

Masterarbeit

Das Lokale Kantenlängenverhältnis – Komplexität und Greedy-Algorithmus

Michael Leeming

Abgabedatum: 30. September 2021
Überarbeitet: 19. Juni 2022
Betreuer: Prof. Dr. Alexander Wolff



Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik I
Algorithmen und Komplexität

Zusammenfassung

Das lokale Kantenlängenverhältnis beschreibt das größte Längenverhältnis zweier adjazenter Kanten in der Zeichnung eines Graphen, im Gegensatz zum globalen, bei dem alle Kantenpaare betrachtet werden. Es wird der Beweis der NP-Schwere für das Entscheidungsproblem aufgezeigt, ob ein Graph eine Zeichnung mit einem vorgegebenen lokalen oder globalen Kantenlängenverhältnis besitzt. Dazu wird ein NP-Schwere-Beweis aus dem Jahr 2001 für das Entscheidungsproblem verwendet, ob ein Graph mit gegebener orthogonaler Repräsentation auf ein Gitter mit vorgegebener Größe gezeichnet werden kann. Dieser Beweis wird so modifiziert, dass die entsprechende Reduktion von Satisfiability auf unser Problem übertragen werden kann. Dazu wird der Bauklotz-Graph eingeführt, der mittels eines bestimmten Kantenlängenverhältnisses in orthogonale Form gebracht werden kann.

Des Weiteren wird ein Greedy-Algorithmus für das Zeichnen von 3-Bäumen mit möglichst geringem lokalem Kantenlängenverhältnis vorgestellt, der in Tests zu guten Ergebnissen führte, für den aber keine Güte bewiesen werden konnte. Dieser Greedy-Algorithmus stellt eine nicht-optimale Heuristik dar.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Schreibkonventionen, Problemstellung, Annahmen	7
2	Komplexität	9
3	Greedy-Algorithmus	23
3.1	Idee, Algorithmus	23
3.2	Implementierung	27
3.2.1	Breitensuche	27
3.2.2	Tiefensuche	27
4	Fazit	31
	Literaturverzeichnis	32

1 Einleitung

„Zum Thema ... gibt es bereits viele Arbeiten.“ Dieser Satz trifft nicht auf das Thema dieser Masterarbeit zu. Die Idee, ein *lokales Kantenlängenverhältnis* einzuführen, hatten zuerst Blažej et al. (2020) [BFL20] und beschrieben dieses als das größte Verhältnis zweier adjazenter Kanten in der Zeichnung eines Graphen. Es scheint überraschend, dass dieses Qualitätsmaß für planare Graphen nicht früher in Betracht gezogen wurde. Gerade als Kriterium für Graphästhetik ist das Kantenlängenverhältnis oft naheliegend, wobei dabei aber besonders das Verhältnis benachbarter Kanten relevant ist, da für die Leserlichkeit eines Teilgraphen wohl kaum von Belang ist, wie die Zeichnung in weit entfernten Graphteilen aussieht. Für dieses Qualitätsmaß für planare geradlinige Zeichnungen von Graphen zeigten sie eine obere Grenze von $4 + \varepsilon$ auf serien-parallelen Graphen auf für ein beliebig kleines $\varepsilon > 0$. Der entsprechende konstruktive Beweis kann mit einem Linearzeitalgorithmus umgesetzt werden. Am Ende wurden die Fragen aufgeworfen, ob die genannte Grenze scharf sei und vorgeschlagen, das lokale Kantenlängenverhältnis auch auf anderen Klassen planarer Graphen zu untersuchen. Da die meisten Innenwinkel der Zeichnungen, die die aufgezeigte Konstruktion erzeugte, bei steigender Knotenanzahl sehr nah gegen 0 oder π gingen, wurde außerdem zur Betrachtung des Zusammenspiels von lokalem Kantenlängenverhältnis und Winkelauflösung angeregt.

Außer der eben erwähnten gibt es bisher keine anderen Arbeiten zum Thema „lokales Kantenlängenverhältnis“. Somit bietet sich hier ein Feld, in dem noch unzählige Aspekte betrachtet bzw. erarbeitet werden können, darunter die bereits aufgeworfenen Fragen. Um in einer weiteren Familie planarer Graphen anzusetzen, bieten sich 3-Bäume an, da sich Blažej et al. auf 2-Bäume konzentrierten, die maximale serien-parallele Graphen darstellen. Bei 3-Bäumen lässt sich aber schon bei kurzer Betrachtung erahnen, dass die Technik, mit der Blažej et al. die Knoten eines 2-Baums für ein möglichst geringes lokales Kantenlängenverhältnis anordneten, schwer auf 3-Bäume zu übertragen sein wird, wenn man an geradliniger Kantenzeichnung festhält. Die Minierung des bereits bekannten Kantenlängenverhältnisses, das sich auf das maximale Verhältnis beliebiger Kanten im Graph bezieht (in dieser Arbeit wie in der von Blažej et al. als *globales Kantenlängenverhältnis* bezeichnet) wurde bereits von Borrazzo et al. [BF20] und Lazard et al. [LLL19] untersucht. Sie zeigten u.a. eine untere Schranke in $\Omega(n)$ für planare Graphen mit n Knoten auf sowie die Existenz von Zeichnungen mit einem globalem Kantenlängenverhältnis echt kleiner als 2 für außenplanare Graphen. Eine NP-Schwere konnte bislang aber nur für das Zeichnen bei vorgegebenen Kantenlängen von Eades und Wormald [EW90] bewiesen werden. Das Zeichnen mit festen Kantenlängen auf planaren 3-fachzusammenhängenden Triangulierungen dagegen ist in Linearzeit lösbar ([CDR07]). Eine Untersuchung der Komplexität des globalen, wie auch des lokalen Kantenlängenverhältnisses wäre somit höchst interessant (Kapitel 2). In allen diesen Arbeiten wurde

natürlich von geradlinigen, planaren Zeichnungen ausgegangen. Anderfalls wäre die Minimierung des Kantenlängenverhältnisses allzu trivial.

Weitere mögliche Gebiete für eine Untersuchung des lokalen Kantenlängenverhältnisses wären beispielsweise Bäume, wiederum im Zusammenspiel mit Winkelauflösung, planare Graphen mit vorgegebenem Maximalgrad oder die Zeichnung bei vorgegebener Maximalanzahl an Kantenkreuzungen. Das lokale Kantenlängenverhältnis ließe sich natürlich auch als *lokales Kantenlängenverhältnis i -ten Grades* definieren für Kanten, die, informell gesprochen, über $i - 1$ Kanten zueinander adjazent sind.

2001 konnte Patrignani [Pat01] die NP-Schwere der flächenminimierenden Zeichnung orthogonaler Repräsentationen von Graphen auf dem Gitter zeigen, was mit die Hauptmotivation zu dieser Masterarbeit einleitet. Denn es scheint zunächst nicht gerade offensichtlich, was an flächenminimierenden orthogonalen Zeichnungen im Zusammenhang mit dem Kantenlängenverhältnis interessant sein könnte, tatsächlich aber – um an dieser Stelle bereits auf eines der Hauptergebnisse vorzugreifen – konnte der Beweis von Patrignani leicht modifiziert und so für einen Beweis der NP-Schwere für die Minimierung des Kantenlängenverhältnisses genutzt werden (sowohl des lokalen als auch des globalen). Das lokale Kantenlängenverhältnis ist wie bereits erwähnt ein weitestgehend unerforschtes Gebiet und in gleicher Weise könnten diese oder andere Arbeiten weitere Möglichkeiten im Bereich der Graphentheorie eröffnen, die sich zum jetzigen Zeitpunkt schwer erahnen lassen.

Zum anderen bietet die Tatsache, dass das Thema „lokales Kantenlängenverhältnis“ ein bisher weitestgehend unerforschtes Gebiet ist, einfach ein spannendes Feld für eine Masterarbeit. Im letzten Absatz wurden dazu schon einige mögliche Ansatzpunkte für Arbeiten genannt. Ebenso sind für die angewandten Techniken keine Grenzen gesetzt – greedy, kräftebasiert, divide-and-conquer, dynamische Programmierung, FTP oder Approximation bieten beispielsweise unzählige Möglichkeiten, von denen diese Arbeit nur auf die erste eingehen kann. Dazu kommt, dass das *lokale Kantenlängenverhältnis von 3-Bäumen* im Bereich der Graphentheorie angesiedelt ist und das Erarbeiten und Auswerten von Algorithmen zum Generieren von Zeichnungen auch durchaus seinen Reiz in seiner Ästhetik hat (Abbildung 3.6).

Inwieweit das lokale Kantenlängenverhältnis als Modell für beispielsweise wirtschaftliche oder soziale Zusammenhänge genutzt werden kann, ist nicht Teil dieser Arbeit, aber durchaus naheliegend. Bei der Betrachtung sozialer Netzwerke z.B. werden Personen mit einem bestimmten Chatverhalten in der Regel eher mit Personen zu tun haben, die wiederum ein ähnliches Verhaltensmuster ihren Bekannten und Freunden gegenüber aufweisen. Oder bei der Verteilung von Sendemasten wäre es wirtschaftlich effizienter, diese so zu verteilen, dass die benötigte Leistung zum Erreichen eines weiter entfernten Masts sich nicht zu sehr von den anderen benachbarten unterscheidet.

Zunächst wird in dieser Arbeit auf Schreibkonventionen und Annahmen eingegangen, die für diese nötig sind. Im anschließenden Kapitel wird die NP-Schwere von GLOBKLV (globales Kantenlängenverhältnis) und LOKKLV (lokales Kantenlängenverhältnis) gezeigt. Dafür wird für einen bestimmten Graphen gezeigt, dass dieser bei einem vorge-

gebenen Kantenlängenverhältnis quadratisch gezeichnet werden muss, damit dieser als Baustein für orthogonale Zeichnungen genutzt werden kann.

Im nächsten Kapitel wird ein nicht-optimaler Greedy-Algorithmus vorgestellt, der zwar sowohl in der Breiten- als auch Tiefensuche auf 3-Bäumen gute Ergebnisse lieferte, für diesen Algorithmus konnte aber keine Schranke für das lokale Kantenlängenverhältnis bewiesen werden. Der Algorithmus ist auf 3-Bäume ausgelegt bzw. implementiert, könnte aber prinzipiell auch für andere Graphklassen genutzt werden.

Falls an dieser Stelle ein wenig Prosa angebracht ist, würde ich gerne mit einem Zitat aus dem 2014 erschienenen Sciencefiction-Film *Interstellar* schließen: „Wir sind Entdecker, Rom. Das ist unser Boot.“

1.1 Schreibkonventionen, Problemstellung, Annahmen

Da der Begriff „lokales Kantenlängenverhältnis“ in dieser Arbeit noch öfter fallen wird, soll er in Volltext mit *lokKLV* abgekürzt werden. Als Funktion in Abhängigkeit eines Graphen oder einer Zeichnung wird es durch ein λ dargestellt (im Gegensatz zu ρ_l in [BFL20]), das globale Kantenlängenverhältnis wie gewohnt durch ein ρ .

Für einen Knoten v bezeichne l_v die Länge der längsten inzidenten Kante zu v , k_v die der kürzesten und x_v und y_v seine x- bzw. y-Koordinate. Für einen Graphen G bezeichne V_G dessen Knotenmenge und E_G dessen Kantenmenge. Für zwei Knoten a und b benutzen wir die Abkürzung ab für die Kante $\{a, b\}$.

