Bachelorarbeit

Ein Algorithmus zur Optimierung einer Flächenaufteilung

Michael May

Abgabedatum:	28. Februar 2020
Betreuer:	Prof. Dr. Alexander Wolff
	Dr. Jonathan Klawitter

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

Zusammenfassung

Die vorliegende Arbeit befasst sich mit dem Problem der Optimierung einer Flächenaufteilung. Dabei soll eine Gesamtfläche in mehrere verschiedene Teilflächen bestimmter Größe unterteilt werden. Die zu unterteilende Fläche wird als Polygon dargestellt, die gewünschten Teilflächen werden durch einen knotengewichteten Graphen gegeben. In dieser Arbeit wird ein Verfahren entworfen, dass aus dieser Eingabe eine Flächenaufteilung des Polygons berechnet. Dafür wird zuerst der gegebene Graph intern trinaguliert und zu diesem Graphen ein um das Polygon erweiterter Dualgraph mit den Facettenschwerpunkten der Innenfacetten als Knoten bestimmt. Der Dualgraph stellt eine Partitionierung des Polygons dar, die dann mithilfe eines kräftebasierten Verfahrens verbessert werden soll. Bei dem Verfahren werden nur die Knoten des ursprünglichen Graphen verschoben. Dabei errechnen sich die Kräfte aus den Fehlern der einzelnen Teilflächen, dem Abstand zum Polygon und einer Kraft zur Erhaltung der Planarität des Graphen. Eine Implementierung dieses Verfahrens zeigt durch Tests mit einigen Beispielen Probleme des Verfahrens unter Berücksichtigung der Einschränkungen auf. Weiter werden Möglichkeiten vorgestellt, den Algorithmus zu verbessern und einige Einschränkungen aufzulösen.

Als Einstieg der Arbeit wird zunächst in Kapitel 1 die Problemstellung näher erörtert und ein Einblick in den aktuellen Stand der Wissenschaft gewährt. Dann folgt eine genaue Beschreibung der Vorgehensweise in Kapitel 2, wie das Problem gelöst werden soll. Dabei werden auch die Einschränkungen, die in dieser Arbeit für das Verfahren getroffen wurden, begründet. In Kapitel 3 wird eine Laufzeitabschätzung gegeben, in Kapitel 4 werden Ergebnisse präsentiert. Abschließend werden in Kapitel 5 noch Möglichkeiten zur Auflösung dieser Einschränkungen vorgestellt und auf weitere Verbesserungen des Algorithmus hingewiesen.

Inhaltsverzeichnis

1 Einleitung

1.1 Problemstellung

Diese Arbeit beschäftigt sich mit einem Problem des *Geographic Information System* (*GIS*), das sich mit der Erfassung, Verarbeitung und Darstellung von räumlichen und geographischen Daten befasst. Konkret soll die Arbeit einen Weg aufzeigen, eine Fläche in Teilflächen bestimmter Größe zu unterteilen; vorher festgelegte Flächen sollen zudem benachbart sein.

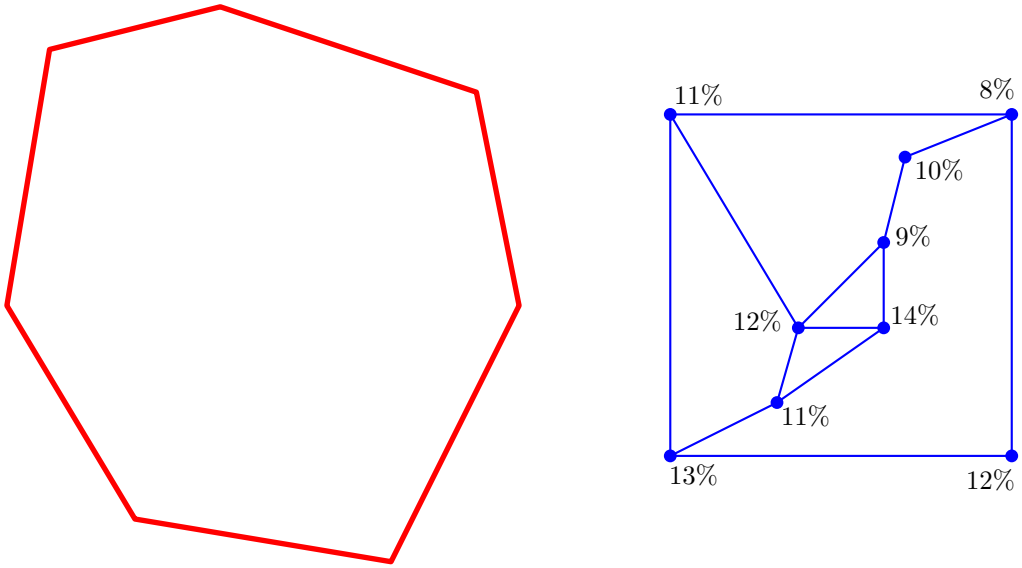

Abb. 1.1: Beispiel für eine Vorgabe zu diesem Algorithmus: Die aufzuteilende Gesamtfläche als konvexes Polygon P (rot) und ein zusammenhängender, knotengewichteter, planarer Graph G (blau), der die gewünschte Aufteilung bestimmt.

Um eine Aufteilung in Teilflächen zu berechnen, wird das Problem zunächst in ein Modell überführt. Dieses enthält die aufzuteilende Fläche als Polygon P im zweidimensionalen Raum. Ein Beispiel für ein solches Polygon zeigt Abbildung 1.1. Es wird davon ausgegangen, dass das Polygon P - also auch die aufzuteilende Fläche - konvex ist, d. h. alle Innenwinkel von P sind kleiner als 180 Grad. Das Ergebnis, die einzelnen Teilflächen, sind am Ende Teilpolygone des Polygons P.

Die Anzahl der Gebiete, deren Anteile an der Gesamtfläche und die Nachbarschaften werden durch einen zusammenhängenden, knotengewichteten Graphen $G = (V, E)$ dargestellt. Jeder Knoten v des gegebenen Graphen steht für eine gesuchte Teilfläche, das

Gewicht des Knotens gibt die anteilige Größe dieser Teilfläche an. Die Gewichte aller Knoten summieren sich folglich zu 100 Prozent auf.

Eine Verbindung e aus der Menge der Kanten E zwischen zwei Knoten gibt an, welche Teilgebiete benachbart sein sollen. Ein Beispiel für einen Graphen, der die gewünschten Teilflächen mit Nachbarschaften darstellt, zeigt Abbildung 1.1. Da der Graph zu einer Aufteilung planar ist, wird im Folgenden davon ausgegangen, dass der gegebene Graph G planar ist.

Im Modell ist nun das Ziel, einen Algorithmus zu entwickeln, der aus einem zusammenhängenden, knotengewichteten, planaren Graphen $G = (V, E)$ und einem Polygon P eine Aufteilung für dieses Polygon berechnet, sodass die Anzahl der Teilflächen der Anzahl der Knoten in G entspricht. Die einzelnen Gebiete sollen dabei die vorgegebene Größe aufweisen. Um die gewünschte Nachbarschaftsrelation zu erfüllen, müssen zudem zwei Teilgebiete a_1 und a_2, die im Graphen G durch v_1 und v_2 dargestellt werden, benachbart sein, wenn eine Kante (v_1, v_2) zwischen diesen Knoten existiert.

