The Many Faces of Planarity
– Matching, Augmentation, and Embedding Algorithms for Planar Graphs –

Ignaz Rutter

Karlsruher Institut für Technologie (KIT)
Fakultät für Informatik
Am Fasanengarten 5
76131 Karlsruhe
rutter@kit.edu

Abstract: Ein Graph ist *planar*, wenn er sich kreuzungsfrei in die Ebene zeichnen lässt. Planarität ist eine zentrale Eigenschaft, nicht nur im Graphenzeichnen, sondern in der gesamten Graphentheorie. Oftmals lassen sich für planare Graphen stärkere theoretische Aussagen beweisen und effizientere Algorithmen angeben als für allgemeine Graphen. Andererseits tritt Planarität oft auch als Nebenbedingung auf und macht Probleme dadurch schwieriger. Eine besondere Rolle spielen planare Graphen in der Visualisierung, da Kreuzungen die Lesbarkeit von Zeichnungen verschlechtern.

In der vorliegenden Dissertation [Rut11] untersuche ich eine Reihe von Problemen, in denen Planarität auf unterschiedliche Weise auftritt. Im Bereich der kombinatorischen Optimierung wird Planarität als Nebenbedingung für Graphaugmentierungsprobleme sowie als Eingaberestriktion für Matching-Probleme betrachtet und beleuchtet inwiefern dies die Komplexität der jeweiligen Probleme verändert. Der zweite Teil der Arbeit befasst sich mit der Visualisierung planarer Graphen. Bisherige Verfahren zur planaren Visualisierung legen häufig zunächst eine kombinatorische Einbettung fest und optimieren dann im Rahmen dieser Einbettung weitere ästhetische Kriterien. Die Einschränkung auf eine einzige anfangs gewählte Einbettung erweist sich dabei häufig als nachteilig. Ich stelle Verfahren vor, die es ermöglichen über alle Einbettungen eines planaren Graphen zu optimieren und unter allen Einbettungen eine zu finden, die für die Visualisierung am besten geeignet ist.

1 Einleitung

Die Graphentheorie ist ohne Zweifel eine der großen Erfolgsgeschichten der diskreten Mathematik, und insbesondere hinsichtlich der automatisierten Verarbeitung auch der Informatik. Graphen und Netzwerke sind ubiquitär; sie werden auch in Bereichen weit jenseits dieser beiden Urpsrungsgebiete eingesetzt, um Relationen zwischen unterschiedlichsten Entitäten zu modellieren, zu studieren und zu verstehen, beispielsweise in der Physik, Biologie, den Sozialwissenschaften, aber auch um IT-Infrastrukturnetzwerke oder Prozessmodelle zu beschreiben. Menschen sind von Natur aus sehr visuell orientiert. Daher geht die Verwendung von Graphen um komplexe Relationen zu beschreiben häufig einher mit einer entsprechenden Visualisierung der Graphen. Bei zunehmender Anzahl von

Kantenkreuzungen in einer Zeichnung reduziert sich die Lesbarkeit drastisch. Daher ist es intuitiv, Kreuzungen gänzlich zu vermeiden; dies führt zur Definition der Klasse der planaren Graphen.

Heutzutage ist Planarität ein zentrales Konzept, nicht nur im Graphenzeichnen, sondern in der gesamten Graphentheorie. Die Charakterisierung planarer Graphen durch Kuratowski [Kur30] im Jahr 1930, kann als Geburtsstunde der modernen Graphentheorie angesehen werden. Die Charakterisierung über verbotene Substrukturen, nämlich K_5 (den vollständigen Graphen mit fünf Knoten) und $K_{3,3}$ (bestehend aus zwei Gruppen zu je drei Knoten, bei dem jeder Knoten mit allen Knoten der anderen Gruppe verbunden ist), zeigt, dass Planarität ein "endliches" Problem ist und führte zu den ersten polynomiellen Erkennungsalgorithmen. Den ersten Linearzeitalgorithmus zur Erkennung von planaren Graphen veröffentlichten Hopcroft und Tarjan 1974 [HT74], inzwischen ist eine Reihe von linearen Planaritätstests bekannt.

