

Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3362

Visualisierung von dynamischen Clustern innerhalb von Netzwerken

Patrick Auwärter

Studiengang: Softwaretechnik
Prüfer: Prof. Dr. Daniel Weiskopf
Betreuerin: Dipl.-Ing. Corinna Vehlow

begonnen am: 20. Juni 2012
beendet am: 20. Dezember 2012

CR-Klassifikation: H.5.m, H.3.3

Kurzfassung

Viele reale Systeme nehmen die Form eines Netzwerks an. Unter einem Netzwerk versteht man eine Struktur, in welcher Elemente in Beziehung zueinander stehen. Netzwerke lassen sich formal als Graphen modellieren. Eine Eigenschaft vieler Netzwerke ist, dass sich ihre Elemente in Gruppen (Cluster) einteilen lassen. Diese Gruppen sind in sich sehr dicht, untereinander aber nur schwach verbunden. In dynamischen Netzwerken durchlaufen die Gruppen einen Lebenszyklus: Sie entstehen, entwickeln sich und verschwinden eventuell wieder. Die Analyse dieser Gruppenentwicklungen kann entscheidend zum Verständnis von realen Systemen beitragen.

Im Rahmen dieser Diplomarbeit wurde daher eine Visualisierung entwickelt, welche es ermöglicht die Evolution von Gruppen in dynamischen Netzwerken nachzuvollziehen. Im Gegensatz zu bestehenden Visualisierungen für Gruppen, kann mit dem entwickelten Ansatz auch die Struktur des Netzwerks zu jedem Zeitpunkt dargestellt werden. Einzelne Elemente können selektiert werden, um so deren Gruppenzugehörigkeit im Verlauf der Zeit sichtbar zu machen. Die Stabilität von Elementen im Bezug auf ihre Gruppenzugehörigkeit kann ebenfalls visuell abgebildet werden. Zudem können dominierende Gruppen, welche über die Zeit hinweg bestehen, detektiert und hervorgehoben werden. Zuletzt wurde die Anwendbarkeit der Visualisierung mithilfe zweier realer Netzwerk-Datensätze untersucht.

Inhaltsverzeichnis

1. Einleitung	9
1.1. Motivation	9
1.2. Ziel	10
2. Grundlagen und Definitionen	13
2.1. Graph	13
2.2. Dynamischer Graph	14
2.3. Gruppen	14
2.4. Evolution von Gruppen	15
2.5. Soziales Netzwerk	16
3. Clustering-Algorithmen	17
3.1. Clustering statischer Graphen	17
3.2. Clustering dynamischer Graphen	18
3.2.1. Lokales Clustering	18
3.2.2. Evolutionäres Clustering	18
3.3. Gruppenverfolgung	19
3.3.1. Dynamische Gruppe	19
3.3.2. Gruppenzuordnung	20
3.3.3. Verfahren	21
3.3.4. Signifikanz	21
4. Verwandte Visualisierungskonzepte	23
4.1. Visualisierung statischer Graphen	23
4.2. Visualisierung dynamischer Graphen	23
4.2.1. Animation	23
4.2.2. Statische Ansichten	24
4.3. Visualisierung der Evolution von Gruppen	27
4.3.1. Ogawa et al.	27
4.3.2. Matchmaker	29
4.3.3. Alluvial Diagrams	33
4.3.4. Reda et al.	35
4.4. Fazit	37
5. Konzept	39
5.1. Detail-Ansicht	39
5.2. Aggregierte Ansicht	40

5.3.	Weitere Bestandteile des Konzepts	41
5.3.1.	Interaktion	41
5.3.2.	Hervorhebungen	41
5.3.3.	Minimierung der Kreuzungen von Transitionen	42
6.	Umsetzung	43
6.1.	Eingabedaten	45
6.2.	Ansichten	46
6.2.1.	Cluster-Ansicht	46
6.2.2.	Aggregierte Ansicht	48
6.2.3.	Detail-Ansicht	49
6.3.	Interaktion	50
6.4.	Einfärbungen	51
6.4.1.	Dynamische Gruppen	51
6.4.2.	Knoten-Stabilität	52
6.5.	Minimierung der Kreuzungen von Transitionen	53
6.5.1.	Globale Minimierung	53
6.5.2.	Lokale Minimierung	54
7.	Fallstudien	57
7.1.	Hypertext 2009	57
7.2.	Fußball-Begegnungen	60
8.	Zusammenfassung und Ausblick	65
A.	Anhang	67
	Literaturverzeichnis	69

Abbildungsverzeichnis

1.1. Netzwerk von Wissenschaftlern [POMo9]	10
2.1. Knoten-Kanten-Diagramme [BLM ⁺ o6]	13
2.2. Dynamischer Graph	14
2.3. Graph mit Gruppenstruktur [NGo4]	15
2.4. Ereignisse bei der Evolution von Gruppen [PBVo7]	15
2.5. Soziales Netzwerk von Freundschaftsbeziehungen	16
3.1. Graph mit überlappenden Gruppen [PDFVo5]	18
3.2. Dynamische Gruppen [GDC10]	20
4.1. Animation [FTo4]	24
4.2. TimeArcTrees [GBDo9]	25
4.3. Layers-Konzept [BCo3]	26
4.4. Mailinglisten-Ansicht von [OMB ⁺ o7]	28
4.5. Matchmaker – Arbeitsablauf [LSP ⁺ 10]	30
4.6. Matchmaker – Bündelung der Verbindungen [LSP ⁺ 10]	31
4.7. Matchmaker – Minimierung der Linienkreuzungen [LSP ⁺ 10]	31
4.8. Matchmaker – Übersichts-Modus [LSP ⁺ 10]	32
4.9. Significance Clustering und Alluvial Diagram [RB10]	34
4.10. Visualisierung von Edler und Rosvall [ER10]	35
4.11. Visualisierung von Reda et al. [RTJ ⁺ 11]	36
5.1. Entwurf der Detail-Ansicht	39
5.2. Entwurf der aggregierten Ansicht	40
6.1. Übersicht der entwickelten Anwendung	43
6.2. Eingabedaten	45
6.3. Cluster-Ansicht	47
6.4. Vergleich der Darstellungen von Transitionen	47
6.5. Aggregierte Ansicht	48
6.6. Detail-Ansicht	49
6.7. Selektion	50
6.8. Kanten-Filter	51
6.9. Einfärbung der dynamischen Gruppen	52
6.10. Einfärbung der Knoten-Stabilität	53
6.11. Globale Minimierung der Transitions-Kreuzungen	54

6.12. Knotenordnung bei der lokalen Minimierung	55
6.13. Vergleich der Verfahren zur Minimierung der Transitions-Kreuzungen	56
7.1. Hypertext 2009 – Weitere Ansichten	58
7.2. Hypertext 2009 – Cluster-Ansicht mit dynamische Gruppen	59
7.3. Fussball-Datensatz – Ausschnitt	61
7.4. Fussball-Datensatz – Afrika-Gruppe	62
7.5. Fussball-Datensatz – Übersicht	63
A.1. Einfärbung der Knoten-Stabilität – Großansicht	68

1. Einleitung

1.1. Motivation

Netzwerke sind in unserer Welt allgegenwärtig und wir selbst sind Teil von ihnen. Unter einem Netzwerk versteht man eine Struktur, in welcher Elemente in Beziehung (Relation) zueinander stehen. Die Elemente werden dabei als *Knoten* und die Verbindungen zwischen den Elementen als *Kanten* bezeichnet. In der Mathematik sind Netzwerke unter dem Begriff *Graph* bekannt. Viele reale Systeme nehmen die Form eines Netzwerks an. Es existieren technologische Netzwerke wie das Internet oder das Stromnetz. Menschen leben in verschiedenen Arten von Beziehungen zu anderen Menschen, bewegen sich also in sozialen Netzwerken. Auch der menschliche Körper selbst kann in eine Vielzahl von biologischen Netzwerken gegliedert werden, so z. B. das Neuronale Netz des Gehirns oder das Netzwerk der Protein-Interaktionen. Netzwerke spielen in vielen wissenschaftlichen Disziplinen wie z. B. der Mathematik, Informatik, Physik, Biologie und Soziologie eine wichtige Rolle.

Eine wichtige Eigenschaft vieler Netzwerke ist, dass diese eine *Gruppenstruktur* (engl. *community structure*) aufweisen [NGo4]. Dies bedeutet, dass einzelne Elemente in enger Verbindung zueinander stehen und als eine zusammengehörende Gruppe angesehen werden können. Diese Gruppen werden auch als *Cluster* bezeichnet. Im Englischen hat sich der Begriff *community* für Gruppen in Netzwerken durchgesetzt. Ein Netzwerk kann meist in eine Vielzahl solcher Gruppen zerlegt werden, wobei diese in sich sehr dicht, untereinander aber nur schwach verbunden sind [NGo4]. Zur Bestimmung der Gruppen wird ein sog. Clustering-Algorithmus verwendet. Die identifizierte Gruppenstruktur kann wichtige Einsichten in die Struktur des realen Systems geben.

Abbildung 1.1 zeigt ein Netzwerk von Wissenschaftlern und deren Mitautorenschaft bei Veröffentlichungen. Hierbei werden die Wissenschaftler durch Knoten repräsentiert und eine Kante zwischen zwei Knoten sagt aus, dass die betreffenden Personen gemeinsam eine wissenschaftliche Arbeit veröffentlicht haben. Durch einen Clustering-Algorithmus wurden Gruppen von Wissenschaftlern bestimmt, welche untereinander sehr dicht verbunden sind. Die Gruppen sind in der Abbildung farblich hervorgehoben. Diese Gruppen können bspw. Hinweise auf verschiedene Forschungsgebiete der Wissenschaftler geben und zeigen, wie auch diese Gebiete untereinander vernetzt sind.

Netzwerke sind in der Regel keine statischen Gebilde, sondern verändern sich im Laufe der Zeit, sind also dynamisch. So wechselt bspw. die Struktur und Größe des Internets ständig. Neue Rechenzentren werden in Betrieb genommen, dafür alte abgeschaltet und in vielen Teilen der Welt erhalten Menschen erstmals Zugang zum World Wide Web. Auch soziale Netzwerke sind einem Wandel unterworfen, da sich menschliche Beziehungen im Laufe

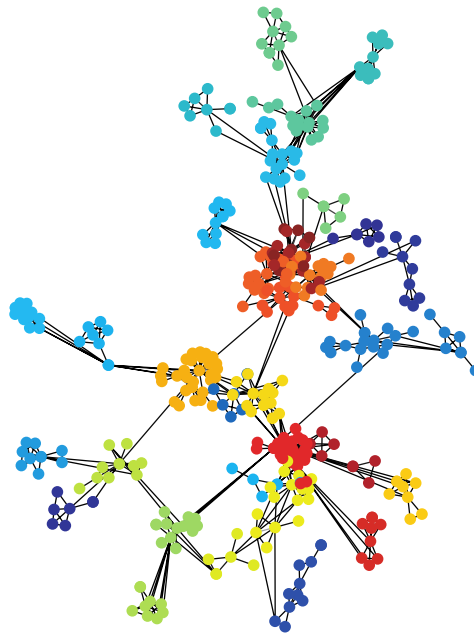


Abbildung 1.1.: Netzwerk von Wissenschaftlern – Die Kreise (Knoten) repräsentieren Wissenschaftler, eine Linie (Kante) zwischen zwei Kreisen deutet an, dass die betreffenden Personen gemeinsam eine wissenschaftliche Arbeit veröffentlicht haben. Die einzelnen Gruppen sind farblich markiert. [POM09]

des Lebens verändern. Die zeitliche Entwicklung von Netzwerken hat zur Folge, dass sich auch die darin enthaltenen Gruppenstrukturen verändern. Die einzelnen Gruppen besitzen Lebenszyklen [PBV07]: Sie entstehen, wachsen an und verschwinden womöglich auch wieder innerhalb der Beobachtungszeit. Das Nachverfolgen dieser Evolution kann in großem Maß zum Verständnis realer dynamischer Netzwerke beitragen.