1.2 Verwandte Arbeiten

Zu dem Problem der Optimierung einer Flächenaufteilung gibt es in der Wissenschaft verschiedene Ansätze. Viele dieser Ansätze arbeiten mit kombinatorischen Optimierungslösungen, wie beispielsweise Santé-Riveira et al. [SRBMCMMB08], die den *simulierten Glühalgorithmus (SGA)* verwenden. Dieser emuliert das Verhalten eines thermodynamischen Systems. Bei diesem Verfahren wird die Grundfläche in I quadratische Einheiten unterteilt und es gibt N verschiedene Anwendungsbereiche, die auf die I Einheiten verteilt werden sollen. Da Teilstücke für einige Nutzungsmöglichkeiten besser geeignet sind als andere, wird jeder Flächeneinheit ein Eignungswert A_{in} zugewiesen; optional kann zusätzlich noch eine Gewichtung W_{in} festgelegt werden. Das Ziel ist eine Maximierung des Eignungswertes für die Gesamtfläche - unter Berücksichtigung der Gewichtung - und eine Maximierung der Kompaktheit, d. h. dass Flächen mit gleicher Nutzung möglichst zusammenhängend sein sollen. Die Kraft des simulierten thermodynamischen Systems, wird dabei durch eine Zielfunktion ersetzt, die maximiert werden soll. Dabei wird der Wert der Zielfunktion des aktuellen Zustandes E_c mit dem Wert eines durch Zufall berechneten Zustandes E_t verglichen. Ist der Wert E_t besser als E_c wird dieser Zustand als neuer aktueller Zustand für die nächste Iteration gewählt. Ist der Wert schlechter, wird dieser Zustand gemäß der Boltzmann-Wahrscheinlichkeitsverteilung als nächster aktueller Zustand gewählt. Der Algorithmus stoppt entweder durch Erfüllung der Abbruchbedingung oder nach einer vorher festgelegten Anzahl an Iterationen. Wenn man nur die Tauglichkeit mithilfe dieses Verfahrens optimiert, ist dieses Verfahren nach Santé-Riveira et al. [SRBMCMMB08] anderen Verfahren deutlich überlegen, hat aber den Nachteil einer deutlich größeren Fragmentierung, d. h. Flächen, die den gleichen Anwendungsbereich haben, sind nicht unbedingt zusammenhängend. Eine große Fragmentierung führt zwangsweise zu sehr vielen kleinen Bereichen, die unterschiedliche Anwendungsbereiche haben. Insgesamt hat dieses Verfahren eine hohe Laufzeit und hängt stark von der Ein-

gabe guter Landnutzungsgebiete ab.

Chaidee et al. [CPST17] verfolgen einen komplett anderen Ansatz. Sie umgehen das Problem der Fragmentierung, indem sie nur zusammenhängende Teilflächen berechnen. Als Eingabe erwartet dieses Verfahren einen Graphen und eine Grundfläche dargestellt durch ein Polygon. Für den Graphen wird ein kraftgerichteter Algorithmus zum Graphenzeichnen aufgerufen, um die bestmögliche Startposition für die Knoten in der Ebene zu erhalten. Mit Hilfe der generierten Knotenpositionen wird daraufhin für den Graphen das Laguerre-Voronoi-Diagramm berechnet, welches ein gewichtetes Voronoi-Diagramm ist. Mit dem Polygon und dem geradlinigen Voronoi-Diagramm ist dadurch eine Flächenaufteilung des gegebenen Polygons entstanden. Nun werden Fehler und Kräfte für die einzelnen Knoten des Graphen bestimmt und eine neue Iteration mit der Berechnung des Voronoi-Diagramms gestartet.

2 Vorgehensweise

Im Folgenden soll ein weiterer Ansatz zum Lösen des Flächenaufteilungsproblems erarbeitet und vorgestellt werden. Dieser arbeitet im Gegensatz zu den in Abschnitt 1.2 vorgestellten Algorithmen nicht mit kombinatorischen Mitteln, wie Santé-Riveira et al., oder Voronoi-Diagrammen, wie Chaidee et al., sondern versucht über die Bestimmung eines Dualgraphen iterativ eine möglichst geschickte Flächenaufteilung zu bestimmen. Der folgende Algorithmus 1 fasst die hier verwendete Vorgehensweise zusammen.

```
1  OptimizeSurfaceDivision(Graph G, Polygon P)
2  begin
3      G+ = triangulateInternally(G);
4      GD = extendedDualGraph(G+);
       /* centers of inner faces as vertices of GD, corners of P and
          intersections with P as vertices of outer face              */
5      foreach face of GD do
6          faceError = calculateErrorOfFace(face);
7      optimize: while totalError(GD)>toleratedError do
8          foreach vertex of G+ do
9              calculateForces(faceError);
               /* calculates forces for each adjacent vertex          */
10             if P.containsNot(vertex) then
11                 calculateForceToPolygon(vertex);
12         foreach vertex of G+ do
13             move();
14         foreach face of GD do
15             faceError = calculateErrorOfFace(face);
```

Algorithmus 1 : Optimize a Surface Division

Der dargestellte Algorithmus lässt sich in zwei größere Schritte unterteilen: Im ersten Schritt wird eine Partitionierung berechnet, indem der Graph G intern trianguliert wird und zu diesem triangulierten Graphen G^+ ein erweiterter Dualgraph bestimmt wird. Die Facetten des Dualgraphen G^D stellen die Teilgebiete der Flächenaufteilung dar. Im zweiten Schritt - der Optimierung - wird die Partitionierung aus dem ersten Schritt mithilfe eines kräftebasierten Verfahrens iterativ verbessert.

Bevor näher auf diese Schritte eingegangen wird, werden Einschränkungen dieser Implementierung aufgezeigt und erklärt. Das Verfahren ist allerdings auf planare Graphen

im Allgemeinen erweiterbar. Danach ist kurz dokumentiert, in welcher Form ein Graph und das Polygon für diese Implementierung bereitgestellt werden müssen.

2.1 Einschränkungen der vorgestellten Implementierung

Um eine zuverlässige Funktion des implementierten Algorithmus zu gewährleisten, müssen folgende Einschränkungen beachtet werden:

Der Teilgraph des Graphen G, der nur aus den die Außenfacette von G bestimmenden Knoten und Kanten besteht, muss zweifach zusammenhängend sein. Damit soll verhindert werden, dass die Außenfacette zu sich selbst benachbart ist. Ist die Außenfacette nicht zweifach zusammenhängend, d. h. gibt es einen Knoten der Außenfacette, der nur über eine Kante erreichbar ist, so ist eine eindeutige Bestimmung einer orthogonalen Geraden zu dieser Kante nicht möglich. Zudem müssen für so eine Gerade zwei Schnittpunkte bestimmt werden. Die Orthogonale schneidet dabei nicht immer das nur das Polygon P. Es können zwei Schnittpunkte mit anderen orthogonalen Halbgeraden zu Kanten der Außenfacette auftreten. Ein Beispiel für einen Graphen, dessen Außenfacette einen Knoten enthält, der nur über eine Kante erreicht werden kann, wird in Abbildung 2.1 gezeigt. Die Mittelsenkrechte zu der Kante, die durch die Einschränkung verhindert werden soll, schneidet zwei orthogonale Halbgeraden zu anderen Kanten der Außenfacette. Der Dualgraph stellt so keine Überschneidungsfreie Partitionierung des Polygons dar. Im Folgenden wird näher auf dieses Problem eingegangen.

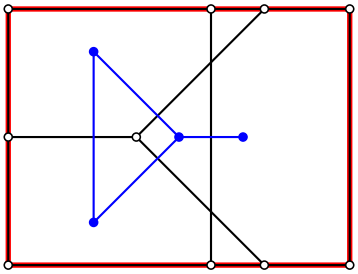

Abb. 2.1: Ein Graph, dessen Außenfacette nur einfach zusammenhängend ist. Die orthogonale Mittelsenkrechte zu dieser Kante schneidet zwei andere Halbgeraden zu Kanten der Außenfacette.

Für die Außenfacette des gegebenen Graphen gilt eine weitere Einschränkung: Die Koordinaten der Knoten dieser Facette müssen ein konvexes Polygon darstellen, da sich sonst die orthogonalen Halbgeraden zu zwei benachbarten Facetten innerhalb des Polygons schneiden können. Abbildung 2.2 skizziert anhand eines Teilgraphen und eines Teils des Polygons beispielhaft diesen Fehler. Die doppelt straffierte Fläche wird dabei beiden Teilpolygonen zugeordnet, obwohl sie zu genau einem Teilpolygon gehören muss. Da die Schnittpunkte mit dem Polygon im Laufe des Algorithmus nicht verändert werden, würde diese Überschneidung bestehen bleiben. Aufgrund dessen, dass die Schnittpunkte nur ein einziges Mal berechnet werden, können auch keine Überschneidungen während eines

Durchlaufs entstehen, obwohl die Außenfacette so verschoben werden kann, dass sie kein konvexes Polygon mehr abbildet.

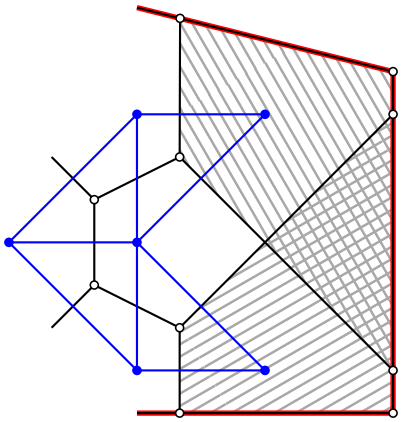

Abb. 2.2: Ein Teilgraph, dessen Knoten der Außenfacette kein konvexes Polygon bilden. Deshalb entsteht ein Schnittpunkt der Halbgeraden innerhalb des Polygons. Die ausgegraute Fläche wird beiden Teilpolygonen zugewiesen.

