Beschriftungsproblem in hierarchisch gelayerten Graphen

Praktikumsbericht

David Dingel

22. Dezember 2021

Inhaltsverzeichnis

Der vorliegende Praktikumsbericht behandelt die Beschriftung von hierarchisch gruppierten Netzwerkplänen. Er baut auf einem vorherigen Resultat von Felix Klesen auf [Klesen18], und soll den dort entwickelten Algorithmus im Bezug auf die Platzierung von Labeln erweitern. Dafür werden zuerst zu dem Anwendungsfall passende Zielsetzungen formuliert, anhand derer die Ergebnisse bewertet werden können. Danach wird in Abschnitt 2 die für diese Arbeit entwickelte Vorgehensweise zur Erstellung der Pläne skizziert. Anschließend wird in Abschnitt 3 deren theoretische Performance entsprechend den Zielsetzungen passender Metriken analysiert, und in Abschnitt 4 werden schließlich Resultate einer Implementierung behandelt, bei der das Programm yFiles genutzt wurde. Zum Abschluss werden in Abschnitt 5 gewisse Schwachstellen besprochen, sowie mögliche Erweiterungen, um ein paar davon zu beseitigen.

1 Aufgabenstellung

Um sein internes IT-Netzwerk zu betreiben, fertigt ein Unternehmen regelmäßig Netzwerkpläne an. Diese sollen jeweils einen gegebenen Teilbereich des Netzwerkes visualisieren. Darin sollen wichtige Informationen zu den Geräten, wie deren IP-Adresse, Standort etc., sowie Informationen zu allen Verbindungen zwischen den Geräten enthalten sein. Die Pläne werden ausgedruckt und archiviert, sodass sie im Falle von Wartungsarbeiten von Technikern zur Hand gezogen werden können.

Aktuell werden all diese Pläne von Hand gezeichnet. Das manuelle Zeichnen ist nicht nur bei der initialen Erstellung zeitintensiv und unsicher, sondern es wird auch besonders kostenintensiv, wenn häufig kleine Änderungen stattfinden. Daher wäre es wünschenswert, diesen Prozess in Zukunft automatisch durchzuführen.

Das Unternehmen nutzt für ihr Netzwerkmanagement bereits das Produkt StableNet® der Firma Infosim®, in dem bereits alle für die Pläne relevanten Informationen hinterlegt sind. Daher bietet es sich an, diese Funktionalität in deren Produktumfang zu integrieren. Auf Anfrage des Unternehmens wurde daher in 2018 bereits eine Arbeit von Felix Klesen durchgeführt ([Klesen18]), in der er zeigen konnte, wie solche Pläne mithilfe von yFiles gelayoutet werden können. Darin beschreibt er ein geeignetes Vorgehen, um ein übersichtliches hierarchisch gruppiertes Layout zu zeichnen, welches größtenteils zur Verwendung in dieser Arbeit übernommen wurde. Eine von diesem Algorithmus erstellte Zeichnung findet sich in Abbildung 1. Hier wurde also bereits ein guter Rahmen geschaffen, um das Vorhaben tatsächlich realisieren zu können. Verbesserungspotenzial bestand jedoch noch bei der Beschriftung der Kanten, denn bei diesem Vorgehen konnte noch nicht sichergestellt werden, dass die Label lesbar bleiben. Im Rahmen dieser Arbeit soll nun behandelt werden, wie der Algorithmus dahingehend erweitert werden kann, dass möglichst übersichtliche, lesbare und nie überlappend beschriftete Pläne entstehen.

1.1 Rahmenbedingungen

Das übergeordnete Ziel ist, hierarchisch gelayerte und gruppierte Graphen zu zeichnen, welcher lesbare beschriftete Kanten haben. Der Algorithmus von Felix Klesen liefert bereits eine hierarchische und gruppierte Positionierung der Knoten, welche hier als gegeben

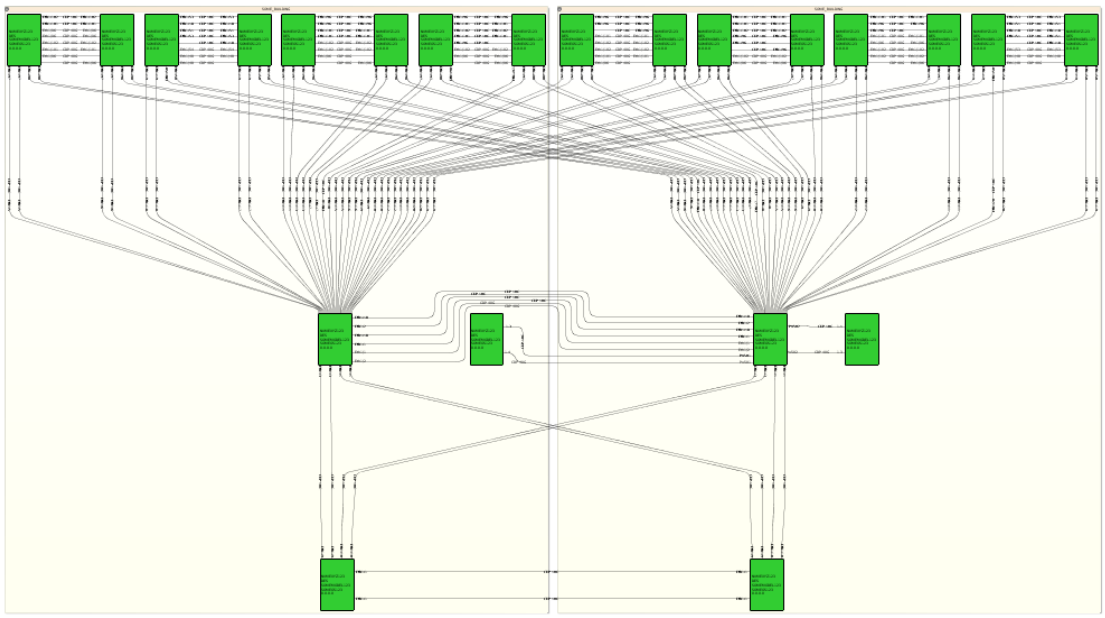


Abbildung 1: Zeichnung des Algorithmus von Felix Klesen

angenommen wird. Auch die Beschriftung von Kanten innerhalb desselben Layers funktioniert mit seinem Algorithmus auch unter Berücksichtigung von Knotengruppierung noch gut. Deswegen wird die Beschriftung von Kanten innerhalb desselben Layers hier nicht betrachtet, sondern ebenfalls als gegeben angenommen. Um diese übersichtliche Beschriftung jedoch nicht nachträglich durch unseren Algorithmus wieder durcheinander zu bringen, erlauben wir unserem Algorithmus hier von Anfang an keine weiteren Umsortierungen von Knoten - zumal deren Umsortierung aufgrund derer Gruppierungen ohnehin nur eingeschränkt möglich wäre. Stattdessen machen wir keine weiteren Annahmen an die Positionierung der Knoten, um diesem initialen Layoutprozess alle nötigen Positionierungen für eine bestmögliche Beschriftung innerhalb desselben Layers zu ermöglichen. Der Rahmen dieser Arbeit beschränkt sich also auf das nachträgliche Zeichnen und Beschriften von Kanten zwischen bereits positionierten Knoten. Die Reihenfolge der Knoten kann also nicht mehr geändert werden, lediglich deren Größe und deren relative Distanz (d.h. Strecken oder Stauchen eines Subgraphen). Ziel ist es, ausgehend von einer bereits errechneten Knotenpositionierung eine Zeichnung zu erstellen, in der die Kanten zwischen den Layern „bestmöglich“ beschriftet sind. Was hier unter „bestmöglich“ verstanden wird, soll später in Abschnitt 1.2 thematisiert werden. Zuvor wollen wir noch die Ausgangssituation formulieren.

In dieser Arbeit werden die Netzwerkdiagramme immer als Graphen modelliert, in denen jedes Gerät einem Knoten und jede Verbindung einer Kante entspricht. Richtungen oder Multikanten werden nicht berücksichtigt.

Außerdem schränken wir die Typen von Graphen, für die der Algorithmus funktionieren soll, noch zusätzlich auf den Anwendungsfall ein: Die für diese Arbeit ausschlag-

gebenden Netzwerkpläne sind immer in drei Layer partitioniert, wobei nur ein einziger Layer mit beiden anderen verbunden ist. Da wir nur die Kanten zwischen den Layern betrachten, handelt es sich hier also um 3-partite Graphen, in denen zwei der Partitionen nicht miteinander verbunden sind.

Aus Gründen der Lesbarkeit legen wir zu Beginn fest, dass die Layer vertikal angeordnet werden (aber eine horizontale Anordnung hätte natürlich ebenso funktioniert). In jedem Layer der gegebenen Positionierung sollen die Knoten auf derselben horizontalen Gerade liegen. Wir beschränken uns außerdem auf den Fall, dass alle Label rechteckig sind und dieselbe Fläche $l \cdot h$ haben, wobei l die Länge und h die Höhe bezeichnet. Zudem sollen sie einheitlich an den Kanten platziert werden, o.B.d.A. ohne Versatz[1] genau auf der Kante liegend, und jede Kante hat ein Label.

1.2 Erwartungen

Ein naheliegendes Ziel für Netzwerkdiagramme ist deren Übersichtlichkeit. Idealerweise sollten die Beschriftungen innerhalb eines Planes „einheitlich“ platziert sein, damit klar erkennbar ist, auf welches Objekt sich das Label bezieht.

Die resultierenden Pläne sollen als standardisierte technische Zeichnungen für ein großes Unternehmen dienen. Es werden daher viele verschiedene Pläne erstellt, welche aber jeweils nur selten in Augenschein genommen werden. Daher ist es in dem hier betrachteten Fall nicht nur erforderlich, dass jeder einzelne Plan in sich selbst möglichst einheitlich strukturiert ist - die Zeichnungen sollten hingegen auch möglichst ähnlich *zueinander* sein, damit wenig Zeit für die Einarbeitung nötig ist.

Abgesehen von der Einheitlichkeit ist auch eine möglichst gute Lesbarkeit relevant. In unserem konkreten Anwendungsfall ist zu bedenken, dass die Pläne auf Papier ausgedruckt werden, statt auf Bildschirmen angezeigt zu werden. Unsere Zeichnung muss also später hinreichend skaliert werden, damit sie auf das gewünschte Papierformat gedruckt werden kann. Die Label werden dadurch um den entsprechenden Faktor kleiner, und nachträgliches hineinzoomen ist dann unmöglich. Wir müssen daher besondere Priorität auf die Lesbarkeit der Label setzen.