1.2. Ziel

Ziel dieser Arbeit ist daher die Entwicklung einer Visualisierung zur Analyse von Gruppen in dynamischen Netzwerken. Die Visualisierung soll die zeitliche Evolution der Gruppen sichtbar und damit nachvollziehbar machen. Zudem soll auch die Struktur des Netzwerks zu jedem Zeitpunkt angezeigt werden können. Einzelne Elemente sollen selektiert werden können, um so deren Gruppenzugehörigkeit im Verlauf der Zeit hervorzuheben. Ebenso sollen auch Gruppen selektiert werden können, um ihre vergangene und zukünftige Entwicklung hervorzuheben. Elemente, welche häufig ihre Gruppenzugehörigkeit ändern, sollen visuell von konstanten Elementen unterschieden werden können. Der Fokus dieser Arbeit soll auf sozialen Netzwerken liegen, auch wenn die Visualisierung später für unterschiedliche Arten von Netzwerken eingesetzt werden kann.

Gliederung der Arbeit

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Grundlagen und Definitionen liefert den theoretischen Hintergrund für das Themengebiet dieser Arbeit.

Kapitel 3 – Clustering-Algorithmen geht auf die verschiedenen Ansätze zum Clustering in Graphen ein.

Kapitel 4 – Verwandte Visualisierungskonzepte gibt einen Überblick bereits bestehender Visualisierungen für Netzwerke und Gruppen.

Kapitel 5 – Konzept stellt das grundlegende Konzept der zu entwickelnden Visualisierung vor.

Kapitel 6 – Umsetzung zeigt wie das Konzept in der Visualisierung konkret umgesetzt wurde.

Kapitel 7 – Fallstudien beschreibt den praktischen Einsatz der entwickelten Visualisierung bei der Analyse zweier realer Datensätze.

Kapitel 8 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor.

2. Grundlagen und Definitionen

2.1. Graph

Netzwerke werden formal durch Graphen beschrieben. Graphen und deren Eigenschaften werden in einem Teilgebiet der diskreten Mathematik erforscht, der Graphentheorie.

Ein Graph ist ein Tupel (V, E) , wobei $V = \{v_1, \dots, v_n\}$ eine Menge von Knoten und $E = \{e_1, \dots, e_m\}$ mit $E \subseteq V^2$ eine Menge von Kanten bezeichnet. Eine Kante ist folglich ein Tupel (v_i, v_j) , welches zwei Knoten verbindet.

Ein Graph kann *gerichtet* oder *ungerichtet* sein. Bei einem ungerichteten Graphen spielt die Ordnung der Knoten einer Kanten keine Rolle, d. h. $(v_i, v_j) = (v_j, v_i)$. Bei einem gerichteten Graphen ist die Ordnung hingegen entscheidend, also $(v_i, v_j) \neq (v_j, v_i)$. Jeder Kante kann – unabhängig davon ob sie gerichtet oder ungerichtet ist – ein Gewicht $w_{v_i, v_j} \in \mathbb{R}$ zugewiesen werden. So erhält man einen *gewichteten Graph*.

Graphen können auch bildhaft dargestellt werden. Hierbei werden meist die Knoten als Kreise und die Kanten als Linien zwischen den Kreisen gezeichnet. Diese Darstellung wird im Englischen als *node-link diagram* bezeichnet; wir verwenden den Begriff *Knoten-Kanten-Diagramm*. Bei einem gerichteten Graphen können die Kanten z. B. als Pfeile abgebildet werden, um so die Richtung der Kante anzuzeigen. Bei gewichteten Graphen kann das Gewicht als Dicke der Kantenlinien dargestellt werden. Abbildung 2.1 zeigt die Knoten-Kanten-Diagramme der oben beschriebenen Typen von Graphen.

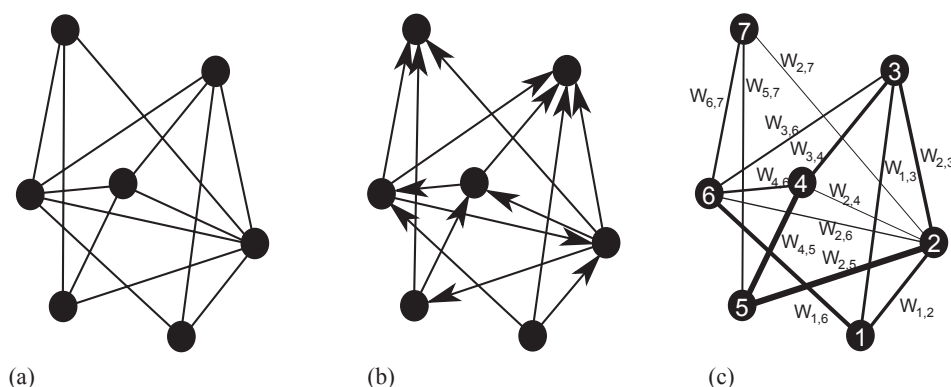


Abbildung 2.1.: Grafische Darstellung eines ungerichteten (a), eines gerichteten (b) und eines gewichteten (c) Graphen als Knoten-Kanten-Diagramm [BLM⁺o6]

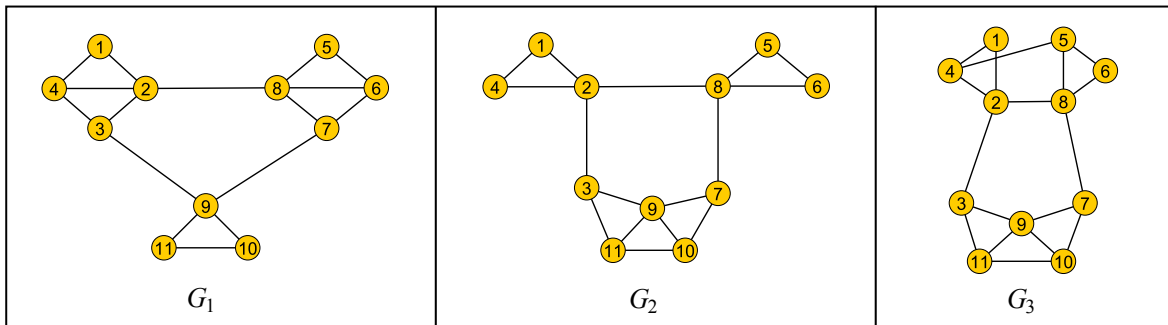


Abbildung 2.2.: Dynamischer Graph über drei Zeitschritte

2.2. Dynamischer Graph

Auch für dynamische Netzwerke muss ein formales Modell gefunden werden. Wie in den Arbeiten [GDC10], [TFSZ11], [LCZ⁺08] und [KH09] wird auch in dieser Arbeit ein dynamisches Netzwerk durch eine Serie von statischen Graphen $G = \{G_1, \dots, G_n\}$ repräsentiert. G wird dann als *dynamischer Graph* bezeichnet. $G_t = (V_t, E_t)$ mit $1 \leq t \leq n$ entspricht einer Momentaufnahme (engl. *snapshot*) des dynamischen Netzwerks zu einem bestimmten Zeitpunkt oder Zeitintervall. Meist ist der zeitliche Abstand zwischen den Momentaufnahmen gleich groß und so repräsentiert G_t den Zeitschritt (engl. *timestep*) t des dynamischen Netzwerks. Die Menge der Knoten in den Zeitschritten kann variieren, so dass im Allgemeinen nicht $V_t = V_{t+1}$ gelten muss. Auf diese Weise können Elemente modelliert werden, die nicht in allen Zeitschritten im dynamischen Netzwerk präsent sind. Abbildung 2.2 zeigt einen dynamischen Graphen über drei Zeitschritte.

2.3. Gruppen

Wie eingangs erwähnt, besitzen viele reale Netzwerke/Graphen die Eigenschaft, dass ihre Knoten in Gruppen (Cluster) eingeteilt werden können [NG04]. Für eine Gruppe gilt, dass „es mehr Kanten ‚innerhalb‘ der Gruppe geben muss als Kanten, die Knoten der Gruppe mit dem Rest des Graphen verbinden“ [For10]. Abbildung 2.3 veranschaulicht diesen Zusammenhang. Es existiert eine Vielzahl von Verfahren, um Gruppen in statischen sowie dynamischen Graphen zu bestimmen (siehe Kapitel 3).

Eine Gruppenstruktur, bei welcher jeder Knoten des Graphen genau einer Gruppe zugeordnet wird, ist formal eine Partition $P = \{C_1, \dots, C_k\}$ der Knoten-Menge V . Eine Partition ist eine Zerlegung einer Menge in nichtleere, disjunkte Teilmengen. Bei einem dynamischen Graphen muss eine Gruppenstruktur für jeden Zeitschritt bestimmt werden. Das Ergebnis eines Clusterings ist also eine Serie von Partitionen $\mathbb{P} = \{P_1, \dots, P_n\}$. $P_t = \{C_{t1}, \dots, C_{tk_t}\}$ mit $1 \leq t \leq n$ ist dabei die Partition von G_t .

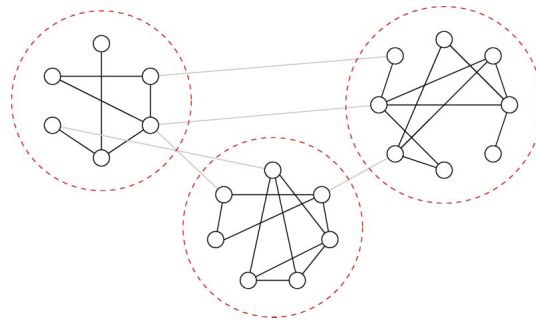


Abbildung 2.3.: Graph mit Gruppenstruktur – Die einzelnen Gruppen sind gestrichelt umrandet. [NG04]

2.4. Evolution von Gruppen

Genau wie sich Netzwerke dynamisch verändern, so wandeln sich auch die Gruppenstrukturen innerhalb der Netzwerke. Palla et al. [PBV07] unterscheiden sechs Ereignisse bei der Evolution von Gruppen: Wachstum, Verkleinerung, Vereinigung, Aufteilung, Entstehung und Auflösung (siehe Abb. 2.4). Die Ereignisse markieren wichtige Punkte im Lebenszyklus einer Gruppe [PBV07][GDC10]. Um die Entwicklung von Gruppen nachverfolgen zu können, müssen diese Ereignisse in den Daten identifiziert werden.

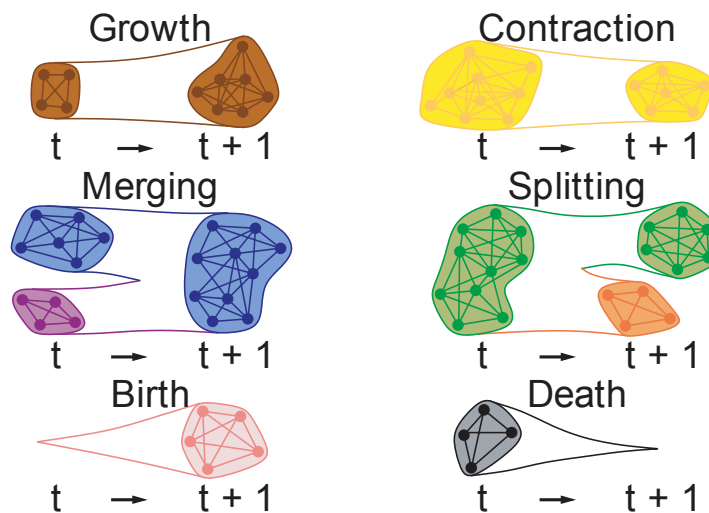


Abbildung 2.4.: Ereignisse bei der Evolution von Gruppen [PBV07]

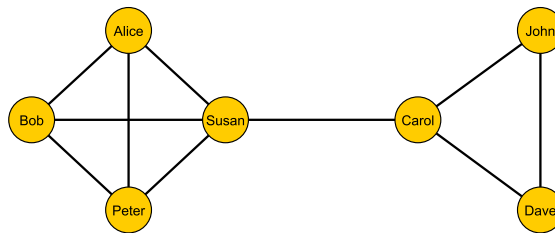


Abbildung 2.5.: Soziales Netzwerk von Freundschaftsbeziehungen – Knoten repräsentieren einzelne Personen und Kanten deuten an, dass die betreffenden Personen befreundet sind. Es lassen sich zwei untereinander dicht verbundene Freundeskreise erkennen.