Bei der hier vorgesteltten Implementierung müssen zudem die Koordinaten der Knoten des Graphen G übergeben werden. Eine Implementierung zur Berechnung der Koordinaten einer planaren Darstellung des Graphen wird in diesem Algorithmus nicht umgesetzt. Dies bringt einen Vorteil mit sich: Für eine Berechnung der Koordinaten müsste auf Verbindungen nach Außen geachtet werden, falls eine Teilfläche im Polygon zu einer Fläche außerhalb des Polygons benachbart sein soll. Durch die Vorgabe der Koordinaten ist eine grobe Platzierung der Teilflächen bereits vorgegeben.

2.2 Bereitstellung des Graphen und des Polygons

Der Implementierung des Algorithmus werden der Graph und das Polygon in Form von zwei CSV-Dateien übergeben. Die Datei, aus der ein Graph eingelesen wird, muss wie folgt aufgebaut sein: Am Anfang der Datei können Kommentare stehen; diese müssen mit '#' beginnen. Danach enthält die Datei alle Knoten des Graphen mit ihrer ID, ihrem Gewicht und ihrer X- und Y-Koordinate in folgender Form:

ID->Gewicht;X;Y

Die ID ist eine Ganzzahl, die größer als Null sein muss. Das Gewicht kann eine Gleitkommazahl sein und wird in Prozent angegeben; die Gewichte aller Knoten addieren sich folglich zu 100 auf. Um die Außenfacette für die Einbettung eindeutig zu bestimmen, muss der erste Knoten in der Datei ein Knoten der Außenfacette sein. Danach folgt eine Leerzeile, die die Übergabe der Knoten beendet.

Darauf folgen die Ajazenzlisten der einzelnen Knoten zeilenweise. Die erste Knoten-ID in einer Zeile legt den Knoten fest, zu welcher diese Adjazenzliste gehört. Alle weiteren

Knoten sind Adjazenzen zu diesem Knoten. Die Adjazenzlisten enthalten zusätzlich die Einbettung des Graphen. Sie müssen die Verbindungen von einer Kante der Außenfacette zur anderen Kante der Außenfacette im Uhrzeigersinn beinhalten. Falls ein Knoten nicht zur Außenfacette gehört, ist es egal, bei welcher Kante die Adjazenzliste beginnt; die Reihenfolge im Uhrzeigersinn muss allerdings eingehalten werden. Eine Adjazenzliste in der CSV-Datei ist eine durch Semikola getrennte Abfolge von Knoten-IDs. Abbildung 2.3 zeigt eine gültige CSV-Datei und den Graphen, der durch diese Datei projiziert werden kann. Die Gewichte und Koordinaten der einzelnen Knoten sind aus dem Beispiel für eine CSV-Datei abzulesen. Der Knoten 1 hat beispielweise ein Gewicht von 18% und liegt an Position $(3, 9)$. Die Einbettung aus der Datei stimmt ebenfalls mit der Einbettung aus der Zeichnung überein.

```
1  # Graph mit 5 Knoten
2  1->18;3;9
3  2->19;9;9
4  3->20;9;5
5  4->21;7;7
6  5->22;3;5
7
8  1;2;5
9  2;3;4;1
10 3;5;4;2
11 4;2;3
12 5;1;3
```

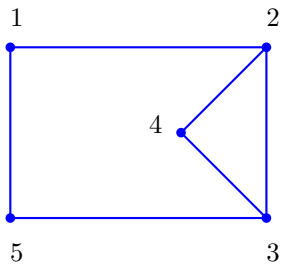

Abb. 2.3: Diese CSV-Datei stellt den daneben abgebildeten Graphen mit 5 Knoten dar. Die Gewichte und Koordinaten sind in der CSV-Datei gegeben.

Das Polygon wird als CSV-Datei mit folgendem Format übergeben: Auch hier können am Anfang der Datei Kommentare stehen, deren Zeilen mit '#' beginnen. Die X- und Y-Koordinaten der Eckpunkte des Polygons folgen zeilenweise durch ein Semikolon getrennt. Diese können als Gleitkommazahlen bereitgestellt werden. Die Eckpunkte müssen gegen den Uhrzeigersinn übergeben werden und es darf in einer Zeile nicht mehr als die X- und Y-Koordinate stehen, für die Eckpunkte sind also keine IDs festzulegen.

2.3 Bestimmung einer Partitionierung

Für das Bestimmen einer Flächenaufteilung werden zwei Teilschritte - die interne Triangulierung und die Berechnung eines erweiterten Dualgraphen - durchgeführt. In Abschnitt 2.3.1 wird erklärt, was mit *interner Triangulierung* gemeint ist und wie diese bestimmt wird. Abschnitt 2.3.2 definiert, was unter einem *erweiterten Dualgraphen* zu verstehen ist und veranschaulicht das Verfahren zum Berechnen des erweiterten Dualgraphen.

2.3.1 Interne Triangulierung des Graphen

Eine interne Triangulierung stellt keine vollständige Triangulierung eines planaren Graphen dar. Bei einer internen Triangulierung werden alle Innenfacetten eines eingebetteten Graphen G trianguliert, die Außenfacette bleibt so bestehen, wie sie der Implementierung übergeben wurde. Allen Innenfacetten werden so lange Kanten hinzugefügt, bis es nur noch Facetten gibt, die durch genau drei Kanten abgeschlossen sind. Im Folgenden ist von den Knoten einer Facette die Rede; damit sind die Knoten der Kanten gemeint, die diese Facette bestimmen. Die Anzahl der Kanten, die eine Facette umschließen, stimmt dabei mit der Anzahl der Knoten zu dieser Facette überein.

Im implementierten Algorithmus wird das Verfahren der internen Triangulierung verwendet, um die unter Umständen bestehenden Verbindungen des Graphen nach außen unangerührt zu lassen. Es wird also sichergestellt, dass die Teilflächen, die einem Knoten der Außenfacette entsprechen, zu Flächen außerhalb des Polygons benachbart sind.

Da bei der Implementierung des Algorithmus keine Koordinaten bestimmt, sondern diese übergeben werden, kann der Graph nicht beliebig trianguliert werden, weil sonst die Planarität von G verletzt werden kann. Deshalb wurde eine geometrische Triangulierung implemetiert. Bei dieser Triangulierungsstrategie wird über alle Facetten des eingebetteten Graphen G iteriert und eine Facette trianguliert, falls diese nicht die Außenfacette ist und sie durch mehr als drei Knoten bestimmt ist. Um eine Facette zu triangulieren, wird per Zufall ein Knoten v_i als Startknoten gewählt. Von diesem Knoten geht man zwei Kanten der Facette entlang, sodass die Facette links der Kante liegt und erhält die zwei Knoten v_{i+1} und v_{i+2}. Eine Kante (v_i, v_{i+2}) wird hinzugefügt, falls der Winkel zwischen den Kanten (v_i, v_{i+1}) und (v_{i+1}, v_{i+2}) im Uhrzeigersinn kleiner als 180 Grad ist und noch keine Kante (v_i, v_{i+2}) in der Kantenmenge von G existiert. Für den Fall, dass keine Kante hinzugefügt wird, dient v_{i+1} als neuer Startknoten. Ist eine Kante hinzugefügt worden, erhält man zwei neue Facetten. Die Facette, die aus den Knoten v_i, v_{i+1} und v_{i+2} besteht, muss nicht weiter betrachtet werden, da sie trianguliert ist. Für die andere Facette ohne den Knoten v_{i+1} beginnt das Hinzufügen einer Kante von Neuem mit der zufälligen Auswahl eines Knotens, wenn diese noch nicht vollständig trianguliert ist. Sobald jede Innenfacette trianguliert wurde, erhält man einen (geometrisch) intern triangulierten Graphen G^+. In Abbildung 2.4 ist ein geometrisch intern triangulierter Graph G^+ zu dem Graphen aus Abschnitt 1.1 dargestellt. Die blauen Kanten sind die Kanten des Graphen G; die grünen Kanten sind im Zuge der Triangulierung hinzugefügt worden.

