Grad 3 gibt es eine Substruktur der Form K_4 in G .*

Beweis. Wir betrachten nun Knoten mit Grad 3, und bezeichnen einen solchen Knoten im Folgenden als v_1 . Aufgrund der Ringstruktur in einem Kanten-maximalem außen 1-gap planarem Graphen wissen wir, dass zwei der Verbindungen von v_1 zu dessen Nachbarn v_2 und v_3 laufen. Außerdem gibt es eine weitere Kante zu einem Knoten v_4 . Da G maximal ist, muss es die Kante v_2v_3 geben. Andernfalls könnte diese Kante wie zuvor die Kante v_1v_2 konstruiert werden, mit dem Unterschied, dass hier v_2v_3 beim Knoten v_1 eine Lücke aufweist, durch die die Kante v_1v_4 hindurch geleitet werden kann. Angenommen es gibt keine Kante v_2v_4 . Man kann dann v_1v_4 so verlegen, dass sie von v_4 bis zur Kreuzung mit v_2v_3 genau so verläuft wie bisher, aber dann anstatt v_2v_3 zu kreuzen parallel an dieser weiterverläuft bis zu v_2 . Dies ist möglich, da alle Kreuzungen die v_1v_4 zwischen v_4 und der Kreuzung zu v_2v_3 hat, genau so beibehalten wie bisher, und entlang der Kante v_2v_3 keine weiteren Kreuzungen auftreten können, da v_1 vom Grad 3 ist (siehe Abbildung 3.3).

Durch diese Verlagerung entsteht ein neuer Graph G' , der weiterhin außen 1-gap plan ist. G' ist maximal, da jede Kante die in G' eingefügt werden kann auch in G eingefügt werden kann, und G bereits maximal ist. Dies ist gegeben, da v_1v_4 in G keine Lücke aufwenden muss um v_2v_3 zu kreuzen, also seine Lücke sowohl in G , als auch in G' in dem Bereich zwischen v_4 und v_2v_3 aufwenden kann. Nun gelten für G' die Voraussetzungen des Satzes wie auch für G . In G' hat nun v_1 aber den Grad 2 und widerspricht damit dem Lemma 3.3. Daraus folgt, dass in G die Kante v_2v_4 existieren muss. Analog kann gezeigt werden dass G die Kante v_3v_4 enthalten muss. Dadurch erhalten wir eine Substruktur der Form K_4 mit den Knoten v_1 , v_2 , v_3 und v_4 . \square

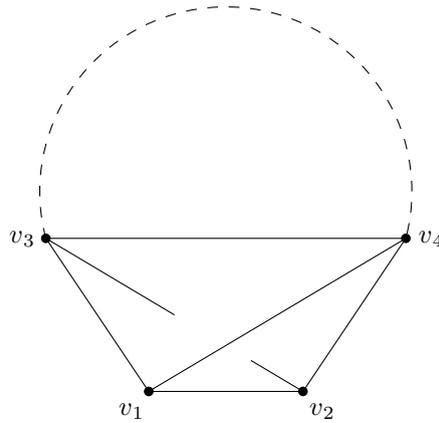


Abbildung 3.4: Benachbarte Knoten mit Grad 3.

Lemma 3.5. *Es gibt keine zwei benachbarten Knoten, die beide den Grad 3 haben.*

Beweis. Angenommen, es gibt zwei Knoten v_1 und v_2 , die direkte Nachbarn sind, und beide den Grad 3 haben. Knoten v_1 hat außerdem den Nachbarn v_3 und v_2 hat den Nachbarn v_4 wie in Abbildung 3.4. Da G maximal ist, und v_1 den Grad 3 hat, müssen die Kanten v_2v_3 und v_1v_4 existieren, wie in 3.4 bewiesen. Da von v_1 und v_2 sonst keine inneren Kanten ausgehen, kann die Kante v_3v_4 unterbrechungsfrei eingezeichnet werden. Nun kann man analog zum Beweis von Lemma 3.3 die beiden Knoten v_1 und v_2 , mit den anliegenden fünf Kanten entfernen. Dadurch erhält man einen kleineren Graph G' , dessen Durchschnittsgrad nach wie vor unter 4 liegt. Dies widerspricht der Annahme, dass G minimal ist. Der neue Graph G' ist nach wie vor maximal, da wir den Graphen G an einer ununterbrochenen Kante aufgetrennt haben, und nur noch eine der beiden Seiten betrachten. Diese ununterbrochene Kante dient nun als neue äußere Kante, kann also weiterhin nur ununterbrochen sein. \square

Theorem 3.6. *Die minimale Dichte eines kanten-maximalen Graphen mit mindestens fünf Knoten liegt bei $2n$ Kanten.*

Beweis. Wir führen hier einen Widerspruchsbeweis. Wenn der Satz gilt, gibt es einen maximalen außen 1-gap planen Graphen mit $n > 4$ Knoten und weniger als $m = 2n$ Kanten. Daraus folgt, dass es einen maximalen außen 1-gap planen Graphen G mit minimaler Knotenanzahl und weniger als $2n$ Kanten gibt. Anhand dieses Graphen bilden wir eine minimales Gegenbeispiel.

Im Folgenden werden wir zeigen, dass ein Graph der unseren ursprünglichen Anforderungen entspricht, nicht weniger als $2n$ Kanten haben kann. Dafür geben dem Graphen nun eine teilweise Orientierung. Der neue Grad eines neuen Knotens wird berechnet als $\deg'(v) = \deg(v) + x - y$, wobei x die Anzahl der eingehenden gerichteten Kanten an v ist, und y die Anzahl der ausgehenden gerichteten Kanten.

Wir führen hierfür die folgenden Invarianten ein:

1. Jeder Knoten v mit ausgehenden gerichteten Kanten hat $\deg(v) \geq 5$.
2. Jeder Knoten v mit ausgehenden gerichteten Kanten hat maximal $\deg(v) - 4$ ausgehende Kanten.
3. Jeder Knoten v mit eingehenden gerichteten Kanten hat $\deg(v) = 3$.
4. Jeder Knoten v hat maximal eine eingehende gerichtete Kante.

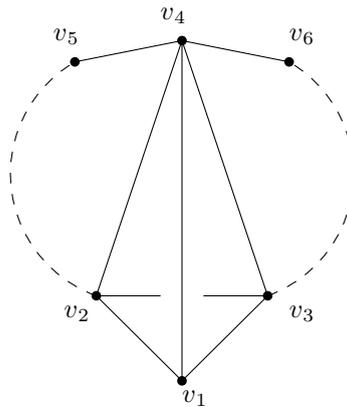


Abbildung 3.5: Fall 1. Es liegen potentiell weitere Knoten auf den gestrichelten Linien.