Zusammengefasst ergeben sich die noch nicht formalen Anforderungen

A Einheitliches Layout innerhalb einer Zeichnung

B Einheitliche Struktur zwischen je zwei Plänen

C Bestmögliche Lesbarkeit

Wie diese Erwartungen in dieser Arbeit interpretiert wurden, soll in 2.1 erläutert werden. Dazu formulieren wir die konkreten Ziele, die wir entsprechend unserer Interpretation gesetzt haben, und anhand derer wir die Resultate beurteilen.

[1] für einen Versatz können die Label nachträglich verschoben werden

2 Umsetzung

2.1 Konkrete Zielsetzung

Wir möchten die intuitiven Wünsche aus Abschnitt 1 nun in konkrete Zielsetzungen überführen. Die Anforderungen bezüglich der Einheitlichkeit lassen sich jedoch nicht auf eindeutige Weise formalisieren, da es dafür erstmal einer klaren Definition von Einheitlichkeit bedarf. Wir haben uns für eine zu unserem Anwendungsfall passenden Interpretation entschieden. Im Folgenden werden daher gewisse Einschränkungen gewählt, die nur einer möglichen Interpretation davon gerecht werden, und nicht als absolut notwendig aufzufassen sind. Vielmehr sind sie recht spezifisch auf den Anwendungsfall zugeschnitten, selbst wenn hier versucht wurde, sie möglichst allgemein zu halten. In diesem Abschnitt formulieren wir die ihr entsprechenden Rahmenbedingungen.

Neben der eingangs erklärten Einschränkung, dass Knoten nicht mehr umpositioniert werden dürfen, wurden die folgenden Einschränkungen gewählt, um A zu gewährleisten:

A.1 Alle Kanten müssen orthogonal verlaufen.

A.2 Zwischen je zwei verbundenen Layern soll der erste Abschnitt jeder Kante immer in Richtung des Ziellayers verlaufen.

A.3 Zwischen je zwei verbundenen Layern sollen alle Label auf der exakt selben Höhe liegen.

A.4 Alle Knoten sollen dieselbe Größe haben, da sonst der unbeabsichtigte Eindruck entstehen könnte, gewisse Elemente wären von höherer Wichtigkeit.

A.5 Innerhalb eines Layers sollen alle Knoten auf derselben horizontal verlaufenden Achse liegen.

A.5 wird sich negativ auf die Lesbarkeit auswirken, sobald ein Layer zu viele Knoten hat - der Graph wird ggf. unnötig breit. Wir werden im Abschnitt 5.3 quantifizieren, wann dieser Effekt genau eintritt. In den Netzwerken des Anwendungsfalls dieser Arbeit ist dies jedoch nie der Fall, weswegen diese Forderung hier vertretbar ist.

Als nächstes möchten wir auch Punkt B so gut wie möglich präzisieren, auch wenn sich das als noch schwieriger gestaltet. Um die Struktur zwischen je zwei Plänen zusätzlich zu vereinheitlichen, lassen wir dem Algorithmus wenig Freiheiten. So legen wir fest

B.1 Das Label muss immer entweder am ersten oder letzten Abschnitt der Kanten sein.

B.2 Die Label sollen in allen Plänen vertikal ausgerichtet sein, damit gegebenenfalls erkennbar ist, auf welchen der Knoten sich eine Texthälfte bezieht.

B.3 Die Reihenfolge der Layer muss in allen Plänen dieselbe sein.

Der wichtigste Faktor ist jedoch die Kantenführung. Da wir in Punkt A.5 bereits die Positionen der Knoten in jedem Layer fixiert haben, hat der Algorithmus nur hier

einen nennenswerten Spielraum. Diesen Freiraum für Individualität beschränken wir, indem wir strenge Regeln für mögliche Kantenverläufe setzen. Zunächst soll jede Kante in Bezug auf sowohl x-Achse als auch y-Achse monoton sein - also nie eine Kehrtwende machen. Für eine bestmögliche Übersichtlichkeit erfordern wir verschärfend, dass jede Kante höchstens drei Abschnitte (also zwei Knicke) hat. Dies verhindert all zu unübersichtliche Kantenverläufe, und in Kombination mit der Orthogonalität impliziert das insbesondere die Monotonie. Ferner erzwingen wir, dass die horizontal verlaufenden Abschnitte auf so wenig verschiedenen y-Koordinaten wie möglich verstreut sind, und dass die Pfade trotzdem klar unterscheidbar sind. All diese Effekte erreichen wir mit einer einzigen hinreichenden Forderung:

B.4 Die Kanten sollen entsprechend Abschnitt 2.4 gebündelt werden.

Die Ziele A und B haben uns zu einer Reihe an Einschränkungen motiviert. Ziel C wird nun im Gegenzug die zu optimierenden Eigenschaften bestimmen.

Wir fordern hier bewusst nicht explizit, die Anzahl an Überschneidungen zu minimieren - wie wir später in Abschnitt 2.4 zeigen werden, sind nämlich nicht alle Überschneidungen zwangsläufig unübersichtlich. Verzichten wir hingegen auf diese Forderung, und nehmen gewisse Überschneidungen in Kauf, eröffnet sich durch die Kantengruppierung ein Weg, die benötigte Fläche stark zu reduzieren und zusätzlich die Übersichtlichkeit und Einheitlichkeit zu verbessern. Abschnitt 3.2 wird dann zeigen, dass die Anzahl an Überschneidungen dadurch trotzdem nur bedingt erhöht wird. Stattdessen widmen wir uns also hier nur der Lesbarkeit der Label.

Die Lesbarkeit der Label ist erfrischenderweise recht leicht zu formalisieren. Die Label werden zunächst als Rechtecke festgesetzter Größe platziert, die erst nach Fertigstellung der Zeichnung im Zuge der Skalierung verkleinert werden. Je weniger die Zeichnung skaliert werden muss, desto größer sind die Label schließlich darauf abgebildet. Die Lesbarkeit ist somit genau durch den Skalierungsfaktor bestimmt, welcher wiederum allein von den Maßen der resultierenden Zeichnung abhängt.

Doch genügt es hier streng genommen noch nicht direkt, einfach eine möglichst geringe Gesamtfläche zu fordern. Falls nämlich zusätzlich ein gewisses Seitenverhältnis eingehalten werden muss, um z.B. eine DinA4 Seite zu bedrucken, kann bereits eine zu lange Seite entscheidend sein, wie Abb. 2 zeigt. Der schwarze Rahmen entspricht dem Seitenverhältnis einer DinA4 Seite. Die Zeichnung nutzt die Fläche nicht gut, da sie zu breit wurde, und ist aufgrund der daraus resultierenden Skalierung kleiner als es nötig wäre.

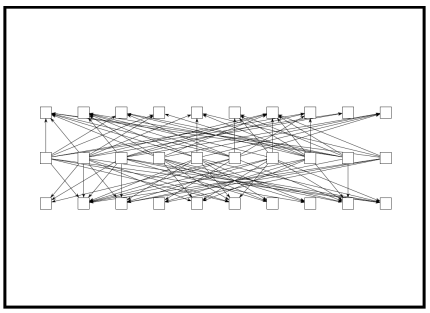


Abbildung 2: Zu breite Zeichnung

Um die Zielsetzungen nicht noch komplizierter zu gestalten, bewerten wir die Resultate dieser Arbeit anhand ihrer Befähigung, die Fläche zu minimieren. In Abschnitt 5.1 formulieren wir jedoch eine mögliche Zielsetzung, die zusätzlich auf das Seitenverhältnis

Bezug nimmt. Anschließend wird untersucht, in welchen Situationen unser Algorithmus unter den genannten Einschränkungen auch entsprechend dieses Maßstabes gut abschneidet, und unter welchen Bedingungen er daran scheitert.

2.2 Konzept

Beim Beschriften von Graphen gibt es zwei Ansätze, das generische und das integrierte Labeling. Beim generischen Labeling wird der Graph zuerst ohne Label vollständig gezeichnet, und anschließend wird versucht, die Label optimal zu positionieren. Dieses Problem stellt sich auch bei orthogonaler Kantenführung als NP-vollständig heraus [Wolff00]. Beim integrierten Labeling hingegen wird bereits beim Positionieren der Knoten und Zeichnen der Kanten Rücksicht auf die Label genommen. In dieser Arbeit wurde ein Hybrid-Ansatz verfolgt: unseren Einfluss auf die Knotenpositionierung beschränken wir aus den in 1.1 genannten Gründen auf ausschließlich den Abstand der Layer, den horizontalen Abstand und die Breite der Knoten. Diese Änderungen können auch im Nachhinein mit wenig Aufwand durchgeführt werden, und werden die Beschriftungen innerhalb desselben Layers nicht beeinträchtigen. Davon abgesehen nehmen wir die restliche Knotenpositionierung als gegeben an. Beim Zeichnen der Kanten hingegen werden wir die Label im Voraus mit einplanen.

Wir geben nun einen groben Überblick über den Ablauf unseres Algorithmus. Betrachten wir dazu zunächst den Spezialfall, nur zwei miteinander verbundene Layer im Graph zu haben. Vorab lassen wir die Knoten positionieren, ohne dass hierbei Rücksicht auf Label und Kanten genommen werden muss. Sind die Knoten bereits platziert, stehen wir also vor der Aufgabe, die Kanten zu zeichnen und zu beschriften. Unser Vorgehen besteht hierbei aus zwei Komponenten.

Zuerst sollen die Label möglichst platzsparend positioniert werden. Dafür werden die Label parallel zueinander direkt am jeweils ersten (also in unserem Fall vertikalen) Abschnitt ihrer Kante platziert. In Abschnitt 2.3 wird dieser Vorgang detailliert betrachtet, und in Abschnitt 3.1 wird abgeschätzt, wie sich diese Strategie auf den Platzbedarf unserer Zeichnung auswirkt.

Die Label sind nun also platziert, jedoch haben wir bis jetzt noch jeweils nur den ersten Abschnitt der Kanten gezeichnet. Als zweites müssen wir uns also darum kümmern, die Kanten nun auch mit ihren Zielknoten im gegenüberliegenden Layer zu verbinden. Hier stellt sich die Aufgabe, einen möglichst übersichtlichen Kantenverlauf zu erzeugen, der trotzdem wenig zusätzlichen Platz einnimmt. Der Lösung dieser Aufgabe widmen wir uns in Abschnitt 2.4. Wir zeigen dort, wie dies durch Bündeln der Kanten an der Zielseite erreicht werden kann. Welche Resultate diese Strategie erzielt, analysieren wir schließlich in Abschnitt 3.2.

