

# Fehlerrate von lokalen Routing- verfahren in teilplanarisierten Graphen unter Verwendung des Log-Normal-Shadowing-Modells

Diplomarbeit

zur Erlangung des Grades einer Diplom-Informatikerin

vorgelegt von

Nicole Bremer

Erstgutachter: Prof. Dr. Hannes Frey,  
Institut für Informatik, Arbeitsgruppe Rechnernetze

Zweitgutachter: Dipl.-Inf. Kevin Reuß,  
Institut für Informatik, Arbeitsgruppe Rechnernetze

Koblenz, im September 2017

# Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja    Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.       

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.       

.....  
(Ort, Datum)

.....  
(Unterschrift)

# Zusammenfassung

Skalierbarkeit und garantierte Auslieferung sind essentielle Eigenschaften eines jeden Routingalgorithmus. Beides bietet bei drahtlosen Ad-hoc Netzwerken die Kombination aus Greedy- und Face-Routing, sofern ein planarer Graph zur Verfügung steht. Doch gerade die fehlerfreie Planarisierung bereitet bei realistischen Netzwerken Schwierigkeiten. Daher soll mit dieser Arbeit die Frage beantwortet werden, zu welcher Fehlerrate es führt, wenn der Graph lediglich mit lokalen Methoden teilplanarisiert wird. Dazu wurde eine Simulationsumgebung geschaffen, um unter Anwendung des Log-Normal-Shadowing-Modells zufällige Konnektivitätsgraphen zu generieren. Diese wurden anschließend durch zwei unterschiedliche, lokale Strategien teilplanarisiert. Es wurden neun verschiedene Settings definiert, die sich aus drei unterschiedlichen Graphendichten und drei unterschiedlichen Werten für den  $\sigma$ -Wert des Log-Normal-Shadowing-Modells ergeben. Für jedes Setting wurde in 2000 Simulationen das Verhalten von Greedy-, Face- und kombiniertem Greedy-Face-Routing untersucht und ausgewertet. Zum Abschluss wurden die Ergebnisse dieser Simulation bewertet und diskutiert.

## Abstract

Scalability and guaranteed delivery are essential properties of each routing algorithm. In the field of wireless ad hoc networks both is offered by a combination of Greedy and Face Routing in case of the availability of a planar graph. However it is especially the faultless planarization which fails in realistic networks. Hence this thesis should find an answer to the question, which failure ratio could be observed, if the underlying graph is just partially planarized. For that purpose there was created a simulation environment to generate random connectivity graphs based on the Log-normal shadowing model. This connectivity graphs are getting partially planarized based on two different local strategies. Nine divergent settings were defined that arised from three differing graph densities and three differing values of the Log-normal shadowing models  $\sigma$ -parameter. There were 2000 simulation cycles for each setting, in which the behavior of Greedy, Face and combined Greedy Face Routing was examined and evaluated. Finally the simulation results were rated and discussed.

# Inhaltsverzeichnis

<b>1</b>	<b>Geographische Routingverfahren</b>	<b>1</b>
1.1	Herausforderungen beim Routen in drahtlosen Ad-hoc Netzwerken . . .	1
1.2	Lokales, positionsbasiertes Routing . . . . .	2
1.2.1	Greedy-Routing . . . . .	3
1.2.2	Face-Routing . . . . .	5
1.2.3	Kombiniertes Verfahren . . . . .	9
1.3	Fehlende Übertragbarkeit der UDG-Annahme . . . . .	11
1.4	Ziele der vorliegenden Arbeit . . . . .	12
<b>2</b>	<b>Routing in realistischeren Settings</b>	<b>14</b>
2.1	Quasi Unit Disk Graphen . . . . .	14
2.2	Log-Normal-Shadowing-Modell . . . . .	15
2.3	Ansätze mit neuen Planarisierungsvarianten . . . . .	16
2.4	Ansätze ohne Planarisierung . . . . .	19
<b>3</b>	<b>Simulationsaufbau</b>	<b>22</b>
3.1	Basiskomponenten . . . . .	22
3.1.1	Log-Normal-Shadowing-Modell . . . . .	22
3.1.2	Planarisierungsvarianten . . . . .	24
3.1.3	Routingvarianten . . . . .	26
3.2	Umsetzung . . . . .	26
3.2.1	Settings . . . . .	27
3.2.2	Simulationsablauf . . . . .	28
3.2.3	Ausgaben der Simulationsdurchläufe . . . . .	32
<b>4</b>	<b>Auswertung der Simulationsergebnisse</b>	<b>35</b>
4.1	Planarisierungsverfahren und die entstandenen Subgraphen . . . . .	35
4.1.1	Kantenanzahl der verschiedenen Graphen . . . . .	35
4.1.2	Anzahl der Schnitte in den verschiedenen Graphen . . . . .	36
4.1.3	Knotengrad in den verschiedenen Graphen . . . . .	38
4.1.4	Bewertung der Planarisierungsverfahren . . . . .	38
4.2	Performance der verschiedenen Routingverfahren . . . . .	39
4.2.1	Erfolgsrate der verschiedenen Routingverfahren . . . . .	39
4.2.2	Stretchfaktor der verschiedenen Routingverfahren . . . . .	41
4.2.3	Bewertung der Routingperformance . . . . .	44

4.3	Face-Routing . . . . .	44
4.3.1	Facewechsel . . . . .	44
4.3.2	Hops und euklidische Distanz pro Face . . . . .	45
4.3.3	Bewertung des Face-Wechselverhaltens . . . . .	46
4.4	Kombiniertes Greedy-Face-Routing . . . . .	46
4.4.1	Umschaltungen . . . . .	46
4.4.2	Hops und euklidische Distanz pro Abschnitt . . . . .	47
4.4.3	Bewertung des Wechselverhaltens . . . . .	49
<b>5</b>	<b>Fazit</b>	<b>50</b>

# Abbildungsverzeichnis

1.1	Prinzip des Greedy-Routings . . . . .	3
1.2	Greedy-Routing $\leftrightarrow$ Shortest-Path-Routing . . . . .	4
1.3	Lokales Minimum . . . . .	4
1.4	Rechte-Hand-Regel . . . . .	6
1.5	Face-Routing . . . . .	6
1.6	Face-Routingfehler in nicht-planaren Graphen . . . . .	7
1.7	Gabriel Graph und Relative Neighbourhood Graph . . . . .	8
1.8	Originalgraph und planarisierte Varianten . . . . .	8
1.9	GFG Beispiel . . . . .	10
1.10	Idealisierte und realistische Übertragungreichweiten . . . . .	12
2.1	Virtuelle Knoten/Kanten . . . . .	17
2.2	Nicht detektierbarer Schnitt bei CLDP . . . . .	18
2.3	Reactive Deflection . . . . .	20
3.1	Log-Normal-Shadowing-Modell . . . . .	23
3.2	Varianten der Kantenentfernung . . . . .	25
3.3	Vergleich der unterschiedlichen teilplanarisierten Subgraphen . . . . .	26
3.4	Beispielgraphen der verschiedenen Settings . . . . .	29
3.5	Teilplanarisierte Subgraphen der verschiedenen Settings . . . . .	30
3.6	Visualisierung Greedy-Routing . . . . .	31
3.7	Visualisierung Face- und GFG-Routing . . . . .	32
4.1	Diagramm: Kantenanzahl . . . . .	36
4.2	Diagramm: Schnitte im Graphen . . . . .	37
4.3	Diagramm: Schnitte pro Kante . . . . .	37
4.4	Diagramm: Knotengrad . . . . .	38
4.5	Diagramm: Erfolgsrate der verschiedenen Routingverfahren . . . . .	39
4.6	Abbruchszenario der implementierten Face-Routingvariante . . . . .	41
4.7	Diagramm: Stretchfaktor (Hops) . . . . .	42
4.8	Diagramm: Stretchfaktor (euklidische Länge) . . . . .	43
4.9	Diagramm: Facewechsel . . . . .	45
4.10	Diagramm: Hopanzahl pro Face . . . . .	45
4.11	Diagramm: Euklidische Distanz pro Face . . . . .	46
4.12	Diagramm: Umschaltungen zwischen Greedy- und Face-Routing . . . . .	47

4.13 Diagramm: Durchschnittliche Hopanzahl pro Abschnitt . . . . .	48
4.14 Diagramm: Durchschnittliche euklidische Länge pro Abschnitt . . . .	48

# Tabellenverzeichnis

3.1	Standardwerte für $\gamma$ und $\sigma$ . . . . .	24
3.2	Feldgrößen der verschiedenen Settings . . . . .	28
4.1	Settingbezeichnungen und Parameter . . . . .	35

# Kapitel 1

## Grundlagen geographischer Routingverfahren

### 1.1 Herausforderungen beim Routen in drahtlosen Ad-hoc Netzwerken

Unter Routing versteht man den Wegwahlprozess, um in einem Netzwerk eine Nachricht von einem Startknoten zu einem Zielknoten zu übermitteln. Dabei stellen drahtlose Ad-hoc Netzwerke eine besondere Herausforderung dar. Diese Netzwerkart besteht typischerweise aus batteriebetriebenen einzelnen Knoten, die in der Lage sind, mit den Knoten innerhalb ihres Sendebereichs via Funkübertragung zu kommunizieren, ohne dass es eine feste Infrastruktur gibt. Aufgrund der beschränkten Übertragungreichweiten wird die Kommunikation zwischen zwei Knoten im Netzwerk im Normalfall durch Multihop-Routing realisiert, bei dem dazwischenliegende Knoten die Nachricht weiterleiten. Gleichzeitig können die Knoten mobil sein, was dazu führt, dass es deutlich häufiger zu Topologieänderungen kommt als bei verdrahteten Netzwerken. Es stehen zusätzlich, vor allem im Anwendungsgebiet der Sensornetzwerke, nur beschränkte Ressourcen zur Verfügung, z.B. im Hinblick auf Batterieleistung, Speichergröße und verfügbare Bandbreite. Diese Eigenschaften führen dazu, dass die in der verdrahteten Netzwerkvariante etablierten Strategien, wie beispielsweise Distanzvektor- oder Link-State-Routing, bei drahtlosen Ad-hoc Netzwerken nicht gut skalieren. Es ist aber eine grundlegende Eigenschaft eines Routingalgorithmus, dass er für eine beliebige Anzahl an Netzwerkknoten funktioniert. Die bei drahtgebundenen Netzwerken typischen Verfahren basieren oftmals - wenn auch auf verschiedenen Hierarchieebenen - auf einer globalen Netzwerksicht. Routingentscheidungen werden anhand von Routingtabellen getroffen, die jederzeit eine konsistente Abbildung des Netzwerks enthalten müssen. Dafür müssen kontinuierlich, immer wenn es zu einer Topologieänderung kommt, Update-Nachrichten ausgetauscht werden. Bei drahtlosen Ad-hoc Netzwerken kommt es aufgrund der Knotenmobilität häufig dazu, dass eine Verbindung zwischen zwei Knoten nicht mehr besteht, weil sich ein Knoten aus dem Übertragungsradius hinausbewegt hat,

oder es ergibt sich eine neue Verbindung, weil ein Knoten den Sendebereich eines anderen betreten hat. Entsprechend müsste man einen großen Aufwand betreiben, um die Routing-Tabellen bei mobilen Ad-hoc Netzwerken zu pflegen und aktuell zu halten. Der ständige Austausch von Kontrollnachrichten würde einen enormen Kommunikationsoverhead verursachen, was in Kombination mit den limitierten Ressourcen solcher Netzwerke nicht praktikabel umsetzbar ist.

## 1.2 Lokales, positionsbasiertes Routing

Um also Skalierbarkeit - und damit eine essentielle Basiseigenschaft eines anwendbaren Routingalgorithmus - zu gewährleisten, nutzt man geographisches Routing, was auch als geometrisches Routing bezeichnet wird. Geographisches Routing ist eine lokale und effiziente Routingstrategie, die darauf basiert, dass man Knoten anhand ihrer Position identifiziert. Die Position erhält man über Lokalisationssysteme, von denen in der Literatur verschiedene Varianten vorgestellt wurden. Am bekanntesten ist vermutlich die Positionsbestimmung über das Global Positioning System (GPS), die mittlerweile recht genaue Werte liefert, aber nur im Außenbereich funktioniert [34]. Darüber hinaus gibt es eine Vielzahl von Ansätzen wie zum Beispiel [35, 39, 5, 4], die entweder infrastrukturbasiert sind und die Positionsbestimmung anhand von Referenzpunkten mit bekannter Position leisten, oder es handelt sich um Ad-hoc Lokalisationssysteme ohne vorgegebene Infrastruktur. Einen Überblick zu verschiedenen Lokalisationssystemen findet man bei [19].

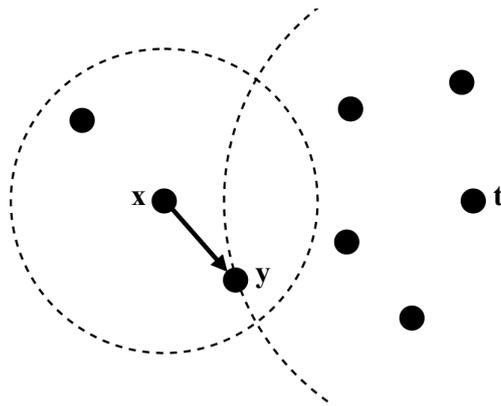
Beim geographischen Routen muss jeder Knoten seine eigene Position kennen. Darüber hinaus kennt er die Positionen seiner direkten Nachbarn, also von allen Nachbar-knoten, mit denen er direkt kommunizieren kann. Der Quellknoten einer Nachricht weiß, welche Position das Ziel der Nachricht hat und vermerkt diese in der Nachricht. Er kennt aber nicht die komplette Route dorthin. Die Quelle wählt einen ihrer Nachbarn aus und sendet die Nachricht dorthin. Dieser Knoten leitet die Nachricht dann wiederum an einen seiner Nachbarn weiter, und dieses Verfahren geht dann solange weiter, bis die Nachricht am Ziel ist oder verworfen wird, falls der Zielknoten nicht erreichbar ist.

Es gibt auch verschiedene nicht-geographischen Ansätzen für Ad-hoc Netzwerke. Diese Routingprotokolle sind proaktiv, reaktiv oder eine Hybridform. Proaktive Verfahren pflegen kontinuierlich die Routen zwischen allen Knoten. Reaktive Strategien bestimmen im Gegensatz dazu den Weg zwischen zwei Knoten erst dann, wenn er zur Datenübertragung benötigt wird. Eine Übersicht über solche nicht-geographischen Strategien bietet [37]. Doch alle diese Verfahren haben das bereits erwähnte Problem mit dem enormen Kommunikationsoverhead. Lokale, positionsbasierte Routingverfahren vermeiden diesen Overhead, da sie lediglich auf präzisen Nachbarschaftsinformationen und der Zielposition beruhen. Die einzige Mindestanforderung ist, dass die Topologieinformationen im 1-Hop-Bereich aktuell gehalten werden müssen. Somit beeinflussen Knoten- und Kantenänderungen immer nur einen kleinen Teil des Netzwerks. Die Menge des benötigten Speichers ist minimal und hängt lediglich von

der Dichte des Netzwerks ab, nicht aber von seiner Gesamtgröße. Daher bieten lokale, geographische Verfahren skalierbare Lösungen für Netzwerke mit eingeschränkten Ressourcen.

### 1.2.1 Greedy-Routing

Die frühen Arbeiten, die sich mit Routing anhand von Positionsinformationen beschäftigen, nutzen eine Weiterleitungsstrategie, die auf dem Routingfortschritt beruht [12, 20, 44]. Unter den lokalen Verfahren ist Greedy-Routing wohl das einfachste. Hierbei analysiert der Sender einer Nachricht seine direkt verbundenen Nachbarn und leitet die Nachricht an denjenigen weiter, der den geringsten euklidischen Abstand zum Ziel hat. Dieser Knoten wählt unter seinen Nachbarn wieder denjenigen mit dem kleinsten Abstand als nächsten Hop aus usw. Auf diese Art wandert die Nachricht immer näher in Richtung Ziel, bis sie schließlich ankommt. Abbildung 1.1 verdeutlicht dieses Prinzip. Der Knoten  $y$  ist der Nachbar von Knoten  $x$ , der den geringsten euklidischen Abstand zum Ziel  $t$  hat.

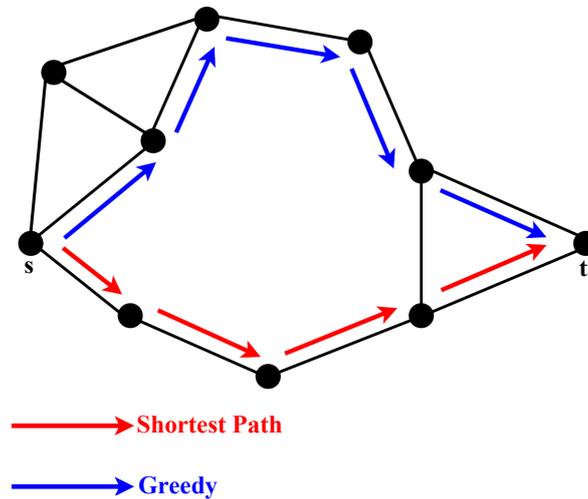


**Abb. 1.1:** Prinzip des Greedy-Routings:  $y$  ist der Nachbarknoten von  $x$ , der am nächsten zum Ziel  $t$  liegt.

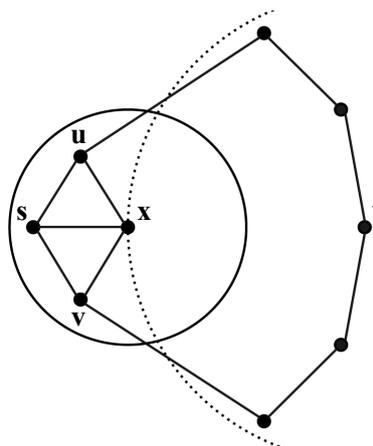
Beim Greedy-Routing geht es ausschließlich darum, pro Weiterleitungsschritt einen Routingfortschritt zu erzielen. Das Ziel ist es nicht, wie beim Shortest-Path-Routing mittels des Dijkstra-Algorithmus, den kürzesten Weg zum Ziel zu finden, denn hierfür würden wieder globale Informationen benötigt. Abbildung 1.2 (nach [42]) zeigt die unterschiedliche Wegwahl von Greedy- und Shortest-Path-Routing im Vergleich.

Neben der Skalierbarkeit ist die garantierte Auslieferung eine weitere wichtige Eigenschaft von Routingalgorithmen. Mit Greedy-Routing kann allerdings keine garantierte Auslieferung zugesichert werden. Man findet leicht Beispieltopologien, bei denen ein Knoten näher am Ziel liegt als alle seine Nachbarn, aber selbst keine Verbindung zum Ziel hat, z.B. aufgrund eines Hindernisses. Abbildung 1.3 zeigt eine solche Situation. Unter all seinen Nachbarn hat Knoten  $x$  den kürzesten Abstand zum Ziel  $t$ . Entsprechend findet Greedy-Routing keinen geeigneten Next-Hop, ob-

wohl sowohl über  $u$  als auch über  $v$  ein Weg zum Ziel existiert. Diese Situation wird als lokales Minimum oder Dead-End bezeichnet. Lokale Minima treten auf, wenn sich zwischen einem Knoten und dem Ziel "Löcher", sogenannte Voids, in der Netzwerktopologie befinden, in denen keine Knoten in Reichweite angesiedelt sind.



**Abb. 1.2:** Gegenüberstellung der unterschiedlichen Wegwahl von Greedy-Routing und Shortest-Path-Routing (nach [42])



**Abb. 1.3:** Greedy-Routing schlägt fehl, da Knoten  $x$  keinen Nachbarn findet, der eine kürzere Distanz zum Zielknoten  $t$  hat als er selbst.

Im Falle eines solchen lokalen Minimums existiert zwar ein Weg zum Ziel, aber dieser würde es notwendig machen, sich zwischenzeitlich vom Ziel zu entfernen, was der

Grundstrategie von Greedy-Routing widerspricht.

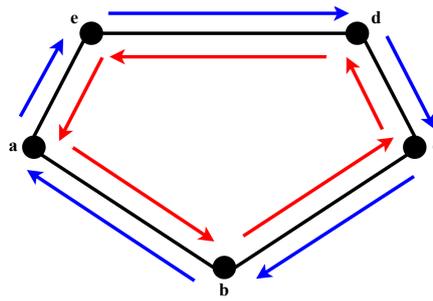
Eine Variante, dieses Problem zu lösen, wäre an dieser Stelle mit globalem Netzwerkwissen zu arbeiten. Damit würde man jedoch die Vorteile des lokalen Ansatzes egalisieren. Bereits Finn [12] erkannte das Problem des lokalen Minimums. Er schlug in seiner Arbeit einen Lösungsansatz vor, der ein limitiertes Fluten über eine eingeschränkte Anzahl an Hops beinhaltet, um auf diesem Weg einen Knoten zu finden, der näher an der Zielposition liegt. Nichtsdestotrotz erzeugt auch eingeschränktes Fluten ein gewisses Maß an Nachrichtenoverhead.

## 1.2.2 Face-Routing

Kranakis et al. [27] präsentierten einen lokalen Algorithmus, den sie Compass Routing nannten. Dieser Algorithmus basiert auf der Annahme, dass man einen Startknoten  $s$  und einen Zielknoten  $t$  hat. Alle weiteren Informationen, die zur Verfügung stehen, sind die aktuelle Position, die Zielposition sowie die Richtung aller vom aktuell betrachteten Knoten ausgehenden Kanten. Nun bewegt man sich auf das Ziel zu, indem man immer die Kante auswählt, die den kleinsten Winkel zu einer virtuellen Verbindungslinie zwischen Start und Ziel, der  $st$ -Linie, aufweist. Allerdings stellten Kranakis et al. fest, dass dieser Algorithmus einen existierenden Weg zum Ziel nicht immer findet. Daher stellten sie als Lösung Compass Routing II vor. Diese Variante sollte die garantierte Auslieferung bereitstellen und stellte einen vielversprechenden Ansatz für den Gebrauch in drahtlosen Ad-hoc Netzwerken dar. Später wurde sie unter der Bezeichnung Face-Routing oder Perimeter-Routing bekannt. Hierbei wird der Graph in Faces unterteilt, also in die polygonalen Flächen, die durch die Kanten zwischen Netzwerkknoten begrenzt sind. Nun werden nacheinander die Faces traversiert, die von der virtuellen  $st$ -Linie geschnitten werden. Dazu geht man am Rand der Faces entlang, bis man auf eine Kante trifft, die von der  $st$ -Linie geschnitten wird. Dort wechselt man in die nächste Face usw. bis man das Ziel erreicht hat. Das Abschreiten einer Face kann anhand der Rechte-Hand-Regel oder anhand der Linke-Hand-Regel erfolgen. Kommt man von einem Knoten  $y$  zu einem Knoten  $x$  wird bei der Rechte-Hand-Regel als nächstes die Kante durchlaufen, die von  $x$  aus gesehen im Uhrzeigersinn, also rechtsherum, von der Kante  $(y, x)$  liegt. Man durchläuft also das Innere einer geschlossenen Face, indem die Kanten entgegen des Uhrzeigersinns durchlaufen werden. Bildlich gesprochen läuft die Rechte-Hand-Regel so ab, dass man durch das Face-Innere geht und dabei immer die rechte Hand an der Kante hat. Die Linke-Hand-Regel funktioniert analog dazu in umgekehrter Richtung.