2.5. Soziales Netzwerk

Ein soziales Netzwerk ist eine abstrakte Repräsentation der sozialen Interaktionen zwischen Akteuren [RTJ⁺11]. Ein Akteur kann eine einzelne Person sein, sowie eine Gruppe von Personen. Auch Staaten und deren politische Beziehungen können als ein soziales Netzwerk aufgefasst werden. Abbildung 2.5 zeigt beispielhaft ein kleines soziales Netzwerk von Freundschaftsbeziehungen.

Der Begriff des sozialen Netzwerks stammt aus den Sozialwissenschaften, welche bereits in den 1920er Jahren begannen, soziale Interaktionen quantitativ zu analysieren [Fre96]. Überhaupt waren die Sozialwissenschaften eine der ersten Disziplinen, die sich – neben der Graphentheorie – der systematischen Erforschung und Formalisierung von Netzwerken widmeten [BLM⁺06]. Der interdisziplinäre Forschungszweig, welcher sich mit der methodischen Analyse von sozialen Netzwerken befasst ist *Social Network Analysis*.

Durch den technologischen Fortschritt der letzten Jahrzehnte haben sich ganz neue Möglichkeiten ergeben, Daten zu erheben. Während in früheren Zeiten aufwändige und zeitintensive Befragungen einzelner Personen durchgeführt werden mussten, können heute einfach Kommunikationsprotokolle verschiedener Dienste wie z. B. E-Mail, Chat oder Telefonie verwendet werden, um Daten im Bezug auf soziale Netzwerke zu gewinnen. Das Internet selbst beinhaltet eine Vielzahl von digitalen sozialen Strukturen, wie Communities oder Foren. Auch die rapide anwachsenden *social networking services* (umgangssprachlich auch als „Soziale Netzwerke“ bezeichnet), wie *facebook*¹ oder *Google+*², könnten in Zukunft als Datenbasis verwendet werden. Neue Technologien können auch die zeitliche Veränderung sozialer Netzwerke detailliert aufzuzeichnen und ermöglichen so einen Wechsel von der Analyse statischer Netzwerke hin zu dynamischen Netzwerken [RTJ⁺11].

¹<http://www.facebook.com/>

²<http://plus.google.com/>

3. Clustering-Algorithmen

Unter einem Clustering-Algorithmus versteht man im Allgemeinen ein Verfahren zur Identifikation von Gruppen (Clustern) in Daten [ELLS11]. Ziel ist es, eine Gruppenstruktur zu bestimmen, wobei die Elemente einer Gruppen ähnliche Eigenschaften besitzen und sich von den Elementen anderer Gruppen unterscheiden [Romo4]. Sowohl das Einteilen der Elemente in Gruppen, als auch das Ergebnis (die Gruppenstruktur), wird als *Clustering* bezeichnet. Clustering-Algorithmen spielen in vielen Disziplinen wie z. B. der Biologie, Medizin, Psychologie oder dem Marketing eine wichtige Rolle [ELLS11]. Hier wird auch oftmals der Begriff *Cluster-Analyse* verwendet.

Diese Arbeit befasst sich speziell mit dem Erkennen und Verfolgen von Clustern in dynamischen Netzwerken bzw. Graphen. Ein Cluster ist hier eine Menge von Knoten. Das algorithmische Berechnen einer Gruppenstruktur in Graphen wird im Englischen auch als *community detection* bezeichnet.

3.1. Clustering statischer Graphen

Ein statisches Netzwerk wird durch einen Graphen modelliert. Es existieren eine Vielzahl von Algorithmen, um Gruppenstrukturen in Graphen zu berechnen, z. B. von Girvan und Newman [GN02], Blondel et al. [BGLLo8] oder Rosvall und Bergstrom [RB08]. Fortunato [For10] liefert einen ausführlichen Überblick der verschiedenen Clustering-Algorithmen für Graphen und erläutert deren Ansätze. Lancichinetti und Fortunato [LF09] vergleichen und bewerten einige Algorithmen mithilfe von speziell konstruierten Benchmark-Graphen.

Standard-Algorithmen ordnen jeden Knoten genau einer Gruppe zu, was mathematisch einer Partition der Knoten entspricht. In vielen realen Graphen ist dies jedoch nicht immer sinnvoll, da die Gruppen teilweise überlappen [PDFV05]. So können bspw. Personen in sozialen Netzwerken gleichzeitig mehreren Gruppen angehören. Als Beispiel kann das Netzwerk der eigenen Bekanntschaftsbeziehungen betrachtet werden: Man ist Teil einer Familie, Teil einer Gruppe von Arbeitskollegen und meist auch Teil verschiedener Freizeitgruppen. Abbildung 3.1 zeigt einen Graphen mit vier sich teilweise überlappenden Gruppen. Die bekannteste Technik zur Bestimmung von überlappenden Gruppen in Graphen ist die *Clique Percolation Method* von Palla et al. [PDFV05].

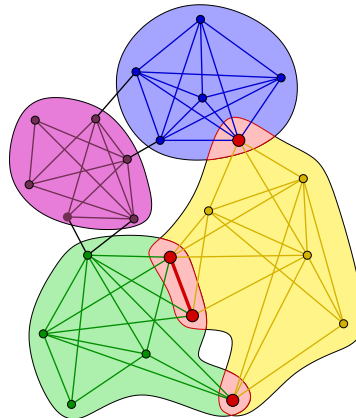


Abbildung 3.1.: Graph mit überlappenden Gruppen – Knoten, welche mehreren Gruppen zugeordnet sind, sind rot eingefärbt. [PDFV05]

3.2. Clustering dynamischer Graphen

3.2.1. Lokales Clustering

Beim Clustering eines dynamischer Graphen muss für jeden Zeitschritt eine Gruppenstruktur bestimmt werden. Der naheliegende Ansatz ist es, für jeden Graphen G_t mithilfe eines statischen Clustering-Algorithmus separat eine Gruppenstruktur P_t zu berechnen. Wir bezeichnen dieses Vorgehen als *lokales Clustering*. Das lokale Clustering hat den Nachteil, dass sich die Gruppenstrukturen P_t und P_{t+1} stark unterscheiden können, was oft an störanfälligen Daten liegt [For10]. Sinnvoll wäre daher eine Art „Glättung“ der Gruppenstrukturen über die Zeit.

3.2.2. Evolutionäres Clustering

In sozialen Netzwerken verändern sich Gruppen meist nicht schlagartig, sondern entwickeln sich schrittweise [BHKLo6]. Um diese Eigenschaft zu berücksichtigen, ist es sinnvoll beim Clustering nicht nur jeweils einen Zeitschritt isoliert zu betrachten, sondern stattdessen mehrere Zeitschritte einzubeziehen. Diese Idee liegt dem von Chakrabarti et al. [CKTo6] entwickelten *evolutionären Clustering* zu Grunde.

Chakrabarti et al. führen zwei Metriken ein: Die *snapshot quality* und die *history cost*. Sei P_t die Partition des Graphen G_t , so bewertet die *snapshot quality* die Qualität von P_t allein im Bezug auf G_t . Die *history cost* ist ein Maß für den Unterschied zwischen P_t und P_{t-1} . Ziel von evolutionären Clustering-Algorithmen ist es für jede Partition P_t die *snapshot quality* zu maximieren und gleichzeitig die *history cost* zu minimieren. Auf diese Weise wird erreicht, dass P_t die Struktur von G_t gut wiedergibt und darüber hinaus nur gering von P_{t-1} abweicht.

Verschiedene statische Clustering-Algorithmen können zu evolutionären Algorithmen umformuliert werden. So wurden evolutionäre Versionen des hierarchischen Clusterings [CKT06] und des spektralen Clusterings [CSZ⁺07] entwickelt. Auch zur Bestimmung der Werte für die snapshot quality und die history cost können unterschiedliche Metriken eingesetzt werden. Die Wahl des konkreten Clustering-Algorithmus, sowie der Metriken ist vom Anwendungsfall abhängig. Lin et al. [LCZ⁺08] und Kim und Han [KH09] erweitern das Konzept von Chakrabarti et al. und machen es möglich, weitere Ereignisse in der Evolution von Gruppen zu erkennen.

3.3. Gruppenverfolgung

Das Ergebnis des Clusterings eines dynamischen Graphen ist eine Gruppenstruktur für jeden Zeitschritt. Verfahren zur *Gruppenverfolgung* (engl. *community tracking*) vergleichen die Gruppen aufeinanderfolgender Zeitschritte und ordnen ähnliche Gruppen einander zu. Auf diese Weise werden Meta-Gruppen bestimmt, welche die Entwicklung einer realen Gruppe über die Zeit beschreiben. Verfahren zur Gruppenverfolgung wurden unter anderem von Greene et al. [GDC10], sowie von Takaffoli et al. [TFSZ11] entwickelt. Das Verfahren von Greene et al. wird im Folgenden genauer erläutert. Die identifizierten Meta-Gruppen werden hier als *dynamische Gruppen* (engl. *dynamic communities*) bezeichnet.

3.3.1. Dynamische Gruppe

Zur Wiederholung: Sei $\mathbb{G} = \{G_1, \dots, G_n\}$ ein dynamischer Graph und $P_t = \{C_{t1}, \dots, C_{tk_t}\}$ mit $1 \leq t \leq n$ die Partition der Knoten eines Zeitschritts t .

Greene et al. definieren eine dynamische Gruppe D als eine Menge von m Gruppen verschiedener Zeitschritte, $D = \{C_{t_1j_1}, \dots, C_{t_mj_m}\}$, wobei die Gruppen zeitlich geordnet sind, formal $t_1 \neq \dots \neq t_m$ und $t_1 < \dots < t_m$. Die einzelnen Gruppen der dynamischen Gruppe werden im Folgenden zur Unterscheidung als *Zeitschritt-Gruppen* bezeichnet. Die Zeitschritt-Gruppen weisen untereinander eine Ähnlichkeit auf, so dass die dynamische Gruppe als „Entwicklungsgeschichte“ ein und derselben realen Gruppe über die Zeit verstanden werden kann. Die Menge aller dynamischen Gruppen eines dynamischen Graphen ist $\mathbb{D} = \{D_1, \dots, D_l\}$.

Abbildung 3.2 zeigt beispielhaft vier dynamische Gruppen, welche über drei Zeitschritte identifiziert wurden:

- $D_1 = \{C_{11}, C_{21}, C_{31}\}$
- $D_2 = \{C_{12}, C_{21}, C_{31}\}$
- $D_3 = \{C_{13}, C_{22}, C_{32}\}$
- $D_4 = \{C_{13}, C_{23}, C_{33}\}$

3. Clustering-Algorithmen

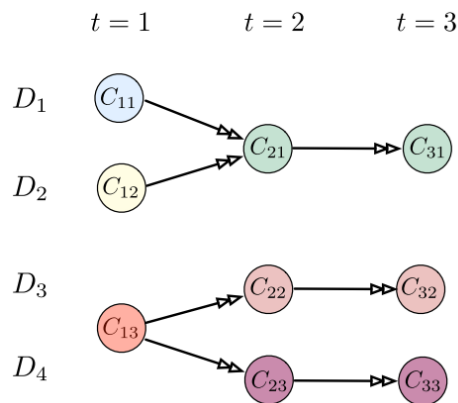


Abbildung 3.2.: Vier beispielhafte dynamische Gruppen über jeweils drei Zeitschritte [GDC10]

Während des Verfahrens zur Gruppenverfolgung werden die dynamischen Gruppen schrittweise aufgebaut. Die *Front* F_i einer dynamischen Gruppe D_i ist diejenige Gruppe, welche D_i zuletzt hinzugefügt wurde. Für die dynamischen Gruppen in Abbildung 3.2 sind die Fronten folgende:

- $F_1 = C_{31}$
- $F_2 = C_{31}$
- $F_3 = C_{32}$
- $F_4 = C_{33}$

Das Beispiel zeigt, dass auch mehrere dynamische Gruppen dieselbe Front besitzen können (vgl. F_1 und F_2).