In dieser Arbeit wird hauptsächlich das *lokKLV* von *3-Bäumen* untersucht. Des Weiteren gehen wir immer von planaren Graphen aus, da der jeweilige Graph sonst nicht kreuzungsfrei gezeichnet werden könnte und gerade die kreuzungsfreie Minimierung des *lokKLV* schwer und damit reizvoll ist. Nun sollen zunächst *k-Bäume* definiert werden. Diese sind wiederum für ein beliebiges $k \in \mathbb{N}_{\geq 2}$ definiert. Sowohl in der Arbeit von Blažej et al. [BFL20] als auch in dieser werden nur die Fälle $k = 2$ und $k = 3$ betrachtet.

Definition 1. *k-Bäume sind rekursiv definierte Graphen:*

- K_k ist ein *k-Baum*.
- Ist (v_1, v_2, \dots, v_k) eine *k-Clique* eines *k-Baums* G , mit $\{v_1, v_2, \dots, v_k\} \subseteq V_G$, so ist auch G' ein *k-Baum* mit $V_{G'} = V_G \cup \{w\}$ und $E_{G'} = E_G \cup \{wv_1, wv_2, \dots, wv_k\}$.

Bildlich gesprochen ist also ein Dreieck ein 3-Baum. Auf einen 3-Baum kann man einen neuen Knoten stapeln, indem man ihn ins Innere einer Facette legt und jeweils mit einer Kante zu den drei ihn umgebenden Knoten verbindet. An sich erfordert die Definition eines 3-Baums nicht, dass der neue Knoten im Inneren der Facette liegt, auf die er gestapelt wird, dies wird im Rahmen dieser Arbeit aber angenommen. Andernfalls wäre die entsprechende Zeichnung nicht planar, wenn der neue Knoten nicht gerade auf der äußeren Facette liegt, was bereits ausgeschlossen wurde. Und auch wenn der neue Knoten auf die äußere Facette gestapelt würde, könnte der so entstandene 3-Baum auch

produziert werden, wenn man nur ins Innere von Facetten stapelt: man fängt mit dem nun neu entstandenen Dreieck als Ursprungsfacette an und stapelt die anderen Knoten in dessen Inneres.

Definition 2. Das lokale Kantenlängenverhältnis $\lambda(\Gamma)$ einer planaren Zeichnung Γ eines planaren Graphen G ist definiert als $\lambda(\Gamma) =_{def} \max_{uv, vw \in E_G} |uv|/|vw|$.

Definition 3. Das lokale Kantenlängenverhältnis $\lambda(G)$ eines planaren Graphen G ist definiert als $\lambda(G) =_{def} \inf_{\Gamma} \lambda(\Gamma)$ mit Γ Zeichnung von G .

Das lokale Kantenlängenverhältnis beschreibt also für die Zeichnung eines Graphen das größte Verhältnis zweier adjazenter Kanten in dieser Zeichnung und für einen Graphen das kleinstmögliche lokale Kantenlängenverhältnis einer Zeichnung dieses Graphen.

Hinweis: In Abbildung 1.1 ist das *lokKLV* für eine Zeichnung von P_3 dargestellt. Für diesen Graph gilt, dass $\rho(P_3) = \lambda(P_3) = 1$ (die Kanten könnten ja auch gleichmäßig gezeichnet werden). Damit ist P_3 auch der kleinste Graph, für den eine Zeichnung Γ existiert mit $\rho(\Gamma) > \lambda(\Gamma)$. Damit nämlich $\rho(\Gamma') \neq \lambda(\Gamma')$ für eine Zeichnung Γ' gilt, muss es mindestens zwei verschiedene Kantenpaare geben (also mindestens drei Kanten), davon eines adjazent zueinander und das andere nicht. K_3 würde hier nicht funktionieren, da dann alle Kantenpaare adjazent wären.

Wir gehen o.B.d.A. davon aus, dass es in den betrachteten Graphen immer mindestens ein Paar adjazenter Kanten gibt, da das *lokKLV* sonst nicht definiert wäre. Entsprechend sei auch das *lokKLV* für eine einzelne Facette definiert, also nur die Kanten betreffend, die zu einem der Eckknoten der Facette inzident sind:

Definition 4. Das lokale Kantenlängenverhältnis $\lambda_f(\Gamma)$ einer Facette f mit den Eckknoten v_1, v_2 und v_3 in der planaren Zeichnung Γ eines planaren Graphen G ist definiert als $\lambda_f(\Gamma) =_{def} \max_{uv, vw \in \{av \in E_G \mid a \in V_G \wedge v \in \{v_1, v_2, v_3\}\}} |uv|/|vw|$.

O.B.d.A. werden ungerichtete, ungewichtete Graphen betrachtet. Ist im Zusammenhang ersichtlich, um welche implizit gemeinte Zeichnung Γ es geht, können $\lambda(\Gamma)$ und $\lambda_f(\Gamma)$ auch mit λ bzw. λ_f abgekürzt werden.

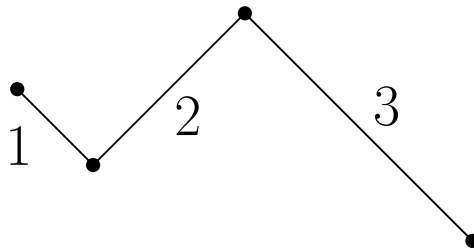


Abb. 1.1: Zeichnung Γ von P_3 mit $\rho(\Gamma) = 3$ und $\lambda(\Gamma) = 2$

2 Komplexität

Die Komplexität des Entscheidungsproblems, ob zu einem planaren Graphen eine Zeichnung mit vorgegebenem globalem oder lokalem Kantenlängenverhältnis existiert, ist bisher offen. In diesem Kapitel wird der Beweis ausgeführt, dass beide Probleme NP-schwer sind. Dieser Beweis nutzt weitestgehend die Idee von Patrignani [Pat01], der die NP-Schwere der flächenminimierenden Zeichnung von orthogonalen Repräsentationen von Graphen aufzeigte (ebenfalls als Entscheidungsproblem), indem er das bereits als NP-vollständig bekannte Problem *Satisfiability* (SAT) auf dieses reduzierte. In gleicher Weise wird SAT hier auf die Minimierung des Kantenlängenverhältnisses reduziert, indem zunächst über ein vorgegebenes Kantenlängenverhältnis eine pseudo-orthogonale Zeichnung erzwungen wird – „pseudo-orthogonal“ in dem Sinn, dass orthogonale Strukturen auf dem Gitter nachgebildet werden können, auch wenn diagonale Kanten enthalten sind. Für den Algorithmus von Patrignani werden als Eingabe beliebige Graphen erlaubt, in die dann Dummy-Knoten eingefügt werden, um Planarität zu erzwingen. Die Dummy-Knoten werden am Ende wieder entfernt. Wir hingegen betrachten weiterhin nur planare Graphen.

Hinweis: Nöllenburg [Nö05] zeigte die NP-Schwere der automatisierten Zeichnung von U-Bahn-Plänen. Dieser Beweis könnte für den NP-Schwere-Beweis von GLOBKLV und LOKKLV (wird im Folgenden noch definiert) evtl. einfacher umgewandelt werden als der von Patrignani. Diese Idee kam allerdings erst nach Fertigstellung dieser Arbeit auf.

Um die Komplexität der zwei genannten Probleme betrachten zu können, sollen diese Probleme zuerst definiert werden:

Definition 5 (GLOBKLV). *Gegeben sei ein planarer Graph G und eine Konstante $k \in \mathbb{R}_{\geq 1}$. Das Problem GLOBKLV ist die Entscheidungsfrage, ob es eine planare Zeichnung Γ von G gibt mit $\rho(\Gamma) \leq k$.*

Definition 6 (LOKKLV). *Gegeben sei ein planarer Graph G und eine Konstante $k \in \mathbb{R}_{\geq 1}$. Das Problem LOKKLV ist die Entscheidungsfrage, ob es eine planare Zeichnung Γ von G gibt mit $\lambda(\Gamma) \leq k$.*

In diesem Kapitel soll bewiesen werden, dass GLOBKLV und LOKKLV beide NP-schwer sind. Davor wird erst ein dazu nötiges Lemma bewiesen. Der in Abbildung 2.1 gezeigte Graph (im Folgenden als *Bauklotz* bezeichnet) muss demnach für $\rho(G) = \sqrt{2}$ oder $\lambda(G) = \sqrt{2}$ quadratisch gezeichnet werden. So können mit diesem dann die orthogonalen Strukturen aus dem Beweis von Patrignani nachgebildet werden.

Lemma 7. *Der in Abbildung 2.1 gezeigte planare Graph G (Wheel-Graph W_5) mit $V(G) = \{v_1, v_2, v_3, v_4, v_5\}$, $E(G) = \{v_1v_2, v_1v_3, v_1v_5, v_2v_4, v_2v_5, v_3v_4, v_3v_5, v_4v_5\}$ kann sowohl bei vorgegebenem $\rho(G) = \sqrt{2}$ als auch bei vorgegebenem $\lambda(G) = \sqrt{2}$ nur so gezeichnet werden, dass v_1, v_2, v_3 und v_4 ein Quadrat bilden und v_5 in der Mitte dieses Quadrats liegt.*

Beweis Lemma 7. Zuerst wird der Fall betrachtet, dass sich v_5 nicht an der äußeren Facette befindet. Jede der vier inneren Facetten bildet dann ein Dreieck (entspr. Abbildung 2.1), welches für $\rho (= \lambda) \leq \sqrt{2}$ keinen Innenwinkel größer 90° besitzen darf. Ist ein Innenwinkel 90° und sind die anliegenden Schenkel a und b gleich lang, so gilt $\rho = \sqrt{2}$ (Abbildung 2.2a). Erhöht man diesen Winkel zwischen den Seiten a und b nun, so wird nur die dritte Seite c verlängert und $\rho = \frac{c}{a} = \frac{c}{b} > \sqrt{2}$ (Abbildung 2.2b). Wird anschließend noch a verlängert, so erhöht sich $\rho = \frac{c}{b}$ weiter, da b konstant bleibt und sich c weiter verlängert. Gleiches gilt für eine Verlängerung von b . Wird a stattdessen verkürzt, so entspricht das für das KLV dem gleichen wie eine Verlängerung von b – der Winkel zwischen a und b bleibt gleich und das Verhältnis der Seite b zu a wird erhöht, entweder durch eine Verlängerung von b oder durch eine Verkürzung von a . Das Verhältnis $\rho = \frac{c}{a}$ erhöht sich also auch in diesem Fall. Gleiches gilt wieder entsprechend für eine Verkürzung von b .

Auf diese Weise kann jedes beliebige Dreieck mit einem Innenwinkel größer 90° erzeugt werden, also durch Erhöhen des 90° -Winkels und Verlängern oder Verkürzen einer der Seiten (abgesehen von Skalierung und Rotation, was für das KLV nicht relevant ist), wobei das KLV streng monoton erhöht wird. Da also keine der vier inneren Facetten des Bauklotzes einen Winkel größer 90° besitzen darf, müssen die Winkel inzident zu v_5 genau 90° sein. Auf gleiche Weise wie zuvor lässt sich argumentieren, dass keine der vier zu v_5 inzidenten Seiten bzw. Kanten verlängert oder verkürzt werden darf. Somit bilden v_1, v_2, v_3 und v_4 ein Quadrat und v_5 liegt in dessen Mitte.