5. Jede gerichtete Kante ist entweder eine Sehne in einer K_4 Substruktur oder eine Kante am Rand des Graphen G .

Wie werden nun die Kanten so richten, dass mit der neuen Gradberechnung jeder Knoten mindestens den Grad 4 hat. Dies würde dazu führen, dass der durchschnittliche Grad nach neuer Berechnung mindestens 4 ist. Da der Durchschnittsgrad basierend auf dem neuen Grad mit dem des alten Grads übereinstimmt, zeigt dies, dass der Graph mindestens $2n$ Kanten hat.

Wir befinden uns nun in einem teilweise gerichteten Graphen G' mit denselben Knoten und Kanten wie in G . Wähle einen Knoten mit Grad 3 und nenne ihn v_1 . Dieser Knoten bildet laut Lemma 3.4 zusammen mit seinen Nachbarn v_2 und v_3 sowie einem weiteren Knoten v_4 eine K_4 Substruktur. Wir wissen, dass v_4 keine eingehenden gerichteten Kanten haben kann, da $\deg(v_4) > 3$. Wir unterscheiden nun, ob auf beiden Seiten der Substruktur weitere Knoten liegen.

Fall 1: Es liegen weitere Knoten zwischen v_2 und v_4 und zwischen v_3 und v_4 . In diesem Fall müssen wir außerdem unterscheiden, ob ein Nachbar von v_4 in G Grad 3 hatte. Die Nachbarn von v_4 nennen wir v_5 und v_6 ; siehe Abbildung 3.5.

Fall 1.1: Kein Nachbar von v_4 hat in G den Grad 3. In diesem Fall haben die Knoten v_2, v_3, v_5 und v_6 in G mindesten den Grad 4. Die Kanten v_2v_4, v_3v_4, v_4v_5 und v_4v_6 können also nicht gerichtet sein. Daraus folgt, dass alle weiteren Kanten die an v_4 anliegen ausgehend sein können, ohne den neuen Grad von v_4 unter 4 zu senken. Die Kante v_1v_4 kann also von v_4 nach v_1 gerichtet werden, womit der Grad von v_1 auf 4 angehoben wird und die Invarianten erfüllt sind.

Fall 1.2: In diesem Fall hat v_5 in G den Grad 3. Der Knoten v_5 hat nun außerdem den Nachbarn v_7 . Dieser Knoten muss in G mindestens den Grad 4 haben, da keine zwei benachbarten Knoten den Grad 3 haben können. Nun ist die Kante v_4v_7 anstatt der Kante v_4v_5 nicht gerichtet. Der Knoten v_4 kann also weiter seine anderen Kanten als ausgehende Kanten haben; siehe Abbildung 3.6. Fall 1.2 kann analog angewendet werden wenn v_6 in G den Grad 3 hat.

Fall 2: Es liegen keine Knoten zwischen v_3 und v_4 . Wir unterscheiden weiter nach dem Grad des Knotens v_3 .

Fall 2.1: Der Knoten v_3 hat in G den Grad 5. Wenn v_3 nicht die Spitze einer K_4 Substruktur ist, kann v_3 nur eine ausgehende Kante zu v_1 haben, d.h. wir können diese Kante zu v_1 richten; siehe Abbildung 3.7. Wenn v_3 hingegen Spitze einer weiteren Substruktur ist, muss diese entweder v_2 oder v_4 enthalten, da die Kanten v_2v_4 und v_1v_4 nur von zwei weiteren

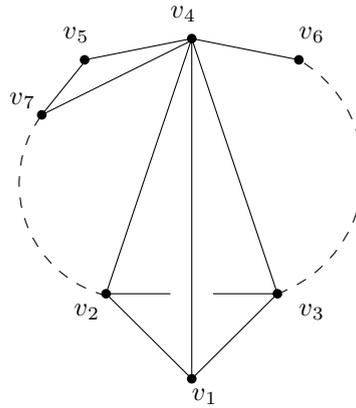


Abbildung 3.6: Fall 1.2. Die Kante von v_5 zu einem weiteren Knoten ist nicht dargestellt, da der zweite Knoten der Kante nicht festliegt.

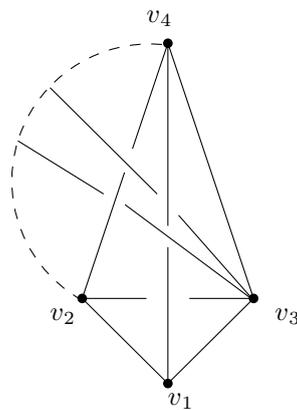


Abbildung 3.7: Fall 2.1. Der Knoten v_3 hat hier exakt den Grad 5.

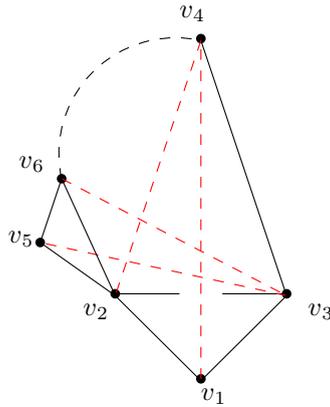


Abbildung 3.8: Fall 2.1.1. Hier ist v_3 die Spitze einer zweiten K_4 Substruktur. Die Kanten markierten Kanten bilden einen undurchdringlichen Kreis.

Kanten durchdrungen werden können. Wir machen hierzu eine weitere Fallunterscheidung. Fall 2.1.1: In diesem Fall bildet v_3 zusammen mit v_2 und zwei weiteren Knoten v_5 und v_6 eine weitere Substruktur. Der Knoten v_5 hat in G den Grad 3; siehe Abbildung 3.8. Die Kanten v_1v_4, v_2v_4, v_3v_5 und v_3v_6 bilden einen Kreis, durch den keine weiteren Kanten führen können. Damit haben v_2 und v_3 in G den Grad 5. Beide Knoten sind außer zu v_2 und v_3 nur zu Knoten mit Grad 4 oder höher, sowie den beiden Knoten v_1 und v_5 mit jeweils Grad 3 verbunden. Daher kann entweder v_1v_3 nach v_1 gerichtet, und v_2v_5 nach v_5 gerichtet werden, oder v_1v_2 nach v_1 und v_3v_5 nach v_5 gerichtet werden. Dabei werden v_1 und v_5 in G' auf Grad 4 angehoben.