2.3 Labelpositionierung

Unser Ansatz ist es, alle Label parallel zueinander an derselben Seite zu platzieren. Dafür wählen wir eines der beiden Layer als das „beschriftete Layer“, und bezeichnen es

im Folgenden stets als B. Hierfür eignet sich dasjenige Layer, welches im Durchschnitt weniger ausgehende Kanten pro Knoten hat, damit entsprechend mehr Platz für Label unter den Knoten bereitsteht. Dass genau diese Wahl von B auch aus anderen Gründen besonders vorteilhaft ist, wird in Abschnitt 3.2 festgestellt.

Haben wir nun B ermittelt, zeichnen wir zunächst von diesem Layer ausgehend den ersten Abschnitt jeder Kante direkt in Richtung des anderen Layers. Der Abschnitt soll lang genug sein, dass genügend Platz für das Label vorhanden ist, also eine Länge von l (der eingangs festgelegten Breite der Label) haben, siehe Abb. 3. Dieser Abschnitt wird später noch verlängert.

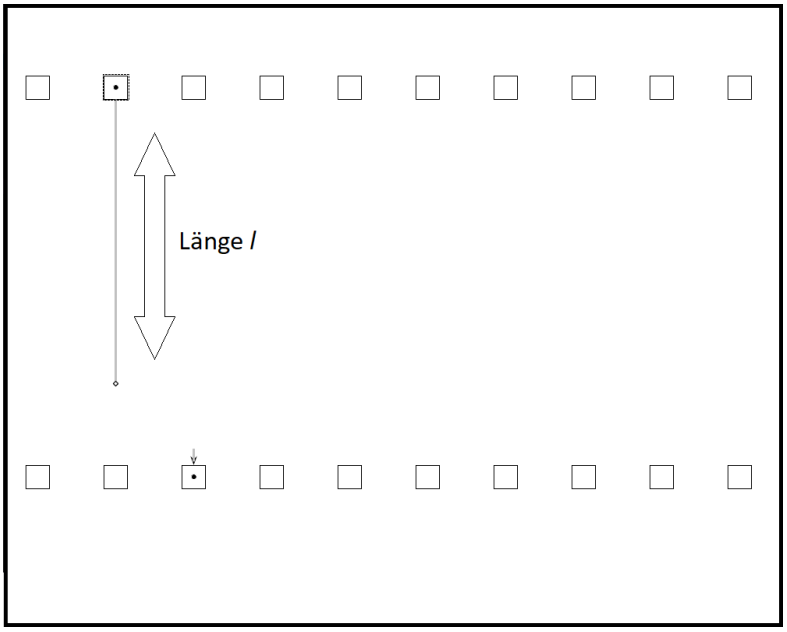


Abbildung 3: Vorläufiges Zeichnen des ersten Abschnittes einer Kante

Nachdem wir nun das erste Teilstück gezeichnet haben, zeichnen wir den Rest der Kante zur anderen Seite. Darum kümmern wir uns im folgenden Abschnitt.

2.4 Kantengruppierung

Da wir etwaige Richtungen der Kanten nicht berücksichtigen, nehmen wir o.B.d.A. in diesem und allen folgenden Abschnitten den Knoten aus der Beschriftungsseite als den Quellknoten, den gegenüberliegenden als den Zielknoten an. Für je zwei miteinander verbundene Layer bezeichnen wir die Knotenmenge der Beschriftungsseite immer mit B, und die der Zielseite mit Z. Ferner bezeichnet E die Menge aller Kanten, die zwischen B und Z verlaufen. Aus Gründen der Lesbarkeit nehmen wir im Rest der Arbeit außerdem an, dass das beschriftete Layer B oben, und die Zielseite Z unten ist.

Nach dem beschrifteten Abschnitt muss die Kante nun zu ihrem Knoten auf der gegenüberliegenden Seite verlaufen. Da bereits viel Platz für den beschrifteten Abschnitt

benötigt wurde, sollen die restlichen Abschnitte möglichst wenig zusätzliche Höhe in Anspruch nehmen. In den zugrundeliegenden Ausgangsplänen waren die beiden Layer üblicherweise beinahe vollständige bipartite Graphen. Dadurch könnten die restlichen Abschnitte schnell unübersichtlich werden, wenn wir sie in einen zu flachen Bereich zwingen. Dies versuchen wir durch Kantengruppierung zu lösen.

Statt zu versuchen, die Anzahl der Kreuzungen zu minimieren, wurde hier ein Ansatz verfolgt, um die Anzahl der *Kanten* drastisch zu reduzieren. Wir gruppieren hierbei alle Kanten aus E, die zum selben Knoten in Z inzident sind, und fassen sie nach dem beschrifteten Abschnitt zu einer einzigen Kante zusammen. Dadurch ersparen wir es uns, jede einzeln an Z anzubinden, ohne den im Plan abgebildeten Informationsgehalt zu mindern. Die Bündelung soll jeweils durch eine Vertikal verlaufende Kante umgesetzt werden, welche dann an alle in ihr gebündelten Kanten angeschlossen wird, siehe Abb. 4.

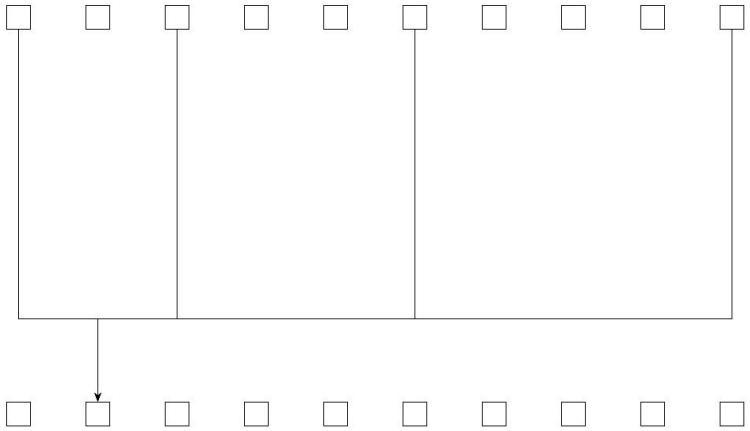


Abbildung 4: Vertikale Bündelung von Kanten zum selben Zielknoten

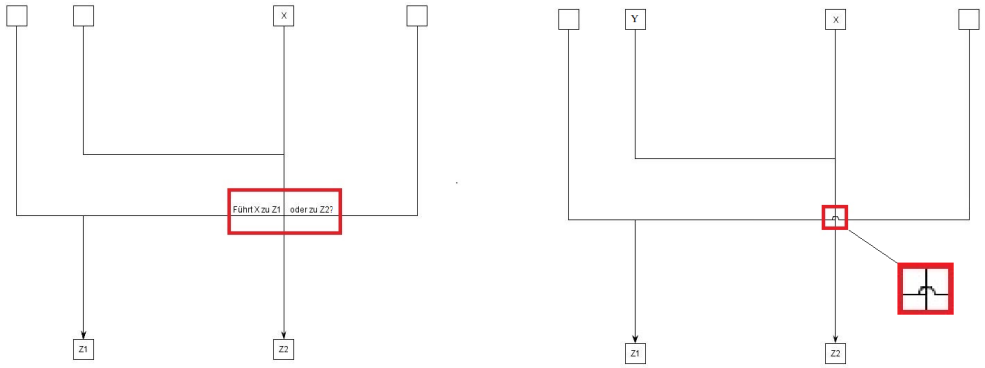
Die gebündelten vertikalen Abschnitte müssen anschließend nur noch durch genau eine Kante vertikal an ihren Zielknoten angeschlossen werden. Damit man verschiedene Kantengruppen voneinander unterscheiden kann, dürfen sie nicht direkt aufeinander liegen. Wir platzieren sie daher mit einem vertikalen Abstand d zueinander. Prinzipiell kann d beliebig klein gewählt werden. Da wir bis jetzt keine absoluten Distanzwerte festgelegt haben, setzen wir d beispielhaft auf h, was eine realistische Flächenabschätzung ermöglicht. Denn weil h hinreichend groß ist, um damit den Text der Label darzustellen, wird es auch als Abstand genügen, um die Kanten klar voneinander zu unterscheiden.

Wir legen also eine (zunächst beliebige) Reihenfolge fest, und platzieren die gebündelten horizontalen Abschnitte entsprechend äquidistant unter ihren jeweiligen beschrifteten Abschnitten. Diese Reihenfolge wird einen Einfluss auf die Anzahl an Kreuzungen haben - das wird später in Abschnitt 3.2 relevant werden. Fürs Erste wollen wir aber nicht näher darauf eingehen, da das für das grundlegende Konzept nicht erheblich ist.

Die Start- und Zielknoten können nun für jedes Bündel jeweils klar identifiziert werden, denn jeder vertikale Abschnitt von oder zu einem Bündel bestimmt durch seine

x-Koordinate eindeutig den angebundenen Knoten. Überschneidungen sind meist eindeutig von Richtungswechseln zu unterscheiden, da jede Kante jeweils nur in ihrer horizontalen Bündelungslinie die Richtung ändert. Die einzige Ausnahme dafür entsteht, wenn ein Knoten aus B genau über einem Knoten aus Z liegt, und er auf dem Weg zu seiner Bündelungslinie vorher eine andere schneidet. Dieser Fall wird in Abbildung 5(a) illustriert.

Hier kann unter Umständen nicht mehr klar unterschieden werden, zu welcher Bündelungslinie er zugehört, und auch die Zuordnung der Bündelungslinien zu ihren Zielknoten kann dadurch zweideutig werden. Daher lassen wir alle Überschneidungen in unserer Zeichnung durch kleine „Brücken“ hervorheben, denn auf diesem Weg bleiben alle Kantenverläufe visuell unterscheidbar.



(a) Unklare Bündelung (b) Eindeutige Bündelung

Abbildung 5: Der selbe Graph links ohne Brücken, rechts mit Brücke. Unter der Brücke erkennt man, dass eine Kante nach links abbiegt. Da jede Kante nur zwei Richtungswechsel hat, kann diese Kante also nur von X kommen, nicht von X's Nachbar Y.

2.5 Zusammenfassung

Um die Knoten aus B und Z zu positionieren, gehen wir also wie folgt vor: Wir bestimmen, wie lang das längste Label zwischen ihnen sein wird (l). Ausgehend davon erhalten wir vorläufig einen minimal nötigen Abstand zwischen den Layern. Dieser muss nun entsprechend der Kantengruppierungen nochmal vergrößert werden, um die Kanten nach dem beschrifteten Abschnitt zu Ende zu zeichnen. Dafür wird entsprechend 2.4 eine Höhe von genau $d \cdot |Z|$ benötigt (da genau $|Z|$ viele Kantengruppen benötigt werden). Zuletzt legen wir die gewünschte Länge des letzten Abschnittes fest, und erhalten insgesamt den Abstand der beiden Layer zueinander. Wir führen den gegebenen Algorithmus für hierarchische Gruppierung durch, und ändern in der daraus resultierenden Positionierung die Abstände der Layer und die Breiten der Knoten. Nachdem wir die vom Algorithmus vorläufig gezeichneten Kanten zwischen den Layern entfernen, können wir zum Abschluss die Schritte aus den Abschnitten 2.3 und 2.4 durchführen, und erhalten die fertige Zeichnung.