3.3.2. Gruppenzuordnung

Um Gruppen aufeinanderfolgender Zeitschritte einander zuordnen zu können (z. B. C_{21} und C_{31} in Abb. 3.2), wird eine Ähnlichkeitsmetrik für Gruppen benötigt. Greene et al. verwenden hierfür den *Jaccard-Koeffizient* [Jac12]. Seien A und B zwei Mengen, so ist deren Ähnlichkeit:

$$\text{sim}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

3.3.3. Verfahren

Der eigentliche Algorithmus zur Identifikation der dynamischen Gruppen ist folgender [GDC₁₀]:

1. Initialisiere \mathbb{D} , indem für jede Zeitschritt-Gruppe $C_{1j} \in P_1$ eine dynamische Gruppe angelegt wird.
2. Für jeden Zeitschritt $t > 1$:
 - a) Für jede Gruppe $C_{tj} \in P_t$:
 - i. Finde alle dynamischen Gruppen D_i mit $\text{sim}(C_{tj}, F_i) > \theta$.
 - ii. Falls keine passende dynamische Gruppe gefunden wurde, erstelle eine neue dynamische Gruppe und füge C_{tj} hinzu.
 - iii. Andernfalls füge C_{tj} zu jeder gefundenen dynamischen Gruppe hinzu.
 - b) Aktualisiere die Front jeder dynamischen Gruppe, so dass diese die zuletzt hinzugefügte Zeitschritt-Gruppe ist. Wurden für eine dynamische Gruppe zwei oder mehr passende Zeitschritt-Gruppen gefunden, so spalte die dynamische Gruppe auf.

Als Eingabe erhält der Algorithmus die Gruppenstrukturen der einzelnen Zeitschritte $\mathbb{P} = \{P_1, \dots, P_n\}$. Diese müssen zuvor durch einen Clustering-Algorithmus bestimmt werden. Zudem muss dem Algorithmus ein Schwellenwert θ (engl. *threshold value*) übergeben werden. Über diesen Schwellenwert kann gesteuert werden, wie groß die Toleranz bei der Zuordnung der Gruppen ist. Im Allgemeinen gilt, dass je höher der Schwellenwert angesetzt wird, desto weniger dynamische Gruppen werden identifiziert. Durch die Verwendung der Fronten F_i ist der Algorithmus in der Lage, Gruppen einander zuzuordnen, welche nicht in allen Zeitschritten vorhanden sind.

3.3.4. Signifikanz

Bei größeren Datensätzen kann der Algorithmus als Ergebnis eine Vielzahl von dynamischen Gruppen liefern. Daher wäre es sinnvoll, eine Rangfolge für die dynamischen Gruppen entsprechend ihrer Bedeutsamkeit (Signifikanz), zu ermitteln. Greene et al. definieren daher ein Signifikanz-Maß, welches die Stabilität der Knotenmitgliedschaften und die Langlebigkeit von dynamischen Gruppen berücksichtigt. Demnach ist die Signifikanz einer dynamischen Gruppe $D = \{C_{t_1j_1}, \dots, C_{t_mj_m}\}$:

$$\text{sig}(D) = \frac{1}{n-1} \sum_{i=1}^{|D|-1} \text{sim}(C_{t_ij_i}, C_{t_{i+1}j_{i+1}})$$

3. Clustering-Algorithmen

Die Signifikanz ist die gemittelte Jaccard-Ähnlichkeit zwischen den einzelnen Zeitschritt-Gruppen einer dynamischen Gruppe. Eine stabile und lang bestehende dynamische Gruppe hat einen Signifikanz-Wert nahe „1“, wohingegen eine weniger stabile und nur punktuell auftretende dynamische Gruppe einen Wert nahe „0“ besitzt.

4. Verwandte Visualisierungskonzepte

4.1. Visualisierung statischer Graphen

Statische Graphen werden, wie bereits in Abschnitt 2.1 erwähnt, meist durch Knoten-Kanten-Diagramme dargestellt. In diesen Diagrammen werden die Knoten durch Kreise und die Kanten durch Linien repräsentiert. Layout-Algorithmen konstruieren die Diagramme, indem sie für einen gegebenen Graphen die Position der Knoten und Kanten in der Darstellung berechnen. In der Vergangenheit wurde eine Reihe von ästhetischen Regeln formuliert, welche die Wahrnehmung und Analyse eines Knoten-Kanten-Diagramms unterstützen sollen. So sollten z. B. die Knoten und Kanten in der Darstellung etwa gleichmäßig verteilt sein und die Anzahl der Kreuzungen von Kanten sollte minimal sein [HMM00]. Ein einzelner Layout-Algorithmus erfüllt nicht alle ästhetischen Regeln, sondern versucht das Layout hinsichtlich einer Teilmenge der Regeln zu optimieren. Einer der am meisten verwendeten Layout-Algorithmen ist das *force-directed placement* von Fruchterman und Reingold [FR91]. Ein Überblick der Layout-Algorithmen wird durch Battista et al. [BETT98] gegeben.

4.2. Visualisierung dynamischer Graphen

Dynamische Graphen wurden in Abschnitt 2.2 formal definiert. Die Ansätze zur Visualisierung von dynamischen Graphen lassen sich in zwei Kategorien einteilen [VLKS⁺11]: *Animationen* und *statische Ansichten*.

4.2.1. Animation

Ein naheliegender Ansatz ist, die einzelnen statischen Graphen zeitlich nacheinander in einer Animation anzuzeigen. Zu beachten ist, dass sich das Layout von G_{t-1} und G_t nicht zu stark ändern sollte, um das mentale Bild des Betrachters (engl. *mental map*) zu bewahren [PHGo6]. Aus diesem Grund kann das Layout von G_t nicht isoliert mit einem statischen Layout-Algorithmus bestimmt werden, sondern muss unter Berücksichtigung des Layouts von G_{t-1} ermittelt werden. Ändern sich die Positionen der Knoten, so kann wie in [MMBdo5] zwischen den Positionen interpoliert werden, um weitere Zwischenbilder zu generieren und eine flüssige Animation zu erreichen.

4. Verwandte Visualisierungskonzepte

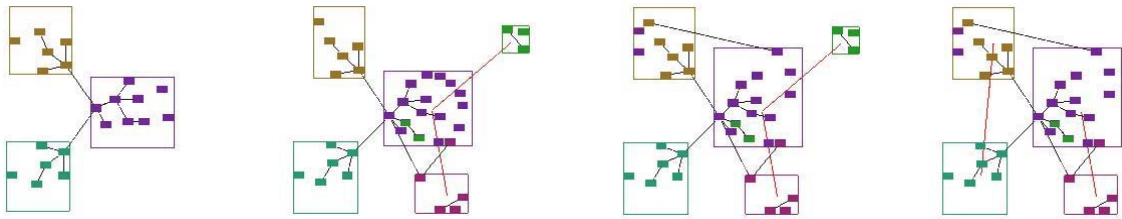


Abbildung 4.1.: Momentaufnahmen einer Animationssequenz aus [FT04] (von links nach rechts)

Frishman und Tal [FT04] befassen sich mit der Animation von geclusterten Graphen. Die Knoten eines Clusters sind dabei immer dicht beieinander angeordnet und von einem Rahmen umgeben (siehe Abb. 4.1). Knoten und Rahmen sind je nach Cluster unterschiedlich eingefärbt, was die Wahrnehmung der Gruppenstruktur unterstützt. Mithilfe eines speziellen Algorithmus, welcher das Layout für die einzelnen Graphen berechnet, wird versucht, den Unterschied zwischen zwei Zeitpunkten möglichst gering zu halten. So fällt es einem Betrachter leichter die Entwicklung der Gruppen zu verfolgen und sein mentales Bild zu bewahren.

Die menschliche Wahrnehmungsfähigkeit, vor allem das begrenzte Kurzzeitgedächtnis, spielt eine wichtige Rolle bei Animationen und schränkt deren Effektivität bei der Visualisierung von dynamischen Graphen ein [VLKS⁺11][BVB⁺11]. So können Betrachter meist nur größere Veränderungen zwischen den einzelnen Graphen wahrnehmen [VLKS⁺11]. Längere Zeitabschnitte können mit Animationen nur unzureichend analysiert werden, da die komplexe Struktur von Graphen früherer Zeitpunkte nur schwer im Gedächtnis behalten werden kann [BBV⁺12]. Zudem können plötzliche, drastische im Änderungen im Layout den Betrachter irritieren und sein mentales Bild zerstören [BVB⁺11].

Soll die Entwicklung einer Gruppe mithilfe einer Animation verfolgt werden, so muss die betreffende Gruppe in jedem Teilbild der Animation identifiziert werden. Ebenso verhält es sich, wenn die Gruppenzugehörigkeit eines einzelnen Knotens im Verlauf der Zeit analysiert werden soll. Verändert sich das Layout von einem Zeitpunkt auf den nächsten stark, fällt es einem Betrachter schwer, vorher beobachtete Gruppen oder Knoten in der neuen Ansicht wiederzufinden. Zwar können Verbesserungen, wie z. B. in der Arbeit von Frishman und Tal [FT04], das Verfolgen der Gruppenentwicklung erleichtern, dennoch erfordert es eine hohe kognitive Anstrengung. Das Analysieren langer Zeitabschnitte und das Erkennen kleiner Änderungen bleibt nach wie vor ein Problem.

4.2.2. Statische Ansichten

Bei statischen Ansichten muss die zeitliche Dimension eines dynamischen Graphen auf eine räumliche Dimension abgebildet werden. Im Folgenden werden zwei mögliche Ansätze genauer erläutert.

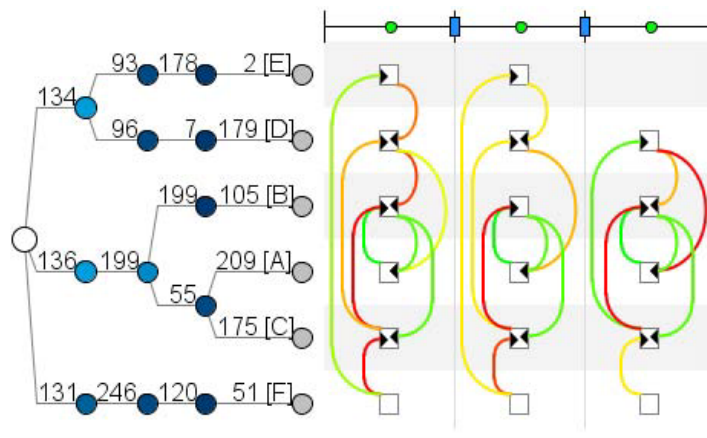


Abbildung 4.2.: TimeArcTrees [GBD09]

Small Multiples

Überträgt man das Konzept der *Small Multiples* [Tuf90] auf dynamische Graphen, so erhält man eine statische Ansicht, bei welcher die einzelnen Graphen G_t nebeneinander oder in einem Gitter angeordnet sind. Wie bei der Animation ist darauf zu achten, dass sich das Layout von G_t auf G_{t+1} nicht zu stark ändert, um das mentale Bild des Betrachters zu bewahren. Abbildung 4.1 zeigt Momentaufnahmen einer Animationssequenz und kann als eine Small-Multiples-Ansicht aufgefasst werden. Gegenüber der Animation bietet das Konzept der Small Multiples den Vorteil, dass auch zeitlich weit auseinanderliegende Graphen miteinander verglichen werden können.

Ein Beispiel für diesen Ansatz stellen die *TimeArcTrees* von Greilich et al. [GBD09] dar (siehe Abb. 4.2). Diese dienen der Visualisierung von gerichteten dynamischen Graphen, deren Knoten hierarchisch geordnet sind (im Englischen als *compound graph* bezeichnet). Jeder Graph G_t wird durch ein Knoten-Kanten-Diagramm dargestellt, wobei die Knoten auf einer vertikalen Linie angeordnet sind. Kanten, welche links der Knoten gezeichnet werden, verlaufen von unten nach oben (*upward edges*), wohingegen die Kanten rechts der Knoten immer von oben nach unten führen (*downward edges*). Das Gewicht einer Kante wird entsprechend einer Skala auf ihre Farbe abgebildet. Die einzelnen Diagramme sind nun von links nach rechts in ihrer zeitlichen Reihenfolge angeordnet. Ganz links ist die Hierarchie der Knoten ebenfalls als Knoten-Kanten-Diagramm zu sehen. Da gleiche Knoten jeweils auf einer horizontalen Linie liegen, lässt sich gut erkennen, wie sich die Kanten im Verlauf der Zeit verändert haben.