Fall 2.1.2: In diesem Fall bildet v_3 zusammen mit v_4 und zwei weiteren Knoten v_5 und v_6 eine weitere K_4 Substruktur. Der Knoten v_5 hat erneut in G den Grad 3. Dieser Fall kann analog zu Fall 2.1.1 bearbeitet werden, mit dem Unterschied dass v_2 und v_4 vertauscht sind.

Fall 2.2: Der Knoten v_3 hat in G den Grad 4. Hier muss noch nach den Graden der Knoten v_2 und v_4 unterschieden werden.

Fall 2.2.1: Der Knoten v_2 hat in G mindestens den Grad 5. Wenn v_2 nicht die Spitze einer K_4 Substruktur ist, müssen wir unterscheiden, ob neben v_2 in G noch ein weiterer Knoten mit Grad 3 liegt.

Fall 2.2.1.1: Der Knoten v_2 hat noch einen weiteren Nachbarn v_7 mit Grad 3 in G . Dieser Knoten bildet damit die Basis einer weiteren K_4 Substruktur. Die Spitze dieser Struktur kann nun entweder v_4 sein oder nicht; siehe Abbildung 3.9

Fall 2.2.1.1.1: Wenn v_4 die Spitze zu v_7 ist, hat v_4 vier Kanten zu Knoten, die nicht gerichtet werden können. Neben den zwei vorher bestehenden Kanten, muss eine Kante zum weiteren Nachbarn von v_7 bestehen. Außerdem hat v_4 eine Kante zu seinem weiteren Nachbarn, v_8 . Wenn dieser mindestens den Grad 4 hat, kann die Kante nicht gerichtet werden, siehe Abbildung 3.10. Wenn v_8 hingegen in G den Grad 3 hat, hat v_4 eine Kante zum anderen Nachbarn von v_8 , der mindestens Grad 4 hat, siehe Abbildung 3.11. Damit kann die Kante zwischen v_7 und v_4 nach v_7 gerichtet werden. Da v_7 nur eine eingehende Kante haben kann, kann die Kante von v_2 nach v_1 gerichtet werden. Damit sind sowohl v_7 als auch v_1 im gerichteten Graph von Grad 4.

Fall 2.2.1.1.2:

Wenn v_4 nicht die Spitze zu v_7 ist, gibt es einen weiteren Knoten v_8 , der diese Spitze ist. Dieser hat Kanten zu den Nachbarn von v_7 , die nicht gerichtet werden können. Außerdem hat v_8 zwei Nachbarn; siehe Abbildung 3.12. Wenn diese in G vom Grad 4 oder höher sind, sind die Kanten ebenfalls ungerichtet. Wenn einer oder beide der von v_8 in G vom Grad 3 sind, ist die Kante von v_8 zu dem Nachbarn des Knotens mit Grad 3 auf jeden Fall

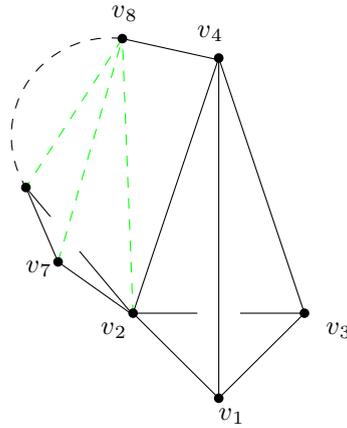


Abbildung 3.9: Fall 2.2.1.1. Für diesen Fall muss unterschieden werden, ob v_4 und v_8 identisch sind. Falls sie identisch sind, sind die markierten Kanten zum Knoten v_4 verbunden.

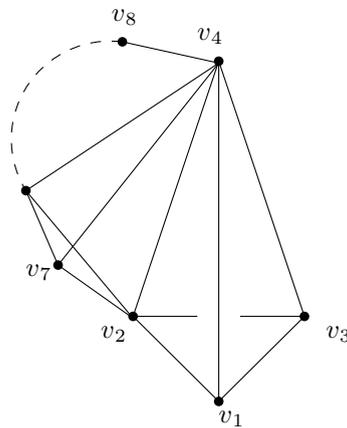


Abbildung 3.10: Fall 2.2.1.1.1. Diese Graphik zeigt den Fall, in dem der Knoten v_8 nicht den Grad 3 hat.

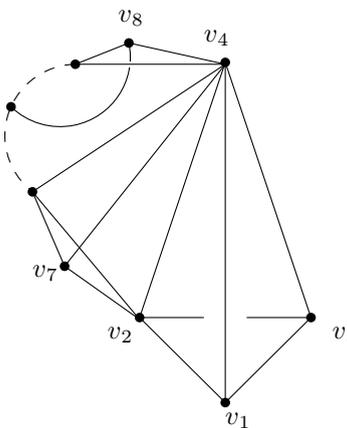


Abbildung 3.11: Fall 2.2.1.1.1. In dieser Graphik ist die Zeichnung für den Fall, dass der Knoten v_8 den Grad 3 hat dargestellt.

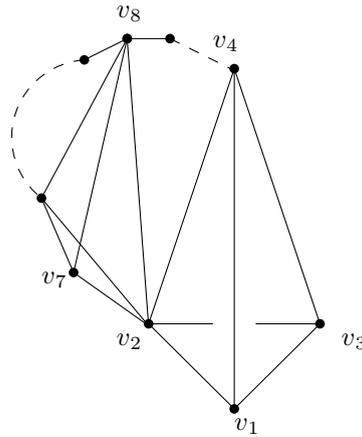


Abbildung 3.12: Fall 2.2.1.1.2. Hier sind zwei separate K_4 Substrukturen vorhanden, die sich nur einen Knoten teilen.

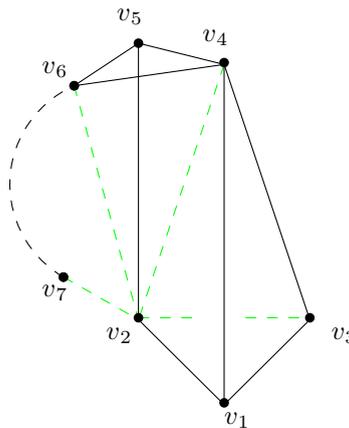


Abbildung 3.13: Fall 2.2.1.3. Hier teilen sich die beiden K_4 Substrukturen die Kante v_2v_4 . Die markierten Kanten sind nicht gerichtet, also kann v_2 alle anderen Kanten nach außen richten, ohne die Invarianten zu verletzen.

ungerichtet. Damit kann die Kante zwischen v_7 und v_8 nach v_7 gerichtet werden. Da v_7 nur eine eingehende Kante haben kann, kann die Kante von v_2 nach v_1 gerichtet werden. Damit sind sowohl v_7 als auch v_1 im gerichteten Graph von Grad 4.