3 Analyse

3.1 Labelpositionierung

Aufgrund unserer Annahme, die Label seien gleich große Rechtecke der Dimension $l \cdot h$, benötigen wir zwangsläufig eine minimale Fläche von $|E| \cdot l \cdot h =: L_{min}$ zwischen den Layern, um alle Label überschneidungsfrei zu platzieren. Ziel ist es, dass die Zeichnung den ihr zur Verfügung gestellten Platz optimal nutzt, also möglichst wenig mehr Fläche benötigt als es diese untere Schranke erfordert. Bei fest gegebener Labelfläche L_{min} ist das Minimieren der Gesamtfläche äquivalent dazu, das Verhältnis von L_{min} zur Gesamtfläche zu maximieren. Wir wollen nun abschätzen, wieviel Platz für die gewählte Labelpositionierung benötigt wird. Für die Analyse ignorieren wir zunächst, ob die Label später aus optischen Gründen einen Mindestabstand zueinander haben sollen.

Da wir die Label parallel zueinander platzieren, und somit keine Fläche verschwendet wird, scheint es auf den ersten Blick, als würde der Platz optimal genutzt werden. Es verbirgt sich hier jedoch eine Problematik, die erst erkennbar wird, sobald sich in B Knoten mit unterschiedlichen Graden befinden.

Jeder Knoten $b \in B$ muss breit genug sein, damit alle von ihm ausgehenden Kanten an derselben Seite anliegen können, und dass diese trotzdem jeweils hinreichenden Abstand zur Nachbarkante haben, um überlappungsfrei beschriftet werden zu können. Daher muss seine Breite mindestens $\deg(b) \cdot h$ betragen. Das Problem liegt nun darin, dass wir nicht nur einen einzelnen Knoten breiter zeichnen dürfen, wenn dieser mehr ausgehende Kanten hat. Stattdessen haben wir in A.4 gefordert, dass alle Knoten die selbe Breite haben – in diesem Fall müssten also auch alle anderen Knoten breiter werden. Es mag scheinen, als wäre diese Forderung unnötig streng - schließlich könnte man den durchaus berechtigten Einwand erbringen, ein Knoten mit deutlich mehr Kanten als seine Nachbarn dürfe auch deutlich wichtiger aussehen. Der Grund, aus dem die einheitliche Breite trotzdem gefordert wurde, ist der, dass es sich jeweils nur um Teilmengen des vollständigen Netzes handelt. Es könnte also durchaus sein, dass die anderen Knoten, die hier einen kleineren Grad haben, eigentlich mindestens genauso gut angebunden sind - in diesem Plan jedoch nur die Anbindungen zu ganz gewissen anderen Knoten betrachtet wurden. In einem anderen Teilnetz könnte es sich genau gegenteilig verhalten, was die Einheitlichkeit zwischen den Plänen beeinträchtigen würde. Für Anwendungsfälle wie diesen gilt es also nach Möglichkeit zu vermeiden, mit einer variierenden Knotenbreite gewisse Implikationen zu übermitteln. Um die Entscheidung zu erleichtern, ob die unnötige Breite in Kauf genommen wird, analysieren wir zunächst, wann und wie sehr dieser Schwachpunkt ins Gewicht fällt.

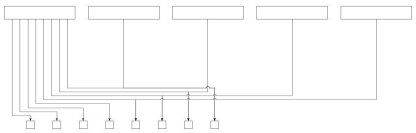


Abbildung 6: Unnötig breite Zeichnung

Alle Label müssen auf derselben y-Koordinate liegen, in jedem Fall wird also eine Gesamtbreite von mindestens $|E| \cdot h$ nötig sein. Wenn alle Knoten die selbe Größe haben

sollen, benötigt jeder davon eine Breite von jeweils $\Delta \cdot h$, wobei Δ den Maximalgrad $\max_{b \in B}(\deg(b))$ bezeichne. Bringen wir mit diesen Werten die mindestens nötige Breite mit der tatsächlich benötigten Fläche in Verbindung, erhalten wir die Ungleichung

$$|E| \cdot h = \sum_{b \in B} \deg(b) \cdot h \leq \sum_{b \in B} \Delta \cdot h = |B| \cdot \Delta \cdot h$$

Von ihr ausgehend können wir nun exakt abschätzen, welchen Faktor c wir zusätzlich benötigen:

$$\begin{aligned} & |B| \cdot \Delta \cdot h \stackrel{!}{=} |E| \cdot h \cdot c \\ \iff \quad & \frac{1}{c} \stackrel{!}{=} \frac{1}{|B| \cdot \Delta} \cdot |E| = \frac{1}{|B| \cdot \Delta} \sum_{b \in B} \deg(b) \\ \iff \quad & \frac{1}{c} \stackrel{!}{=} \frac{1}{|B|} \cdot \sum_{b \in B} \frac{\deg(b)}{\Delta} \end{aligned}$$

Das heißt, der durchschnittliche Wert von $\frac{\deg(b)}{\Delta}$, also die durchschnittliche Abweichung vom Maximalgrad, bestimmt, wieviel mehr Platz unser Ansatz erfordert. Falls der Grad der Knoten einheitlich ist, erhalten wir sogar eine Approximationsgüte von 1. Sollten jedoch Graphen mit stark unterschiedlichen Knotengraden beschriftet werden, stellt dies tatsächlich einen Schwachpunkt dar. Eine mögliche Lösung, die immer eine Approximationsgüte von 1 erhält, und dabei trotzdem Punkt A.4 einhält, behandeln wir in Abschnitt 5.2. In den dieser Arbeit zugrundeliegenden Plänen sind die Knotengrade der beschrifteten Layer fast immer gleich, da es sich um doppelt redundante Anbindungen handelt. Daher stellt dies hier kein Problem dar, und wir erhalten bisher eine bestmögliche Platznutzung.

Doch haben wir bisher nur die resultierende Breite behandelt. Zwar wurde für die Labelpositionierung auch noch keine zusätzliche Höhe in Anspruch genommen, doch ist das wenig erstaunlich - schließlich musste sich hierfür auch noch nicht um die eigentlichen Kantenverläufe gekümmert werden. Sie allein beeinflussen die resultierende Höhe, und daher widmen wir uns ihnen nun im folgenden Abschnitt.

3.2 Kantengruppierung

Wir analysieren nun, wie sich die Kantengruppierung auf die resultierende Höhe und die Anzahl an Kreuzungen auswirkt.

Vor der Bündelung ist aufgrund unserer Wahl der Beschriftungsseite für ihre Knoten die durchschnittliche Anzahl ausgehender Kanten geringer $(*)$. Mit doppelter Abzählung

erhalten wir eine äquivalente, doch kürzere Formulierung:

$$\frac{1}{|B|} \cdot \underbrace{\sum_{b \in B} |\{w \in E \mid \alpha(w) = b\}|}_{=|E|} = \frac{1}{|B|} \cdot \sum_{b \in B} \deg(b)$$

$$\overset{(*)}{\leq} \frac{1}{|Z|} \cdot \sum_{z \in Z} \deg(z) = \frac{1}{|Z|} \cdot \underbrace{\sum_{z \in Z} |\{w \in E \mid \omega(w) = z\}|}_{=|E|}$$

$$\iff |B| \geq |Z| \tag{1}$$

wobei $\alpha(w)$ Quellknoten und $\omega(w)$ Zielknoten einer Kante $w \in E$ bezeichnen.

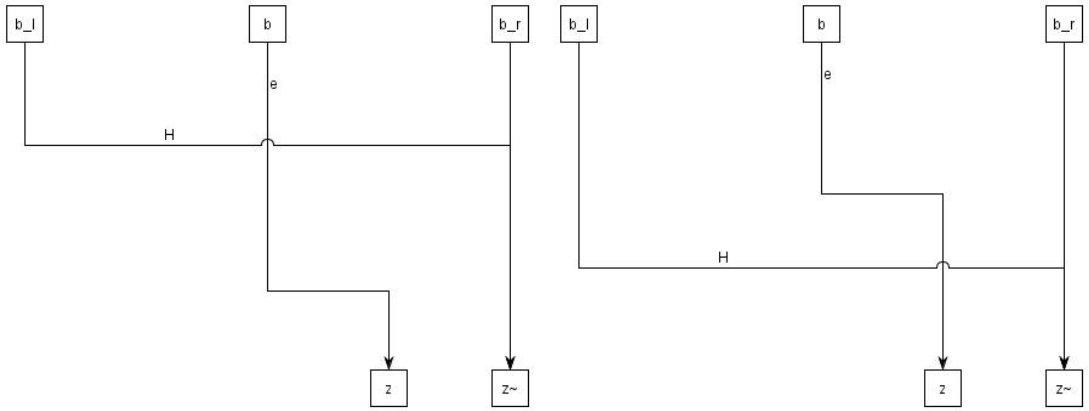
Das ist auch die effizientere Methode, den beschrifteten Layer B zu wählen. Ist wie in den zugrundeliegenden Ausgangsplänen jeder Knoten von B mit jedem Knoten von Z verbunden, so ist $|E| = |B| \cdot |Z|$. Nach der Gruppierung beläuft sich die Anzahl an Kanten, die im verbleibenden Abschnitt untergebracht werden müssen, nur noch auf $|Z|$ viele (da alle Kanten, die zu einem Knoten $|Z|$ verlaufen, zu einer Kante zusammengefasst werden). Der Mehrwert der Gruppierung eröffnet sich durch die folgende Gleichung:

$$|Z| = \sqrt{|Z| \cdot |Z|} \overset{(1)}{\leq} \sqrt{|B| \cdot |Z|} = \sqrt{|E|}$$

Wir erhalten daher im *Worst Case* eine Verbesserung auf $\sqrt{|E|}$ viele Kanten. Neben der Übersichtlichkeit hat das auch einen positiven Effekt auf unsere benötigte Höhe, denn sie hängt nur von der Anzahl Bündelungslinien ab. Und da zu jedem Zielknoten genau eine Bündelungslinie konstruiert wird, gibt es davon exakt $|Z|$ viele. Sollen die Bündelungslinien einen Abstand von d zueinander haben, benötigen wir also für die Kantenverläufe nur noch eine Höhe von maximal $d \cdot |Z| \leq d \cdot \sqrt{|E|}$ [2].

