

Visualisierung und Analysemöglichkeiten von georeferenzierten sozialen Netzen am Beispiel standortgebundener Infrastrukturprojekte

Bachelorarbeit
von

Tobias Haaß

An der Fakultät für Informatik
Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

Erstgutachter:	Prof. Dr. Dorothea Wagner
Zweitgutachter:	Prof. Dr. Peter Sanders
Betreuender Mitarbeiter:	Dr. Martin Nöllenburg
Zweiter betreuender Mitarbeiter:	Dr. Robert Görke

Bearbeitungszeit: 13. Juli 2011 – 12. November 2011

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe und die Satzung des Karlsruhe Institut of Technology zur Sicherung guter wissenschaftlicher Praxis beachtet habe.

Datum

Unterschrift

Inhaltsverzeichnis

1. Einleitung	3
1.1. Überblick	3
1.2. Soziale Netzwerke	4
1.3. Beschreibung der Daten	4
1.4. Anforderungen an die Visualisierung	5
1.5. Möglichkeiten zur Visualisierung sozialer Netzwerke	7
2. Vorgehensweise bei der Visualisierung	9
2.1. Ablauf	9
2.2. Visualisierung der Akteurseigenschaften	11
2.3. Projektknoten	13
3. Positionierung der Knoten	15
3.1. Bestimmung der geographischen Koordinate	16
3.2. Bestimmung der Größe der Begrenzungspolygone	17
3.2.1. Bundesland (Begrenzungspolygon für R_0)	17
3.2.2. Begrenzungspolygone von R_1 und R_2	18
3.3. Umriss des Bundeslandes	19
3.3.1. Douglas-Peucker Algorithmus	19
4. Entzerren von nahen Knoten	21
4.1. Ansatz	21
4.2. Grundidee zur Umsetzung	23
4.2.1. Umgebung U	27
4.3. Expandieren auf ursprüngliche Größe	28
4.4. Auswirkung auf den Umriss des Polygons	29
4.5. Resultat	30
4.6. Alternative Variante der Expansion	30
5. Übereinanderliegende Knoten	33
5.1. Kräftebasierte Verfahren	33
5.2. Verwendung eines kräftebasierten Verfahrens	34
5.3. Benutzung des SmartOrganicLayouter	34
5.4. Auftretende Probleme mit dem SmartOrganicLayouter	36
5.4.1. Zurückschieben von Knoten in ihre vordefinierte Region	36
6. Beschriftung der Knoten	39
7. Analysemöglichkeiten	43
7.1. Sektoren und Georeferenz	43
7.2. Clustering	44
7.2.1. Analyse der Projektstruktur	45

8. Zusammenfassung und Ausblick	47
Anhang	51
A. Fallstudien	51
B. Programm zum Erstellen des Layouts	56
B.1. Aufbau der Daten	56
B.1.1. Akteure	56
B.1.2. Projektknoten	57
B.1.3. Kanten	57
B.2. Programmoberfläche	57
B.3. Ablauf	58
Literaturverzeichnis	59

Kurzbeschreibung

In dieser Bachelorarbeit geht es um das Visualisieren standortgebundener Infrastrukturprojekte, die als soziales Netzwerk modelliert werden. Solche Projekte sind sehr kostenintensiv und benötigen daher umfangreiche Planungen. Mit dieser Arbeit wird ein Werkzeug geliefert um ein Infrastrukturprojekt zu analysieren. Zunächst wird das Projekt als Graph modelliert. Für die Zeichnung wird jedem Knoten eine initiale Position zugeordnet, die von seinem Herkunftsort aus der Realität abhängt. Bei den betrachteten Projekten hat sich gezeigt, dass sich viele Akteure am Projektstandort ansammeln. Um für eine bessere Übersicht zu sorgen, wird die nähere Umgebung um den Standort entzerzt. Des Weiteren werden Knotenüberlappungen mit einem kräftebasierten Algorithmus aufgelöst. Abschließend werden für jeden Knoten Beschriftungen eingefügt und eine Position für diese berechnet. Um die Struktur des Graphen zu analysieren wird das Projekt als Correlation-Clustering Problem betrachtet und ein ganzzahliges lineares Optimierungsproblem gelöst. Die beschriebenen Verfahren wurden in Java implementiert und als Ergebnis werden drei Fallstudien vorgestellt.

1. Einleitung

1.1. Überblick

In unserer globalisierten Welt werden Handlungen vermehrt als Projekte bezeichnet. Gerade in der Wirtschaft werden Aufgaben von kleineren bis größeren Projektteams durchgeführt. Umfangreiche Planungen, Ressourcen- und Aufgabenteilung, sowie Steuerungsmaßnahmen sind nötig um die gesteckten Ziele zu erreichen. In großen Infrastrukturprojekten obliegen diese Maßnahmen einer Vielzahl unterschiedlicher Akteure. Der erfolgreiche Abschluss eines Projekts hängt maßgeblich davon ab, wie gut die Zusammenarbeit einzelner Akteure koordiniert wird. Je mehr Akteure beteiligt sind, desto schwieriger wird es alle Interessen zu berücksichtigen. Bei Großprojekten wie Stuttgart 21 gilt es frühzeitig Konflikte zu erkennen um daraus Maßnahmen ableiten zu können, mit denen Schwierigkeiten vermieden werden können.

Aufgrund der Standortgebundenheit von infrastrukturellen Bauprojekten ist zu erwarten, dass vor allem lokal angesiedelte Akteure eine wichtige Rolle einnehmen. So sind kommunale Entscheidungsträger, Anwohner, lokale Medien und viele weitere Akteure im Projekt involviert. Dennoch haben auch regionale, überregionale, nationale bis hin zu internationalen Akteuren Einfluss auf das Projekt. Deshalb müssen für eine umfassende Analyse die verschiedenen sozialen Ebenen berücksichtigt werden, auf denen Akteure einen unterschiedlich starken Einfluss auf die Entscheidungsprozesse haben können.

Ziel dieser Arbeit ist es, eine Möglichkeit zu zeigen, wie man solche Projekte darstellen kann. Die Projekte werden mit Hilfe von Ansätzen aus der Soziologie als Netzwerke modelliert und daraus eine Visualisierung in Form einer zweidimensionalen Zeichnung entwickelt. Diese Zeichnungen können dann als Werkzeug für eine Analyse dienen. Die Visualisierung von Netzwerken ist eine wichtige Disziplin in der theoretischen Informatik. Durch eine geschickte Wahl des Designs kann man den Informationsgehalt von Daten leichter einem Anwender vermitteln. Das besondere an den betrachteten Netzwerken ist die Eigenschaft, dass die Akteure schon eine initiale Position, abgeleitet durch ihre geographische Koordinate, besitzen. Ein völlig beliebige Platzierung mit Hilfe gängiger Layoutverfahren ist daher nicht möglich. Durch die Georeferenzierung erreicht man eine Visualisierung, die näher an der Realität ist und erhält gleichzeitig neue Möglichkeiten für die Analyse standortgebundener sozialer Netzwerke. Es werden ebenfalls Methoden gezeigt, wie man mit Hilfe der Visualisierung und einer Betrachtung der Struktur des zugrundeliegenden Graphen eine Analyse des Projekts durchführen kann. Mittels des erlangten Wissen aus so einer Analyse, kann man dann eventuell Konflikte im sozialen Gefüge früher erkennen und

auch abwenden, was die Erfolgchancen für das Projekt steigern oder die Kosten senken kann.

1.2. Soziale Netzwerke

Soziale Netzwerke sind heute dank Plattformen wie Facebook und MySpace in aller Munde. Hier bezeichnet der Begriff jedoch eine eigene Unterart von Graphen. Genauer gesagt ist ein Graph ein mathematisches Konzept um Netzwerke zu modellieren. Ein Graph G besteht aus einem Tupel (V, E) , wobei V aus einer Menge von Objekten, genannt Knoten, besteht und die Menge E auftretende Verknüpfungen zwischen Knotenpaaren (u, v) repräsentiert. Ein Element aus E wird als Kante bezeichnet. Die Kanten eines Graphen können entweder *gerichtet* oder *ungerichtet* sein. Bei einer Kante $e = (u, v)$ entspricht dies einer symmetrischen bzw. asymmetrischen Beziehung zwischen den Knoten u und v . Im Folgenden wird von einem ungerichteten Graphen ausgegangen. Soziale Netze sind ebenfalls mathematische Konstrukte, bestehend aus einem Tupel von Knoten und Kanten. Für gewöhnlich werden die Knoten auch Akteure genannt, da sie mit einer Teilmenge aus den restlichen Knoten interagieren oder sonst eine abstrakte Beziehung zwischen ihnen herrscht. Solche Beziehungen werden wieder durch Kanten ausgedrückt. Die Menge aller Akteure wird mit $A = (a_0, \dots, a_n)$ bezeichnet. Sie können im Graph Menschen, Organisationen und andere Gruppierungen repräsentieren.

Auch in der Soziologie gibt es Ansätze, die sozialen Strukturen und Handlungen netzwerktheoretisch zu analysieren. So lange die Datenmenge überschaubar ist, kann eine Visualisierung sinnvoll zu einem besseren Verständnis beitragen. Zur Visualisierung wird der Graph in eine Zeichnung überführt. Jedem Knoten u wird hierzu meistens ein graphischer Punkt an der Position p_u zugeordnet. Eine Kante $e = (u, v)$ wird durch eine Verbindungslinie der Positionen p_u und p_v symbolisiert. Gerichtete Kanten macht man üblicherweise mit Hilfe von Pfeilen deutlich.

Durch die Interpretation als Graph erhält man zusätzlich die Möglichkeit, bekannte Analysemethoden aus der Algorithmik auf das Netzwerk anzuwenden. Um Informationen aus dem Graph zu erhalten kann dessen Struktur analysiert werden, um Aufschluss über die Beziehungen zwischen Akteuren zu bekommen. Daraus können dann eventuell Muster abgeleitet werden, mit deren Hilfe man Aussagen über das Netzwerk treffen kann. Verbindet man die gewonnenen Erkenntnisse mit soziologischem Wissen kann man Thesen über das reale Projekt ableiten.

In dieser Arbeit werden Infrastrukturprojekte als Netzwerke modelliert. Hierfür wurde auf einen Ansatz aus der Techniksoziologie zurückgegriffen. In der Akteurs-Netzwerk-Theorie (ANT) werden Technik und Gesellschaft nicht mehr komplett voneinander getrennt betrachtet [Lat07]. Vielmehr werden beide Bereiche einander gleichgestellt. So treten in den Netzwerken nicht nur menschliche Akteure auf, sondern beispielsweise auch technische Elemente, Gesetze und Umwelteinflüsse, die untereinander in Beziehung stehen und ein gemeinsames Netzwerk z.B. in Form eines Infrastrukturprojekts bilden.

1.3. Beschreibung der Daten

Um ein Projekt modellieren zu können, benötigt man Daten aus der Realität. Um an Datenressourcen zu gelangen, gibt es verschiedene Wege. Es bietet sich an die Websites der Projektbetreiber nach enthaltenen Informationen auszuwerten. Auf diese Weise erfährt man etwas über die Entwicklung, das Projektmanagement, den Verlauf des Projekts und die Investoren. Gezieltes Sammeln und Verarbeiten von Beiträgen aus den Medien (Rundfunk, TV, Zeitschriften, ...) verhilft einem zu weiteren Daten. Weitere Möglichkeiten sind

Umfrageergebnisse, Stellungnahmen von Umweltverbänden und Protokolle von politischen Entscheidungsträgern. Das Modellieren ist also mit viel Aufwand und evtl. auch Kosten verbunden.

Für die Bachelorarbeit wurden drei anonymisierte Datensätze von Infrastrukturprojekten betrachtet und als soziales Netzwerk interpretiert. Sie beinhalten Projektdetails wie den Standort und einer Liste von teilnehmenden Akteuren. Dabei existieren für jeden Akteur folgende Eigenschaften:

- Name des Akteurs
- Zuteilung in Regionen
- Postleitzahl
- Einteilung in Sektoren
- Funktion im Projekt
- Haltung gegenüber dem Projekt
- Beziehung zwischen Akteuren

Mit Hilfe der Postleitzahl kann der Ort jedes Akteurs ermittelt werden. Da der ermittelte Ort nicht für jeden Akteur die gleiche Bedeutung hat, wurden zusätzlich Regionen definiert. Je nach Region hat die Postleitzahl einen unterschiedlich starken Einfluss auf die Position in der Zeichnung. Die Beziehung zwischen Akteuren gibt Aufschluss darüber, ob sich zwei Akteure negativ, neutral oder positiv gegenüber stehen. Des Weiteren existiert in der Soziologie schon eine Zuordnung von Akteuren in bestimmte Sektoren, z.B. Wirtschaft, Zivilgesellschaft und Politik. Die verschiedenen Sektoren können beispielsweise von Niklas Luhmanns [Luh84] „funktionalen Teilsysteme“ abgeleitet werden. Jedes Teilsystem besteht aus Akteuren mit ähnlichen Interessen und Ressourcen, die wiederum Beziehungen zwischen Akteuren beeinflussen.

Neben menschlichen Akteuren treten vor allem Firmen und Institutionen in den Daten auf. Auch besondere Ereignisse bzw. Gesetzesentscheidungen, die andere Akteure und damit ein Projekt im Gesamten beeinflussen, dürfen für eine umfangreiche Modellierung nicht vergessen werden. Zusätzlich treten noch diverse Tageszeitungen und andere Printmedien in Erscheinung, die durch ihre Berichterstattung ebenfalls positiven oder negativen Einfluss auf das Projekt ausüben können. Eine Übersicht auftretender Akteurstypen wird in Tabelle 1.1 aufgelistet. Die betrachteten Projekte bestehen aus etwa 30 – 40 Akteuren mit einer ähnlichen Anzahl Kanten.

Institutionen	Personen	Ereignisse
Projektentwickler	Bürgermeister	Gesetze
Investoren/ Industriefirmen	Politiker	Politikwechsel
Hoch-/ Tiefbau	Bürger	Naturereignisse
Zeitungen	Journalisten	Technische Störfälle
Gemeinderäte	etc.	etc.
Umweltverbände		
etc.		

Tabelle 1.1.: Beispiele von Akteurstypen

1.4. Anforderungen an die Visualisierung

Im Folgenden werden einige Ziele genannt, die für die Visualisierung wichtig sind. In der Literatur [BETT99, KW01] werden verschiedene Gütekriterien genannt, nach denen man

ein Layout optimieren kann. Ein Layout oder eine Zeichnung ist das algorithmische Kernproblem um Graphen zu visualisieren. Dabei wird jedem Knoten bzw. jeder Kante eine geeignete Position in der Ebene zugeordnet. Die wichtigsten Kriterien für ein gutes Layout sind die folgenden:

Kreuzungen Vermeidung von kreuzenden Kanten und sich überlappenden Knoten.

Platzverbrauch Es wird versucht möglichst wenig Platz für die Zeichnung zu verwenden. Eine homogene Verteilung der Knoten ohne viel Freiraum erzeugt einen besseren Eindruck.

Knickminimierung Dabei wird versucht eine Kante möglichst geradlinig zu zeichnen. Bei vielen Knicken hat das menschliche Auge Schwierigkeiten Linien zu verfolgen.

Winkelauflösung Maximiere den kleinsten Winkel zweier inzidenter Kanten zum gleichen Knoten. Durch den größeren Winkel zwischen den Kanten kann das Auge den Linien leichter folgen.

Neben diesen üblichen Charakteristiken, die für die Lesbarkeit eines Graphlayouts von entscheidender Bedeutung sind, gibt es hier noch einige Besonderheiten. Im Gegensatz zu anderen sozialen Netzwerken ist in diesem Anwendungsfall die Georeferenz eines Knotens ein wichtiger Faktor. Es gibt Anwendungen bei denen es egal ist an welcher Position sich ein Akteur am Ende in der Zeichnung befindet. Wir wollen aber jedem Knoten einen Ort auf einer stark abstrahierten Landkarte zuordnen. Dieser Ort muss bei der Positionierung der Knoten auf jeden Fall beachtet werden. Abweichungen werden nötig sein um Überlagerungen von Knoten zu vermeiden. Sie sollten aber möglichst gering sein um eine gute geographische Relation zwischen den Akteuren zu gewährleisten. Kreuzende Kanten werden zwar versucht zu vermeiden, weil die Knoten aber eine vorgegebene Position haben und diese vergleichsweise wichtiger ist, dürfen Kreuzungen auftreten. Dahingegen wird aufgrund der Georeferenz die Winkelauflösung nicht betrachtet. Kanten werden als gerade Verbindungslinie zwischen zwei Knoten gezeichnet. Somit treten Knicke erst gar nicht auf, weshalb eine Knickminimierung nicht notwendig ist.

Da die Relevanz der Georeferenzierung nicht für alle Akteure gleich ist werden verschiedene Regionen R_0, \dots, R_j definiert und eine Partitionierung der Akteure A in disjunkte Teilmengen $A = A_0 \cup \dots \cup A_j$ vorgenommen. Die Regionen sollen ähnlich einer Zielscheibe hierarchisch angeordnet werden, wobei die Region R_i alle Akteure aus A_i enthält. Die innere Region R_0 ist dabei in der Mitte der Zeichnung zu finden. Dort sind alle Akteure, die für das Projekt von zentraler Bedeutung sind und die Bindung zum zugehörigen Ort besonders wichtig ist. Nach außen hin kann die Georeferenzierung stärker vernachlässigt werden. Genauso enthält R_0 auch einen Projektknoten, der den Projektstandort und einige technische Eigenschaften des Bauprojekts zeigt. Zur besseren Orientierung wird die innere Region durch das Bundesland, in dem sich der Projektstandort befindet, umrandet. Alle weiteren Regionen entsprechen konzentrisch angeordneten konvexen Hüllen des Bundeslandes (Abbildung 1.1). Ähnlich zu der Georeferenz sollten auch die zugewiesenen Regionen nicht vertauscht werden. Im weiteren Verlauf werden nur drei Regionen betrachtet. Das Verfahren kann aber problemlos auf mehr Regionen erweitert werden.

Der Zeichnung soll außerdem anzusehen sein, wie es sich mit der grundsätzlichen Haltung jedes Akteurs gegenüber dem Projekt verhält, also ob er für oder gegen das Projekt ist. Um das Projekt zu analysieren ist die Zugehörigkeit jedes Akteurs zu seinem Sektor noch wichtig und sollte aus der Zeichnung ebenfalls abzulesen sein. Eine Zuordnung zum gleichen Sektor zeugt im Wesentlichen von ähnlichen Interessen und Ressourcen der jeweiligen Akteure.

Für das Interpretieren von Projektdaten ist es außerdem wichtig, dass die gleichen Daten bei erneuter Berechnung auch das gleiche Layout ergeben. Sonst muss man sich immer

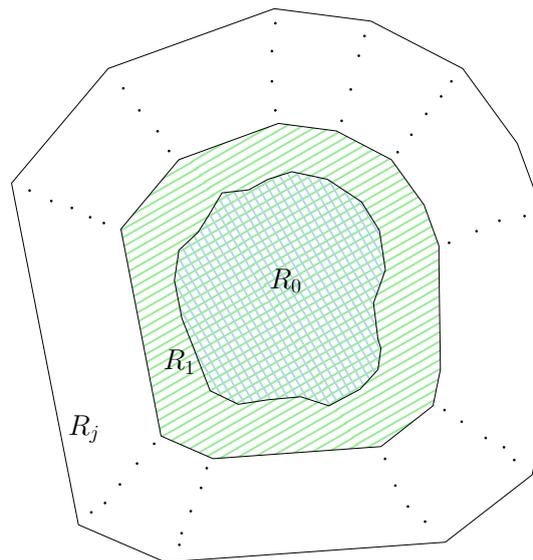


Abbildung 1.1.: Aufbau der Regionen

wieder neu orientieren und einlesen. Das heißt alle Verfahren müssen stets deterministisch sein.

1.5. Möglichkeiten zur Visualisierung sozialer Netzwerke

Das Besondere beim Zeichnen von sozialen Netzen ist die Tatsache, dass oft nicht klar ist wie ein taugliches Layout dabei aussieht. So kann eine gute Zeichnung dem Betrachter helfen Sachverhalte besser zu verstehen, während eine schlechte Zeichnung eher zu Verwirrungen führt. Für die Untersuchung und das Verständnis von Netzwerken ist die Netzwerkvisualisierung daher ein grundlegendes Werkzeug. Entscheidend dabei ist, welche Information man hervorheben will. Einen allgemeinen Überblick über Graphvisualisierungen bieten [BETT99] und [KW01]. In [BW03] werden einige Algorithmen vorgeschlagen um speziell soziale Netzwerke zu visualisieren.

Beliebt sind vor allem kräftebasierte Verfahren. Sie eignen sich besonders gut, wenn man keine besonderen Ansprüche an die Positionierung der Knoten stellt. Man kann durch einfache Implementierungen Zeichnungen von Graphen generieren, die Rücksicht auf gängige Optimalitätskriterien nehmen. Ein kurzer Überblick über solche Verfahren wird später noch gegeben (Kapitel 5.1). In 1.2c ist ein Beispiel für die Anwendung eines kräftebasierten Ansatzes zu sehen. Bekannte Implementierungen werden in [KK89] und [FR91] vorgestellt. Eine weitere Möglichkeit für die Visualisierung wird in Beispiel 1.2d gezeigt. Dort ist ein radiales Layout zu sehen. Radiale Zeichnungen geben vor allem darüber Aufschluss, wie wichtig ein Akteur für das Netzwerk ist und welche Rolle er dort spielt. Nähere Informationen dazu sind in [BKW03] zu finden. Andere Beispiele werden in den Abbildungen 1.2b und 1.2a gezeigt. Dort werden die Akteure kreisförmig bzw. hierarchisch angeordnet. Damit will man vor allem Beziehungen zwischen Akteuren verdeutlichen. Weitere Methoden für ein Layout werden in [HHE07] genannt. Dort wird außerdem eine Studie vorgestellt, die verschiedene Visualisierungsarten miteinander vergleicht und feststellt, welche Auswirkungen sie auf die Wahrnehmung von Eigenschaften des Netzwerks haben.