Fall 2.2.1.2: Wenn v_2 nur zu einem Knoten, der in G Grad 3 hat, verbunden ist, kann man direkt die Kante v_1v_2 nach v_1 richten.

Wenn v_2 hingegen Spitze einer K_4 Substruktur ist, müssen wir noch unterscheiden, ob v_4 Teil dieser Struktur ist.

Fall 2.2.1.3: In diesem Fall besteht die zweite Substruktur aus den Knoten v_2, v_4, v_5 und v_6 , wobei v_5 in G den Grad 3 hat. Der Knoten v_2 hat neben v_1 einen weiteren Nachbarknoten, den wir hier v_7 nennen; siehe Abbildung 3.13. Wir müssen nun weiter unterscheiden ob der Knoten v_7 in G vom Grad 3 oder höher ist.

Fall 2.2.1.3.1: Wenn der Grad von v_7 in G mindestens 4 ist, hat v_2 vier Kanten die nicht gerichtet werden können. Diese Kanten sind v_2v_3, v_2v_4, v_2v_6 und v_2v_7 . Daraus folgt, dass alle anderen Kanten, die an v_2 anliegen, ausgehend sein können, ohne dass der Grad von v_2 unter 4 fällt. Damit haben sowohl v_1 als auch v_5 eine eingehende Kante und ihr Grad wird auf 4 angehoben.

Fall 2.2.1.3.2: Wenn der Knoten v_7 in G vom Grad 3 ist, ist auch der andere Nachbar von v_7 relevant, und wird als v_8 bezeichnet. Wenn der Knoten v_8 nicht identisch mit dem Knoten

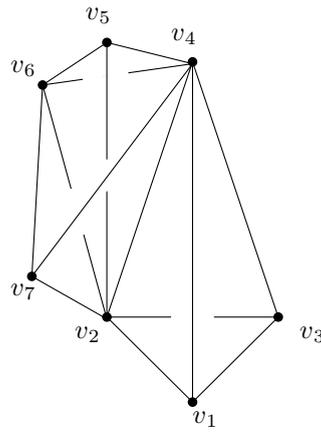


Abbildung 3.14: Fall 2.2.1.3.2.

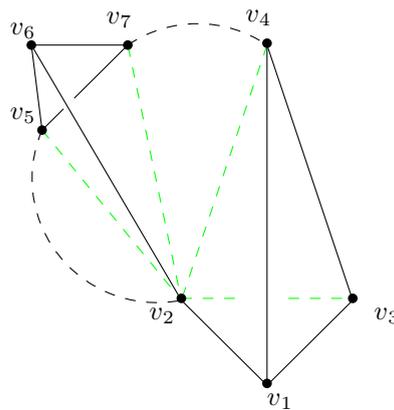


Abbildung 3.15: Fall 2.2.1.4. Die markierten Kanten sind nicht gerichtet, also kann v_2 alle anderen Kanten nach außen richten, ohne die Invarianten zu verletzen.

v_6 ist, hat v_2 eine Kante zu v_8 , die nicht gerichtet werden kann. Dadurch kann v_2 wieder seine anderen Kanten wieder zu v_1 und v_5 richten. Sind v_8 und v_6 hingegen identisch, ist die einzige potentielle Spitze des K_4 , der zu v_7 gehört, der Knoten v_4 . Dadurch ergibt sich aber ein Graph, der exakt sieben Knoten enthält, aber bereits mehr als 14 Kanten hat. Damit entspricht dieser Graph nicht unseren Voraussetzungen, da er nicht weniger als $2n$ Kanten besitzt; siehe Abbildung 3.14.

Fall 2.2.1.4: Hier besteht die zweite Substruktur aus v_2 zusammen mit v_5 , v_6 und v_7 , wobei v_6 in G den Grad 3 hat; siehe Abbildung 3.15. Hier hat v_2 die Kanten v_2v_3 , v_2v_4 , v_2v_5 und v_2v_7 , die nicht gerichtet sein können. Dadurch können alle anderen Kanten an v_2 ausgehende Kanten sein. Die Kante v_2v_6 kann also nach v_6 gerichtet sein, und die Kante v_1v_2 kann nach v_1 gerichtet sein. Dadurch sind sowohl v_1 als auch v_5 im neuen Grad bei mindestens 4.

Fall 2.2.2: In diesem Fall hat v_4 in G mindestens den Grad 5. Dieser Fall funktioniert analog zu Fall 2.2.1, indem die Fälle symmetrisch aufgebaut werden.

Fall 2.2.3: In diesem Fall haben sowohl v_2 als auch v_4 in G den Grad 4; siehe Abbildung 3.16. Es ist hier möglich, eine weitere Kante in den Graphen G einzufügen. Man kann beim Nachbarknoten v_5 von v_2 zu zeichnen beginnen, und entlang der ununterbrochenen Außenkante v_2v_5 zeichnen, dann bei v_2 Richtung wechseln entlang der Kante v_2v_4 bis zu v_4 . Dies ist möglich, da durch den Grad von v_2 keine Kante von v_2 aus in diesen Bereich führt. Die neue Kante v_4v_5 kann ihre eigene Lücke aufwenden, um die Kante, die von v_3 ausgeht hindurchzulassen. Dadurch ist die Maximalität von G nicht gegeben, und der Fall kann so nicht existieren. \square

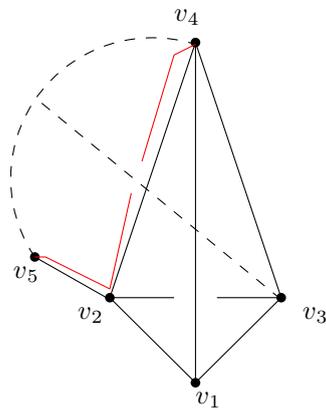


Abbildung 3.16: Fall 2.2.3. Diese Graphik zeigt, dass die rot gepunktete Linie in die bestehende Zeichnung eingefügt werden kann.