Nun muss noch ausgeschlossen werden, dass v_5 an der äußeren Facette liegt. O.B.d.A. würden v_1, v_2, v_3 und v_4 dann wie in Abbildung 2.3a dargestellt eine „Brücke“ oder „Bogen“ bilden und v_5 würde oberhalb von dieser gesetzt und mit allen Knoten verbunden werden. Zunächst fixiere man v_2 und v_3 mit Abstand 1 (Abbildung 2.3b). Zu v_2 adjazente Knoten müssen dann zwischen zwei Kreisen mit Radien $\sqrt{2}$ und $1/\sqrt{2}$ um v_2 liegen, ebenso zu v_3 adjazente Knoten zwischen entsprechenden Kreisen um v_3 . Der

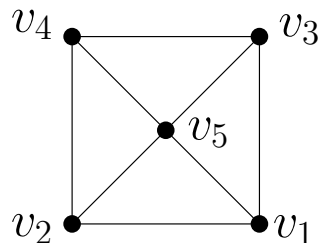


Abb. 2.1: Bauklotz-Graph

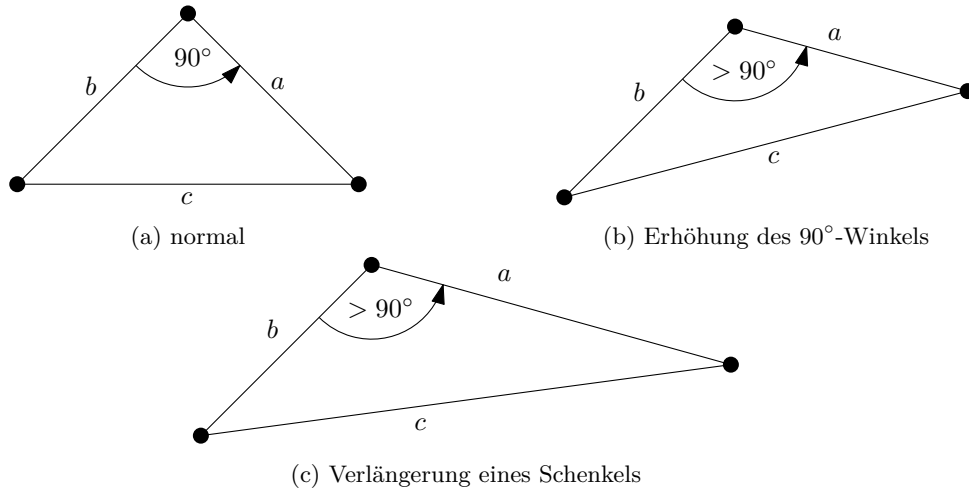


Abb. 2.2: Streng monotone Erhöhung des KLV

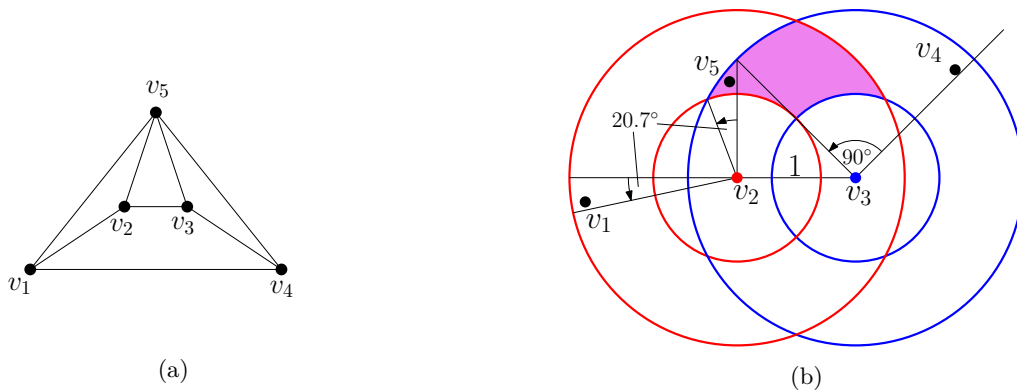


Abb. 2.3: v_5 an der äußeren Facette

Knoten v_5 müsste im Schnitt dieser beiden Mengen liegen (in Abbildung 2.3b violett dargestellt). Dann muss von v_1 und v_4 mindestens einer der beiden Knoten unterhalb von v_2 und v_3 liegen, ansonsten wäre die Zeichnung nicht planar (da v_5 an der äußeren Facette liegt, würde v_1v_4 sonst zwei der Kanten v_2v_3 , v_2v_5 und v_3v_5 schneiden). Nimm dafür o.B.d.A. v_1 unterhalb dieser Linie an. Wie bereits ausgeführt, darf kein Winkel größer 90° sein, v_5 müsste also in den kleinen violetten Bereich links oberhalb von v_2 gelegt werden. Der entsprechende Kreissektor um v_2 hat einen Winkel von etwa 20.7° . Genauso viel Spielraum hätte dann v_1 links unterhalb von v_2 . Dieser Winkel ergibt sich als $\frac{180^\circ}{\pi} \arctan 1/\sqrt{7}$ für den äußersten Punkt des violetten Bereichs $(-\frac{1}{4}, \frac{\sqrt{7}}{4})$, wenn v_2 in $(0,0)$ liegt und v_3 in $(1,0)$. Zuletzt müsste v_4 so gelegt werden, dass auch der Winkel zwischen v_3v_5 und v_3v_4 höchstens 90° beträgt. Unter all diesen Eingrenzungen von v_1 , v_4 und v_5 kann die Kante v_1v_4 nicht mehr planar gelegt werden, sie würde nämlich oberhalb von v_2v_3 , aber unterhalb von v_5 verlaufen und somit die Kante v_3v_5 und eine der Kanten v_2v_3 und v_2v_5 schneiden. \square

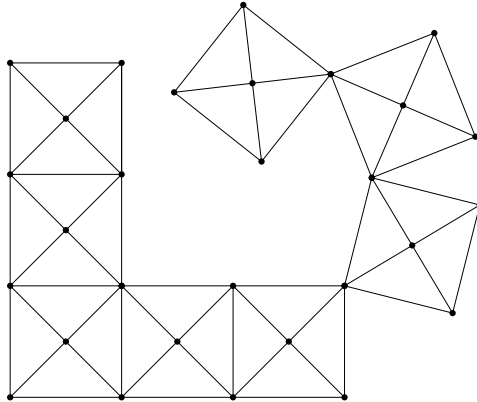


Abb. 2.4: Bauklotz-Verbindung

Korollar 8. *Verbindet man mehrere Bauklötze miteinander, indem man einen oder zwei Knoten des einen Bauklotzes mit einem bzw. zwei Knoten des anderen Bauklotzes identifiziert, bleiben Form und Größe der Bauklötze erhalten.*

Beweis. Bei einem Bauklotz angefangen haben die Außenseiten o.B.d.A. Länge 1 und die Diagonalstreben Länge $1/\sqrt{2}$. Über den einen oder die zwei verbindenden Knoten darf also auch die Diagonalstrebe des neuen Bauklotzes nicht kürzer werden als die Außenseite des alten (siehe Abbildung 2.4). \square

Hierbei unterscheiden wir zwischen *Bauklotz-Blöcken* (Verbindungen über zwei Knoten), die eine feste orthogonale Form haben und *Bauklotz-Ketten* (Verbindungen über einen Knoten), bei denen jedes Kettenglied zu seinem Vorgänger einen Winkel α haben kann mit $-90^\circ < \alpha < 90^\circ$.

Um die Struktur einer orthogonalen Zeichnung auf dem Gitter nachzubauen, muss allerdings noch beachtet werden, dass in der Nachbildung jeder Knoten und jede Kante der ursprünglichen Zeichnung nun die Breite und Höhe 1 besitzt. Damit sich die neuen Kanten und Knoten nicht überlappen, reicht es aber aus, alle ursprünglichen Positionen von Knoten und Kantenknicken zu verdoppeln, sodass zwischen zwei Knoten im Ursprungsgraph mindestens ein zusätzlicher Dummy-Bauklotz liegt, um die Kante zu repräsentieren (siehe Abbildung 2.5).

Reduktion von SAT auf OAC:

In diesem Abschnitt wird lediglich die Reduktion von SAT auf OAC (Orthogonal Area Compaction) beschrieben, im darauf folgenden die Details, die bei der Anpassung für GLOBKLV und LOKKLV beachtet werden müssen. Es sei also betont, dass in diesem Abschnitt noch kein eigener Beitrag enthalten ist, sondern nur die Idee von Patrignani [Pat01] präsentiert wird. Die entsprechenden Lemmas und Sätze werden genannt, aber lediglich deren Beweisidee skizziert. An dieser Stelle soll dem Leser die Grundidee des Beweises begreiflich werden, Details können in Patrignanis Arbeit eingesehen werden.

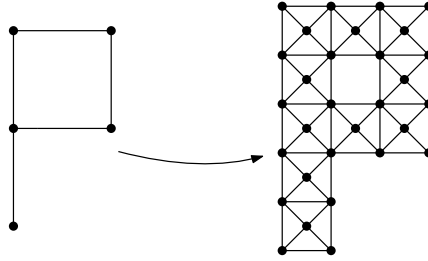


Abb. 2.5: Nachbildung einer orthogonalen Struktur mit Bauklötzen

Die Eingabe für OAC ist die orthogonale Repräsentation H eines Graphen G und eine Konstante K , d.h. für eine orthogonale Zeichnung von G weist H jedem Knoten (Maximalgrad 4) für jede inzidente Kante einen Winkel $\in \{0, \pi/2, \pi, 3\pi/2\}$ zu, in dem die Kante von dem Knoten abgeht. Statt einer entsprechenden Erweiterung auf Knicke werden Knicke im Folgenden durch Dummy-Knoten ersetzt. Entschieden werden soll die Frage, ob es für H eine Zeichnung auf dem Gitter mit einer Fläche kleiner gleich K gibt.

Die Eingabe für SAT ist eine aussagenlogische Formel ϕ über den Variablen x_1, \dots, x_n und Klauseln C_1, \dots, C_m in konjunktiver Normalform. Entschieden werden soll die Frage, ob ϕ erfüllbar ist.

Mit der *minimalen Fläche* einer Komponente ist im Folgenden die Fläche gemeint, die mindestens benötigt wird, um die gesamte Komponente auf dem Gitter zu zeichnen. Im Folgenden wird zur Erklärung des Beweises eine Formel mit $n = 3$ und $m = 2$ benutzt.

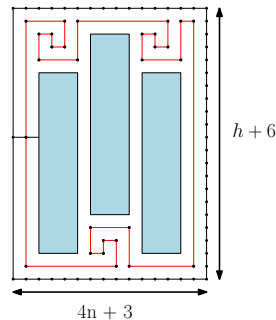


Abb. 2.6: Kolben-Gadget; rot: Gürtel, blau: Kolben, schwarz: Rahmen

Abbildung 2.6 zeigt ein *Kolben-Gadget*, bei dem jede Spalte, also jeder *Kolben* innerhalb des *Rahmens* entweder ganz nach oben oder unten geschoben ist, da für die *Wirbel* des *Gürtels*, der um die Kolben gelegt ist, innerhalb des $(4n + 3) \times (h + 6)$ großen Rahmens nur dann entsprechend entgegengesetzt oberhalb oder unterhalb der Kolben Platz ist (für Höhe h und Breite 3 der Kolben). Jeder Kolben steht für die Belegung *wahr* (oben) oder *falsch* (unten) einer der Variablen.