Die verschiedenen Layoutverfahren können dem Leser unterschiedliche Eigenschaften deutlich machen. Gesucht ist letztendlich eine Zeichnung, welche die Schlüsseleigenschaften des Netzwerks besonders hervorhebt. Für unsere Anforderungen ist jedoch keines der genannten Verfahren direkt geeignet, da wir die Knoten vor allem unter dem Gesichtspunkt

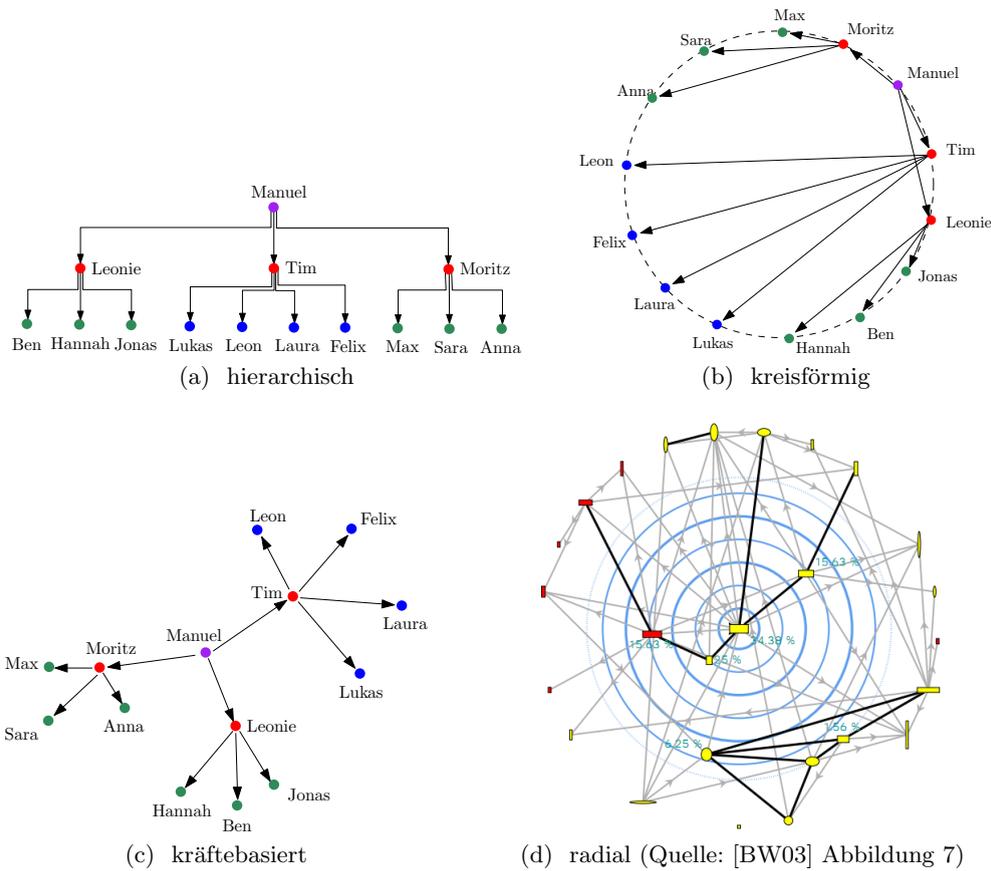


Abbildung 1.2.: Beispiele für soziale Netzwerke, visualisiert durch verschiedene Ansätze.

der Georeferenzierung zeichnen wollen. So wird, wie bei Ortschaften auf einer Landkarte, jedem Akteur eine Position zugeordnet. Erstellt man mit diesen Positionsangaben eine Zeichnung (vgl. Abbildung 1.3) erkennt man schnell einige Nachteile. Als erstes fällt auf, dass sich viele Knoten auf engem Raum ansammeln können. Dies ist ein typisches Merkmal der vorhandenen Infrastrukturprojekte. Neben der Konzentration vieler Akteure an einem Ort gibt es auch einige Überlappungen. Außerdem können große freie Flächen entstehen, wenn die Lage einzelner Akteure weit voneinander entfernt ist. Es müssen also Verfahren gefunden werden, um diese Nachteile zu umgehen.

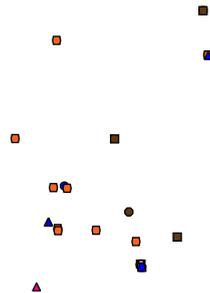


Abbildung 1.3.: Beispiel für eine Zeichnung, bei der die Knoten an dem Herkunftsort ihrer zugehörigen Akteure platziert werden.

2. Vorgehensweise bei der Visualisierung

2.1. Ablauf

Im Folgenden wird ein Überblick über den groben Ablauf der einzelnen Visualisierungsschritte gegeben. Ein Skizze wird in Abbildung 2.1 gezeigt. Zu Beginn wird für jeden Knoten eine initiale Position gewählt. Wie die einzelnen Knoten platziert werden wird in Kapitel 3 beschrieben. In Abbildung 2.1a wird beispielhaft eine initiale Positionierung der Knoten innerhalb eines imaginären Bundeslandes gezeigt.

Als Merkmal von Infrastrukturprojekten wird eine Konzentration vieler Akteure am Projektstandort angenommen. Ein Grund dafür können wirtschaftliche Motive sein. So bemühen sich viele lokal angesiedelte Firmen um eine Beteiligung an den Projekten. Außerdem haben Akteure aus der Nähe auch ein besonderes Interesse am Projekt. Oft ist dieses Interesse politischer Natur und betrifft insbesondere Genehmigungen und Fördergelder. Bürgerinitiativen und die lokale Presse beteiligen sich ebenfalls durch Diskussionen über das Bauvorhaben. Aus diesem Grund sammeln sich viele Knoten im repräsentierenden Graphen in der Nähe des Projektstandortes. Um die Übersicht zu wahren, müssen also besonders Knoten in der Nähe des Projektstandortes voneinander entzerrt werden (Abbildung 2.1b). Dies soll alle Knoten betreffen, die sich im gleichen Bundesland wie der Projektstandort befinden. Dementsprechend wird die Entzerrung als algorithmisches Problem formuliert und betrachtet. Kapitel 4 zeigt wie man damit umgehen kann. Als Nächstes werden der Zeichnung die äußeren Regionen mit den entsprechenden Akteuren hinzugefügt. Die Platzierung dieser Knoten wird ebenfalls in Kapitel 3 beschrieben.

Das Entzerren der Projektumgebung reicht jedoch meist noch nicht aus um die Knoten weit genug voneinander zu trennen. Außerdem verhindert es keine Überlagerung von Knoten, die Akteure mit der gleichen Postleitzahl repräsentieren und damit die gleichen Koordinaten besitzen. Solche Knoten gibt es tatsächlich auch recht häufig, da einige Akteure unterschiedliche Funktionen im Projekt haben. Diese sind aber jeweils durch einen eigenen Knoten repräsentiert, der die gleiche Koordinate besitzt. In Kapitel 5 wird ein Verfahren beschrieben, um die Überlappungen aufzulösen (Abbildung 2.1d). Man könnte den dort verwendeten Algorithmus auch ohne das Verfahren aus Kapitel 4 benutzen und damit auch ein Entzerren erreichen. Allerdings wird durch das Entzerren eine bessere initiale Position der Knoten für den Algorithmus ausgewählt. Würde man den vorherigen Schritt weglassen, wird die Georeferenz der Knoten stärker missachtet.

Um das ganze Layout leichter lesbar zu machen, wird am Ende noch jedem Knoten ein Label zugewiesen (Abbildung 2.1e). Die Platzierung von Beschriftungen in einem Graphen

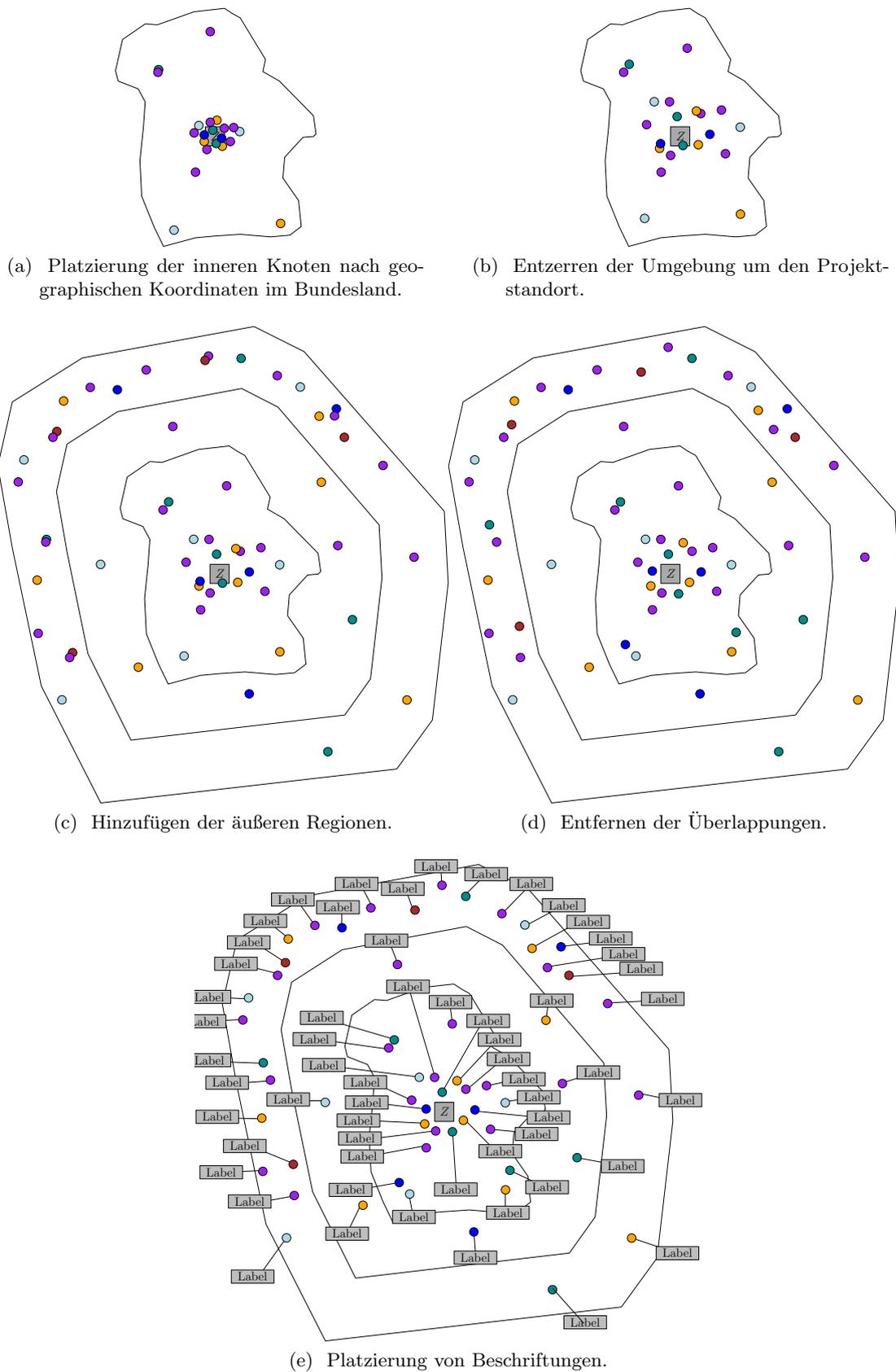


Abbildung 2.1.: Übersicht über die einzelnen Visualisierungsschritte.

ist ein Problem auf das man bei automatischen Layouts oft stößt. Hier wurde die Regionszugehörigkeit der Knoten ausgenutzt, um möglichst wenige Überschneidungen zu erhalten. Die Beschreibung dazu findet man in Kapitel 6.

2.2. Visualisierung der Akteurseigenschaften

Im Allgemeinen stehen für die Visualisierung einige wenige Möglichkeiten zur Verfügung, um bestimmte Informationen zu vermitteln. Bei einer wichtigen Information sollte man deshalb eine Eigenschaft wählen, die dem Betrachter besonders gut auffällt und sich von anderen Visualisierungsmöglichkeiten gut abhebt. Jaques Bertin [Ber11] hat sich damit in der Kartographie beschäftigt und das theoretische Konzept der graphischen Variablen formuliert. Dabei unterscheidet er sechs Variablen, die er verschiedenen Charakteristiken zuordnete (Tabelle 2.1).

Variable	assoziativ	selektiv	geordnet	quantitativ
Größe	-	×	×	×
Helligkeit	-	×	×	-
Muster	×	×	×	-
Farbe	×	×	-	-
Richtung	×	×	-	-
Form	×	-	-	-

Tabelle 2.1.: Eigenschaften graphischer Variablen nach Bertin.

Assoziativ bezeichnet dabei die Eigenschaft einer Variable, sofort alle Objekte einer Gruppe identifizieren zu können, die sich nur an Hand von dieser Variablen unterscheiden lassen. So lassen sich ebenso große und gleichfarbige Objekte leicht gruppieren, selbst wenn sie sich in ihrer Form unterscheiden. Die Variable Form ist daher assoziativ. Außerdem wirken assoziative Variablen auf die menschliche Wahrnehmung trotz Unterscheidung mit der gleichen Intensität. Bei der Größe trifft dies nicht zu, da wir große Objekte deutlich leichter ins Auge fassen als kleine Objekte. Die selektive Eigenschaft einer Variablen hilft uns dabei sofort zu erkennen, welche Objekte bezüglich dieser Eigenschaft eine Gruppe bilden. Besonders leicht fällt uns dies bei Farben, zum Beispiel die Gruppe aller roten Objekte. Dagegen erlangt man bei Formen nur schwer einen schnellen Überblick. Graphische Variablen mit der Ordnungseigenschaft erlauben uns Vergleiche zwischen Objekten anzustellen und diese zu sortieren. Während man Objekte nach der Größe sortieren kann, ist dies bei Farben nicht ohne zusätzliche Information möglich. Als letzte Eigenschaft bleibt die Quantitativität. Sie erlaubt uns Aussagen über Zahlenverhältnisse zweier Objekte zu treffen. So kann man der Größe relativ leicht ablesen, ob etwas doppelt oder viermal so groß ist. Im Bezug auf die Visualisierung von sozialen Netzwerken kann diese Einteilung helfen, zusätzliche Informationen für jeden Knoten dem Leser anzubieten.

Da für die Analyse des Netzwerkes die Zugehörigkeit zu den Sektoren von zentraler Bedeutung ist, werden Knoten, die dem gleichen Sektor zugeordnet werden, auch mit der gleichen Farbe ausgefüllt. Für diese Zuordnung spielt vor allem die Selektivität eine wichtige Rolle. So kann sich der Leser schnell einen Überblick über alle Akteure eines Sektors verschaffen.

Für die Funktion innerhalb des Projekts ist die Selektivität nicht ganz so wichtig. Daher gibt die Form der Knoten darüber Aufschluss, welche Funktion der Akteur im Projekt hat. Jedem Knoten wird dazu eine einfache geometrische Figur, z.B. ein Quadrat oder ein Dreieck, zugeordnet. Auf diese Weise erhält ein Knoten u eine Fläche. Jede geometrische

Figur wird dabei gerade so groß gewählt, dass die Bounding Box aller Figuren gleich groß ist. Somit nimmt jeder Knoten in etwa die gleiche Fläche ein. Im weiteren Verlauf wird mit $F(u)$ Bezug zu dieser Fläche genommen.

Des Weiteren erhalten alle Knoten eine Beschriftung, die verrät, um welchen Akteur es sich bei dem Knoten handelt. Die Hintergrundfarbe der Beschriftung unterscheidet sich dabei. Damit wird gezeigt, wie sich der Akteur gegenüber dem Projekt verhält (grün = Befürworter des Projekts, rot = Gegner des Projekts, gelb = neutral). Die Farbe wird je nach Stärke der Haltung abgestuft, d.h. ein helleres Grün zeigt eine weniger starke Befürwortung des Projekts als ein dunkleres Grün. Auch die Farbe der Kanten zwischen den Knoten sagt etwas über die Beziehung der Akteure aus. So zeigt eine rote Kante zwischen Knoten u und v , dass sich die zugehörigen Akteure negativ gegenüber stehen. Umgekehrt verhält sich eine grüne Kante. Hier kommt die graphische Variable der Helligkeit zum Tragen, mit der man leicht eine Ordnung erstellen kann (von rot über gelb nach grün).

Zur besseren Übersicht wird dem Graphen noch eine Legende hinzugefügt. Die Legende selbst besteht ebenfalls aus Knoten, die entsprechend der zu zeigenden Eigenschaft eingefärbt bzw. geformt sind. Die Beschriftung dieser Knoten verrät dem Betrachter wie die verwendeten Eigenschaften zu interpretieren sind. Ein Beispiel für eine Legende findet sich in Abbildung A.1.

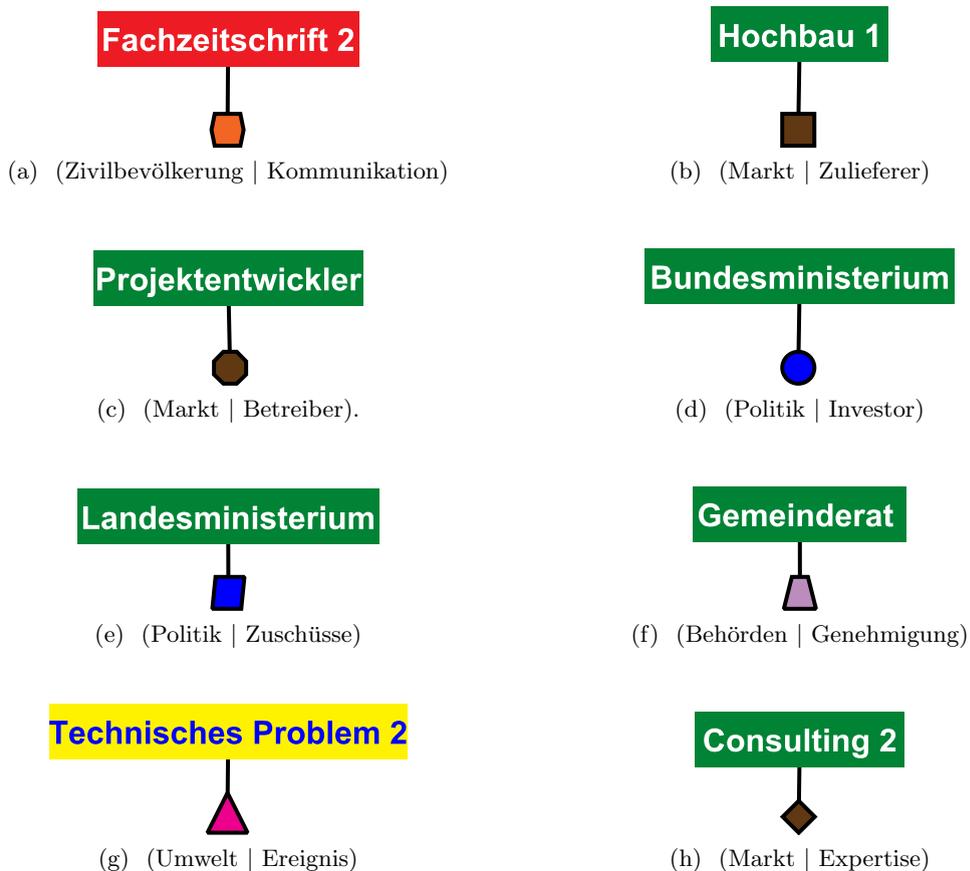
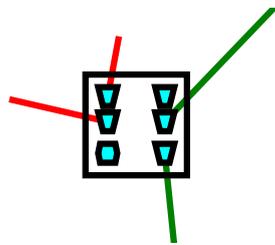


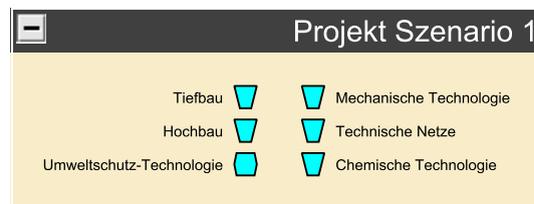
Abbildung 2.2.: Beispiele für Akteure (Sektor | Funktion).

2.3. Projektknoten

Der Projektstandort wird durch einen eigenen Knoten dargestellt. An ihm kann man grundlegende technische Eigenschaften über das Bauprojekt ablesen. Durch die Beschreibung der technischen Eigenschaften kann der Knoten verhältnismäßig groß werden. Dadurch nimmt er sehr viel Platz im Zentrum weg, was schlecht für die Georeferenzierung der Knoten ist. Um ein besseres Layout des Netzwerks zu erhalten wird für den Projektstandort ein Stellvertreter des eigentlichen Projektknotens gezeichnet. Dieser muss nicht viel größer als alle anderen Akteure sein (Abbildung 2.3a). Etwas abgesetzt neben der Zeichnung des Graphs wird ein weiterer Projektknoten eingefügt. Er dient nur zur besseren Lesbarkeit und ist selbst kein Bestandteil des eigentlichen Netzwerks (Abbildung 2.3b).



- (a) Die kleinere Variante taucht als Knoten im Graph auf. Damit wird der Standort symbolisiert und technische Eigenschaften gezeigt. Diese können wiederum durch Kanten mit anderen Akteuren verbunden sein.



- (b) In der großen Variante des Projektknoten kann man die technischen Eigenschaften ablesen und so zuordnen welcher Akteur, mit welcher Eigenschaft verbunden ist.

Abbildung 2.3.: Projektstandort als Knoten.

3. Positionierung der Knoten

Die meisten Akteure können durch eine Postleitzahl eindeutig einem Ort zugewiesen werden. Damit lässt sich aus einer Datenbank die zugehörige geographische Koordinate auslesen. Hierzu befindet sich eine Beschreibung in Abschnitt 3.1. Auf diesen Koordinaten beruht dann die initiale Position des jeweiligen Knotens in der Zeichnung. Die initiale Position wird für die weitere Berechnung auch als Wunschposition des Knotens angesehen und ist daher sehr wichtig um die Georeferenzierung für das Ergebnis zu wahren.

Allerdings lässt sich nicht jedem Akteur eine gültige Position zuweisen, da einige Institutionen oder Ereignisse nicht zweifelsfrei einer Postleitzahl zugeordnet werden können. Ein Beispiel dafür wäre ein Gesetz. Einige andere Akteure sind außerdem nicht in Deutschland beheimatet. Deshalb wurden alle Akteure zusätzlich einer Region zugeordnet. Es werden drei Regionen unterschieden: R_0 , R_1 und R_2 (Abbildung 1.1).

Zu den Akteuren A_0 gehören alle Akteure, die sich in dem gleichen Bundesland wie der Projektstandort befinden. Ihre Position wird wie gehabt durch ihre Postleitzahl bestimmt. Für die Visualisierung wird zusätzlich noch das Bundesland des Projektstandortes als Polygon P_0 eingefügt. Damit lassen sich für den Betrachter die Knoten geographisch besser einordnen. Weitere Informationen dazu sind in Abschnitt 3.3 zu finden. Jeder Knoten u der einen Akteur aus A_0 repräsentiert, erhält zunächst seine Wunschposition als Koordinate.

Nun findet eigentlich ein weiterer Zwischenschritt statt, bei dem das Polygon P_0 verändert wird (Näheres dazu in Kapitel 4). Wir gehen hier erst einmal davon aus, dass dieser Schritt bereits durchgeführt wurde und wir als Ergebnis das veränderte Polygon P'_0 erhalten. Als Abgrenzung der folgenden Region R_1 wird die konvexe Hülle von P'_0 berechnet und vergrößert (Abschnitt 3.2). Man erhält das Polygon P_1 . Danach werden alle Akteure aus A_1 als Knoten dem Graphen hinzugefügt. Zu diesen gehören Akteure, die sowohl geographisch als auch entscheidungstechnisch noch wichtig für das Projekt sind, aber nicht mehr im gleichen Bundesland beheimatet sind oder für die keine genaue Position bestimmt werden kann. Damit die Georeferenz auch hier eine Rolle spielt wird die Position wie in Abbildung 3.1 gezeigt bestimmt. Die Position der Knoten wird nur für den Bereich $P_1 \setminus P'_0$ ermittelt. Hierzu wird ein Vektor zwischen Projektstandort und Wunschposition gebildet. Dieser gibt die Richtung für die Platzierung des Knotens an. Die Länge wird nur auf die Fläche außerhalb von P'_0 beschränkt, also den im Bild rot markierten Bereich. Zusätzlich wird an den Rändern eine Toleranzgrenze beachtet, damit der Knoten nicht zu nah an den Begrenzungspolygonen P_1 und P'_0 liegt. Dies hat sonst negative Auswirkungen bei dem im Kapitel 5 beschriebenen Verfahren, welches auftretende Knotenüberlappungen auflöst.