4. Lösung mittels Gurobi

4.1 Aufbau des Programms

Wir haben außerdem die Software Gurobi verwendet, um außen 1-gap planare Graphen zu vervollständigen und zu erkennen. Gurobi ermöglicht es, ganzzahlige lineare Programme zu lösen. Wir haben ein Java Programm entwickelt, das auf die Programmierschnittstellen von Gurobi zugreift, um uns bei der Arbeit mit außen 1-gap planaren Graphen zu unterstützen. Beim Start des Programms hat der Anwender verschiedene Eingabemöglichkeiten. Das Programm ermöglicht die Eingabe eines Graphen durch Benennung der enthaltenen Knoten. Außerdem können bereits existierende Kanten zwischen verschiedenen Knoten übergeben werden. Der Nutzer kann wählen, ob das Programm den erhaltenen Graphen maximieren soll, also so viele Kanten hinzufügen wie möglich ist. Alternativ kann der Nutzer auch nur überprüfen, ob der eingegebene Graph außen k -gap planar ist. Es ist außerdem möglich zu wählen, ob das Programm die eingegebenen Knoten in der vorgegebenen Reihenfolge behalten muss, oder ob eine besser geeignete Reihenfolge gewählt werden darf.

Das Programm erstellt die folgenden Variablen um einen Graphen $G = (V, E)$ zu modellieren:

k : Diese Variable wird verwendet, um zu erkennen für wie groß k mindestens sein muss, so dass der zu prüfende Graph k -gap planar ist. Diese Variable liegt zwischen 0 und einer oberen Grenze.

$x_{u,v}$: Diese Variable ist 1, wenn der Knoten u in der Reihenfolge der äußeren Knoten vor dem Knoten v kommt, ansonsten ist die Variable 0. Für jedes Knotenpaar existieren zwei dieser Variablen mit umgekehrten Indizes. $u, v \in V$

$e_{\{u,v\}}$: Diese Variable ist 1, wenn die Kante zwischen Knoten u und Knoten v im Graph existiert. Für jedes Knotenpaar existiert nur eine dieser Variablen. Hat der Nutzer bereits Kanten als existent vorgegeben, werden die entsprechenden Variablen bereits auf 1 festgelegt. $u, v \in V$

$C_{u,v,x,y}$: Diese Variable ist 1, wenn die Kante zwischen u und v eine Kreuzung mit der Kante zwischen x und y hat und eine Lücke aufweist um die Kante xy durchzulassen. Bei dieser Variable gibt es für jedes Kantenpaar zwei Variablen. $u, v, x, y \in V$ paarweise verschieden

Die Menge V ist hierbei die Menge aller Knoten des Graphen.

Wir betrachten nun die Option, dass die Reihenfolge der Knoten vom Nutzer fest vorgegeben ist, und das Programm nur prüfen soll, ob es sich um einen außen k -gap planaren Graphen handelt. Dabei werden die Variablen zur Knotenreihenfolge nicht vergeben, da diese nicht benötigt werden. Außerdem werden nur die Kantenvariablen erstellt, für Kanten, die der

Nutzer explizit angegeben hat, da das Programm den Graphen nicht vervollständigen wird. Kreuzungsvariablen werden angelegt für Kanten, die sich kreuzen. Da die Reihenfolge der Knoten fix ist, ist es möglich bereits im Voraus festzustellen, ob sich Kanten kreuzen können.

Um das erstellte Modell zu lösen, wird Gurobi versuchen die Variable k zu minimieren. Als Ergebnis wird hierbei nur festgestellt, ob es eine Lösungsmöglichkeit gibt, und ab welchem Wert von k eine solche Lösung existiert.

Um die Berechnung durchzuführen, werden Gleichungen als Einschränkungen benötigt. Im folgenden werden diese Gleichungen angegeben.

$$e_{\{u,v\}} + e_{\{x,y\}} - C_{u,v,x,y} - C_{x,y,u,v} \leq 1 \quad (4.1)$$

für alle sich kreuzenden Kanten uv und xy .

$$\sum_{xy \in E} C_{u,v,x,y} \leq 1 \quad (4.2)$$

für alle Kanten uv .

Anschließend wird das Modell mit Gurobi optimiert und das Ergebnis ausgegeben.

Das Programm kann außerdem prüfen, ob es sich um einen außen k -planaren Graphen handelt. Dazu wird die Gleichung

$$C_{u,v,x,y} - C_{x,y,u,v} = 0 \quad (4.3)$$

für alle Kantenpaare u, v und x, y in E . Dadurch wird jede Kreuzung auf beiden Kanten gezählt und nach dem minimieren von k gibt das Ergebnis einen außen k -planaren Graphen zurück.

Im Weiteren wird die Gleichung 4.1 nicht mehr benötigt, da alle folgenden Graphen außen k -gap planar sein sollen. Betrachten wir nun die Veränderungen, wenn das Programm die Anzahl an Kanten maximieren soll. Die Reihenfolge der Knoten ist weiterhin festgelegt. Hier werden zusätzlich die Kantenvariablen für alle möglichen Knotenpaare erstellt. Kreuzungsvariablen werden erneut nur erstellt für Kanten, die sich basierend auf der Knotenreihenfolge kreuzen können. Hier wird der Term $\sum_{e_{u,v} \in E} e_{u,v}$ maximiert. Die Gleichungen bleiben hier die selben. Das Ergebnis das Gurobi liefert ist hier ein Graph, der aus dem eingegebenen Graphen entsteht mit maximaler Anzahl an Kanten. Es kann andere Graphen mit gleicher Kantenanzahl geben, diese werden hier aber nicht ausgegeben.