2.3.2 Bestimmung des erweiterten Dualgraphen zu G^+

Der Dualgraph eines planaren Graphen G hat zu jeder Facette in G einen Knoten. Zwei Knoten des Dualgraphen G^D sind adjazent, wenn die Facetten, die durch die Knoten in G^D repräsentiert werden, benachbart sind, d. h. diese Facetten eine gemeinsame Kante besitzen. Bei einem erweiterten Dualgraphen zu einem Graphen G wird der Knoten, der die Außenfacette des ursprünglichen Graphen G darstellt durch mehrere Knoten ersetzt.

Für den Algorithmus wird der Dualgraph zum intern triangulierten Graphen G^+ aus

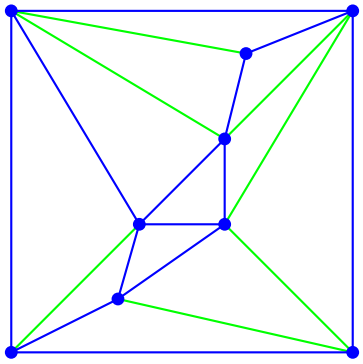

Abb. 2.4: Geometrisch intern triangulierter Graph G^+ (blau und grün) mit 9 Knoten. Die grünen Kanten sind durch die Triangulierung hinzugefügt worden.

Abschnitt 2.3.1 durch das Polygon P erweitert. Die Knoten des Dualgraphen, die den Innenfacetten des Graphen G^+ entsprechen, sind von der Erweiterung nicht betroffen. Für diese Knoten wird festgelegt, dass sie auf dem Schwerpunkt der Facette liegen, zu welcher der Knoten gehört. Der Knoten, der die Außenfacette von G^+ repräsentiert, wird durch die Eckpunkte des Polygons und Schnittpunkte mit dem Polygon ersetzt. Abbildung 2.5 veranschaulicht die folgende Erklärung zur Ermittlung eines Schnittpunktes mit dem Polygon. Einen Schnittpunkt mit dem Polygon erhält man, indem eine zur Außenfacette von G^+ benachbarte Facette f_n und deren Schwerpunkt c_n bestimmt wird. Nun wird die orthogonale Gerade durch den Punkt c_n zu der Kante, die f_n und die Außenfacette von G^+ gemeinsam haben, ermittelt. Aufgrund der Konvexität des Polygons schneidet die Gerade das Polygon in zwei Punkten. Als neuer Knoten v_{ip} wird nur der Schnittpunkt, der ausgehend von v_c in gleicher Richtung liegt, wie der Schnittpunkt mit der Kante der Außenfacette, dem Dualgraphen G^D hinzugefügt.

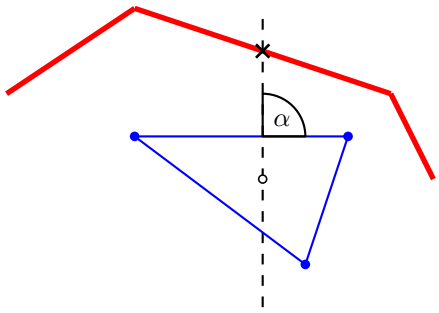

Abb. 2.5: Bestimmung eines Schnittpunkts der orthogonalen Halbgeraden ausgehend von Schwerpunkt einer Facette des Graphen G (blau) mit dem Polygon P (rot). Der Winkel α ist dabei 90 Grad.

Für alle Knoten, die Schwerpunkte einer Facette von G^+ sind, gilt: Eine Kante in G^D wird für zwei Knoten hinzugefügt, falls die jeweiligen Facetten in G^+ benachbart sind. Für die Knoten auf dem Polygon gilt, dass zwischen zwei Knoten eine Kante hinzugefügt

wird, wenn sie beim Ablaufen des Polygons in eine Richtung aufeinander folgen. Einer Facette des Graphen G^+, Alle Knoten, die eine Facette von G^+ repräsentieren, die zu der Außenfacette von G^+ benachbart sind, haben eine weitere Kante. Diese Kante verläuft von dem Schwerpunkt der Facette c_n zu dem Schnittpunkt v_{ip} mit dem Polygon, der zu diesem Schwerpunkt berechnet wurde.

Abbildung 2.6 zeigt für das Beispiel aus Abschnitt 1.1 den intern triangulierten Graphen G^+ (blau und grün) und den um das Polygon P (rot) erweiterten Dualgraphen G^D (schwarz). Hier erkennt man auch, dass der Dualgraph eine erste Flächenaufteilung des Polygons P darstellt. Die Facetten des Dualgraphen stellen dabei die Teilflächen einer Partitionierung dar.

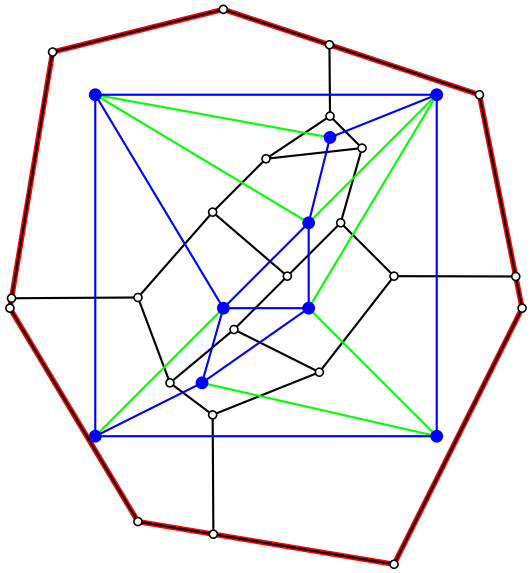

Abb. 2.6: Intern triangulierter Graph G^+ (blau und grün) mit 9 Knoten und der um das Polygon P (rot) erweiterte Dualgraph (schwarz).

Diese Partitionierung erfüllt die Kriterien der Anzahl an Teilflächen und deren Nachbarschaften. Beides wird durch die Dualität der Graphen G^+ und eines kontrahierten Graphen G_k^D zu G^D bewiesen. Eine Kontraktion aller Knoten von G^D, die die Außenfacette von G^+ repräsentieren, also aller Knoten, die auf dem Polygon liegen, liefert den eigentlichen Dualgraphen G_k^D zu G^+. Der Dualgraph zu G_k^D ist wieder G^+, d. h. dass die Anzahl der Facetten in G_k^D der Anzahl der Knoten in G^+ entspricht und die Facetten des Dualgraphen G^D die gewünschten Nachbarschaften aufweisen.

Der erweiterte Dualgraph G^D hat eine Facette mehr als G_k^D, nämlich seine Außenfacette. Dies stellt aber kein Problem dar, weil die Außenfacette keine Teilfläche im Polygon repräsentiert und komplett außerhalb des Polygons liegt.

2.4 Optimierung der Partitionierung

Das Ziel des Optimierungsschritts besteht darin, die in Abschnitt 2.3 bestimmte erste Partitionierung zu verbessern, sodass die Teilflächen unter Beachtung einer Fehlertoleranz die gewünschte Größe aufweisen. Hierfür wird ein kräfte-basiertes Verfahren aus dem Bereich der Visualisierung von Graphen verwendet.

Bei diesem Algorithmus werden nur die Knoten des triangulierten Graphen G^+ iterativ durch das kräftebasierte Verfahren verschoben. Ein Knoten des Dualgraphen, der durch den Schwerpunkt einer Facette von G^+ bestimmt ist, wird indirekt verschoben, weil sich die Koordinaten des Schwerpunkts ändern. Die für das kräftebasierte Verschieben eines Knotens benötigte Kraft setzt sich aus drei Teilkräften zusammen. Die Summe dieser Teilkräfte ist die resultierende Kraft für diesen Knoten in einem Iterationsschritt. Eine Teilkraft errechnet sich aus dem Fehler der Teilflächen, die zweite Teilkraft wird durch die Position des Knotens zum Polygon bestimmt und die letzte Teilkraft erhält die Planarität von G^+. Zunächst wird in Abschnitt 2.4.1 auf die Berechnung des Fehlers einer Teilfläche eingegangen und anschließend die einzelnen Teilkraftberechnungen in Abschnitt 2.4.2, Abschnitt 2.4.3 und Abschnitt 2.4.4 für einen Knoten beschrieben.