Betrachten wir nun den Effekt auf die Anzahl an Kreuzungen. Weil nur die Bündelungsabschnitte horizontal verlaufen, sind die einzig möglichen Kreuzungen solche eines vertikalen Abschnittes mit einer anderen Bündelungslinie. Wir betrachten nun jeweils, wann die vertikalen Abschnitte eine Bündelungslinie kreuzen. Seien $b \in B$ und $z \in Z$ durch eine Kante $e := (b, z)$ miteinander verbunden. e hat zwei Möglichkeiten, sich mit einer anderen Kante zu schneiden: entweder auf dem ersten Abschnitt, dem Weg von b zur eigenen Bündelungslinie (Abb. 7(a)), oder dem dritten Abschnitt, dem Weg von der eigenen Bündelungslinie zu z (Abb. 7(b)).

[2] In einigen Fällen wäre es sogar möglich, manche horizontale Abschnitte überschneidungsfrei nebeneinander zu legen, um noch weniger Höhe in Anspruch zu nehmen. Ein Beispiel findet sich in Abb. 8



(a) Schnitt auf dem ersten Abschnitt (b) Schnitt auf dem letzten Abschnitt

Abbildung 7: Die möglichen Schnitte einer Kante e mit einer Bündelungslinie H

Sei nun H eine andere Bündelungslinie, die von e geschnitten wird, und welche die Kanten zum Zielknoten $\tilde{z} \neq z$ bündelt. Damit die geschnittene Bündelungslinie überhaupt horizontal durch diesen Schnitt verläuft, muss sie selbst bereits Knoten links und rechts vom Schnitt aneinander anbinden.

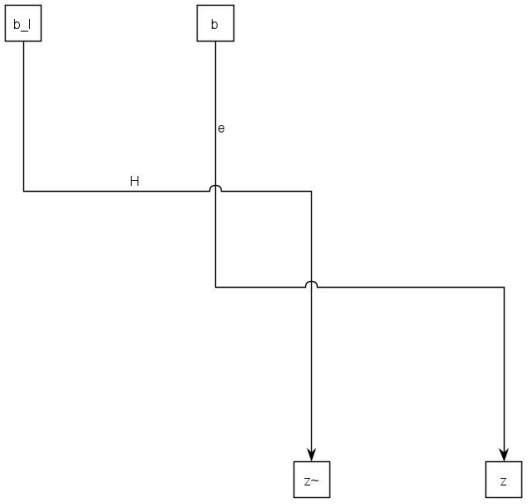
Wir behandeln zunächst den legitimen Fall, dass eine horizontal verlaufende Bündelungsline nur dann geschnitten wird, wenn sie sowohl links als auch rechts vom Schnitt zu je mindestens einem Knoten aus B inzident ist $(**)$.

Angenommen, der erste Abschnitt von e überschneidet sich mit H (Abb. 7(a)). Weil der schneidende Abschnitt vertikal verläuft, muss der Schnitt senkrecht unter b liegen. Nach $(**)$ existieren dann also Knoten b_l links von b und b_r rechts von b, die über die Bündelungslinie H an einen anderen Knoten $\tilde{z}$ angebunden waren. Falls z links von $\tilde{z}$ ist, hätten sich dann auch ohne die Kantengruppierung (b, z) und $(b_l, \tilde{z})$ schneiden müssen. Falls hingegen z rechts von $\tilde{z}$ ist, so wäre analog ein Schnitt von (b, z) und $(b_r, \tilde{z})$ unausweichlich. Das heißt, wenn solch ein Schnitt bei Zeichnung mit Kantengruppierung vorliegt, dann auch bereits ohne die Kantengruppierung. Wir erzeugen durch die Kantengruppierung also keine Schnitte dieser Art, die nicht ohnehin entstanden wären.

Betrachten wir die zweite Möglichkeit, also dass sich e mit H auf dem Weg von seiner Bündelungslinie zu z schneidet (Abb. 7(b)). Weil dieser Abschnitt vertikal verläuft, muss der Schnitt dann senkrecht über z liegen, H bindet also wieder nach $(**)$ Knoten b_l links von z und b_r rechts von z an $\tilde{z}$ an. Analog zum ersten Fall würde also auch hier bereits ohne Kantengruppierung zwangsläufig ein Schnitt entstehen.

Zunächst mag es scheinen, als wäre die Voraussetzung $(**)$ sowieso immer erfüllt – schließlich muss ein horizontal durch einen Punkt verlaufender Abschnitt einer monotonen Kante Knoten links und rechts davon anbinden. Leider scheitert dies an einem Detail: zwar muss die Bündelungslinie tatsächlich zu Knoten links und rechts vom Schnitt inzident sein – jedoch müssen nicht beide aus B kommen.

Es kann nämlich vorkommen, dass die geschnittene Bündelungslinie nur diese x-Koordinate passiert, um zum Zielknoten zu gelangen; nicht, um einen weiteren Startknoten auf der anderen Seite anzubinden. Ein einfaches Beispielszenario dafür wäre, wenn nur ein Startknoten $b_l \in B$ links von b durch H an $\tilde{z}$ angebunden ist, und $\tilde{z}$ links von z liegt. Ist nun die Bündelungslinie von e unter der Bündelungslinie von $(b_l, \tilde{z})$ platziert, so entstehen zwei Schnitte, die keinesfalls nötig wären. Und tatsächlich zeigten die von unserer Implementation erstellten Pläne diesen Effekt gelegentlich auf. Die Reihenfolge der Bündelungslinien spielt also durchaus eine Rolle bei der Beseitigung von Schnitten. Aber selbst wenn beide Knoten aus B kommen, können sie vom *selben* Knoten kommen - in dem Fall wäre der Schnitt auch nicht immer notwendig, da wir die ausgehenden Kanten eines Knotens beliebig umsortieren können. Also birgt nicht nur die vertikale Reihenfolge der Bündelungslinien potenziell unnötige Schnitte, sondern auch die horizontale Reihenfolge der Kanten ausgehend vom selben Knoten kann Schnitte verursachen, die nicht nötig gewesen wären. Beispiele dafür finden sich in Abb. 15. Wir müssen also sowohl eine ideale vertikale Reihenfolge der Bündelungslinien ermitteln, als auch eine bestmögliche horizontale Reihenfolge innerhalb jedes Knoten. Es scheint, als wäre dieses Problem schwer lösbar zu sein, insbesondere weil auch das in der Implementation gewählte Programm yFiles hieran scheitert.



3.3 Fläche

Fassen wir nun zusammen, wie die von unserer Zeichnung benötigte Fläche ausfällt. Wieder ignorieren wir optisch wünschenswerte Parameter, also Mindestabstände zwischen den Labeln und Mindestlängen der letzten Abschnitte (von den Bündelungslinien zu ihren Zielknoten). Abstände zwischen den Labeln können erzielt werden, indem sie direkt im Labeltext inkludiert werden, und Mindestlängen der letzten Abschnitte können durch nachträgliche Verschiebung erreicht werden. Seien L_1, L_2 und L_3 die Layer, L_1 mit L_2 durch die Kanten aus E_1 und L_2 mit L_3 durch die Kanten aus E_2 verbunden.

Berechnen wir dafür zuerst die nötige Breite und anschließend die Höhe. Die nötige Breite hängt nur von der Phase der Labelplatzierung ab. Wir gehen hier von einer Approximationsgüte von 1 aus, denn entweder, die Knotengrade sind schon einheitlich, oder wir können den Fix aus Abschnitt 5.2 anwenden. Dieser wird die nötige Höhe jedoch geringfügig vergrößern - da diese zusätzliche Höhe aber beliebig klein gewählt werden kann, ignorieren wir sie hier. Dadurch ergibt sich eine nötige Breite von $\max_{i \in \{1,2\}} \{|E_i| \cdot h\}$.

Nun zur nötigen Höhe. Unabhängig von dem spezifischen Graphen wird immer für jedes Layer die Höhe eines Knoten benötigt, welche wir h_V nennen, und für die zwei Verbindungen jeweils die Höhe der Label l. Die Höhe hängt ansonsten nur noch von den horizontalen Bündelungslinien ab. Sollen die horizontal verlaufenden Bündelungslinien einen vertikalen Abstand von d $(= h)$ zueinander haben, benötigen wir hierfür in der ganzen Zeichnung eine Höhe von insgesamt $\sqrt{|E|} \cdot h$. Wir erhalten damit eine Höhe von $3 \cdot h_V + 2 \cdot l + \sum_{i \in \{1,2\}} \sqrt{|E_i|} \cdot h$. Die insgesamt benötigte Fläche (ohne zusätzliche Mindestabstände) entspricht also

$$\left(\max_{i \in \{1,2\}} \{|E_i| \cdot h\} \right) \cdot \left(3 \cdot h_V + 2 \cdot l + \sum_{i \in \{1,2\}} \sqrt{|E_i|} \cdot h \right)$$

4 Implementierung

In der spezifischen Implementierung wurde die Knotenplatzierung durch das hierarchische Layouting vom Programm yFiles umgesetzt, welches dafür den Sugiyama Algorithmus einsetzt. Wir zeigen nun, welche Pläne der Algorithmus für gewisse Netzwerktopologien zeichnet.

In diesem Szenario soll eine fiktive Firma ein Firmennetz betreiben, an das Filialen in vielen verschiedenen Standorten angebunden sind. Die Geräte sind in drei Schichten unterteilt: Die Core-Devices sind die zentralen Geräte, jedes Gerät benötigt eine Anbindung an zumindest einen Core-Knoten. Damit nicht jedes Gerät eine Direktverbindung zu den Core-Devices benötigt und Filialen untereinander kommunizieren können, ohne dafür jedes Mal den Umweg über ein Core-Device gehen zu müssen, gibt es die sogenannten Metro-Devices. Sie fungieren als Vermittlungsstellen, an die schließlich die Filialen angebunden sind. Abbildung 14 zeigt das gesamte Netz. Wir gruppieren die Knoten nach den Ländern, in denen sich das Gerät befindet. Die folgenden Abbildungen zeigen Zeichnungen verschiedener Teilgraphen.