Wenn die Knotenreihenfolge hingegen frei ist, ergeben sich mehrere Änderungen. Einerseits werden nun die Variablen zur Knotenreihenfolge erstellt. Es ist darauf zu achten, dass ein Knoten als Start festgelegt wird, und alle diesen Knoten betreffenden Variablen so festgelegt werden, dass dieser Knoten vor allen anderen Knoten kommt. Hier müssen nun für alle Kantenpaare Variablen für die Kreuzungen angelegt werden, da sich die Reihenfolge während des Optimierungsprozesses ändern kann. Das Optimierungsziel bleibt das Gleiche wie in den entsprechenden vorherigen Fällen. Hier muss Gleichung 4.1 entfernt werden, da nicht klar ist, welche Kanten sich kreuzen. Stattdessen müssen neue Gleichungen eingeführt werden. Für diese Gleichungen ist aber zuerst notwendig, die Transitivität der Knoten zu gewährleisten:

$$x_{u,v} + x_{v,x} - x_{u,x} \leq 1 \quad (4.4)$$

für alle $u, v, x \in V$, wobei u, v und x paarweise verschieden sind. Nun kann die folgende Gleichung für alle paarweise verschiedenen Knoten u, v, x und $y \in V$ eingeführt werden.

$$e_{\{u,v\}} + e_{\{x,y\}} + x_{u,x} + x_{x,v} + x_{v,y} - x_{u,v,x,y} - x_{x,y,u,v} \leq 4 \quad (4.5)$$

Diese Gleichung sorgt dafür, dass bei den der Gleichung entsprechenden Reihenfolgen der Knoten jeder Kreuzung zweier existierender Kanten eine der beiden Kanten ihre Lücke verwendet um die andere Kante hindurchzulassen. Mit diesen zusätzlichen Gleichungen kann nun Gurobi wieder ein Ergebnis für das Modell suchen.

4.2 Auswertung

Um unsere Anwendung zu testen, haben wir Graphen erzeugt und getestet. In dieser Analyse haben wir das Programm im Modus zur Erkennung ohne Maximierung des Graphen ausgeführt. Die erzeugten Graphen bestehen alle aus $n = 10$ Knoten. Wir haben je 50 Graphen mit einer durchschnittlichen Dichte d von 1, 2 und 3 erstellt. Die Dichte ist dabei definiert als $d = m/n$ wobei m für die Anzahl an Kanten im Graphen steht. Wir analysieren damit die Laufzeit unseres Programms für die verschiedenen Einstellungen, und die Ergebnisse bezüglich des minimalen Wertes für k der nötig ist, um die Graphen als außen k -gap planare Graphen, bzw. außen k -planare Graphen darzustellen. Die Graphen wurden erzeugt, indem zehn Knoten erzeugt wurden, und alle möglichen Kanten mit der gleichen Wahrscheinlichkeit p hinzugefügt werden. Dabei wurde p so gewählt, dass die Anzahl der Kanten der gewünschten Dichte entspricht. Bei unseren Ergebnissen für die Graphen mit durchschnittlicher Dichte 3 haben wir nicht alle Graphen auf k -gap Planarität getestet, da die Tests mehr als 15 Minuten gedauert haben, und dabei das Zeitlimit, das wir für die Berechnung der Graphen gesetzt haben, überschritten. Deshalb sind in der dritten Kategorie nur 13 Datensätze vorhanden. Zur Einordnung der Ergebnisse bezüglich der Laufzeit sind hier einige Grunddaten des Systems, auf dem der Test durchgeführt wurde:

- CPU: intel i7 950 @3.07GHz
- 12GB RAM
- Windows 10
- Java 1.8 Update 151

Zur Darstellung unserer Ergebnisse haben wir Box-Whisker-Plots verwendet. Die Grafik in 4.1 zeigt die Laufzeit in derer unser Programm den Test auf außen k -gap Planarität durchgeführt hat. Diese Darstellung zeigt die Laufzeiten aufgeteilt nach den Gruppierungen für die Dichtewerte mit denen die Graphen generiert wurden. Für jede der Gruppen gibt es eine Box in der Mitte, in dessen Umfang sich 50% der Datenpunkte befinden. Eine Linie im inneren der Box markiert den Durchschnittswert für diese Gruppe. Die Linien die von der Box ausgehen sind maximal 1.5 mal so lang wie die Box. Diese Whisker enden am letzten Datenpunkt der noch innerhalb ihrer maximalen Reichweite liegen. Datenpunkte die außerhalb der Whisker liegen werden als Ausreißer betrachtet und einzeln als Punkte dargestellt. Im Vergleich dazu zeigt Grafik 4.2 die Laufzeit für außen k -Planarität. Diese Plots zeigen bereits, dass der Test auf k -gap Planarität deutlich länger dauert. Dazu haben wir noch den Plot 4.3, der das Verhältnis der Dauer darstellt. Es ergibt sich, dass für dichtere Graphen der Test auf k -Planarität immer besser wird. Dieses Ergebnis deckt sich mit unseren Erwartungen, da bei dem Test auf außen k -Planarität deutlich weniger Varianz vorhanden ist, da hier keine Lückenverteilung festgelegt werden muss, sondern jede Kreuzung beider Kanten angerechnet wird. Allerdings ist zu beachten, dass wir nur einen kleinen Datensatz untersuchen, diese Ergebnisse also bei ausführlicheren Tests variieren könnten. Außerdem haben wir noch die Plots für die minimalen k -Werte der Graphen.

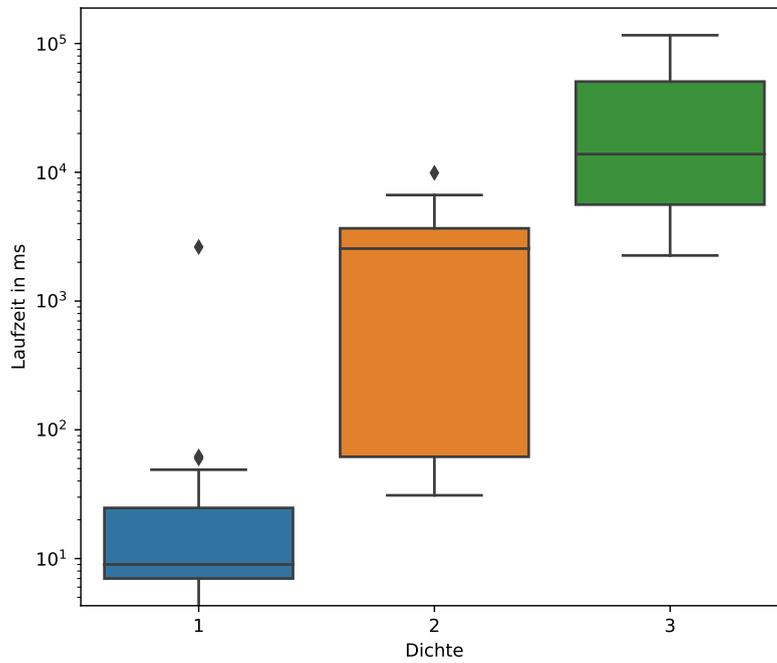


Abbildung 4.1: Diese Grafik zeigt die benötigte Berechnungszeit zum testen der Graphen auf außen k -gap Planarität abhängig von deren Dichte auf einer logarithmischen Skala.