2.4.1 Berechnung des Fehlers einer Teilfläche

Der Fehler e_v der Teilfläche zum Knoten v wird im Folgenden als „Fehler eines Knotens“ bezeichnet, weil jede Teilfläche durch genau einen Knoten in G^+ repräsentiert wird. Der Fehler eines Knotens wird in jedem Iterationsschritt durch die Differenz des Soll-Anteils t_v und des momentanen Ist-Anteils a_v berechnet. Es gilt also:

$$e_v = t_v - a_v.$$

Ist die Teilfläche zum Knoten v zu klein, ergibt sich folglich ein positiver Fehler, ist die Teilfläche zu groß entsprechend ein negativer Fehler. Aus dem Fehler jeder einzelnen Teilfläche kann ein Gesamtfehler e für die komplette Flächenaufteilung berechnet werden. Der Gesamtfehler ist die Summe der Beträge der Fehler der einzelnen Teilflächen:

$$e = \sum_{v \in V} |e_v|.$$

Würde man auf die Bildung des Betrages verzichten, wäre die Summe der Fehler gleich null, was suggerieren würde, dass die Flächenaufteilung insgesamt keinen Fehler hat, also richtig ist, obwohl die einzelnen Teilflächen noch eine falsche Größe aufweisen. Deshalb müssen hier die Beträge der Fehler summiert werden.

2.4.2 Kraft aus dem Fehler für einen Knoten

Aus dem Fehler einer Teilfläche zum Knoten w werden Kräfte für alle adjazenten Knoten von w bestimmt. Der Knoten w soll dabei auf den adjazenten Knoten v eine abstoßende

Kraft ausüben, wenn die Teilfläche zum Knoten w zu klein ist. Es wird dabei unterschieden, ob die Kraft einen Knoten anzieht oder abstößt; eine anziehende Kraft wird halbiert. Die Kraft auf einen adjazenten Knoten v kann also folgendermaßen errechnet werden:

$$F_v(w) = \frac{\overrightarrow{wv}}{|\overrightarrow{wv}|} \cdot \frac{e_w}{deg(w)} \cdot \alpha.$$

Hierbei stellt $\overrightarrow{wv}$ den Vektor von v nach w im Zweidimensionalen dar; $deg(w)$ gibt den Grad des Knotens w an. Der Parameter α ist ein globaler Faktor zwischen 0 und 1, damit die Kräfte innerhalb eines Iterationsschrittes nicht zu groß werden. Für einen Knoten v ergibt sich daraus folgende Kraft in Abhängigkeit zu allen adjazenten Knoten:

$$F_v(adj(v)) = \sum_{\substack{w \in V \\ (w,v) \in E}} F_v(w).$$

2.4.3 Kraft für einen Knoten außerhalb des Polygons

Eine weitere Kraft, die auf den Knoten v wirkt, wird durch die Position des Knotens v zum Polygon P bestimmt. Diese Kraft ist nötig, damit die Knoten von G^+ nahe am Polygon gehalten werden, sodass die Schwerpunkte der Facetten, also die Knoten des Dualgraphen, das Polygon P nicht verlassen. Diese Kraft wirkt nur auf den Knoten v, wenn dieser nicht mehr innerhalb des Polygons liegt. Es gibt mehrere Möglichkeiten die Richtung dieser Kraft zu bestimmen. Eine erste Möglichkeit ist eine Kraft, die senkrecht zu einer nahegelegenen Polygonkante in Richtung des Polygons wirkt. Abbildung 2.7 zeigt allerdings einen Fall, bei dem diese Berechnung nicht zum Ziel führt. Egal, welche der beiden Polygonkanten ausgewählt wrd, um eine Kraft orthononal zu dieser Polygonkante zu berechnen, wird v nicht anhand eines ins Polygon gerichteten Vektors verschoben.

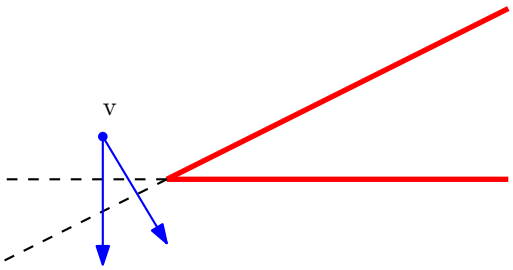

Abb. 2.7: Kräfte (blau) die orthogonal zu den Polygonkanten (rot) wirken; diese sind zum Polygon, aber nicht in das Polygon gerichtet.

Deshalb wurde eine andere Richtungsmöglichkeit gewählt. Bei dieser wird der Knoten v zum Schwerpunkt c des Polygons P verschoben. Die Stärke der Kraft ist dabei abhängig von der euklidischen Distanz des Knotens v zu dem Schnittpunkt s der Strecke (v, c) mit dem Polygon P. Die euklidische Distanz zwischen den Punkten v und s wird mit $d(v, s)$

abgekürzt. Damit errechnet sich die Kraft für einen Knoten, der außerhalb des Polygons liegt, durch folgenden Term:

$$F_v(P) = \frac{\overrightarrow{vc}}{|\overrightarrow{vc}|} \cdot d(v, s) \cdot \alpha.$$

Dabei stellt $\overrightarrow{vc}$ den Vektor von Knoten v zum Schwerpunkt c des Polygons dar. Der Faktor α ist hier derselbe globale Faktor zwischen 0 und 1 wie oben. Weil diese Berechnung zur Folge hat, dass ein Knoten höchstens bis zu einer Polygonkante geschoben wird, multipliziert sich noch ein Faktor auf. In der Implementierung wurde dieser Faktor auf $0,05$ gesetzt Für einen Knoten, der innerhalb des Polygons P liegt, wird die euklidische Distanz auf 0 gesetzt. Damit ist die Kraft für einen Knoten, der im Polygon liegt, gleich null.

2.4.4 Kraft zum Erhalt der Planarität für einen Knoten

Um die Planarität von G^+ zu erhalten, wird eine abstoßende Kraft zwischen einem Knoten v und einer Kante (a, b) berechnet. Die Berechnung dieser Kraft stammt aus [**?**] von F. Bertault., in dem ein Algorithmus entwickelt wird, der Graphen schön zeichnet, d. h. falls der zu zeichnende Graph eine Symmetrie besitzt, wird diese in der Zeichnung dargestellt und die Kantenlängen werden, soweit möglich, angeglichen. Für diesen Algorithmus wird ebenfalls ein kräftebasiertes Verfahren verwendet. Für den Erhalt der Planarität wird in dem hier vorgestellten Algorithmus aus [**?**] nur die abstoßende Kraft zwischen einem Knoten und einer Kante genutzt. Die Kraft ist null, solange der Abstand zwischen dem Knoten und der Kante größer als eine vorher festgelegte Konstante γ ist. Der Abstand des Knotens und der Kante wird bestimmt durch den Abstand der Punkte v und i_v, dem Schnittpunkt der Orthogonalen zu der Kante (a, b) durch den Knoten v mit der Strecke (a, b). Liegt der Schnittpunkt nicht auf der Strecke (a, b), so ist die abstoßende Kraft ebenfalls null. Andernfalls berechnet sich die Kraft durch folgenden Term:

$$F_v((a,b)) = \begin{cases} -\dfrac{(\gamma - d(v, i_v))^2}{d(v, i_v)} \cdot (x(i_v) - x(v)) & i_v \in (a,b) \wedge d(v, i_v) < \gamma \\ 0 & \text{sonst} \end{cases}$$

In der Implementierung wurde γ auf den Wert $0,5$ festgelegt.

2.4.5 Gesamtkraft für einen Knoten

Für einen Knoten v errechnet sich demnach die Gesamtkraft durch die Summe seiner Teilkräfte:

$$F_v = F_v(adj(v)) + F_v(P) + F_v((a,b)).$$

In jedem Iterationschritt werden die Kräfte für die einzelnen Knoten von G^+ neu berechnet und die Knoten anhand ihrer resultierenden Kräfte F_v verschoben. Für den

Faktor α wurde in der Implementierung für die Berechnung der Kräfte aus den Fehlern und der Kräfte für Knoten außerhalb des Polygons der Wert $\alpha = 0,1$ gesetzt.

3 Laufzeitabschätzung

In diesem Kapitel wird auf die Laufzeit des implementierten Algorithmus mithilfe der O-Notation abgeschätzt. Die Laufzeit wird in Abhängigkeit zur Anzahl der Knoten im gegebenen Graphen $G = (V, E)$ angegeben; es gilt $n = |V|$. In der Teilabschätzung zur Berechnung eines Dualgraphen tritt zusätzlich die Variable k auf, die als Anzahl der Eckpunkte des Polygons definiert ist.

In Abschnitt 3.1 wird die Triangulierung und die Berechnung des erweiterten Dualgraphen betrachtet. Dieser Schritt wird für einen vorgegebenen Graphen G und ein Polygon P nur einmal durchgeführt. Danach wird die Laufzeit eines Iterationsschritts der Optimierung in Abschnitt 3.2 untersucht.