Die planaren Graphen bilden wahrscheinlich eine der am besten untersuchten Graphklassen. Eine Fülle von Literatur zeigt das immense Interesse an ihren Eigenschaften, Zeichenalgorithmen und Optimierungsalgorithmen, die speziell auf planare Graphen zugeschnitten sind. Beispielsweise besitzen planare Graphen gute Zerlegungseigenschaften, lassen sich mit wenigen Farben färben und viele Lösungen für Standardprobleme, die häufig als Subroutine in anderen Algorithmen eingesetzt werden, lassen sich auf planaren Graphen besonders effizient implementieren. Hierzu zählen beispielsweise Matching- und Flussalgorithmen. Zudem dienen planare Graphen oft als Sprungbrett für die Entwicklung effizienter Algorithmen auf allgemeineren Graphklassen, etwa Graphen mit beschränktem Genus oder den sogenannten H-minorenfreien Graphen.

Planarität ist eine Eigenschaft mit vielen unterschiedlichen Aspekten, da sie beispielsweise im Rahmen von kombinatorischen Optimierungsproblemen verschiedene Rollen einnehmen kann, beispielsweise die einer zusätzlichen Nebenbedingung aber auch die einer Eingaberestriktion. Die Nützlichkeit von Planarität und den damit einhergehenden Grapheigenschaften variiert stark mit dem betrachteten Problem und insbesondere mit der Rolle, die Planarität in dem Problem spielt. Um dem Rechnung zu tragen, ist die Arbeit in zwei Teile gegliedert. Im ersten Teil stehen Fragestellungen der kombinatorischen Optimierung im Vordergrund. Dort treten zwei Facetten von Planarität zutage: Einerseits wird Planarität als zusätzliche, hilfreiche Eigenschaft der Eingabe ausgenutzt, andererseits tritt Planarität auch als Nebenbedingung auf, deren Einhaltung durch die Problemstellung gefordert wird und die Probleme häufig schwieriger macht. Der zweite Teil befasst sich mit dem Zeichnen von planaren Graphen. Dabei spielt die Wahl der Einbettung eines planaren Graphen eine wesentliche Rolle für die Qualität der Darstellung. Ich stelle eine Reihe von Verfahren vor, die möglichst gute Einbettungen von planaren Graphen für verschiedene Zeichenstile berechnen.

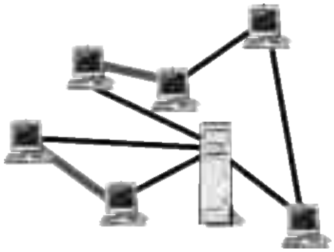


Abbildung 1: Durch Einfügen der dicken Kanten wird das Netzwerk gegen den Ausfall einer einzelnen Kante abgesichert.

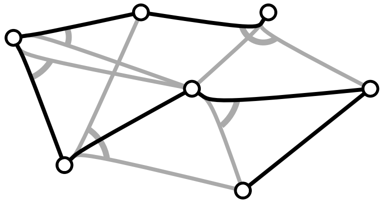


Abbildung 2: Ein Schaltergraph, Kanten desselben Schalters sind gemeinsamen Fußpunkt mit einem Bogen verbunden.

2 Kombinatorische Optimierung auf planaren Graphen

Der erste Teil der Arbeit beschäftigt sich mit verschiedenen kombinatorischen Optimierungsproblemen auf planaren Graphen, dabei tritt Planarität entweder als Restriktion der Eingabe oder als zusätzliche Nebenbedingung auf. Es werden drei unterschiedliche Fragestellungen behandelt.