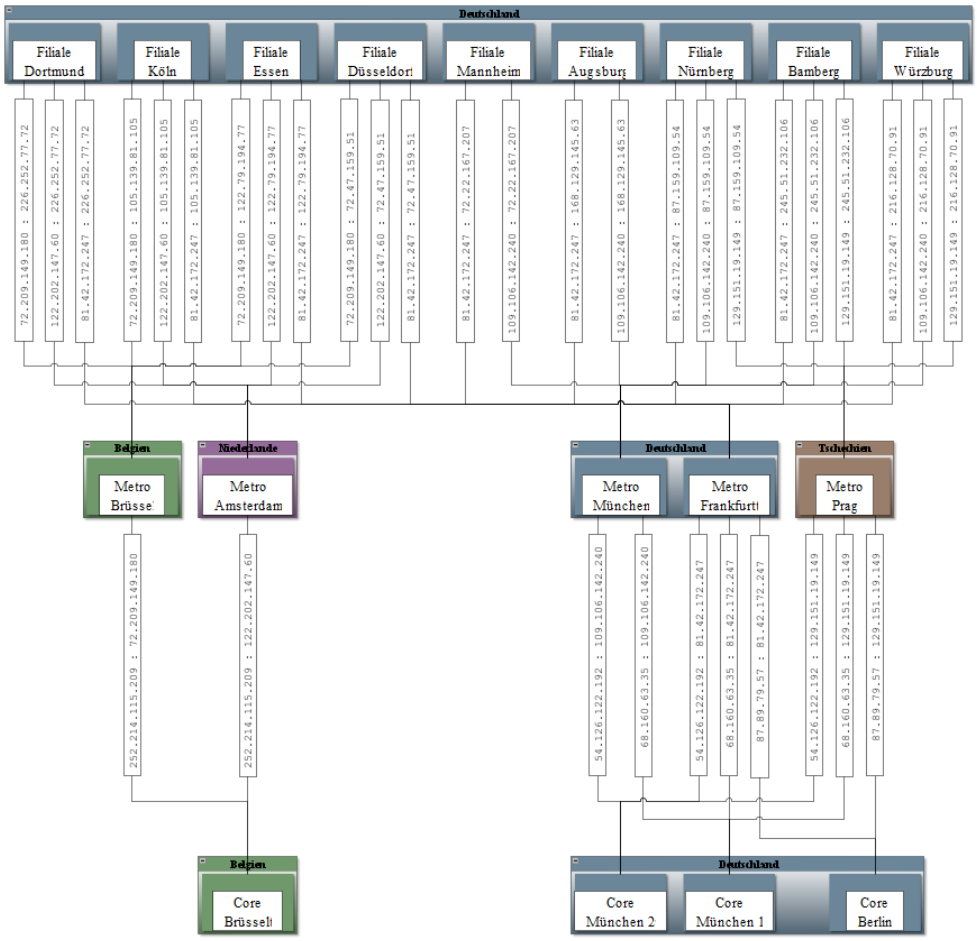


Abbildung 8: Trotz großer Kantenzahl bleibt der Graph überschaubar.