Lemma 9. *In jeder orthogonalen Zeichnung des Kolben-Gadgets auf dem Gitter mit (minimaler) Fläche $(4n + 3) \times (h + 6)$ gilt, dass jeder Kolben mit (minimaler) Fläche $3 \times (h - 1)$ entweder „oben“ oder „unten“ gezeichnet ist (entspr. Abbildung 2.6).*

Beweisidee. Durch die Reihenfolge der Links- und Rechtsknicke des Gürtels hat jeder seiner Wirbel mindestens die Höhe 3. So bleibt für die Kolben nur noch deren minimaler Platz unterhalb oder oberhalb der Wirbel übrig. \square

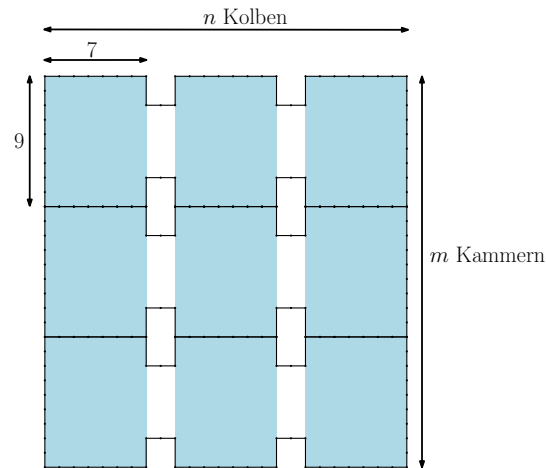


Abb. 2.7: In Kammern unterteilte Kolben

Die Kolben werden nun in m Kammern der Größe 9×9 unterteilt (Abbildung 2.7), eine für jede der m Klauseln, und mit den benachbarten Kammern des nächsten Kolbens verbunden, sodass jede Zeile eine Klausel darstellt. Die Kammern sind jetzt breiter als die zuvor gezeigten Rechtecke. Somit kommen im Gürtel nun zwei Wirbel auf eine Kammer statt nur einer. Die erste und die letzte Kammer benötigen keinen Übergang zum Vorgänger bzw. Nachfolger und sind somit jeweils um 1 schmaler (in Abbildung 2.7 wären das die Übergänge zu den Kammern des „nullten“ und vierten Kolbens). Hinweis: Lemma 9 müsste nun für eine minimale Fläche $(9n + 2) \times (h + 6)$ umformuliert werden.

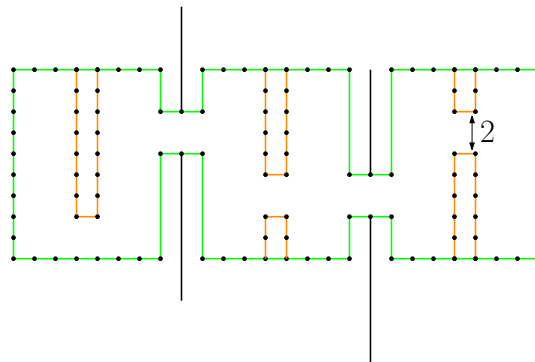


Abb. 2.8: Kammern (grün) mit Hindernissen (orange) und Scharnieren (schwarz)

Abbildung 2.8 zeigt, wie die Kammern nun erweitert werden. In jede **Kammer** wird ein **Hindernis** eingebaut entsprechend dem Vorkommen der Variabel in der Klausel – kommt

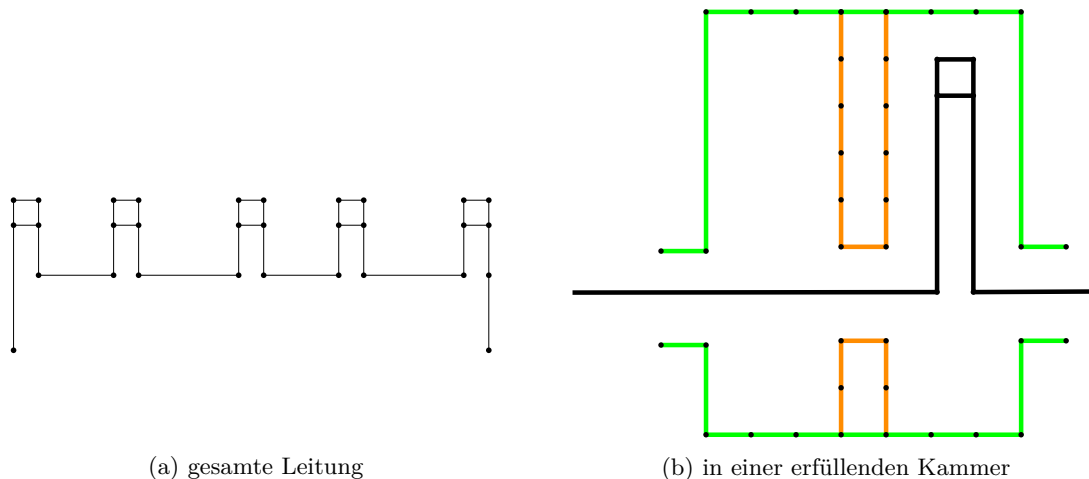


Abb. 2.9

die Variable gar nicht in der Klausel vor, ist die Lücke ganz unten (1. Kammer in Abbildung 2.8), kommt sie positiv vor, ist die Lücke etwas erhöht (2. Kammer) und kommt sie negativ vor, ist die Lücke am höchsten (3. Kammer). Durch *Scharniere* oberhalb und unterhalb der Verbindungen zwischen den Kammern (in Abbildung 2.8 schwarz) wird erzwungen, dass diese Verbindungen später die Höhe 2 haben, da sie sonst nicht mehr in den Rahmen passen würden. Die Verbindung kann dann nur entweder *wahr-konform* (auf Höhe der Lücke entspr. einer Kammer mit positiver Variable, siehe 2. Verbindung in Abbildung 2.8) oder *falsch-konform* (auf Höhe der Lücke entspr. einer Kammer mit negativer Variable, siehe 1. Verbindung) sein. Auch die Kammer an sich ist entweder wahr- oder falsch-konform, d.h. ihre beiden Verbindungen sind entweder beide oben oder beide unten. Benachbarte Kammern können demnach leicht versetzt zueinander sein, wenn eine wahr- und die andere falsch-konform ist (Abbildung 2.10).

Zuletzt wird eine *Leitung* aus $2n - 1$ A-Formen durch die Kammern einer Klausel gelegt (Abbildung 2.9a). Es muss also mindestens eine Kammer mit nur einer A-Form auskommen (Abbildung 2.9b).

Alle Klauseln werden zusammengefügt und zusammen mit dem Gürtel in einen Rahmen der Größe $(9n + 2) \times (9m + 7)$ gefasst.

Abbildung 2.10a zeigt alle Komponenten zusammen für die Variablenbelegung $x_1 = 1$, $x_2 = 1$, $x_3 = 0$ für die Formel $(\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_3)$. Abbildung 2.10b entspricht der nicht-erfüllenden Belegung $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, der Graph kann in der vorgegebenen Fläche nicht planar gezeichnet werden.

Lemma 10. *Eine wahr-konforme Kammer mit einem wahr-Hindernis oder eine falsch-konforme Kammer mit einem falsch-Hindernis kann eine oder zwei A-Formen der Leitung beinhalten. In allen anderen Fällen muss die Kammer zwei A-Formen beinhalten.*

Beweis. Damit auf eine A-Form verzichtet werden kann, muss der Durchgang der Kammer und der Durchgang des Hindernisses auf der gleichen Höhe sein. Das ist nur in den genannten Situationen der Fall. \square

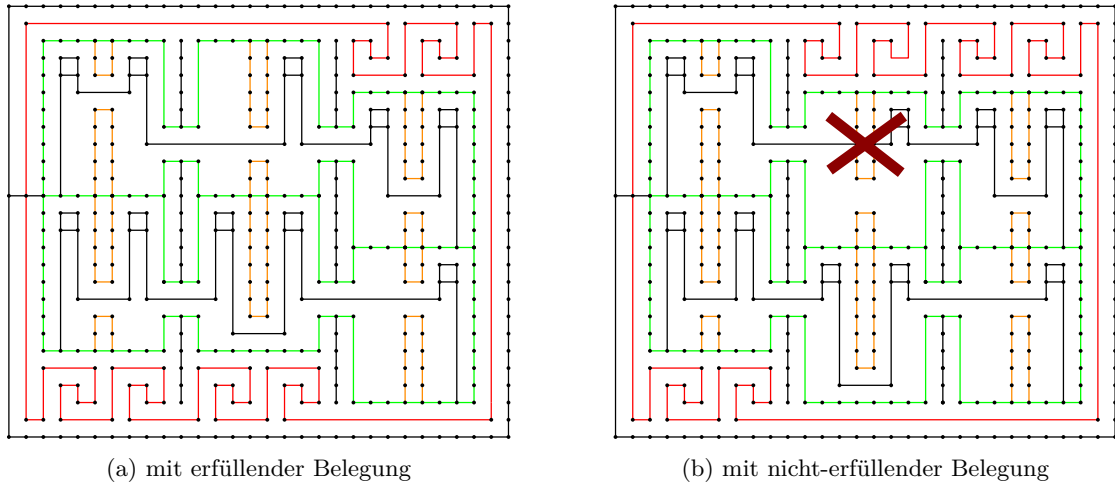


Abb. 2.10: Fertige Konstruktion; schwarz: Rahmen, Scharniere, Leitungen, grün: Kammern, rot: Gürtel, orange: Hindernisse

Lemma 11. *Eine Klausel-Konstruktion besitzt genau dann eine erfüllende Konfiguration T , wenn eine Übertragung der Variablenwerte von T mindestens ein wahres Literal in der entsprechenden Klausel erzeugt.*

Beweisidee. Eine Leitung besitzt nur $2n - 1$ A-Formen, d.h. mindestens eine Kammer muss mit nur einer A-Form auskommen – eine Kammer, deren entsprechende Variable die Klausel erfüllt. Würden in der Konfiguration mehrere Variablen die Klausel erfüllen, können die übrigen A-Formen beliebig auf die freien Räume in den Kammern verteilt werden.

Lemma 12. *Die eben beschriebene orthogonale Repräsentation zu einer Formel ϕ hat genau dann eine orthogonale Zeichnung auf dem Gitter der Fläche höchstens $(9n + 4) \times (9m + 7)$, wenn ϕ erfüllbar ist.*

Beweisidee. Das Lemma folgt aus den vorangegangenen. Wenn die Größe der äußeren Grenze wie definiert gewählt wird, passen die Kammern, wenn sie nach Lemma 11 erfüllbar sind, gerade unter oder über die Wirbel des Gürtels.

Die NP-Schwere von *OAC* folgt aus Lemma 12 und der Tatsache, dass die Reduktion in Polynomialzeit durchgeführt werden kann.