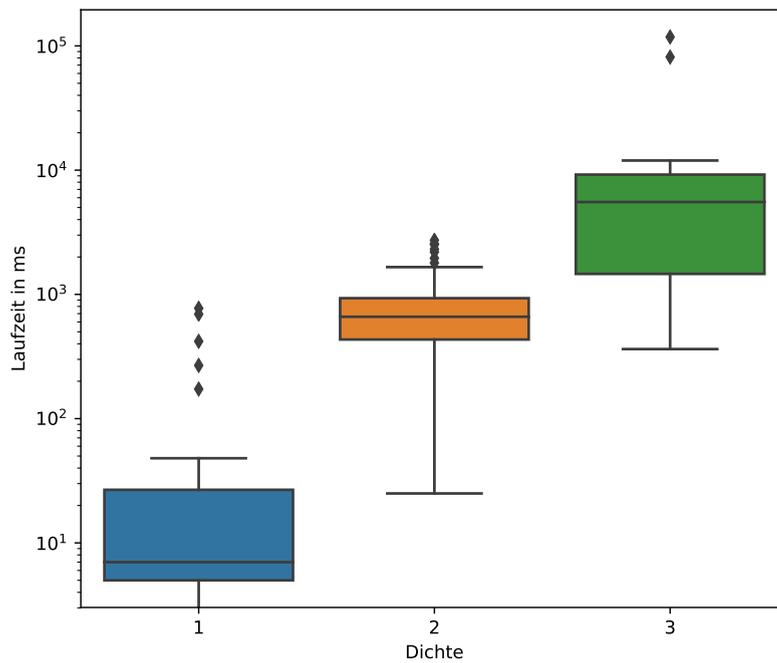


Abbildung 4.2: Diese Grafik zeigt die benötigte Berechnungszeit zum testen der Graphen auf außen k -Planarität abhängig von deren Dichte auf einer logarithmischen Skala.

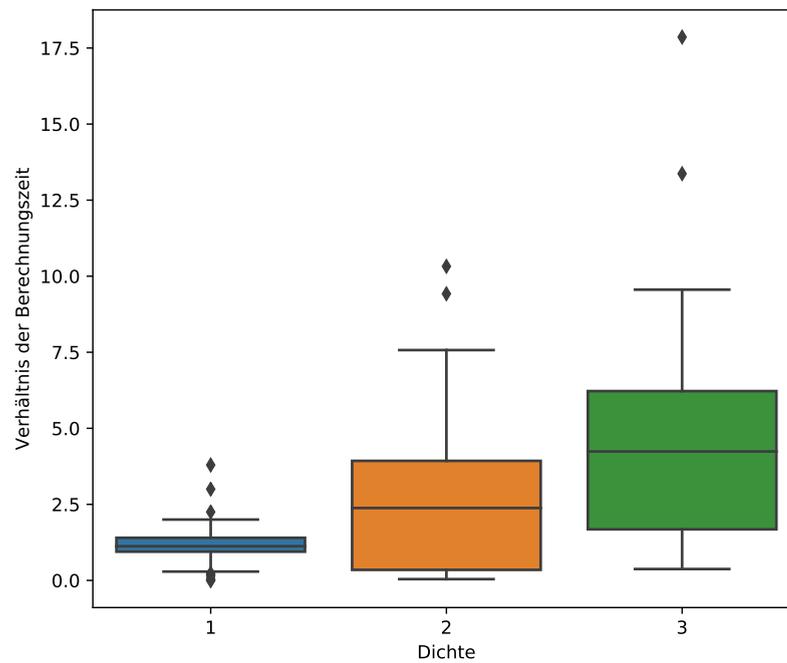


Abbildung 4.3: Diese Grafik zeigt das Verhältnis zwischen den Berechnungszeiten der Graphen. Um diese Daten zu erhalten, teilen wir die Berechnungszeit eines jeden Graphen zur Überprüfung auf außen k -gap Planarität durch die Berechnungszeit für außen k -Planarität.

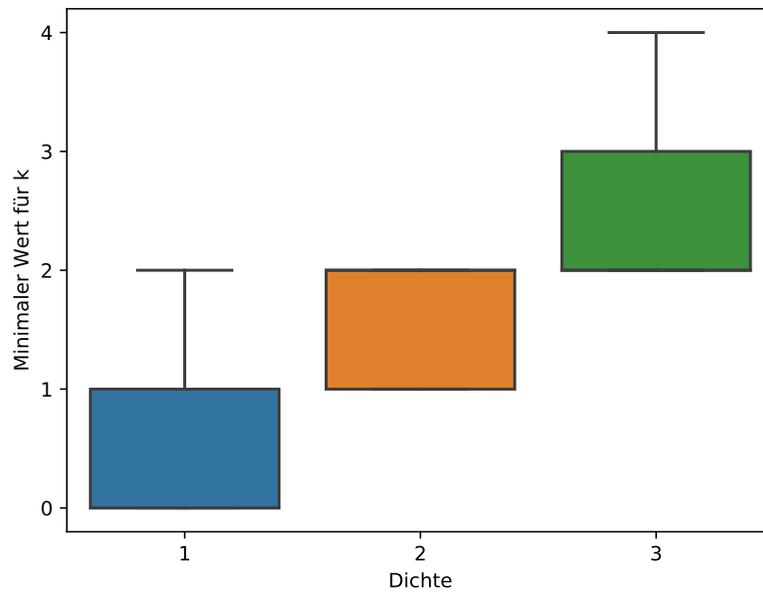


Abbildung 4.4: Diese Grafik zeigt die minimalen Werte für k , die nötig sind, um die überprüften Graphen als außen k -gap planare Graphen darzustellen.

Dabei zeigt 4.4 die Werte für außen k -gap planare Graphen, und 4.5 für außen k -planare Graphen. Dabei hätten wir entsprechend der Ergebnisse in der Arbeit von Bae et. al. [BBC⁺18] erwartet, dass die k -Werte für außen k -gap planare Graphen um den Faktor 2 kleiner sind. Unsere Ergebnisse zeigen allerdings, dass diese Werte bei unseren Graphen um bis zu den Faktor 4 kleiner sind als bei den entsprechenden außen k -planaren Graphen. Das Verhältnis ist dargestellt in 4.6.