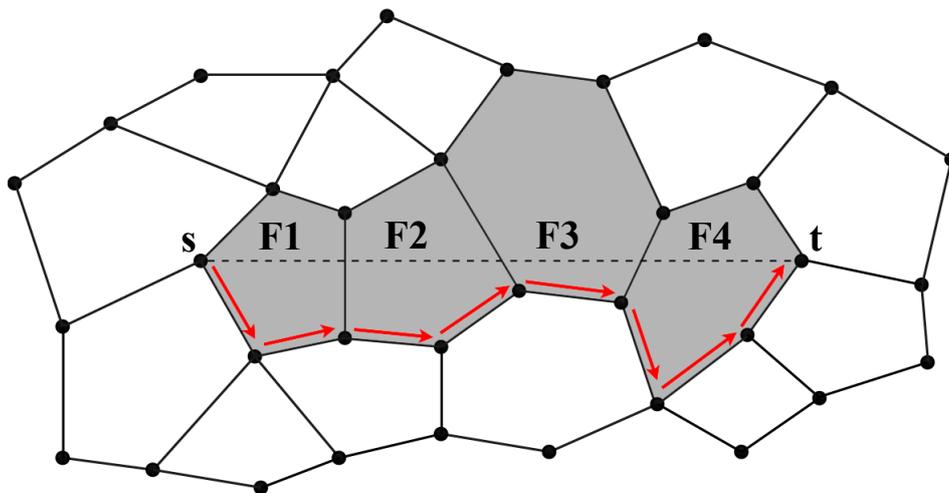
Abbildung 1.4 illustriert die Rechte-Hand-Regel. Beginnt man beispielsweise bei Knoten  $a$  würde das Innere der Face durchlaufen, indem man als erstes die Kante zu Knoten  $b$  entlang geht, dann zu Knoten  $c$  usw. Diesen Fall repräsentieren die roten Pfeile. Es kann aber auch notwendig sein, die äußere, nicht durch Kanten begrenzte, Face entlangzulaufen. Diese Variante wird durch die blauen Pfeile dargestellt und führt zur Knotenabfolge  $a - e - d - c - b$ .

Auch der zweite Bestandteil des Face-Routings, der Wechsel in eine neue Face, geschieht anhand von Regeln. Eine Vielzahl von Varianten wurden untersucht und



**Abb. 1.4:** Ablauf der Face-Traversierung unter Verwendung der Rechten-Hand-Regel

validiert. Karp und Kung wechseln bei ihrem GPSR-Algorithmus [24] zum Beispiel in die nächste Face ehe die st-Linie überquert wird und wählen als nächste Kante diejenige aus, die linksherum von der geschnittenen Kante liegt. Kuhn et al. [28] wählen und untersuchen dagegen eine Variante bei der die komplette Face erkundet wird. Die Face-Traversierung wird daraufhin an dem Punkt durch einen Face-Wechsel neu gestartet, der von den durchlaufenen am nächsten zur Zielposition liegt.

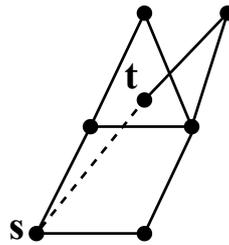


**Abb. 1.5:** Face-Routing: Eine Nachricht wird durch den Wechsel zwischen Face-Traversierung und Face-Wechsel schrittweise immer weiter Richtung Ziel geleitet

Die Abbildung 1.5 zeigt einen beispielhaften Ablauf von Face-Routing. Beginnend bei Startknoten  $s$  werden die grau hinterlegten Faces der Reihe nach anhand der Rechten-Hand-Regel durchquert. Die Face-Wechsel finden statt, ehe die st-Linie überschritten wird. Auf diese Art nähert man sich sukzessive dem Zielknoten  $t$ .

Allerdings ist es egal, ob man die Face entsprechend der Rechte- oder Linke-Hand-Regel durchläuft und welche Face-Wechsel-Strategie man zugrunde legt: Damit Face-

Routing korrekt arbeitet, muss der zugrundeliegende Graph planar sein. Das bedeutet, er darf keine schneidenden Kanten enthalten. Abbildung 1.6 (nach [26]) zeigt einen Fall, in dem der Graph noch schneidende Kanten aufweist, wodurch kein Weg zum Zielknoten gefunden wird. Egal ob die Rechte-Hand-Regel oder die Linke-Hand-Regel zur Anwendung käme, Face-Routing würde in dieser Topologie in einer Schleife ablaufen und fehlschlagen.



**Abb. 1.6:** Da der dargestellte Graph noch einen Schnitt enthält, läuft Face-Routing in dieser Topologie in einer Schleife ab und findet keinen Weg zum Zielknoten  $t$  (Abbildung nach [26]).

Der Verbindungsgraph eines drahtlosen Netzwerks ist allerdings nicht planar und enthält typischerweise viele schneidende Kanten. Daher muss zunächst mittels eines Planarisierungsalgorithmus ein planarer Subgraph  $S(G)$  aus dem realen Netzwerkgraphen  $G$  erstellt werden. Die Konstruktion einer solchen Substruktur  $S(G)$  ausgehend von der Originalstruktur  $G$  wird als Topologiekontrolle bezeichnet. Wünschenswerte Eigenschaften bei der Topologiekontrolle sind, dass die Algorithmen lokal sind und auf verteilte Art und Weise ablaufen. Unter diesen Voraussetzungen muss ein Knoten nur die eigene Position und die seiner direkten Nachbarn kennen und kann die Entscheidung, ob eine Kante Bestandteil des planaren Subgraphen ist, unabhängig von anderen Knoten treffen. Verbreitete Planarisierungsmethoden sind der Gabriel Graph (GG) [15] und der Relative Neighbourhood Graph (RNG) [46]. Beide Varianten verfolgen die gleiche Grundidee: Ein Knoten  $u$  kann die Kante  $(u, v)$  zu einem Nachbarknoten  $v$  dann löschen, wenn es einen alternativen Weg zu  $v$  über einen weiteren Nachbarknoten  $w$  gibt. Dieser Knoten  $w$  wird als Witness-Knoten bezeichnet und muss innerhalb einer festgelegten Fläche zwischen  $u$  und  $v$  liegen. Der Gabriel Graph und der Relative Neighbourhood Graph unterscheiden sich in der Art und Weise, wie diese Fläche bestimmt wird.

Beim Gabriel Graph existiert eine Kante  $(u, v)$  zwischen den Knoten  $u$  und  $v$ , wenn es keinen weiteren Knoten  $w$  gibt, der innerhalb des Kreises mit dem Durchmesser  $\overline{uv}$  liegt:

$$\forall w \neq u, v : d^2(u, v) < [d^2(u, w) + d^2(v, w)]$$

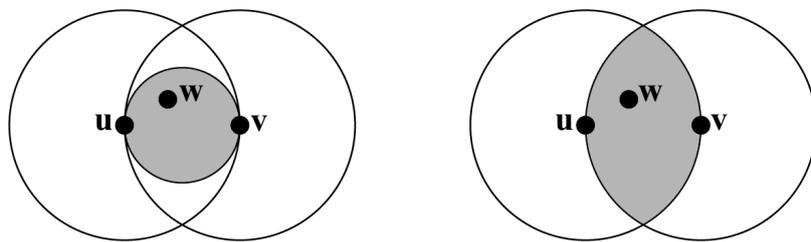
wobei  $d(u, v)$  dem Abstand zwischen Knoten  $u$  und Knoten  $v$  entspricht.

Beim Relative Neighbourhood Graph existiert eine Kante  $(u, v)$  zwischen den Kno-

ten  $u$  und  $v$ , wenn der Abstand  $d(u, v)$  zwischen ihnen kleiner oder gleich dem Abstand zu jedem anderen Knoten  $w$  ist, unabhängig davon ob  $u$  oder  $v$  weiter von  $w$  entfernt ist:

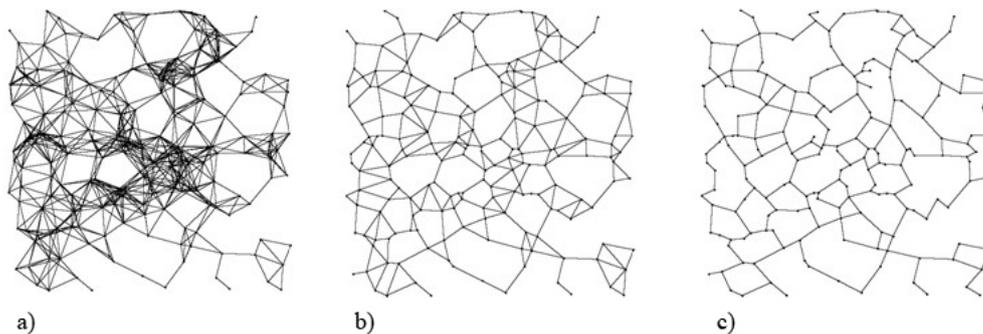
$$\forall w \neq u, v : d(u, v) \leq \max[d(u, w), d(v, w)]$$

Beim GG darf also innerhalb des Kreises mit Durchmesser  $\overline{uv}$  kein weiterer Knoten enthalten sein, damit die Kante  $(u, v)$  Teil des planaren Graphen ist, und beim RNG darf dafür kein weiterer Knoten in der Schnittfläche enthalten sein, die sich ergibt, wenn man jeweils um  $u$  und  $v$  einen Kreis mit dem Radius  $\overline{uv}$  zieht.



**Abb. 1.7:** Gabriel Graph (links) und Relative Neighbourhood Graph (rechts): Befindet sich ein Witness-Knoten  $w$  im schattierten Bereich ist die Kante  $(u, v)$  kein Bestandteil des planaren Graphen.

Abbildung 1.7 verdeutlicht dieses Vorgehen für den Gabriel Graphen (links) und für den Relative Neighbourhood Graphen (rechts). Befindet sich im grau hinterlegten Bereich ein Witness-Knoten  $w$  ist die Kante  $(u, v)$  nicht Teil des planaren Graphen.



**Abb. 1.8:** a) Originalgraph, b) GG, c) RNG (aus [24])

Der Unterschied zwischen einem Originalgraphen und den planarisierten Graphen ist gut in Abbildung 1.8 (aus [24]) zu sehen. Auf der linken Seite ist der vollständige Netzwerkgraph aus 200 Knoten abgebildet, die zufällig auf einer 2000 x 2000 Meter großen Fläche verteilt wurden (a). Im Vergleich dazu sieht man in der Mitte den daraus erhaltenen Gabriel Graphen (b) und rechts den entsprechenden Relative Neighbourhood Graphen (c).

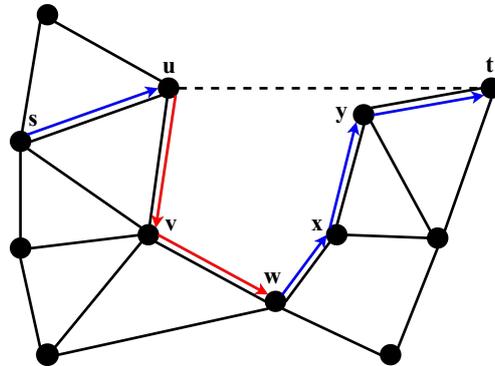
Sowohl der GG als auch der RNG gehen davon aus, dass der zugrundeliegende Graph ein Unit Disk Graph (UDG) ist. Das bedeutet, wenn man es auf einen Netzwerkgraphen überträgt, dass die Knoten in der euklidischen Ebene angeordnet sind und man für alle Knoten einen identischen, kreisförmigen Übertragungsradius  $r$  annimmt. Zwei Knoten  $u$  und  $v$  können also genau dann miteinander kommunizieren, wenn ihr euklidischer Abstand  $d(u, v)$  kleiner oder gleich dem eindeutigen Senderradius  $r$  ist und entsprechend werden sie in dem Fall im Netzwerkgraphen durch eine Kante miteinander verbunden.

Wurde unter Anwendung der UDG-Annahme mithilfe eines Planarisierungsalgorithmus der planare Subgraph  $S(G)$  erstellt, kann Face-Routing angewendet und so die Auslieferung garantiert werden, sofern es einen Pfad zum Zielknoten gibt. Gibt es keinen Weg zum Ziel, wird irgendwann eine Anfangskante ein zweites Mal in gleicher Richtung durchlaufen, was als Zeichen für einen Routingfehlschlag interpretiert werden kann.

### 1.2.3 Kombiniertes Verfahren

Face-Routing hat den gravierenden Nachteil, dass die auf diese Art generierten Pfade bei Weitem nicht optimal sind. Meist werden sehr ineffiziente Wege gefunden, bei denen Pakete auf dem Weg zum Ziel einen großen Umweg zurücklegen. Im Gegensatz dazu stellt Greedy-Routing eine effizientere Methode dar, hat aber das bereits erläuterte Problem mit dem lokalen Minimum. Aus diesem Grund schlugen Bose et al. mit Greedy-Face-Greedy (GFG) [3] sowie Karp und Kung mit GPSR [24] vor, Greedy-Routing und Face-Routing miteinander zu kombinieren. Diese Variante soll die Vorteile des jeweiligen Verfahrens nutzen, um die Nachteile des anderen auszugleichen. GFG bzw. GPSR umfassen dabei als Basiskomponenten das Greedy-Routing, die Planarisierung und das Face-Routing. Zunächst wird versucht mit Greedy-Routing auf dem Netzwerkgraphen  $G$  die Nachricht mit jedem Schritt näher ans Ziel zu bringen. Erreicht Greedy-Routing ein lokales Minimum und findet keinen Nachbarknoten, der eine kürzere Distanz zum Ziel aufweist, nutzt man Face-Routing als Recovery-Strategie. Das Face-Routing findet dann auf der Grundlage des planarisierten Subgraphen  $S(G)$  statt. Hierbei werden solange die Faces traversiert, bis man entweder das Ziel erreicht oder einen Knoten findet, der einen kleineren Abstand zum Ziel hat als der Knoten, bei dem das lokale Minimum erreicht wurde, je nachdem was früher eintritt. Im ersten Fall wurde der Routingauftrag erfolgreich ausgeführt und entsprechend kann der Algorithmus an dieser Stelle enden. Im zweiten Fall wird wieder auf Greedy-Routing gewechselt und weiter über den Originalgraphen  $G$  geroutet, bis man entweder das Ziel erreicht hat oder aufgrund eines neuen lokalen Minimums wieder auf Face-Routing umschaltet. Abbildung 1.9 (nach [42]) zeigt die verschiedenen Phasen von kombiniertem Greedy-Face-Routing. Vom Startknoten  $s$  wird zunächst via Greedy-Routing an Knoten  $u$  weitergeleitet. Aufgrund eines lokalen Minimums kommt nun Face-Routing zum Einsatz und durchläuft die Knoten  $v$  und  $w$ . Der Knoten  $w$  liegt näher am Ziel  $t$  als Knoten  $u$ , so dass nun wieder Greedy-Routing verwendet wird, was schließlich über den Weg

$x - y - t$  zum Ziel führt.



**Abb. 1.9:** Phasen bei kombiniertem Greedy-Face-Routing: Greedy  $s \rightarrow u$ , Face  $u \rightarrow w$ , Greedy  $w \rightarrow t$  (Abbildung nach [42])

Der Vorschlag, Greedy-Routing mit Face-Routing als Recovery-Strategie zu kombinieren, liefert einen vielversprechenden Ansatz für lokale, positionsbasierte Routingverfahren. Entsprechend viel wurde und wird in diesem Bereich erforscht, vorgeschlagen und überprüft. Ein Anliegen dabei ist es, die Performance des Verfahrens zu verbessern. Datta et al. [10] wollen in ihrer Arbeit die Performance steigern, indem sie die durchschnittlich benötigte Hop-Anzahl verringern. Dafür kombinieren sie eine möglichst frühe Rückkehr zum Greedy-Routing nach einem lokalen Minimum mit einer Strategie, bei der der Weg abgekürzt werden soll, indem man die lokalen Informationen analysiert und direkt zum letzten bekannten Hop routet. Zusätzlich läuft GFG lediglich über die Knoten der verbundenen Dominating Set mit Ausnahme des Start- und Endknotens. Die Dominating Set eines Graphen  $G = (V, E)$ , besteht aus den Knoten  $V$  und den Kanten  $E$ , ist eine Teilmenge  $D$  von  $V$ , so dass jeder Knoten entweder in  $D$  enthalten ist, oder direkt mit einem Knoten verbunden ist, der in  $D$  enthalten ist. Von einer verbundenen Dominating Set  $D$  spricht man, wenn jeder Knoten  $v_1 \in D$  jeden anderen Knoten  $v_2 \in D$  erreichen kann ( $v_1 \neq v_2$ ). Kuhn et al. stellen zunächst mit AFR [29] eine asymptotisch optimale Weiterentwicklung von Compass Routing II vor, gefolgt von den Verbesserungen GOAFR [31] und GOAFR+ [28]. Frey und Stojmenovic geben in [14] einen Überblick über verschiedene Face-Routing-Ansätze und setzen sich mit der Frage auseinander, welche Varianten für die Basiskomponenten in Frage kommen und sinnvoll sind. Hierbei geht es um die Wahl der ersten Kante beim Start von Face-Routing sowie die Entscheidung, ob die Rechte- oder Linke-Hand-Regel zum Einsatz kommt. Auch für den Zeitpunkt des Face-Wechsels ergeben sich verschiedene Alternativen, um dann nach einem Face-Wechsel wieder vor der Entscheidung zu stehen, welche Kante als erste durchlaufen werden soll und ob dies entsprechend der Rechte- oder Linke-Hand-Regel geschieht. In [13] findet man einen Überblick über geographische Routingverfahren und deren Verbesserungen im Bereich Greedy- und Face-Routing. In [42] vergleicht Stojmenovic verschiedene Routingansätze für positionsbasiertes Ruten

in Ad-hoc Netzwerken.

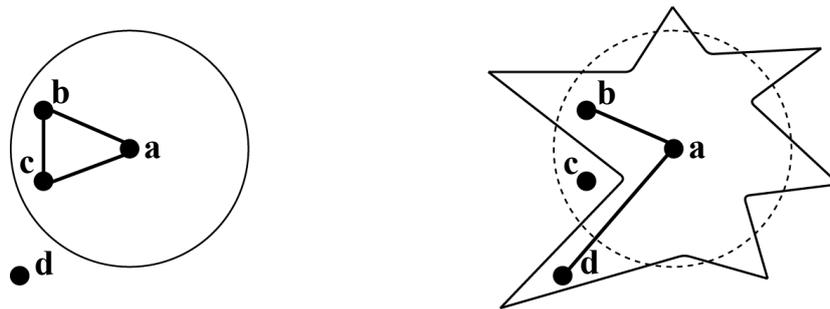
Die zugrunde gelegte Routingmetrik beeinflusst die Leistungsfähigkeit eines Routingalgorithmus. Folglich wird auch dieser Bereich erforscht. Li et al. [32] untersuchen zum Beispiel Routingmetriken im Zusammenspiel mit Greedy-, Face- und GFG-Routing.

Die Notwendigkeit eines planaren Graphen für die korrekte Arbeitsweise von Face-Routing ist der Grund dafür, dass Topologiekontrolle eine essentielle Basiskomponente ist und entsprechend auch weitreichend untersucht wird. So wurde in [9] eine Topologiekontrolle für drahtlose Sensornetzwerke mit veränderlicher Topologie vorgestellt, die nur geringen Wartungsaufwand verursacht. Gao et al. stellen in [17] den Restricted Delaunay Graphen (RDG) als neuen Routinggraphen vor und empfehlen, die Knoten in Clustern zu gruppieren und dann den RDG zwischen den Cluster-Heads zu erstellen. Beim Clustering werden die vorhandenen Knoten in mehrere disjunkte Mengen, die sogenannten Cluster, unterteilt. Der Cluster-Head fungiert dabei nach außen als Stellvertreterknoten für den gesamten Cluster und repräsentiert eine weitere Hierarchieebene. Auf diese Art soll der Stretchfaktor des Graphen verbessert werden. Als Stretchfaktor bezeichnet man das Verhältnis vom gefundenen zum kürzesten Pfad. Erkennt bei der Planarisierung durch GG oder RNG nur einer der beiden Endknoten einer potenziell entfernbaren Kante einen Witness-Knoten, führt dies zu inkonsistenten Entscheidungen. Ein Lösungsvorschlag für dieses Problem ist es, eine Kante nur dann zu entfernen, wenn beide Endknoten den Witness-Knoten erkennen und der Entfernung zustimmen [22, 41, 23]. Diese - als Mutual-Witness-Erweiterung bezeichnete - Übereinkunft kann leicht mit lokalen Mitteln überprüft werden, indem die Knoten ihre Nachbarschaftslisten untereinander austauschen. Auf diese Art können alle Knoten prüfen, ob ein bestimmter Nachbar einen anderen Knoten als gemeinsamen Nachbarn hat.

### 1.3 Fehlende Übertragbarkeit der UDG-Annahme auf realistische Drahtlosnetzwerke

All diese vorgestellten Varianten beinhalten die Tatsache, dass Face-Routing einen planaren Graphen benötigt, um die Auslieferung einer Nachricht garantieren zu können. Die Planarisierungsalgorithmen beruhen dabei auf der Unit-Graph-Annahme. Unit Disk Graphen bilden allerdings ein idealisiertes Netzwerk ab, indem davon ausgegangen wird, dass jeder Knoten den gleichen perfekten, kreisförmigen Übertragungsradius hat. Diese Vorstellung von einem idealisierten Netzwerk ist allerdings keine deckungsgleiche Abbildung von realen Netzwerken. Das Unit-Graph-Modell stellt eine starke Vereinfachung dar und eignet sich daher sehr gut, um theoretische Ansätze und Möglichkeiten zu untersuchen. Die Ergebnisse lassen sich aber nicht unmittelbar auf reale, drahtlose Ad-hoc Netzwerke übertragen. In der Praxis findet man ungleichmäßige Funkreichweiten [47, 49, 50] oder Szenarien, in denen lange und stabile sowie kurze und instabile Verbindungen auftreten [16]. Wetterkonditionen können die Signalausbreitung beeinträchtigen und so der Grund für

ungleichmäßige Funkreichweiten sein. Ebenso können Hindernisse wie Gebäude oder Berge die Ursache dafür sein, dass zwei Knoten keine direkte Verbindung haben, obwohl ihr Abstand zueinander kleiner als die eigentlich mögliche Übertragungsreichweite ist. Abbildung 1.10 (nach [18]) zeigt schematisch einen solchen Fall. Auf der linken Seite sind die Verbindungen entsprechend dem UDG-Modell dargestellt. Die rechte Seite zeigt einen Ansatz, der eher in der Realität zu finden ist. Hier kann es durchaus vorkommen, dass Knoten  $a$  zu dem nahen Knoten  $c$  keine Verbindung hat, wohl aber zu dem entfernten Knoten  $d$ .



**Abb. 1.10:** Verbindungen beim idealisierten UDG-Modell (links) und mögliche reale Verbindungen (rechts) (Abbildung nach [18])

Des Weiteren kann es zu Fehlern bei der Bestimmung der Knotenposition kommen, die die jeweiligen Knoten durch ein Lokalisationssystem erfahren. Lokalisationsfehler führen ebenfalls dazu, dass die Unit-Graph-Annahme verletzt wird [41].