2.1 Graphaugmentierung

Robustheit ist ein grundlegendes Kriterium beim Aufbau von Infrastruktur-Netzen, wie etwa Computernetzwerken. Ein häufig angewandtes Kriterium ist zum Beispiel zu fordern, dass der Graph zumindest *zweifach kantenzusammenhängend* ist, also nicht durch Ausfall einer einzelnen Kante in mehrere Zusammenhangskomponenten zerfällt. Andererseits ist man daran interessiert, die Kosten möglichst gering zu halten, sodass ein Netzwerk, in dem alle Knoten paarweise miteinander verbunden sind, zu teuer ist. Zudem werden Infrastruktur-Netzwerke häufig nicht vollständig neu aufgebaut, sondern ein bestehendes Netzwerk soll durch zusätzliche Komponenten möglichst kostengünstig erweitert werden, um neuen Anforderungen gerecht zu werden. Man ist also daran interessiert einen gegebenen Graphen, etwa durch Hinzufügen möglichst weniger Kanten, so zu modifizieren, dass er gewisse zusätzliche Eigenschaften erhält. Es wurde bereits eine Reihe solcher *Graphaugmentierungs-Probleme* im Zusammenhang mit Robustheit betrachtet. Dabei soll insbesondere der Zusammenhangsgrad eines gegebenen Graphen durch Hinzufügen möglichst weniger Kanten erhöht werden. Abbildung 1 zeigt ein Beispielnetzwerk, welches durch Einfügen der dicken Kanten zweifach kantenzusammenhängend wird. Hierbei handelt es sich um eine klassische und gut untersuchte Problemstellung. In meiner Arbeit fordere ich nun zusätzlich, dass der augmentierte Graph planar bleibt. Dieses Problem tritt beispielsweise im Graphenzeichnen auf, da viele Zeichenalgorithmen für planare Graphen zweifachen Zusammenhang voraussetzen oder zumindest für diese

Art von Graphen besondere Qualitätsgarantien angeben. Die Forderung möglichst wenige Kanten hinzuzufügen sorgt dafür, dass diese Qualitätsgarantien möglichst gut erhalten bleiben.

Ich untersuche den Komplexitätsstatus einer Reihe von planaren Augmentierungsproblemen. Inbesondere zeige ich, dass das Problem einen planaren Graphen durch Hinzufügen einer minimalen Anzahl von Kanten zweifach kantenzusammenhängend zu machen NP-schwer ist. Dies beantwortet eine offene Frage von Kant, der eine analoge Aussage für zweifachen Knotenzusammenhang gezeigt hat [KB91]. Weiter wird das analoge Problem betrachtet, bei dem der Graph *geometrisch eingebettet* ist, also jeder Knoten bereits eine fest zugewiesene Position hat und die Kanten geradlinig gezeichnet werden müssen. In der Arbeit wird gezeigt, dass einen geometrischen Graphen planar zweifach zusammenhängend zu machen selbst für Bäume NP-schwer ist, und auch das erhöhen zu c-fachem Zusammenhang für $c \geq 3$ schwer ist. Fordert man jedoch zusätzlich, dass die Knoten sich in konvexer Lage befinden, so ist das Problem effizient lösbar. In diesem Fall lässt sich nicht nur die Anzahl der zusätzlich nötigen Kanten minimieren, sondern auch das allgemeinere Kostenminimierungsproblem in $O(n^2)$ Zeit lösen, bei dem jeder möglichen zusätzlichen Kante ein positiver Kostenwert zugeordnet wird. Dieses Problem selbst ohne die Planaritätsbedingung im allgemeinen NP-schwer, und zwar sogar dann, wenn die Gewichte auf die Menge $\{1, 2\}$ beschränkt sind.

2.2 Schaltergraphen

Schaltergraphen bieten erweiterte Modellierungsmöglichkeiten gegenüber gewöhnlichen Graphen. Ein *Schalter* besteht aus einer Menge von Kanten, die sich einen gemeinsamen Knoten teilen. Eine *Konfiguration* wählt aus jedem Schalter eine Kante aus. Ein Schaltergraph beschreibt also eine Familie von Graphen und eine Konfiguration beschreibt ein konkretes Mitglied dieser Familie. Abbildung 2 zeigt einen Schaltergraphen, wobei Kanten, die zum selben Schalter gehören, durch einen Bogen am gemeinsamen Knoten miteinander verbunden sind. Die hervorgehobenen Kanten bilden eine Konfiguration dieses Schaltergraphen, deren resultierender Graph zusammenhängend ist. So lassen sich beispielsweise Computer- und Telefonnetzwerke oder auch Eisenbahnnetze mit Weichen auf sehr natürliche Weise als Schaltergraphen modellieren.