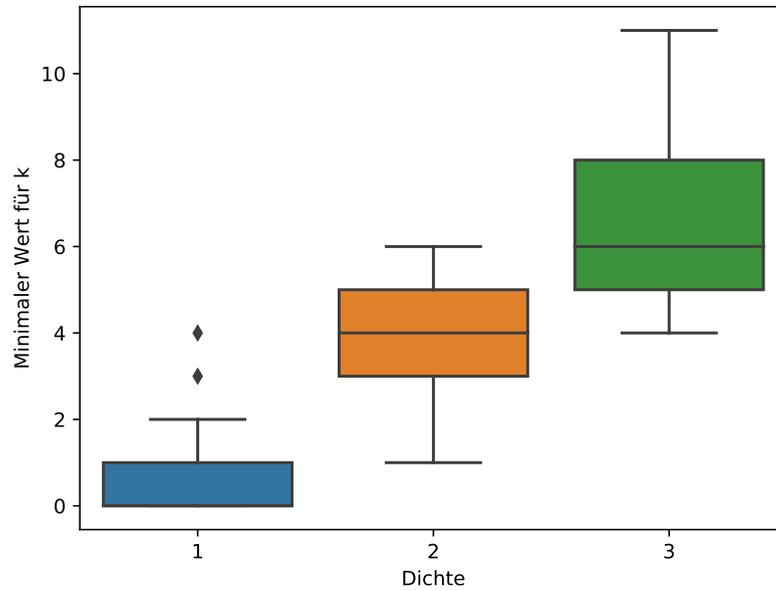


Abbildung 4.5: Diese Grafik zeigt die minimalen Werte für k , die nötig sind, um die überprüften Graphen als außen k -planare Graphen darzustellen.

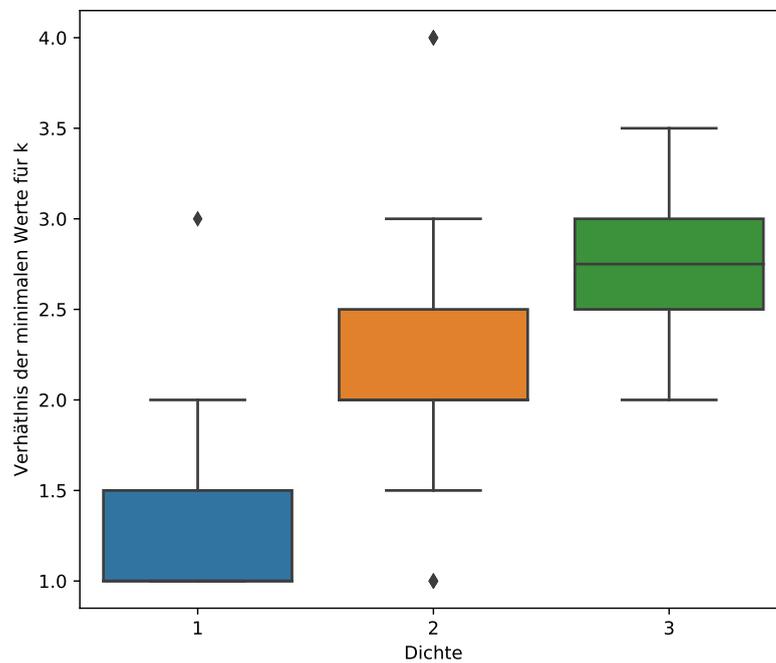


Abbildung 4.6: Diese Grafik vergleicht die minimalen Werte nötig, um die Graphen als außen k -gap planar, bzw als außen k -planar darzustellen. Die dargestellten Daten sind der Quotient aus dem k für k -Planarität geteilt durch den Wert für k -gap Planarität.

5. Zusammenfassung

Wir haben in dieser Arbeit außen 1-gap planare Graphen eingeführt und untersucht. Zuerst haben wir in Kapitel 3.1 die Obergrenze der Dichte von außen 1-gap planaren Graphen mit $3n - 5$ bestimmt, indem wir die Kanten im inneren des Graphen nach außen kopiert haben, und dann bekannte Eigenschaften von 1-gap planaren Graphen angewendet haben. Anschließend zeigten wir eine Zeichenvorschrift, mit Hilfe derer Graphen mit mindestens $3n - 6$ Kanten erzeugt werden können. Desweiteren muss jeder Graph mit einer vollständigen Einbettung mindestens $2n$ Kanten haben, wie in Kapitel 3.2 gezeigt wurde. Dies haben wir mit einem Widerspruchsbeweis bewiesen, in dem wir gezeigt haben, dass es keine Möglichkeit gibt, dass ein Graph mit weniger als $2n$ Kanten vollständig ist. Neben dem theoretischen Teil der Arbeit haben wir ein ganzzahliges lineares Programm erstellt, das außen k -gap planare und außen k -planare Graphen erkennen, und außen 1-gap planare Graphen maximieren kann.

Neben diesen Ergebnissen bleiben noch einige Fragen offen, wie zum Beispiel, ob die Obergrenze der Dichte scharf ist. Momentan ist der Unterschied bereits sehr klein, mit nur einer Kante Unterschied bei einigen n Werten. Dennoch wäre es interessant, ob es möglich ist, die Grenze so zu definieren, dass sie für jedes n scharf ist. In dieser Arbeit haben wir uns auf außen 1-gap planare Graphen beschränkt, es bleibt also offen, wie sich die ermittelten Werte und Eigenschaften an außen k -gap planare Graphen anpassen lassen. Wir haben die Erkennung von außen k -gap planaren Graphen nur mittels unserem ganzzahligen linearem Programm durchgeführt, es bleibt also zu zeigen, ob es einen effizienten Algorithmus gibt, der es ermöglicht solche Graphen in Polynominalzeit zu erkennen.

Literaturverzeichnis

- [ABB⁺16] Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth und Josef Reislhuber: *Outer 1-Planar Graphs*. *Algorithmica*, 74(4):1293–1320, Apr 2016, ISSN 1432-0541. <https://doi.org/10.1007/s00453-015-0002-1>.
- [BBC⁺18] Sang Won Bae, Jean Francois Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter *et al.*: *Gap-planar graphs*. *Theoretical Computer Science*, 2018.
- [BEG⁺13] Franz J. Brandenburg, David Eppstein, Andreas Gleißner, Michael T. Goodrich, Kathrin Hanauer und Josef Reislhuber: *On the Density of Maximal 1-Planar Graphs*. In: Walter Didimo und Maurizio Patrignani (Herausgeber): *Graph Drawing*, Seiten 327–338, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg, ISBN 978-3-642-36763-2.