Die Unit-Graph-Annahme kann also aus zwei Gründen nicht zutreffend sein: Ein Knoten hat eine falsche Einschätzung seiner eigenen Position oder aber es kommt zu Unregelmäßigkeiten innerhalb der angenommenen, einheitlichen Funkreichweite. Das Ergebnis kann eine fehlerhafte Planarisierung sein, bei dem die folgenden Defizite auftreten können: Es ist möglich, dass der Graph unterteilt wird, weil eine Kante fälschlicherweise gelöscht wurde, oder eine irrtümlich nicht gelöschte Kante führt dazu, dass der planarisierte Graph immer noch Schnitte enthält. Haben Knoten unterschiedliche Erkenntnisse im Bezug auf vermeintliche Witness-Knoten, können unidirektionale Verbindungen die Konsequenz daraus sein. All diese Planarisierungsfehler können zu persistenten Routingfehlern führen, sodass das Face-Routing fehlschlägt und für mindestens ein Knotenpaar keinen Weg findet [26, 25, 40].

## 1.4 Ziele der vorliegenden Arbeit

Um die Kluft zwischen Theorie und Praxis zu schließen, ergaben sich mehrere Ansätze, die ein wirklichkeitsgetreueres Modell als den Unit Disk Graphen zugrunde legen. Das nächste Kapitel gibt hierzu einen Überblick zu verwandten Arbeiten. Allerdings kam es auch schon zu der Vermutung, dass lokale Strategien nicht ausreichen, um Graphen mit beliebiger Struktur zu planarisieren [11, 40]. Santi [38]

kommt nun aber zu dem Ergebnis, dass lokale Mechanismen zur Topologiekontrolle eine essentielle Voraussetzung für die Skalierbarkeit von Drahtlosnetzwerken sind. Daraus ergibt sich die Frage, welchen Stellenwert der planare Graph für den Erfolg bei der Kombination aus Greedy- und Face-Routing eigentlich hat. Mit dieser Arbeit soll die Frage beantwortet werden, wie hoch die Fehlerrate bei diesem Routingverfahren ist, wenn der Graph lediglich teilplanarisiert ist. Mit lokalen Methoden ist es möglich, ohne viel Aufwand einen gewissen Anteil an Kantenschnitten zu eliminieren, indem man lediglich die Verbindungen der Endpunkte der beiden schneidenden Kanten betrachtet. Die Entscheidung, ob zwei Kanten eine direkte Verbindung zueinander haben, wird hierbei nicht auf Grundlage des Unit-Graph-Modells getroffen. Stattdessen wird das Log-Normal-Shadowing-Modell (LNSM) genutzt, wodurch eine realistischere Abbildung von drahtlosen Ad-hoc Netzwerken erreicht wird.

Auf dieser Grundlage wurden in Simulationen Konnektivitätsgraphen erstellt und diese mit zwei unterschiedlichen Verfahren teilplanarisiert. Für diese Graphen wurde die Erfolgsrate von Greedy-Routing, Face-Routing und kombiniertem GFG-Routing getestet, indem jeweils genau ein Routingauftrag von einem zufälligen Startknoten an einen zufälligen Zielknoten simuliert wurde. Für die Testläufe wurden neun verschiedene Settings definiert, die sich im Hinblick auf die Graphendichte und die Standardabweichung  $\sigma$  der normalverteilten Zufallsvariablen  $X_\sigma$  des Log-Normal-Shadowing-Modells unterscheiden. Für jedes Setting wurden 2000 Durchläufe simuliert und ausgewertet.

Kapitel 2 stellt verwandte Arbeiten vor, die sich mit geographischen Routingverfahren unter realistischeren Modellen als dem Unit Disk Graphen beschäftigen. In Kapitel 3 werden die Grundlagen der Simulation erläutert, sowie der genaue Ablauf und die Ausgaben der Simulationsdurchläufe dargelegt. Kapitel 4 umfasst die Ergebnisse dieser Simulationsdurchläufe und deren Auswertung. Schließlich erfolgt in Kapitel 5 eine Bewertung der Erkenntnisse.

# Kapitel 2

## Geographische Routingverfahren in realistischeren Settings

Im Folgenden soll ein Überblick über Arbeiten gegeben werden, die sich mit geographischen Routingverfahren unter realistischeren Annahmen als dem Unit-Disk-Graph-Modell beschäftigten.

### 2.1 Quasi Unit Disk Graphen

Zu den frühen Arbeiten, die einen Schritt in Richtung realistischer Modellierung gehen, gehören die Arbeiten von Barrière et al. [2] und Kuhn et al. [30], die den Quasi Unit Disk Graph (QUDG) nutzen. Der Quasi Unit Disk Graph ist eine Verallgemeinerung des Unit Disk Graphen und lässt Schwankungen in der Übertragungreichweite zu. Dafür definieren Barrière et al. eine minimale Übertragungreichweite  $r$  und eine maximale Übertragungreichweite  $R$ . Die von ihnen vorgestellte Variante des Face-Routings garantiert die Auslieferung, solange das Verhältnis der maximalen zur minimalen Übertragungreichweite höchstens  $\sqrt{2}$  beträgt. Knoten, deren Abstand  $d$  zueinander kleiner oder gleich der minimalen Übertragungreichweite  $r$  ist, können direkt miteinander kommunizieren. Knoten mit einem Abstand von  $d$  größer als  $R$  sind nicht miteinander verbunden. Gilt für die Entfernung zweier Knoten  $r < d \leq R$  kann es sein, dass sie miteinander kommunizieren können oder auch nicht. Das vorgestellte Routingprotokoll besteht aus drei Phasen. In der Completion-Phase werden zunächst virtuelle Kanten zu dem Originalgraphen  $G$  hinzugefügt. Damit soll erreicht werden, dass auch der planarisierte Graph später noch verbunden ist. Aus dem so entstandenen Supergraphen  $S(G)$  wird in der Extraction-Phase der Gabriel Graph gebildet, ehe in der Routing-Phase kombiniertes Greedy- und Face-Routing zum Einsatz kommt. Abbildung 2.1 (nach [33]) auf Seite 17 zeigt dieses Verfahren in der mittleren Abbildung (b). In der Completion-Phase werden die gestrichelten, virtuellen Kanten hinzugefügt, ehe in der Extraction-Phase die schneidenden Kanten entfernt werden. Für die Einschränkung, dass das Verhältnis der Übertragungreichweiten höchstens  $\sqrt{2}$  beträgt, konnte die Richtigkeit des Protokolls bewiesen werden. Allerdings versagt die vorgestellte Strategie, wenn diese Einschränkung nicht gilt oder

unidirektionale Kanten auftreten. Ein weiteres Problem dieses Ansatzes ist die Tatsache, dass virtuelle Kanten durch tatsächlich existierende Pfade realisiert werden müssen und diese beliebig lang werden können.

Kuhn et al. [56/12] stellen ein weiteres Verfahren im Bereich der Quasi Unit Disk Graphen vor. Setzt man bei einem  $d$ -QUDG den Wert von  $d$  auf mindestens  $1/\sqrt{2}$ , kann man lokal virtuelle Kanten einsetzen und die klassischen Strategien zu geographischem Routing nutzen. Dabei sind die gleichen Performance-Garantien zu erreichen, wie auch beim Unit Disk Graphen. Zur Topologiekontrolle wird ein Algorithmus vorgestellt, der aus dem Netzwerkgraphen  $G$  zunächst die verbundene Dominating Set (CDS) extrahiert (nach dem Verfahren wie in [1] beschrieben). Anschließend wird durch Clustering die verwendete Knotenanzahl reduziert und man erhält den Subgraphen  $H(G)$ . Als Routingalgorithmus wird GEcho vorgestellt, eine Kombination aus Greedy-Routing und Echo-Flooding [8, 36]. Die Motivation dieser Arbeit war es, ein Modell zu nutzen, das die grundlegenden Eigenschaften von Ad-hoc Netzwerken erfasst, aber trotzdem noch einfach genug ist, um bindende theoretische Ergebnisse zu erhalten. Entsprechend zeigt das Verfahren auch eine realistischere Variante als den reinen Unit Disk Graphen auf. Allerdings ist auch der  $d$ -QUDG noch eine starke Vereinfachung der Realität, da das Verhältnis von Maximal- und Mindestreichweite in real existierenden Netzwerken nicht auf  $\sqrt{2}$  beschränkt ist.

## 2.2 Log-Normal-Shadowing-Modell

Eine Variante, um Konnektivitätsgraphen zu erhalten, die eher der Realität entsprechen als der UDG oder der QUDG ist das Log-Normal-Shadowing-Modell. Auf diesem Modell basieren die Arbeiten von Xu et al. [48] und Hekmat und van Mieghem [18]. Das Log-Normal-Shadowing-Modell ist ein Verfahren, das nicht nur die Beziehung zwischen gesendeter und empfangener Leistung bei drahtlosen Signalen und der Distanz zwischen den Knoten nutzt, sondern auch Parameter bereitstellt, um es entsprechend verschiedener Umgebungsstrukturen zu konfigurieren. Damit ist es ein allgemeines Modell, das sowohl innerhalb von Gebäuden als auch Outdoor angewendet werden kann. Allerdings basieren diese Parameter auf Erfahrungen und Testläufen und sind somit nicht selbstanpassungsfähig. Daher haben Xu et al. eine Funktion aufgestellt, um ein Log-Normal-Shadowing-Modell mit dynamischer Varianz (LNSM-DV) zu generieren. Dazu nutzen sie die Least-Square-Methode um die Koeffizienten im Modell abzuschätzen, wodurch sich LNSM-DV dynamisch entsprechend der Umgebung anpassen kann. Es konnte experimentell nachgewiesen werden, dass das LNSM durch diese Erweiterung im Hinblick auf Genauigkeit und Selbstanpassung verbessert werden konnte. Sie merken allerdings an, dass die Least-Square-Methode eine gewisse Menge an Stichproben benötigt, was im Knotenlayout entsprechend Berücksichtigung finden sollte.

Hekmet und van Mieghem haben die Konnektivität in drahtlosen Ad-hoc Netzwerken untersucht, wenn der Netzwerkgraph auf dem Log-Normal-Shadowing-Modell

aufbaut. Sie haben dieses Modell ausgewählt, da die Verbindung zwischen zwei Knoten nicht nur von deren Entfernung zueinander abhängig ist. Eine zufällige Abweichung simuliert Schwankungen in der Sendeleistung, wie sie in der Realität aufgrund von Hindernissen, ungleichmäßigen Senderadien oder Ähnlichem vorkommen. Dazu beschreiben sie einen Parameter  $\xi$ , der sich auf die Schwankungen des Funksignals bezieht.  $\xi$  berechnet sich aus der Standardabweichung der Normalverteilung durch den Pfadverlustkoeffizienten<sup>1</sup>. Ein kleiner Wert für  $\xi$  repräsentiert geringe Schwankungen des Funksignals, ein großer demzufolge stärkere. Entsprechend konnten in der Simulation bei großen  $\xi$ -Werten eine erhöhte Wahrscheinlichkeit für lange Kanten festgestellt werden. In der Arbeit wird angemerkt, dass die 1-Konnektivität, also die volle Konnektivität, eine zu strenge Bedingung bei Ad-hoc Netzwerken ist. Daher schlagen sie vor, die Größe der Giant Component als Maß für die Konnektivität eines Netzwerkgraphen zu nutzen. Ein unzusammenhängendes Netzwerk kann aus verschiedenen, in sich verbundenen, Inseln oder Clustern bestehen. Der größte dieser Cluster wird als Giant Component bezeichnet. Dazu haben sie eine Gleichung hergeleitet, um die Größe der Giant Component als eine Funktion über dem durchschnittlichen Knotengrad und dem Parameter  $\xi$  zu berechnen. Diese Formel kann genutzt werden, um die benötigte durchschnittliche Menge an Nachbarknoten zu bestimmen, die realisiert werden muss, um ein gewünschtes prozentuales Maß an Konnektivität zu erhalten. Der Durchschnittsknotengrad kann schließlich über die Justierung der Übertragungsstärke oder eine Anpassung in der Dichte des Netzwerkgraphen beeinflusst werden.

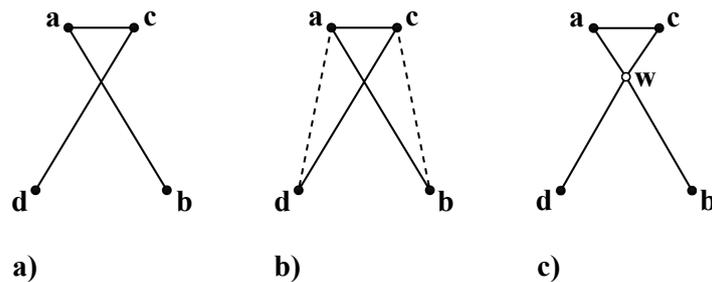
## 2.3 Ansätze mit neuen Planarisierungsvarianten

Fehlschläge bei geographischen Routingverfahren unter realistischen Bedingungen lassen sich auf Fehler in der Planarisierung zurückführen [26, 40, 33]. Daher untersuchen die folgenden Arbeiten die Ursachen und Auswirkungen von fehlerhafter Planarisierung und stellen Lösungsvorschläge für dieses Problem vor.

Lillis et al. [33] nutzen in ihrer Betrachtung ebenfalls den  $d$ -QUDG mit  $d \geq 1/\sqrt{2}$ . Es wird ein lokales Protokoll zur Topologiekontrolle vorgestellt, dessen Basisidee recht einfach ist: Jeder Kantenschnitt wird durch einen virtuellen Knoten ersetzt. Das Verfahren ist in Abbildung 2.1 (nach [33]) auf der rechten Seite (c) dargestellt. Am Schnittpunkt der beiden Kanten  $(a, b)$  und  $(c, d)$  wird ein virtueller Knoten  $w$  hinzugefügt. Die beiden schneidenden Kanten werden entfernt und durch neue Kanten  $(w, a)$ ,  $(w, b)$ ,  $(w, c)$  und  $(w, d)$  ersetzt. Zusätzlich muss ein Knoten der Menge  $\{a, b, c, d\}$  als Stellvertreter für den virtuellen Knoten  $w$  fungieren, da ein Pfad durch den virtuellen Knoten  $w$  in dem Netzwerk dadurch realisiert wird, dass er über dessen Stellvertreterknoten führt. Da Stellvertreterknoten nicht nur die eigene Nachbarschaftstabelle, sondern auch die des virtuellen Knotens pflegen müssen, sollte jeder Knoten nur für eine begrenzte Menge an virtuellen Knoten als Stellvertreter dienen,

<sup>1</sup>Eine detailliertere Beschreibung der einzelnen Parameter des Log-Normal-Shadowing-Modells findet sich in Kapitel 3 dieser Arbeit.

und die Tabellen sollten nicht sehr groß werden. Um dies zu gewährleisten, wird eine Clustering-Technik (ähnlich zu [17]) vorgeschlagen und aufgezeigt. Das geographische Routing erfolgt dann zwischen den Cluster-Heads des entstandenen Subgraphen. Diese Strategie führt dazu, dass der Nachrichtenverkehr sehr ungleichmäßig verteilt ist, wodurch die Cluster-Heads mit der Weiterleitung überlastet werden könnten. Als Lösung wird vorgeschlagen, die Rolle der Cluster-Heads rotieren zu lassen oder aber die Nachricht zwar von Cluster zu Cluster, nicht aber zwingend an die Cluster-Heads zu senden. Doch auch wenn durch eine Erweiterung des Algorithmus die Problematik der ungleichmäßigen Lastenverteilung zu lösen wäre, bleibt immer noch die Diskrepanz zwischen dem QUDG und realistischen Netzwerken. Die Realisierung einer Route über virtuelle Knoten durch die Stellvertreterknoten führt dazu, dass lange Pfade entstehen können. Lediglich mit der Beschränkung, dass  $d$  mindestens  $1/\sqrt{2}$  beträgt konnte nachgewiesen werden, dass die Hop-Distanz von einem Stellvertreter zu einem anderen beschränkt ist. Es ergibt sich also eine ähnliche Problematik wie in dem Ansatz von Barrière et al. bezüglich der realen Umsetzung von virtuell eingefügten Elementen.

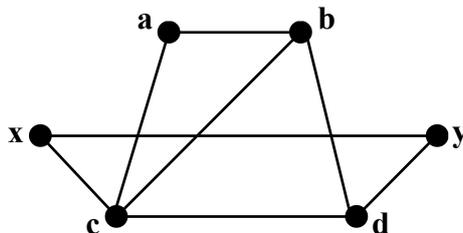


**Abb. 2.1:** Zur Planarisierung des Originalgraphen (a) werden nach dem Verfahren von Barrière et al. virtuelle Kanten hinzugefügt, ehe die Schnitte gelöscht werden (b). Lillis et al. nutzen einen virtuellen Knoten am Schnittpunkt und ersetzen die schneidenden Kanten durch vier neue, mit dem virtuellen Knoten verbundene, Kanten (c) (Abbildung nach [33]).

Seada et al. [40] analysieren in ihrer Arbeit die Richtigkeit von geographischem Routing unter nicht-idealen Voraussetzungen. Dabei betrachten sie detailliert die Auswirkungen von Positionsfehlern, die im Umfang von 1-10 % der Funkreichweite als durchaus realistisch angesehen werden können. Sie kommen zu dem Ergebnis, dass Planarisierungsfehler in Form von unterteilten Graphen oder zurückbehaltenen Kantenschnitten die Ursache für die Fehlschläge bei geographischen Routingverfahren sind. In den betrachteten Szenarien reichte es aus, dass lediglich ein Knoten die falschen Positionsinformationen hatte, um die beschriebenen Planarisierungsfehler zu beobachten. Probleme sind bei geographischen Routingverfahren in dünnen Graphen wahrscheinlicher zu beobachten als in dichten. Dabei stellt vor allem die Unterteilung des Graphen bei der Planarisierung ein Problem dar. Daher schlagen sie als Lösung einen lokalen Ansatz vor, der der Mutual-Witness-Erweiterung ent-

spricht. Eine Kante darf nur dann entfernt werden, wenn beide Endpunkte definitiv mit dem Witness-Knoten verbunden sind. Jedoch wird dadurch nicht das Problem der Kantenschnitte gelöst. Ihre Überlegungen führen zu dem Ergebnis, dass lokale Mittel nicht ausreichen, um aus einem beliebigen Netzwerkgraphen einen verbundenen, planaren Graphen zu generieren. Stattdessen kann per Deduktion gezeigt werden, dass jede Suche über eine feste Hopanzahl  $k$  hinweg nicht ausreicht, um alle Schnitte in einem beliebigen Konnektivitätsgraphen zu erkennen.

Auch Kim et al. [26] beschäftigen sich in ihrer Arbeit mit dem Problem, dass Verletzungen der UDG-Annahme zu Planarisierungsfehlern und somit zu dauerhaften Routingmisserfolgen führen. In Simulationen konnten sie feststellen, dass kombiniertes GFG-Routing bei den Planarisierungsstrategien Gabriel Graph und Relative Neighbourhood Graph dazu führen, dass bei fehlender Einhaltung der UDG-Annahme nur knapp 70 % der Knoten miteinander kommunizieren können. Der Einsatz der Mutual-Witness-Erweiterung führt dazu, dass sich immerhin knapp 90 % der Knoten gegenseitig erreichen können. Allerdings sind 10 % unverbundene Knoten immer noch ein inakzeptabler Wert. Da die Fehler auf unzureichende Planarisierungsverfahren zurückzuführen sind, stellen sie eine neue Planarisierungsvariante vor, das Cross-Link Detection Protocol (CLDP). Die Grundidee von CLDP ist, dass jeder Knoten jede seiner Kanten überprüft, indem er die zu testende Kante durchläuft und anschließend entsprechend der Rechte-Hand-Regel den Graphen solange traversiert, bis er wieder an dem Knoten ankommt. Wird unterwegs ein Schnitt detektiert, wird gegebenenfalls eine der beiden Kanten entfernt, falls der Graph dadurch nicht unterteilt würde. Wird eine Kante bei der Traversierung doppelt durchlaufen - einmal in jeder Richtung - ist dies ein Zeichen dafür, dass die Kante nicht entfernt werden darf, da es sonst zu einer Aufteilung des Graphen kommen würde. In Simulationen konnte der Vorteil von CLDP gezeigt werden, da zuverlässig Routingpfade zwischen allen Knotenpaaren gefunden wurden. CLDP bietet also garantierte Auslieferung, hat aber den großen Nachteil, dass es dafür die nicht-lokale Untersuchung des Netzwerkgraphen benötigt. Es werden also unter Umständen sehr lange Pfade durchlaufen, um die Schnitte aufzuspüren. Außerdem finden sich Topologien, bei denen Schnitte nicht erkannt werden. Ein Beispiel hierfür ist in Abbildung 2.2 (nach [26]) dargestellt. Bei der Überprüfung der Kante  $(b, c)$  bzw.  $(c, b)$  können die Endknoten den Schnitt mit der Kante zwischen  $x$  und  $y$  nicht feststellen.



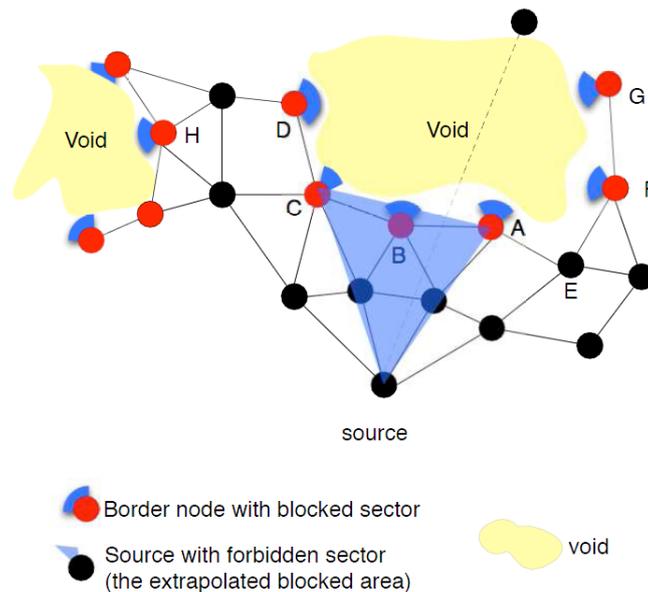
**Abb. 2.2:** Beispieltopologie für einen nicht mit CLDP erkennbaren Schnitt (Abbildung nach [26]).