Aufgrund ihres Aufbaus eignen sich Schaltergraphen gut, um graphentheoretische Probleme zu modellieren, die strukturelle Entscheidungen beinhalten. Hat man einen Schaltergraphen vorliegen, ist man daran interessiert herauszufinden, ob seine Familie einen Graphen enthält, der eine gegebene Grapheigenschaft besitzt. Beispielsweise ist man daran interessiert, ob man die Verbindungen der Schalter so auswählen kann, dass zwei gegebene Teilnehmer eines Telefonnetzwerks miteinander verbunden sind. Eine weitere Problemstellung ergibt sich beispielsweise daraus, alle Teilnehmer einer Telefonkonferenz, oder gar alle Teilnehmer des Netzwerks, zusammenzuschalten. Schaltergraphen wurden von Groote und Ploeger eingeführt [GP08] und die vorliegende Arbeit beantwortet eine Reihe von offenen Fragen aus ihrer Arbeit.

Ich gebe effiziente Algorithmen an, mit denen überprüft werden kann, ob eine Verbindung zwischen zwei Knoten hergestellt werden kann, und auch, ob sich das gesamte Netzwerk zusammenschalten lässt; für beliebige Teilmengen hingegen ist das Problem NP-schwer. Zudem wird eine Reihe anderer Eigenschaften hinsichtlich ihrer Komplexität untersucht. So ist es beispielsweise NP-schwer zu entscheiden, ob ein Schaltergraph eine planare Konfiguration besitzt und ob man den Graphen so konfigurieren kann, dass er eulersch ist. Letzteres ist vor allem deswegen erstaunlich, da ein Graph genau dann eulersch ist, wenn er zusammenhängend ist und alle Knoten geraden Grad haben. Konfigurationen für die letztere beiden Eigenschaften können mit Algorithmen aus der Arbeit jedoch separat effizient gefunden werden.

2.3 Große Matchings in planaren Graphen

Ein *Matching* ist eine Teilmenge der Kanten eines Graphen, bei der jeder Knoten zu höchstens einer Kante dieser Menge inzident ist. Das Finden von möglichst großen Matchings ist ein gut untersuchtes Problem aus der kombinatorischen Optimierung, welches häufig als Subroutine in anderen Algorithmen zum Einsatz kommt. Der beste bekannte Algorithmus für dieses Problem hat eine Laufzeit von $O(\sqrt{n}m)$ [MV80]. Erst seit kurzem sind bessere Algorithmen, beispielsweise für planare Graphen, bekannt [MS06]. Allerdings basieren diese Verfahren auf schneller Matrix-Multiplikation, einem nicht sehr praxis-tauglichen Werkzeug. In der Praxis werden daher fast ausschließlich einfachere, langsamere Verfahren mit einer Laufzeit von $O(nm)$ eingesetzt.

Bei der Verwendung eines Matching-Algorithmus als Subroutine ist es häufig nicht zwingend nötig ein größtes Matching zu finden, sondern es genügt ein großes Matching (mit garantierter Mindestgröße) zu finden. Nishizeki und Baybars [NB79] zeigten, dass in planaren Graphen mit festem Minimalgrad stets Matchings einer bestimmten Mindestgröße existieren. Keines der bisherigen Verfahren ist jedoch in der Lage ein solches Matching, von dem man ja weiß, dass es existiert, schneller zu finden als ein größtes Matching. Ich stelle ein Verfahren vor, das ein solches Matching in linearer Zeit berechnet. Zunächst wird ein Verfahren angegeben, das auf allgemeinen Graphen mit Minimalgrad 3 arbeitet und für planare Graphen eine Mindestqualität garantiert, allerdings gelang es nicht mit einem solchen generischen Algorithmus die scharfe Schranke von Nishizeki und Baybars zu erreichen. Erst mit einem modifizierten Verfahren, das Planarität auch auf einer grundlegenderen Ebene ausnutzt und das Verfahren durch die inhärente in einer kombinatorischen Einbettung kodierten Informationen steuert, gelang es in linearer Zeit die scharfe Schranke von Nishizeki und Baybars erreicht. Das Verfahren lässt sich zudem auf größere Minimalgrade verallgemeinern und liefert für diese bessere Schranken.

3 Einbettungen von planaren Graphen

Der zweite Teil der Arbeit beschäftigt sich mit der Problemstellung, Einbettungen von planaren Graphen zu finden, die möglichst gut für bestimmte Visualisierungsarten geeignet sind. Diese Art von Problemen ist inhärent schwierig, da planare Graphen im Allgemeinen exponentiell viele planare Einbettungen besitzen. Es werden Einbettungsprobleme für unterschiedliche Zeichenstile untersucht, sowohl für topologische Zeichnungen, bei denen Kanten als beliebige Kurven gezeichnet werden dürfen, als auch für orthogonale Zeichnungen, bei denen Kanten nur aus horizontalen und vertikalen Streckensegmenten zusammengesetzt werden.