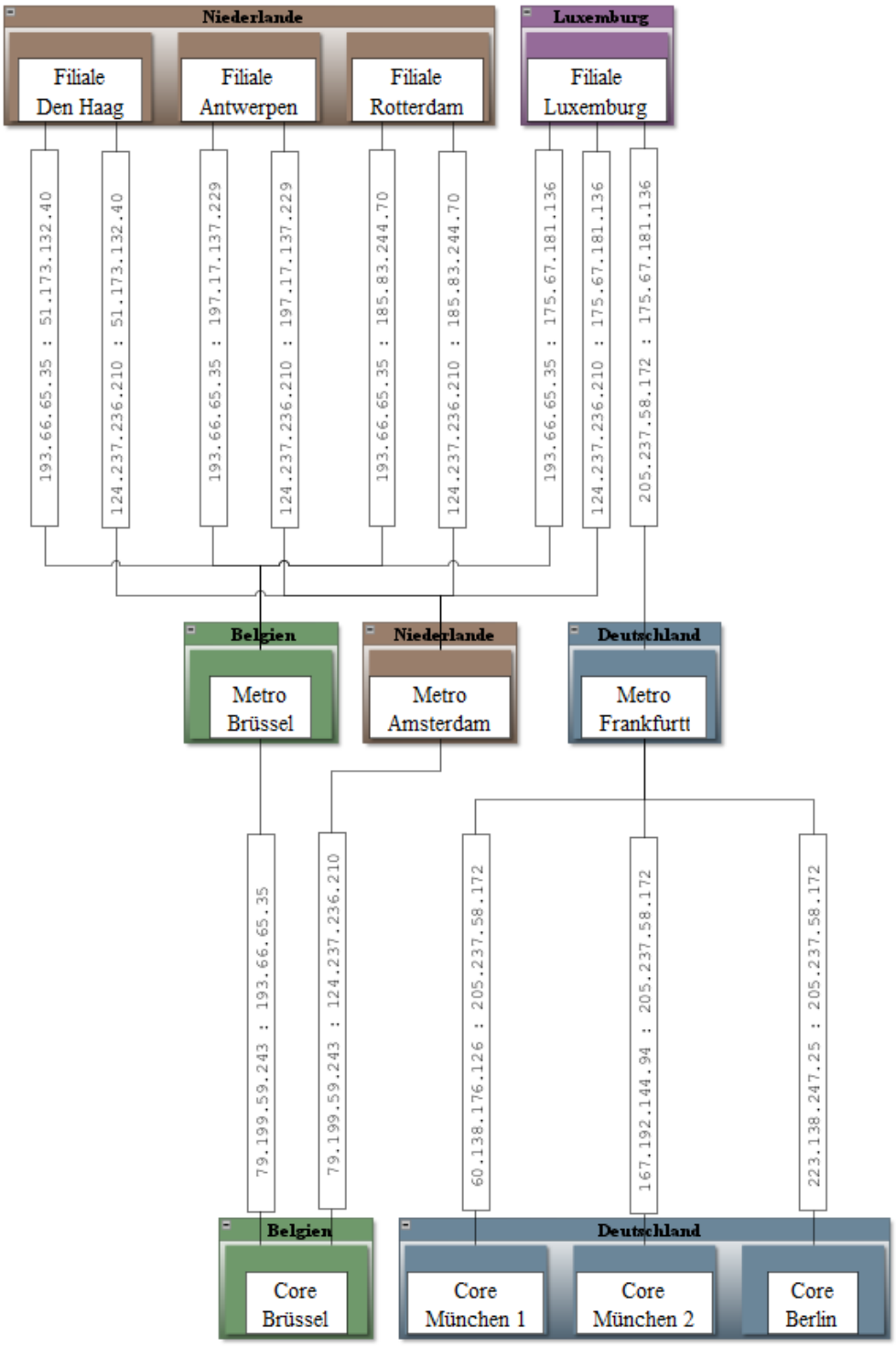


Abbildung 9: Filialen in BeNeLux Staaten

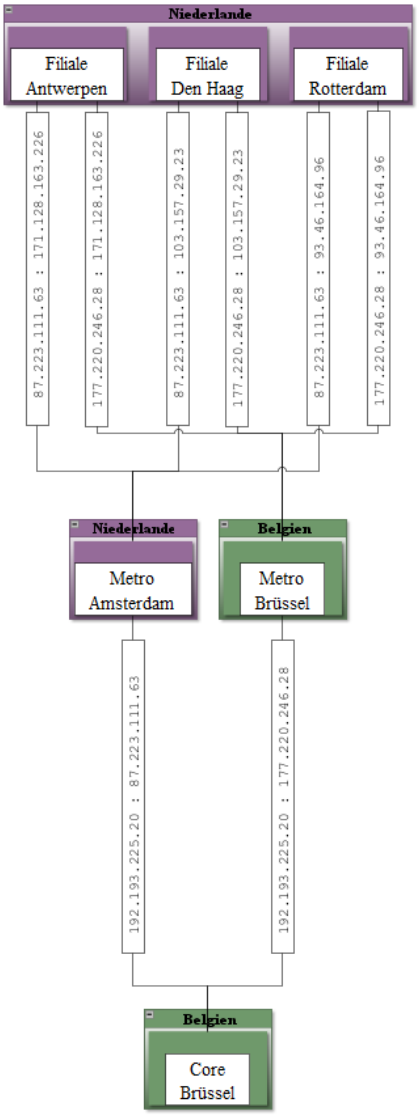


Abbildung 10: Niederländische Filialen

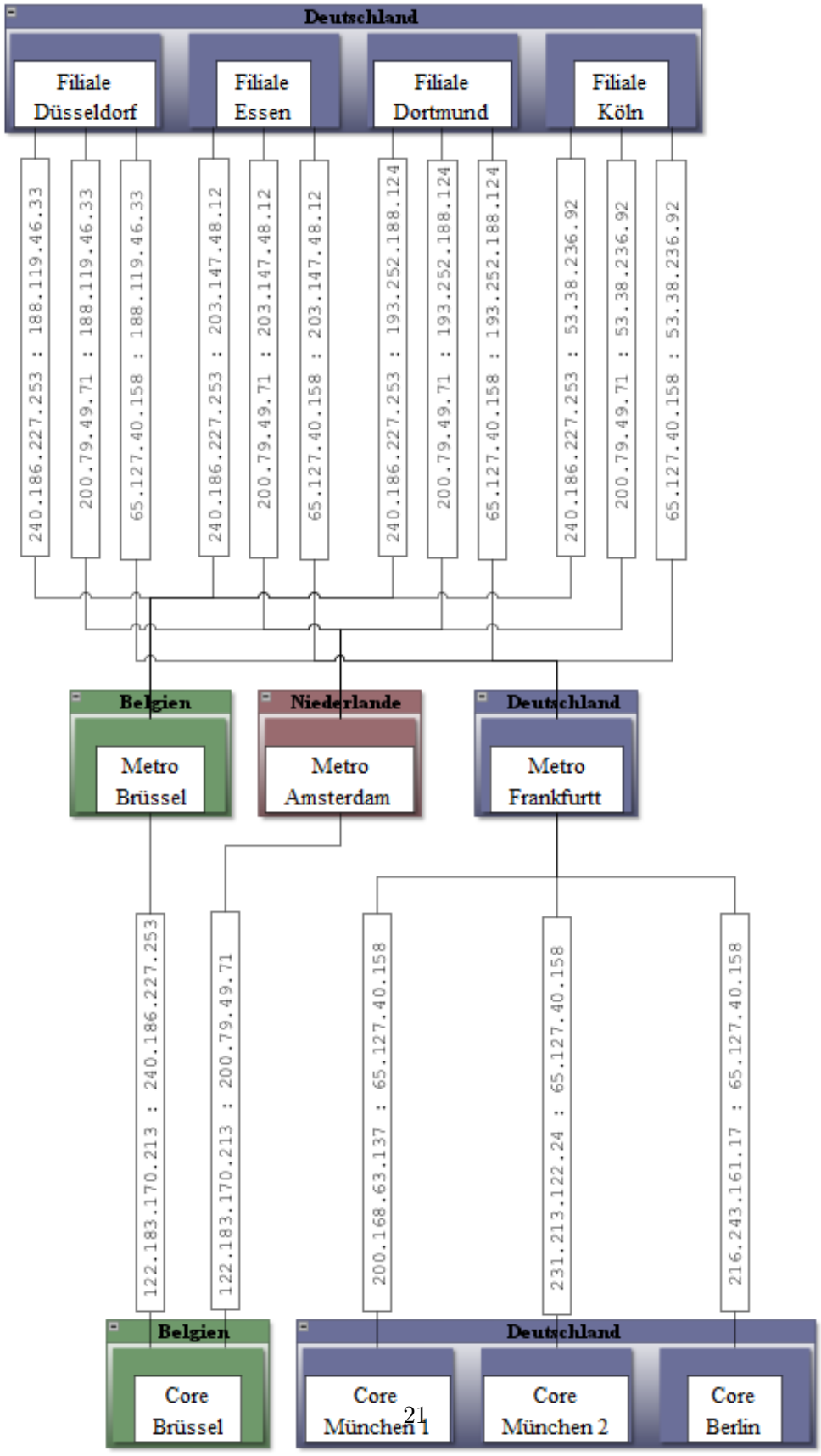


Abbildung 11: Filialen in Nordrhein-Westfalen

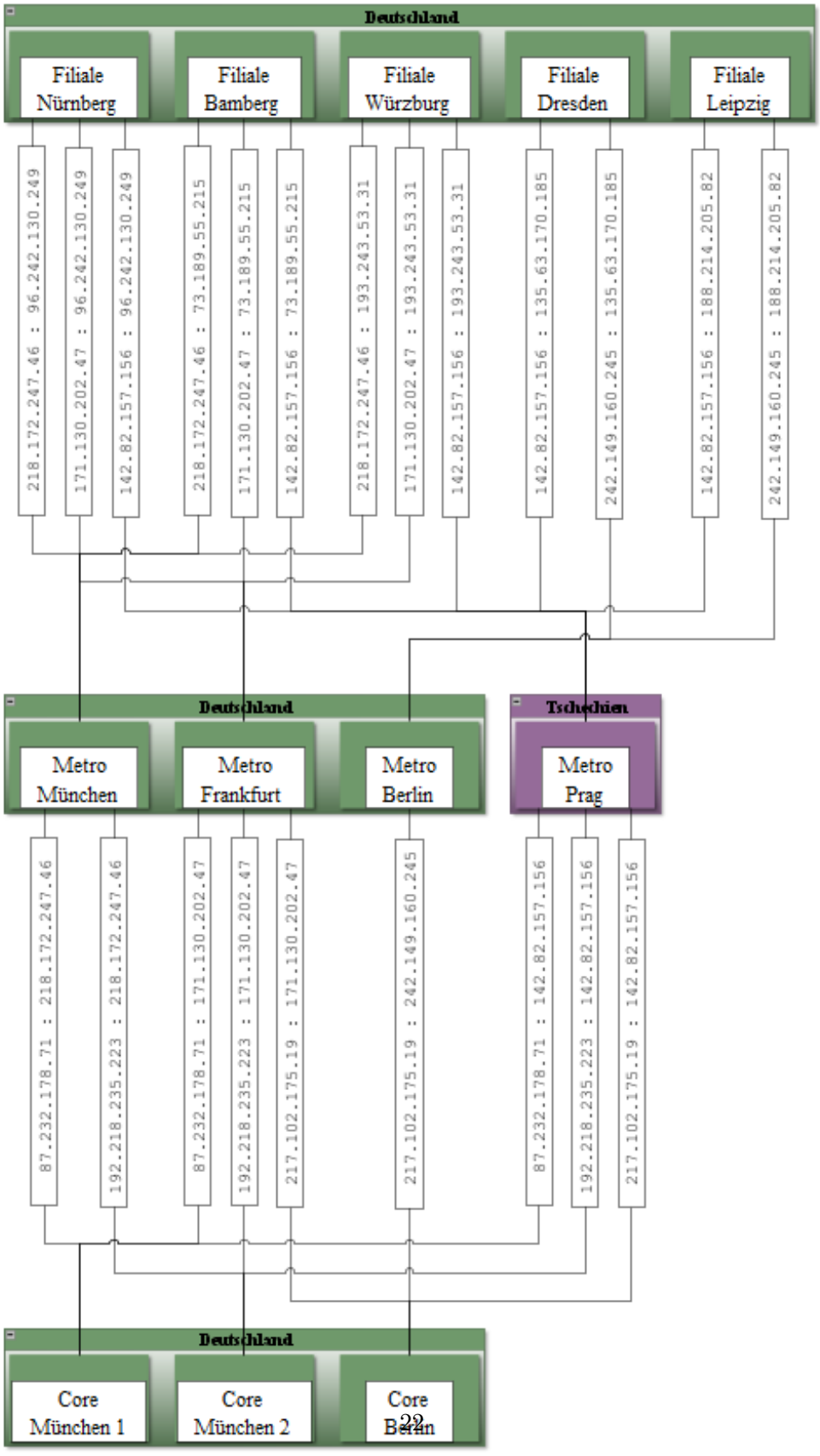


Abbildung 12: Filialen in Mittel- und Ostdeutschland, mit einem Metro-Device in Prag