Small Multiples unterliegen teilweise den gleichen Beschränkungen wie Animationen [OMB⁺07]: Änderungen in großen, komplexen Graphen können übersehen werden und drastische Änderungen im Layout zeitlich naher Graphen machen es dem Betrachter schwer, die Entwicklung zu verfolgen. Um die Evolution von Gruppen nachzuvollziehen, müssen diese – wie bei der Animation – in jedem Teilbild zunächst identifiziert werden. Sind einzelne

4.3. Visualisierung der Evolution von Gruppen

Im Folgenden werden Visualisierungen vorgestellt, deren Ziel es ist, die Entwicklung von Gruppen in Daten sichtbar zu machen. Nicht alle Visualisierungen wurden speziell für Gruppen in dynamischen Graphen entwickelt, dennoch können die zugrundeliegenden Konzepte auf dynamische Graphen übertragen werden.

4.3.1. Ogawa et al.

Ogawa et al. [OMB⁺07] stellen eine Visualisierung zur Analyse der sozialen Interaktionen in Open-Source Software-Projekten vor. An diesen Projekten arbeiten meist viele hundert Software-Entwickler mit, die sich in Gruppen unterschiedlichen Aufgaben widmen. Ziel der Visualisierung ist es, die Evolution dieser Gruppen von Entwicklern sichtbar zu machen und so die Struktur und den Verlauf des Zusammenarbeitens in großen Software-Projekten besser zu verstehen. Als Eingabedaten für die Visualisierung dienen das Quellcode-Repository und die Mailingliste des Projektes. Die Mailingliste kann als gewichteter dynamischer Graph modelliert werden, indem der E-Mail-Austausch in jedem Monat zusammengefasst wird. Die Entwickler werden dabei durch Knoten und eine E-Mail zwischen zwei Entwicklern durch eine Kante repräsentiert. Das Kantengewicht entspricht der Anzahl der im jeweiligen Monat ausgetauschten E-Mails. In jedem Zeitschritt der Mailingliste können, basierend auf dem Graphen, Gruppen von Entwicklern identifiziert werden, die einen regen Austausch unterhalten haben.

Die aus den Eingabedaten erzeugte Visualisierung besteht aus zwei Ansichten: Der *Mailinglisten-Ansicht* und der *Repository-Ansicht*. In der Mailinglisten-Ansicht wird die Entwicklung der Gruppen visualisiert und in der Repository-Ansicht wird das Repository baumartig, wie in gängigen Datei-Explorern, dargestellt. Unser Fokus soll auf der Mailinglisten-Ansicht liegen. Bevor die Mailinglisten-Ansicht generiert werden kann, müssen die einzelnen Zeitschritte geclustert werden. Hierfür wird der *Markov Cluster Algorithmus* (MCL) [Donoo] verwendet. Die Visualisierung selbst basiert auf dem Konzept der *Sankey Diagrams*. Sankey Diagrams werden hauptsächlich dazu verwendet, um den Energie- oder Materialfluss durch physische Systeme darzustellen. Ogawa et al. adaptieren die Diagramme nun, um den „Fluss“ von Entwicklern zwischen den Gruppen aufeinanderfolgender Zeitschritte sichtbar zu machen.

Abbildung 4.4 zeigt die Mailinglisten-Ansicht für 12 Zeitschritte. Die Zeitachse verläuft in vertikaler Richtung. In jedem Zeitschritt werden die Entwickler durch kleine Ovale dargestellt, die auf einer horizontalen Linie angeordnet sind. Die identifizierten Gruppen sind durch Abstände voneinander getrennt und von links nach rechts in aufsteigender Größe platziert. Zwischen den Zeitschritten werden Linien gezeichnet, welche immer genau zwei Gruppen verbinden. Die Breite einer Linie entspricht der Anzahl an Entwicklern, welche von einer Gruppe in die andere übergehen. Die Evolution der Gruppen kann nun nachvollzogen werden, indem die Ansicht von oben nach unten betrachtet wird.

4. Verwandte Visualisierungskonzepte

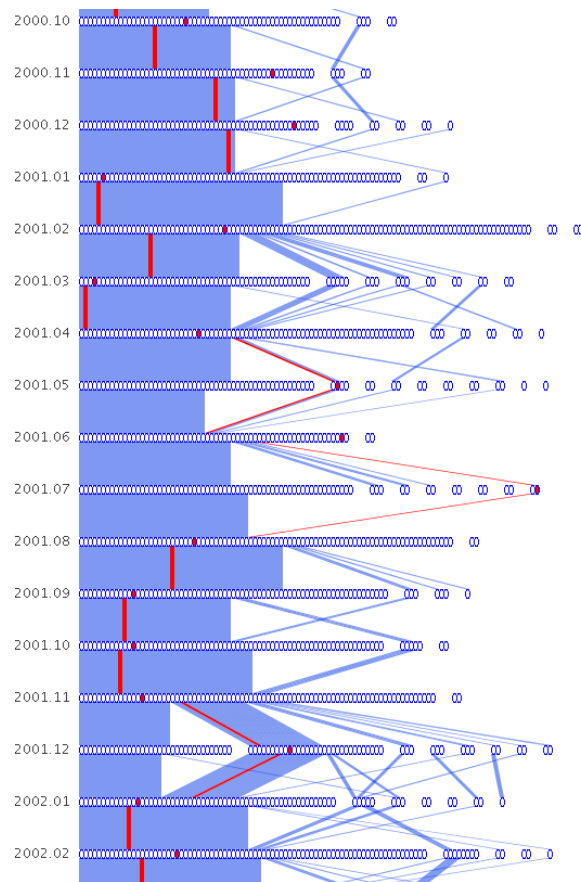


Abbildung 4.4.: Mailinglisten-Ansicht von [OMB⁺07]

Da die Gruppen einfach nur nach ihrer Größe angeordnet sind, kann es dazu kommen, dass sich die Linien zwischen zwei Zeitschritten kreuzen. Vor allem breite Linien können mehrere schmale Linien verdecken. Dadurch kann die Entwicklung einzelner Gruppen nur schwer nachvollzogen werden. Um das Problem der Überlagerung von Linien zu mindern, werden diese semi-transparent gezeichnet, sodass an Kreuzungen alle Linien weiterverfolgt werden können.

Mit der Ansicht kann auf vielfältige Weise interagiert werden. Der visuelle Abstand zwischen zwei Zeitschritten kann über einen Schieberegler angepasst werden, um so die vertikale Ausdehnung der Ansicht festzulegen. Der Abstand kann zunächst sehr klein gewählt werden, um einen Überblick zu gewinnen. Anschließend kann der Zwischenraum vergrößert werden, um klar den Übergang der Elemente verfolgen zu können. Durch die Selektion eines Entwicklers wird dieser in allen Zeitschritten hervorgehoben und auch innerhalb der Linien markiert (rot in Abb. 4.4).

Bewertung

Die Visualisierung von Ogawa et al. besitzt eine einfache und klare Darstellung. Die zu den einzelnen Zeitschritten identifizierten Gruppen und ihre Entwicklung sind gut erkennbar. Bei größeren Datensätzen mit vielen kleinen Gruppen, könnten die Kreuzungen von Linien allerdings zunehmen und das Nachvollziehen der Entwicklung erschweren. Kreuzen sich viele Linien zwischen zwei Zeitschritten, kann nicht mehr auf einen Blick erkannt werden, wohin einzelne Elemente wandern. Die Linien müssen gedanklich „entwirrt“ werden, indem sie einzeln verfolgt werden. Daher wäre es sinnvoll, die Gruppen nicht einfach ihrer Größe nach anzuordnen, sondern so zu platzieren, dass die Kreuzungen der Linien zwischen Zeitschritten minimiert werden.

Ein weiterer Schwachpunkt betrifft die Selektion: Wird ein Element selektiert, so verlaufen die hervorgehobenen Linien nicht genau zwischen den hervorgehobenen Elementen, d. h. der Start- bzw. Endpunkt der Linie ist meist ein anderes Element. Dies liegt daran, dass die Linien zwischen zwei Zeitschritten nicht direkt gleiche Elemente miteinander verbinden, sondern nur andeuten, *wie viele* Elemente insgesamt von einer Gruppe zu einer anderen übergehen. Das visuelle Resultat ist ein „zerstückelter“ Pfad, der mit dem Auge nur schwer verfolgt werden kann (siehe Abb. 4.4). Bei mehreren gleichzeitig selektierten Elementen kann zudem eine Linie nicht mehr eindeutig einem Element zugeordnet werden.

Außerdem bietet die Visualisierung keine Möglichkeit, sich die Graphen der einzelnen Zeitschritte anzeigen zu lassen. Gerade um die Entwicklung von sozialen Gruppen besser erforschen zu können, wäre dies allerdings notwendig. So könnte die interne Struktur der Gruppen genauer untersucht werden. Es wäre bspw. zu erwarten, dass sehr langlebige Gruppen über dichte interne Verbindungen verfügen.

4.3.2. Matchmaker

Lex et al. [LSP⁺10] haben die Visualisierung *Matchmaker* entwickelt, um multidimensionale, quantitative Daten zu analysieren. Die Visualisierung ermöglicht es, verschiedene Gruppen von Dimensionen miteinander zu vergleichen.

Hinweis: Die Begriffe „Gruppe“ und „Cluster“ sind in Matchmaker nicht synonym zu verstehen. Als Gruppe wird eine „Gruppe von Dimensionen“ bezeichnet. Gruppen sind weiter unterteilt in Cluster.

Als Eingabe erhält Matchmaker eine Matrix aus Zahlenwerten, wobei die Spalten Dimensionen und die Zeilen Datenelementen (Tupeln) entsprechen (siehe Abb. 4.5a). Zunächst definiert der Benutzer mehrere Gruppen von Dimensionen (siehe Abb. 4.5b). Eine Dimension der Daten kann dabei Teil mehrerer Gruppen sein. Die Gruppen werden in der Praxis meist so gebildet, dass die beteiligten Dimensionen einen Zusammenhang aufweisen, z. B. Ergebniswerte desselben Experiments sind. Anschließend wird jede Gruppe separat durch einen Clustering-Algorithmus in einzelne Cluster zerlegt (siehe Abb. 4.5c).

4. Verwandte Visualisierungskonzepte

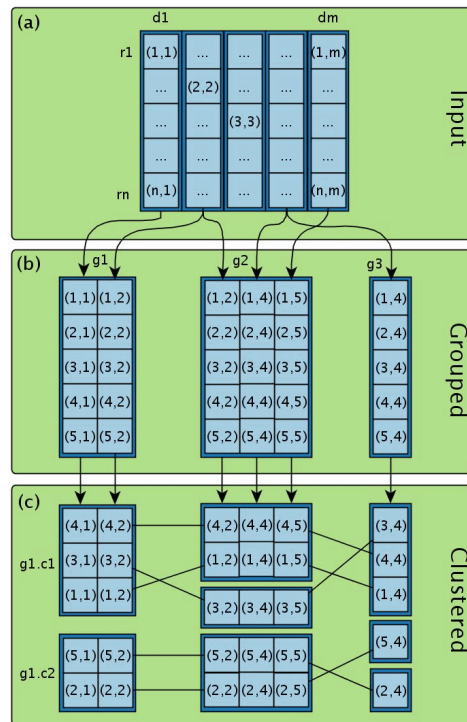


Abbildung 4.5.: Arbeitsablauf in Matchmaker – Die tabellarischen Eingabedaten (a) werden beliebig gruppiert (g_1, g_2, g_3 in (b)). Die Gruppen werden dann einzeln geclustert (z. B. c_1, c_2 für g_1) und sind damit für die Analyse bereit. [LSP⁺ 10]

Nach der Vorverarbeitung durch den Benutzer, können die geclusterten Gruppen visualisiert werden. Die Gruppen werden als *Heatmaps* dargestellt, wobei die Zahlenwerte auf die Farbe abgebildet werden. Die Cluster sind in den Gruppen entsprechend ihres Mittelwertes vertikal angeordnet. Auf die gleiche Weise sind auch die einzelnen Elemente in den Clustern platziert. Um nun die Gruppen vergleichen zu können, sind Elemente, welche Teil der gleichen Zeile der Eingabedaten sind, durch Linien verbunden. Um die Anzahl der Linien und deren Kreuzungen zwischen zwei Gruppen zu minimieren, werden die Linien teilweise zusammengefasst, sodass nur noch wenige, breite Bündel zwischen den Gruppen verlaufen (siehe Abb. 4.6).