3.1 Planarität partiell eingebetteter Graphen

Durch die Abwesenheit von Kreuzungen sind planare Zeichnungen besonders gut lesbar. Zudem gibt es zahlreiche Algorithmen, die es erlauben zu einem planaren Graphen eine planare Einbettung sowie eine zugehörige Zeichnung zu berechnen. Diese Verfahren erlauben dem Anwender jedoch keine Kontrolle über das resultierende Layout. Wenn sich Netzwerke über die Zeit verändern, ist es wichtig, dass sich die Visualisierung der stabilen Teile eines Netzwerks möglichst wenig verändern, um dem Benutzer eine gute Orientierung zu ermöglichen. In meiner Arbeit betrachte ich daher die grundlegende Frage, ob eine gegebene planare Zeichnung eines Teils eines Netzwerkes sich auf planare Art und Weise zu einer Zeichnung des gesamten Netzwerkes erweitern lässt. Dabei darf der bereits vorgegebene Teil der Zeichnung nicht verändert werden. Die Komplexität dieses Problems hängt stark vom verwendeten Zeichenstil ab. Eine frühere Arbeit zeigte bereits, dass dieses Zeichnungs-Erweiterungsproblem für geradlinige Zeichnungen NP-schwer ist [Pat06]. Ich betrachte das entsprechende Problem für *topologischen Zeichnungen*, bei denen die Knoten eines Graphen durch Punkte und seine Kanten durch beliebige Jordankurven zwischen ihren Endpunkten repräsentiert werden. Abbildung 3 zeigt ein Beispiel eines bereits gezeichneten Teilgraphen mit der Aufgabe, einige weitere Kanten auf planare Weise in die Zeichnung einzufügen (die Instanz ist lösbar, aber nicht ganz offensichtlich; versuchen Sie es!). Versucht man zusätzlich noch die Kante 1–8 hinzuzufügen, so wird die Instanz unlösbar, obwohl der zugrundeliegende Graph selbst noch planar ist. Das partielle Einbettungsproblem unterscheidet sich also in diesem Punkt vom gewöhnlichen Planaritätstest.

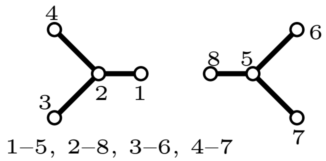


Abbildung 3: Ein partiell eingebetteter Graph. Die Aufgabe besteht darin, die gegebenen Kanten auf planare Weise zusätzlich in die Zeichnung einzufügen.

Für den Fall topologischer Zeichnungen zeige ich, dass das Problem äquivalent ist zu einem entsprechenden Erweiterungsproblem für kombinatorische Einbettungen. Ein *partiell eingebetteter Graph* (PEG) lässt sich als Tripel $(G, H, \mathcal{H})$ beschreiben, dabei ist G ein Graph und $H \subseteq G$ ein Teilgraph mit einer vorgegebenen planaren Einbettung $\mathcal{H}$. Die Frage besteht nun darin, ob eine planare Einbettung $\mathcal{G}$ von G existiert, deren Einschränkung auf H mit $\mathcal{H}$ übereinstimmt im Sinne folgender beider Eigenschaften: 1) Um jeden Knoten v von H ist die zirkuläre Reihenfolge seiner Nachbarn in H im Uhrzeigersinn um v dieselbe wie in $\mathcal{H}$, und 2) Für jeden (gerichteten) Kreis C in H liegen links bzw. rechts von C in $\mathcal{G}$ dieselben Knoten aus H wie in $\mathcal{H}$. In diesem Fall sagen wir, der PEG sei *planar*.