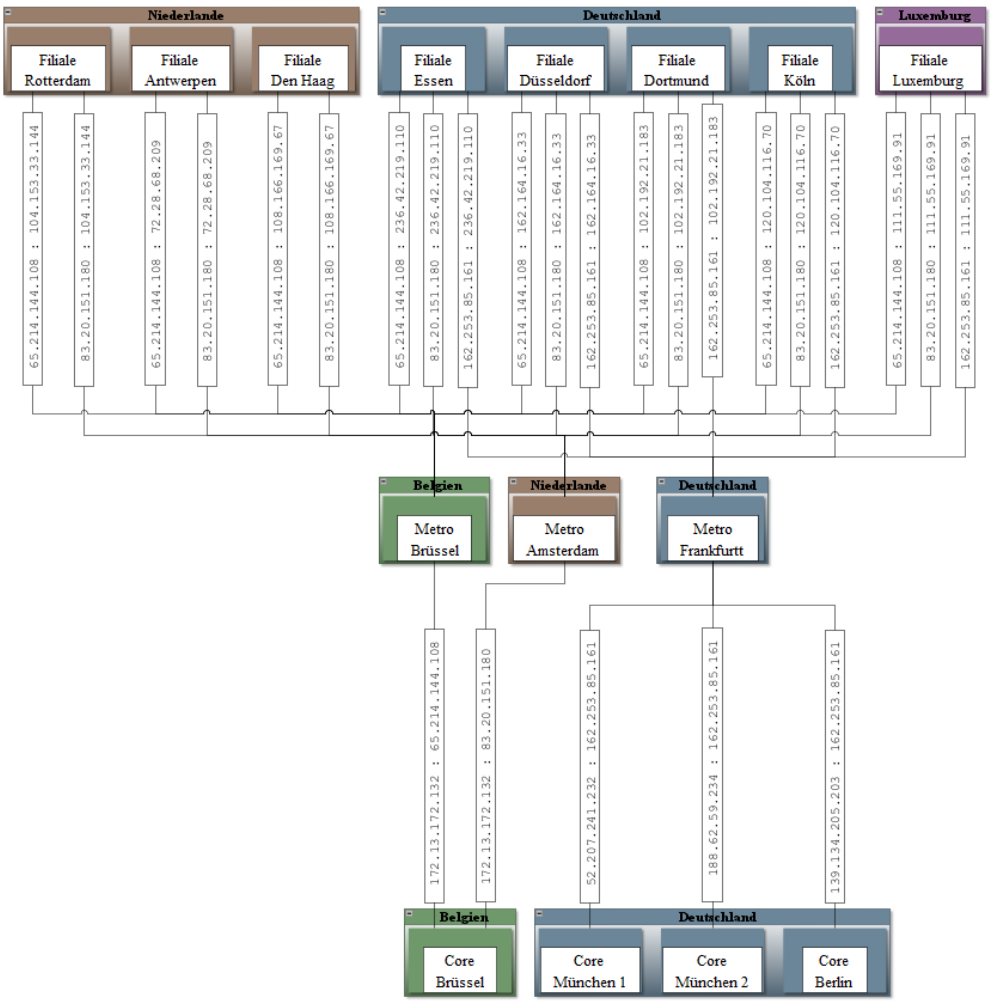


Abbildung 13: Westliche Filialen

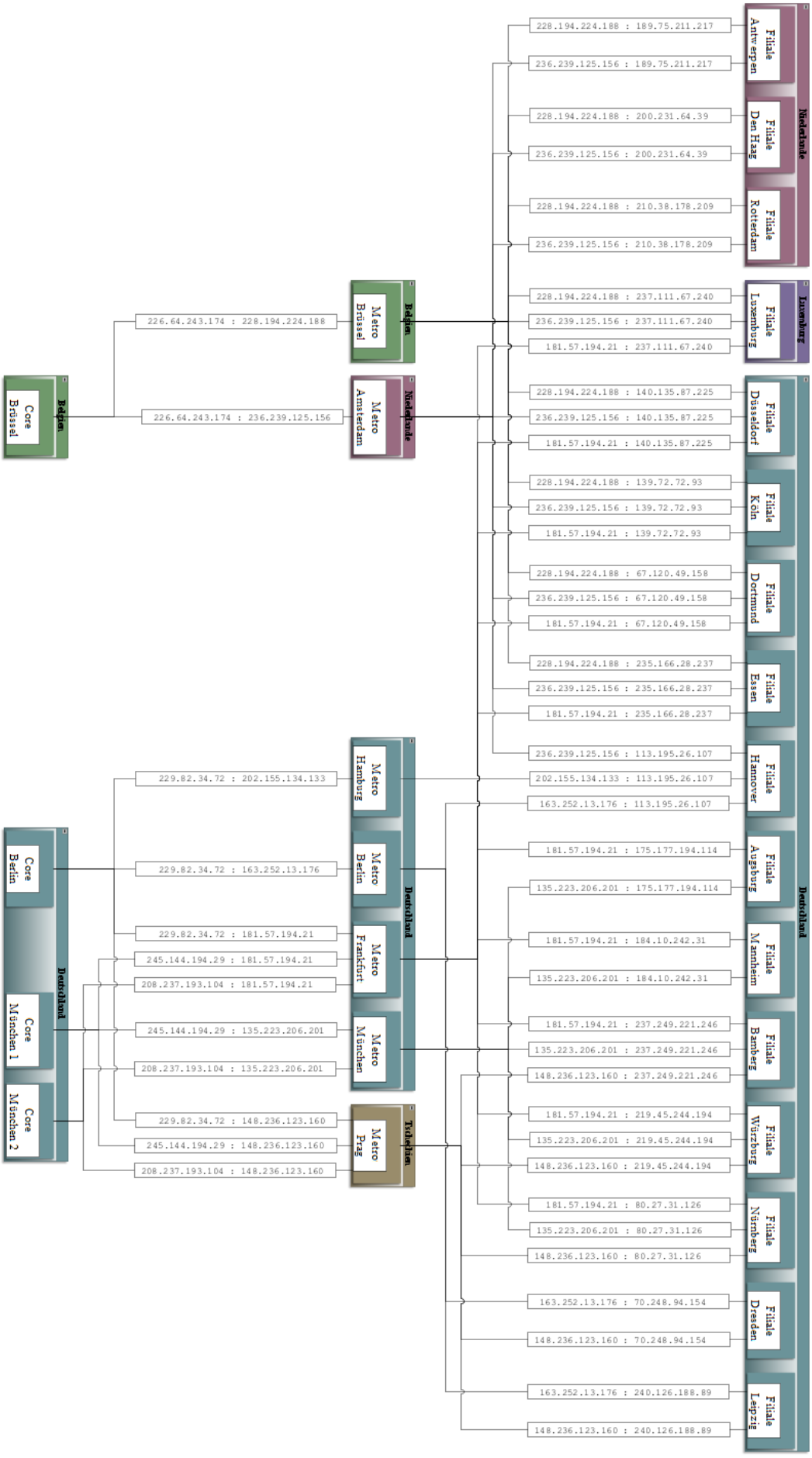


Abbildung 14: Das gesamte Netzwerk

5 Ausblick

5.1 Seitenverhältnis

Wir haben diese Arbeit nur im Bezug auf die Fläche analysiert. Doch haben wir bereits festgestellt, dass nicht nur die Fläche, sondern auch das Seitenverhältnis zu berücksichtigen ist. Wie diese Zielsetzung zu handhaben ist soll nun kurz betrachtet werden.

Bei der bisherigen Analyse wurden die Beschriftungen von Kanten innerhalb desselben Layers außer Acht gelassen. Das führt dazu, dass die resultierende Breite unserer Zeichnung noch nicht verlässlich ist, weil die Pläne dafür vermutlich noch etwas in die Breite wachsen müssen. Um sicherzustellen, dass möglichst viel Platz für solche Änderungen bereit steht, könnten die Pläne im Querformat gezeichnet werden, ohne dabei die gesamte Breite einzunehmen. Man erreicht dies beispielsweise, indem man zunächst ein möglichst quadratisches Seitenverhältnis als Ziel setzt. Die dazu äquivalente mathematische Zielsetzung wäre ein minimaler Umfang.

5.2 Uneinheitliche Knotengrade

In Abschnitt 3.1 wurde festgestellt, dass bei stark voneinander abweichenden Knotengraden im beschrifteten Layer eine unnötig breite Zeichnung entstehen wird.

Aber auch wenn solche Graphen beschriftet werden sollen, ließe sich dieses Problem durch eine leichte Abschwächung unserer Zielsetzung lösen. Das Problem liegt daran, dass die Kanten senkrecht aus dem Knoten führen müssen. Damit dann mehrere Kanten unter ihm liegen können, muss er also breit genug sein, damit sie alle aus ihm herausführen können. Wir könnten aber beispielsweise erlauben, dass vor dem beschrifteten Abschnitt jeder Kante noch ein einziger nicht-orthogonaler Abschnitt sein darf (siehe Abbildung 16). Dadurch könnten die beschrifteten Abschnitte in einem einzelnen Port gebündelt werden, statt dass jede Kante einen eigenen Port direkt senkrecht über ihr erhält. Der Knoten könnte dadurch eine konstante Breite behalten, da er sich nicht mehr über alle von ihm ausgehenden Kanten erstrecken muss.

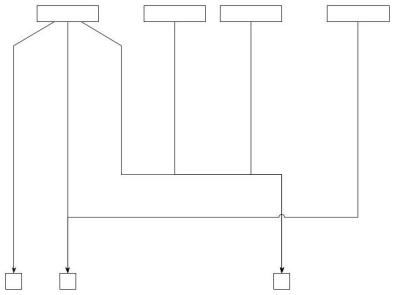
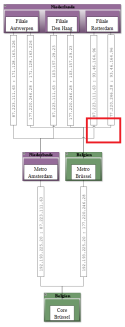


Abbildung 16: Schräge Abschnitte erlauben schmalere Knoten

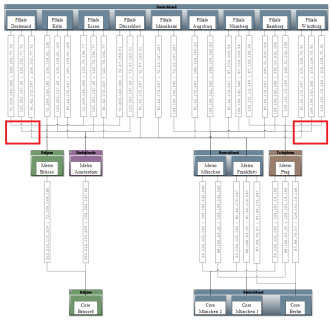
5.3 Schwachstellen

5.3.1 Seitenverhältnis

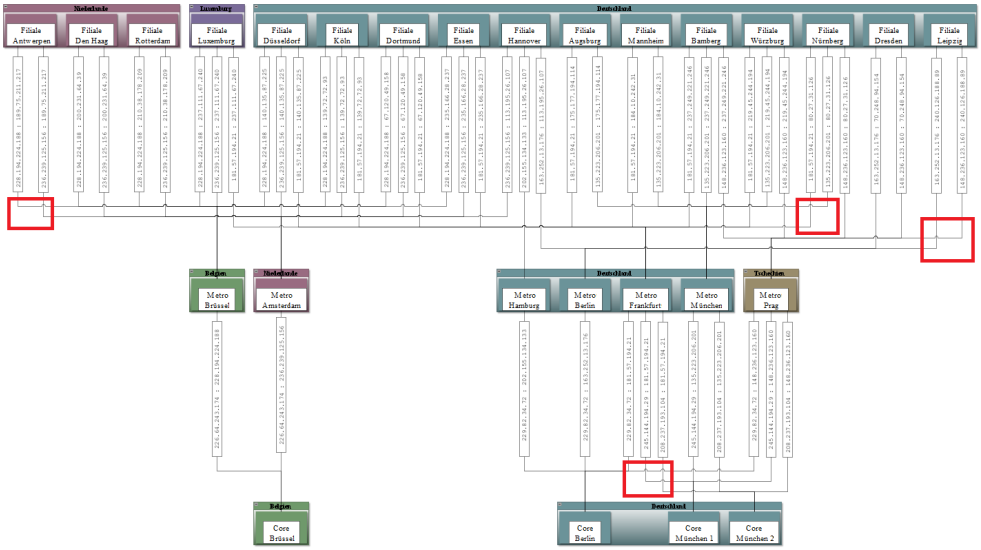
Zielen wir wie in Abschnitt 5.1 besprochen auf einen minimalen Umfang, schneidet unser Algorithmus bei größer werdenden Graphen schlecht ab. Denn wir haben bereits festgestellt, dass die verwendete Höhe ein Wachstumsverhalten von $\sqrt{E}$ hat - die Breite



(a) Netz aus Abbildung 10



(b) Netz aus Abbildung 8



(c) Netz aus Abbildung 14

Abbildung 15: In jedem der rot markierten Bereiche sind unnötige Überschneidungen gezeichnet worden. Alle der unnötigen Schnitte sind Resultat einer ungeschickten horizontalen Kantenreihenfolge.

wächst hingegen linear in E. In dieser Arbeit hatten alle Pläne eine konstante Anzahl an Layern (3), jedoch eine variierende Anzahl Knoten und Kanten. Mit zunehmender Anzahl an Kanten würde daher das Seitenverhältnis immer schlechter werden.

5.3.2 Knoten auf selber Achse

Wir positionieren alle Knoten desselben Layers auf derselben Höhe. Das ist besonders problematisch im Bezug aufs Seitenverhältnis, aber auch die Fläche kann bereits darunter leiden (siehe 5.3.3). Das Seitenverhältnis leidet, wenn beispielsweise viele Knoten existieren, die nicht zu allen benachbarten Layern Verbindungen haben. Oder auch wenn

$$\max_{i \in \{1,\dots,k-1\}} (E_i \cdot h) > (k-1) \cdot l + \sum_{i=1}^{k-1} d \cdot |Z_i|$$

wobei k die Anzahl Layer, Z_i die i-te Zielseite und E_i die Kanten zwischen Layer i und $i+1$ bezeichnen für $i \in \{1, \dots, k-1\}$. Denn sobald diese Ungleichung gilt, wird der Graph breiter sein als hoch, und das Seitenverhältnis wird negativ beeinflusst. Aber auch ungeachtet des Seitenverhältnisses kann bereits ein zu breites Layer schwerwiegende Folgen haben, wie der nächste Punkt zeigt.

5.3.3 Fläche

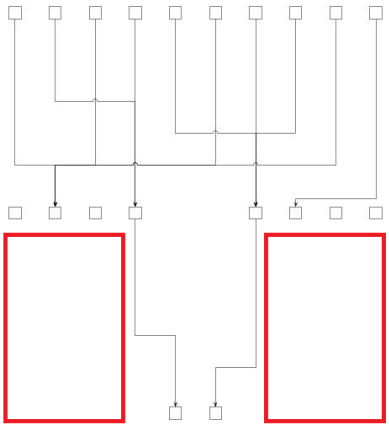


Abbildung 17: Nur die oberen Layer nutzen die Breite.

Auch ist die resultierende Fläche noch nicht ganz ideal. Das liegt daran, dass die größere der Kantenmengen E_i bestimmt, wie breit die gesamte Zeichnung wird. Der Bereich der kleineren wird dann trotzdem genauso breit werden wie der der größeren. Es könnten zwei der Layer die Breite mit ihren Labeln voll ausnutzen, wohingegen die anderen beiden viel weniger benötigt hätten, aber genau so breit gezeichnet wurden. Ein gutes Beispiel dafür ist Abbildung 14. Hier kann es in manchen Situationen sogar vorteilhaft sein, eine der beiden Verbindungen horizontal zu beschriften. Damit das einen Vorteil bringt, muss sie jedoch sehr klein sein, und sowieso kleiner als die Andere sein.

Die Ursache liegt darin, dass erst die Kanten auf einer Seite ungebündelt zu den Labeln führen müssten, wodurch eine Breite von $E_2 \cdot d$ benötigt wird, und erst anschließend auf der anderen Seite der Label gebündelt werden könnten. Die horizontale Ausrichtung benötigt also zwangsläufig bereits mehr Breite, als es *im selben Layer* die vertikale Ausrichtung täte. Das lohnt sich also erst, wenn die Kantenmenge hier hinreichend klein ist.

Literatur

[Klesen18] Felix Klesen. *Visualisierung von Netzwerkgraphen*. Juli 2018.

[Wolff00] Alexander Wolff. *A Simple Proof for the NP-Hardness of Edge Labeling*. 2000. URL: https://www1.pub.informatik.uni-wuerzburg.de/pub/wolff/pub/w-spnph-00.pdf.