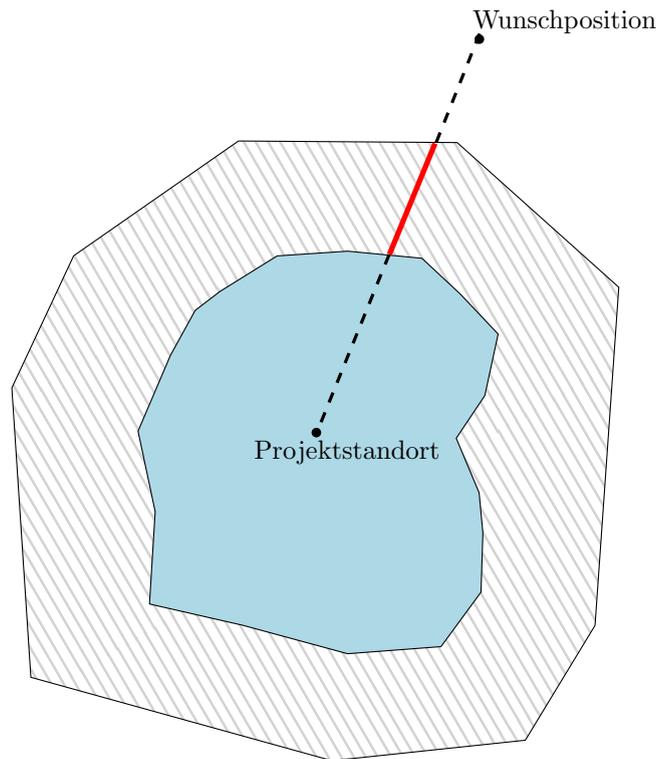


Abbildung 3.1.: Bestimmung der Position von Akteuren aus A_1 . Die Berechnung der Position für A_i mit $i > 1$ erfolgt analog.

Dort besteht die Gefahr, dass der Knoten aus der Region herausrutschen kann, wenn er sich zu nah oder gar auf dem Rand befindet. Die letztendlich berechnete Position auf der roten Strecke wird zufällig gewählt. Damit wird im Sinne der Georeferenz nur die grobe Richtung garantiert. Einzelne geographische Relationen zwischen Akteuren müssen dadurch nicht mehr korrekt sein. So kann es in den Regionen R_i ($i > 0$) zu einer Nord-/Süd- bzw. West-/Ost-Vertauschung kommen. Da dies aber nur für die Akteure aus A_0 wirklich wichtig ist, wird dieser Nachteil bewusst in Kauf genommen.

Bei Knoten, denen man keine Postleitzahl und damit auch keine Wunschposition zuordnen kann, werden die Koordinaten komplett zufällig gewählt. Es wird nur noch sichergestellt, dass sich die Position auch innerhalb der zugewiesenen Region befindet. Damit die Positionierung aber trotzdem deterministisch bleibt wird vor der Berechnung ein Zufallszahlengenerator mit dem gleichen Anfangswert benutzt. Sofern die Daten gleich bleiben, werden dadurch die identischen Positionen berechnet.

Im nächsten Schritt werden die Akteure aus R_2 positioniert. Dort sind alle übrigen Akteure aus Deutschland oder dem Ausland untergebracht. Für das Begrenzungs-polygon wird das vorher berechnete Polygon P_1 nochmals vergrößert. Die Berechnung der Positionen aller Knoten in dieser Region läuft dann analog zu denen aus R_1 ab.

3.1. Bestimmung der geographischen Koordinate

Die verwendeten Datensätze über Postleitzahlen und ihre zugehörigen geographischen Koordinaten stammen von OpenGeoDB [Ope11], einem offenen Projekt zur Ermittlung von Geodaten. Die verwendeten Daten beschränken sich jedoch nur auf Deutschland und beinhalten über 8200 Datensätze. Dort sind nicht alle deutschen Postleitzahlen erfasst. Insbesondere Firmen und Institutionen mit einer eigenen Postleitzahl sind darin nicht integriert.

Falls dies zu Problemen führte, wurde eine beliebige andere Postleitzahl des gleichen Standortes des Akteurs gewählt. Beispielsweise wurde statt der firmeneigenen Postleitzahl, die der Stadt gewählt, in der die Firma ihren Sitz hat. Für ausländische Firmen wurde ein Ort an der deutschen Grenze gewählt um zumindest die grobe geographische Richtung vorgeben zu können. Da diese Akteure der Region R_2 zugeordnet werden, reicht dieses Vorgehen auch völlig aus. Dort kommt es ohnehin nicht auf geographische Genauigkeit an.

In den Datensätzen gibt es zu jeder Postleitzahl die zugehörigen Breiten- und Längengrade (φ bzw. λ). Es gibt in der Kartografie viele Projektionen um daraus eine Karte zu machen. Dabei versucht jede Projektion bestimmte Eigenschaften wie Flächen- und Entfernungstreue zu optimieren. Eine Projektion, die dabei jede Eigenschaft optimal umwandelt gibt es nicht. Bei der Implementierung wurde die Mercatorprojektion angewendet. Sie gehört zur Klasse der Zylinderprojektionen und ist folgendermaßen definiert:

$$\begin{aligned} x(\lambda) &= \lambda - \lambda_0 \\ y(\varphi) &= \frac{1}{2} \cdot \ln \left(\frac{1 + \sin(\varphi)}{1 - \sin(\varphi)} \right) \end{aligned}$$

Dabei müssen λ und φ im Bogenmaß vorliegen. Mit λ_0 kann das Kartenzentrum festgelegt werden. Für die Zeichnung ist es hier aber im Prinzip egal, wo sich das Zentrum befindet. Karten, die mit Hilfe der Mercatorprojektion erstellt werden, sind richtungs- und winkeltreu. Der Nachteil dieser Projektion zeigt sich erst dann, wenn man Karten in der Nähe der Pole darstellen will. Dort verzerrt sich die Fläche sehr stark und Grönland wird dadurch beispielsweise ähnlich so groß wie Afrika abgebildet. Mehr zu Kartenprojektionen findet man in [BS95]. Mit Hilfe der Mercatorprojektion wird dann jedem Akteur $a_i \in A$ ($i = 1, \dots, n$) die Position

$$pos_i = (s \cdot x(\lambda_i) | s \cdot y(\varphi_i)) \quad (3.1)$$

zugewiesen. Die Koordinaten werden dabei noch mit einem skalaren Faktor s multipliziert. Dies entspricht einer Änderung des Maßstabs. Der Faktor wird in Abhängigkeit von der Anzahl der Akteure aus A_0 und der Fläche des Bundeslandes P_0 gewählt (Gl. 3.3).

3.2. Bestimmung der Größe der Begrenzungspolygone

3.2.1. Bundesland (Begrenzungspolygon für R_0)

Die Größe des Bundeslandpolygons richtet sich nach drei Orten und hängt deshalb von der Berechnung der geographischen Koordinaten ab. Die Berechnung wiederum richtet sich aber auch nach der Fläche des Bundeslandpolygons (Faktor s aus Gl. 3.1). Damit wird sichergestellt, dass genügend Platz für Knoten innerhalb des Bundeslandes ist. Um beides zu bestimmen wird zunächst ein initialer Wert s_0 gewählt und die daraus resultierende Fläche des Bundeslandes $\Gamma_{s_0}(P_0)$ bestimmt. Mit $\Gamma(\cdot)$ wird die Fläche eines Polygons bezeichnet. Die Größe soll sich am Ende nach der Anzahl der Akteure aus A_0 und der Fläche, die diese einnehmen, orientieren:

$$F(A_0) = |A_0| \cdot F(u) \cdot \tau \quad (\tau \geq 1) \quad (3.2)$$

Wenn man nun

$$s = \sqrt{\frac{F(A_0)}{\Gamma_{s_0}(P_0)}} \quad (3.3)$$

wählt und daraus die Koordinatenberechnung neu bestimmt, dann gilt am Ende $F(A_0) = \Gamma_s(P_0)$. Der Faktor τ wurde experimentell bestimmt. Er kann allerdings auch vom Benutzer gewählt werden, um damit die Ausgabegröße des Graphen zu bestimmen. Durch ihn ordnet man einem Knoten mehr Raum zu als er rein flächenmäßig eigentlich benötigt.

3.2.2. Begrenzungspolygone von R_1 und R_2

Den gleichen Ansatz könnte man auch für die beiden anderen Regionen R_1 und R_2 nutzen. Zunächst wird die konvexe Hülle des Bundeslandes berechnet und mit einer zentrischen Streckung vergrößert. Mit der Vergrößerung versucht man ausreichend Platz für die Knoten und ihre Beschriftungen zu bieten. Allerdings werden, wie später beschrieben (vgl. Kapitel 6), teilweise noch Beschriftungen aus der Region R_i in R_{i+1} verschoben, weshalb für diese Beschriftungen zusätzlicher Platz benötigt wird. Man erhält für das Begrenzungspolygon P_1 von R_1 durch

$$q = (\alpha \cdot |A_1| + \beta \cdot |A_0|) \cdot F(u) \cdot t + \Gamma(P_0) \quad (3.4)$$

$$d = \sqrt{\frac{q}{\Gamma(P_0)}} \quad (3.5)$$

den Streckungsfaktor d . Mit t kann man hier ähnlich wie τ aus Gleichung 3.2 einem Knoten mehr Platz zuordnen. Damit sollte nach Anwendung einer zentrischen Streckung um d , die Fläche des Polygons groß genug sein um ausreichend Platz für die enthaltenen Knoten und Beschriftungen zu bieten. Die Koeffizienten α und β approximieren dabei die notwendige Mindestfläche. Wenn man davon ausgeht, dass ein Beschriftungsknoten etwa das sechsfache der Fläche eines normalen Knotens einnimmt und die Beschriftungen von Akteuren aus A_i zur einen Hälfte in der gleichen Region R_i und zur anderen Hälfte nach R_{i+1} verschoben werden, dann erhält man:

$$\alpha = (1 + 6 \cdot 0,5) = 4 \quad \text{und} \quad \beta = 6$$

Das Problem bei dieser Methode ist die wenig intuitive Bestimmung von t . Bei einer zentrischen Streckung mit Streckungsfaktor d vergrößert sich die Fläche um das d^2 -fache. Dennoch kann dabei der zur Verfügung stehende Platz für die Knoten gering ausfallen. In Abbildung 3.2a ist ein Beispiel mit der fünffachen Minimalgröße für jeden Knoten bzw. Beschriftung gezeigt ($t = 5$). Wie man sehen kann gibt es kaum genug Platz für alle Knoten. Würde man nun noch Beschriftungen einzeichnen gäbe es vermutlich Überschneidungen mit anderen Knoten. Erst mit der 15-fachen Minimalgröße ist die Region groß genug um auch noch Platz für Beschriftungen zu bieten. Die Schwierigkeit liegt dabei in der Abschätzung der zusätzlichen Fläche, die bei einer zentrischen Streckung erzeugt wird.

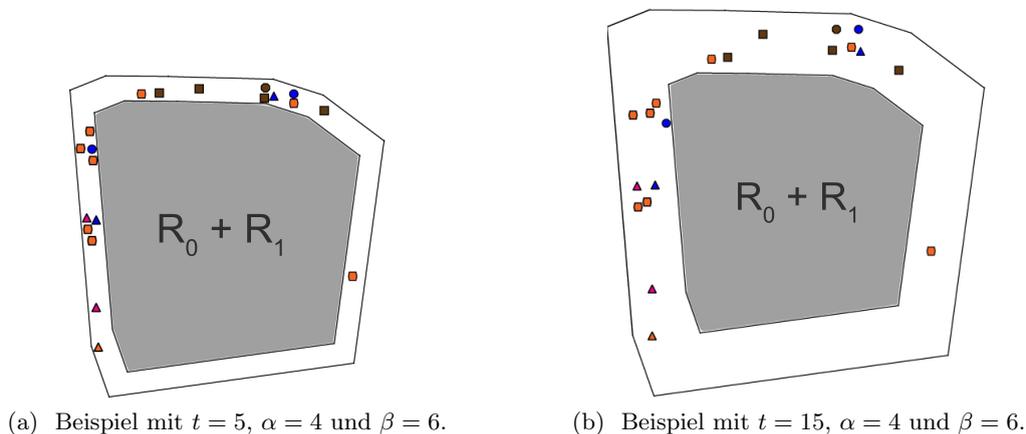


Abbildung 3.2.: Berechnung der Größe von R_2 mit Gleichung 3.5.

Da man nicht davon ausgehen kann, dass die Knoten innerhalb einer Region gleich verteilt sind, lässt sich t auch nicht so einfach automatisch bestimmen. Es ist also notwendig,

dass der Benutzer selbst noch etwas ändern kann und dafür sollte man besser erkennen können, wie sich die Änderung auswirkt. Deshalb wird die Größe der Begrenzungspolygone P_1 und P_2 nicht in Abhängigkeit von der Fläche bestimmt, sondern von der Breite. Der Streckungsfaktor für das Begrenzungspolygon $P_i = (p_0, \dots, p_n)$ lässt sich dann wie folgt bestimmen:

$$d = 1 + \frac{F(u) \cdot t}{\left(\sum_{p \in P_{i-1}} |\zeta p|\right) \cdot \frac{1}{n}} \quad (3.6)$$

Es werden alle Abstände zwischen dem Streckungszentrum ζ und jedem Eckpunkt des Polygons P_{i-1} gebildet und daraus der durchschnittliche Abstand berechnet. Mit t bestimmt man dann, wie viele Knoten in einer Region in etwa nebeneinander Platz haben werden. Diese Vorgehensweise wird sowohl für das Begrenzungspolygon von R_1 als auch für das von R_2 benutzt.

3.3. Umriss des Bundeslandes

Die Umrisse aller deutschen Bundesländer liegen in Form von Polygonen in Textdateien bereit. Das entsprechende Polygon wird eingelesen und dem Graph hinzugefügt. Die Schwierigkeit dabei ist die korrekte Platzierung des Polygons relativ zu den Akteuren. Denn für die Bundesländer liegen keine Angaben über die geographische Position vor. Deshalb wurden für alle Bundesländer jeweils drei Orte aus einer Karte gewählt. Zwei davon geben dabei die nördlichste und südlichste Position des Bundeslandes vor. Der dritte Ort beeinflusst die Ost-/West-Verschiebung. Aus den Orten und der Postleitzahl kann dann wieder eine geographische Koordinate ermittelt werden und sowohl die Position als auch der Maßstab ausreichend genau berechnet werden. Eine exakte Berechnung ist für diese Anwendung nicht notwendig. Die Berechnung der Koordinaten wird dabei so gewählt, dass die Fläche des Bundeslandes groß genug ist um alle Knoten zu beinhalten (Abschnitt 3.1).

Auch die Form des Polygons muss nicht besonders detailliert sein und deshalb wird der Douglas-Peucker-Algorithmus verwendet um die Anzahl der Knoten zu verringern.

3.3.1. Douglas-Peucker Algorithmus

Der Douglas-Peucker Algorithmus (DPA) ist ein weit verbreiteter Algorithmus um Linienzüge zu vereinfachen [DP73]. Grundlage ist das Teile und Herrsche-Prinzip, welches in vielen Algorithmen seine Anwendung findet.

Zu Beginn werden die Start- bzw. Endpunkte p_0 und p_n des Eingabepolygons $P = (p_0, \dots, p_n)$ als Strecke betrachtet. Dadurch hat man eine erste einfache Approximation $p_0 p_n$ (Abbildung 3.3b) des Ausgangspolygons. Nun wird der maximale Abstand aller p_j ($0 < j < n$) zu $\overline{p_0 p_n}$ berechnet (Abbildung 3.3c). Falls der Abstand größer als ein vorher festgelegter Grenzwert ϵ ist, wird der gefundene Punkt p_i der Approximation hinzugefügt (Abbildung 3.3d). Anschließend wird rekursiv für jede entstandene Punktmenge $P_0 = (p_0, \dots, p_i)$ und $P_1 = (p_{i+1}, \dots, p_n)$ analog verfahren bis keine weiteren Punkte der Approximation hinzugefügt werden und der Algorithmus terminiert. Die Laufzeit des Algorithmus für diese Variante liegt in $\mathcal{O}(n^2)$.

In der Literatur gibt es zahlreiche Beschreibungen für optimierte Varianten des obigen Schemas um das Ergebnis zu verbessern oder an bestimmte Merkmale anzupassen. Hersberger und Snoeyink [HS92] präsentieren eine verbesserte Version mit einer Laufzeit von $\mathcal{O}(n \cdot \log(n))$. Bei der Implementierung wurde aber darauf verzichtet, da die Resultate

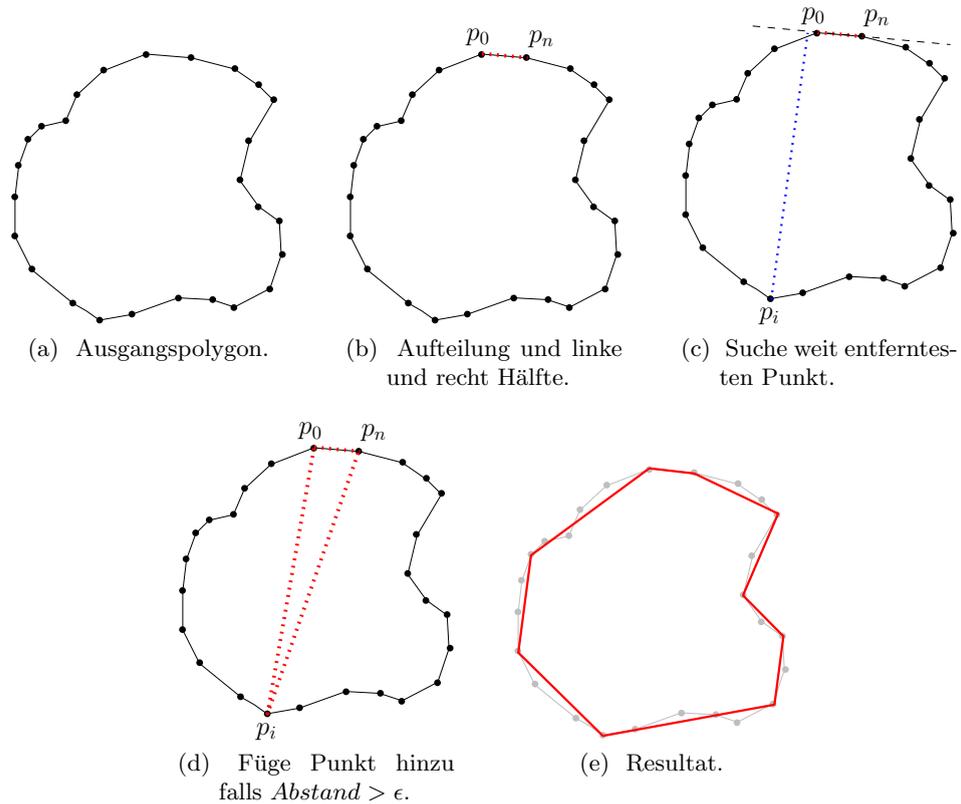


Abbildung 3.3.: Douglas-Peucker Algorithmus.

schon zufriedenstellend sind und die Instanzen von P zu klein in der Anzahl der Knoten sind, um einen Leistungsgewinn erwarten zu können.

Ein grundsätzliches Problem des DPA kann bei verschiedenen Anwendungen der Grenzwert ϵ sein, da dieser vor dem Verfahren manuell festgelegt werden muss. Auch hier gibt es Ansätze um automatisch einen guten Schwellwert zu finden [XID10]. Für die verwendeten Umrisse der Bundesländer reicht es allerdings aus, einen experimentell ermittelten Schwellwert für alle Umrisse zu verwenden, da diese in der Struktur (maximale Abstände) grundsätzlich ähnlich sind.

4. Entzerren von nahen Knoten

Gegeben sei eine Menge V von Knoten, die eine vordefinierte Position haben und sich innerhalb eines Polygons P' befinden. Zusätzlich existiert ein spezieller Knoten z , der als zentraler Knoten angesehen werden kann. Er muss aber nicht notwendigerweise in der Mitte der Zeichnung liegen. Eine wichtige Eigenschaft der Knotenmenge ist das Ansammeln vieler Knoten in der Nähe von z (vgl. Bsp. 4.1). Ziel ist es nun, die Knoten so zu verschieben, dass die Umgebung um z etwas ausgedehnt wird und nahe Knoten voneinander wegbewegt werden. Genauso sollen weit entfernte Knoten von z näher heranrücken. Dabei soll auch das Polygon verzerrt werden um die Verschiebung der Knoten und damit eine Änderung zur Vorgabe deutlich zu machen. Der grobe Umriss des Polygons soll dabei aber beibehalten werden, da er maßgeblich zur besseren Lesbarkeit des Layouts beitragen kann. Eine zu starke Verfälschung kann sonst zur Unkenntlichkeit des Umriss führen.

Die relative Positionierung zwischen den Knoten ist ebenfalls wichtig und sollte durch das Verfahren nicht verändert werden. Der Algorithmus soll also die orthogonale Ordnung aufrecht erhalten [MELS95].

Orthogonale Ordnung. *Angenommen ein Algorithmus ändert im Layout die Position von Knoten u von (x_u, y_u) zu (x'_u, y'_u) . Dann erhält der Algorithmus die orthogonale Ordnung, wenn für jedes Knotenpaar $u, v \in V$ gilt:*

$$\begin{array}{llll} x_u < x_v & \text{iff} & x'_u < x'_v & \wedge \\ y_u < y_v & \text{iff} & y'_u < y'_v & \wedge \\ x_u = x_v & \text{iff} & x'_u = x'_v & \wedge \\ y_u = y_v & \text{iff} & y'_u = y'_v & \end{array}$$

4.1. Ansatz

Eine erste Idee um eine Entzerrung zu erreichen wäre die Nachbildung eines Fischaugenobjektivs. Hierfür bietet es sich an die Koordinaten der Knoten in Polarkoordinaten (δ, r) umzuwandeln und den Radius r mit Hilfe einer passenden Funktion g zu verändern. Der Winkel δ soll dabei konstant gehalten werden. Als Funktionen kommen typische nicht lineare Funktionen in Frage, wie zum Beispiel der Logarithmus oder die m -te Wurzel. Diese stauchen große Entfernungen stärker zusammen als kleinere und bewirken, dass sich weit entfernte Akteure in der Zeichnung näher in Richtung Zentrum verschieben. Damit man

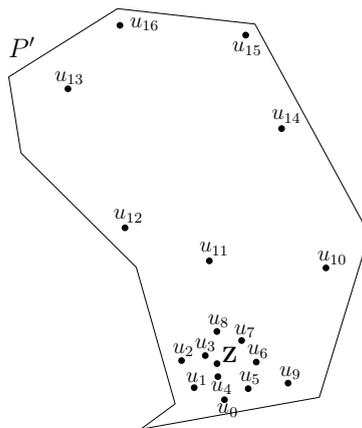


Abbildung 4.1.: Beispiel für eine Eingabemenge $V = (z, u_0, \dots, u_{16})$.

im Gesamten aber keine Verkleinerung bewirkt und dadurch große leere Flächen entstehen, wird das ganze mit einem konstanten Faktor c wieder linear hochskaliert. Dadurch erhält man die Funktion

$$f(\delta, r) = (\delta, g(r) \cdot c).$$

Mit ihr ordnet man einem existierenden Knoten u in der Zeichnung eine neue Position zu. Eine wichtige Eigenschaft der Funktion f ist die Monotonie. Damit ist sichergestellt, dass ein Knoten u nach der Transformation auch näher als ein anderer Knoten v ist, falls er es vorher auch war (bzw. umgekehrt). Wie man leicht einsehen kann trifft die Monotonie bei Verwendung von Polarkoordinaten und den oben genannten Funktionen für f zu, da sowohl die m -te Wurzel als auch der Logarithmus streng monoton steigen.