Überraschenderweise lässt sich das Problem zu testen ob ein PEG planar ist, anders als viele andere Lösungserweiterungsprobleme, effizient lösen, und zwar sogar in linearer Zeit. Das Verfahren hierzu verwendet zunächst eine Zerlegung in (zweifache) Zusammenhangskomponenten und anschließend den SPQR-Baum, eine Datenstruktur zur Repräsentation aller planaren Einbettungen von zweifachen zusammenhängenden planaren Graphen [DT96]. Die oben angegebenen Bedingungen lassen sich dann in diese Zusammenhangskomponenten und auch in den SPQR-Baum "projizieren", wo die Existenz einer Einbettung dann durch lokale Betrachtungen entschieden werden kann. Hierdurch ergibt sich ein relativ einfacher polynomieller Algorithmus, der sich mit Hilfe weiterer algorithmischer Techniken auf lineare Laufzeit beschleunigen lässt.

In einem weiteren Schritt werden zudem, nach dem Vorbild von Kuratowski, der zeigte, dass ein Graph genau dann planar ist, wenn er weder den Graph $K_{3,3}$ noch K_5 enthält, die planaren partiell eingebetteten Graphen durch verbotene Substrukturen charakterisiert. Hierzu werden zunächst die grundlegenden Minor-Operationen auf partiell eingebettete Graphen erweitert. Die neuen PEG-Minor-Operationen definieren eine Ordnung auf der Menge der PEGs, mit der Eigenschaft, dass alle Elemente, die kleiner sind als ein planarer PEG, ebenfalls planar sind. Es muss daher eine Menge von minimalen nicht-planaren PEGs geben, mit der Eigenschaft, dass jeder nicht-planare PEG mindestens eine dieser *verbotenen Substrukturen* enthält. Zusätzlich zu den beiden Graph $K_{3,3}$ und K_5, die sich aus der Forderung der Planarität ergeben, identifiziere ich sieben weitere solche verbotenen Substrukturen (siehe Abbildung 4) und zeige, dass dies genau die minimalen nicht-planaren PEGs sind, die oben angesprochene Menge also endlich und in der Tat recht klein ist. Neben einer genauen kombinatorischen Charakterisierung der planaren PEGs über verbotene Substrukturen ergibt sich hieraus auch ein effizienter *zertifizierender* Planaritätstest für PEGs, der zu einer gegebenen Eingabe entweder eine gültige Einbettungserweiterung berechnet, oder eine verbotene Substruktur extrahiert und so belegt, dass eine gültige Erweiterung nicht existiert. Gerade für komplexe und daher fehleranfällige Algorithmen wie Planaritätstests hat sich ein solches Vorgehen bei der Implementierung von Algorithmen bewährt.

3.2 Simultane Einbettungen

Liegen zwei (oder mehr) Graphen auf derselben Knotenmenge vor, so ist man häufig daran interessiert, diese Graphen miteinander zu vergleichen, beispielsweise durch Angabe einer Zeichnung, die die Ähnlichkeiten möglichst gut hervorhebt. Selbst für Paare planarer Graphen ist der Vereinigungsgraph im Allgemeinen nicht planar. Daher sucht man nach einer sogenannten *simultanen Einbettung mit festen Kanten*, das heißt die Knoten der beiden

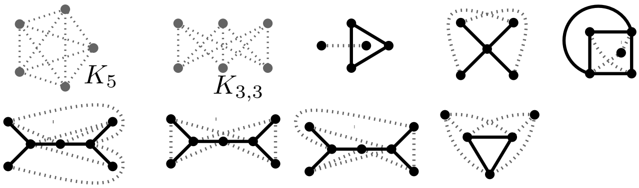


Abbildung 4: Die minimalen nicht-planaren PEGs. Der fest eingebettete Teilgraph H ist schwarz gezeichnet, die zusätzlichen Knoten und Kanten, deren Einbettung noch entschieden werden muss sind hell und gepunktet.

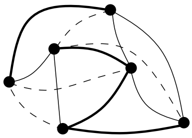


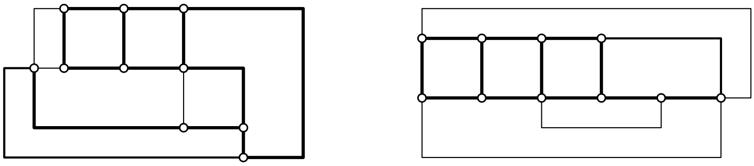
Abbildung 5: Simultane Einbettung zweier Graphen G_1 (durchgezogen) und G_2 (gestrichelt) gemeinsame Kanten sind fett gezeichnet.