Reduktion von SAT auf GLOBKLV und LOKKLV:

Anhand von Abbildung 2.4 wurde bereits gezeigt, wie eine orthogonale Struktur mit Bauklötzen nachgebildet werden kann. Es soll die Konstruktion eines Graphen G_ϕ für den Beweis also so modifiziert werden, dass die entsprechende Formel ϕ genau dann erfüllbar ist, wenn G_ϕ eine Zeichnung Γ mit $\rho(\Gamma) = \lambda(\Gamma) = \sqrt{2}$ besitzt. Im Folgenden werden die Bauklötze in den Abbildungen aber nicht dargestellt. Die Knoten und Kanten

als solche abzubilden, wäre weniger ein Mehrgewinn für das Verständnis, sondern würde die Zeichnungen nur unübersichtlicher machen. Dabei sind aber noch folgende Punkte zu beachten:

Äußere Grenze schließen: In Patrignanis Konstruktion ist der linke Rand der äußeren Grenze nur durch einen einzigen Knoten abgeschlossen. Das reicht auch aus, wenn durch die Winkelangaben einer orthogonalen Repräsentation festgelegt ist, dass die anderen Konstrukte rechts von ihr liegen sollen. Ansonsten könnten diese aber auch auf die äußere Facette gelegt werden. Um dies zu verhindern, müsste noch eine zusätzliche Bauklotz-Wand aufgezogen werden (Abbildung 2.11). In den dadurch neu entstandenen Raum der Breite 1 passt keines der genannten Konstrukte hinein.

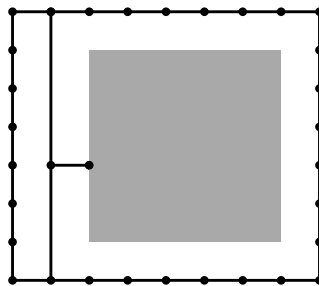


Abb. 2.11: Zusätzliche Wand auf der linken Seite

Vereinfachung – Stopper statt Wirbel: Patrignani musste etwas umständlich beweisen, dass durch die Reihenfolge der Links- und Rechtsknicke die Wirbel sowohl unten als auch oben platziert werden können, gleichzeitig aber immer „am Stück“ gezeichnet werden müssen. Darauf kann jetzt verzichtet werden, indem die Wirbel durch Stopper ersetzt werden. Diese müssen nicht massiv mit Bauklötzen ausgefüllt werden, es würde aber auch nicht schaden oder die Polynomialzeit der Konstruktion beeinflussen. Dafür muss dann aber auf die Dummy-Knoten in den Ecken des Gürtels verzichtet werden, damit die Stopper weiterhin nach oben oder unten verschoben werden können (Abbildung 2.12).

Flexible Kanten durch Bauklotz-Ketten ersetzen: Als nächstes müssen Kanten flexibler Länge durch Bauklotz-Ketten ersetzt werden. In Abbildung 2.13 sind diese rot dargestellt. Den größten Streckungsfaktor haben dabei die Kanten zwischen zwei aufeinanderfolgenden Stoppern. Im kürzesten Fall hat so eine Kante die Länge 2, im längsten $(9n - 8) + (9m + 5)$. Damit ergibt sich für alle Kanten ein Streckungsfaktor $\leq \frac{(9n-8)+(9m+5)}{2}$. Diese Kanten flexibler Länge durch einfache Kanten-Ketten zu ersetzen, hätte den Vorteil, dass diese leichter zusammengefaltet werden können. Allerdings könnten für die Problemstellung LOKKLV die Kanten mit jedem Glied der Kette um den Faktor $1/\sqrt{2}$ kürzer werden. Dadurch könnten die Stopper so klein werden, dass sie ihren Zweck nicht mehr erfüllen, nämlich die Position der Kolben auf *wahr* oder *falsch* zu erzwingen. Daher müssen Bauklotz-Ketten verwendet werden.

In diesem Fall muss aber erst Platz zum Zusammenfalten geschafft werden, um also z.B. im Fall der Kette zwischen zwei Stoppern diese von einer Länge $(9n - 8) + (9m + 5)$

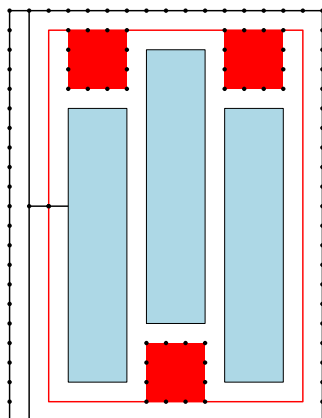


Abb. 2.12: Stopper statt Wirbel

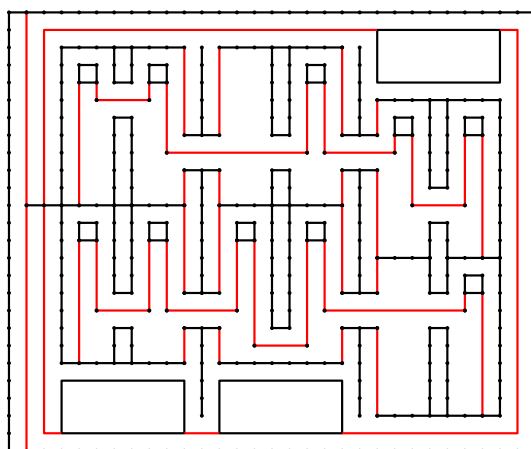


Abb. 2.13: Kanten flexibler Länge

auch auf eine Strecke der Länge 2 und Höhe 1 stauchen zu können. Dazu soll der Graph vor dem Einfügen der Ketten erst vergrößert werden, d.h. aus mehr Bauklötzen aufgebaut werden, damit die Ketten im Verhältnis feiner werden und enger zusammengefaltet werden können.

Abbildung 2.14 zeigt eine Möglichkeit der Kettenfaltung. Ob man die Faltung einer solchen Kette auch platzsparender umsetzen könnte, ist für diesen NP-Schwere-Beweis egal, solange nur polynomieller Platz verbraucht wird und eine polynomiell begrenzte Anzahl Bauklötze benötigt wird. Diese Faltung hat außerdem den Vorteil, dass die Verbindung zwischen zwei Bauklötzen immer auf mittlerer Höhe der Bauklötze besteht und es daher egal ist, ob eine Zeile der Kette eine gerade oder ungerade Anzahl Bauklötze enthält.

Der benötigte Platz zum Falten in diesem Beispiel ist für ein $\varepsilon_1 > 0$ in der Höhe begrenzt durch $(2 + \varepsilon_1)$ mal die Anzahl Zeilen und für ein $\varepsilon_2 > 0$ in der Breite durch $8\sqrt{2} + 2 + \varepsilon_2 < 14$. Da hier acht Bauklötze pro Zeile „verbaut“ werden können, wird der

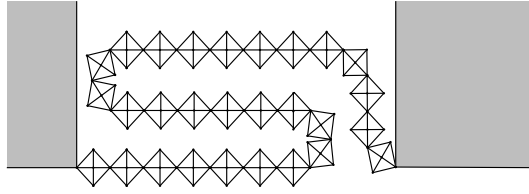


Abb. 2.14: Falten einer Bauklotz-Kette

Graph unter Berücksichtigung des o.g. größten Streckungsfaktors um das $\frac{(9n-8)+(9m+5)}{16}$ -fache vergrößert, mindestens aber um das 14-fache, um Platz für mindestens eine Zeile zu schaffen. Erst dann werden die Bauklotz-Ketten in der Länge eingefügt, die die jeweilige flexible Kante für ihre längste Strecke braucht.

A-Formen durch L-Formen ersetzen: Wie in Abbildung 2.15 illustriert, ist die Flexibilität des Graphen allerdings zu hoch. Es könnten durch Verwendung der Bauklotz-Ketten nämlich A-Formen durch Kammern geführt bzw. eingespart werden, sodass eine Kammer die Klausel erfüllen würde, wo die entsprechende Variable das nicht rechtfertigt.

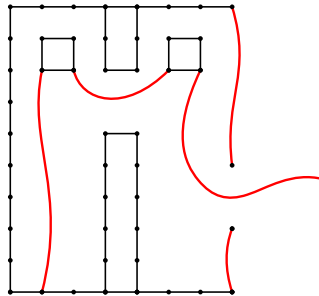


Abb. 2.15: Zu hohe Flexibilität

Die A-Formen werden nun durch jeweils zwei L-Formen ersetzt, bei denen sichergestellt wird, dass diese nur an den dafür vorgesehenen Stellen platziert werden können – auch bei flexiblen Bauklotz-Ketten. Die Kante zwischen den beiden L-Formen in Abbildung 2.16b muss auch eine flexible Länge haben für den Fall, dass z.B. eine wahr-konforme Kammer ein falsch-Hindernis enthält und sich die L-Formen daher auf verschiedenen Höhen befinden.

Im Gegensatz zur Darstellung in Abbildung 2.16 sollen die L-Formen aber den vorhandenen Platz weitmöglichst ausnutzen und wie in Abbildung 2.17 gezeigt erweitert werden, sodass die L-Formen weder ineinander gespiegelt werden noch um 90° oder 180° gedreht und so anders eingesetzt werden können.

Die einzigen ehemaligen flexiblen Kanten der Kammern, die nun durch Bauklotz-Ketten ersetzt sind, sind in Abbildung 2.17 als Kurven gekennzeichnet. Da der Graph (einschließlich der Scharniere) zuvor um das mindestens 14-fache vergrößert wurde und die erwähnten Bauklotz-Ketten weiterhin einen Abstand von 1 sowohl zu den Scharnieren als auch zu den L-Formen haben, besteht durch diese nicht genug Flexibilität, um die

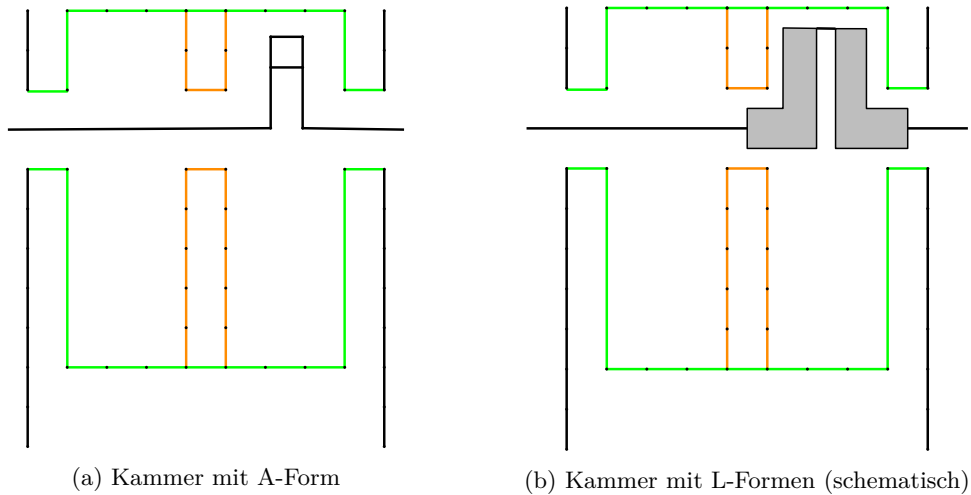


Abb. 2.16

L-Formen an einer anderen als die ihnen zugedachten Lücken zu platzieren.