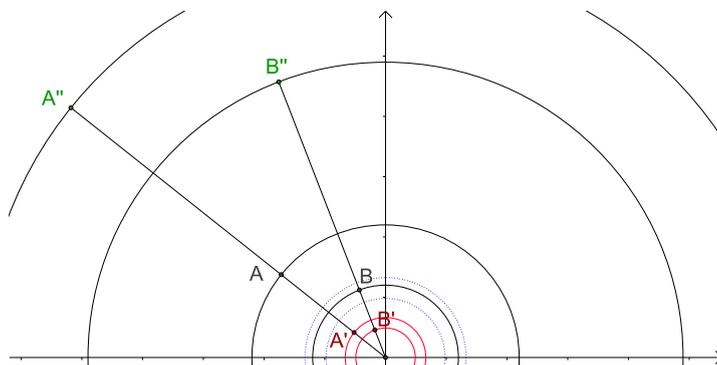


Abbildung 4.2.: A und B zeigen die ursprüngliche Position. Dann wird der Radius geeignet zusammengestaucht (hier: Quadratwurzel). Daraus resultieren die Punkte A' und B' . Am Ende werden die Abstände mit einem skalaren Faktor c wieder vergrößert (A'' bzw. B'').

Ein Problem, welches aber auftreten kann, lässt sich in Abbildung 4.2 erkennen. Dort liegt A ursprünglich noch nördlich von B . Dies ändert sich aber durch das Anwenden der Transformation f . Nun ist das Resultat der Transformation B'' nördlich von A'' . Ähnlich lässt sich auch ein Beispiel für eine Ost-/Westvertauschung konstruieren. Hier wird also die orthogonale Ordnung verletzt. Dieser Effekt kann zu ungünstigen Vertauschungen führen und ist daher unerwünscht. Deshalb werden kartesische Koordinaten verwendet. Diese garantieren während der Transformation keine Vertauschungen hervorzurufen und die Monotonie der Abstände bleibt ebenfalls erhalten. Im kartesischen Koordinatensystem

werden die Knoten in x- und in y-Richtung unabhängig voneinander mit der gleichen Funktion verändert. Deshalb wird die Funktion f in

$$f(x, y) = (g(x) \cdot c, g(y) \cdot c).$$

undefiniert.

Im Folgenden wird dieser erste Ansatz etwas verfeinert, um unter anderem den konstanten Faktor c nicht völlig beliebig wählen zu müssen.

4.2. Grundidee zur Umsetzung

Aufbauend auf den ersten Ansatz werden zunächst Knoten durch eine geeignete Funktion in Richtung Zentrum verschoben. Dabei werden jedoch alle Knoten $u \in V_U \subseteq V$ in einer gewissen Umgebung U um das Zentrum z ignoriert. Die Größe von U könnte von der verwendeten Knotengröße abhängig gemacht werden. Durch diese Unterteilung entstehen acht verschiedene Regionen Υ_j ($j = 0, \dots, 7$), welche in Abbildung 4.3 (a) gekennzeichnet sind. Für jede dieser Region wird eine eigene Funktion benötigt. Anschließend werden nun die Abstände aller Knoten, inklusive der Knoten aus V_U , wieder vergrößert. Nach der Vergrößerung soll die ursprüngliche Größe der Bounding Box des Polygons P' wieder eingenommen werden. Damit füllt man den leeren Platz wieder aus. Die gesamte Skalierung gliedert sich also in zwei Teile und hat den Effekt, dass Knoten innerhalb der Umgebung U , im Verhältnis zum Rest, eine stärkere Vergrößerung erfahren und das Zentrum so entzerrt wird.

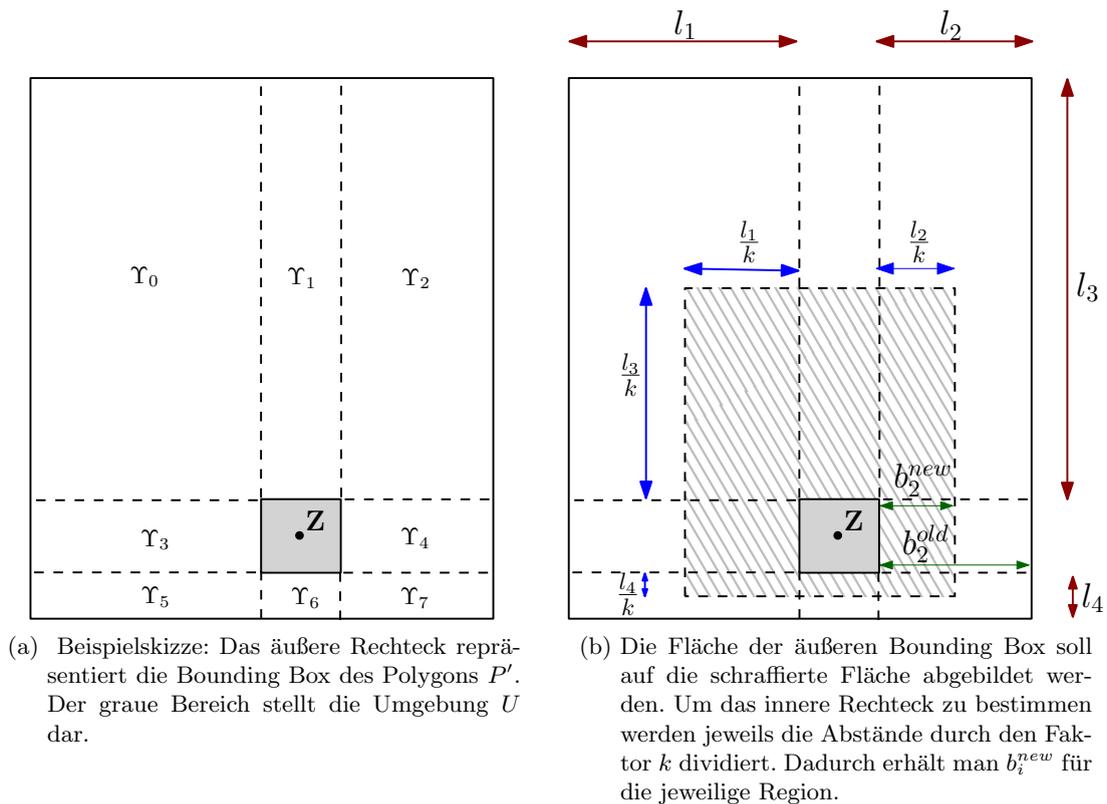


Abbildung 4.3.: Grundidee für die Skalierung.

Da die Knoten nach der Entzerrung nicht mehr an der Position sind, die durch ihre vordefinierte Koordinate eigentlich vorgesehen war, ist es erwünscht das Polygon ebenfalls

mitzuskalieren. So entsteht für die Entzerrung ein weiteres Ziel. Um den groben Umriss des Polygons beizubehalten spielt auch hier die Monotonie der Skalierungsfunktion eine wichtige Rolle. Dadurch wird sichergestellt, dass alle Knoten auch nach der Transformation sich noch innerhalb von P' befinden.

Das erste Ziel soll es sein, weit entfernte Knoten näher an das Zentrum z zu ziehen. Durch die anschließende Vergrößerung der geschrumpften Umgebung erreicht man eine bessere Ausnutzung des zur Verfügung stehenden Raumes. Die Knoten in der Nähe von z erhalten mehr Platz im Vergleich zu den Knoten, die weiter weg sind. Damit dabei der bereits oben genannte konstante Faktor c automatisch bestimmt werden kann, wird zunächst die Bounding Box des Polygons wie in Abbildung 4.3 (b) verkleinert. Die Seitenlängen l_i zwischen der Bounding Box und der Umgebung U werden durch k geteilt und man erhält so eine kleinere Fläche B_k (blau schraffiert). Dadurch erhält man auch die Abstände zwischen B_k und U , die mit $b_i^{new} = \frac{l_i}{k}$ bezeichnet werden (für $i = 1, \dots, 4$). Die Abstände zwischen U und der Bounding Box werden mit $b_i^{old} = l_i$ benannt. Nun sollen alle Knoten und P' genau in diese neue Umgebung B_k verschoben werden. Dies kann man durch eine geeignete Wahl von c erreicht werden. Eine passende Verschiebung ist aber nur möglich, wenn man für jede Richtung ein eigenes c_i berechnet. Die Herleitung sieht wie folgt aus:

$$g(b_i^{old}) \cdot c_i \stackrel{!}{=} b_i^{new} \stackrel{!}{=} b_i^{old} \cdot \frac{1}{k} \quad (4.1)$$

$$\Rightarrow c_i = \frac{b_i^{new}}{g(b_i^{old})} \quad (4.2)$$

Die Funktion $g(x)$ soll weit entfernte Knoten näher zum Zentrum verschieben. Mit Hilfe von c_i werden dann ursprüngliche Randpunkte genau auf die entsprechende Kante der gewünschten Umgebung B_k gestreckt. Damit lassen sich die Funktionen für die verschiedenen Regionen nach folgendem Schema kombinieren (s. Tabelle 4.1):

$$f_j(x, y) = (f_{1,i}(x), f_{2,i'}(y)) = (g(x) \cdot c_i, g(y) \cdot c'_i). \quad (4.3)$$

Region Υ_j	$f_{1,i}(x)$	$f_{2,i'}(y)$
Υ_0	$f_{1,1}(x)$	$f_{2,3}(y)$
Υ_1	$f(x) = x$	$f_{2,3}(y)$
Υ_2	$f_{1,2}(x)$	$f_{2,3}(y)$
Υ_3	$f_{1,1}(x)$	$f(y) = y$
Υ_4	$f_{1,2}(x)$	$f(y) = y$
Υ_5	$f_{1,1}(x)$	$f_{2,4}(y)$
Υ_6	$f(x) = x$	$f_{2,4}(y)$
Υ_7	$f_{1,2}(x)$	$f_{2,4}(y)$

Tabelle 4.1.: Zuordnungen der Funktionen $f_{1,i}(x)$ und $f_{2,i'}(x)$ zu den Regionen Υ_j .

Für g kommen zwei herkömmliche Funktionen in Frage, deren Eigenschaften und Eignung im Folgenden beleuchtet werden.

Wurzelfunktion

Um die geforderten Eigenschaften zu erfüllen wurde folgende Funktion angewendet:

$$g(x) := \sqrt[m]{d(x)} \quad (4.4)$$

$$c_i = \sqrt[m]{(b_i^{old})^{m-1}} \cdot \frac{1}{k}$$

Dabei bezeichnet $d(x)$ den Abstand zur jeweiligen Kante der Umgebung U . In Abbildung 4.4 wird beispielhaft die Anwendung obiger Funktion gezeigt. Die zugehörige Wertetabelle ist in Tabelle 4.2 dargestellt. Man erkennt, dass Knoten, die etwas weiter vom Zentrum z entfernt sind, näher an z heranrücken. Umgekehrt werden Knoten in der Nähe von z etwas nach außen gedrückt. Knoten innerhalb von U behalten natürlich ihren Abstand bei, da sie, wie bereits beschrieben, für die Verkleinerung zunächst ignoriert werden.

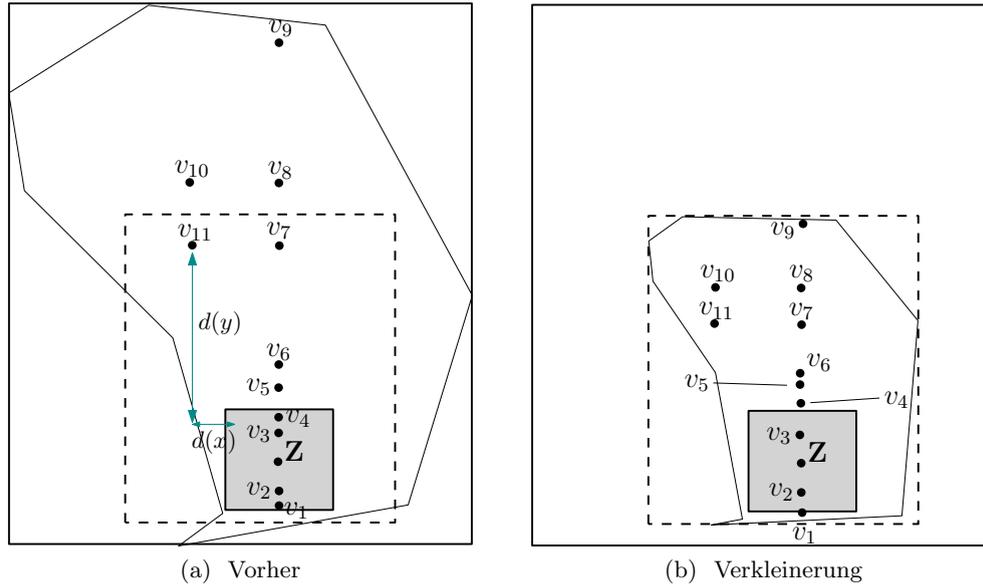


Abbildung 4.4.: Einige Knoten, die durch eine Funktion wie in Gl. 4.4 transformiert wurden. Auch das Polygon P' wird auf die gleiche Weise verkleinert, so dass es in die Umgebung U passt. Die Abstände zwischen U und Bounding Box wurden hier halbiert ($k = 2$).

Knoten	Vorher	Verkleinern mit $f(x, y)$		Expandieren mit $h(x, y)$		
	Abstand d_v	Abstand d_n	$\frac{d_n}{d_v}$	Abstand d'_n k = 2	$\frac{d'_n}{d_v}$ k = 2	$\frac{d''_n}{d_v}$ k = 4
v_1	9,0	10,07	1,12	15,58	1,72	2,06
v_2	5,0	5,0	1,0	7,7	1,54	2,11
v_3	5,0	5,0	1,0	9,34	1,87	3,3
v_4	10,0	15,35	1,53	28,67	2,87	3,77
v_5	30,0	31,05	1,04	57,99	1,93	2,12
v_6	34,0	33,05	0,97	61,74	1,82	1,97
v_7	52,0	40,61	0,78	75,87	1,46	1,53
v_8	65,0	45,14	0,69	84,32	1,30	1,34
v_9	98,0	54,72	0,56	102,22	1,04	1,05
v_{10}	70,77	51,75	0,73	95,90	1,35	1,41
v_{11}	59,0	47,85	0,81	88,55	1,50	1,58

Tabelle 4.2.: Übersicht über die Abstände einiger Knoten zu z bei Verwendung von $f(x, y) = (g(x) \cdot c_i, g(y) \cdot c'_i)$ mit $g(x) := \sqrt{d(x)}$ für die Verkleinerung und $h(x, y)$ (Gl. 4.10) für die Expansion. Die Bezeichnung der Knoten bezieht sich dabei auf Abbildung 4.4.

Eine genauere Analyse zeigt, dass für die Skalierung drei relevante Regionen entstehen (vgl. Abbildung 4.5). In der folgenden Gleichung bezeichnet z wie vorher auch das Zentrum und

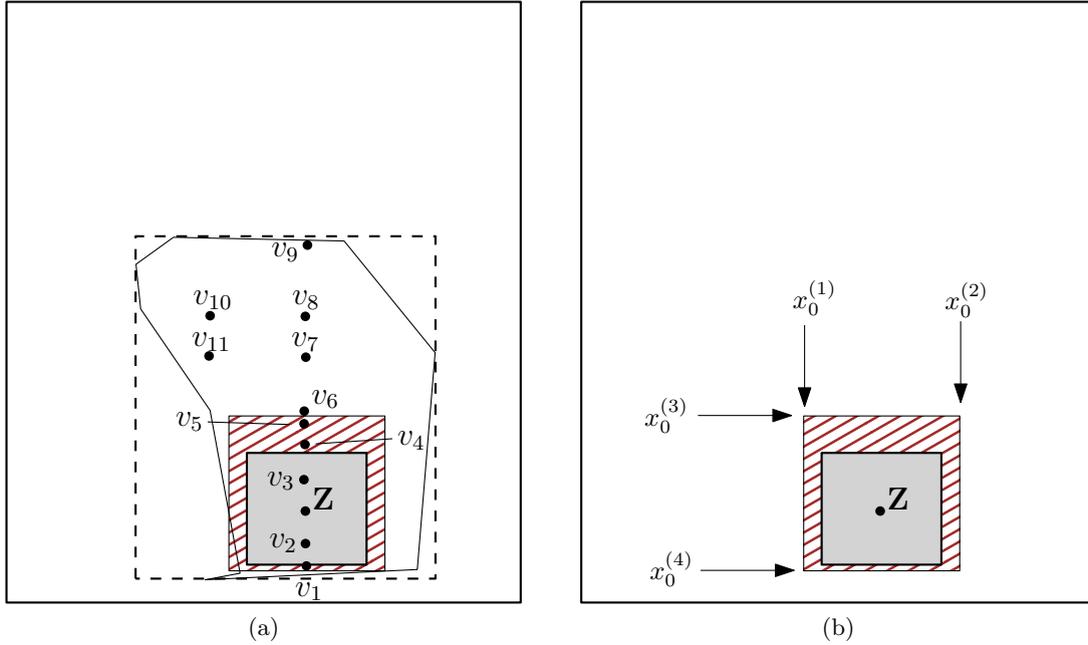


Abbildung 4.5.: Die rot schraffierte Box entsteht durch die Fixpunkte $x_0^{(i)}$, die es für jede Seite gibt. Bei allen Punkten innerhalb dieser Umgebung vergrößert sich der Abstand zum Zentrum, außerhalb verhält es sich umgekehrt.

\mathbb{V} sei die Menge aller Knoten, die für die Transformation zugelassen sind ($\mathbb{V} = V \setminus V_U$). Dies trifft also auf alle Knoten zu, die sich innerhalb des Polygons befinden, aber außerhalb der Umgebung U liegen. Dann ergibt sich für die m -te Wurzel und für jedes $i \in \{1, \dots, 4\}$ ein Fixpunkt $x_0^{(i)} = \sqrt[m-1]{c_i^m}$. Dieser Fixpunkt beeinflusst folgende Eigenschaft:

$$\forall v \in \mathbb{V} : |\bar{v}z| \leq \sqrt[m-1]{c_i^m} \Rightarrow f_{o,i}(|\bar{v}z|) \geq |\bar{v}z| \quad (4.5)$$

$$\forall v \in \mathbb{V} : |\bar{v}z| > \sqrt[m-1]{c_i^m} \Rightarrow f_{o,i}(|\bar{v}z|) < |\bar{v}z| \quad o \in \{1, 2\} \quad (4.6)$$

Dies hat zur Folge, dass alle Punkte mit einer kleineren Entfernung zum Zentrum als $x_0^{(i)}$ vom Zentrum wegbewegt werden (Gl. 4.5). Umgekehrt rücken die ursprünglich weiter entfernten Knoten näher an z heran (Gl. 4.6). An dem gezeigten Beispiel lässt sich auch erkennen, dass v_4 im Verhältnis zu v_5 stärker nach außen gedrängt wird, obwohl beide vor der Transformation im rot schraffierten Bereich sind. Das heißt ebenfalls, dass je näher der Punkt an z , desto stärker wird er von z auch entfernt. Also wird genau der Effekt erreicht, der gewünscht ist. Zusammenfassend gesagt, wandern alle Punkte aus \mathbb{V} in Richtung der Kanten der rot schraffierten Box (Abbildung 4.5).

Nun bleibt noch zu klären welche Wurzelfunktion am geeignetsten scheint. An Hand von Beispielen stellt man fest, dass bei einer stärkeren Wurzel die oben beschriebenen Eigenschaften verstärkt werden. Betrachtet man das Grenzwertverhalten für den entscheidenden Fixpunkt $x_0^{(i)}$

$$\begin{aligned} \lim_{m \rightarrow \infty} x_0^{(i)} &= \lim_{m \rightarrow \infty} \sqrt[m-1]{c_i^m} = \lim_{m \rightarrow \infty} c_i^{\frac{1}{1-\frac{1}{m}}} = c_i \\ \lim_{m \rightarrow \infty} c_i &= \lim_{m \rightarrow \infty} \sqrt[m]{(b_i^{old})^{m-1}} \cdot \frac{1}{k} = b_i^{old} \cdot \frac{1}{k} = b_i^{new} \end{aligned} \quad (4.7)$$

erkennt man, dass dieser der vorher festgelegte Bereich ist, auf den verkleinert werden soll.

Da immer noch die gleichen Eigenschaften wie oben gelten, hat dies zur Folge, dass alle Punkte gegen die Begrenzung konvergieren.

Dadurch entsteht zum einen der Nachteil eines zu stark verfälschten Umriss des Polygons. Noch schwerwiegender ist das Ansammeln aller Knoten in Richtung der Begrenzung. Ein Entzerren wird dadurch also nicht erreicht. Deshalb sollte eine möglichst niedrige Wurzelfunktion benutzt werden. In Beispielen hat die Quadratwurzel zufriedenstellende Ergebnisse geliefert.

Natürlicher Logarithmus

Die zweite mögliche nicht lineare Funktion wäre der Logarithmus.

$$g(x) := \ln(d(x))$$

$$c_i = \frac{b_i^{old}}{k \cdot \ln(b_i^{old})}$$

Auch hier gibt es wieder einen Fixpunkt, den man allerdings nicht mit gängigen Funktionen berechnen kann.

$$s(x) := x \cdot e^x \tag{4.8}$$

$$LW(x) := s^{-1}(x) \tag{4.9}$$

$LW(x)$ ist die lambertsche W-Funktion, definiert als die Umkehrung von Gleichung 4.8. Mit einem Computeralgebrasystem erhält man den Fixpunkt $x_0^{(i)} = -c_i \cdot LW(-\frac{1}{c_i})$.

Insgesamt betrachtet weist diese Funktion ähnliche Eigenschaften wie die Wurzelfunktion auf, hat aber zusätzlich noch eine Besonderheit. In Abbildung 4.6 werden die Verläufe beider Funktionen gegenübergestellt. Zu beachten ist dabei, dass der Funktionsgraph nicht die jeweilige Funktion, sondern das Verhältnis der verwendeten Funktion im Vergleich zur Eingabe wiedergibt. Man erkennt, dass es sogar zwei Fixpunkte gibt. Zwischen ihnen wächst der Abstand aller Knoten zum Zentrum z . In dem übrigen Bereich wird er kleiner. Je nach Definition von U muss der erste Fixpunkt aber für die gewünschte Anwendung keine Auswirkung haben. Da die Knoten dort für die Skalierung ignoriert werden, spielt es keine Rolle wie sich die Funktion in der Nähe von 0 verhält. Wenn allerdings die Umgebung U sehr klein ist, werden nahe Knoten sogar noch näher zu z bewegt, was natürlich nicht gewünscht ist. Um Abhilfe zu schaffen könnte man die Funktion durch zusätzliche Parameter noch etwas manipulieren. Beispielsweise könnte man einen konstanten Wert auf die Eingabe addieren um die Funktion etwas nach links „zu schieben“.

Der weitere Verlauf zeigt ähnliche Eigenschaften wie die Quadratwurzel, jedoch mit stärkerer Wirkung. Dies kann wieder negativen Einfluss auf den Umriss des Polygons haben und ihn zu stark verfälschen (s. Abschnitt 4.4).

Insgesamt betrachtet scheint also die Quadratwurzel die bessere Wahl zu sein und wird deshalb auch bei der Implementierung verwendet.

4.2.1. Umgebung U

Die Größe von U hat ebenfalls eine Auswirkung auf das Ergebnis obiger Funktionen. Insbesondere wird durch sie der Parameter c_i beeinflusst (Gl. 4.2), welcher wiederum direkt den Fixpunkt $x_0^{(i)}$ festlegt, der die Grenze zwischen Ausdehnen und Verkleinern vorgibt. Im Hinblick auf die gezeigte Analyse wäre es somit günstig c_i möglichst groß werden zu lassen, dann wird auch die Umgebung, in der die größtmögliche Vergrößerung herrscht, ebenfalls größer. Um dies zu erreichen sollte b_i^{old} möglichst groß werden und k klein bleiben. Aus diesem Grund wurde bei der Implementierung die Projektumgebung sehr klein gemacht, denn dadurch wird b_i^{old} größer. Man hat also im Vergleich zu Abbildung 4.3a im Wesentlichen nur noch vier und keine acht Regionen mehr.