Die Bündelung der Linien wird nicht durch Umsortieren der Elemente in den Clustern erreicht, sondern geschieht rein visuell durch die Einführung sog. Support-Punkte (grün in Abb. 4.7). Jedem Cluster werden, entsprechend seiner Anzahl an Elementen, Support-Punkte zugeordnet. Zusammengehörige Elemente werden nicht mehr direkt durch Linien verbunden, sondern durch die Support-Punkte geleitet. Die Support-Punkte sind nach dem Ziel-Cluster ihrer Verbindung vertikal geordnet. Dieser Ansatz verringert die Anzahl der Kreuzungen zwischen den Support-Punkten benachbarter Gruppen und verlagert die Kreuzungen in einen schmalen Bereich zwischen den Clustern und ihren zugehörigen Support-Punkten.

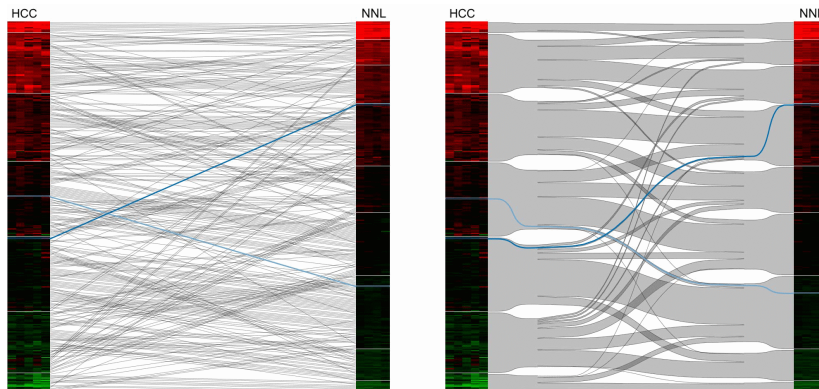


Abbildung 4.6.: Bündelung der Verbindungen in Matchmaker – Links: Die zusammenhängenden Elemente zweier Gruppen werden naiv durch Linien verbunden. Rechts: Anstatt einfacher Linien werden geschwungene Spline-Kurven verwendet und diese zu Bündeln zusammengefasst. [LSP⁺ 10]

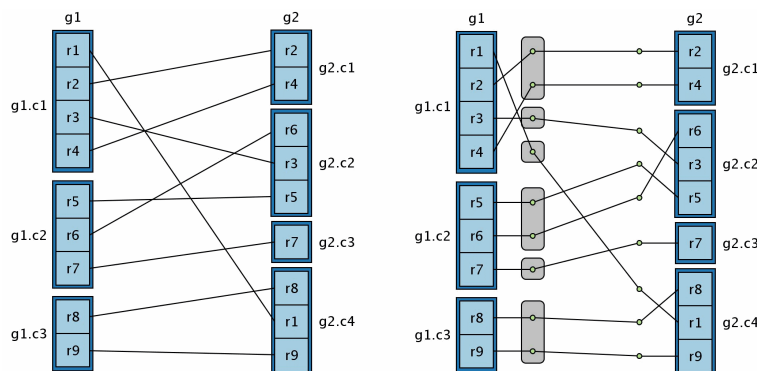


Abbildung 4.7.: Minimierung der Linienkreuzungen in Matchmaker – Links: Naiver Ansatz mit direkten Verbindungen. Rechts: Durch die Einführung von Support-Punkten (grün), werden die Linienkreuzungen in den Bereich zwischen den Clustern (blau) und den zugehörigen Support-Punkten verlagert. [LSP⁺ 10]

Matchmaker verfügt über einen Übersichts- und einen Detail-Modus. Der Übersichts-Modus zeigt mehrere, der Detail-Modus immer genau zwei Gruppen. Abbildung 4.8 zeigt den Übersichts-Modus bei der Visualisierung eines medizinischen Gen-Datensatzes. Im Detail-Modus können einzelne Cluster durch Selektion vergrößert und detailreicher dargestellt werden. In beiden Modi besteht die Möglichkeit einzelne Elemente oder Linien, sowie ganze Cluster zu selektieren, um so zugehörige Elemente in den anderen Gruppen hervorzuheben.

4. Verwandte Visualisierungskonzepte

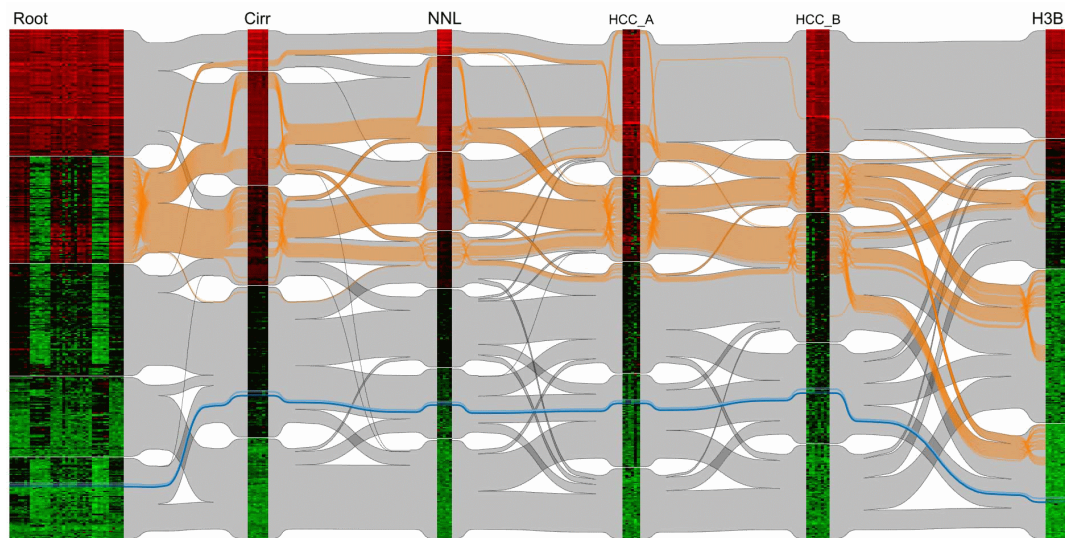


Abbildung 4.8.: Matchmaker im Übersichts-Modus – Dargestellt ist ein medizinischer Gen-Datensatz. Die Root-Gruppe beinhaltet alle, die anderen Gruppen jeweils nur einzelne Dimensionen. Das zweite Cluster der Root-Gruppe ist selektiert, was dazu führt, dass alle Elemente des Clusters in der ganzen Ansicht hervorgehoben werden (orange). [LSP⁺ 10]

Bewertung

Besonders hervorzuheben ist bei Matchmaker die Strategie zur Minimierung der Linienkreuzungen zwischen benachbarten Gruppen. Das Problem wird nicht durch komplexe mathematische Verfahren gelöst, sondern durch einen relativ einfachen, visuellen Ansatz. Durch die Einführung von Support-Punkten werden die Kreuzungen auf einen schmalen Streifen der Anzeige verdrängt, sodass zwischen Clustern gut wahrnehmbare Bündel verlaufen und die Entwicklung der Cluster einfach analysiert werden kann. Ein Problem dieser Technik ist allerdings, dass im Allgemeinen nicht mehr ersichtlich ist, welche beiden Elemente innerhalb von Clustern durch Linien verbunden sind. Erst durch die Selektion eines Elements in einem Cluster kann herausgefunden werden, welches das zugehörige Element in der benachbarten Gruppe ist.

Matchmaker dient der Visualisierung mehrdimensionaler, quantitativer Daten und kann somit nicht direkt bei der Analyse von Gruppen in dynamischen Graphen eingesetzt werden. Das grundlegende visuelle Konzept, die Interaktionsmöglichkeiten und die Technik zur Minimierung der Kreuzungen sind allerdings übertragbar.

4.3.3. Alluvial Diagrams

Rosvall und Bergstrom [RB10] befassen sich mit dem Erkennen und Visualisieren von Veränderungen in dynamischen Netzwerken. Ihr Vorgehen besteht aus zwei Teilen: dem *Significance Clustering* und der Visualisierung durch *Alluvial Diagrams*.

Durch das Significance Clustering können wirkliche Änderungen in der Netzwerkstruktur von zufälligen Schwankungen unterschieden werden. Beim Significance Clustering wird jeder Graph G_t des dynamischen Netzwerks zunächst mit einem herkömmlichen Clustering-Algorithmus partitioniert. Daraufhin wird aus jedem Original-Graphen G_t eine Menge von sog. Bootstrap-Graphen $\{B_{t1}, \dots, B_{tm}\}$ generiert. Einen Bootstrap-Graphen erhält man, indem die Kantengewichte des Original-Graphen entsprechend einer Poisson-Verteilung neu abgetastet werden. Ein Bootstrap-Graph entspricht also einer zufällig leicht „gestörten“ Variante des Original-Graphen. Jeder Bootstrap-Graph B_{ti} wird ebenfalls geclustert. Auf Basis dieser Bootstrap-Graphen können signifikante Gruppen des Original-Graphen G_t identifiziert werden, indem Teilmengen der Knoten gesucht werden, welche in nahezu allen Bootstrap-Netzwerken der gleichen Gruppe zugeordnet werden. Das Significance Clustering wird für jeden Graphen G_t des Eingabe-Netzwerks wiederholt.

Abbildung 4.9 (oben) zeigt die Graphen zu zwei Zeitpunkten eines dynamischen Netzwerkes. Jeder Graph durchläuft zunächst einzeln das Significance Clustering. Das Ergebnis ist eine Partition der Knoten, welche in Abbildung 4.9 (mitte) farblich dargestellt ist. Der Farbton deutet die Gruppenzugehörigkeit eines Knotens an und die Helligkeit seine Signifikanz. Je höher die Signifikanz, also je häufiger ein Knoten in den Bootstrap-Graphen derselben Gruppe zugeordnet wurde, desto dunkler seine Farbe.

Das Ergebnis des Significance Clusterings wird als Alluvial Diagram visualisiert (siehe Abb. 4.9 unten). Jeder Zeitpunkt des Eingabe-Netzwerks wird durch eine Spalte im Diagramm repräsentiert (in Abb. 4.9 beschriftet mit „Time 1“ und „Time 2“). Die Gruppen der Zeitpunkte werden auf Blöcke in den Spalten abgebildet. Die Größe einer Gruppe (definiert z. B. durch den *PageRank* Gruppe oder schlicht die Anzahl der Knoten) wird auf die Höhe des Blocks abgebildet. Die Gruppen sind vertikal sortiert, wobei die größte Gruppe ganz unten angeordnet ist. Um die Veränderungen in der Gruppenstruktur zwischen zwei Zeitpunkten sichtbar zu machen, sind die Gruppen mit Bändern verbunden. Ein Band repräsentiert eine Menge von Knoten, welche gemeinsam von einer Gruppe der Zeit t zu einer anderen der Zeit $t + 1$ übergehen. Die geschwungene Form der Bänder wird durch die Verwendung von Spline-Kurven erreicht und erleichtert das Verfolgen der Gruppenentwicklung. Abbildung 4.9 (unten) zeigt deutlich, wie sich die orangefarbene und die rote Gruppe im zweiten Zeitpunkt vereinigen. Von der blauen Gruppe spaltet sich eine lila Gruppe ab.

Edler und Rosvall haben das Konzept der Alluvial Diagrams in einer Visualisierungsanwendung umgesetzt [ER10]. In der Visualisierung können einzelne Knoten oder ganze Gruppen selektiert werden, welche dann farblich global hervorgehoben werden. Auf diese Weise kann die Entwicklung einzelner Elemente oder Gruppen genauer nachvollzogen werden.