## 2.4 Ansätze ohne Planarisierung

Auf vielfältige Weise hat sich gezeigt, dass die Planarisierung ein großes Problem darstellt. Mit lokalen Mitteln ist sie nach aktuellem Wissen nicht für beliebige Netzwerkgraphen umsetzbar, während sich bei nicht-lokalen Methoden wiederum Schwierigkeiten mit der Skalierbarkeit ergeben. Der eigentliche Vorgang erzeugt Overhead und führt zu Distanzverzerrungen und langen Pfaden aufgrund der Kantenentfernung. Ein weiterer kritischer Punkt ist die Frage der Anpassungsfähigkeit an die Netz-dynamik (Einfügen/Entfernen von Knoten/Kanten). Folglich gibt es Bemühungen, geographische Routingverfahren zu realisieren, die keine Planarisierung benötigen. Solch einen Routingalgorithmus, der ohne Planarisierung auskommt, stellen Casari et al. vor [7]. ALBA-R ist eine verteilte Variante, die auf beliebige Graphen angewendet werden kann und den Designansatz der Cross-Layer-Methode nutzt. Hierbei bricht man die strikte Schichtenarchitektur auf. Dadurch soll ein umfangreicherer Blick auf das Netzwerk ermöglicht werden, um somit anhand von mehr verfügbaren Informationen, bessere Routingentscheidungen treffen zu können. Möchte ein Knoten eine Nachricht weiterleiten, sendet er eine Relay-Selection-Message an alle seine Nachbarn. Jeder Nachbar kann nun seine eigene "Brauchbarkeit" im Bezug auf diesen Routingauftrag auf der Grundlage seines aktuellen Status (z.B. Paketfehlerrate, Restenergie, aktuelle Queue-Belegung usw.) berechnen. Ein Routingalgorithmus kann nun die Wahl des geeigneten nächsten Hops durch eine Kombination aus geographischen Daten und der Frage, welcher Knoten den besten Nutzen im Bezug auf eine bestimmte Metrik hat, treffen. Beispiele für diese Art von Routingverfahren sind GeRaF [51] und ALBA [6], die allerdings beide fehlschlagen, wenn sie ein lokales Minimum erreichen. Daher wird die Rainbow-Erweiterung für ALBA vorgeschlagen, bei der Knoten, in Abhängigkeit von ihrer Weiterleitungsfähigkeit, farblich markiert werden. Grundsätzlich sind die Knoten als gelb markiert. Stellt ein Knoten allerdings fest, dass er ein lokales Minimum ist, wechselt er seine Färbung zu rot, wodurch die Wahrscheinlichkeit, als nächster Hop ausgewählt zu werden, drastisch sinkt. Ein roter Knoten versucht nicht mehr, Pakete in Zielrichtung zu leiten, sondern zu einem gelben Knoten. Findet er in seiner Nachbarschaftsliste keinen gelben Knoten, wechselt seine Kolorierung zu blau. Auf diese Art wird durch die Farbgebung eine Bewertungshierarchie von "besseren" hin zu "schlechteren" Knoten gebildet. Ein Knoten leitet Pakete nur an solche Nachbarn weiter, die mindestens die gleiche Farbe haben wie er selbst, bevorzugt aber grundsätzlich die "bessere" Färbung. Zusätzlich werden die Nachbarn eines Knotens in zwei Mengen unterteilt, die alternierend durchsucht werden. Die Menge der genutzten Farbabstufungen ist frei wählbar. Mehr Farben führen zu einem besseren Routingergebnis, erhöhen allerdings gleichzeitig den Aufwand. Es konnte nachgewiesen werden, dass ALBA-R schleifenfrei ist und sowohl in dünnen als auch in dichten Graphen zu garantierter Auslieferung führt. Da keine Planarisierung benötigt wird, ist der Overhead geringer als bei Verfahren, die Face-Routing als Basiskomponente enthalten.

Eine weitere Variante, die keinen planaren Graphen benötigt, schlagen Theoleyre et al. vor [45]. Sie setzen bei Reactive Deflection an dem Punkt an, wenn Greedy-

Routing beim Erreichen eines lokalen Minimums fehlschlägt und schlagen eine Strategie vor, die Voids reaktiv erkennen und daraufhin vermeiden soll. Begreift ein Knoten sich als lokales Minimum für eine bestimmte Zielrichtung, propagiert er diese Erkenntnis in Richtung der Knoten, die im Pfad vor ihm liegen und gilt daraufhin als blockiert für diese Zielrichtung. Die Knoten nutzen ihn dann nicht mehr als nächsten Hop, sondern wählen den nicht-blockierten Nachbarknoten, der die kürzeste Entfernung zum Zielknoten hat. Knoten, die nur noch blockierte Folgeknoten haben, werden ebenfalls als blockiert markiert. Um die Menge an überflüssigen Paketübermittlungen und die Zeit, die das Netzwerk benötigt um zu konvergieren, zu verringern, wurde das Verfahren um einen Optimierungsschritt erweitert. Hierbei wird die Größe des blockierten Bereichs extrapoliert. Hat ein Knoten Informationen darüber, dass es in einer Richtung blockierte Knoten gibt, kann er den minimalen Winkel bestimmen, der die blockierten Knoten umfasst, und daraufhin alle Knoten vermeiden, die in diesen Winkel hineinfallen. Abbildung 2.3 (aus [45]) zeigt ein Beispiel, in dem das Wissen eines Knotens die 3-Hop-Nachbarschaft umfasst. Knoten B ist in Zielrichtung blockiert. Daraufhin erzeugt die Quelle eine Liste, die Knoten B und alle seine blockierten Nachbarn umfasst. Aus den Positionsinformationen dieser Knoten kann nun der blockierte Bereich berechnet werden.



**Abb. 2.3:** Beispiel für die Erkennung einer Void mit der Reactive-Deflection-Methode (aus [45])

In Simulationen wurde die Performance von Reactive Deflection im Vergleich zum Greedy-Routing getestet. Das Wissen um blockierte Knoten umfasste dabei die 3-Hop-Nachbarschaft. Bei Reactive Deflection konnte eine geringere Paketverlustrate als bei Greedy-Routing festgestellt werden. Bei beiden Verfahren ist allerdings ein Schwellenwert bezüglich der Graphendichte zu erkennen. Liegt der durchschnittliche Knotengrad unterhalb von diesem Wert, verzeichnet man einen starken Anstieg

der Paketverlustrate. Bei Reactive Deflection ist dieser Schwellenwert etwa 30 % niedriger als bei klassischem Greedy-Routing. Doch auch hier kann der Verlust von Paketen nicht komplett vermieden werden. Der Vergleich der Pfadlängen ist weniger aussagekräftig, da bei Greedy-Routing nur die erfolgreichen Versuche betrachtet wurden. Aufgrund der Umgehung von Voids umfasst die Pfadlänge bei Reactive Deflection entsprechend mehr Hops. Abgesehen davon, dass auch mit diesem Verfahren nicht die Auslieferung garantiert werden kann, bezieht es sich auch speziell auf Netze unter der Annahme, dass genügend Speicher, Energie und Rechenpower zur Verfügung stehen. Daher stellt sich grundlegend die Frage, ob es bei limitierter Ressourcenverfügbarkeit, wie es vor allem bei Sensornetzen typisch ist, überhaupt Anwendung finden könnte.

# Kapitel 3

## Simulationsaufbau

Die Simulation der verschiedenen Routingverfahren unter voneinander abweichenden Bedingungen stellt die Untersuchungsgrundlage dieser Arbeit dar. Daher wird in diesem Kapitel zunächst die Theorie der Basiskomponenten, also des Log-Normal-Shadowing-Modells, der Planarisierungsvarianten und der verwendeten Routingverfahren, beschrieben. Anschließend wird erläutert, wie diese in der Simulation praktisch umgesetzt wurden, welche Bedingungen für die Simulationsdurchläufe gelten und welche Settings untersucht wurden. Schließlich werden die Ausgaben und Visualisierungsmöglichkeiten der Simulationsumgebung abgegrenzt.

### 3.1 Basiskomponenten

#### 3.1.1 Log-Normal-Shadowing-Modell

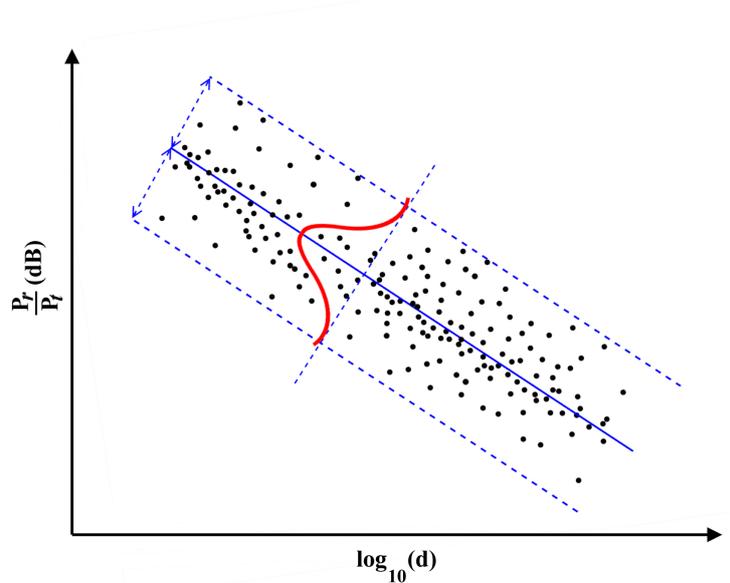
Das Log-Normal-Shadowing-Modell ist ein Funkmodell, das neben der Signalabschwächung über die Distanz zusätzlich eine statistische Abweichung der Signalstärke um einen Mittelwert herum in Betracht zieht. Sendet eine Quelle  $x$  an einen Empfänger  $y$ , kann der Pfadverlust, also die Stärke der Signalabnahme über die Distanz, berechnet werden. Ein Knoten kann eine Nachricht korrekt empfangen, wenn die Empfangsstärke größer oder gleich einem definierten Schwellenwert  $c$  ist. Betrachtet man nur diesen von der Distanz abhängigen Pfadverlust, würde der entstandene Konnektivitätsgraph einem Unit Disk Graphen entsprechen, sofern für alle Knoten der gleiche Schwellenwert  $c$  angenommen wurde. Um das Vorhandensein von Hindernissen, Reflexionspfaden und der Mehrwegeausbreitung mit in Betracht zu ziehen, nutzt das Log-Normal-Shadowing-Modell eine normalverteilte Zufallsvariable  $X_\sigma$ . Auf diese Weise wird berücksichtigt, dass ein Signal durch Dämpfung abgeschwächt oder auch durch Reflexion verstärkt werden kann. Der Gesamtpfadverlust zwischen dem Sender  $x$  und dem Empfänger  $y$  berechnet sich dann gemäß der folgenden Formel:

$$\overline{\text{PL}}(x, y) = \text{PL}_0 + 10 \gamma \log_{10} \frac{d}{d_0} + X_\sigma$$

Dabei repräsentieren die Variablen Folgendes:

- $PL_0$ : Pfadverlust bis zur Referenzdistanz  $d_0$
- $\gamma$ : Pfadverlustkoeffizient
- $d$ : Distanz zwischen den Knoten  $x$  und  $y$
- $d_0$ : Referenzdistanz in Metern
- $X_\sigma$ : normalverteilte Zufallsvariable mit der Standardabweichung  $\sigma$

Abbildung 3.1 illustriert die Berechnung durch das Log-Normal-Shadowing-Modell. Die blaue, durchgezogene Linie entspricht der Signalabschwächung über die Distanz. Hinzu kommt die normalverteilte Zufallsvariable. Der Abstand dieser beiden parallelen, blauen, gestrichelt dargestellten Linien wird durch die Standardabweichung  $\sigma$  bestimmt. Ein großer  $\sigma$ -Wert repräsentiert entsprechend stärkere und ein kleiner  $\sigma$ -Wert nur geringe Schwankungen.



**Abb. 3.1:** Das Log-Normal-Shadowing-Modell zieht neben der Signalabschwächung über die Distanz (blaue durchgezogene Linie) auch eine statistische Abweichung mit in Betracht.

Die beiden Knoten  $x$  und  $y$  können direkt miteinander kommunizieren, wenn die folgende Bedingung erfüllt ist:

$$\overline{PL}(x, y) \geq c,$$

wobei  $c$  den Schwellenwert in Dezibel angibt. Der Pfadverlust ist eine Funktion die anzeigt, wie die Signalstärke mit steigender Distanz abfällt. Der Pfadverlustkoeffizient  $\gamma$  beschreibt dabei, wie stark diese Funktion fällt. Auf diese Art ist es möglich,

das Modell an verschiedene Umgebungen anzupassen. Mit Hilfe der Standardabweichung  $\sigma$  der Normalverteilung kann das Ausmaß der Schwankungen aufgrund von äußeren Einflüssen wie Hindernissen usw. beeinflusst werden. Tabelle 3.1 (aus [21]) enthält Referenzwerte für  $\gamma$  und  $\sigma$  für unterschiedliche Geländeprofile.

Location	Average of $\gamma$	Average of $\sigma^2$ [dB]
Engineering building	1.9	5.7
Apartment hallway	2.0	8.0
Parking structure	3.0	7.9
One-sided corridor	1.9	8.0
One-sided patio	3.2	3.7
Concrete canyon	2.7	10.2
Plant fence	4.9	9.4
Small boulders	3.5	12.8
Sandy flat beach	4.2	4.0
Dense bamboo	5.0	11.6
Dry tall underbrush	3.6	8.4

**Tabelle 3.1:** Standardwerte für  $\gamma$  und  $\sigma$  (aus [21])

Aufgrund der vielen Parameter ist das Log-Normal-Shadowing-Modell ein flexibles und an verschiedene Szenarien anpassbares Funkmodell. Durch die Einbeziehung der Zufälligkeit in Form der normalverteilten Zufallsvariablen können realistische Settings simuliert und nachgestellt werden. Daher ist es sehr gut als Grundlage für die Erstellung eines realitätsnahen Konnektivitätsgraphen geeignet, um anschließend zu untersuchen, wie sich die verschiedenen Routingverfahren verhalten.

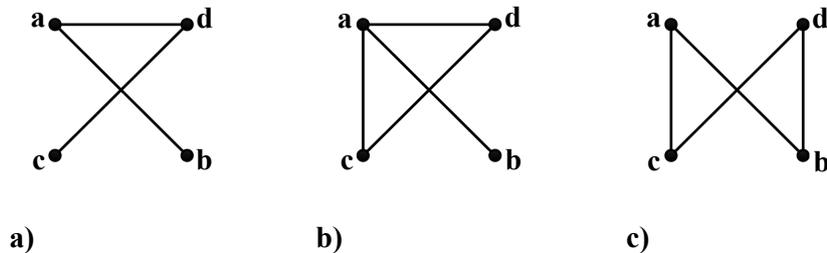
### 3.1.2 Planarisierungsvarianten

Die Planarisierung soll anhand einer sehr einfachen Strategie erfolgen, die auf der einen Seite niemals zur Unterteilung des Graphen führen darf, und auf der anderen Seite nur lokale Informationen nutzt. Wenn ein Schnitt zweier Kanten erkannt wurde, wurden daher lediglich die Verbindungen der vier beteiligten Endknoten betrachtet, um zu entscheiden, ob eine Kante gelöscht werden darf. Dabei können bei einem Schnitt zwischen den Kanten  $(a, b)$  und  $(c, d)$  drei verschiedene Fälle auftreten:

- Variante 1:

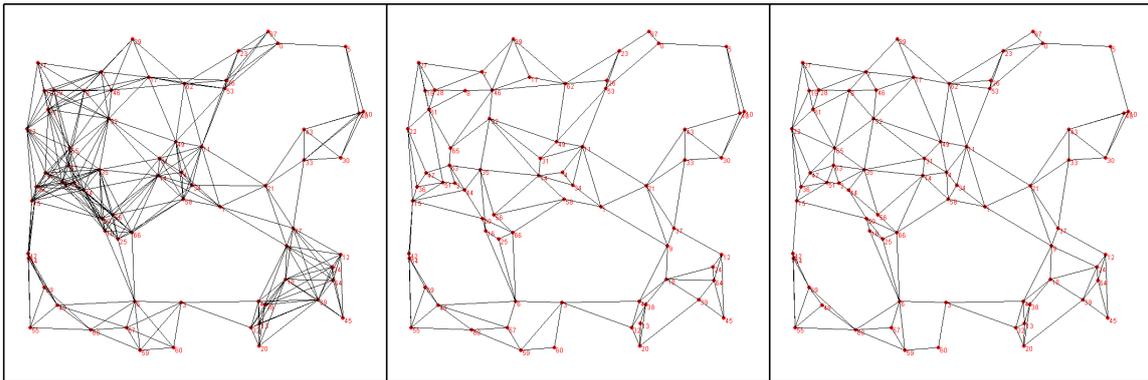
Abgesehen von den beiden schneidenden Kanten  $(a, b)$  und  $(c, d)$  gibt es zwischen den Knoten  $a$ ,  $b$ ,  $c$  und  $d$  höchstens eine weitere Verbindung. Diese Variante ist auf der linken Seite in Abbildung 3.2 (a) dargestellt. Außer den beiden sich überschneidenden Verbindungen gibt es lediglich eine Kante zwischen Knoten  $a$  und Knoten  $d$ . In dem Fall darf keine der beiden schneidenden Kanten entfernt werden, da sonst entweder Knoten  $c$  oder Knoten  $b$  nicht mehr erreichbar wäre.

- Variante 2:  
Beide Endknoten einer Kante sind mit dem gleichen Endknoten der anderen Kante verbunden. Eine Visualisierung dieses Falles ist in der Mitte von Abbildung 3.2 (b) dargestellt. Sowohl Knoten  $c$  als auch Knoten  $d$  sind mit  $a$  verbunden. Bei dieser Konstellation kann die Kante  $(c, d)$  gelöscht werden, da zwischen  $c$  und  $d$  eine alternative Verbindung über  $a$  existiert.
- Variante 3:  
Die Endknoten der einen Kante sind jeweils mit unterschiedlichen Endknoten der schneidenden Kante verbunden. In Abbildung 3.2 auf der rechten Seite existiert jeweils eine Verbindung zwischen  $a$  und  $c$  und eine zwischen  $b$  und  $d$ . Bei dieser Anordnung existiert ein Weg zwischen allen beteiligten Knoten unabhängig davon, welche der schneidenden Kanten gelöscht wird.



**Abb. 3.2:** Anhand der weiteren Verbindungen der Endknoten der schneidenden Kanten, ergeben sich verschiedene Varianten für die Kantenlöschung. Entweder darf keine Kante gelöscht werden (a) oder es ist festgelegt, welche Kante gelöscht werden darf (b) oder es darf eine beliebige der schneidenden Kanten gelöscht werden (c).

Für die dritte Variante wurden zwei verschiedene Verfahren zur Auswahl der zu löschenden Kante realisiert. Bei dem ersten Verfahren wird per Zufallsgenerator die Kante ausgewählt, die nicht im teilplanarisierten Subgraphen enthalten sein soll (Verfahren 1). Das zweite Verfahren bestimmt die Länge der schneidenden Kanten und entfernt die längere von beiden (Verfahren 2). Beide Verfahren erzeugen unterschiedliche teilplanarisierte Subgraphen, die dann jeweils mit den verschiedenen Routingvarianten getestet wurden. Abbildung 3.3 zeigt beispielhaft die unterschiedlichen teilplanarisierten Subgraphen. Auf der linken Seite ist der Originalgraph zu sehen. Der in der Mitte dargestellte Subgraph entstand durch die Anwendung von Verfahren 1. Wurde hingegen die längere Kante gelöscht, falls die zu löschende Kante frei gewählt werden konnte, ergab sich der teilplanarisierte Subgraph wie er auf der rechten Seite dargestellt ist.



**Abb. 3.3:** Auf der linken Seite ist der Originalgraph zu sehen. Der teilplanarisierte Subgraph in der Mitte entstand durch die Verwendung von Verfahren 1, der Subgraph auf der rechten Seite durch Anwendung von Verfahren 2.

### 3.1.3 Routingvarianten

Als Routingverfahren kommen die gängigen Varianten Greedy-Routing, Face-Routing und kombiniertes GFG-Routing zum Einsatz.

Greedy-Routing wählt unter seinen Nachbarn denjenigen als nächsten Hop, der die kürzeste euklidische Entfernung zum Ziel hat und schlägt fehl, wenn es ein lokales Minimum erreicht.

Face-Routing traversiert den Graphen entsprechend der Rechte-Hand-Regel. Der Wechsel in die nächste Face findet statt, bevor die st-Linie überschritten wird. Allerdings wird die Auswahl der nächsten Kante anhand der Rechte-Hand-Regel nicht unbedingt strikt bis zum Zielknoten beibehalten. Wenn ein Knoten feststellt, dass er direkt mit dem Zielknoten verbunden ist, wird auch die Kante zum Zielknoten als nächste gewählt, unabhängig davon, ob diese Wahl der Rechte-Hand-Regel entspricht oder nicht. Wird bei einem Routingversuch hingegen die erste Kante der aktuellen Face ein zweites Mal in gleicher Richtung durchlaufen, erkennt der Algorithmus, dass er sich in einer Schleife befindet. Der Routingversuch wird abgebrochen und als Fehlschlag gewertet.

Bei der kombinierten Fassung aus Greedy- und Face-Routing wird Face-Routing klassisch als Recovery-Strategie genutzt, wenn bei Greedy-Routing ein lokales Minimum erreicht wird. Sobald beim Face-Routing dann ein Knoten erreicht wird, der eine geringere Distanz zum Zielknoten hat als das lokale Minimum, wird wieder zu Greedy-Routing gewechselt.

## 3.2 Umsetzung

Die Simulationsumgebung wurde als Java-Programm umgesetzt. Dabei wurden die folgenden Bausteine realisiert:

- Zufällige Platzierung einer festgelegten Knotenmenge innerhalb einer Fläche,

deren Größe frei wählbar ist

- Generierung des Konnektivitätsgraphens auf Basis des Log-Normal-Shadowing-Modells
- Teilplanarisierung des Netzwerkgraphen unter Verwendung von Verfahren 1 und Verfahren 2
- Abwicklung eines Routingauftrags durch die drei Routingverfahren (Greedy-Routing, Face-Routing, kombiniertes Greedy-Face-Routing)

### 3.2.1 Settings

Die Herausforderung geographischer Routingalgorithmen sind gerade die dünnen Netzwerke. Ist der Netzwerkgraph recht dicht, ist die Wahrscheinlichkeit sehr groß, dass die Knoten beim Greedy-Routing einen geeigneten Weiterleitungsknoten finden. Je dünner der Graph ist, umso höher ist die Wahrscheinlichkeit, dass Greedy-Routing auf ein lokales Minimum trifft und Face-Routing als Recovery-Strategie zum Einsatz kommt. Die verbliebenen Schnitte in den teilplanarisierten Graphen beeinflussen lediglich die Routingabschnitte, in denen die Flächen mittels Face-Routing traversiert werden. Entsprechend sind für die Beantwortung der Frage, wie häufig Face-Routing in teilplanarisierten Graphen fehlschlägt, vor allem die Graphen interessant, die eine eher geringe Dichte aufweisen. Daher wurden für die Simulation bezüglich der Graphendichte drei verschiedene Varianten ausgewählt:

- sparse:  
Im Großen und Ganzen ist der Graph zusammenhängend, allerdings können vereinzelt Partitionen vorkommen.
- medium:  
Eine Unterteilung des Graphen ist möglichst selten, jedoch sind isolierte Knoten immer noch möglich.
- dense:  
Partitionen sollten nicht mehr vorkommen und isolierte Knoten sind selten.

Der zweite variable Parameter der Simulation ist die Standardabweichung  $\sigma$  der Normalverteilung. Hier sollten jeweils Szenarien ohne Schwankung, mit mittlerer und mit starker Schwankung untersucht werden. Daher wurden in Anlehnung an Tabelle 3.1 auf Seite 24 für  $\sigma$  die Werte 0.0, 5.0 und 10.0 untersucht.