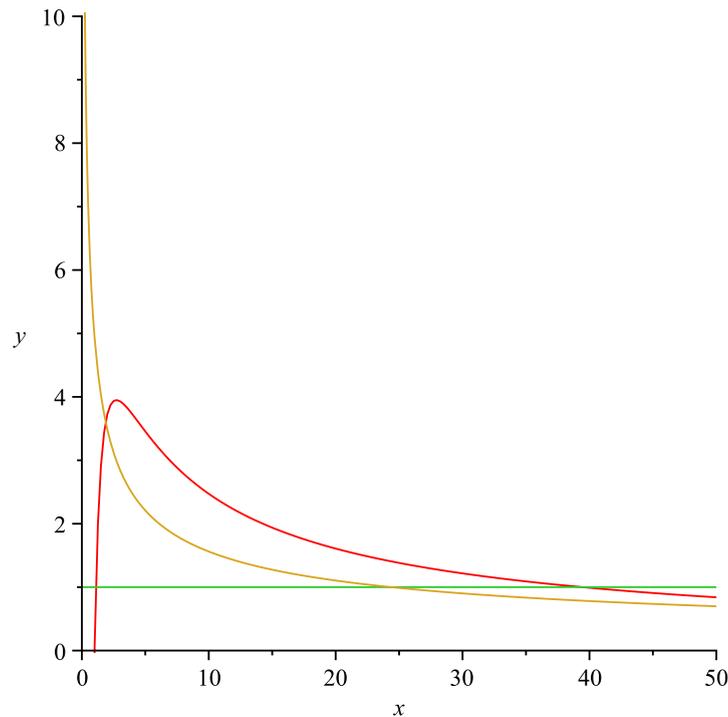


Abbildung 4.6.: Der rote Funktionsgraph zeigt die Funktion $\frac{\ln(x) \cdot c_i}{x}$. Zum Vergleich zeigt die braune Kurve die Funktion $\frac{\sqrt{x} \cdot c_i}{x}$. Man kann der Zeichnung also das Verhältnis zur Eingabe entnehmen. D.h. für den Bereich oberhalb von $y = 1$ (grün) werden Knoten vom Zentrum wegbewegt. Für den Fall unterhalb der Kurve gilt die Umkehrung.

4.3. Expandieren auf ursprüngliche Größe

Um die Entzerrung der Knoten abzuschließen wird die gesamte Umgebung wieder auf ihre ursprüngliche Größe ausgedehnt. Dieses mal sind also auch Knoten aus V_U betroffen. Die Ausdehnung geschieht nach folgendem Schema:

$$h_j(x, y) = (h_{1,i}(x), h_{2,i'}(y))$$

$$h_{o,i}(x) = \frac{f_{o,i}(x)}{b_i^{new}} \cdot b_i^{old} = \frac{f_{o,i}(x)}{\frac{1}{k} \cdot b_i^{old}} \cdot b_i^{old} = f_{o,i}(x) \cdot k \quad o \in 1, 2$$

Es sei darauf hingewiesen, dass diese Formel hier etwas vereinfacht angegeben wurde, um deren Eigenschaften leichter überprüfbar zu machen. Bei der Implementierung waren teilweise noch einige Besonderheiten zu beachten, die sich aber nicht auf folgende Betrachtung auswirken. Beispielsweise muss die Größe der Umgebung U miteingerechnet werden. Dennoch lässt sich daran schon einmal erkennen, dass Knoten innerhalb von U um maximal das k -fache gestreckt werden (da $f_{o,i}(x) = x$). Dies wird aber nur für den Fall erreicht, wenn U eine kleine Epsilonumgebung um z ist. Entgegen der Schlussfolgerung aus Abschnitt 4.2.1 würde der gesamten Transformation ein größeres k helfen, um größere Abstände zu erreichen. Mit Hilfe der letzten beiden Spalten aus Tabelle 4.2 lässt sich das an einem konkreten Beispiel erkennen. Letztendlich wird eine Kompromisslösung gewählt.

An Hand der folgenden Gleichungen lässt sich eine weitere wichtige Eigenschaft der nun kompletten Transformation erkennen ($k > 1$).

$$\forall v \in V_U : f_{o,i}(|\bar{vz}|) \cdot k = |\bar{vz}| \cdot k \Rightarrow h_{o,i}(|\bar{vz}|) > |\bar{vz}| \quad (4.10)$$

$$\forall v \in \mathbb{V} \wedge |\bar{vz}| \leq x_0^{(i)} : f_{o,i}(|\bar{vz}|) \cdot k \stackrel{Gl. 4.5}{>} |\bar{vz}| \cdot k \Rightarrow h_{o,i}(|\bar{vz}|) > |\bar{vz}| \quad (4.11)$$

$$\begin{aligned} \forall v \in \mathbb{V} \wedge |\bar{vz}| > x_0^{(i)} : b_i^{old} > |\bar{vz}| &\iff \sqrt{b_i^{old}} > \sqrt{|\bar{vz}|} \\ &\iff \sqrt{|\bar{vz}|} \cdot \sqrt{b_i^{old}} > |\bar{vz}| \\ &\iff \frac{\sqrt{|\bar{vz}|} \cdot \sqrt{b_i^{old}}}{k} > \frac{|\bar{vz}|}{k} \\ &\iff \sqrt{|\bar{vz}|} \cdot c > \frac{|\bar{vz}|}{k} \\ &\iff f_{o,i}(|\bar{vz}|) > \frac{|\bar{vz}|}{k} \\ &\iff f_{o,i}(|\bar{vz}|) \cdot k > |\bar{vz}| \\ &\Rightarrow h_{o,i}(|\bar{vz}|) > |\bar{vz}| \end{aligned} \quad (4.12)$$

Daraus lässt sich ableiten, dass durch die komplette Transformation h alle Knoten vom Zentrum wegbewegt werden. Dies ist zwar nicht optimal, aber da das an diese Transformation anschließende Verfahren (s. Kapitel 5) die Positionen der Knoten erneut verändert, spielt es für das Gesamtergebnis keine entscheidende Rolle.

Wichtig ist aber noch die Erkenntnis, dass diese Eigenschaft verlangt das Polygon nicht nur aus den bereits genannten Gründen mitzuskalieren. Da alle Knoten nach außen wandern, könnten sie das Polygon verlassen, sofern es in einem Bereich deutlich kleiner als die Bounding Box ist. Denn dort ist die Vergrößerung zu stark als dass die Knoten noch innerhalb des Polygons bleiben würden.

4.4. Auswirkung auf den Umriss des Polygons

Wie man in Abbildung 4.7 erkennt hat der Parameter k kaum nennenswerte Auswirkungen auf den Umriss nach der Skalierung (vgl. 4.7b + 4.7c und 4.7f + 4.7g). Daher lässt sich im Hinblick auf die zu erwartende stärkere Vergrößerung k auch etwas größer wählen ohne negative Auswirkungen auf den Umriss befürchten zu müssen.

Gravierender wirkt sich jedoch die verwendete Funktion $g(x)$ aus. Der natürliche Logarithmus (4.7d und 4.7h) sorgt dafür, dass sich der Umriss des Polygons an die Bounding Box anschmiegt. Verglichen damit ist die Wirkung der Quadratwurzel etwas gutmütiger und eignet sich folglich auch etwas besser.

Es lassen sich aber auch Beispiele konstruieren, bei denen je nach Lage von z im Ergebnis starke Einkerbungen auftreten. Diese erwecken einen unschönen Gesamteindruck und werden deshalb versucht zu entfernen (Abbildung 4.8d). Sie entstehen, wenn das Polygon mehrmals die Koordinaten des Zentrums kreuzt (vgl. Abbildung 4.8a). Durch das Expandieren (Abschnitt 4.3) werden solche Kurven in gegensätzliche Richtungen gestreckt und dadurch verstärkt. Um das Auftreten solcher Einkerbungen zu verhindern, wird das Polygon vor der Skalierung verändert. Dazu werden die Knoten, die für eine Einkerbung sorgen können, zunächst identifiziert (Abbildung 4.8c). Ob die Knoten dann entfernt werden hängt davon ab, ob sie wichtig für den Umriss sind. Um das festzustellen wird als erstes der maximal entfernte Knoten der Kurve zu der Mittellinie gesucht. Dadurch erhält man die Länge a . Des Weiteren wird die Fläche berechnet, den dieser Teil des Polygons einnimmt. Sollte die Fläche unter einem vorher festgelegten Schwellwert liegen und a

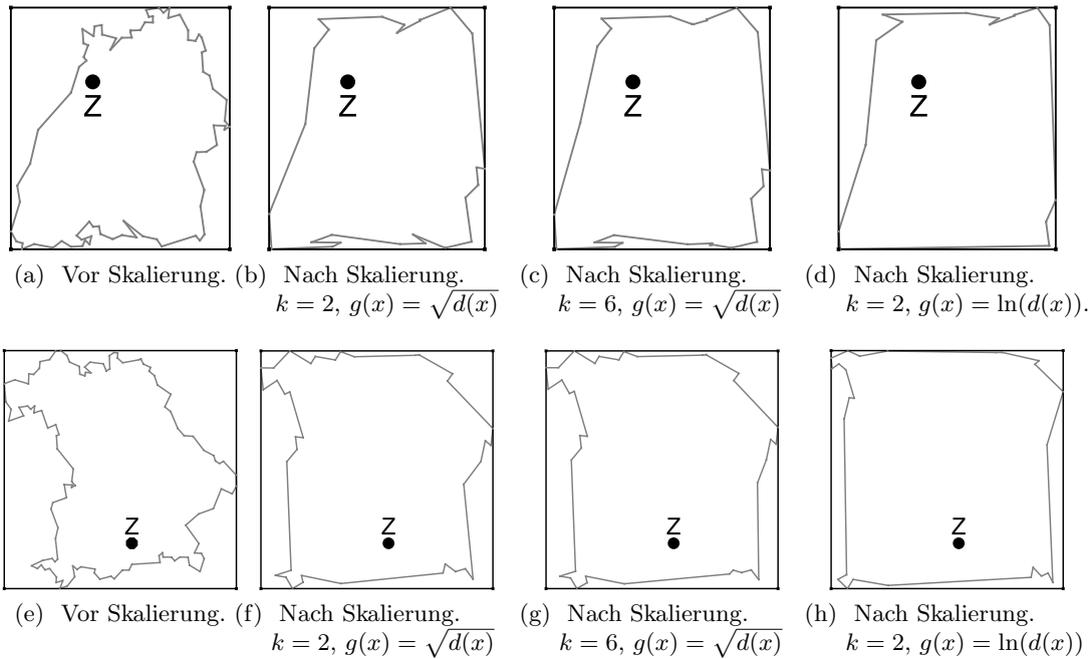


Abbildung 4.7.: Auswirkung der Skalierung auf die Bundesländer Baden-Württemberg ((a) – (d)) und Bayern ((e) – (h)).

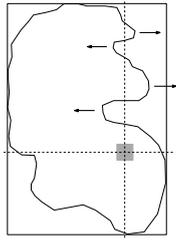
ebenfalls unter einem zweiten Schwellwert liegt, wird die Kurve entfernt. Die Schwellwerte werden in Abhängigkeit der Knotengröße so definiert, dass dadurch nur kleine Teile aus dem Polygon entfernt werden, die sonst zu den beschriebenen Einkerbungen führen. Wenn man das Verfahren auf Abbildung 4.8d anwendet erhält man als Resultat Abbildung 4.8e.

4.5. Resultat

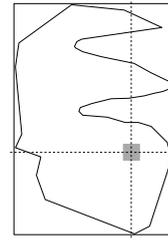
An Beispiel 4.9 lässt sich erkennen, wie sich obiges Verfahren auf die Knoten auswirkt. Die vielen sich überlappenden Knoten im unteren Bereich von 4.9a wurden etwas voneinander entfernt. Man könnte dies noch etwas verbessern, wenn man kein vorgegebenes Zentrum z als Eingabe erhält, sondern den Schwerpunkt einer solchen Knotenanhäufung als Zentrum wählt. Dadurch wird sichergestellt, dass sich die Knoten in alle Richtungen um z ausbreiten können. Im gezeigten Beispiel ist das nicht der Fall. Hier werden die Knoten um z im Wesentlichen nur in eine Richtung gedrückt, da sie im Vergleich zu z alle eine ähnliche Lage aufweisen. Mit einer Änderung wäre also eine ausgeglichene Entzerrung gegeben. Hier ist es allerdings erwünscht das Projektzentrum z nicht zu ändern. Am Ende wird auf das Polygon nochmal der Douglas-Peucker-Algorithmus angewendet. Die Überlappungen, die weiterhin auftreten, müssen aber durch ein weiteres Verfahren entfernt werden. Dies ist in Kapitel 5 beschrieben.

4.6. Alternative Variante der Expansion

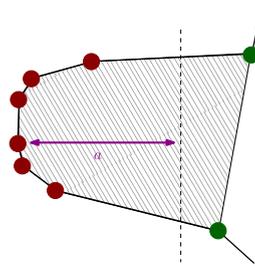
Das beschriebene Verfahren wurde für die Anwendungsfälle noch leicht abgeändert. Während der Entwicklungsphase hat sich herausgestellt, dass es aus ästhetischen Gründen sinnvoller ist, wenn sich der Projektstandort im Zentrum der Zeichnung des sozialen Netzwerks befindet. Um dies umzusetzen wird nach der Verkleinerung (Abschnitt 4.2) die komplette geschrumpfte Umgebung so verschoben, dass sich das Zentrum z in der Mitte des Polygons befindet (vgl. Abbildung 4.10).



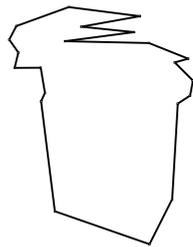
(a) Durch das mehrmalige Kreuzen der Mittellinie werden beim Vergrößerungsschritt aus kleineren Kurven tiefe Einkerbungen im Polygon. Hier wird das Polygon von der Skalierung gezeigt.



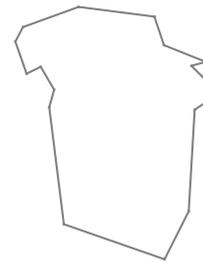
(b) Nach der Skalierung sind ausgeprägte Einkerbungen zu sehen.



(c) Sowohl der maximale Abstand a und die schraffierte Fläche werden berechnet. Sind beide unter vorher festgelegten Schwellwerten, werden die rot markierten Knoten aus dem Polygon gelöscht.

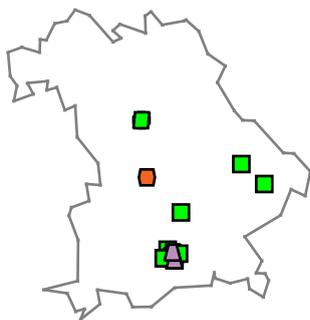


(d) Beispiel für unschöne Einkerbungen.

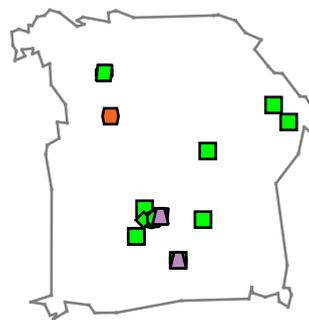


(e) Das Verfahren mit den Schwellwerten aus 4.8c angewendet auf 4.8d.

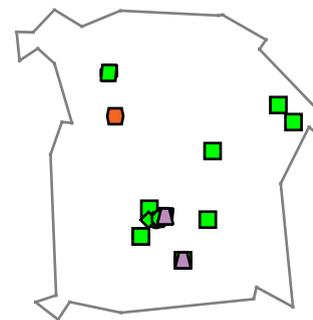
Abbildung 4.8.: Negative Auswirkung der Streckung auf das Polygon.



(a) Beispiel vor Skalierung.



(b) Nach Skalierung ($k = 2$).



(c) Anwendung des Douglas-Peucker-Algorithmus.

Abbildung 4.9.: Resultat

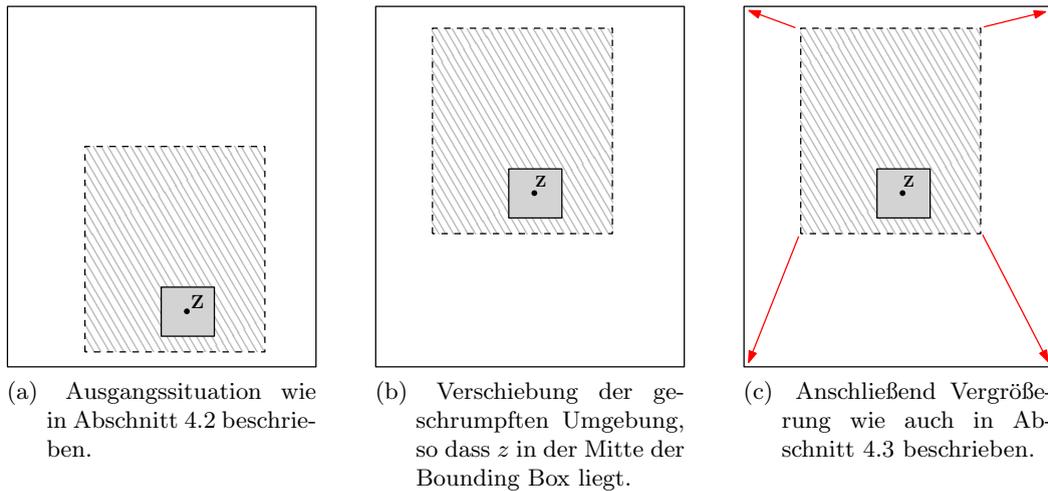


Abbildung 4.10.: Beschreibung der Änderung, die im Vergleich zum obigen Verfahren, bei der Implementierung gemacht wurde.

Dabei kann es im Gegensatz zu vorher vorkommen, dass die geschrumpfte Umgebung außerhalb der Bounding Box liegt. Durch die anschließende Vergrößerung auf die ursprüngliche Größe wird dies aber wieder korrigiert. Bei der Vergrößerung sind im Vergleich zu oben keine weiteren Änderungen mehr notwendig gewesen. Ein positiver Nebeneffekt, der durch das Verschieben in die Mitte erreicht wird, zeigt sich in Kapitel 5. Dort konnte es vorkommen, dass durch eine Lage des Zentrums am Rand, einige Knoten aus dem Polygon heraus gedrückt wurden. Nach der Verschiebung ist um das Zentrum allerdings mehr Platz für die Knoten und dadurch wird dieses Problem verringert.

5. Übereinanderliegende Knoten

Bei den verwendeten Graphen stößt man oft auf das Problem, dass Knoten initial die gleiche Position im Layout haben. Ursache dafür sind gleiche Postleitzahlen bei unterschiedlichen Akteuren. Solchen Akteuren werden dann die gleiche initiale Position zugeordnet. Um das Layout aber insoweit lesbar zu machen, dass man jeden Knoten erkennen kann, ist es notwendig Überlagerungen von Knoten aufzulösen.

Um dies zu erreichen wird ein automatischer Layoutalgorithmus aus der Graphenbibliothek *yFiles* [yWo11] benutzt und passend parametrisiert. Der Algorithmus gehört zur Klasse der kräftebasierte Verfahren [BETT99]. Eine kurze Einführung hierfür liefert Abschnitt 5.1. Verbreitete Implementierungen eines kräftebasierten Verfahrens werden in [KK89] und [FR91] vorgestellt. In Abschnitt 5.3 wird näher auf die Benutzung des verwendeten Algorithmus eingegangen.

5.1. Kräftebasierte Verfahren

In Anlehnung an die Physik wird bei kräftebasierten Verfahren der Graph als Modell verstanden, in dem Kräfte herrschen, die der Algorithmus versucht auszugleichen. So kann man sich Knoten als elektrisch geladene Teilchen vorstellen, die sich gegenseitig abstoßen. Dahingegen verhalten sich Kanten wie Federn, die dafür sorgen, dass sich adjazente Knoten nicht zu weit voneinander entfernen. Der Algorithmus simuliert nun ein System aus den beschriebenen Kräften und gibt am Ende eine Konfiguration des Graphen zurück, bei der die wirkenden Kräfte möglichst minimal sind. Dies geschieht oft durch mehrere Iterationen über alle Knoten, wobei in jedem Durchgang die Knoten nacheinander etwas verschoben werden. Die Verschiebung bewirkt eine Verringerung der Summe aller wirkenden Kräfte. Das Ergebnis entspricht einer Zeichnung, bei der die Knoten einen ausreichenden Abstand zu ihren Nachbarn haben. Ein Beispiel dazu findet sich in Abbildung 5.1(a) und (b). Dort ergab sich eine deutlich klarere Zeichnung mit weniger Kreuzungen. Die auf einen Knoten wirkende Kraft lässt sich mathematisch wie in Gleichung beschreiben:

$$K(v) = \sum_{(u,v) \in E} f_{uv} + \sum_{(u,v) \in V \times V} g_{uv} \quad (5.1)$$

Dabei ist f_{uv} die Federkraft zwischen den Knoten u und v . Die Abstoßungskräfte zwischen u und v werden durch die Summe über g_{uv} berechnet. In [BETT99] wird folgende Berech-

nung der Kräfte für den x -Wert von v mit der Distanzfunktion $d(p_i, p_j)$ vorgeschlagen:

$$f_{uv} = k_{uv}^{(1)}(d(p_u, p_v) - l_{uv}) \cdot \frac{x_0 - x_u}{d(p_u, p_v)} \quad (5.2)$$

$$g_{uv} = \frac{k_{uv}^{(2)}}{(d(p_u, p_v))^2} \cdot \frac{x_v - x_u}{d(p_u, p_v)} \quad (5.3)$$

Die Berechnung des y -Werts erfolgt analog. In der Formel bezeichnet l_{uv} die natürliche Länge der Feder zwischen den Knoten u und v bei der keine Kraft wirkt. Mit $k_{uv}^{(1)}$ wird die Steifheit der Feder modelliert. Je größer $k_{uv}^{(1)}$, desto eher tendieren die Knoten u und v zu einem Abstand von l_{uv} . Schließlich wird mit $k_{uv}^{(2)}$ die Abstoßungskraft zwischen u und v festgelegt.

Ziel ist es nun für das Gesamtsystem

$$\min \left(\sum_{v \in V} K(v) \right)$$

ein, zumindest lokales, Minimum zu finden. Weitere Verfeinerungen sind möglich, z.B. lassen sich unterschiedliche Steifheiten der Federn simulieren, die in Abhängigkeit zu den gewünschten Kantenlängen stehen. Oder es lassen sich unterschiedliche Abstoßungskräfte bei verschiedene Teilmengen der Knoten simulieren.

5.2. Verwendung eines kräftebasierten Verfahrens

Die Zielstellung hier ist es alle Überdeckungen aufzulösen ohne dabei aber die vordefinierte Position zu sehr abzuwandeln. Außerdem sollen die Knoten die ihnen zugewiesene Region nicht verlassen können. Mit Hilfe von so genannten Ankerknoten kann man einen Layoutalgorithmus dazu zwingen eine vordefinierte Knotenposition zu beachten [Lyo92].