Graphen werden an dieselben Positionen gezeichnet, gemeinsame Kanten werden durch dieselbe Kurve dargestellt und jede Zeichnung für sich genommen ist planar, siehe Abbildung 5. Für Tripel von planaren Graphen ist das entsprechende simultane Einbettungs-Problem NP-schwer [GJP$^+$06]. Obwohl sich in den letzten Jahren viele Wissenschaftler mit diesem Problem beschäftigt haben, ist der Komplexitätsstatus des Problems für Paare allgemeiner planarer Graphen noch ungeklärt. Bis dato existieren nur für sehr eingeschränkte Graphklassen effiziente Algorithmen. So lässt sich die Existenz einer simultanen Einbettung effizient überprüfen, wenn beide Graphen außenplanar sind oder wenn einer der beiden Graphen höchstens einen Kreis enthält.

Ich zeige, dass sich das Problem auch dann effizient, und zwar sogar in linearer Zeit, entscheiden lässt, wenn der Durchschnitt beider Graphen zweifach zusammenhängend ist. Dies ist einer der ersten Algorithmen, der zeigt, dass das Problem auch auf einer größeren Klasse von Graphen effizient lösbar ist. Zudem führe ich den Fall, dass der Durchschnitt zusammenhängend ist, zurück auf den Fall, dass der Durchschnitt ein Baum ist und alle weiteren Kanten zwischen Blättern dieses Baumes verlaufen. Dies zeigt neue Richtungen für algorithmische Ansätze für den allgemeinen Fall auf und grenzt die möglichen Bereiche für eine Suche nach potenziell schwierigen Instanzen weiter ein.

3.3 Orthogonale Zeichnungen mit Flexibilitäts-Bedingungen

Der letzte Abschnitt der Arbeit befasst sich mit dem *orthogonalen Zeichnen* von planaren Graphen, wobei Knoten auf dem Gitter platziert werden sollen und die Kanten aus horizontalen und vertikalen Streckensegmenten zusammengesetzt werden. Der Übergang von einem horizontalen auf ein vertikales Segment bzw. umgekehrt wird dabei als Knick



(a) (b)

Abbildung 6: Zwei unterschiedliche orthogonale Zeichnungen eines planaren Graphen.

bezeichnet. Da Kanten mit vielen Knicken die Lesbarkeit der Zeichnungen verschlechtern, ist man bestrebt die Anzahl der Knicke in der Zeichnung möglichst klein zu halten. Traditionell versucht man daher entweder die Gesamtanzahl an Knicken oder auch die maximale Anzahl an Knicken pro Kante zu minimieren. Beide Probleme sind NP-schwer, da es NP-schwer ist zu entscheiden, ob sich ein Graph gänzlich ohne Knicke zeichnen lässt [GT01]. Es ist aber bekannt, dass man mit höchstens zwei Knicken pro Kante stets auskommt [BK94]. Dagegen war bisher unklar, ob man effizient entscheiden kann, ob sich ein Graph mit nur einem Knick pro Kante orthogonal zeichnen lässt.

In dieser Arbeit wird ein neues Problem dieser Art eingeführt, bei dem jeder Kante eine individuelle Maximalzahl von Knicken zugeordnet wird, ihre *Flexibilität*. Dies umfasst das Problem der Zeichenbarkeit mit einem Knick pro Kante, ist aber noch wesentlich allgemeiner, da es dem Benutzer die genaue Kontrolle überlässt, welche Kanten wieviele Knicke haben dürfen. In Abbildung 6 ist ein Beispielgraph mit zwei verschiedenen orthogonalen Zeichnungen gegeben, die Wichtigkeit der Kanten ist durch ihre Dicke angegeben. Obwohl Zeichnung 6a hinsichtlich beider Qualitätsmaße besser ist als Zeichnung 6b, ist letztere übersichtlicher, da wichtige Kanten weniger Knicke haben. Erlaubt man starre Kanten, also solche mit Flexibilität 0, so ist das Problem offensichtlich NP-schwer. Mein Verfahren erlaubt es die Existenz einer Zeichnung im Rahmen der gegebenen Flexibilitäten effizient zu entscheiden, sofern man jeder Kante mindestens einen Knick erlaubt. Dies umfasst das Problem der Zeichenbarkeit mit einem Knick pro Kante und schließt damit die bisherige Lücke zwischen dem Schwere-Resultat für 0-Knick-Zeichenbarkeit und dem Algorithmus für das Zeichnen mit zwei Knicken pro Kante. Zudem liefert es aber einen wesentlich allgemeineren Ansatz zur Berechnung orthogonaler Zeichnungen, der dem Benutzer eine stärkere Kontrolle über die engültige Zeichnung gewährt.