3.1 Laufzeit des Partitionierungsschrittes

Das geometrische Triangulieren benutzt eine Methode, um eine Facette des gegebenen Graphen zu triangulieren. Für die Triangulation einer Facette wird bei dieser Implementierung $O(n^2)$ Zeit benötigt, da, um eine Kante zu finden, die hinzugefügt werden darf, im Worst Case alle Knoten der Facette, also höchstens alle n Knoten, besucht werden müssen. Das Triangulieren der verbleibenden Facette dauert dann noch $O(n-1)$ Zeit, was mit $O(n)$ abzuschätzen ist. Damit ergibt sich für die Triangulation einer Facette eine Worst Case Laufzeit von $O(n^2)$. Das geometrische Triangulieren ruft die Methode zum Triangulieren einer Facette für jede Facette des gegebenen Graphen auf. Die Anzahl der Facetten in einem planaren Graphen ist höchstens $2n-4$, weshalb die Aufrufe der Methode zum Triangulieren durch n abgeschätzt werden können. Die Laufzeit der kompletten Triangulation eines gegebenen Graphen ist allerdings kleiner $O(n^3)$, weil mit steigender Anzahl der Facetten die Laufzeit zum Triangulieren einer Facette bei gleichbleibender Anzahl an Knoten sinkt. Die Laufzeit für das Triangulieren wird deshalb mit $O(n^2 \log n)$ abgeschätzt.

Die Laufzeit für die Berechnung des erweiterten Dualgraphen kann mit $O(n^2)$ abgeschätzt werden, weil die Bestimmung der Knoten in $O(n \cdot k)$ Zeit möglich ist und das Hinzufügen der Kanten des Dualgraphen $O(n^2)$ Zeit kostet. Dies ist darauf zurückzuführen, dass zu jeder Facette des ursprünglichen Graphen alle Nachbarfacetten bestimmt werden.

Eine Verlinkung der Facetten des Dualgraphen mit den Knoten des Graphen ist für die Fehlerbestimmung nötig. Die Methode, die diese Verlinkung herstellt, benötigt bei der umgesetzten Implementierung des Algorithmus eine Laufzeit von $O(n^4)$. Eine Aufgabe für die Zukunft wäre eine Verbesserung dieser Methode, sodass diese nicht langsamer als die restlichen Schritte der Partitionierung ist.

Insgesamt folgt eine Laufzeit von $O(n^4)$ für die Berechnung einer Partitionierung; mit einer schnelleren Methode zur Verlinkung der Facetten des Dualgraphen mit den Knoten im Graphen könnte man eine Laufzeit von $O(n^2 \log n)$ erreichen.

3.2 Laufzeit der Optimierung

Bei der Optimierung werden die Schritte der Fehlerberechnung, der Kräfteberechnung und des Verschiebens (i) Iterationsschritte oft ausgeführt. Für das Berechnen des Fehlers einer Teilfläche wird $O(1)$ Zeit abgeschätzt. Da dies für jede Teilfläche durchgeführt wird, ergibt sich eine Laufzeit von $O(n)$ für die Berechnung der einzelnen Fehler. Damit folgt auch, dass der Gesamtfehler in $O(n)$ Zeit berechnet werden kann.

Aus den einzelnen Fehlern wird eine Teilkraft für jeden Knoten berechnet, wofür $O(n)$ Zeit benötigt wird, weil für einen Knoten eine Kraft für alle adjazenten Knoten berechnet wird. Damit wird jede Kante des Graphen zweimal betrachtet, einmal für jede Richtung. Die Anzahl der Kanten in einem maximalen planaren Graphen ist $3n - 6$, weshalb die Laufzeit für die Berechnung dieser Kraft mit $O(n)$ abgeschätzt werden kann.

Für die Berechnung der Kraft, die von der Position des Knotens v zum Polygon abhängt, kann für einen Knoten $O(1)$ Zeit veranschlagt werden. Im Worst Case liegt jeder Knoten außerhalb des Polygons, was zu einer Laufzeit von $O(n)$ führt.

Die Laufzeit für die Berechnung der Kraft zum Erhalt der Planarität kann mit $O(n^2)$ abgeschätzt werden, da für jeden Knoten die abstoßende Kraft zu jeder Kante berechnet werden muss.

Das Verschieben eines Knotens beansprucht $O(n)$ Zeit, da jeder Knoten des Graphen verschoben werden muss. Zusätzlich müssen auch die Knoten im Dualgraphen, die eine Innenfacette von G darstellen, angepasst werden. Die Anzahl der zu verschiebenden Knoten im Dualgraphen liegt bei höchstens $2n - 3$. Damit folgt eine Laufzeit von $O(n)$. Der Optimierungsschritt benötigt demnach insgesamt $O(n^2)$ Zeit pro Iterationsschritt.

4 Resultate

In diesem Abschnitt werden einige Testergebnisse dieser Implementierung gezeigt und diskutiert. Der vorgestellte Algorithmus liefert eine Flächenaufteilung, in der die gewünschten Nachbarschaftsbeziehungen erfüllt sind. Allerdings können aufgrund der Einschränkung, dass die Schnittpunkte mit dem Polygon im Optimierungsschritt nicht verändert werden, die geforderten Größen der Teilflächen nicht immer erreicht werden. Sollte doch eine gewünschte Aufteilung entstehen, ist dies auf die Eingabe guter Koordinaten der einzelnen Knoten des Graphen G zurückzuführen.

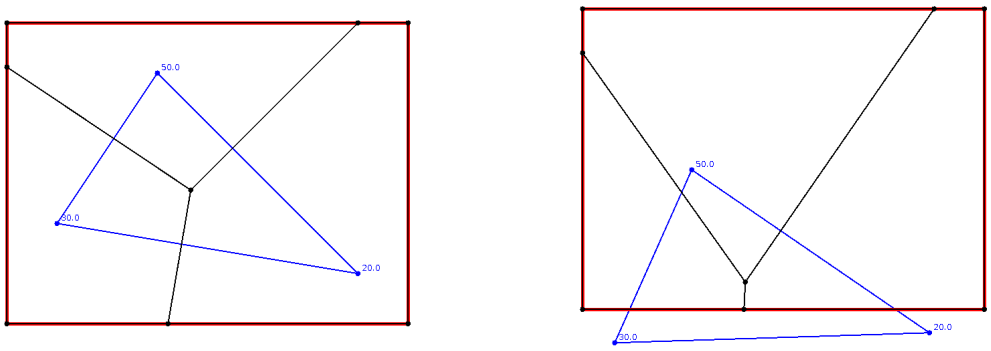

Abb. 4.1: Startzustand und Ergebnis nach 1000 Iterationsschritten mit einem Graphen bestehend aus drei Knoten und einem Rechteck als Eingabe. Die Knoten sind mit ihren jeweiligen Gewichten beschriftet.

Abbildung 4.1 zeigt ein Minimalbeispiel mit drei Knoten und einem Rechteck als Polygon. Die Teilflächen sollen dabei 50%, 30% und 20% der Gesamtfläche einnehmen. Nach dem beschriebenen Verfahren des Algorithmus sind die Schnittpunkte mit dem Polygon nach der Berechnung des Dualgraphen fest. Eine Verschiebung der Knoten ändert bei diesem Minimalbeispiel deshalb nur einen Eckpunkt aller Teilpolygone, den Schwerpunkt der einzigen Innenfacette, und kann deshalb zu keinem Ergebnis kommen, das der gewünschten Aufteilung entspricht. Die Fläche, die am Ende 50% der Gesamtfläche einnehmen soll, kann diese Größe mit dieser Implementierung nicht erreichen, denn die Knoten des Graphen würden mit der umgesetzten Kräfteberechnung so verschoben werden, dass der Schwerpunkt der Innenfacette außerhalb des Polygons liegt, bevor diese Teilfläche den gewünschten Anteil einnimmt.

Da bei diesem Graphen keine Kreuzung der orthogonalen Kanten zu den Kanten der Außenfacette möglich ist, wurde für diesen Graphen getestet, ob eine Neuberechnung des erweiterten Dualgraphen in jedem Iterationsschritt zu einem besseren Ergebnis führt.