Um die erste Bedingung zu erfüllen wird für jeden Knoten u ein Ankerknoten o_u an der gleichen Position erstellt. Dieser darf durch den Algorithmus nicht verschoben werden. Er soll den zugehörigen Knoten u lediglich zu sich heranziehen, damit er sich nicht zu weit wegbewegt. Dies geschieht mit einer Kante (u, o_u) zwischen Anker- und Originalknoten. Für diese Kante sollte die Federkraft f_{uo_u} recht hoch sein und die Abstoßungskraft zwischen u und o_u relativ gering. Ein Beispiel für die Verwendung von Ankerknoten findet sich in Abbildung 5.1(c). Im Vergleich zu (b) ohne Ankerknoten sind die Knoten dort deutlich näher an ihrer ursprünglichen Position. Die Ankerknoten sind also nötig, da sonst die Knoten keinen Bezug zu ihrer Ausgangssituation mehr haben.

Um die Knoten nun noch in ihrer zugewiesenen Region zu halten, sollte nach Möglichkeit dem Algorithmus kenntlich gemacht werden, welche Bereiche für die Positionierung der Knoten möglich sind und welche nicht.

5.3. Benutzung des SmartOrganicLayouter

Auf eine eigene Implementierung eines kräftebasierten Verfahrens wurde verzichtet. Stattdessen wurde der SmartOrganicLayouter aus der *yFiles*-Bibliothek [yWo11] benutzt. Um die Ausgabe nach eigenen Wünschen anzupassen, gibt es einige Einstellmöglichkeiten, die in geeigneter Weise gewählt wurden um das Ergebnis positiv zu beeinflussen.

Dem Algorithmus wird ein Graph G als Parameter übergeben, aus dessen Knotenmenge V alle Knoten verschoben werden dürfen. Mit Hilfe einer weiteren Knotenmenge V' kann man aber Ausnahmen definieren, die der SmartOrganicLayouter nicht bewegen darf. Zusätzlich

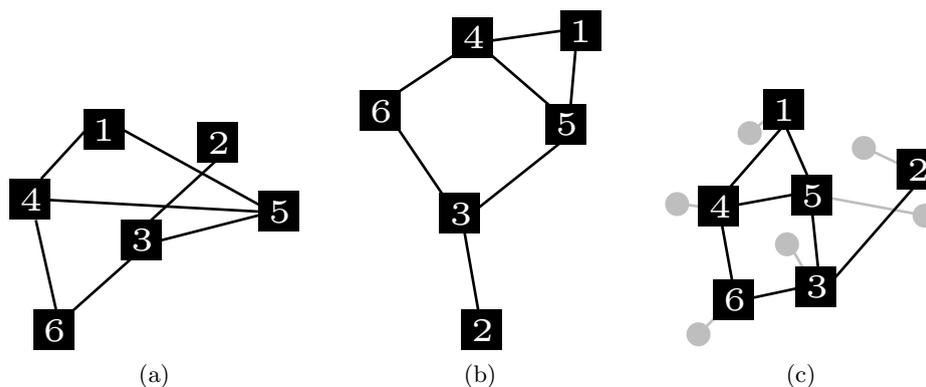


Abbildung 5.1.: Abbildung (a) zeigt den Graph G . In Abbildung (b) wurde ein kräftebasiertes Verfahren auf G angewendet. Im Gegensatz dazu wurde in Abbildung (c) für jeden Knoten aus G zusätzlich ein Ankerknoten (grau) erstellt, der die gleiche Position wie sein Original einnimmt. Anschließend wurde das gleiche Verfahren wie in (b) eingesetzt.

kann für jede Kante e aus E eine Wunschlänge ω_e definiert werden. Diese beeinflusst die Federkraft f_e , sodass e nach dem Layoutverfahren möglichst die vorgegebene Länge ω_e hat. Außerdem kann man dem Layouter fest vorgegebene Regionen zuordnen, in denen die Knoten sich am Ende befinden dürfen. Es stehen dabei aber nur einfache geometrische Figuren, wie Rechtecke, zur Verfügung. Um die Knoten in ihrer gewünschten Region zu halten reicht dies nicht aus, da hier ein komplexeres Polygon als Grenze erwünscht ist. Damit man dennoch das Ziel erreichen kann, die Knoten innerhalb ihrer Region zu positionieren, wird sowohl die Grenze nach innen, als auch nach außen als Polygon dem Graphen hinzugefügt. Zusätzlich werden diese Abgrenzungspolygone mit weiteren Knoten angereichert. Damit bewirkt man eine höhere abstoßende Kraft, die die zu bewegendenden Knoten innerhalb der Region von den umgebenden Polygonen erfährt. Damit werden die Knoten nach innen gedrückt und ein Verlassen aus der Region wird verhindert. Die Knoten, die zu den Polygonen gehören, dürfen nicht verschoben werden. Die Abgrenzungspolygone werden der Menge V' hinzugefügt, damit sie während dem Algorithmus nicht bewegt werden.

Eine weitere Besonderheit ist die getrennte Betrachtung von verschiedenen Zusammenhangskomponenten. Diese werden während des Algorithmus voneinander räumlich getrennt und auch separat betrachtet. Da das zugrundeliegende soziale Netzwerk nicht notwendigerweise einer Zusammenhangskomponente entspricht, aber eine Trennung nicht erwünscht ist, wird jeder Knoten und die Abgrenzungspolygone mit dem Projektzentrum verbunden. Die Wunschlänge ω dieser Kanten wird auf 80% ihrer ursprünglichen Länge gesetzt. Somit wird genügend Flexibilität geboten ohne aber dem Layouter zu viel Spielraum zu lassen. Gleichzeitig sorgt man aber auch dafür, dass die Knoten in der Nähe des Projektzentrums bleiben. Aus dem gleichen Grund wird die Wunschlänge der normalen Kanten, die der Graph enthält ebenfalls auf 80% ihrer originalen Länge gesetzt. Für die Kanten kann man noch eine zusätzliche Option aktivieren, damit der Algorithmus auftretende Kreuzungen versucht zu verhindern.

Wie in Abschnitt 5.2 bereits beschrieben, wird nun für jeden Knoten u aus V ein Ankerknoten o_u erstellt. Diese werden ebenfalls der Menge V' hinzugefügt und können deshalb nicht vom Algorithmus verschoben werden. Durch ihre Verbindungskante $e = (u, o_u)$ sorgen sie dafür, dass der originale Knoten nicht zu weit von seiner initialen Position entfernt wird. Die gewünschte Kantenlänge von e wird dabei auf einen Wert kleiner als die Knotengröße

gesetzt. Damit erhält man sozusagen eine starke Feder, die den zu verschiebenden Knoten in der Nähe seiner Ursprungsposition hält.

Der SmartOrganicLayouter kennt drei unterschiedliche Kräfte. Zu den bereits bekannten Feder- und Abstoßungskräften kommt noch eine Gravitationskraft hinzu. Dieses liegt im Schwerpunkt aller Knoten aus V und zieht alle Knoten zu sich heran. Das Zusammenspiel der Kräfte lässt sich nur über einen Wert steuern, die sogenannte *compactness*. Die *compactness* kann mit einem Wert aus dem Intervall $(0..1]$ belegt werden. Dabei sorgt ein höherer Wert für eine stärkere Gravitationskraft. Zusätzlich werden die Federkräfte gegenüber den Abstoßungskräften erhöht. Da nur eine minimale Verschiebung erwünscht ist, sollten die Abstoßungskräfte eher niedrig sein, aber hoch genug um Überlappungen aufzulösen. Folglich sollte der *compactness*-Wert eher höher sein $(0,6 - 0,8)$. Hier wurde bei der Implementierung etwas experimentiert und ein Wert genommen, der ein optisch ansprechendes Ergebnis liefert. Die Unterschiede fallen aber ohnehin gering aus, da durch die Ankerknoten dem Algorithmus schon einige Einschränkungen vorgegeben sind. Nachdem der Layouter gestartet wurde und sein Ergebnis zurückgeliefert hat werden alle eingefügten Ankerknoten und -kanten wieder gelöscht. Ebenfalls gelöscht werden die mit zusätzlichen Knoten angereicherten Begrenzungspolygone.

Die drei Teilgraphen in den Regionen R_0 , R_1 und R_2 werden jeweils separat gezeichnet. In jedem Schritt werden zunächst einmal das innere und äußere Begrenzungspolygon dem Graphen hinzugefügt, welche die Grenzen der entsprechenden Region darstellen. Im Falle des Layouts für die Knoten $V = A_0$ wäre dies das zugehörige Bundesland für das äußere Polygon. Ein inneres Begrenzungspolygon ist hier überflüssig und wird weggelassen. Wenn man das obige Verfahren auf alle drei Regionen angewendet hat, besitzen alle Knoten neue Positionen, die aber immer noch keine überschneidungsfreie Zeichnung garantieren.

5.4. Auftretende Probleme mit dem SmartOrganicLayouter

Damit man mit dem Algorithmus das geforderte Ziel, Überlappungen von Knoten aufzulösen, auch tatsächlich erreicht, ist es erforderlich Knotenüberlappungen explizit zu verbieten. Um während des Verfahrens auftretende Überlappungen aufzulösen ist dem SmartOrganicLayouter ein weiterer Algorithmus nachgeschaltet, der diese Überlappungen auflöst. Der Nachbearbeitungsschritt kann jedoch nicht weiter konfiguriert werden. Für die Anwendung ist dies aber problematisch, da es dabei vorkommen kann, dass Knoten aus der vordefinierten Region herausgeschoben werden. Der Grund dafür liegt darin, dass hier kein kräftebasiertes Verfahren mehr angewendet wird. So können es einzelne Knoten schaffen das Begrenzungspolygon zu überwinden, weil dort gerade genügend Platz vorhanden ist, um auftretende Überschneidungen aufzulösen. Leider gibt es beim SmartOrganicLayouter keine Möglichkeit um dies zu verhindern. Deshalb wird versucht mit einem eigenen einfachen Verfahren (s. Abschnitt 5.4.1) die Fehler nach der Berechnung des SmartOrganicLayouter zu korrigieren. Negativ zu bewerten ist außerdem die Positionierung der Knoten, die durch den SmartOrganicLayouter gewählt wird. Es ist nicht unbedingt klar, wie stark die geographische Relation zwischen den Knoten dabei vertauscht wird. Ein speziell auf diese Anwendung angepasstes kräftebasiertes Verfahren dürfte wohl bessere Ergebnisse liefern, übersteigt aber den Rahmen dieser Bachelorarbeit.

5.4.1. Zurückschieben von Knoten in ihre vordefinierte Region

Zunächst werden sämtliche Knoten identifiziert, die durch obiges Verfahren ihre vordefinierte Region verlassen haben. Vor der Anwendung des SmartOrganicLayouter merkt man sich zusätzlich noch die Position σ_u , die der Knoten vorher inne hatte. Ziel ist es nun alle Knoten möglichst nah wieder an ihre ursprüngliche Position σ zu verschieben, ohne

dass sie andere Knoten überdecken, die sich nun zwischen dem Knoten und σ befinden. (Abbildung 5.2a).

Im Folgenden ist der aktuell zu verschiebende Knoten mit u_0 bezeichnet. Als erstes wird ein Schlauch um diesen Knoten und die Zielposition σ_{u_0} gebildet. Die Breite des Schlauches wird doppelt so hoch wie der Knotendurchmesser gewählt. Dadurch können aufkommende Hindernisse als Punkte behandelt werden. Die Länge des Schlauchs entspricht dem Abstand zwischen u_0 und σ_{u_0} , zuzüglich des halben Knotendurchmessers. Sonst würde man Hindernisse hinter σ_{u_0} außer Acht lassen. Nun wird die Menge \mathbb{H} aller Knoten gebildet, deren Mittelpunkte innerhalb des Schlauchs und der Region liegen. Trifft dies auf keinen Knoten zu, kann man direkt die Position σ_{u_0} für u_0 wählen und der Knoten befindet sich wieder innerhalb seiner Region. Wenn die Menge allerdings nicht leer ist, wird diese nach dem Abstand zu u_0 sortiert. Das Hindernis $H \in \mathbb{H}$ mit der kürzesten Entfernung wird ausgewählt. Die neue Position kann man mit Hilfe des Satzes von Pythagoras berechnen (vgl. Abbildung 5.2d).

Diese einfache Korrektur der Position hat auch ihre Schwächen. Es ist beispielsweise nicht sichergestellt, dass sich innerhalb des Schlauchs genügend freier Raum für den Knoten u_0 befindet. Sollten dort zu viele Hindernisse sein, kann der Knoten nicht verschoben werden. Allerdings treten die falsch positionierten Knoten eher selten auf. Die Zeichnung wird auch groß genug gewählt, dass genügend Platz für alle Knoten innerhalb einer Region ist (s. Abschnitt 3.2). So kann man davon ausgehen, dass komplizierte Fälle gar nicht erst auftreten und ein ausgereifteres Verfahren nicht notwendig ist.

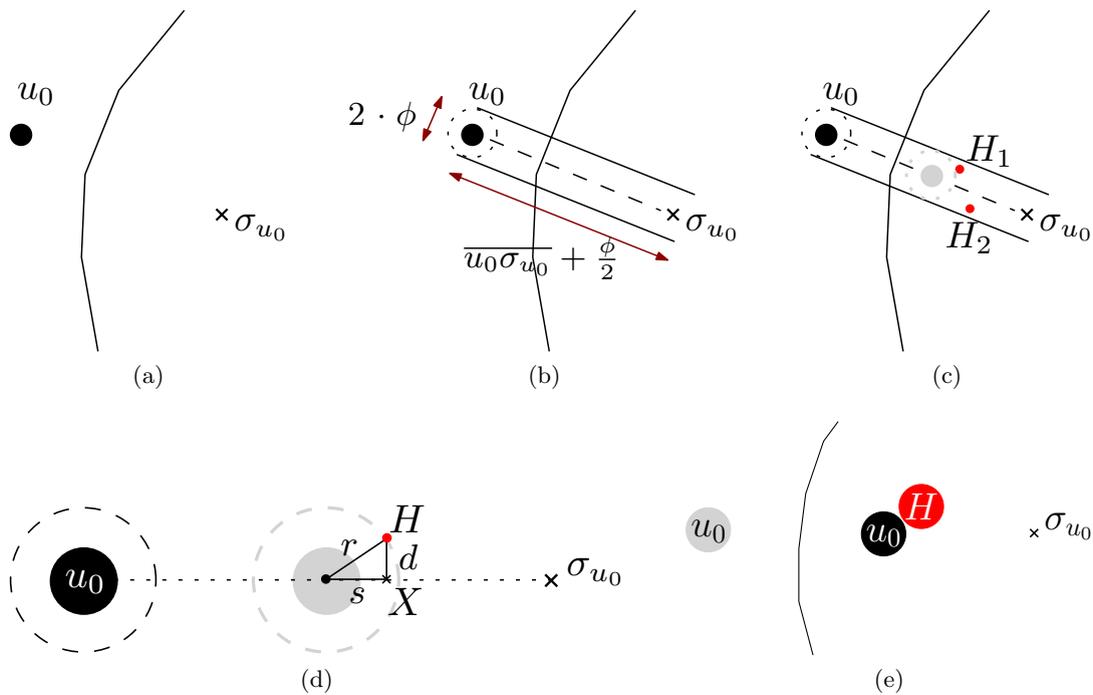


Abbildung 5.2.: Abbildung (a) zeigt den Knoten u_0 , welcher nach dem SmartOrganicLayouter außerhalb seiner zugewiesenen Region liegt. Ursprünglich hatte er die Position von σ_{u_0} . Nun wird versucht, seine Position zu korrigieren, indem eine Position zwischen u_0 und σ_{u_0} gesucht wird, so dass kein anderer Knoten überdeckt wird. In Abbildung (b) wird die Region gezeigt, in der sich eventuell Hindernisse befinden können. Alle Hindernisse werden nach Entfernung zu u_0 sortiert. In Abbildung (d) wird eine möglichst nahe Position zu σ_{u_0} bestimmt, ohne dass das nächste Hindernis H überdeckt wird. Dafür ist es notwendig X zu bestimmen. X ist der Schnittpunkt zwischen der Strecke $\overline{u_0 \sigma_{u_0}}$ und ihrer orthogonalen Strecke durch H . Mit Hilfe von $s = \sqrt{r^2 - d^2}$ lässt sich dann die gesuchte neue Position bestimmen. Die Strecke r hat die Länge des Knotenradius ϕ damit die Hindernisse nicht als Knoten, sondern als Punkte betrachtet werden können. Abbildung (e) zeigt das Ergebnis, bei dem u_0 nun wieder eine Position innerhalb der Region hat.

6. Beschriftung der Knoten

Die Aufgabe, automatisch eine Position für die Beschriftung jedes Knotens zu finden, ohne dass dadurch eine andere Beschriftung oder ein Knoten überdeckt wird, wurde unter anderem in [MS91] untersucht. Die Autoren haben dort eine vereinfachte Variante unter der Annahme von einer festen Beschriftungsgröße und vier möglichen Positionen für die Beschriftung als Problem definiert. Damit konnten sie zeigen, dass das Problem NP-vollständig ist. Gute Lösungen sind daher schwer zu finden. In [AHS05] werden verschiedene heuristische Methoden zur Erzeugung guter Beschriftungen beschrieben. Unter anderem wird ein kräftebasierter Ansatz erwähnt, der auch hier Verwendung findet. Bei der Positionierung der Beschriftungen wird eine Strategie verfolgt, bei der diese möglichst um den Graphen herum platziert werden. Damit wird versucht den Platz im Inneren überwiegend für die Darstellung der Knoten zu verwenden.

Die Platzierung der Beschriftungen gliedert sich in zwei Schritte. Zu Beginn wird mit Hilfe einer einfachen Heuristik für jede Beschriftung eine initiale Position berechnet. Dabei wird zunächst keine Rücksicht auf Überlappungen mit anderen Knoten genommen. Anschließend wird erneut ein kräftebasiertes Verfahren benutzt, um die endgültige Lage der Beschriftungsknoten festzulegen. Das Verfahren soll dafür sorgen, dass sich die Beschriftungen dort anordnen, wo noch genügend Platz ist.

Bei der Berechnung der initialen Positionen aus dem ersten Schritt werden alle Beschriftungen in zwei Kategorien unterteilt. Zum einen gibt es die Gruppe der Beschriftungen B_0 , deren Knoten sich in der Region R_0 ansammeln. Dies sind also alle Beschriftungen für die Knoten aus A_0 . In der zweiten Gruppe B_1 sind alle übrigen Beschriftungen, die zu Knoten aus dem äußeren Bereich des Graphen gehören ($A_1 \cup A_2$).

Alle Beschriftungen aus B_1 werden initial abhängig von ihrer Position in gegensätzlicher Richtung zu der Position der Bounding Box des Bundeslandes positioniert (vgl. Abbildung 6.1a). Wenn sich die Bounding Box rechts vom Knoten befindet, wird die Beschriftung links vom Knoten angeordnet. Entsprechend wird mit den anderen Richtungen verfahren. Auf diese Weise kann man die Beschriftungen aus B_1 in acht Regionen unterteilen. Durch die Wahl dieser initialen Position wird sichergestellt, dass die Beschriftungen der äußeren Knoten während des kräftebasierten Verfahrens hauptsächlich nach außen gedrückt werden. Somit wird innen mehr Platz für die dortigen Beschriftungen und Knoten geschaffen und das Risiko von Überlappungen zwischen Beschriftungen und Knoten wird verringert. Die Wahl der Bounding Box als Indikator macht zwar eine einfache Abfrage möglich, lässt aber auch einen Sonderfall zu (vgl. Abbildung 6.1b). Es kann der Fall eintreten, dass sich

ein Knoten aus $A_1 \cup A_2$ innerhalb der Bounding Box befindet. Eine Aussage über die Lage des Knoten im Vergleich zur Bounding Box ist dann nicht mehr möglich. Tritt diese Problematik auf, wird die Position in Abhängigkeit zum Mittelpunkt der Bounding Box bestimmt. Dadurch sind aber nur noch vier verschiedene Regionen möglich.

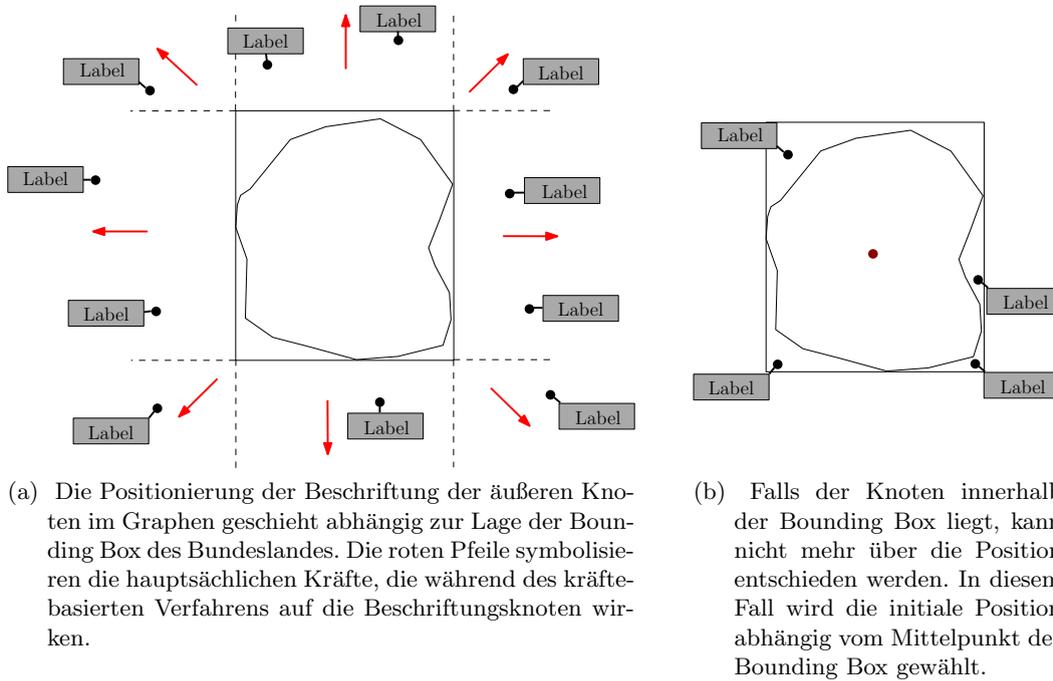


Abbildung 6.1.: Bestimmung der initialen Position der Beschriftung von äußeren Knoten.