4. Verwandte Visualisierungskonzepte

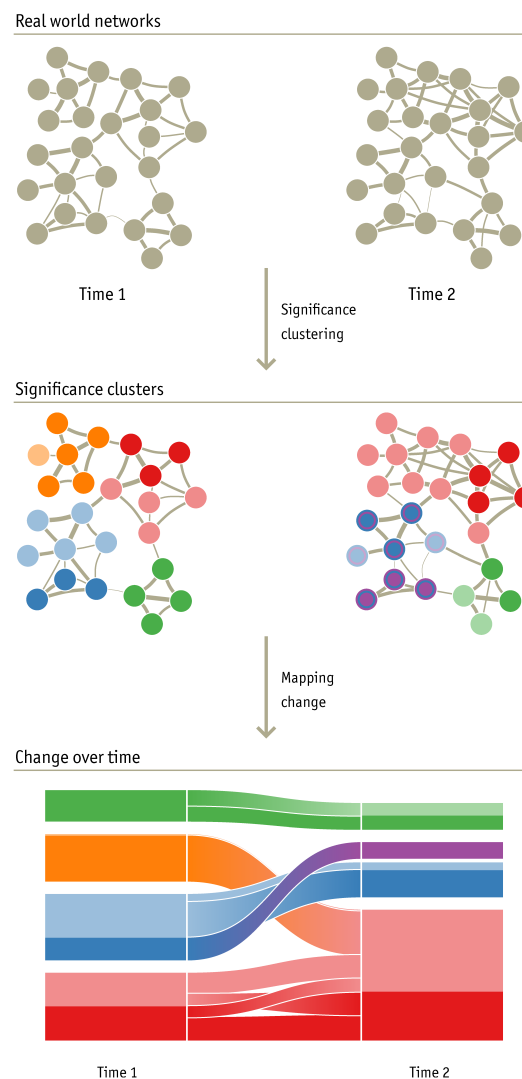


Abbildung 4.9.: Significance Clustering und Alluvial Diagram [RB10]

Bewertung

Mithilfe der Alluvial Diagrams wird die Evolution von Gruppen in einer sehr schlichten, reduzierten Weise dargestellt. Der Fokus wird so auf das Wesentliche gelenkt. Wie bei Ogawa et al. [OMB⁺07] werden auch hier die Gruppen der einzelnen Zeitpunkte nach ihrer Größe vertikal angeordnet, was in Bezug auf mögliche Bänder-Kreuzungen nicht optimal ist. In der Visualisierung von Edler und Rosvall können die Kreuzungen manuell minimiert werden, indem die Gruppen interaktiv verschoben werden können. Bei größeren Datensätzen mit vielen Verbindungen zwischen den Zeitpunkten kann dies allerdings sehr aufwendig werden. Eine automatische Anordnung ist hier vorzuziehen.

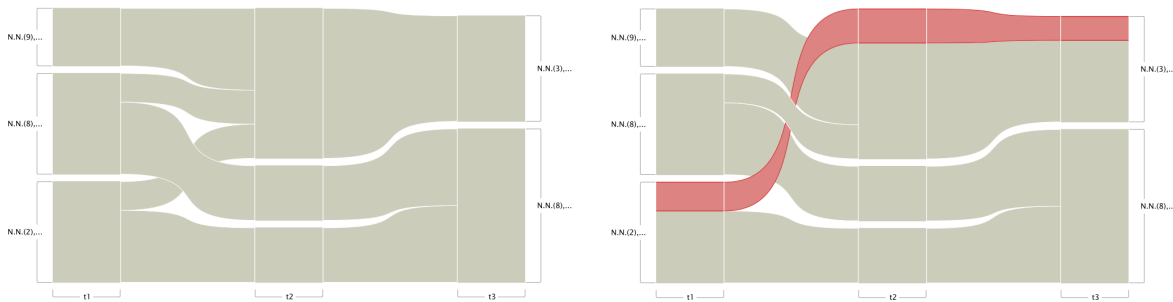


Abbildung 4.10.: Selektion eines Knotens in der Visualisierung von Edler und Rosvall – Links: Diagramm vor der Selektion. Rechts: Diagramm nach der Selektion eines Knotens. Die Kanten zwischen Zeitpunkt „t1“ und „t2“ verlaufen nun leicht anders. [ER10]

Wie erwähnt bietet die Visualisierung von Edler und Rosvall die Möglichkeit mit den Diagrammen zu interagieren. Hierbei tritt allerdings ein Problem zu Tage: Wird ein Knoten oder eine Gruppe ausgewählt, so muss oft die Ansicht „umgebaut“ werden, um den Pfad zusammenhängend darzustellen (siehe Abb. 4.10). Dies kann das mentale Bild des Betrachters zerstören und so die Analyse erschweren.

Die Visualisierung bietet neben dem Alluvial Diagramm noch eine weitere Ansicht, in welcher die Struktur des Netzwerks zu jeweils einem Zeitpunkt als Knoten-Kanten-Diagramm angezeigt werden kann. Die Struktur kann dabei sowohl auf Netzwerk-, als auch auf Gruppen-Ebene (hier *module level* genannt) dargestellt werden. Auf diese Weise kann allerdings nicht die Struktur zu zwei Zeitpunkten miteinander verglichen werden.

4.3.4. Reda et al.

Reda et al. [RTJ⁺11] stellen eine Visualisierung der Gruppenevolution in sozialen Netzwerken vor. Das visuelle Konzept orientiert sich dabei an den *Parallelen Koordinaten* [ID90]. Als Eingabedaten für die Visualisierung dienen die Gruppenstrukturen in den einzelnen Zeitschritten eines Netzwerks. Diese Gruppenstrukturen werden zunächst, durch das Framework von Tantipathananandh und Berger-Wolf [TBW09], über die Zeit hinweg „geglättet“ (ähnlich dem evolutionären Clustering aus Abschnitt 3.2.2). Auf diese Weise wird versucht einzelne, kurzlebige Interaktionen herauszufiltern und eine stabilere, globale Gruppenstruktur zu gewinnen (als *community interpretation* bezeichnet), welche letztendlich visualisiert wird.

Abbildung 4.11 zeigt die Visualisierung eines Datensatzes zum Abstimmungsverhalten im US-Repräsentantenhaus. Entlang der x-Achse verläuft die Zeit. Entlang der y-Achse sind die zuvor identifizierten Gruppen angeordnet (mit A-H beschriftet). Ein Linienzug (Pfad) innerhalb der Darstellung repräsentiert ein Element des Netzwerks und zeigt dessen Gruppenzugehörigkeit über die Zeit. Wechselt ein Element die Gruppe, so ist dies als Liniestück mit steilem Winkel erkennbar. Die Linien werden innerhalb einer Gruppe dicht

4. Verwandte Visualisierungskonzepte

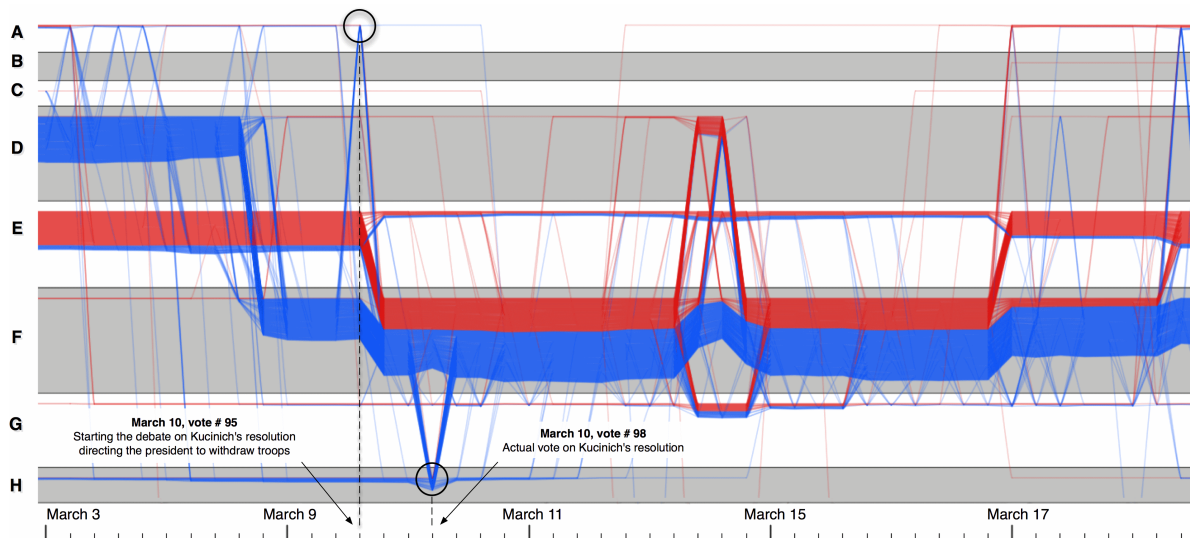


Abbildung 4.11.: Visualisierung des Abstimmungsverhaltens im US-Repräsentantenhaus – Blaue Pfade repräsentieren demokratische und rote Pfade republikanische Abgeordnete. [RT]⁺¹¹

beieinander als Bündel dargestellt. Einzelne Ereignisse, wie das Aufspalten einer Gruppe oder das Abwandern von Elementen zu anderen Gruppen sind auf diese Weise gut erkennbar. Durch Selektion einzelner Pfade, werden diese hervorgehoben, indem alle übrigen Pfade nahezu transparent dargestellt werden. Somit kann die Gruppenzugehörigkeit einzelner Elemente besser nachverfolgt werden.

Bewertung

Die Verwendung der Parallelen-Koordinaten-Metapher macht die Darstellung intuitiv verständlich. Zudem ist das Konzept sehr platzsparend, da einzelne Elemente nur durch Linien repräsentiert werden. Somit eignet sich dieser Ansatz auch zur Darstellung größerer Datensätze. Wie bei anderen untersuchten Arbeiten kann es auch hier zum Problem der Kreuzung von Linien kommen. Der Layout-Algorithmus bietet hierfür keine automatische Optimierung. Stattdessen besteht die Möglichkeit die Gruppen (A-H in Abb. 4.11) interaktiv in ihrer vertikalen Position zu verändern und so die Kreuzungen manuell zu minimieren. Ein derartiges Vorgehen bietet sich aber nur lokal und bei kleineren Datensätzen an.

4.4. Fazit

Um die in diesem Kapitel untersuchten Visualisierungskonzepte besser bewerten und vergleichen zu können, wurde eine Reihe von Anforderungen an eine Visualisierung für Gruppen in dynamischen Netzwerken identifiziert. Zu den Anforderungen gehören im Einzelnen:

- A1 - Die Entwicklung der Gruppen ist nachvollziehbar:** Ermöglicht es die Visualisierung, die Entwicklung von Gruppen mit geringer kognitiver Anstrengung nachvollziehen zu können?
- A2 - Dominante Gruppen können identifiziert werden:** Bei der Analyse von dynamischen Netzwerken ist es ein Ziel, dominante Gruppen, welche über die Zeit hinweg bestehen, von kurzlebigen Gruppen zu unterscheiden. Aus diesem Grund sollte die Visualisierung es ermöglichen, dominante Gruppen hervorzuheben.
- A3 - Die Gruppenzugehörigkeit Einzelner ist verfolgbar:** Kann die Gruppenzugehörigkeit einzelner Elemente über die Zeit hinweg mit geringer kognitiver Anstrengung verfolgt werden?
- A4 - Ausreißer-Elemente sind identifizierbar:** Können konstante Elemente von Elementen, welche häufig ihre Gruppenzugehörigkeit ändern („Ausreißer“), unterschieden werden?
- A5 - Die Netzwerkstruktur ist sichtbar:** Kann die Struktur des Netzwerks in den einzelnen Zeitschritten z. B. als Knoten-Kanten-Diagramm angezeigt werden? Nur so können Zusammenhänge zwischen der Entwicklung der Gruppen und der Netzwerkstruktur hergestellt werden.
- A6 - Die Linien-Kreuzungen zwischen Zeitschritten werden minimiert:** Diese Anforderung betrifft nur die in Abschnitt 4.3 vorgestellten Visualisierungen. Bei diesen Visualisierungen sind die Gruppen der einzelnen Zeitschritte durch Linien miteinander verbunden. Ein automatisches Verfahren sollte dafür sorgen, dass die Anzahl der Linien-Kreuzungen minimiert wird. Nur so kann die Entwicklung der Gruppen gut nachvollzogen werden.

Tabelle 4.1 zeigt, inwiefern die untersuchten Konzepte die gestellten Anforderungen erfüllen. So bietet z. B. keines der Konzepte eine Möglichkeit, dominante Gruppen einfach zu identifizieren (A2). Ebenso können auch Ausreißer-Elemente in keiner Visualisierung gesondert hervorgehoben werden (A4). Die Netzwerkstruktur in den einzelnen Zeitschritten ist nur bei den Visualisierungen für dynamische Graphen (Abschnitt 4.1) sichtbar; diese Visualisierungen bieten sich allerdings nicht zur Analyse der Gruppenentwicklung an (vgl. A1). Ein großes Problem der Visualisierungen der Evolution von Gruppen (Abschnitt 4.3) ist, dass es keinen ausgereiften Ansatz gibt, die Linien-Kreuzungen zwischen Zeitschritten zu minimieren.