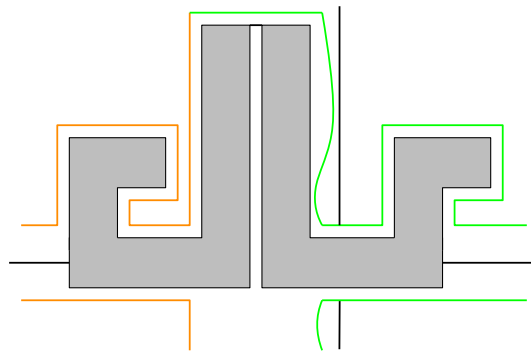


Abb. 2.17: L-Formen mit max. Größe; flexible Bauklotz-Ketten neben Scharnieren betont

Schließlich können alle Lemmas des ursprünglichen Beweises wiederverwendet werden:

Lemma 13 (entspr. Lemma 10). *Eine wahr-konforme Kammer mit einem wahr-Hindernis oder eine falsch-konforme Kammer mit einem falsch-Hindernis kann bei einer Bauklotz-Konstruktion zwei oder vier L-Formen der Leitung beinhalten. In allen anderen Fällen muss die Kammer vier L-Formen beinhalten.*

Beweis. An den kombinatorischen Möglichkeiten, die L-Formen zu verteilen, ändert sich zum vorigen Beweis nichts. Durch die L-Formen ist sichergestellt, dass die Linien, die im Beweis von Patrignani gezeichnet wurden, von den gleichen Bauklotzketten zwischen zwei L-Formen überlaufen werden wie die Kante zwischen den beiden entsprechenden Knoten der orthogonalen Konstruktion. Durch die Verwendung von Bauklotz-Ketten könnte allerdings noch ein Problem entstehen. Bei OAC kann eine A-Form nur übersprungen werden, wenn die Öffnung des Hindernisses und die Verbindung zur vorigen

oder nächsten Kammer auf der gleichen Höhe sind, da die Kante, die durch die Kammer geführt werden soll, als waagrecht definiert wird (Abbildung 2.18a). Da Bauklotz-Ketten in jedem Glied der Kette aber auch in ihrem Winkel flexibel sind, könnte ein Hindernis auch umgangen werden (Abbildung 2.18b). Wie Abbildung 2.18c zeigt, kann dies aber verhindert werden, wenn die Kette nach dem Vergrößern des Graphen mit genau der Länge eingesetzt wird, dass die entsprechende nächste A- bzw. L-Form erreicht werden kann. Für das Umgehen eines Hindernisses wäre die Kette dann zu kurz.

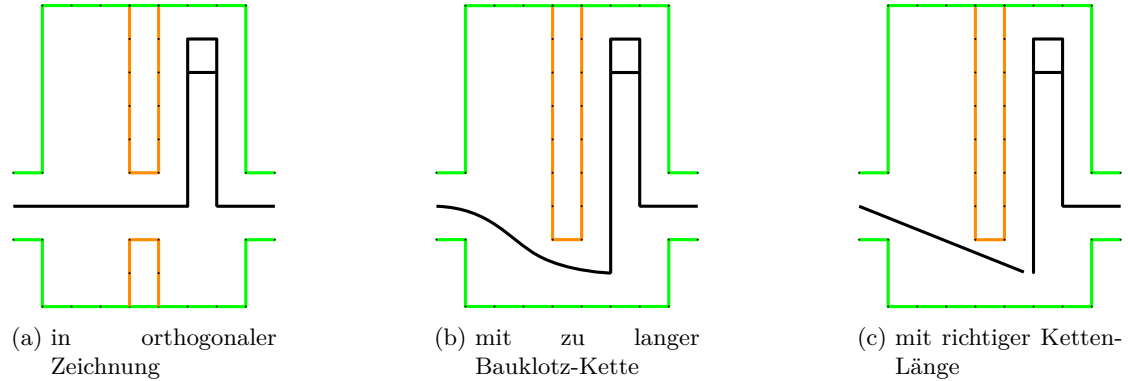


Abb. 2.18: Möglichkeiten, ein Hindernis zu umgehen

□

Lemma 14 (entspr. Lemma 11). *Eine Klausel-Konstruktion in einer Bauklotz-Konstruktion besitzt genau dann eine erfüllende Konfiguration T , wenn eine Übertragung der Variablenwerte von T mindestens ein wahres Literal in der entsprechenden Klausel erzeugt.*

Beweis. Am Widerspruchsbeweis von Patrignani ändert sich unter der Verwendung von Lemma 14 statt Lemma 10 nichts. □

Lemma 15 (entspr. Lemma 12). *Die eben beschriebene Bauklotz-Konstruktion zu einer Formel ϕ hat genau dann eine Zeichnung Γ mit $\rho(\Gamma) = \lambda(\Gamma) = \sqrt{2}$, wenn ϕ erfüllbar ist.*

Beweis. Statt des Beweises der an voriger Stelle zur Argumentation bezüglich der benötigten Fläche geführt werden musste, folgt dieses Lemma aus den vorangegangenen Lemmas und der Tatsache, dass über die Bauklötze hinweg sowohl ein konstantes *globKLV* und *lokKLV* $\rho(\Gamma) = \lambda(\Gamma) = \sqrt{2}$ für eine Zeichnung Γ garantiert wird. Da die Größen der Bauklotz-Blöcke also weiterhin den Größen der entsprechenden Komponenten auf dem Gitter entsprechen, passt die Konstruktion unter den gleichen Umständen wie in OAC in den Rahmen – wenn unter Verwendung von Lemma 14 die Konfiguration alle Klauseln erfüllt. □

Satz 16. *GLOBKLV und LOKKLV sind NP-schwer.*

Beweis. Die NP-Schwere von GLOBKLV und LOKKLV folgt aus Lemma 15 und der Tatsache, dass die Reduktion in Polynomialzeit umgesetzt werden kann.

Bis hierhin wurde der Beweis der NP-Schwere von GLOBKLV und LOKKLV unter der Annahme durchgeführt, dass die irrationale Zahl $\sqrt{2}$ zur Bestimmung der Knotenpositionen exakt berechnet werden kann. Man könnte aber ebenso einfache Quadrate als Bauklötze verwenden, also bei denen der Mittelknoten fehlt. An der Konstruktion würde das nichts ändern.

Die Knotenpositionen könnten aber aufgrund des Zusammenfaltens der Bauklotzketten auf nicht-rationalen Koordinaten liegen. Diese Positionen müssen aber nicht exakt berechnet werden, es muss nur sichergestellt werden, dass dazu genug Platz vorhanden wäre. \square

3 Greedy-Algorithmus

3.1 Idee, Algorithmus

Der erste Ansatz, der sich im Rahmen dieser Masterarbeit zur Minimierung des *lokKLV* ergab, war ein Greedy-Algorithmus für 3-Bäume, der beim Stapeln eines neuen Knotens in die Facette eines 3-Baums die in dieser Facette zu diesem Zeitpunkt bestmögliche Position wählt. Dabei bestimmen sechs Werte bzw. Funktionen das *lokKLV*, das in Abhängigkeit der neuen Knotenposition erzielt wird: für alle drei einschließenden Knoten a , b und c muss die Position des neuen Knotens w jeweils sowohl im Verhältnis zu der längsten als auch zu der kürzesten bisherigen Kante von $v \in \{a, b, c\}$ berücksichtigt werden. Die Funktion für das Verhältnis zur längsten Kante von v ist dementsprechend definiert als:

$$L_v(w) =_{def} \frac{l_v}{\sqrt{(x_v - x_w)^2 + (y_v - y_w)^2}}$$

Das Verhältnis zur kürzesten Kante von v ist definiert als:

$$K_v(w) =_{def} \frac{\sqrt{(x_v - x_w)^2 + (y_v - y_w)^2}}{k_v}$$

Das *lokKLV* in Abhängigkeit des neuen Knotens w ist dann in jedem Punkt das Maximum dieser sechs Funktionen, welches es innerhalb des Dreiecks der einschließenden Facette f zu minimieren gilt (siehe Abbildung 3.1):

$$\Lambda_f(w) =_{def} \max\{L_a(w), L_b(w), L_c(w), K_a(w), K_b(w), K_c(w)\}$$

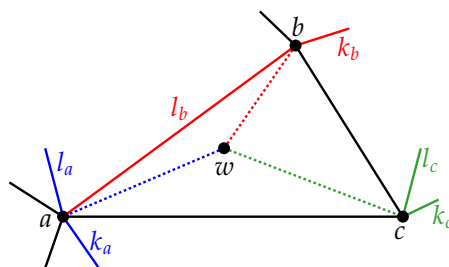


Abb. 3.1: Parameter zur Berechnung des $\Lambda_f(w)$

Das *lokKLV* muss dann aktualisiert werden auf $\lambda \leftarrow \max\{\lambda, \Lambda_f(w)\}$. Abbildung 3.2a zeigt die beiden Funktionen L_v und K_v für einen einzelnen Knoten v . Innerhalb eines Dreiecks wird dann also das minimale Maximum über sechs dreidimensionale Funktionen gesucht (Abbildung 3.2b). Die drei gelb dargestellten Funktionen (im Folgenden „Trichter“ genannt) berechnen das Verhältnis zur jeweils längsten Kante des Eckpunkts, die drei blauen dargestellten („Kegel“) das Verhältnis zur jeweils kürzesten Kante.

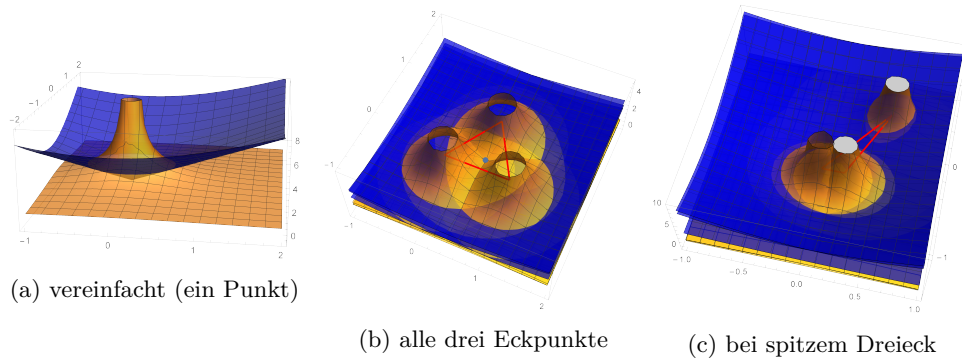


Abb. 3.2: Finde minimales Maximum innerhalb des Dreiecks

Anmerkung: die Kegel in Abbildung 3.2b sind so flach, dass sie innerhalb des rot hervorgehobenen Dreiecks nie ein Maximum darstellen, also praktisch ignoriert werden könnten. Dies ergaben auch Tests (Abschnitt 3.2), sowohl in der Tiefen- als auch in der Breitensuche. Gerade für die Breitensuche ist es aber schwer zu beweisen, dass die Kegel generell ignoriert werden können, wenn man mit einem gleichseitigen Dreieck beginnt, da die kürzeste Kante eines Eckpunkts ja durch das Stapeln eines Knotens in einer benachbarten Facette bereits verkürzt worden sein könnte. Wählt man als Anfangsdreieck eines mit einer entsprechend kurzen Seite, kommen die Kegel natürlich zum Tragen (siehe Abbildung 3.2c). Des Weiteren hätten die Kegel aber nie eine direkte Auswirkung auf das *lokKLV*, da immer eine der beiden angrenzenden Dreiecksseiten länger als die Distanz zum neuen Knoten w sein muss und damit schon vorher ein größeres *lokKLV* verursacht hätte. Je nach „Greedy-Mentalität“ könnte man die Trichter also ohnehin vernachlässigen.