Um die Beschriftungen B_0 zu positionieren könnte man, ähnlich wie oben, die Beschriftungen abhängig von der Position des Knotens zum Mittelpunkt der Bounding Box platzieren. Dies hat aber den Nachteil, dass je nach Verteilung der Knoten, die zugehörigen Beschriftungen sehr einseitig angeordnet werden (vgl. Abbildung 6.2a). Damit wird der gerade im Inneren geringe Platz schlecht ausgenutzt. Zwar könnte man statt des Mittelpunkts den Schwerpunkt aller Knoten nehmen, allerdings wird das Ergebnis dadurch nicht wesentlich besser. Gesucht ist also eine Möglichkeit die Beschriftungen gleichmäßiger um das Bundesland zu verteilen. Um dies zu erreichen wird zunächst der Schwerpunkt S aller zugehörigen Knoten berechnet. Von diesem Schwerpunkt aus wird ein Strahl durch jeden Knoten konstruiert, der anschließend den Umkreis der Bounding Box des Bundeslandes schneidet (Abbildung 6.2b). Damit wird jedem Knoten ein Punkt auf dem Kreis zugeordnet. Abhängig von diesem Punkt werden die Knoten sortiert. Nun beginnt man von einem beliebigen Knoten aus und ordnet der zugehörigen Beschriftung l_0 den berechneten Schnittpunkt zu. Für die weiteren Beschriftungen l_i nimmt man folgende Position:

$$\gamma = \frac{360^\circ}{|B_0|} \quad (6.1)$$

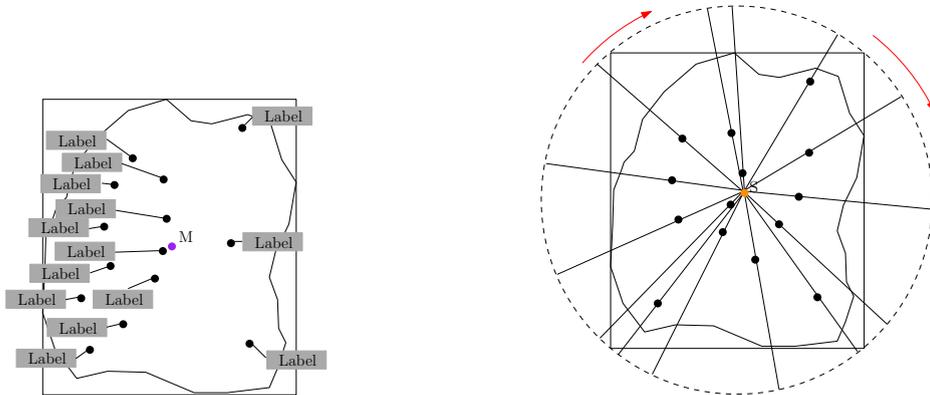
$$l_i \leftarrow l_0 + i * \gamma \quad (6.2)$$

Somit wird eine gleichmäßige Verteilung gewährleistet (Abbildung 6.2c) und es existiert für jeden Beschriftungsknoten eine Ausgangsposition.

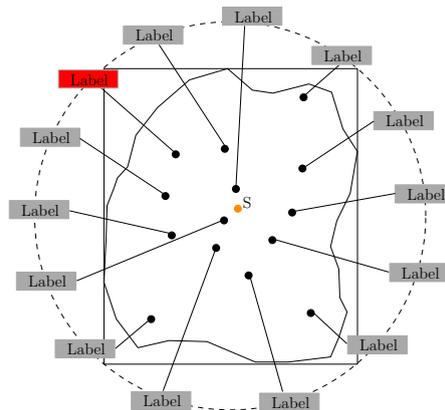
Um nun eine konfliktfreie Positionierung zu ermöglichen (keine Überdeckungen) werden am Ende alle Beschriftungen durch ein kräftebasiertes Verfahren verschoben. Hier kommt wie in Abschnitt 5.3 erneut der SmartOrganicLayouter zum Einsatz. Wieder werden Ankerknoten benutzt um die Knoten möglichst nah ihrer oben beschriebenen Initialposition

zu belassen. Der Layouter soll lediglich dort die Beschriftungen verschieben, wo es eine Überdeckung mit anderen Knoten gibt. Die eigentlichen Knoten des Graphs werden der Menge V' hinzugefügt und können nicht verschoben werden.

Aus praktischen Gründen wird jedes Label mit dessen Knoten noch durch eine Kante verbunden. Zum einen ist die Kante essentiell um beim Lesen des Layouts überhaupt erst feststellen zu können, welche Beschriftung zu welchem Knoten gehört. Zum anderen wirkt die Kante während des kräftebasierten Verfahrens wie eine Feder und sorgt dafür, dass das Label in der Nähe des Knoten bleibt. Der Layouter wird auch noch so konfiguriert, dass im Ergebnis möglichst keine Überschneidungen zwischen Knoten und Kanten auftreten.



- (a) Auswirkung wenn man für innere Beschriftungen ein ähnliches Verfahren wie bei den äußeren Labeln anwendet. Der Nachteil dabei ist die unausgeglichene Verteilung der Label. Während links kaum Platz für die Label sind, bietet die rechte Seite größere ungenutzte Flächen.
- (b) Aus allen Knoten innerhalb des Bundeslandes wird der Schwerpunkt S berechnet. Die Schnittpunkte der Strahlen liefern eine grobe Orientierung für die Platzierung der Beschriftungen auf dem Umkreis. Hierfür werden die Knoten u abhängig des Schnittpunktes von dem Strahl durch u und S und dem Umkreis im Uhrzeigersinn sortiert.



- (c) Nach der Sortierung aus (b) wird ein beliebiger Knoten ausgewählt und dessen Beschriftung l_0 (rot) auf den berechneten Schnittpunkt mit dem Umkreis gesetzt. Die folgenden Label werden auf dem Kreis um γ Grad (Gl. 6.1) verschoben und man erhält eine gleichmäßige Verteilung der Label um das Bundesland.

Abbildung 6.2.: Bestimmung der initialen Position der Beschriftung von inneren Knoten.

7. Analysemöglichkeiten

7.1. Sektoren und Georeferenz

Eine erste einfache Analyse des Projekts bezieht sich auf die unterschiedlichen Sektoren und die Lage der Akteure. In den Daten existiert für jeden Akteur eine Haltung gegenüber dem Projekt. Es gibt sowohl negative als auch positive Wertungen mit verschiedenen Stufen. Die Haltung aller Akteure innerhalb eines Sektors werden zusammengefasst und ein Durchschnittswert für den Sektor von -100% bis +100% gebildet. So lassen sich eventuell problematische Bereiche schnell identifizieren. Mehrere Beispiele sind in Abbildung 7.1 gezeigt.



Abbildung 7.1.: Sektoranalyse verschiedener Projekte.

Was die Sektoranalyse betrifft steht hier das Projekt aus Szenario 1 am besten da (Abbildung 7.1a, Abbildung A.1). In der Bevölkerung gibt es aber hier noch Verbesserungspotential. Wirft man einen genaueren Blick auf die einzelnen Akteure, erkennt man in diesem Bereich vor allem einige landesweite Zeitungen, die negativ über das Infrastrukturprojekt berichten. Es ist aber davon auszugehen, dass dies wenig negatives Licht auf das Bauprojekt wirft, da vor allem lokale Entscheidungsträger wichtig sind und diese durchaus überzeugt von dem Vorhaben sind. Als Betreiber kann man hier aber vielleicht reagieren und mehr Öffentlichkeitsarbeit leisten, um mehr Rückenwind aus der Bevölkerung zu

bekommen.

Im Gegensatz dazu scheint die Lage in Projekt 2 insgesamt negativer zu stehen (Abbildung 7.1b, Abbildung A.2). Dort wirken sich Betriebsstörungen negativ auf den Gesamteindruck aus, was man in allen Bereichen spürt. Die Unterstützung aus der Wirtschaft ist verglichen mit Projekt 1 ebenfalls nicht so stark. Insgesamt könnte man mehr Lobbyarbeit leisten oder durch Bürgschaften mehr Vertrauen erwerben.

Im dritten Projekt erkennt man an Hand der Lage der Knoten weitere Unterschiede zu den vorher genannten Projekten (Abbildung 7.1c, Abbildung A.3). Dort gibt es wenig lokal beteiligte Akteure, dafür viele überregionale oder nationale Akteure aus dem Bereich der Politik und Verwaltung. Dies könnte ein Anzeichen für die falsche Wahl des Standortes sein, da das Projekt von eigentlich wichtigen Stellen gefördert wird. Es ist also ebenfalls nützlich zu erfahren, wie lokal ein Projekt verankert ist, was man an Hand der Dichte und Verteilung der Knoten gut überblicken kann.

Wichtig für die Sektoranalyse sind korrekte Daten. Im Prinzip werden diese nur gesammelt und entsprechend visualisiert. Eine Analyse, die die Struktur des Graphen zu Grunde legt, wird im nächsten Abschnitt gezeigt.

7.2. Clustering

Zur Analyse der Struktur im sozialen Netzwerk wird der Graph als Instanz des *Correlation-Clustering* Problem betrachtet [BBC02]. Dabei wird jede Kante $e = (u, v)$ im Graph mit $\langle + \rangle$ oder $\langle - \rangle$ markiert. Im Allgemeinen symbolisiert eine mit $\langle + \rangle$ markierte Kante, eine Zusammengehörigkeit, gleich welcher Art, zwischen den Knoten u und v . Gegensätzlich verhält es sich bei den $\langle - \rangle$ Kanten. Ziel des Problems ist es nun eine Aufteilung aller Knoten in Gruppen zu finden, so dass der Kantenmarkierung möglichst gut entsprochen wird. So sollen alle Knoten u und v , die mit einer $\langle + \rangle$ Kante verbunden sind, am Ende dem gleichen Cluster zugeordnet werden. Ein Cluster bezeichnet eine Gruppe von Objekten mit ähnlichen Eigenschaften. Dahingegen sollen die Knoten w und z bei einer $\langle - \rangle$ Markierung in unterschiedlichen Gruppen sein. Dabei entsteht ein Optimierungsproblem, wobei man die Anzahl falsch zugeordneter Beziehungen versucht zu minimieren. Darüber hinaus kann auch für jede Kante e ein Gewicht c_e definiert werden, welches ebenfalls bei der Optimierung einfließt. Das Besondere von *Correlation-Clustering* im Vergleich zu vielen anderen partitionierenden Verfahren ist die Unabhängigkeit von einem externen Parameter κ , welcher vorher festlegt, wie viele Cluster existieren sollen. So werden durch das Verfahren automatisch genau so viele Cluster gebildet, wie für eine optimale Lösung notwendig sind.

Die Autoren von [BBC02] haben dort bewiesen, dass das Problem NP-hart ist. In [DEFI06] wird dieses Problem als lineares Programm formuliert und zusätzlich ein Algorithmus angegeben, der durch Runden eine $\mathcal{O}(\log n)$ Approximation erreicht. Bei den betrachteten Daten ist man durch die relativ dünnen Graphen nicht auf eine Rundung angewiesen und kann in akzeptabler Zeit eine genaue Lösung berechnen.

Im Folgenden bezeichnet $E^{\langle - \rangle}$ die Menge aller $\langle - \rangle$ Kanten. Analog ist die Menge $E^{\langle + \rangle}$ definiert. Für jedes Knotenpaar existiert eine binäre Variable x_{uv} , die genau dann 0 ist, wenn u und v sich im selben Cluster befinden. Im umgekehrten Fall ist die Variable 1. Damit lässt sich das Problem wie folgt als ILP formulieren:

Minimiere:

$$\sum_{e \in E^{\langle - \rangle}} c_e \cdot (1 - x_e) + \sum_{e \in E^{\langle + \rangle}} c_e \cdot x_e \quad (7.1)$$

unter der Bedingung:

$$x_{uv} \in \{0, 1\} \quad \forall uv \in V^2 \quad (7.2)$$

$$x_{uv} + x_{vw} \geq x_{uw} \quad \forall u, v, w \in V \quad (7.3)$$

$$x_{uv} = x_{vu} \quad \forall uv \in V^2 \quad (7.4)$$

Gleichung 7.1 setzt genau die Forderung um, die falsch zugewiesenen Knoten zu minimieren. Für $\langle - \rangle$ Kanten ist die Summe minimal, wenn $x_{uv} = 1$ gilt, also genau dann, wenn u und v in verschiedenen Clustern sind. Dagegen wird die zweite Summe minimal, wenn $x_{uv} = 0$ gilt. Weiterhin muss für alle x_e die Transitivität gelten (Gl. 7.3) um widersprüchliche Zuordnungen so vermeiden.

Bezüglich der Anwendung werden alle Kanten mit $\langle + \rangle$ markiert, deren Akteure sich positiv gesinnt sind. Umgekehrt geschieht dies bei negativen Kanten.

7.2.1. Analyse der Projektstruktur

Um die Projekte mit der Clusteranalyse strukturell voneinander zu unterscheiden, sind in den vorhandenen Beispielen leider zu wenige Daten vorhanden. So lässt sich keine Aussage darüber treffen, ob sich negativ verlaufene Projekte, von erfolgreichen Projekten im Bezug auf ihre Clusterstruktur unterscheiden.

Es lassen sich aber einige Merkmale feststellen, die ein erfolgreiches Projekt vermutlich hat. Im Abbildung 7.2 wird ein Beispiel gezeigt, wie eine Instanz eines *Correlation-Clustering*-Problems visualisiert wird. Die Akteure, die dem Projekt gegenüber positiv gesinnt sind, erkennt man an der grünen Umrandung der Knoten. Wenn es wenige Cluster gibt, in denen sich die positiv eingestellten Akteure sammeln, kann dies ein Zeichen von guter Zusammenarbeit sein. Umgekehrt kann eine Verteilung in viele verschiedene Cluster über eine Uneinigkeit Erkenntnisse liefern. Insbesondere die Akteure aus dem Bereich der Politik sind für ein erfolgreiches Projekt von entscheidender Bedeutung. Wenn also Knoten aus der Politik zusammen mit den Betreibern in dem gleichen Cluster landen, kann dies ebenfalls ein sehr gutes Zeichen für das Gelingen des Projekts sein. Optimal aus Sicht des Projekts wäre es wohl, wenn alle positiv gestimmten Akteure in ein gemeinsames Cluster zugeordnet, während ihre Gegner in vielen verschiedenen Clustern aufgeteilt sind. Damit scheinen die Gegenparteien untereinander sehr zerstritten und ein gemeinsames Vorgehen um das Projekt zu verhindern scheint schwierig.

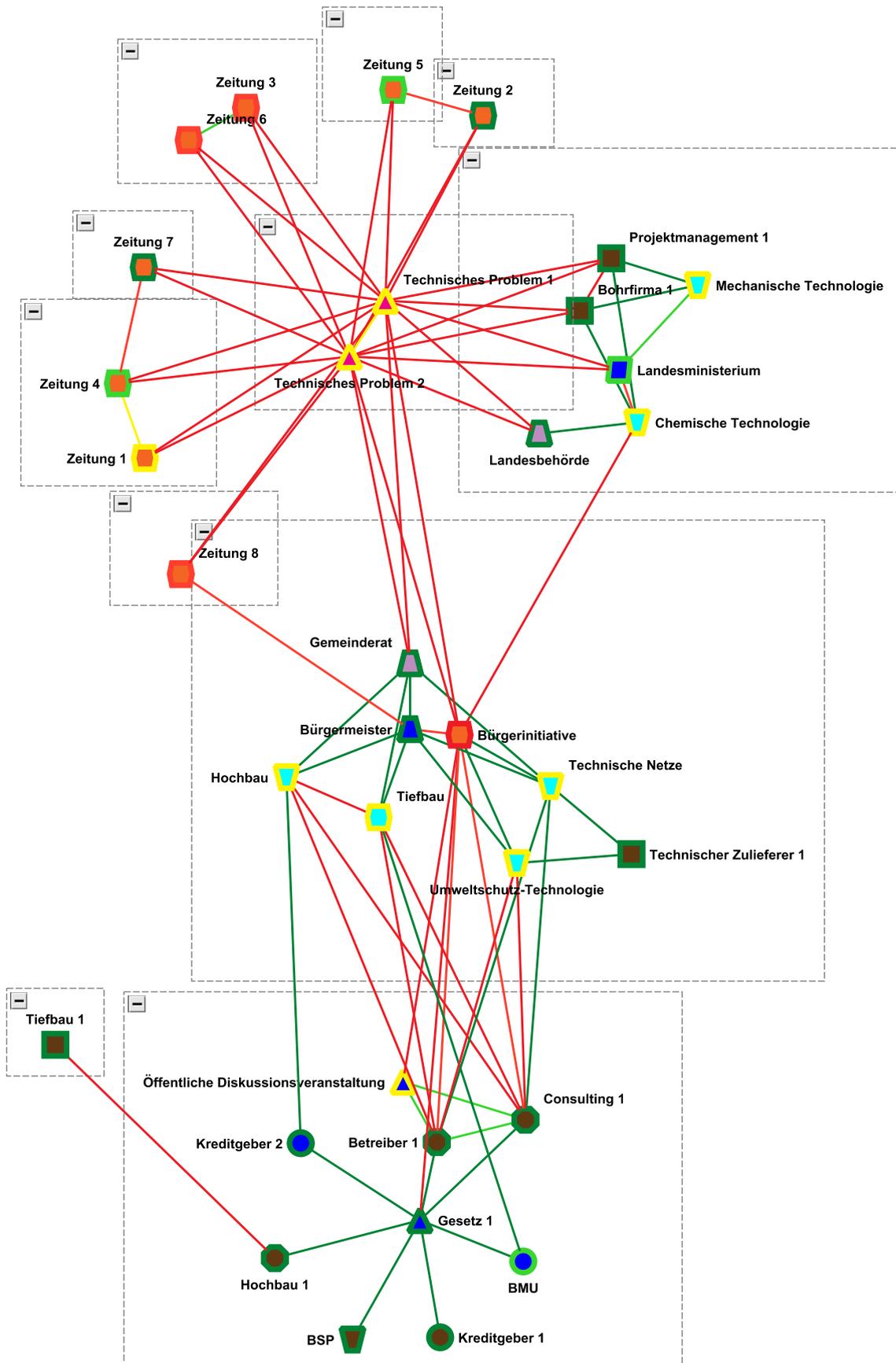


Abbildung 7.2.: Beispiel Clustering. Die Farbe der Knoten gibt erneut die Zugehörigkeit zu den Sektoren an, während die farbliche Umrandung die Haltung des Akteurs gegenüber dem Projekt veranschaulicht.

8. Zusammenfassung und Ausblick

Zunächst wurde ein Überblick über soziale Netzwerke gegeben und eine Möglichkeit beschrieben, wie man z.B. ein Infrastrukturprojekt als Netzwerk zeichnen kann. Einige Beispiele, die mit den beschriebenen Verfahren entstehen, sind im Anhang zu finden.

Das Erstellen des Layouts gliedert sich in verschiedene Schritte. Nachdem jedem Knoten seine geographische Koordinate zugeordnet wurde (Abbildung 8.1a) ist es nötig die Region um den Projektstandort auszudehnen, um alle Akteure sichtbar zu machen. Gleichzeitig werden in dieser Phase alle Akteure so verschoben, dass sich der Standort am Ende in der Mitte der Zeichnung befindet. Um die Änderung der geographischen Position sichtbar zu machen, wird auch der Umriss des Bundeslandes verfälscht (Abbildung 8.1b).

Als nächstes werden die weiteren Regionen eingezeichnet und die Akteure aus A_1 und A_2 platziert (Abbildung 8.1c). Die Position orientiert sich dabei ebenfalls an der geographischen Koordinate, ist aber auch vom Zufall abhängig.

Nachdem nun alle Knoten eine initiale Position besitzen, müssen die noch vorhandenen Überlappungen aufgelöst werden (Abbildung 8.1d). Dazu wurde ein kräftebasiertes Verfahren eingesetzt. Dabei kann es vorkommen, dass Knoten ihre vordefinierte Region verlassen. Diese müssen zur Korrektur wieder zurückgeschoben werden. Hierzu wurde ein sehr einfaches Verfahren benutzt, welches auch ausreicht, da dieser Fall in den betrachteten Beispielen nicht häufig auftritt. Nun können, falls gewünscht, noch Kanten eingezeichnet werden (Abbildung 8.1e). Insgesamt betrachtet ist das Layout mit Kanten aber leider zu unübersichtlich. Durch den Wunsch der Georeferenzierung kann man wenig Rücksicht auf Kreuzungen und Winkelauflösung nehmen. Wenn man darauf aber mehr Wert legen will, könnte man die Knoten in den äußeren Regionen freier platzieren und dabei mehr auf Kreuzungen der Kanten achten.

Abschließend werden noch die Beschriftungen eingezeichnet. Für eine initiale Position wurde die hierarchische Anordnung der Knoten in Regionen ausgenutzt. Die endgültige Position richtet sich dann erneut nach einem kräftebasierten Algorithmus (Abbildung 8.1f).

Der mehrfache Einsatz des kräftebasierten Verfahrens nimmt fast die komplette Berechnungszeit des Layouts in Anspruch. Während das Einlesen der Daten in etwa 15 % der Laufzeit in Anspruch nimmt, schlägt das Auflösen der Überlappungen mit etwa 80 % der Laufzeit zu Buche. Beim Platzieren der Beschriftungen werden zusätzliche 4 % benötigt, also deutlich weniger obwohl das gleiche kräftebasierte Verfahren zum Einsatz kommt (vgl. Tabelle A.1, A.2 und A.3). Der Grund dafür liegt im Auffüllen der Begrenzungs-polygon mit

zusätzlichen Knoten, um zu verhindern, dass die Knoten, die man verschieben will, nicht ihre Region verlassen. Auf diese Weise hat man allein beim Auflösen der Überlappungen aus Region R_2 über 3000 Knoten, wohingegen das Resultat aus insgesamt nur noch 100 – 200 Knoten besteht. Wenn man durch ein eigens angepasstes kräftebasiertes Verfahren das Anreichern der Begrenzungspolygone unnötig macht, kann man so eventuell einiges an Berechnungszeit einsparen. Um die Ergebnisse zu verbessern steckt im kräftebasierten Verfahren ebenfalls das meiste Potential. Wenn man dort ein Verfahren benutzt, welches speziell für georeferenzierte Graphen besser geeignet ist, kann man am Ende womöglich bessere Ergebnisse erzielen.

Als Möglichkeit die Struktur des Graphen zu analysieren wurde das Correlation-Clustering vorgestellt. Es hat die Eigenschaft, dass das Verfahren selbstständig die benötigte Anzahl an Clustern bildet. So muss man vorher nicht festlegen muss, wie viele Cluster gebildet werden sollen, was das Ergebnis flexibler macht.

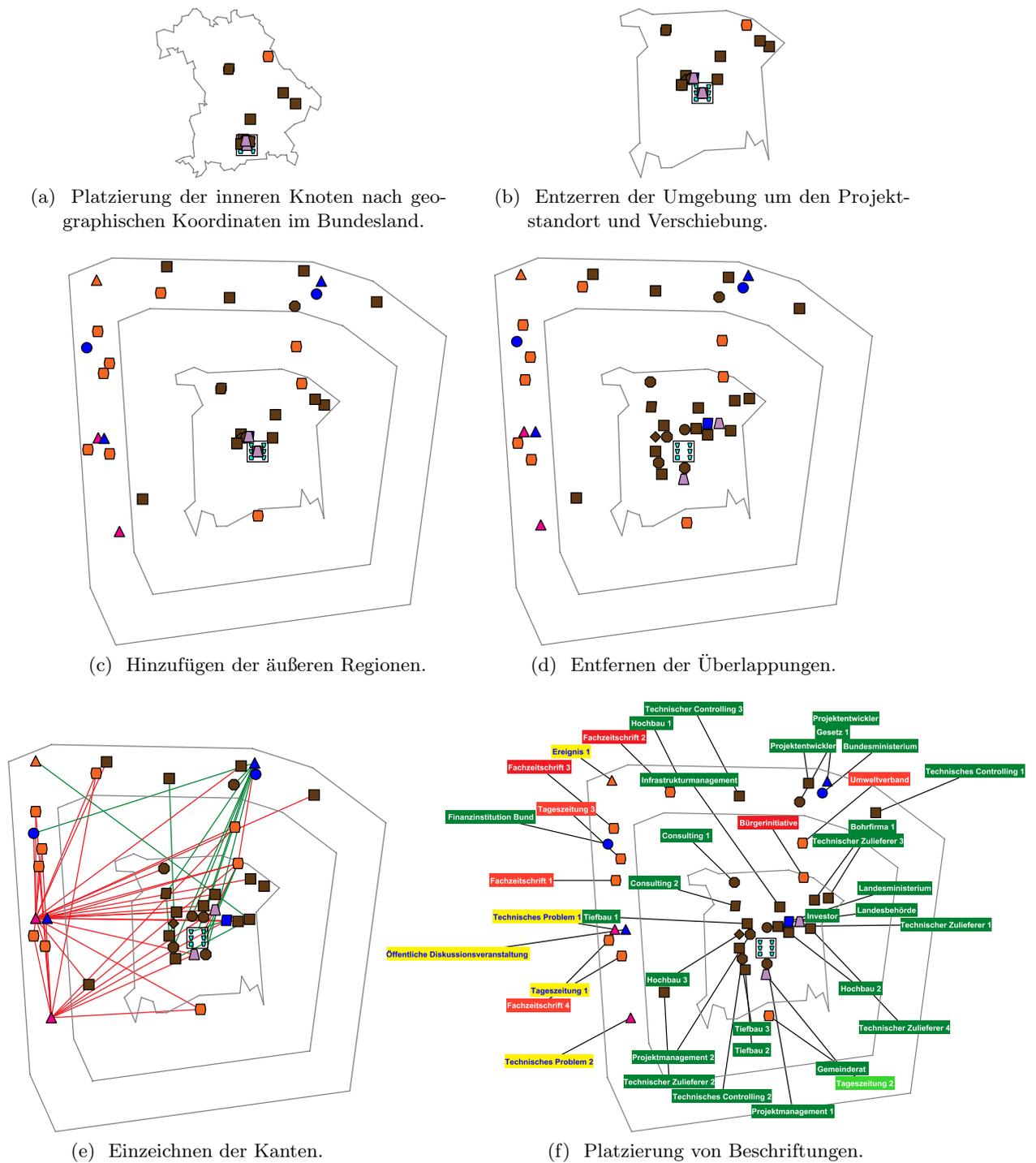


Abbildung 8.1.: Übersicht über die Visualisierungsschritte.