Durch die Begutachtung der graphischen Darstellung von verschiedenen generierten Netzwerkgraphen, wurde die Feldgröße für eine feste Knotenanzahl so angepasst, dass für den jeweiligen  $\sigma$ -Wert die gewünschte Graphendichte erreicht wurde. Dabei wurde darauf geachtet, dass durch die gewählten Feldgrößen die Eigenschaften für sparse, medium und dense gerade so erreicht wurden. Für die neun verschiedenen Settings ergaben sich auf diese Weise die Belegungen für die Feldgröße wie in Tabelle 3.2 dargestellt.

$\sigma$	sparse	medium	dense
0.0	6250 x 6250	5000 x 5000	4400 x 4400
5.0	9750 x 9750	8400 x 8400	7500 x 7500
10.0	28000 x 28000	24000 x 24000	22000 x 22000

**Tabelle 3.2:** Feldgrößen der verschiedenen Settings

Abbildung 3.4 auf Seite 29 zeigt Beispieltopologien aller neun Settings. Die obere Reihe illustriert von links nach rechts den Graphen mit aufsteigender Knotendichte für  $\sigma = 0.0$ , die mittlere Reihe entsprechend für  $\sigma = 5.0$ , und die untere stellt die Graphendichten für  $\sigma = 10.0$  dar. Die teilplanarisierten Subgraphen dieser Beispieltopologien sind auf Seite 30 in Abbildung 3.5 zum Vergleich dargestellt.

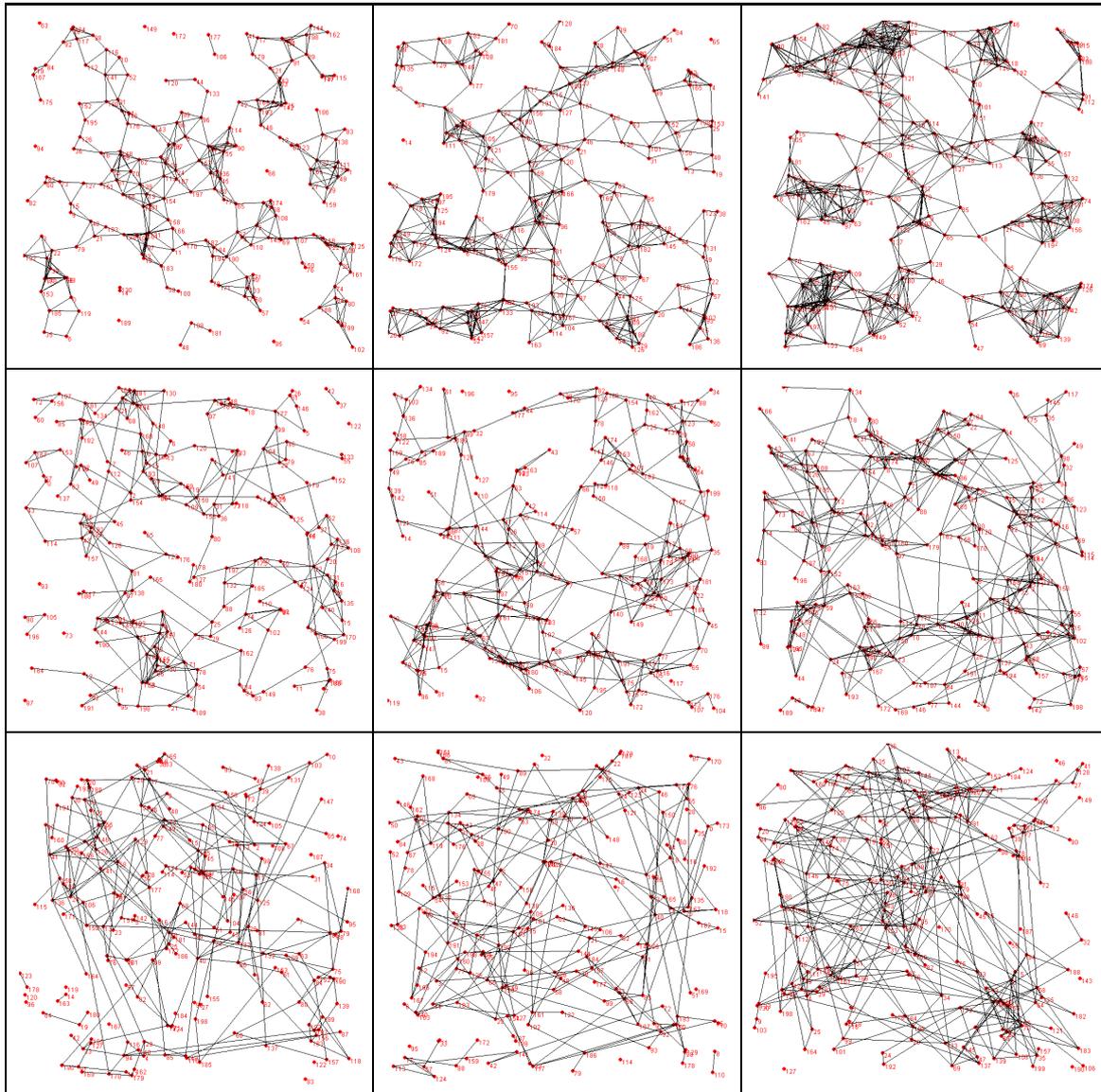
### 3.2.2 Simulationsablauf

Für jedes der neun Settings wurden jeweils 2000 Durchläufe simuliert. Feste Parameter dabei waren:

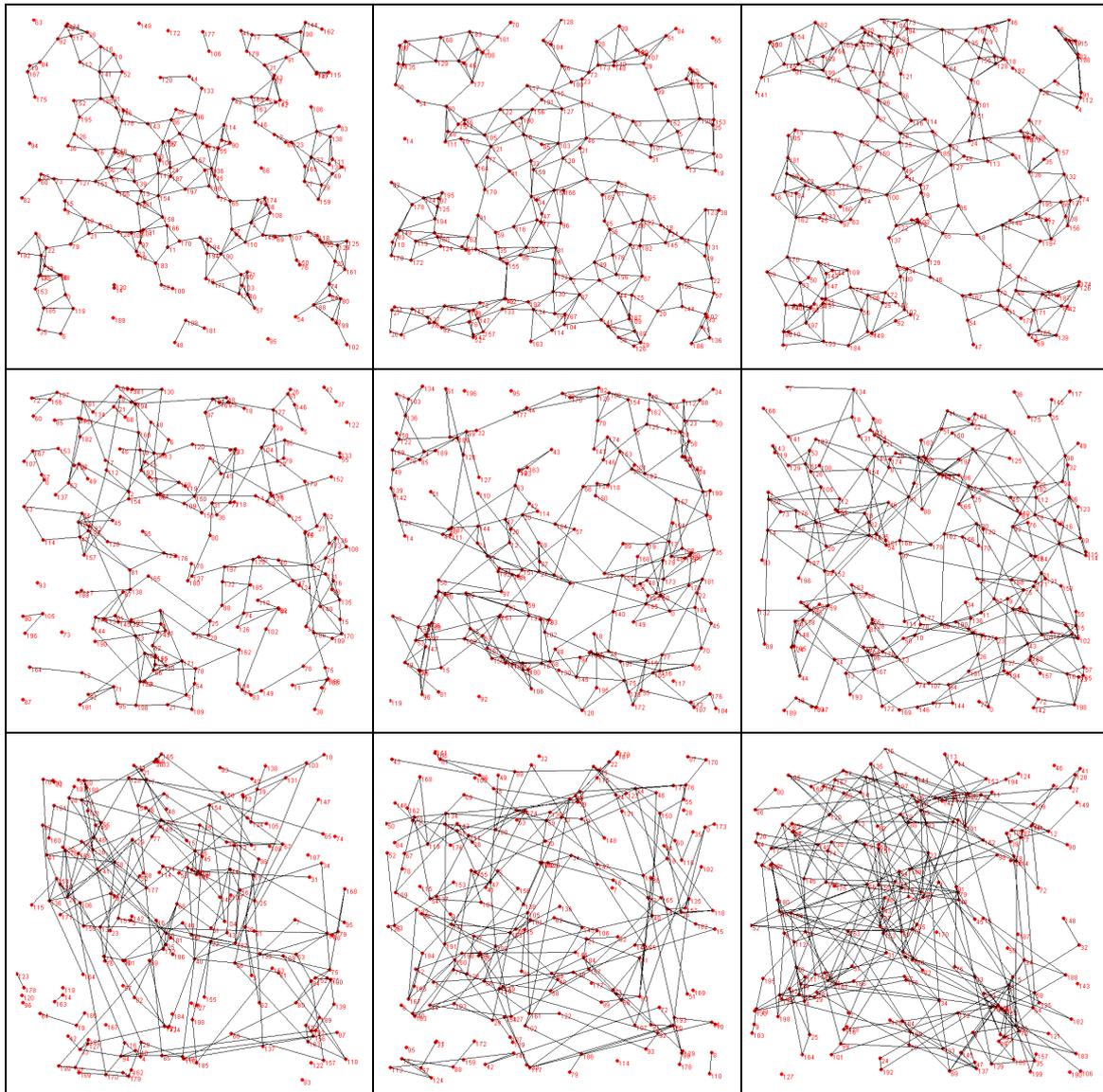
- Anzahl der Knoten: 200
- Pfadverlust bis zur Referenzdistanz  $d_0$ : 40 dB
- Pfadverlustkoeffizient  $\gamma$ : 2.0
- Referenzdistanz  $d_0$ : 1 Meter
- Empfangsschwellenwert  $c$ : -95 dBm

Für jeden Durchlauf wurden die 200 Knoten zufällig über die, dem Setting entsprechende, Feldgröße verteilt. Anschließend wurde für jedes Knotenpaar  $x$  und  $y$  der Pfadverlust  $\overline{\text{PL}}(x, y)$  berechnet und mit dem Empfangsschwellenwert  $c$  verglichen, um zu entscheiden, ob die Kante  $(x, y)$  Bestandteil des Konnektivitätsgraphen  $G$  ist oder nicht. Jede Kante wurde als bidirektional angenommen. Wenn Knoten  $x$  also mit Knoten  $y$  kommunizieren kann, impliziert das, dass auch Knoten  $y$  mit Knoten  $x$  verbunden ist. Durch die Anwendung der beiden Planarisierungsverfahren ergaben sich die teilplanarisierten Subgraphen  $S_1(G)$  und  $S_2(G)$ . Alle drei Graphenvarianten wurden in Form einer eigenen Adjazenzmatrix gepflegt.

Jeder Durchlauf sollte einen einzelnen Routingauftrag simulieren. Dazu wurde zufällig ein Start- und ein Endknoten ausgewählt. Mittels Breitensuche wurde geprüft, ob ein Pfad zwischen diesen beiden Knoten besteht. Falls diese Prüfung positiv ausfiel, wurde der Ablauf der drei verschiedenen Routingverfahren simuliert und im Erfolgsfall der durchlaufene Pfad zurückgegeben. Sowohl Face- als auch kombiniertes Greedy-Face-Routing wurden jeweils einmal unter Verwendung des Subgraphen  $S_1(G)$  und einmal für den Subgraphen  $S_2(G)$  simuliert. In jedem Simulationsdurchlauf berechnete das Programm verschiedene Parameter, die in Kapitel 3.2.3 aufgezählt werden. Die Ausgabe erfolgte in Form einer Excel-Tabelle, wobei jede Zeile



**Abb. 3.4:** Von links nach rechts sind in jeder Reihe Beispielgraphen für die Graphendichten sparse, medium und dense dargestellt. Die erste Reihe zeigt Graphen für einen  $\sigma$ -Wert von 0.0, die zweite für  $\sigma$  gleich 5.0, und für die letzte Reihe war ein  $\sigma$ -Wert von 10.0 gesetzt.

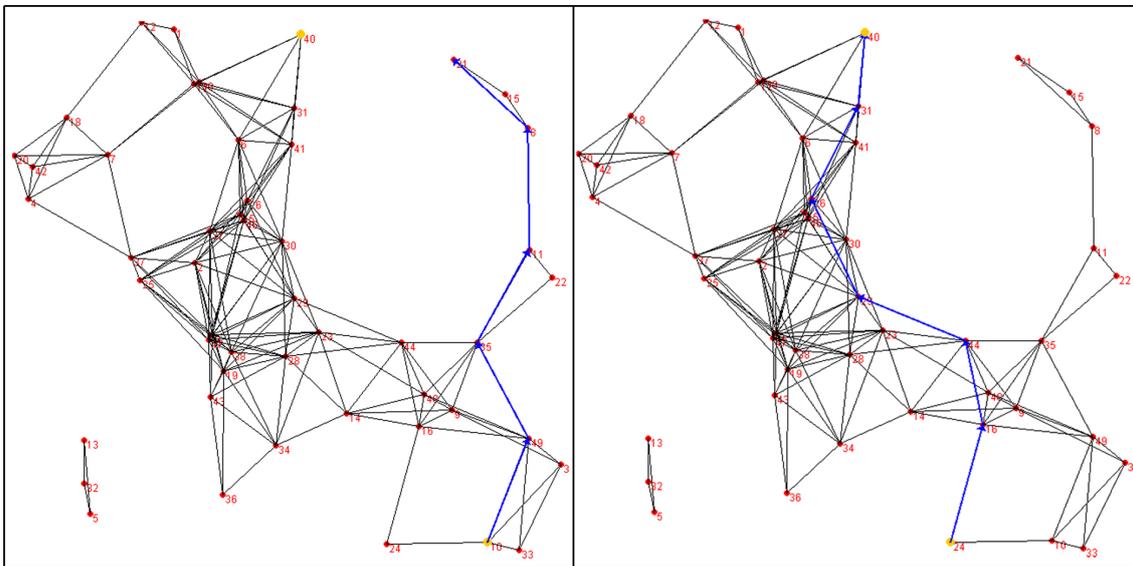


**Abb. 3.5:** Die teilplanarisierten Subgraphen der entsprechenden Originalgraphen aus Abbildung 3.4 sind durch Anwendung des Planarisierungsalgorithmus in der Variante 1 entstanden.

einen einzelnen Durchlauf widerspiegelte.

Zur besseren Nachvollziehbarkeit und zur bildlichen Darstellung wurde die Möglichkeit implementiert, die einzelnen Graphenvarianten und den zurückgelegten Pfad des jeweiligen Routingalgorithmus graphisch darzustellen.

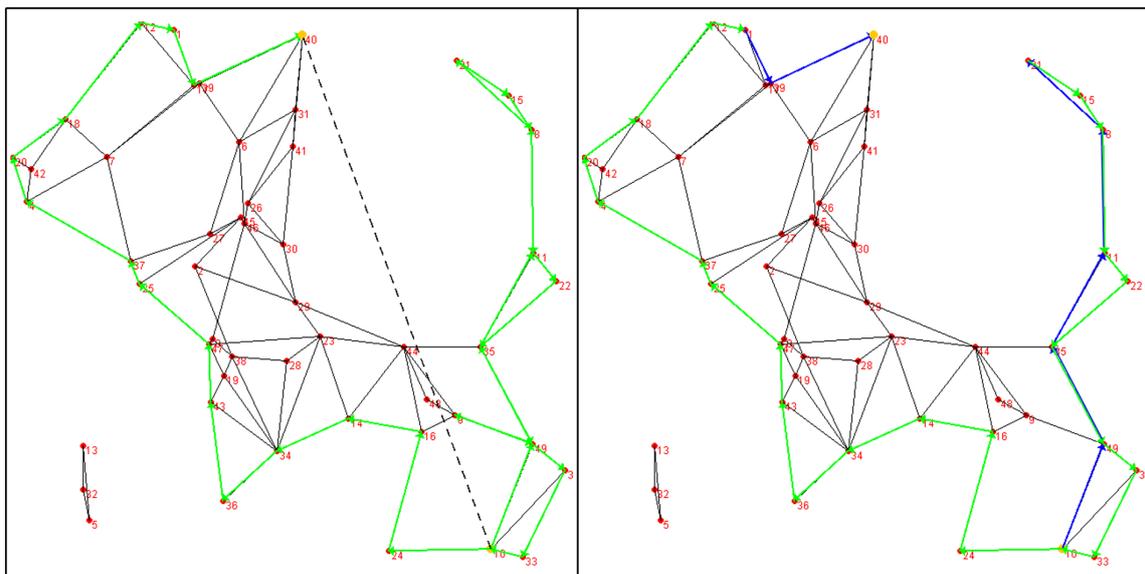
Abbildung 3.6 zeigt die Visualisierung von zwei verschiedenen Greedy-Routingaufträgen in derselben Netzwerktopologie. Links soll eine Nachricht von der Quelle 10 zum Zielknoten 40 gesendet werden. Knoten 10 erkennt Knoten 43 als denjenigen seiner Nachbarn, der den kürzesten Abstand zum Ziel hat, und leitet die Nachricht entsprechend dorthin weiter. Über die Knoten 35, 11 und 8 erreicht die Nachricht schließlich den Knoten 21. Dieser findet keinen Nachbarknoten, der eine kürzere Distanz zum Ziel hat als er selbst, und der Routingauftrag schlägt fehl, da ein lokales Minimum erreicht wurde. Der Routingauftrag in der rechten Darstellung hat ebenfalls Knoten 40 als Ziel. Diesmal ist Knoten 24 der Ursprung der Nachricht, ein direkter Nachbar von Knoten 10. Obwohl die Nachricht einen ähnlichen Ursprung hat wie im linken Beispiel, kann dieser Routingauftrag über die Knoten 16, 44, 29, 26 und 31 erfolgreich ausgeführt werden, so dass die Nachricht schließlich den Zielknoten 40 erreicht. Bestünde eine Verbindung zwischen Knoten 10 und Knoten 16, hätte auch der erste Routingauftrag zum Ziel geführt, da Knoten 16 näher am Ziel liegt als der Knoten 49 und folglich zu Beginn als nächster Hop ausgewählt worden wäre.



**Abb. 3.6:** Zwei unterschiedliche Greedy-Routingaufträge in einem Netzwerk. Auf der linken Seite möchte Knoten 10 eine Nachricht an Knoten 40 schicken. Unterwegs wird ein lokales Minimum erreicht und Greedy-Routing schlägt fehl. Auf der rechten Seite ist das Ziel dasselbe, aber Knoten 24 ist der sendende Knoten. Hier ist Greedy-Routing erfolgreich und erreicht den Zielknoten.

Abbildung 3.7 visualisiert einen Routingauftrag mit Startknoten 10 und Zielknoten 40. Auf der linken Seite ist die Wegwahl beim Face-Routing dargestellt. Die

gestrichelte Linie repräsentiert die virtuelle st-Linie. Hier wird zunächst auf Basis der Rechte-Hand-Regel Knoten 49 als nächster Hop gewählt. Anschließend führt der Weg über mehrere Zwischenknoten bis zu Knoten 21. An dieser Stelle gelangt der Algorithmus durch die Rechte-Hand-Regel an die "Außenbegrenzung" des Netzwerks und folgt diesen Netzwerkaußenkanten bis zum Zielknoten 40. Auf der rechten Seite ist der Routingpfad von kombinierten Greedy-Face-Routing aufgezeigt. Greedy-Routing wird auf dem Originalnetzwerkgraphen ausgeführt, Face-Routing auf dem teilplanarisierten Subgraphen. Zur besseren Übersicht wird hier lediglich der Subgraph des Netzwerks dargestellt. Greedy-Routingschritte sind mit blauen Pfeilen visualisiert, Face-Routingschritte mit grünen. Vom Startknoten 10 wird via Greedy-Routing zunächst der Weg über die Knoten 49, 25, 11 und 8 genommen, ehe bei Knoten 21 ein lokales Minimum erreicht wird. Daraufhin wird Face-Routing als Recovery-Strategie gestartet, und es wird entsprechend der Rechte-Hand-Regel der äußeren Begrenzung des Netzwerks gefolgt. Mit Knoten 1 wird schließlich ein Knoten erreicht, der eine geringere Distanz zum Ziel hat als Knoten 21, bei dem das lokale Minimum aufgetreten ist. Entsprechend wird wieder zu Greedy-Routing gewechselt und schließlich auch der Zielknoten 40 erreicht.



**Abb. 3.7:** Die Wegwahl für den Routingauftrag mit Startknoten 10 und Zielknoten 40 wird hier einmal für Face-Routing (links) und einmal bei kombiniertem Greedy-Face-Routing (rechts) dargestellt. Die grünen Pfeile entsprechen dabei Face-Routingschritten, die blauen visualisieren Greedy-Routingschritte.

### 3.2.3 Ausgaben der Simulationsdurchläufe

Für jeden Simulationsdurchlauf hat das erstellte Programm die folgenden Werte berechnet und ausgegeben:

## 1. Graphenbezogene Parameter

- durchschnittlicher Knotengrad (jeweils für den Originalgraphen und beide Varianten der teilplanarisierten Subgraphen)
- Anzahl der Kanten (jeweils für den Originalgraphen und beide Varianten der teilplanarisierten Subgraphen)
- Anzahl der Schnitte (jeweils für den Originalgraphen und beide Varianten der teilplanarisierten Subgraphen)

## 2. Daten zum Start- und Zielknoten

- Existiert ein Pfad vom Start- zum Zielknoten?
- minimale Hopanzahl des Pfades Start-Ziel
- euklidischer Abstand zwischen dem Start- und Zielknoten

3. Erfolg der einzelnen Routingverfahren<sup>2</sup>

- War Greedy-Routing erfolgreich?
- War Face-Routing erfolgreich?
- War GFG-Routing erfolgreich?

## 4. Greedy-Routing

- Anzahl der benötigten Hops bis zum Ziel
- euklidische Länge des Start-Ziel-Pfades
- Stretchfaktor (jeweils bezüglich der benötigten Hopanzahl und für die durchlaufene Pfadlänge)

## 5. Face-Routing

- Anzahl der benötigten Hops bis zum Ziel
- euklidische Länge des Start-Ziel-Pfades
- Anzahl der Face-Wechsel
- durchschnittliche Hops pro Face
- durchschnittlich zurückgelegte euklidische Länge pro Face
- Stretchfaktor (jeweils bezüglich der benötigten Hopanzahl und für die durchlaufene Pfadlänge)

## 6. kombiniertes Greedy-Face-Routing

- Anzahl der Umschaltungen zwischen Greedy- und Face-Routing

---

<sup>2</sup>Alle Werte, die Face- und kombiniertes Greedy-Face-Routing betreffen, wurden zweifach berechnet, da die Simulation sich hier jeweils einmal auf den nach Verfahren 1 teilplanarisierten Graphen bezog, und einmal auf den teilplanarisierten Subgraphen, der durch Verfahren 2 entstanden ist.