Zur Berechnung des minimalen Maximums kommen 43 Kandidatenpunkte infrage:

- ein Kandidat für den Schnitt aller drei Trichter
- sechs Kandidaten für den Schnitt zweier Trichter mit einer Dreiecksseite (drei Paare von Trichtern, zwei Schnitte pro Trichterpaar mit Dreieck)
- 36 Kandidaten für den Schnitt zweier Trichter mit einem Kegel (drei Paare von Trichtern, drei Kegel, vier Schnitte pro Trichterpaar mit Kegel)

Alle diese Kandidaten lassen sich in $O(1)$ berechnen:

O.B.d.A. wird nun die Berechnung des Schnitts der beiden Trichter von b und c mit

dem Kegel von a erläutert. Der Schnitt zweier Trichter ist eine invertierte Parabel (im Zweidimensionalen von der Form $\frac{1}{p_2x^2+p_1x+p_0c}$ für entsprechende Parameter p_0, p_1, p_2). Um für den Schnitt der Trichter von b und c nicht umständlich die entsprechenden Funktionen

$$\frac{l_b}{\sqrt{(x_b - x)^2 + (y_b - y)^2}} = \frac{l_c}{\sqrt{(x_c - x)^2 + (y_c - y)^2}}$$

gleichsetzen zu müssen mit parametrisiertem

$$y = \frac{y_c - y_b}{x_c - x_b}(x - x_b) + y_b$$

und nach x aufzulösen, wird das Koordinatensystem wie folgt transformiert:

- Translation($-x_b, -y_b$)
- Rotation($-\frac{y_c}{x_c}$)
- Streckung($\frac{1}{l_b}$)

b wird also in den Nullpunkt gesetzt, c auf die x-Achse rotiert und $l_b = 1$ gesetzt (siehe Abbildung 3.3). Im Folgenden beziehen sich die genannten Variablen auf deren Werte nach dieser Transformation. Somit wird $x_b = y_b = y_c = 0$ und $l_b = 1$ gesetzt und der Funktionswert der invertierten Schnittparabel ist nur noch durch ihre x-Koordinate x_T bestimmt. Für x_T können sich zwei Werte $\frac{x_c \pm l_c x_c}{1 - l_c^2}$ ergeben, von denen einer zwischen 0 und x_c liegen muss. Dieser wird für x_T gewählt. Für den Schnitt mit dem Kegel von a wird die Trichterfunktion von b dann mit der Kegelfunktion gleichgesetzt und x_T eingesetzt.

$$\frac{\sqrt{(x_T - x_a)^2 + (y - y_a)^2}}{l_a} = \frac{1}{\sqrt{x_T^2 + y^2}}$$

y hängt nun also (wegen der zuvor durchgeführten Transformation) nur noch von den Parametern von a ab. Nach Auflösen nach y ergeben sich dann bis zu 4 Kandidaten $(x_T, y_{T_i}), i \in \{1, 2, 3, 4\}$. Für alle Kandidaten muss dann noch überprüft werden, ob sie innerhalb des Dreiecks liegen. Für die entsprechende Facette f ist λ_f dann das minimale Maximum der z -Werte der validen Kandidatenpunkte. Andere Punkte als diese Schnittpunkte können für ein minimales Maximum nicht infrage kommen, da eine invertierte Parabel keine lokalen Minima besitzt.

Die anderen Schnittpunkte (drei Trichter; zwei Trichter und Dreieck) lassen sich jeweils durch das Lösen von Polynomen zweiten Grades finden. Die Berechnung des Schnittpunkts zweier Trichter mit einem Kegel dagegen stellt hier durch das Lösen eines Polynoms vierten Grades auch die komplizierteste dar. In der in Abschnitt 3.2 präsentierten Implementierung wurden dazu 14 ineinander verschachtelte Terme benötigt (Überprüfung der Kandidatenpunkte nur angedeutet):

Algorithmus 1: SchnittMitKegel(Vertex v_3 , Double x_T , List<Vertex> $candidates$)

Eingabe: dritter Knoten v_3 , x-Koordinate der invertierten Parabel x_T

Ausgabe: Kandidatenpunkte werden in $candidates$ übergeben

$$x_3 = v_3.x$$

$$y_3 = v_3.y$$

$$k_3 = v_3.k$$

$$b = -2y_3$$

$$c = x_3^2 - 2x_3x_T + 2x_T^2 + y_3^2$$

$$d = -2x_T^2y_3$$

$$e = -k_3^2 + x_3^2x_T^2 - 2x_3x_T^3 + x_T^4 + x_T^2y_3^2$$

$$v_1 = c^2 - 3bd + 12e$$

$$v_2 = 2c^3 - 9bcd + 27d^2 + 27b^2e - 72ce$$

$$v_3 = \sqrt[3]{2v_1}$$

$$v_4 = (v_2 + \sqrt{-4v_1^3 + v_2^2})^{\frac{1}{3}}$$

$$v_5 = \frac{v_3}{3v_4} + \frac{v_4}{3\sqrt[3]{2}}$$

$$v_6 = \frac{b^2}{4} - \frac{2c}{3}$$

$$v_7 = \sqrt{v_6 + v_5}$$

$$v_8 = \frac{b^2}{4} - \frac{4c}{3} - v_5$$

$$v_9 = \frac{-b^3 + 4bc - 8d}{4v_7}$$

$$v_{10} = -\frac{b}{4}$$

$$y_{T_1} = v_{10} - \frac{v_7 + \sqrt{v_8 - v_9}}{2}$$

$$y_{T_2} = v_{10} - \frac{v_7 - \sqrt{v_8 - v_9}}{2}$$

$$y_{T_3} = v_{10} + \frac{v_7 - \sqrt{v_8 - v_9}}{2}$$

$$y_{T_4} = v_{10} + \frac{v_7 + \sqrt{v_8 - v_9}}{2}$$

prüfeKandidat(y_{T_1} , $candidates$)

prüfeKandidat(y_{T_2} , $candidates$)

prüfeKandidat(y_{T_3} , $candidates$)

prüfeKandidat(y_{T_4} , $candidates$)

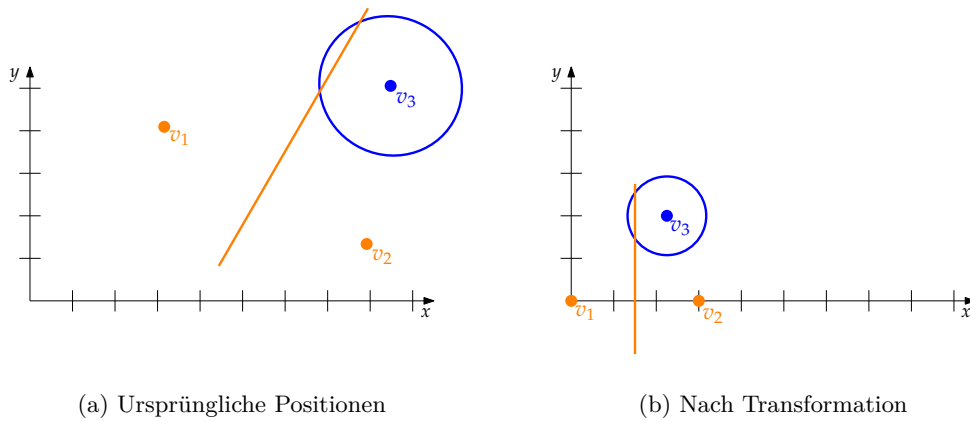


Abb. 3.3: Transformation des Koordinatensystem zur Berechnung der Schnitte

Tiefe	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Laufzeit	1	1	5	6	6	8	11	50	64	209	372	917	2555	7518	24193
<i>lokKLV</i>	2.0	4.0	5.0	6.9	9.3	10.7	13.3	17.5	22.4	27.5	31.7	39.3	48.0	58.8	70.8

Abb. 3.4: Ergebnisse der Breitensuche, Laufzeit in *ms*

3.2 Implementierung

Der in diesem Kapitel präsentierte Greedy-Algorithmus wurde in Java implementiert und für ein gleichseitiges Startdreieck auf Tiefensuche und auf Breitensuche laufen gelassen. Das verwendete Laptop hatte einen Intel-Core-i3-8130U-Prozessor mit 2.20 GHz, 8 GB Arbeitsspeicher, und lief auf der JRE System Library [JavaSE-1.7] unter Windows 10.

3.2.1 Breitensuche

Die Breitensuche konnte nur bis zu einer Baumtiefe $t = 15$ durchgeführt werden, da die Anzahl der Knoten mit der Baumtiefe exponentiell auf $3 + \sum_{i=0}^{14} 3^i = 7.174.456$ ansteigt und ab $t \geq 16$ der Stack überlief.

Abbildung 3.5 lässt vermuten, dass das *lokKLV* quadratisch mit der Baumtiefe wächst, also $\text{lokKLV} \in \Theta(\log^2 n)$ für die Anzahl n der Knoten.

3.2.2 Tiefensuche

Auch wenn intuitiv die Vermutung naheliegt, dass es egal sein müsste, welche der drei Facetten man für die Tiefensuche auswählt, so macht das insofern einen Unterschied, da jede Facette seine Eckknoten lokal als v_1 , v_2 und v_3 definiert. Sei nun f_{12} die Facette, die nach dem Stapeln eines neuen Knotens v' zwischen v_1 , v_2 und v' aufgespannt wird und f_{13} , f_{23} die entsprechend anderen Facetten. Der Greedy-Algorithmus wurde nun so implementiert, dass die Eckknoten einer neuen Facette wie folgt identifiziert werden:

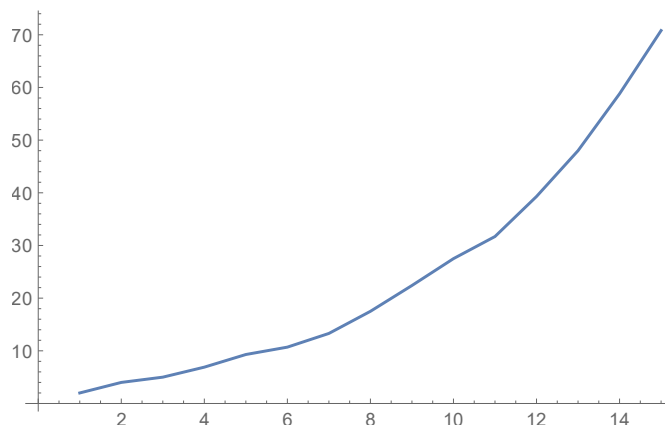


Abb. 3.5: Entwicklung des *lokKLV* bei Breitesuche

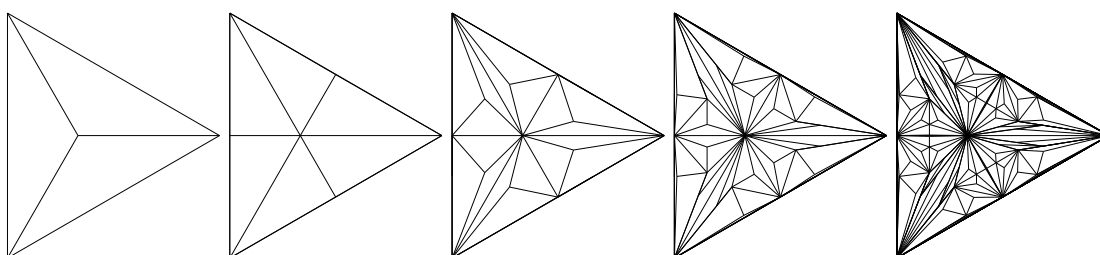


Abb. 3.6: Zeichnungen bei Breitesuche und Baumtiefen 1 bis 5

- $f_{12}: (v'_1, v'_2, v'_3) \leftarrow (v_1, v_2, v')$
- $f_{13}: (v'_1, v'_3, v'_3) \leftarrow (v_1, v_3, v')$
- $f_{23}: (v'_1, v'_2, v'_3) \leftarrow (v_2, v_3, v')$

Das führt dazu, dass eine Tiefensuche über f_{12} zu einer Zeichnung führt, in der bereits der erste gestapelte Knoten an die Kante v_1v_2 gelegt wird und alle weiteren Knoten versetzt jeweils in die dadurch entstehende extrem dünne Facette hineingelegt werden (siehe Abbildung 3.7) – ein Knoten dürfte an sich natürlich nicht *auf* einer anderen Kante liegen, das ist allerdings nur eine Frage des entsprechenden ε , das man für den Abstand zur Kante wählt, das Problem bleibt das gleiche. Das *lokKLV* bleibt aber ab Baumtiefe $t = 2$ konstant bei 3.95. Die Laufzeit steigt konstant mit Anzahl der Knoten bzw. Baumtiefe t und liegt für $t = 1\,000\,000$ bei 3.6s, ebenso verhält sich die Laufzeit für die Tiefensuche über f_{13} oder f_{23} .