Anhang

A. Fallstudien

Als Ergebnis wurden aus drei Datensätzen jeweils eine Visualisierung erstellt (Abbildungen A.1–A.3). Es wurden ebenfalls Messungen durchgeführt, um Aussagen über die Laufzeit treffen zu können. Dabei wurde die Zeit für die wichtigsten Visualisierungsschritte mehrmals gemessen und anschließend der Durchschnitt gebildet (Tabellen A.1, A.2 und A.3). Zur besseren Lesbarkeit wurden die Kanten in der Zeichnung weggelassen. Wegen der vorgegebenen initialen Position der Knoten, kann man diese nicht beliebig verschieben und deshalb Kantenkreuzungen auch nicht vermeiden. Aus diesem Grund ist das Layout bei vielen Kanten eher ungeeignet.

Daten einlesen	Entzerren	Überlappungen	Beschriftungen	Rest	Gesamtzeit
1.074 ms	2 ms	5766 ms	310 ms	95 ms	7247 ms

Tabelle A.1.: Laufzeitmessung für Szenario 1 (Abbildung A.1).
Anzahl Knoten: 41. Anzahl Kanten: 57

Daten einlesen	Entzerren	Überlappungen	Beschriftungen	Rest	Gesamtzeit
1090 ms	2 ms	4762 ms	342 ms	103 ms	6299 ms

Tabelle A.2.: Laufzeitmessung für Szenario 2 (Abbildung A.2).
Anzahl Knoten: 31. Anzahl Kanten: 35

Daten einlesen	Entzerren	Überlappungen	Beschriftungen	Rest	Gesamtzeit
977 ms	2 ms	5141 ms	283 ms	85 ms	6488 ms

Tabelle A.3.: Laufzeitmessung für Szenario 3 (Abbildung A.3).
Anzahl Knoten: 29. Anzahl Kanten: 77

Bei allen drei Beispielen wurden die Überlappungen von Knoten komplett aufgelöst. Ebenfalls treten keine Überlappungen der Beschriftungen auf, allerdings ist es teilweise nicht so einfach abzulesen, welche Beschriftung zu welchem Knoten gehört. So können die Verbindungskanten zwischen Beschriftung und Knoten sich kreuzen oder von anderen Knoten überdeckt werden, wie z.B bei der Beschriftung *Projektentwickler* aus Szenario 3 (A.3). Hier müsste man also für eine bessere Lesbarkeit der Zeichnung noch nachbessern. Besonders bei Projekten mit noch mehr Akteuren als in Szenario 1 (A.1), verliert man durch eine ungünstige Platzierung der Beschriftungen schnell den Überblick. Es zeigt sich auch, dass sich manche Knoten in ihrer Form zu sehr ähneln, um die verschiedenen Funktionen der Akteure leicht ablesen zu können. Für die Analyse wurde aber vor allem Wert auf

die verschiedenen Sektoren gelegt und hier erreicht man mit der graphischen Variablen der Farbe eine bessere Unterscheidbarkeit. Über die Haltung aller Akteure gegenüber dem Projekt kann man sich dagegen relativ schnell einen Überblick verschaffen, da hier die Farbgebung sehr intuitiv ist und einem schnell ins Auge fällt. Eine weitere Eigenschaft, die in den Anforderungen formuliert wurde, ist der Platzverbrauch. Die Knoten und Beschriftungen sind in den vorliegenden Beispielen weitestgehend gleich verteilt und es gibt wenig ungenutzten freien Raum. Eine Ausnahme bilden hier jedoch die äußeren Regionen. Betrachtet man ein Projekt aus Bayern, wie in Szenario 1, mit dem Auftreten von Akteuren überwiegend aus Deutschland, dann ist es klar, dass sich die Knoten mehr westlich und nördlich des Bundeslandes ansammeln, während die Bereiche im Süden und Osten frei bleiben. Der Grund dafür ist die Georeferenzierung der Knoten, die eine homogene Verteilung der Knoten verhindert.

Die Georeferenzierung war nach der Formulierung der Anforderungen vor allem für die Akteure im Bundesland wichtig. Mit Hilfe eines Vergleichs der Abbildungen 8.1b und 8.1d lässt sich erkennen, dass die orthogonale Ordnung nicht für jedes Knotenpaar erfüllt wird. Hier ist nicht so ganz klar, wie sich das kräftebasierte Verfahren beim Auflösen der Überlappungen auf die orthogonale Ordnung auswirkt. Jedenfalls müsste ein angepasstes Verfahren implementiert werden, um eine Zeichnung zu erhalten, die der Georeferenzierung besser entspricht. Wie schon in Kapitel 8 beschrieben, würde sich das auch positiv auf die Laufzeit auswirken. Bei allen Fallstudien benötigt das Auflösen der Überlappungen die meiste Zeit (s. Tabelle A.1 – A.3).

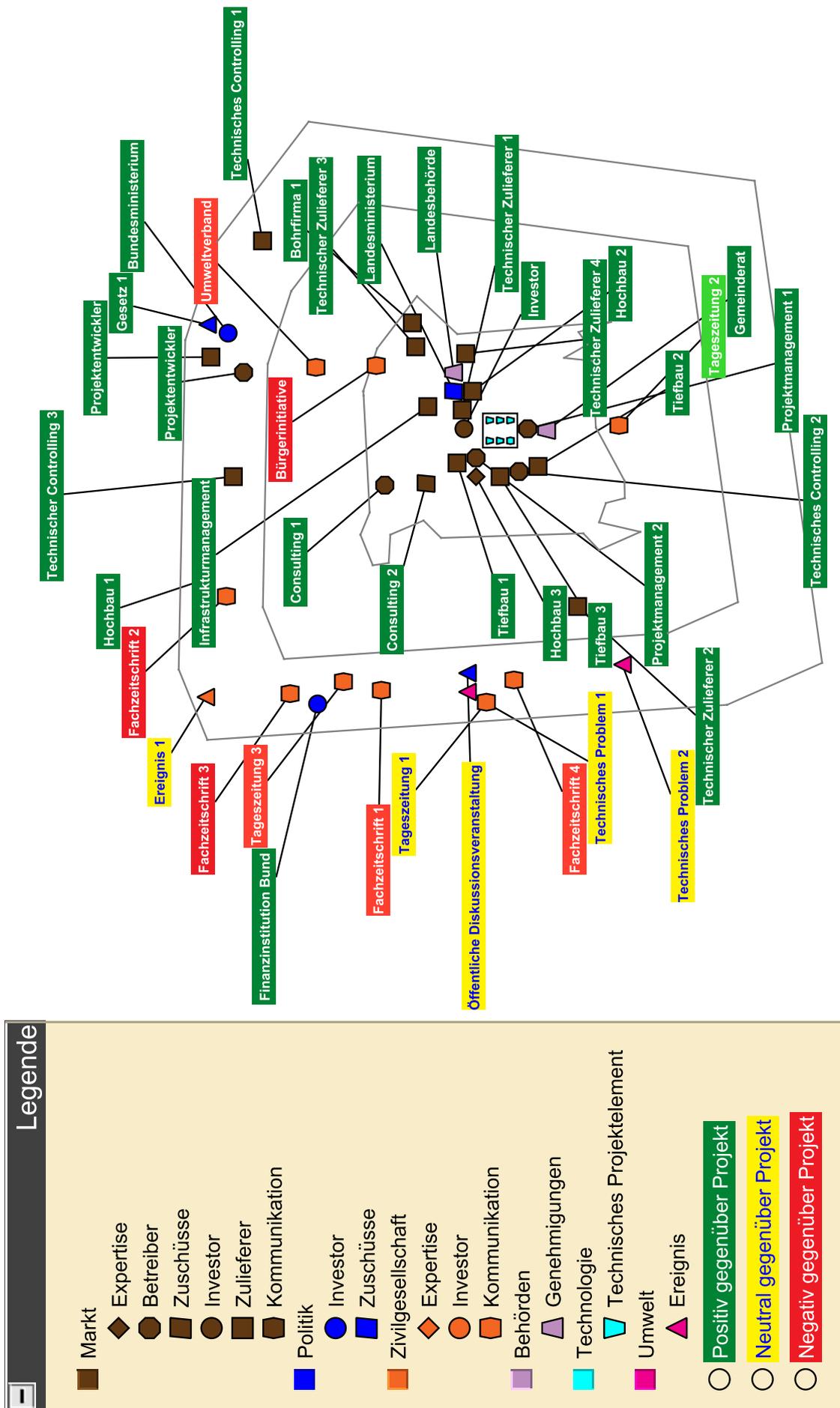


Abbildung A.1.: Szenario 1 ohne Kanten

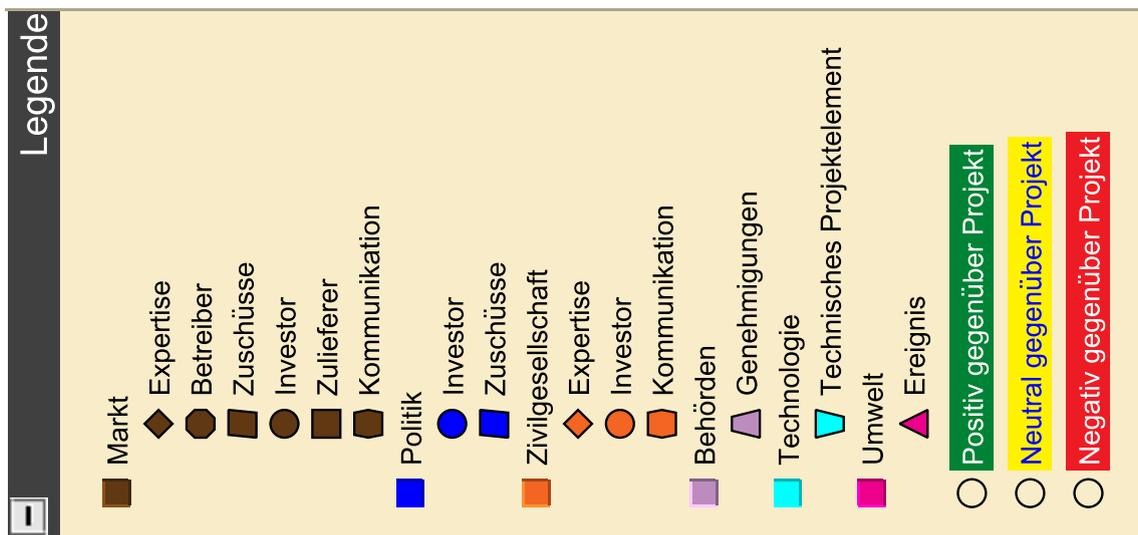


Abbildung A.3.: Szenario 3 ohne Kanten

B. Programm zum Erstellen des Layouts

Für die Visualisierung und das Erstellen des Graphen wird die *yFiles*-Bibliothek für Java der Firma yWorks genutzt. Damit lassen sich die Graphen und ihre Zeichnung im GraphML-Format erstellen. Zur Anzeige kann dann der kostenlose Editor *yEd*¹ benutzt werden.

B.1. Aufbau der Daten

Die kompletten Daten, die man visualisieren will, werden in drei Textdateien aufgeteilt. Die einzelnen Spalten werden durch das & -Zeichen getrennt.

B.1.1. Akteure

Projekt	ID Institution	Name	Region	PLZ	ID Typ	Sektor	Haltung	Funktion
---------	----------------	------	--------	-----	--------	--------	---------	----------

Tabelle B.4.: Eigenschaften der Akteure. Jede weitere Zeile erzeugt einen neuen Akteur.

Projekt Name des Projekts. Diese Information dient in erster Linie der Übersicht und wird nicht weiter verwendet.
(Typ: String)

ID Institution Eine eindeutige Nummer, über die der Akteur identifiziert werden kann.
(Typ: Integer)

Name Der Name des Akteurs. Der Inhalt wird als Knotenbeschriftung benutzt.
(Typ: String)

Region Ordnet dem Akteur seine Region zu. Es sind drei Regionen möglich: R_0 (9), R_1 (99), R_2 (999).
(Typ: Integer)

PLZ Die Postleitzahl des Akteurs. Mit dieser Information wird die initiale Position des Knoten im Graph gewählt. Sollte keine Postleitzahl vorhanden sein, kann man mit 999 eine gewünschte zufällige Positionierung kenntlich machen.
(Typ: Integer)

ID Typ Legt fest, welche Funktion der Knoten im Projekt hat. Dadurch wird die Form des Knoten beeinflusst.
(Typ: Integer)

Sektor Ordnet dem Akteur einen Sektor zu. Möglich sind: *Market*, *Politics*, *Civil Society*, *Authorities*, *Technology*, *Environment*. Die Wahl des Sektors beeinflusst die Knotenfarbe.
(Typ: String)

Haltung Gibt dem Akteur eine Haltung gegenüber dem Projekt. Abstufungen von [-4..4] sind möglich. Positive Werte zeigen, dass der Akteur für das Projekt ist. Die Haltung wird durch die Hintergrundfarbe der Knotenbeschriftung deutlich gemacht.
(Typ: Integer)

Funktion Eine Beschreibung der Funktion, die der Akteur während dem Projekt inne hat. In *yEd* taucht die Beschreibung auf, wenn man mit der Maus über einen Knoten fährt.
(Typ: String)

¹http://www.yworks.com/en/products_yed_about.html

B.1.2. Projektknoten

Der Projektknoten besteht neben den notwendigen Informationen noch aus weiteren Technologiemarken. Diese Knoten geben Aufschluss über technische Eigenschaften des Projekts. Da diese Knoten im Programm wie andere Akteure behandelt werden, sind die gleichen Eigenschaften wie in Tabelle B.4 notwendig. Im Unterschied dazu haben aber die Spalten Region und PLZ keinen Einfluss auf das Ergebnis.

B.1.3. Kanten

Source ID	Target ID	Beschreibung	Haltung	Gewicht
-----------	-----------	--------------	---------	---------

Tabelle B.5.: Eigenschaften der Kanten

Jede Zeile fügt dem Graph eine neue Kante hinzu. In Source ID bzw. Target ID trägt man die Institution ID der jeweiligen Akteure ein. Im Beschreibungsfeld kann man einen Text eintragen, der eingeblendet wird, sobald man mit der Maus über eine Kante fährt. Die Spalte für Haltung legt fest, wie die Beziehung zwischen den beiden Akteuren ist. Es sind positive und negative Werte möglich. Auch neutrale Beziehungen können durch eine 0 kenntlich gemacht werden. Durch das Feld werden auch die (+) und (-) Kanten für das *Correlation-Clustering* festgelegt. Je nach Haltung wird die Kante in rot, gelb oder grün eingefärbt. Das Gewicht muss vom Typ Double sein und wird ebenfalls beim *Correlation-Clustering* verwendet.

B.2. Programmoberfläche



Abbildung B.4.: Programmoberfläche

Verzeichnis Gibt den Ort an, an dem sich die Daten zu den Bundesländern und den Postleitzahlen befinden.

Akteure Gibt den Ort an, an dem sich die Daten zu den Akteuren befinden.

Projekt Gibt den Ort an, an dem sich die Daten zu dem Projekt befinden.

Kanten Gibt den Ort an, wo sich die Daten zu den Kanten befinden.

Umriss Es sollte das Bundesland ausgewählt werden, an dem sich der Projektstandort befindet. Das Bundesland wird dann als Umriss dargestellt. Es stehen fast alle deutschen Bundesländer zur Verfügung.

Legende Gibt an, ob dem Graph zusätzlich eine Legende hinzugefügt werden soll. An ihr kann man sehen, wie die Farben bzw. Formen den Sektoren und Funktionen zugewiesen werden.

- Regionen** Gibt an, ob die Begrenzungspolygone der Regionen R_i eingezeichnet werden sollen.
- Labels** Gibt an, ob Beschriftungen für jeden Knoten erstellt werden sollen.
- Kanten** Gibt an, ob die definierten Kanten gezeichnet werden sollen.
- Projektknoten** Gibt an, ob der zur besseren Übersicht größere Projektknoten gezeichnet werden soll (Abbildung 2.3b).
- Sektoranalyse** Schaltet die Sektoranalyse an bzw. ab (Abschnitt 7.1).
- Zoom** Hier kann definiert werden, wie viel Platz man einem Knoten im Bundesland zuordnet (Parameter τ in Gl. 3.2). Je größer der Wert, desto größer werden die Abstände zwischen den Knoten.
- Breite** Hier kann die Breite der Regionen R_1 und R_2 verändert werden, falls zu wenig Platz für die Knoten und Beschriftungen ist (Parameter t in Gl. 3.6).

B.3. Ablauf

Nachdem die benötigten Daten eingelesen sind, wird zunächst der Parameter s für die Koordinatenberechnung (Gl. 4.8) bestimmt. Danach können die Akteure innerhalb des Bundeslandes als Knoten hinzugefügt werden. Im nächsten Schritt wird die in Kapitel 4 beschriebene Entzerrung auf das Bundesland und die Akteure angewendet. Um die noch bestehenden Überlappungen aufzulösen wird das Verfahren aus Abschnitt 5.3 angewendet. Falls der Benutzer zusätzlich noch die Beschriftung der Knoten anzeigen will, wird die Position jeder Beschriftung wie in Kapitel 6 berechnet. Abschließend werden eventuell noch die Legende, Sektoranalyse und die Regionen dem Graphen zugefügt. Die Ergebnisse für drei Projekte sind im Anhang zu finden. Dort wurde wegen der schlechteren Übersicht auf das Einzeichnen von Kanten verzichtet. Der Graph wird im GraphML-Format gespeichert und kann mit dem kostenlosen Editor *yEd* geöffnet werden. Das Betrachten der Layouts eignet sich dort besser, da man die einzelnen Elemente ein- und ausblenden kann.

Literaturverzeichnis

- [AHS05] Kamran Ali, Knut Hartmann und Thomas Strothotte: *Label Layout for Interactive 3D Illustrations*. Journal of the WSCG, 13:2005, 2005.
- [BBC02] N. Bansal, A. Blum und S. Chawla: *Correlation clustering*. In: *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, Seiten 238 – 247, 2002.
- [Ber11] Jacques Bertin: *Semiology of Graphics : Diagrams, Networks, Maps*. Esri Press, Redlands, Calif., 1. ed. Auflage, 2011.
- [BETT99] Giuseppe Di Battista, Peter Eades, Roberto Tamassia und Ioannis G. Tollis: *Graph drawing : Algorithms for the visualization of graphs*. Prentice Hall, 1999.
- [BKW03] Ulrik Brandes, Patrick Kenis und Dorothea Wagner: *Communicating Centrality in Policy Network Drawings*. IEEE Transactions on Visualization and Computer Graphics, 9:241–253, April 2003.
- [BS95] Lev M. Bugayevskiy und John P. Snyder: *Map projections : A Reference Manual*. Taylor & Francis, 1995.
- [BW03] Ulrik Brandes und Dorothea Wagner: *Visone – Analysis and Visualization of Social Networks*. In: Michael Jünger und Petra Mutzel (Herausgeber): *Graph Drawing Software*, Seiten 321–340. Springer-Verlag, 2003.
- [DEFI06] Erik D. Demaine, Dotan Emanuel, Amos Fiat und Nicole Immorlica: *Correlation Clustering in General Weighted Graphs*. Theoretical Computer Science, 361(2-3):172 – 187, 2006.
- [DP73] David H. Douglas und Thomas K. Peucker: *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*. Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2):112–122, 1973.
- [FR91] Thomas M. J. Fruchterman und Edward M. Reingold: *Graph drawing by force-directed placement*. Software: Practice and Experience, 21(11):1129–1164, 1991, ISSN 1097-024X.
- [HHE07] Weidong Huang, Seok Hee Hong und Peter Eades: *Effects of Sociogram Drawing Conventions and Edge Crossings in Social Network Visualization*. J. Graph Algorithms Appl., 11(2):397–429, 2007.
- [HS92] John Hersherberger und Jack Snoeyink: *Speeding Up the Douglas-Peucker Line-Simplification Algorithm*. Technischer Bericht, Vancouver, BC, Canada, Canada, 1992.
- [KK89] Tomihisa Kamada und Satoru Kawai: *An algorithm for drawing general undirected graphs*. Information Processing Letters, 31(1):7 – 15, 1989, ISSN 0020-0190.

- [KW01] Michael Kaufmann und Dorothea Wagner: *Drawing Graphs: Methods and Models (Lecture Notes in Computer Science)*, Band 2025 der Reihe *Lecture Notes in Computer Science*. Springer, 2001.
- [Lat07] Bruno Latour: *Eine neue Soziologie für eine neue Gesellschaft. Einführung in die Akteur-Netzwerk-Theorie*. Suhrkamp, 2007.
- [Luh84] Niklas Luhmann: *Soziale Systeme: Grundriß einer allgemeinen Theorie*. Suhrkamp, 1984.
- [Lyo92] Kelly A. Lyons: *Cluster busting in anchored graph drawing*. In: *Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research - Volume 1, CASCON '92*, Seiten 7–17. IBM Press, 1992. <http://dl.acm.org/citation.cfm?id=962198.962200>.
- [MELS95] Kazuo Misue, Peter Eades, Wei Lai und Kozo Sugiyama: *Layout Adjustment and the Mental Map*. *Journal of Visual Languages and Computing*, 6(2):183 – 210, 1995, ISSN 1045-926X. <http://www.sciencedirect.com/science/article/pii/S1045926X85710105>.
- [MS91] Joe Marks und Stuart Shieber: *The Computational Complexity of Cartographic Label Placement*. Technischer Bericht TR-05-9, Center for Research in Computing Technology, Harvard University, 1991.
- [Ope11] OpenGeoDB. <http://opengeodb.giswiki.org/wiki/OpenGeoDB>, August 2011.
- [XID10] Wang Xiao-li und Zhang De: *Selecting Optimal Threshold Value of Douglas-Peucker Algorithm Based on Curve Fit*. In: *Networking and Distributed Computing (ICNDC), 2010 First International Conference on*, Seiten 251–254, oct. 2010.
- [yWo11] yWorks. http://www.yworks.com/en/products_yfiles_about.html, August 2011.