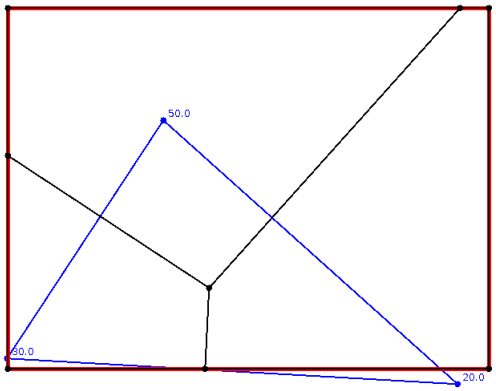

Abb. 4.2: Flächenaufteilung mit iterativer Neuberechnung des Dualgraphen nach 500 Iterationsschritten.

Wie Abbildung 4.2 zeigt, liefert der angepasste Algorithmus schon nach 500 Iterationsschritten ein ähnlich gutes Ergebnis für diesen Graphen. Wechselt der Schnittpunkt der oberen Kante in den weiteren Iterationsschritten auf die rechte Kante des eingegebenen Rechtecks, wird die Flächenaufteilung weiter verbessert. Trotzdem hängt das Ergebnis stark von der Eingabe ab, aber es ist zu vermuten, dass eine Verbesserung auf diese Weise eine Verbesserung der Ergebnisse für alle Graphen darstellen kann. In Kapitel 5 wird eine Möglichkeit beschrieben, die Erstellung eines Dualgraphen zu verbessern, sodass ein Verschieben von Schnittpunkten auf dem Polygon durch iterative Neuberechnung des Dualgraphen für alle Arten von Graphen möglich ist. Nichtsdestotrotz muss nach Einarbeitung der Verbesserung getestet werden, ob die Vermutung bestätigt werden kann.

Eine Ausführung mit dem Beispielgraphen aus der Problemstellung zeigt ein weiteres Problem der Implementierung auf, denn für dieses Beispiel liefert die Implementierung ebenfalls kein Ergebnis, das der gewünschten Flächenaufteilung entspricht. Abbildung 4.3 zeigt die erste Partitionierung des Polygons und die Partitionierung nach 1000 Iterationsschritten. Das Problem wird anhand der Teilfläche unten links beschrieben. Diese Fläche soll nach der Optimierung der Partitionierung 13% der Gesamtfläche einnehmen. In der ersten Partitionierung ist diese Fläche zu groß, weshalb sie auf alle adjazenten Knoten eine anziehende Kraft auswirkt. Im Laufe der Optimierung wird diese Fläche trotzdem größer, weil sich der Knoten zu dieser Fläche kaum verschiebt. Der Grund dafür ist, dass sich die anziehenden Kräfte der Knoten der Außenfacette und die abstoßende Kraft des verbliebenen Knotens fast annullieren. Da die zu diesem Knoten adjazenten Knoten der Außenfacette eine starke Verschiebung nach außen erfahren, verschieben sich die Schwerpunkte der Facetten, deren Teil der Knoten unten links ist, in eine Richtung, sodass die Teilfläche zu diesem Knoten größer statt kleiner wird. Aus diesem Grund wurde eine Unterscheidung von abstoßenden und anziehenden Kräften implementiert. Anziehende Kräfte werden im Gegensatz zu abstoßenden Kräften halbiert. Abbildung 4.3 zeigt allerdings schon das Ergebnis mit dieser Verbesserung, die kaum eine Auswirkung auf die

Verschiebung des Knotens unten links hatte. Tests mit einem Verhältnis von 1 : 4 und noch kleinerem Verhältnis der anziehende Kräfte zu den abstoßenden Kräften bewirkte ebenfalls keine wesentliche Verbesserung. Aufgrund solcher nicht konvergierenden Flächen wurde in der Implementierung auf eine Optimierung bis zur Abbruchbedingung verzichtet. Stattdessen wurde mit einer festen Anzahl an Iterationsschritten gearbeitet. Die anderen Teilflächen dieses Beispiels näherten sich aber jeweils den gewünschten Größen an.

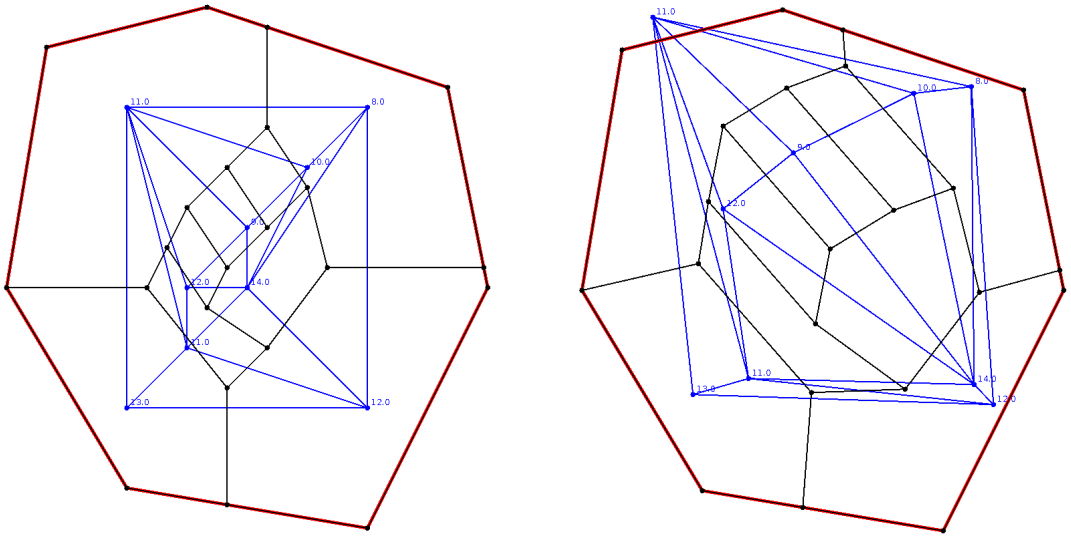

Abb. 4.3: Ausführung des Algorithmus mit dem Beispiel aus der Problemstellung.

Bei einem Durchlauf mit dem Anwendungsbeispiel von Chaidee et al. [CPST17] ist ein weiterer Fehler aufgetreten, der bei dieser Implementierung allgemein auftreten kann. Trotz der Planarität des gegebenen Graphen und der Kraft zur Erhaltung der Planarität ist ein Knoten des erweiterten Dualgraphen über eine Kante des Dualgraphen geschoben worden. So sind sich überschneidende Kanten in Dualgraphen entstanden. Dies kann darauf zurückgeführt werden, dass im Algorithmus die Planaritätsbedingung aus Laufzeitgründen nur für einen Knoten v und „benachbarte“ Kanten berücksichtigt wurde. Benachbarte Kanten meinen Kanten, die zwischen adjazenten Knoten des Knotens v verlaufen. Da die Knoten des Dualgraphen die Eckpunkte der Teilflächen darstellen, würde ohne eine Korrektur dieses Problems, dass man einfach lösen kann, indem man die Planaritätsbedingung für alle Kanten testet, der Eckpunkt einer Teilfäche innerhalb einer anderen Teilfläche liegen.

Vergleich mit Chaidee et al.

Nun wird der Algorithmus dieser Arbeit mit dem Algorithmus von Chaidee et al. [CPST17] verglichen, weil sie ähnlich aufgebaut sind und beide auf Verfahren der Visualisierung von

Graphen basieren. Wegen der oben beschriebenen Ergebnisse ist allerdings abzusehen, dass der Algorithmus für die Flächenaufteilung von Chaidee et al. besser funktioniert als der hier implementierte Algorithmus. Der Gesamtfehler der Flächenaufteilung, der mit der Implementierung dieser Arbeit berechnet wurde, ist deutlich größer als der Restfehler nach der Ausführung mit dem Algorithmus von Chaidee et al. Abbildung 4.4 zeigt in blau den Graphen der Eingabe, der auf die Einschränkungen für die Implementierung dieser Arbeit angepasst wurde und das Polygon der Eingabe in rot. Die grünen Kanten sind die Kanten, die durch die geometrische Triangulierung hinzugefügt wurden. Die einzelnen Knoten sind zudem mit ihren Gewichten beschriftet. Abbildung 4.5 zeigt die Partitionierung vor der Optimierung ohne den Graphen. Betrachtet man auf der linken Seite die unteren beiden Teilflächen und die dazu gehörenden Gewichte im Graphen, stellt man fest, dass die kleine Dreiecksfläche am Ende $11,6\%$ der Gesamtfläche einnehmen soll und die Fläche in der Ecke des Polygons nur $0,7\%$ der Gesamtfläche bedecken soll. Für die kleine Dreiecksfläche gibt es bei dieser Implementierung nicht die Möglichkeit, die gewünschte Größe zu erreichen, weil zwei der drei Eckpunkte des Teilpolygons, das diese Fläche darstellt, fest sind. Eine Abhilfe könnte die Verschiebung der Schnittpunkte sein. Allerdings sind dafür die Aufhebung der Einschränkungen aus Abschnitt 2.1 nötig. Eine Idee hierfür wird in Kapitel 5 beschrieben.