Literatur

[BK94] Therese Biedl und Goos Kant. A better heuristic for orthogonal graph drawings. In *Proc. 2nd Europ. Symp. Algorithms (ESA'94)*, Jgg. 855 of *Lecture Notes Comput. Sci.*, Seiten 24–35. Springer-Verlag, 1994.

[DT96] Giuseppe DiBattista und Roberto Tamassia. On-Line Maintenance of Triconnected Components with SPQR-Trees. *Algorithmica*, 15:302–318, 1996.

[GJP+06] Elisabeth Gassner, Michael Jünger, Merijam Percan, Marcus Schaefer und Michael Schulz. Simultaneous Graph Embeddings with Fixed Edges. In F. V. Fomin, Hrsg., *WG '06*, Jgg. 4271 of *Lecture Notes Comput. Sci.*, Seiten 325–335, 2006.

[GP08] Jan F. Groote und Bas Ploeger. Switching graphs. In *Proceedings of the 2nd Workshop on Reachability Problems (RP'2008)*, ENTCS, Seiten 119–135, 2008.

[GT01] Ashim Garg und Roberto Tamassia. On the Computational Complexity of Upward and Rectilinear Planarity Testing. *SIAM J. Comput.*, 31(2):601–625, 2001.

[HT74] John E. Hopcroft und Robert E. Tarjan. Efficient Planarity Testing. *J. ACM*, 21(4):549–568, 1974.

[KB91] Goos Kant und Hans L. Bodlaender. Planar Graph Augmentation Problems. In Frank Dehne, Jörg-Rüdiger Sack und Nicola Santoro, Hrsg., *Proc. 2nd Workshop Algorithms and Data Structures (WADS'91)*, Jgg. 519 of *Lecture Notes Comput. Sci.*, Seiten 286–298. Springer-Verlag, 1991.

[Kur30] Kazimierz Kuratowski. Sur le problème des courbes gauches en topologie. *Fund. Math.*, 15:271–283, 1930.

[MS06] Marcin Mucha und Piotr Sankowski. Maximum Matchings in Planar Graphs via Gaussian Elimination. *Algorithmica*, 45(1):3–20, 2006.

[MV80] Silvio Micali und Vijay V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matchings in general graphs. In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci. (FOCS'80)*, Seiten 17–27, 1980.

[NB79] Takao Nishizeki und Ilker Baybars. Lower bounds on the cardinality of the maximum matchings of planar graphs. *Discrete Math.*, 28(3):255–267, 1979.

[Pat06] Maurizio Patrignani. On Extending a Partial Straight-Line Drawing. *Found. Comput. Sci.*, 17(5):1061–1069, 2006.

[Rut11] Ignaz Rutter. *The Many Faces of Planarity – Matching, Augmentation, and Embedding Algorithms for Planar Graphs –*. Dissertation, Fakultät für Informatik, Karlsruher Institut für Technologie (KIT), July 2011.

Ignaz Rutter, geboren 1981, studierte Informatik an der Universität Karlsruhe. Er schloss 2007 als Diplom-Informatiker ab und wurde mit dem Absolventenpreis für den besten Studienabschluss im akademischen Jahr 2006/2007 ausgezeichnet. Seit 2007 arbeitet er als wissenschaftlicher Mitarbeiter am Lehrstuhl Prof. Wagner am jetzigen Karlsruher Institut für Technologie (KIT). Die Promotion "summa cum laude" zum vorliegenden Thema erfolgte im Juli 2011. Im Januar 2012 wurde er mit dem Hermann-Billing-Preis für eine herausragende Dissertation an der Fakultät für Informatik des KIT im Jahr 2011 ausgezeichnet. Seine Interessensgebiete umfassen Algorithmen, Graphentheorie, Graphenzeichnen und kombinatorische Optimierung.