- Anzahl der Greedy-Anteile
- Anzahl der Face-Anteile
- Anzahl der benötigten Hops bis zum Ziel
- euklidische Länge des Start-Ziel-Pfades
- Stretchfaktor (jeweils bezüglich der benötigten Hopanzahl und für die durchlaufene Pfadlänge)
- Gesamtzahl der Hops in Greedy-Phasen
- Gesamtzahl der Hops in Face-Phasen
- durchschnittliche Hops pro Greedy- und pro Face-Phase
- zurückgelegte euklidische Länge in allen Greedy-Phasen
- zurückgelegte euklidische Länge in allen Face-Phasen
- durchschnittlich zurückgelegte euklidische Länge pro Greedy- und pro Face-Phase

# Kapitel 4

## Auswertung der Simulationsergebnisse

In diesem Kapitel werden die Ausgaben der Simulationsdurchläufe präsentiert und bewertet. Die Settings gemäß Tabelle 3.2 auf Seite 28 werden im Folgenden zur besseren Übersichtlichkeit durchnummeriert. Die Nummerierung mit den dazugehörigen Parametern ist in Tabelle 4.1 dargestellt.

Bezeichnung	$\sigma$ -Wert	Graphendichte
Setting 1	0.0	sparse
Setting 2	0.0	medium
Setting 3	0.0	dense
Setting 4	5.0	sparse
Setting 5	5.0	medium
Setting 6	5.0	dense
Setting 7	10.0	sparse
Setting 8	10.0	medium
Setting 9	10.0	dense

**Tabelle 4.1:** Settingbezeichnungen und Parameter

### 4.1 Planarisierungsverfahren und die entstandenen Subgraphen

#### 4.1.1 Kantenanzahl der verschiedenen Graphen

Abbildung 4.1 zeigt die durchschnittliche Kantenanzahl des Originalgraphen  $G$  sowie der beiden teilplanarisierten Subgraphen  $S_1(G)$  und  $S_2(G)$  in den verschiedenen Settings. Mit steigender Dichte enthalten die Graphen mehr Kanten. Es ist jedoch

erkennbar, dass sich diese Tendenz mit steigendem  $\sigma$ -Wert abschwächt. Enthält der dichte Graph unter Setting 3 noch die 1,95-fache Kantenmenge des dünnen Graphen aus Setting 1, ist bei einem  $\sigma$ -Wert von 10.0 nur noch die 1,53-fache Kantenmenge im dichten Graphen zu verzeichnen. Diese Beobachtung beruht auf der Tendenz, dass mit steigendem  $\sigma$ -Wert aufgrund der höheren Schwankungen auch längere Kanten ausgebildet werden können. Auf diese Art können auch in dünneren Graphen Knoten miteinander kommunizieren, die eine größere Distanz zueinander haben. Gleichzeitig erhöht sich die Wahrscheinlichkeit, dass Knoten mit geringer Distanz keine direkte Verbindung zueinander haben. Entsprechend beeinflusst die Graphendichte die Gesamtzahl der vorhandenen Kanten nicht so stark, wie es bei kleinen  $\sigma$ -Werten der Fall ist.

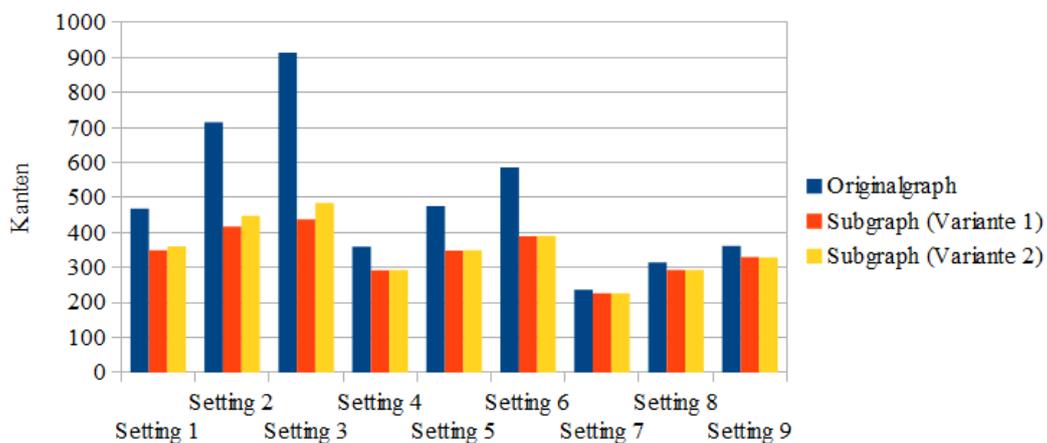


Abb. 4.1: Durchschnittliche Kantenanzahl

#### 4.1.2 Anzahl der Schnitte in den verschiedenen Graphen

Die durchschnittliche Gesamtanzahl an Schnitten in den verschiedenen Settings ist in Abbildung 4.2 dargestellt. Je dichter der Graph ist, umso mehr Schnitte sind zu finden. Allerdings schwächt sich dieses Verhältnis mit steigendem  $\sigma$ -Wert ab. Wird keine Schwankung mit in die Berechnung einbezogen ( $\sigma = 0$ ), weist der dichte Graph die 7,78-fache Menge an Schnitten im Gegensatz zum dünnen Graphen auf. Bei großer Schwankung ( $\sigma = 10.0$ ) ist es nur noch die 3,06-fache Menge. Auch dies ist mit der Tendenz zu längeren Kanten bei großen  $\sigma$ -Werten zu erklären. In den Settings 1 bis 3 können mit den lokalen Planarisierungsverfahren mehr als 99 % der Schnitte beseitigt werden. Bei einem mittleren Wert für die Standardabweichung  $\sigma$  werden durch die beiden Planarisierungsstrategien immerhin noch durchschnittlich 67,2 % der Schnitte eliminiert. Bei einem  $\sigma$ -Wert von 10.0 ergeben sich hingegen unregelmäßigere Graphenstrukturen. Aufgrund der sehr einfachen, lokalen Strategie der Planarisierungsverfahren 1 und 2 können in diesen "chaotischen" Graphen nur

noch durchschnittlich 13,2 % der Schnitte beseitigt werden.

Löscht man die längere Kante, sofern man die Wahl hat, welche der schneidenden Kanten gelöscht werden soll, führt dies zu einem etwas besseren Ergebnis als die zufällige Löschung. Mit Verfahren 2 konnten im Durchschnitt 2,1 % mehr Schnitte entfernt werden als mit Verfahren 1.

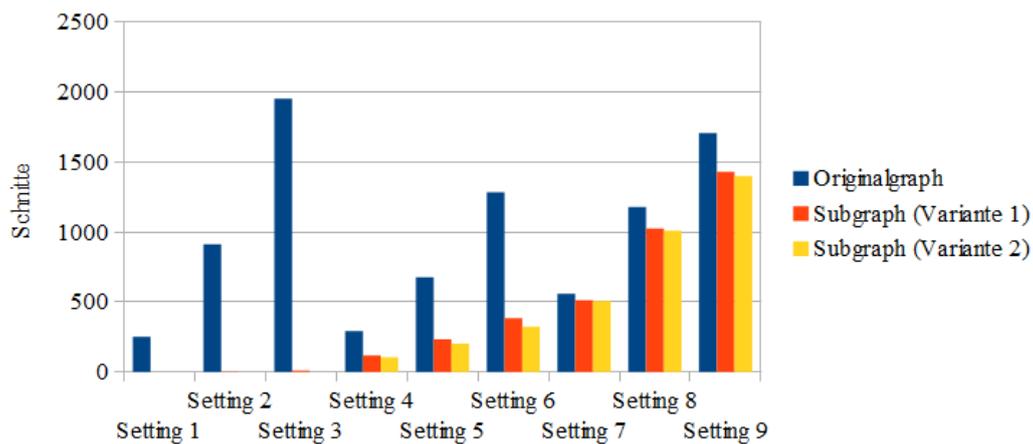


Abb. 4.2: Durchschnittliche Anzahl an Kantenschnitten

Zur besseren Veranschaulichung stellt Abbildung 4.3 dar, wie viele Schnitte durchschnittlich pro Kante in den verschiedenen Settings aufgetreten sind. Dabei wird deutlich, dass nicht nur dichtere Graphen zu mehr Kantenschnitten führen, sondern auch eine Vergrößerung des  $\sigma$ -Wertes bei ähnlicher Graphendichte. Die Tendenz, längere Kanten auszubilden, führt dazu, dass diese dann auch öfter von anderen Kanten geschnitten werden. Auch hier liefert das Planarisierungsverfahren 1 ein leicht besseres Resultat als das Verfahren 2.

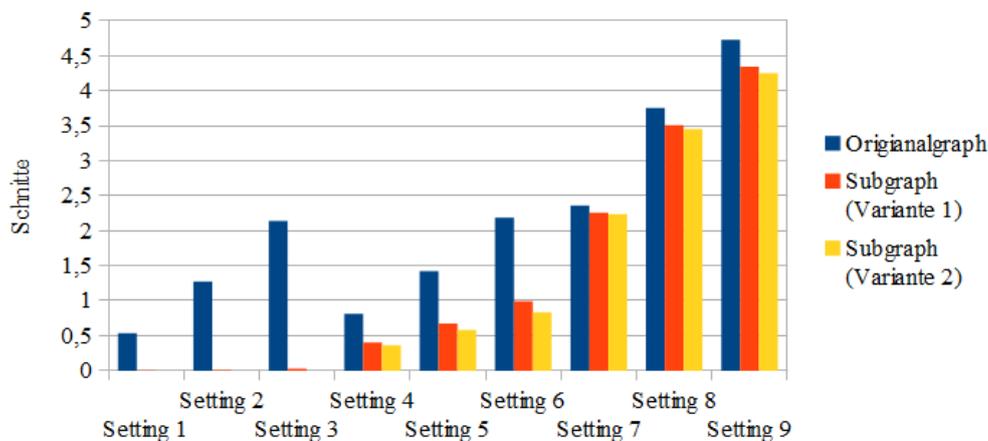


Abb. 4.3: Durchschnittliche Anzahl an Schnitten pro Kante

### 4.1.3 Knotengrad in den verschiedenen Graphen

Ein weiterer interessanter Messwert ist der durchschnittliche Knotengrad, also die durchschnittliche Menge an direkten Nachbarn eines Knotens, in den verschiedenen Settings. Abbildung 4.4 stellt den Knotengrad der verschiedenen Graphen in den unterschiedlichen Settings dar. Mit steigender Graphendichte steigt auch die durchschnittliche Anzahl an Nachbarn, mit denen jeder Knoten direkt kommunizieren kann. Gleichzeitig erhält man bei ähnlicher Graphendichte und steigendem  $\sigma$ -Wert einen geringeren Knotengrad. Im Schnitt benötigt jeder Knoten hier also weniger Nachbarn, um zu verhindern, dass der Graph unterteilt ist oder isolierte Knoten auftreten. Eine höhere Standardabweichung repräsentiert größere Schwankungen, wodurch auch Knoten, die einen größeren Abstand zueinander haben, mit einer höheren Wahrscheinlichkeit verbunden sind, als es bei geringen Schwankungen der Fall wäre. Werden viele Schnitte durch die Planarisierungsvarianten erkannt, wie es bei den Settings 1 bis 3 der Fall ist, führt Verfahren 2 dazu, dass es im Verhältnis mehr kurze Kanten gibt. Folglich ist bei diesen Settings der Knotengrad in  $S_2(G)$  ein wenig größer als in  $S_1(G)$ .

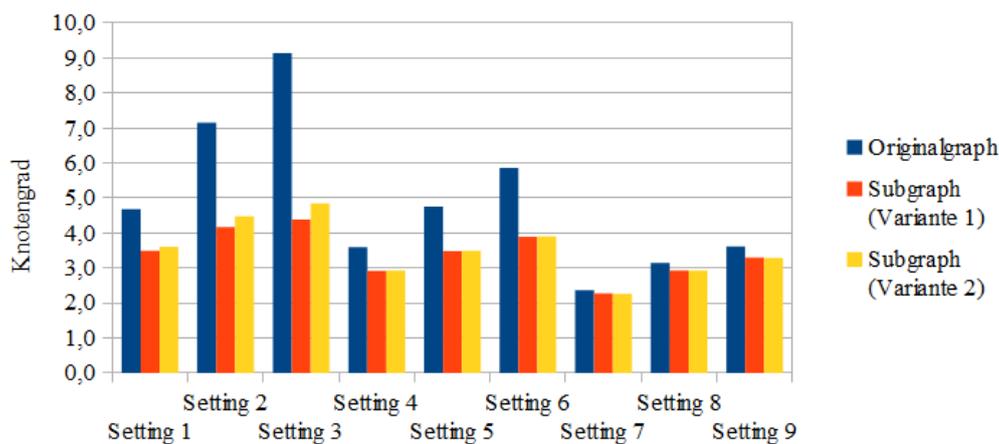


Abb. 4.4: Durchschnittliche Anzahl an direkt verbundenen Nachbarn eines Knotens

### 4.1.4 Bewertung der Planarisierungsverfahren

Es hat sich gezeigt, dass die beiden sehr einfachen, verwendeten Planarisierungsverfahren recht zuverlässig funktionieren, wenn  $\sigma$  einen Wert von Null hat, und über 99 % der Schnitte entfernen. Dieses Szenario entspricht allerdings der idealisierten Annahme, dass die Kommunikationsfähigkeit zweier Knoten lediglich von der Distanz abhängig ist. Zieht man nun die Varianz aufgrund von äußeren Umständen (Hindernisse, Reflexionspfade, Mehrwegeausbreitung) mit in Betracht, schneiden die beiden Planarisierungsstrategien deutlich schlechter ab. Ungeachtet dessen sollte im

Rahmen dieser Arbeit untersucht werden, wie sich geographische Routingverfahren in genau solchen teilplanarisierten Graphen verhalten, die noch Kantenschnitte enthalten. Jedoch könnte die Anzahl der zurückbleibenden Schnitte mit einer immer noch sehr einfachen Methode, die nicht nur die Endpunkte der schneidenden Kanten betrachtet, weiter reduziert werden. Eine mögliche Variante wäre es, zu untersuchen, ob es zwischen zwei dieser Endpunkte einen alternativen  $k$ -Hop Weg gibt, wobei  $k$  lediglich kleine Werte umfassen sollte.

## 4.2 Performance der verschiedenen Routingverfahren

### 4.2.1 Erfolgsrate der verschiedenen Routingverfahren

In Abbildung 4.5 sind die Erfolgsraten der unterschiedlichen Routingverfahren in den verschiedenen Settings dargestellt.

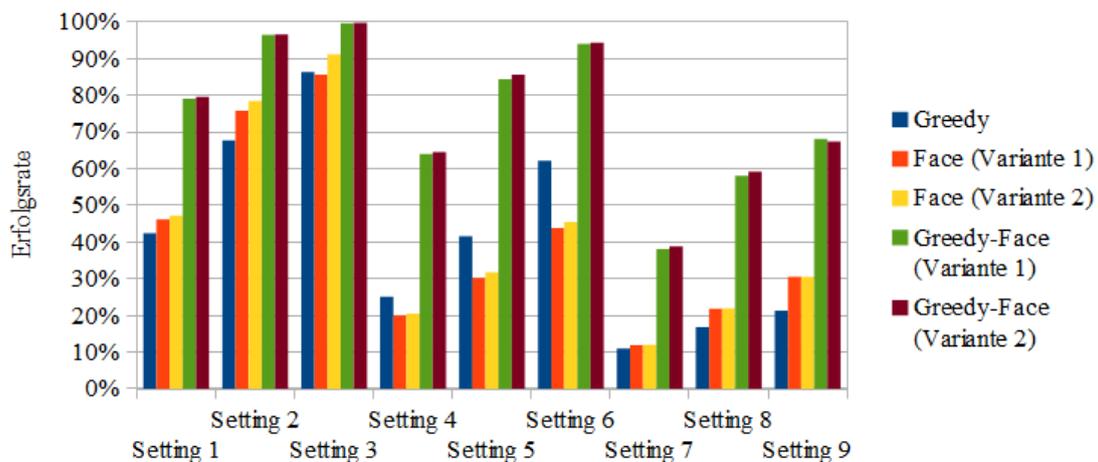


Abb. 4.5: Erfolgsrate der verschiedenen Routingverfahren

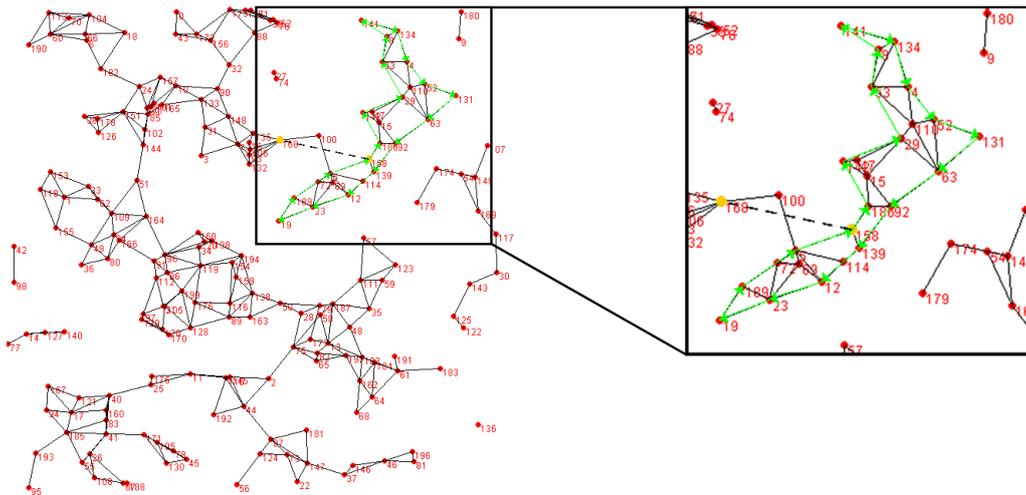
Bei kombiniertem Greedy-Face-Routing konnte in allen Settings die höchste Auslieferungsrates festgestellt werden. Dabei ist die Wahrscheinlichkeit eines erfolgreich abgeschlossenen Routingauftrags in den dichten Graphen höher als in den dünnen. Greedy-Routing schneidet am besten in den Settings ab, in denen die Verbindungen innerhalb der Konnektivitätsgraphen lediglich auf der Knotendistanz beruhen und die Standardabweichung der Normalverteilung den Wert Null hat. Die Wahrscheinlichkeit für einen erfolgreichen Routingauftrag ist in dichten Graphen im Gegensatz zu dünnen mehr als doppelt so hoch. Diese Tendenz lässt sich für alle  $\sigma$ -Werte beobachten. Mit steigendem  $\sigma$ -Wert sinkt die Erfolgsrate beim Greedy-Routing allerdings im Allgemeinen drastisch, vor allem in dünnen Graphen. So erreicht Greedy-

Routing in Setting 7 nur noch eine Erfolgsrate von 10,97 %, also nur noch gut ein Viertel der Erfolgsrate in Setting 1 (42,39 %). Reines Greedy-Routing erlangt in realistischen Settings folglich deutlich häufiger ein lokales Minimum und schlägt infolgedessen fehl. Je höher der  $\sigma$ -Wert ist, umso größer ist das Verhältnis zwischen kurzen und langen Kanten. Dieses Ungleichgewicht führt dazu, dass seltener ein geeigneter nächster Hop gefunden wird.

Überraschend war die niedrige Erfolgsquote von Face-Routing, die in den Settings 4 bis 6 sogar noch schlechter war, als die Erfolgsrate von Greedy-Routing. Anhand der Simulation sollte zwar untersucht werden, inwiefern verbleibende Schnitte den Erfolg von Face-Routing, und damit auch von kombiniertem Greedy-Face-Routing, beeinflussen, jedoch war diese sehr niedrige Erfolgsrate, vor allem in den Settings 4 bis 9, unerwartet. In den Subgraphen der Settings 1 bis 3 wurden zudem über 99 % der Schnitte erkannt und entfernt. Daher konnte die relativ hohe Anzahl an Face-Routingfehlschlägen nicht mit dem Vorhandensein von Kantenschnitten erklärt werden. Aus diesem Grund wurden die einzelnen Ergebnisse der Simulationen von Setting 1 noch einmal genau analysiert. Das Resultat dieser Begutachtung war, dass in den nach Verfahren 1 planarisierten Graphen bei 38,7 % der Face-Routingfehlschläge noch Schnitte im Graphen vorhanden waren. Entsprechend geschahen 61,3 % der Routingfehler in einem planaren Graphen. Bei der Planarisierung nach Verfahren 2 enthielten sogar 99,4 % der Graphen, in denen es zu Fehlschlägen beim Face-Routing kam, keine Schnitte. Diese Ergebnisse stehen jedoch im Gegensatz zu der Grundthese, dass Face-Routing die Auslieferung garantieren kann, sofern ein planarer Graph vorliegt. Bei weiteren Simulationen konnte der Widerspruch reproduziert und ein Szenario festgestellt werden, bei dem Face-Routing in der implementierten Variante trotz eines planaren Graphen fehlschlägt. Falls es eine Topologie erforderlich macht, dass eine Kante  $(u, v)$  durchlaufen wird, die von der virtuellen  $st$ -Linie geschnitten wird, um den Zielknoten  $t$  zu erreichen, stellt der Algorithmus zunächst den Schnitt mit der  $st$ -Linie fest. Daraufhin wird der Schnittpunkt zwischen der  $st$ -Linie und der Kante  $(u, v)$  als neuer Startpunkt der  $st$ -Linie berechnet und die nächste Kante entsprechend der Rechte-Hand-Regel ausgewählt. Irgendwann wird schließlich wieder der Knoten  $u$  erreicht, und der Algorithmus versucht die Kante  $(u, v)$  zu durchlaufen. Dabei wird ein Schnitt mit der  $st$ -Linie detektiert und infolgedessen gerät Face-Routing in eine Schleife und bricht ab. Abbildung 4.6 illustriert einen solchen Fall. Startknoten 158 möchte eine Nachricht an Zielknoten 168 senden. Dazu müsste die Kante  $(5, 100)$  durchlaufen werden, die von der  $st$ -Linie geschnitten wird. Das Verfahren erkennt schließlich eine Schleife und bricht ab.

Der zeitliche Rahmen dieser Arbeit ließ es leider nicht zu, eine Face-Routingvariante zu implementieren, die lediglich aufgrund von Kantenschnitten fehlschlägt und die Simulationsläufe für alle neun Settings erneut zu starten. Auch wenn das beschriebene Szenario in dichteren Graphen seltener vorkommt als in dünnen, da die Wahrscheinlichkeit für das Vorhandensein eines Alternativweges größer ist, lässt sich nicht mit Gewissheit sagen, dass die Routingfehlschläge in dichten Graphen aufgrund von Kantenschnitten geschehen. Die Tatsache, dass Face-Routing allerdings in den

Graphen, die im Durchschnitt besonders viele Schnitte pro Kante aufweisen, besonders häufig fehlschlägt, lässt die Schlussfolgerung zu, dass sich Kantenschnitte relativ negativ auf die Erfolgsrate von Face-Routing auswirken. Um genauere Aussagen darüber zu treffen, welche durchschnittliche Schnittanzahl pro Kante noch zu annehmbaren Erfolgsquoten führt, müsste das Verhalten des Routingverfahrens auf Basis einer Face-Routingvariante untersucht werden, die lediglich aufgrund von Schnitten im teilplanarisierten Graphen fehlschlagen kann.