Bei der Tiefensuche über die Facette f_{13} hingegen nähert sich der jeweils neu gestapelte Knoten immer näher dem Eckknoten v_1 des obersten Levels an (siehe Abbildung 3.8). Das *lokKLV* liegt dabei stets knapp unter der Anzahl der Knoten, für $n = 1\,000\,000$ bei 999 997.9.

Bei der Tiefensuche über die Facette f_{23} erfolgt die Wahl der neuen Facette spiralförmig (siehe Abbildung 3.9). Ab Baumtiefe $t = 25$ liegt das *lokKLV* konstant bei 48.6.

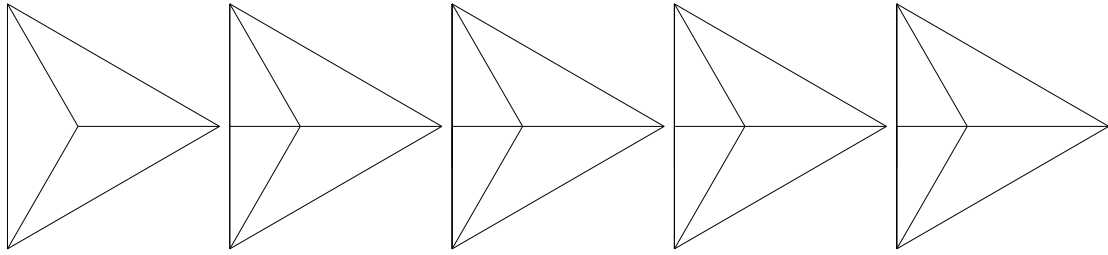


Abb. 3.7: Zeichnungen bei Tiefensuche über Facette f_{12} und Baumtiefen 1 bis 5

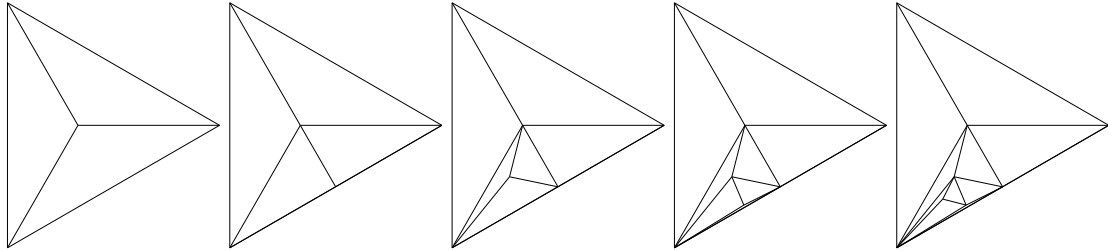


Abb. 3.8: Zeichnungen bei Tiefensuche über Facette f_{13} und Baumtiefen 1 bis 5

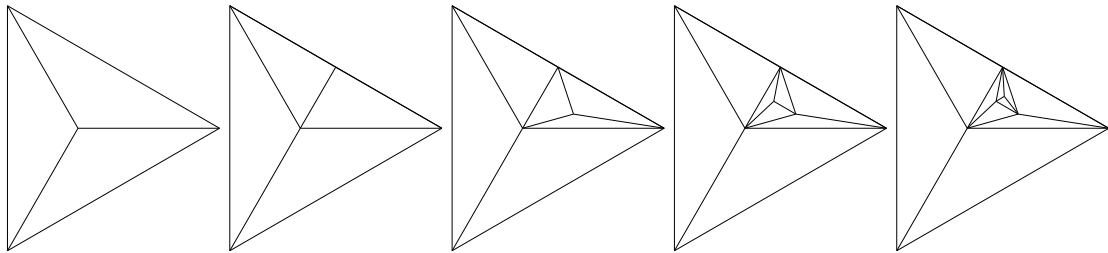


Abb. 3.9: Zeichnungen bei Tiefensuche über Facette f_{23} und Baumtiefen 1 bis 5

Wie Abbildung 3.10 zeigt, ist der Greedy-Algorithmus nicht optimal. Das Verhältnis der grün markierten Kanten und das der rot markierten Kanten entspricht den Trichterfunktionen, an denen das *lokKLV* entsteht. Das *lokKLV* könnte in dieser Facette und damit für den gesamten Graphen verbessert werden, wenn der mittlere Knoten in Pfeilrichtung verschoben würde.

Hinweis (nach Abschluss dieser Arbeit): Um Rundungsfehler zu minimieren, hätte der Algorithmus besser implementiert werden können. Für das Dreieck, das der Facette auf oberster Bauebene entspricht, wurden Knotenpositionen $v_1 = (0, 0)$, $v_2 = (0, 1)$, $v_3 = (\frac{\sqrt{3}}{2}, 0.5)$ gewählt, die einem Abstand von 1 entsprechen. Je tiefer der Baum wird, desto näher liegen die Knoten beieinander und desto ungenauer kann das *lokKLV* berechnet werden. Daher hätte man zu Beginn den Maximalwert der Double-Variablen besser ausnutzen können.

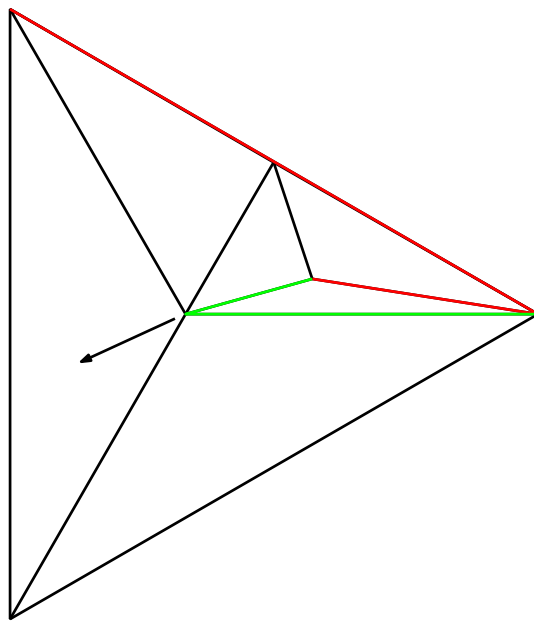


Abb. 3.10: Verbesserung des Greedy-Ergebnisses durch Verschiebung des mittleren Knotens

4 Fazit

Es konnte gezeigt werden, dass die Entscheidungsprobleme GLOBKLV und LOKKLV NP-schwer sind, allerdings nicht, ob diese in NP enthalten sind. Die Vermutung liegt nahe, dass sie es nicht sind, da die genannten Probleme nicht auf das Gitter begrenzt sind und es i.A. überabzählbar unendlich viele mögliche Zeichnungen eines Graphen gibt. Es bleibt also offen, ob GLOBKLV und LOKKLV in NP liegen. Auch zu der Computergestützten Umsetzung der Beweis-Konstruktion gibt es wegen der irrationalen Zahl $\sqrt{2}$ bereits genannte Schwierigkeiten. Es müsste zunächst untersucht werden, wie genau $\sqrt{2}$ angenähert werden muss, um eine korrekte Konstruktion zu erhalten.

Der präsentierte Greedy-Algorithmus stellt eine nach Tests gute Heuristik auf 3-Bäumen dar. Dieser Algorithmus ist aber nicht auf die Anwendung auf 3-Bäume beschränkt. In den Ergebnissen der Tiefensuche konnte betrachtet werden, in welchen Fällen das *lokKLV* wesentlich schlechter wurde. Unter dieser Kenntnis könnte der Algorithmus auch auf Graphklassen angewendet werden, wo diese Fälle vermieden werden können. Des Weiteren konnte keine Schranke für die Qualität des Algorithmus bewiesen werden, was zu der Frage führt, ob das dadurch erzeugte *lokKLV* für Graphen mit n Knoten in $O(\log^2 n)$ liegt, wie die Testergebnisse nahelegen. Die Qualität des Greedy-Algorithmus könnte auch durch ein Postprocessing verbessert werden, indem die Facette bzw. Facetten, in denen das *lokKLV* entsteht, vergrößert werden, indem am *lokKLV* beteiligte Knoten in größere Facetten verschoben werden.

Bereits in der Einleitung wurde vorgeschlagen, das *lokKLV* auf weiteren Graphklassen zu untersuchen oder verschiedene algorithmische Techniken anzuwenden. Besonders divide-and-conquer könnte dabei zu guten Ergebnissen führen, da das *lokKLV* definitionsgemäß lokal betrachtet und optimiert werden kann.

Literaturverzeichnis

- [BF20] Manuel Borrazzo und Fabrizio Frati: On the planar edge-length ratio of planar graphs. *Journal of Computational Geometry*, 11:137–155, 2020, 10.20382/jocg.v11i1a6.
- [BFL20] Václav Blažej, Jiří Fiala und Giuseppe Liotta: On the Edge-Length Ratio of 2-Trees. In: David Auber und Pavel Valtr (Herausgeber): *Graph Drawing and Network Visualization*, Seiten 85–98, Cham, 2020. Springer International Publishing, ISBN 978-3-030-68766-3, 10.1007/978-3-030-68766-3.
- [CDR07] Sergio Cabello, Erik Demaine und Günter Rote: Planar Embeddings of Graphs with Specified Edge Lengths. *Journal of Graph Algorithms and Applications*, 11, Januar 2007, 10.7155/jgaa.00145.
- [EW90] Peter Eades und Nicholas Wormald: Fixed edge-length graph drawing is NP-hard. *Discrete Applied Mathematics*, 28:111–134, August 1990, 10.1016/0166-218X(90)90110-X.
- [LLL19] Sylvain Lazard, William Lenhart und Giuseppe Liotta: On the edge-length ratio of outerplanar graphs. *Theoretical Computer Science*, 770:88–94, 2019, 10.1016/j.tcs.2018.10.002.
- [Nö05] Martin Nöllenburg: Automated Drawing of Metro Maps. Diplomarbeit, Universität Karlsruhe, August 2005.
- [Pat01] Maurizio Patrignani: On the Complexity of Orthogonal Compaction. *Computational Geometry*, 19, 2001, 10.1016/S0925-7721(01)00010-4.

Erklärung

Hiermit versichere ich die vorliegende Abschlussarbeit selbstständig verfasst zu haben, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben, und die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde unter Erlangung eines akademischen Grades vorgelegt zu haben.

Würzburg, den 30. September 2021

.....
Michael Leeming