Die Größe der Teilflächen entspricht nicht den gewünschten Anteilen, aber die Implementierung zu dieser Arbeit hat einen Vorteil gegenüber dem Algorithmus von Chaidee et al.:
Die gewünschten Nachbarschaften sind alle vorhanden. In Abschnitt 2.3 wird der Grund dafür schon einmal genannt. Die Dualität von einem Graphen und dessen Dualgraphen beweißt, dass die Nachbarschaften alle vorhanden sind. Bei Chaidee et al. sind nicht alle gewünschten Nachbarschaften vorhanden, sie berechnen hierfür einen Kompatibilitätswert, der maximiert werden soll. Da die Nachbarschaften bei der Vorgehensweise dieser Thesis nur einmal durch die Bestimmung des erweiterten Dualgraphen festgelegt werden, würde sich ein solcher Wert für die Berechnung der Flächenaufteilung bei diesem Verfahren nicht ändern.

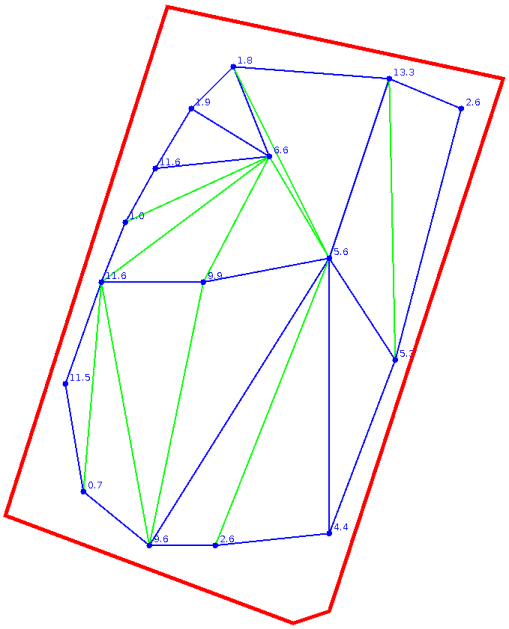

Abb. 4.4: Eingabe des Graphen G (blau) und des Polygons P rot des Anwendungsbeispiels aus Chaidee et al. mit der geometrischen Triangulierung des Graphen (grün). Die Knoten des Graphen sind mit den jeweiligen Gewichten beschriftet.

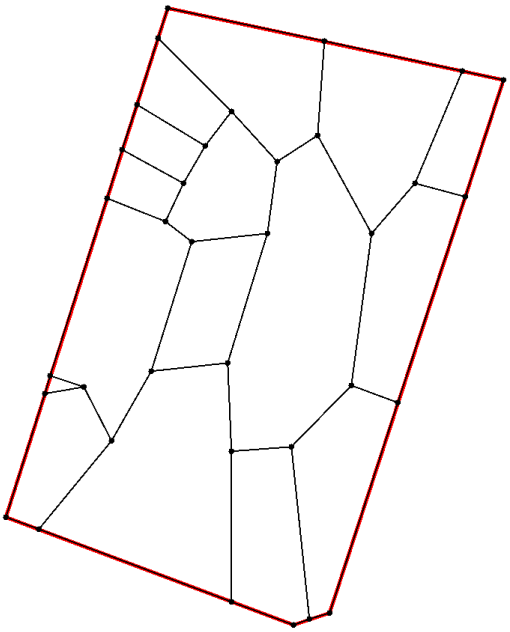

Abb. 4.5: Berechneter Dualgraph zu der oben gezeigten Eingabe des Anwendungsbeispiels von Chaidee et al. ohne Darstellung des ursprünglichen Graphen. Der Dualgraph stellt eine erste Partitionierung des Polygons P (rot) dar.

5 Ausblick

Ein Verbesserungspunkt an diesem Verfahren sollte die Laufzeitverbesserung darstellen, vor allem des Schrittes, der eine Beziehung zwischen der Facette des erweiterten Dualgraphen und dem entsprechenden Knoten im Graphen herstellt.

Außerdem sollten die Einschränkungen für diese Implementierung aufgelöst werden, damit das Verfahren auf alle planaren Graphen angewandt werden kann.

Dabei muss insbesondere darauf eingegangen werden, die Schnittpunkte der Orthogonalen mit dem Polygon anzupassen, sodass keine Kreuzungen von Orthogonalen wie in Abschnitt 2.1 gezeigt entstehen. Nutzt man dieses Anpassen der Schnittpunkte auch im Optimierungsschritt, können bessere Ergebnisse erzielt werden. Eine Idee zur Aufhebung der Einschränkung, dass die Außenfacette ein konvexes Polygon abbilden muss, wäre die Schnittpunkte von orthogonalen Geraden zu Kanten der Außenfacette innerhalb des Polygons als neuen Knoten des Dualgraphen zu definieren. Von diesem Knoten aus könnte man einen neuen Schnittpunkt mit dem Polygon bestimmen, indem man die Winkelhalbierende zu den orthogonalen Halbgeraden durch diesen Knoten bestimmt. Den Schnittpunkt der Winkelhalbierenden mit dem Polygon fügt man als neuen Knoten zum Dualgraphen G^D hinzu. Dann wird eine Kante von den Schnittpunkten der Orthogonalen und zu dem Schnittpunkt der Winkelhalbierenden mit dem Polygon G^D hinzugefügt. Abbildung 5.1 zeigt das Beispiel aus Abbildung 2.2 in dieser verbesserten Variante, die auch eine nicht-konvexe Außenfacette des Graphen zulässt.

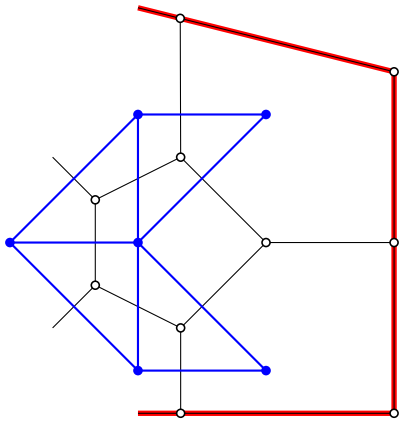

Abb. 5.1: Lösungsansatz für die Aufhebung der Einschränkung der Konvexität der Außenfacette.

Dadurch könnte auch die Bedingung, dass die Außenfacette zweifach zusammenhängend sein muss, fallen gelassen werden. Für einen Knoten, der nur über eine Kante erreich-

bar ist, bestimmt man die Mittelsenkrechte dieser Kante und berechnet zwei Schnittpunkte. Falls die Mittelsenkrechte eine Orthogonale zu einer Außenfacettenkante schneidet, wendet man den oben beschriebenen Lösungsansatz an, um eine Flächenaufteilung zu erhalten, die sich nicht überschneidet. Schneidet die Mittelsenkrechte in einer Richtung nur das Polygon kann dieser Schnittpunkt als Knoten übernommen werden.

Wie man in Kapitel 4 gesehen hat, ist durch die einmalige Bestimmung von Schnittpunkten mit dem Polygon nicht immer eine befriedigende Flächenaufteilung möglich. Der Algorithmus sollte also auch dahingegen verbessert werden, dass z. B. durch eine zusätzliche Verschiebung der Schnittpunkte mit dem Polygon eine Konvergenz zu der gewünschten Flächenaufteilung erreicht wird.

Literaturverzeichnis

[CPST17] S. Chaidee, P. Pakawanwong, V. Suppakitpaisarn und P. Teerasawat: Interactive land-use optimization using Laguerre Voronoi diagram with dynamic generating point allocation. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W7:1091–1098, 2017. https://doi.org/10.5194/isprs-archives-XLII-2-W7-1091-2017.

[SRBMCMMB08] I. Santé-Riveira, M. Boullón-Magán, R. Crecente-Maseda und D. Miranda-Barrós: Algorithm based on simulated annealing for land-use allocation. *Computers & Geoscience*, 34(3):259–268, 2008. https://doi.org/10.1016/j.cageo.2007.03.014.

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 28. Februar 2020

.....
Michael May