**Abb. 4.6:** Beispieltopologie für ein Szenario, das bei der implementierten Face-Routingvariante zu einem Abbruch aufgrund einer Schleifenerkennung führt

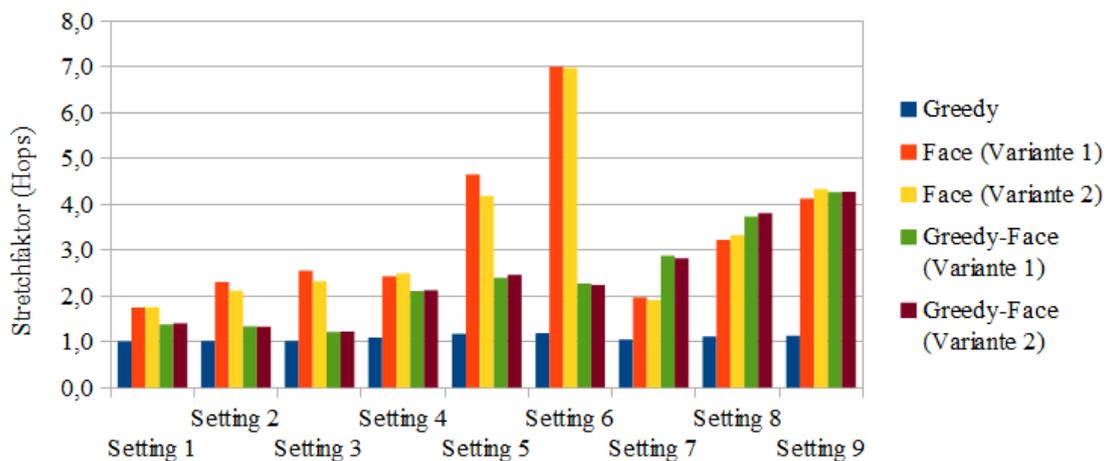
Aufgrund des aufgeführten Abbruchverhaltens des Face-Routingalgorithmus, ergibt sich auch für kombiniertes Greedy-Face-Routing keine eindeutige Aussage bezüglich der Misserfolgsquote aufgrund von Kantenschnitten. Zwar ist von allen drei untersuchten Routingverfahren das kombinierte Verfahren das erfolgreichste, jedoch sind auch hier mit hoher Wahrscheinlichkeit Fehlschläge beobachtet worden, die auf dem Abbruchverhalten in dem beschriebenen Szenario bei Face-Routing beruhen. Die Tatsache, dass kombiniertes Greedy-Face-Routing vor allem in den realistischen Settings deutlich besser abschneidet als reines Greedy- oder Face-Routing, legt die Vermutung nahe, dass mit einem Face-Routingalgorithmus, der dieses Abbruchverhalten nicht zeigt, und einer etwas erweiterten Planarisierungsstrategie unter Umständen annehmbare Erfolgsquoten zu erzielen wären. Es lässt sich jedenfalls nicht ausschließen, dass eine Teilplanarisierung des Netzwerkgraphen ausreicht, um unter Anwendung von kombiniertem Greedy-Face-Routing eine gute bis sehr gute Auslieferungsrate zu erreichen.

#### 4.2.2 Stretchfaktor der verschiedenen Routingverfahren

Ein gutes Maß für die Effizienz von Routingverfahren ist der Stretchfaktor, also das Verhältnis vom gefundenen Pfad zum kürzesten Pfad. Der Stretchfaktor

wurde in zwei verschiedenen Varianten als Messgröße verwendet. Zum einen wurde das Verhältnis von tatsächlich vollzogenen Hops zu der minimalen Hopanzahl zwischen Start- und Zielknoten ermittelt. Die minimale Hopdistanz wurde mittels Breitensuche ermittelt. Auf der anderen Seite wurde der Stretchfaktor im Bezug auf die zurückgelegte euklidische Länge ermittelt. Als Referenz für die kürzeste Distanz diente hierbei der euklidische Abstand zwischen Start- und Zielknoten. Berücksichtigt wurden nur erfolgreich abgeschlossene Routingaufträge.

Der Hop-Stretchfaktor ist im Diagramm in Abbildung 4.7 dargestellt. Bei Greedy-Routing konnte mit einem Durchschnittswert von 1,09 für alle neun Settings der geringste Stretchfaktor beobachtet werden. In den realitätsnahen Settings 4 bis 9 war der Stretchfaktor im Schnitt ein wenig höher als in den Settings 1 bis 3, bei denen die Verbindung zwischen zwei Knoten lediglich von der Distanz abhängig war.



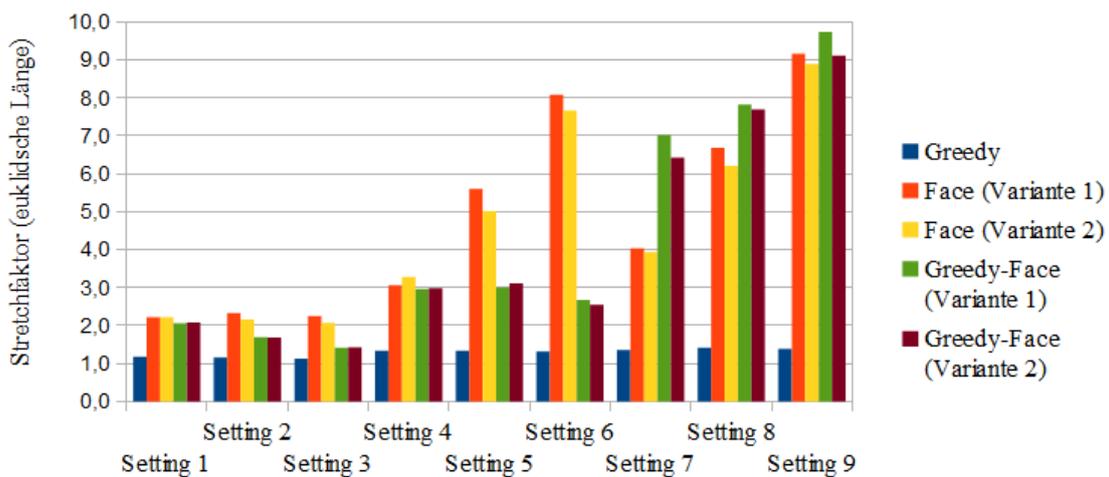
**Abb. 4.7:** Stretchfaktor im Bezug auf die benötigten Hops bis zum Ziel

Der Stretchfaktor ist bei reinem Face-Routing hoch, zumindest in den Settings 1 bis 6. In den Settings 7 bis 9 liegt er hingegen meist unter dem Stretchfaktor von kombiniertem Greedy-Face-Routing. Die Ursache dafür könnte in dem Abbruchverhalten der Face-Routingimplementierung liegen. In diesen Settings schlägt Face-Routing sehr häufig fehl, und es ist nicht zu sagen, ob Schnitte der Grund dafür sind oder das Szenario, in dem eine Kante durchlaufen werden müsste, die von der st-Linie geschnitten wird. Gleichzeitig deutet die Tatsache, dass gerade diese Settings sehr viele Schnitte enthalten, darauf hin, dass die verbliebenen Kantenschnitte die Hauptursache für die Routingfehlschläge sind. Entsprechend könnte es sein, dass Face-Routing nur dann erfolgreich war, wenn nicht allzu viele Hops benötigt wurden, um das Ziel zu erreichen, denn in diesen Fällen ergaben sich weniger Möglichkeiten, aufgrund von Kantenschnitten Fehlschläge zu produzieren. Daher kann dies als Indiz dafür gewertet werden, dass reines Face-Routing in teilplanarisierten Graphen mit steigender Hop-Distanz häufiger versagt. Der Stretchfaktor war im Bezug auf die beiden

verwendeten Planarisierungsmethoden fast identisch. In den nach Verfahren 1 teilplanarisierten Subgraphen betrug der Stretchfaktor im Durchschnitt 3,33, in den durch Verfahren 2 entstandenen Subgraphen 3,27.

Bei dem kombinierten Verfahren liegt der Stretchfaktor der Settings 1 bis 6 in einem moderaten Bereich zwischen dem von Greedy- und dem von Face-Routing. Lediglich in den Settings 7 bis 9 liegt er im Schnitt sogar höher als der bei Face-Routing beobachtete Wert. Auch dies lässt sich mit der These erklären, dass Face-Routing bei vielen Hops eher fehlschlug. Im kombinierten Verfahren wird durch den Rückwechsel auf Greedy-Routing mit höherer Wahrscheinlichkeit das Ziel erreicht. Entsprechend ist kombiniertes Greedy-Face-Routing auch bei höheren Hopzahlen tendenziell erfolgreicher, wodurch sich ein größerer Stretchfaktor ergibt.

Nimmt man die zurückgelegte euklidische Distanz als Grundlage, ergibt sich für den Stretchfaktor ein ähnliches Bild. Das Verhalten der verschiedenen Routingvarianten in den unterschiedlichen Settings ist in Abbildung 4.8 illustriert.



**Abb. 4.8:** Stretchfaktor in Bezug auf die euklidische Länge des zurückgelegten Pfades

Das Verhältnis zwischen den verschiedenen Routingverfahren stellt sich analog zu dem, auf der Hopanzahl basierenden, Stretchfaktor dar. Es ist lediglich der Unterschied zu erkennen, dass das kombinierte Verfahren in den Settings 7 bis 9 einen sehr hohen Stretchfaktor im Vergleich zu dem Verhältnis gegenüber Face-Routing in den Settings 4 bis 6 aufweist. Diese Tatsache begründet sich damit, dass die Feldgröße bei den Settings mit einem  $\sigma$ -Wert von 10.0 sehr groß ist. Da als Grundlage für die kürzeste euklidische Distanz der Abstand von Start- und Zielknoten gewählt wurde, ergeben sich in diesen Settings mehr Möglichkeiten, lange Pfade zurückzulegen, als bei den Settings mit geringerer Feldgröße. Gleichzeitig versagte Face-Routing hier deutlich häufiger, sobald es eine längere Distanz zu durchlaufen hatte.

### 4.2.3 Bewertung der Routingperformance

Greedy-Routing hat in der Simulation die bekannte Schwäche vor allem in dünnen Graphen gezeigt. Je größer die Standardabweichung der normalverteilten Zufallsvariablen  $X_\sigma$  war, umso schlechter war auch die Erfolgsquote von Greedy-Routing. Wenn das Ziel allerdings erreicht wurde, war die Wegwahl recht effizient und verursachte nur einen geringen Stretchfaktor.

Die Auswertung der Ergebnisse von Face-Routing gestaltet sich aufgrund der beschriebenen Face-Routingimplementierung schwierig. Für die beobachtete große Anzahl an Routingfehlschlägen kann nicht gesagt werden, ob diese aufgrund von Kantenschnitten oder durch das Abbruchverhalten bei dem beschriebenen Szenario begründet sind. Allerdings schlägt Face-Routing vor allem in den Settings fehl, in denen es durchschnittlich die meisten Schnitte pro Kante gab. In diesen Settings ist eine signifikant geringere Erfolgsrate erzielt worden als in den Settings, in denen fast keine Schnitte in den teilplanarisierten Subgraphen übrig blieben. Diese Tatsache stützt die These, dass die Routingfehlschläge mit hoher Wahrscheinlichkeit eher auf Kantenschnitte zurückzuführen sind.

Die Tatsache, dass die Fehlschläge beim Face-Routing nicht eindeutig einer bestimmten Ursache zuzuordnen sind, beeinflusst die Auswertung der Ergebnisse bei kombiniertem Greedy-Face-Routing. Allerdings schneidet diese Variante gerade in den realistischeren Settings deutlich besser ab als reines Greedy- oder Face-Routing. Diese Beobachtung lässt den Schluss zu, dass mit einer veränderten Face-Routingvariante und einer leicht erweiterten Planarisierungsstrategie deutlich bessere Erfolgsquoten erzielt werden könnten, so dass eine Teilplanarisierung für den praktischen Einsatz zumindest in Betracht kommen könnte. Der höhere Stretchfaktor in den Settings 7 bis 9 bei kombiniertem Greedy-Face-Routing wirkt zunächst negativ, kann aber durch die deutlich höhere Erfolgsrate relativiert werden. Bei längeren Pfaden ist es wahrscheinlicher, dass Greedy- oder Face-Routing versagen, während mit dem kombinierten Verfahren dennoch ein Weg zum Ziel gefunden wird.

## 4.3 Face-Routing

### 4.3.1 Facewechsel

Das Diagramm in Abbildung 4.9 zeigt die durchschnittlichen Facewechsel pro Routingauftrag. Im Vergleich wurden in dichteren Graphen mehr Facewechsel vollzogen als in dünneren, was die logische Konsequenz der höheren Kantenzahl in diesen Settings ist. Mit steigendem  $\sigma$ -Wert fällt die Anzahl der Facewechsel allerdings drastisch ab. Diese Beobachtung deckt sich mit der Vermutung, dass Face-Routing in diesen Settings nur dann erfolgreich war, wenn lediglich eine kurze Distanz zurückgelegt werden musste. Vor allem in den Settings 7 bis 9, in denen der  $\sigma$ -Wert eine starke Schwankung repräsentiert, wurden bei den erfolgreichen Face-Routingversuchen im Schnitt weniger als zwei Facewechsel vollzogen. Bei Routingaufträgen, die einen längeren Pfad erforderlich machen, kam es also offensichtlich zu Fehlschlägen. Die-

se Misserfolge sind mit hoher Wahrscheinlichkeit auf verbliebene Kantenschnitte zurückzuführen. Im teilplanarisierten Subgraphen  $S_2(G)$  zeigten sich im Schnitt 5 % mehr Facewechsel als im Subgraphen  $S_1(G)$ , der durch Verfahren 1 entstanden ist. Dies beruht auf der Tendenz von Verfahren 2, tendenziell eher Graphen zu erzeugen, die mehr Kanten enthalten, die dafür allerdings kürzer sind.

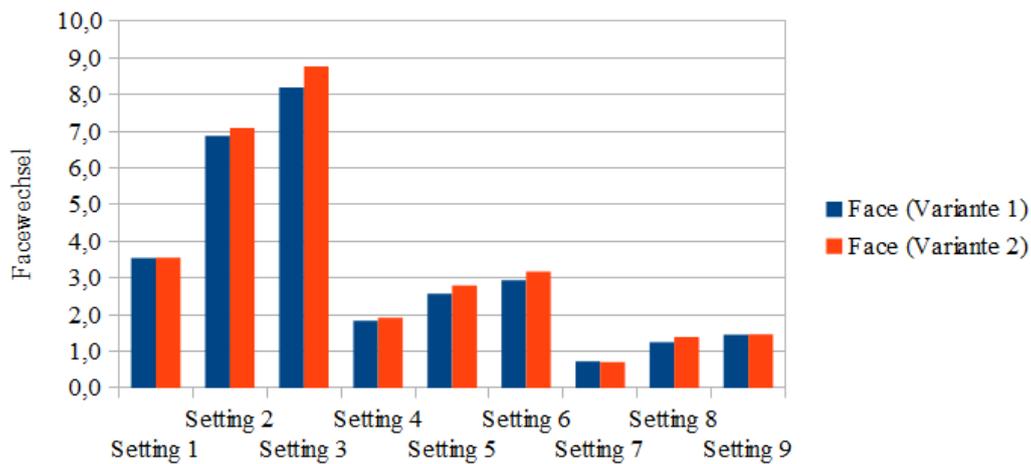


Abb. 4.9: Durchschnittliche Facewechsel pro Routingauftrag

### 4.3.2 Hops und euklidische Distanz pro Face

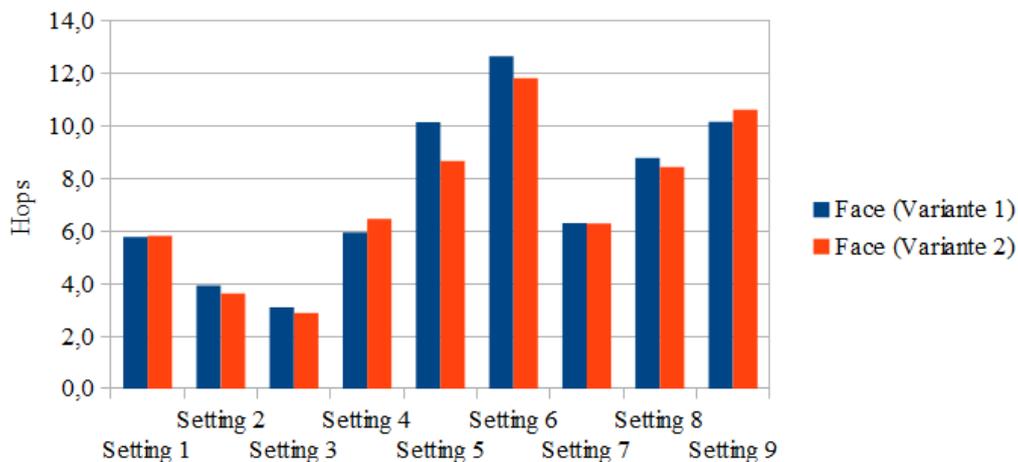


Abb. 4.10: Durchschnittliche Hopanzahl pro Face

Abbildung 4.10 zeigt die durchschnittliche Hopanzahl pro Face. In den Settings 1 bis 3 kam es mit steigender Graphendichte auch zu mehr Facewechseln und weniger Hops

pro Face. In den realitätsnahen Settings 4 bis 9 stieg hingegen die durchschnittliche Hopanzahl pro Face mit der Graphendichte. Das gleiche Verhältnis zeigt sich in der durchschnittlich pro Face zurückgelegten euklidischen Distanz, wie es in Abbildung 4.11 dargestellt ist. Die große Diskrepanz zwischen den Settings 1 bis 3, 4 bis 6 und 7 bis 9 ergibt sich aus der unterschiedlich großen Feldfläche bei diesen drei Varianten.

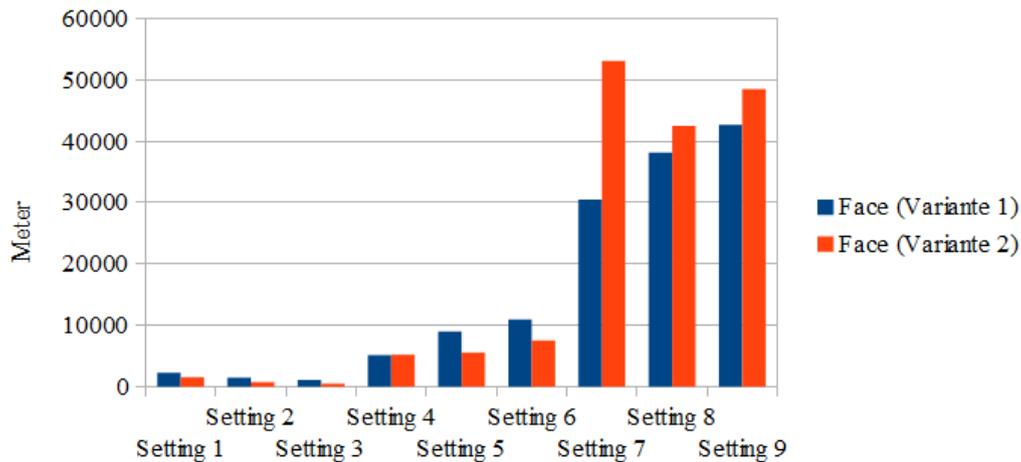


Abb. 4.11: Durchschnittliche euklidische Distanz pro Face

### 4.3.3 Bewertung des Face-Wechselverhaltens

Es konnten in dichteren Graphen mehr Facewechsel beobachtet werden als in dünneren, aber mit steigendem  $\sigma$ -Wert fiel die Anzahl drastisch ab. Der Grund dafür ist vermutlich, dass Face-Routing in den realistischeren Settings wahrscheinlich in den Fällen zum Erfolg geführt hat, in denen ein kürzerer Pfad zum Ziel führte.

## 4.4 Kombiniertes Greedy-Face-Routing

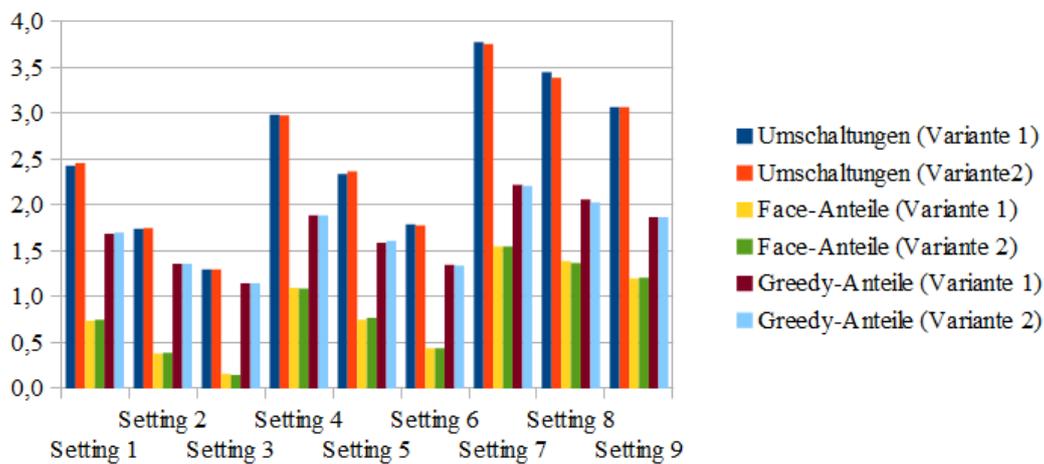
### 4.4.1 Umschaltungen

Abbildung 4.12 zeigt wie oft bei kombiniertem Greedy-Face-Routing durchschnittlich zwischen Greedy- und Face-Routing gewechselt wurde und wie viele Face- bzw. Greedy-Anteile im Schnitt zu verzeichnen waren.

Es zeigt sich, dass es in dichteren Graphen weniger Umschaltungen zwischen Greedy- und Face-Routing gibt als in dünnen Graphen. Bei den dichten Graphen der Settings 3, 6 und 9 ist der Knotengrad im Schnitt knapp 1,75-fach so hoch wie bei den dünnen Graphen der Settings 1, 4 und 7. Entsprechend ist die Wahrscheinlichkeit, dass Greedy-Routing ein lokales Minimum erreicht und Face-Routing als Recovery-Strategie genutzt wird, bei dünnen Graphen höher, da jeder Knoten im Durchschnitt

weniger Nachbarn hat. Gleichzeitig sind mit steigendem  $\sigma$ -Wert mehr Umschaltungen zu beobachten. In diesen Settings nimmt die Erfolgsrate von reinem Greedy-Routing stark ab. Infolgedessen muss häufiger auf Face-Routing zurückgegriffen werden, um ein lokales Minimum zu überwinden. Das führt dazu, dass auch die Gesamtanzahl an Wechslen zwischen Greedy- und Face-Routing steigt.

In jedem Setting gab es mehr Greedy- als Face-Anteile. Dieses Ergebnis begründet sich damit, dass Face-Routing lediglich eine temporäre Recovery-Strategie für den Fall darstellt, dass Greedy-Routing an einem lokalen Minimum keinen geeigneten Next-Hop findet.



**Abb. 4.12:** Durchschnittliche Wechsel zwischen Face- und Greedy-Routing

Ob der teilplanarisierte Subgraph durch die Planarisierungsvariante 1 oder 2 erstellt wurde, hat auf die Anzahl der Wechsel zwischen Greedy- und Face-Routing keinen nennenswerten Einfluss.

#### 4.4.2 Hops und euklidische Distanz pro Abschnitt

Das Diagramm in Abbildung 4.13 visualisiert die durchschnittliche Hopanzahl pro Greedy- bzw. Face-Abschnitt. Es zeigt sich, dass die Anzahl an Greedy-Hops mit steigendem  $\sigma$ -Wert fällt. Repräsentiert die normalverteilte Zufallsvariable  $X_\sigma$  also höhere Schwankungen, erreicht Greedy-Routing schneller ein lokales Minimum, und es muss auf Face-Routing als Recovery-Strategie zurückgegriffen werden. Im Verhältnis dazu steigt die Anzahl der Hops in den Face-Routingabschnitten bei höheren  $\sigma$ -Werten. Ein  $\sigma$ -Wert von 10.0 erzeugt "chaotisch" wirkendere Graphen als ein  $\sigma$ -Wert von 0.0. Folglich bewegt sich Face-Routing bei der Graphen-Traversierung weniger kontinuierlich auf das Ziel zu und findet erst später einen Knoten, der näher am Zielknoten gelegen ist als der Knoten, bei dem auf Face-Routing gewechselt wurde.

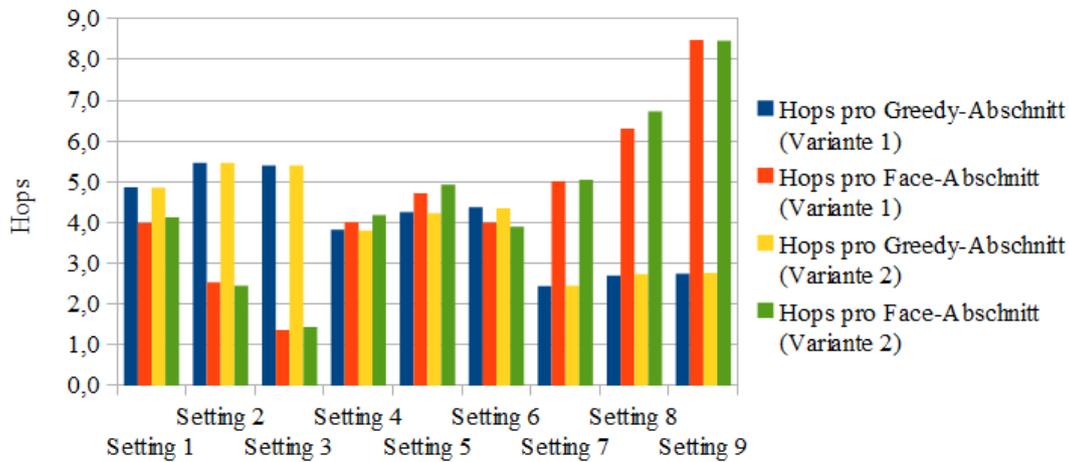


Abb. 4.13: Durchschnittliche Hopanzahl pro Greedy- bzw. Face-Abschnitt

In Abbildung 4.14 wird die durchschnittlich zurückgelegte euklidische Distanz pro Greedy- bzw. Face-Abschnitt dargelegt. Während die Hopanzahl pro Greedy-Abschnitt mit steigendem  $\sigma$ -Wert fallend war, ist die zurückgelegte euklidische Distanz steigend. Setzt man die durchschnittlich zurückgelegte euklidische Länge ins Verhältnis zur Feldgröße, entspricht sie jeweils ungefähr der halben Größe des simulierten Bereiches. Im Verhältnis wurde also die gleiche Strecke zurückgelegt, jedoch wurden bei größeren  $\sigma$ -Werten dafür weniger Hops benötigt. Diese Beobachtung ist damit zu begründen, dass größere  $\sigma$ -Werte auch längere Kanten ermöglichen.

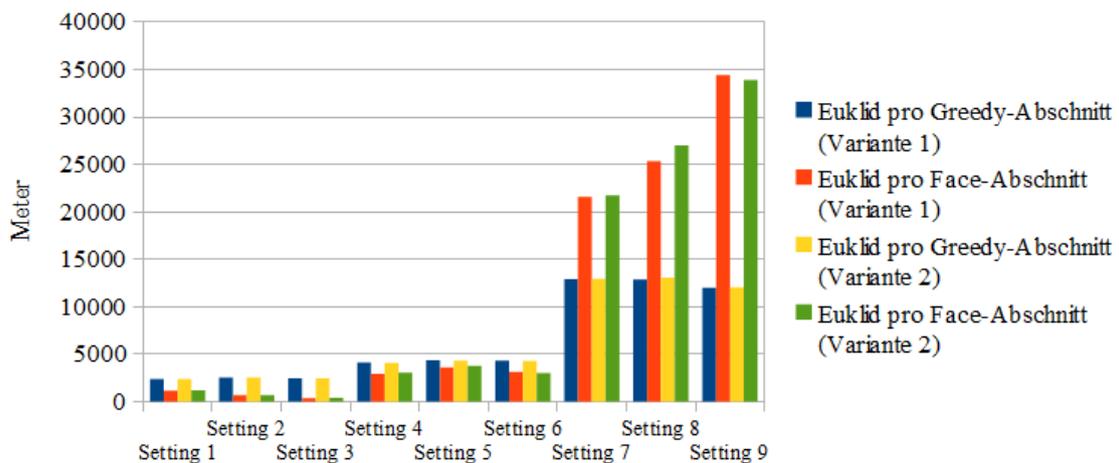


Abb. 4.14: Durchschnittlich zurückgelegte euklidische Länge pro Greedy- bzw. Face-Abschnitt

Die euklidische Distanz pro Face-Routingabschnitt nimmt zu, wenn durch die Zufallsvariable  $X_\sigma$  größere Schwankungen repräsentiert werden. Dieser Trend zeigt sich auch dann, wenn man die zurückgelegte Distanz im Verhältnis zur Feldgröße betrachtet. Die Ursache könnte auch hier in der unregelmäßigeren Graphenstruktur liegen, die durch höhere  $\sigma$ -Werte hervorgerufen wird.

### 4.4.3 Bewertung des Wechselverhaltens zwischen Greedy- und Face-Abschnitten

In dichten Graphen ist die Wahrscheinlichkeit geringer, dass Greedy-Routing ein lokales Minimum erreicht. Entsprechend treten hier weniger Wechsel zwischen Greedy- und Face-Routing auf als in dünnen Graphen. Das Bestreben, möglichst oft Greedy-Routing aufgrund der effizienteren Routingpfade zu nutzen, spiegelt sich auch im Verhältnis von Greedy- zu Face-Abschnitten wider. Face-Routing dient lediglich als Recovery-Strategie und kommt daher seltener zum Einsatz. Trotz sinkender Hopanzahl bei steigendem  $\sigma$ -Wert wird in den Greedy-Abschnitten im Verhältnis zur Feldgröße in etwa die gleiche Distanz zurückgelegt, während sich der durchlaufene Pfad in den Face-Abschnitten mit steigendem  $\sigma$ -Wert verlängert.

# Kapitel 5

## Fazit

Unter realistischen Netzwerkbedingungen hat sich die Planarisierung mit lokalen Mitteln als schwierige Aufgabe erwiesen, und es bestehen aktuell noch keine praktikabel einsetzbaren Verfahren. Daher beschäftigte sich diese Arbeit mit der Frage, welche Auswirkungen es auf die Erfolgsrate von lokalen Routingverfahren hat, wenn der Graph lediglich teilplanarisiert wird. Die Basis des Konnektivitätsgraphen stellte dabei das Log-Normal-Shadowing-Modell dar, mit dessen Hilfe sich realistische Settings nachbilden lassen. Die Untersuchungsgrundlage bildete eine Simulationsumgebung, die in Form eines Java-Programms realisiert wurde. Für neun verschiedene Settings wurden jeweils 2000 Durchläufe simuliert und die Ergebnisse ausgewertet. Die umgesetzten lokalen Planarisierungsstrategien betrachteten lediglich die Endpunkte der beiden schneidenden Kanten und deren Verbindungen zueinander, um zu entscheiden, ob eine Kante entfernt werden kann oder nicht. In den tendenziell unrealistischen Settings 1 bis 3, bei denen keine äußeren Einflüsse wie Hindernisse usw. in Betracht gezogen wurden, konnten mit diesen Verfahren über 99 % der Schnitte entfernt werden. Unter Berücksichtigung einer gewissen Varianz in den Settings 4 bis 9 verblieben deutlich mehr schneidende Kanten in den Subgraphen.

Eine leicht erweiterte Planarisierungsvariante sollte dazu führen, dass weniger Schnitte im Subgraphen zurückbleiben. Eine mögliche Erweiterung der Planarisierungsverfahren könnte nicht nur eine Überprüfung der Verbindungen zwischen diesen Endpunkten, sondern auch die Suche nach einem alternativen Weg im  $k$ -Hop Bereich umfassen. Hierbei wäre es interessant zu testen, bis zu welchem  $k$  noch eine Verbesserung bezüglich der Eliminierung von Schnitten bei angemessenem Aufwand zu erzielen ist.

In der Simulationsumgebung wurde Greedy-, Face- und kombiniertes Greedy-Face-Routing umgesetzt, um zu ermitteln, wie sich die verschiedenen Routingverfahren unter realistischen Bedingungen verhalten, wenn der Subgraph lediglich teilplanarisiert ist.

Die Fehlerrate von Greedy-Routing stieg mit sinkender Graphendichte und steigendem  $\sigma$ -Wert stark an. Falls ein Weg zum Ziel gefunden wurde, erwies sich die Wegwahl aber wiederum als sehr effizient und verursachte nur einen geringen Stretchfaktor.

Es wurde eine Face-Routingvariante implementiert, die aus zwei Gründen fehlschlagen konnte. Misserfolge können zum einen durch verbliebene Kanten im teilplanarisierten Subgraphen verursacht werden. Darüber hinaus schlägt der Face-Routingalgorithmus fehl, wenn es die Topologie erforderlich macht, dass auf dem Weg zum Ziel eine bestimmte einzelne Kante durchlaufen wird, die gleichzeitig einen Schnitt mit der  $st$ -Linie aufweist. Dieses Szenario ist in dünnen Graphen wahrscheinlicher als in dichten. Gleichzeitig wies Face-Routing gerade in den Topologien eine sehr niedrige Erfolgsrate auf, in denen viele Kantenschnitte im teilplanarisierten Subgraphen zurückblieben. Diese Tatsache führt zu der Schlussfolgerung, dass die verbliebenen Restkanten die Erfolgsquote von Face-Routing doch recht stark negativ beeinflussen.

Aufgrund der umgesetzten Face-Routingvariante können Misserfolge des kombinierten Routingverfahrens nicht eindeutig auf Kantenschnitte zurückgeführt werden. Trotzdem lag die Erfolgsquote von kombiniertem Greedy-Face-Routing in allen Settings deutlich über der der getesteten Einzelvarianten. Dieses Ergebnis zeigte sich auch bei den Netzwerkgraphen, bei denen in der teilplanarisierten Variante viele Schnitte zurückblieben.

Die Beobachtung, dass kombiniertes Greedy-Face-Routing vor allem in in den realistischen Settings eine deutlich höhere Erfolgsrate liefert als Greedy- oder Face-Routing alleine, führt zu der These, dass auch eine Teilplanarisierung des Netzwerkgraphen ausreichend sein könnte, um praktikable Auslieferungsraten zu erreichen. Dazu wäre es sinnvoll, das Verhalten der verschiedenen Routingverfahren in realistischen Settings erneut zu untersuchen, diesmal jedoch eine Face-Routingvariante zu verwenden, die lediglich aufgrund von Kantenschnitten fehlschlägt. Kombiniert man diese Variante mit einer verfeinerten Planarisierungsstrategie, ist es nicht auszuschließen, dass gute bis sehr gute Erfolgsquoten bezüglich der Auslieferung erzielt werden können.

# Literaturverzeichnis

- [1] Khaled M. Alzoubi, Peng-Jun Wan und Ophir Frider: "*Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks*", In: Proc. 3<sup>rd</sup> ACM Int. Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC), 2002
- [2] Lali Barrière, Pierre Fraigniaud, Lata Narayanan und Jaroslav Opatrny: "*Robust position-based routing in wireless ad hoc networks with irregular transmission ranges*", In: Wireless Communications and Mobile Computing, Seite 141-153, 2003
- [3] Prosenjit Bose, Pat Morin, Ivan Stojmenović und Jorge Urrutia: "*Routing with Guaranteed Delivery in Ad Hoc Wireless Networks*", In: Wireless Networks, Vol. 7, Seite 609-616, November 2001
- [4] Jehoshua Bruck, Jie Gao und Anxiao Jiang: "*Localization and Routing in Sensor Networks by Local Angle Information*", In: ACM Transactions on Sensor Networks (TOSN), Vol. 5, Februar 2009
- [5] Nirupama Bulusu, John Heidemann, Deborah Estrin und Tommy Tran: "*Self-Configuring Localization Systems: Design and Experimental Evaluation*", In: ACM Transactions on Embedded Computing Systems (TECS), Volume 3, Seite 24-60, Februar 2004
- [6] Paolo Casari, Michele Nati, Chiara Petrioli und Michele Zorzi: "*ALBA: An adaptive load-balanced algorithm for geographic forwarding in wireless sensor networks*", In: Proc. IEEE Military Communications Conference (MILCOM), Seite 1-9, Oktober 2006
- [7] Paolo Casari, Michele Nati, Chiara Petrioli und Michele Zorzi: "*Efficient Non-Planar Routing around Dead Ends in Sparse Topologies using Random Forwarding*", In: Proc. IEEE International Conference on Communications, Seite 3122-3129, 2007
- [8] Ernest J. H. Chang: "*Echo Algorithms: Depth Parallel Operations on General Graphs*", In: IEEE Transactions on Software Engineering, Vol. SE-8, No. 4, Seite 391-401, Juli 1982

- [9] Yuan-Po Cheng, Yao-Jen Tang and Ming-Jer Tsai: "*LF-GFG: Location-Free Greedy-Face-Greedy Routing with Guaranteed Delivery and Lightweight Maintenance Cost in a Wireless Sensor Network with Changing Topology*", In: IEEE Transactions on Wireless Communications, Vol. 13, Seite 7025-7036, September 2014
- [10] Susanta Datta, Ivan Stojmenović und Jie Wu: "*Internal node and shortcut based routing with guaranteed delivery in wireless networks*", In: Proc. 21<sup>st</sup> International Conference on Distributed Computing Systems Workshops, April 2001
- [11] Stephane Durocher, David Kirkpatrick und Lata Narayanan: "*On routing with guaranteed delivery in three-dimensional ad hoc wireless networks*", In: Wireless Networks, Vol. 16, Seite 227-235, Januar 2010
- [12] Gregory G. Finn: "*Routing and Addressing Problems in Large Metropolitan-Scale Internetworks*", Technical Report ISI/RR-87-180, Information Sciences Inst. (ISI), März 1987
- [13] Hannes Frey: "*Scalable Geographic Routing Algorithms for Wireless Ad Hoc Networks*", In: IEEE Network, Vol. 18, Seite 18-22, Juli 2004
- [14] Hannes Frey und Iven Stojmenović: "*On Delivery Guarantees and Worst-Case Forwarding Bounds of Elementary Face Routing Components in Ad Hoc and Sensor Networks*", In: Proc. 12<sup>th</sup> ACM Annual International Conference on Mobile Computing and Networking (MobiCom), Vol. 59, Seite 390-401, 2006
- [15] K. Ruben Gabriel und Robert R. Sokal: "*A New Statistical Approach to Geographic Variation Analysis*", Systematic Zoology 18, Seite 259-278, 1969
- [16] Deepak Ganesan, Deborah Estrin, Alec Woo und David Culler: "*Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks*", August 2002
- [17] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang und An Zhu: "*Geometric Spanner for Routing in Mobile Networks*", In: IEEE Journal on Selected Areas in Communications, Vol. 23, Seite 174-185, Januar 2005
- [18] R. Hekmat und P. van Mieghem: "*Connectivity in Wireless Ad-hoc Networks with a Log-normal Radio Model*", In: Mobile Networks and Applications, Vol. 11, Seite 351-360, Juni 2006
- [19] Jeffrey Hightower und Gaetano Borriello: "*Location Systems for Ubiquitous Computing*", In: Computer, Vol. 24, Seite 57-66, August 2001
- [20] Ting-Chao Hou und Victor O. K. Li: "*Transmission Range Control in Multihop Packet Radio Networks*", In: IEEE Transactions on Communications, Vol. 34, Seite 38-44, Januar 1986

- [21] Holger Karl und Andreas Willig: *"Protocols and Architectures for Wireless Sensor Networks"*, 1. Auflage, John Wiley & Sons: West Sussex, 2005
- [22] Brad Nelson Karp: *"Geographic Routing for Wireless Networks"*, Phd Thesis, Harvard University, Oktober 2000
- [23] Brad Karp: *"Challenges in Geographic Routing: Sparse Networks, Obstacles, and Traffic Provisioning"*, Präsentation beim DIMACS Pervasive Networking Workshop, Mai 2001
- [24] Brad Karp und H. T. Kung: *"GPSR: Greedy Perimeter Stateless Routing for Wireless Networks"*, In: Proc. 6<sup>th</sup> ACM Annual International Conference on Mobile Computing and Networking (MobiCom), Seite 243-254, August 2000
- [25] Young-Jin Kim, Ramesh Govindan, Brad Karp und Scott Shenker: *"On the Pitfalls of Geographic Face Routing"*, In: Proc. Workshop on Foundations of Mobile Computing (DIALM-POMC), Seite 34-43, 2005
- [26] Young-Jin Kim, Ramesh Govindan, Brad Karp und Scott Shenker: *"Geographic Routing Made Practical"*, In: Proc. 2<sup>nd</sup> Conference on Symposium on Networked Systems Design & Implementation - Volume 2, Seite 217-230, Mai 2005
- [27] Evangelos Kranakis, Harvinder Singh und Jorge Urrutia: *"Compass Routing on Geometric Networks"*, In: Proc. 11<sup>th</sup> Canadian Conference on Computational Geometry, Seite 51-54, August 1999
- [28] Fabian Kuhn, Roger Wattenhofer, Yan Zhang und Aaron Zollinger: *"Geometric Ad-Hoc Routing: Of Theory and Practice"*, In: Proc. 22<sup>nd</sup> Annual Symposium on Principles of Distributed Computing (PODC), Seite 63-72, Juli 2003
- [29] Fabian Kuhn, Roger Wattenhofer und Aaron Zollinger: *"Asymptotically Optimal Geometric Mobile Ad-Hoc Routing"*, In: Proc. 6<sup>th</sup> International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), Seite 24-33, September 2002
- [30] Fabian Kuhn, Roger Wattenhofer und Aaron Zollinger: *"Ad-Hoc Networks Beyond Unit Disk Graphs"*, In: Proc. 2003 joint Workshop on Foundations of Mobile Computing, Seite 69-78, 2003
- [31] Fabian Kuhn, Roger Wattenhofer und Aaron Zollinger: *"Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing"*, In: Proc. 4<sup>th</sup> ACM international Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc), Seite 267-278, Juni 2003
- [32] Yujun Li, Yaling Yang und Xianliang Lu: *"Rules of Designing Routing Metrics for Greedy, Face, and Combined Greedy-Face Routing"*, In: IEEE Transactions on Mobile Computing, Vol. 9, Seite 582-595, April 2010

- [33] Kevin M. Lillis, Sriram V. Pemmaraju und Imran A. Pirwani: "*Topology Control and Geographic Routing in Realistic Wireless Networks*" Ad Hoc & Sensor Wireless Networks, Vol. 6, Seite 265-297, 2008
- [34] Pratap Misra und Per Enge: "Global Positioning System: Signals, Measurement and Performance", 2. Auflage, Ganga-Jamuna Press: Massachusetts, 2012
- [35] Dragos Niculescu und Badri Nath: "*Ad Hoc Positioning System (APS)*", IEEE Global Telecommunication Conference GLOBECOM, November 2001
- [36] David Peleg: "*Distributed Computing, A Locality-Sensitive Approach*", 1. Auflage, Siam: Philadelphia, 2000
- [37] Elizabeth M. Royer und C-K Toh: "*A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks*", In: IEEE Personal Communications, Vol. 6, Seite 46-55, April 1999
- [38] Paolo Santi: "*Topology Control in Wireless Ad Hoc and Sensor Networks*", In: ACM Computing Surveys (CSUR), Val. 37, Seite 164-194, Juni 2005
- [39] Andreas Savvides, Chih-Chieh Han und Mani B. Strivastave: "*Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors*", In: Proc. 7<sup>th</sup> Annual International Conference on Mobile Computing and Networking (MOBICOM), Seite 166-179, Juli 2001
- [40] Karim Seada, Ahmed Helmy und Ramesh Govindan: "*Modeling and analyzing the correctness of geographic face routing under realistic conditions*", In: Ad Hoc Networks, Vol. 5, Seite 855-871, August 2007
- [41] Karim Seada, Ahmed Helmy und Ramesh Govindan: "*On the Effect of Localization Errors on Geographic Face Routing in Sensor Networks*", ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 7, Oktober 2003
- [42] Ivan Stojmenović: "*Position-Based Routing in Ad Hoc Networks*", In: IEEE Communications Magazine, Val. 40, Seite 128-134, August 2002
- [43] Milos Stojmenovic und Amiya Nayak: "*Localized routing with guaranteed delivery and a realistic physical layer in wireless sensor networks*", In: Computer Communications, Vol. 29, Seite 2550-2555, 2006
- [44] Hideaki Takagi und Leonard Kleinrock: "*Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals*", In: IEEE Transactions on Communications, Val. 32, Seite 246-257, März 1984
- [45] Fabrice Theolayre, Eryk Schiller und Andrzej Duba: "*Efficient Greedy Geographical Non-Planar Routing with Reactive Deflection*", IEEE International Conference on Communications, Juni 2009

- [46] Godfried T. Toussaint: "*The Relative Neighbourhood Graph of a Finite Planar Set*", In: Pattern Recognition, Vol. 12, Seite 261-268, 1980
- [47] Alec Woo, Terence Tong und David Culler: "*Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks*", In: Proc. 1<sup>st</sup> International Conference on Embedded Networks Sensor Systems (SenSys), Seite 14-27, November 2003
- [48] Jiuqiang Xu, Wei Liu, Fenggao Lang, Yuanyuan Zhang und Chenglong Wang: "*Distance Measurement Model Based on RSSI in WSN*", In: Wireless Sensor Networks, Vol. 2, Seite 606-611, August 2010
- [49] Jerry Zhao und Ramesh Govindan: "*Understanding Packet Delivery Performance In Dense Wireless Sensor Networks*", In: Proc. 1<sup>st</sup> International Conference on Embedded Networks Sensor Systems (SenSys), Seite 1-13, November 2003
- [50] Gang Zhou, Tian He, Sudha Krishnamurthy und John A. Stankovic: "*Impact of Radio Irregularity on Wireless Sensor Networks*", In: Proc. 2<sup>nd</sup> International Conference on Mobile Systems, Applications, and Services (MobiSys), Seite 125-138, Juni 2004
- [51] Michele Zorzi und Ramesh R. Rao: "*Geographic Random Forwarding (GeReF) for Ad Hoc and Sensor Networks: Energy and Latency Performance*", In: IEEE Transactions on Mobile Computing, Vol. 2, Seite 349-365, Oktober 2003