Da die untersuchten Konzepte die gestellten Anforderungen nur unzureichend erfüllen, soll im Rahmen dieser Arbeit ein neues Konzept zur Visualisierung der Gruppen-Evolution in dynamischen Netzwerken entwickelt werden.

	Animation	Small Multiples	Layers	Ogawa et al.	Matchmaker	Alluvial Diagrams	Reda et al.	Diese Arbeit
A1: Die Entwicklung der Gruppen ist nachvollziehbar	●	●	●	●	●	●	●	●
A2: Dominante Gruppen können identifiziert werden	●	●	●	●	●	●	●	●
A3: Die Gruppenzugehörigkeit Einzelner ist verfolgbar	●	●	●	●	●	●	●	●
A4: Ausreißer können identifiziert werden	●	●	●	●	●	●	●	●
A5: Die Netzwerkstruktur ist sichtbar	●	●	●	○	○	○	○	●
A6: Die Linien-Kreuzungen zwischen Zeitschritten werden minimiert	-	-	-	○	●	○	○	●

Tabelle 4.1.: Übersicht der vorgestellten Visualisierungskonzepte
 (●: Anforderung erfüllt ●: Anforderung teilweise erfüllt ○: Anforderung nicht erfüllt)

5. Konzept

Auf Grundlage der Erkenntnisse aus Kapitel 4 wurde ein Konzept für die zu entwickelnde Visualisierung erarbeitet. Das Konzept orientiert sich an den Visualisierungen aus Abschnitt 4.3, insbesondere an den Alluvial Diagrams (Abschnitt 4.3.3). Im Vergleich zu den Alluvial Diagrams wird auch die Struktur des Netzwerks in jedem Zeitschritt als Knoten-Kanten-Diagramm, wie beim Small-Multiples-Ansatz (Abschnitt 4.2.2), dargestellt. Ziel des erarbeiteten Konzepts ist es, die Darstellung der Gruppenentwicklung und der Netzwerkstruktur in eine statische Ansicht zu integrieren. Das Konzept besteht aus zwei verschiedenen Ansichten: Der *Detail-Ansicht* und der *aggregierten Ansicht*.

5.1. Detail-Ansicht

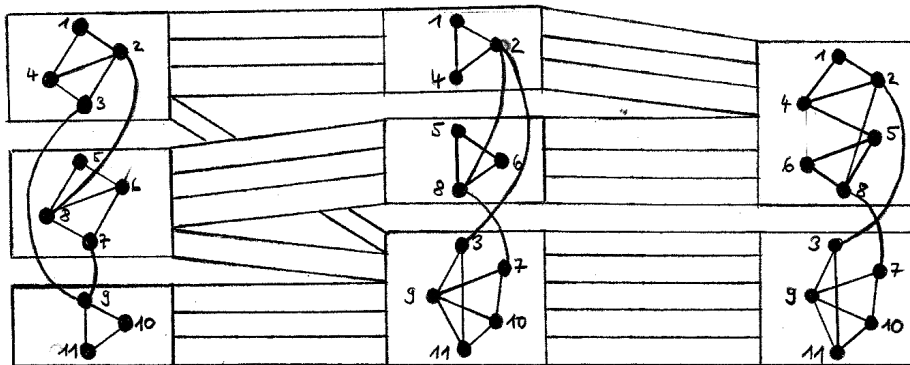


Abbildung 5.1.: Entwurf der Detail-Ansicht

Abbildung 5.1 zeigt den Entwurf der Detail-Ansicht. Die Ansicht visualisiert den dynamischen Graphen aus Abbildung 2.2 (Seite 14). Jeder Zeitschritt-Graph G_t wird durch eine Spalte repräsentiert. Je Spalte werden die durch einen Clustering-Algorithmus identifizierten Gruppen als Rechtecke dargestellt. Innerhalb der Rechtecke sind die Knoten der Gruppe vertikal angeordnet, d. h. jeder Knoten besitzt eine eindeutige vertikale Position. Die Kanten zwischen den Knoten derselben Gruppe werden als gerade Linien gezeichnet (*Intra-Cluster-Kanten*), Kanten zwischen den Knoten verschiedener Gruppen als gebogene Linien (*Inter-Cluster-Kanten*). Gleiche Knoten zweier benachbarter Graphen G_t und G_{t+1} sind durch Linien miteinander verbunden, d. h. $v_i \in V_t$ wird mit $v_i \in V_{t+1}$ verbunden. Diese Linien zwischen Zeitschritten werden im

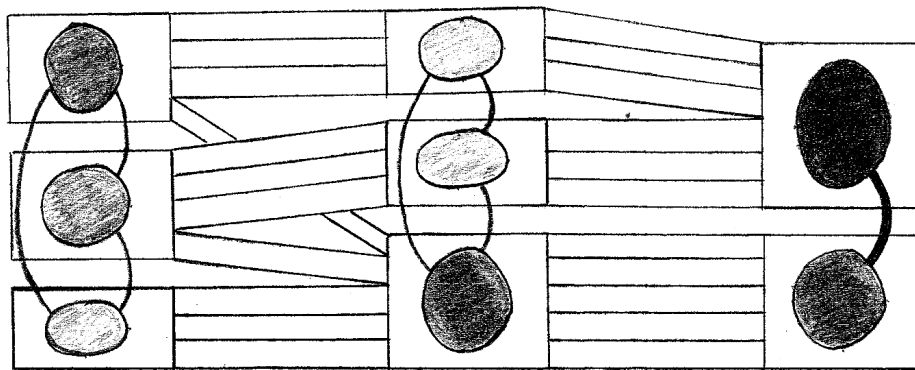


Abbildung 5.2.: Entwurf der aggregierten Ansicht

Folgenden als *Transitionen* bezeichnet. So lässt sich genau erkennen, in welcher Gruppe sich ein Knoten im nächsten Zeitschritt befindet. Auch mehrere Knoten, welche gemeinsam von einer Gruppe zu einer anderen übergehen, können identifiziert werden.

Die Darstellung ermöglicht es, die Ereignisse bei der Entwicklung von Gruppen nachzuvollziehen. Zu den Ereignissen gehören: Wachstum, Verkleinerung, Vereinigung, Aufteilung, Entstehung und Auflösung (vgl. Abschnitt 2.4). Abbildung 5.1 zeigt, wie die beiden oberen Gruppen vom ersten zum zweiten Zeitschritt kleiner werden, da jeweils ein Knoten zur untersten Gruppe wechselt. Vom zweiten zum dritten Zeitschritt vereinigen sich die oberen beiden Gruppen, während die unterste Gruppe unverändert bleibt.

5.2. Aggregierte Ansicht

Der Entwurf der aggregierten Ansicht ist in Abbildung 5.2 dargestellt. Im Gegensatz zur Detail-Ansicht, stellt die aggregierte Ansicht den Graphen G_t nicht mehr auf Ebene der Knoten, sondern auf Ebene der Gruppen dar. Die Gruppen werden als Ellipsen innerhalb der Rechtecke gezeichnet. Die Füllfarbe der Ellipsen wird entsprechend einer Metrik für Gruppen in Graphen bestimmt. So kann die Füllfarbe z. B. andeuten, wie dicht die Knoten der Gruppe untereinander verbunden sind. Alle Inter-Cluster-Kanten zwischen zwei Gruppen werden zu einer aggregierten Kante zusammengefasst. Das Gewicht der aggregierten Kante ist die Anzahl bzw. das summierte Gewicht der Einzelkanten und wird auf die Dicke der Linie abgebildet.

Die aggregierte Ansicht ermöglicht es, die Entwicklung der Gruppenstruktur zu verfolgen und zeigt darüber hinaus zu jedem Zeitschritt auch die Stärke der Verbindungen der Gruppen untereinander. Sind zwei Gruppen, bspw. in einem sozialen Netzwerk, stark miteinander verbunden, so ist anzunehmen, dass sich die Gruppen zu einem späteren Zeitpunkt evtl. vereinigen, oder zumindest einzelne Elemente zwischen den Gruppen überwechseln. Derartige Hypothesen können mithilfe der aggregierten Ansicht überprüft werden. Wie

erwähnt, kann die Füllfarbe der Ellipsen z. B. den „inneren Zusammenhalt“ einer Gruppe andeuten. Diese Darstellung kann ebenfalls genutzt werden, um Zusammenhänge zwischen der internen Struktur einer Gruppe und ihrer Entwicklung herzustellen. Es ist anzunehmen, dass Gruppen, deren Knoten untereinander nur schwach verbunden sind, ein höheres Risiko zur Aufteilung oder Auflösung haben, als dicht verbundene Gruppen. Die aggregierte Ansicht soll als Einstiegspunkt der Analyse dienen und einen Überblick über das dynamische Netzwerk bieten. Anschließend kann auf die Detail-Ansicht gewechselt werden, um so die interne Struktur der Gruppen miteinbeziehen zu können.

5.3. Weitere Bestandteile des Konzepts

5.3.1. Interaktion

In beiden Ansichten soll die Darstellung vergrößert werden können, um so Einzelheiten besser zu erkennen. Vor allem in der Detail-Ansicht ist dies zur Analyse der internen Struktur einer Gruppe wichtig. In der vergrößerten Darstellung soll der angezeigte Bildausschnitt zudem möglichst intuitiv verschoben werden können. Einzelne Knoten sollen selektiert werden können, um deren Gruppenzugehörigkeit im Verlauf der Zeit hervorzuheben. Des Weiteren sollen auch einzelne Transitionen selektiert werden können, was denselben Effekt hat. Zudem sollen Gruppen selektiert werden können, wodurch die zugehörigen Knoten in allen Zeitschritten hervorgehoben werden.

5.3.2. Hervorhebungen

Dominante Gruppen, welche über die Zeit hinweg bestehen, sollen in beiden Ansichten hervorgehoben werden können. Um diese dominanten Gruppen zu identifizieren soll das in Abschnitt 3.3 vorgestellte Verfahren zur Gruppenverfolgung von Greene et al. [GDC10] implementiert werden. Das Ergebnis des Verfahrens ist eine Menge von dynamischen Gruppen, welche entsprechend ihrer Signifikanz geordnet werden können. Die signifikantesten dynamischen Gruppen sollen in den Ansichten eingefärbt werden.

Ebenso sollen Knoten, welche oft ihre Gruppenzugehörigkeit ändern („Ausreißer“), farblich hervorgehoben werden können, um diese von „konstanten“ Knoten zu unterscheiden. Konstante Knoten verbleiben über die Zeit in vorwiegend einer (dynamischen) Gruppe. Zunächst muss eine Metrik definiert werden, welche ein Maß für die Stabilität eines Knotens im Bezug auf seine Gruppenzugehörigkeit ist. Mithilfe dieser Metrik kann für jeden Knoten ein Stabilitätswert berechnet werden. Der Wert soll anschließend, basierend auf einer Farbskala, in den Ansichten visualisiert werden.

5.3.3. Minimierung der Kreuzungen von Transitionen

Die vertikale Anordnung der Gruppen, sowie der Knoten innerhalb der Gruppen, soll so gewählt werden, dass die Anzahl der Kreuzungen von Transitionen minimiert wird. Wie bereits in Abschnitt 4.3 erwähnt, erschweren viele Linien-Kreuzungen zwischen benachbarten Zeitschritten das Nachvollziehen der Gruppenentwicklungen. Im Rahmen der Umsetzung ist ein geeignetes Verfahren zu implementieren.

6. Umsetzung

Basierend auf dem in Kapitel 5 vorgestellten Konzept wurde eine Visualisierungsanwendung implementiert. Das Ergebnis der Umsetzung wird im Folgenden vorgestellt. Die Anwendung wurde in der Programmiersprache *Java* (Version 6.0) entwickelt. Java bietet mit *Java 2D* eine Programmierschnittstelle zur Erstellung zwei-dimensionaler Grafikanwendungen. So stellt Java 2D unter anderem Klassen und Methoden zum Zeichnen geometrischer Primitive, zum Compositing und zur Farbverwaltung bereit. Die Klasse *java.awt.Graphics2D* dient als Renderer und erzeugt die letztendliche Ausgabe für den Bildschirm. Java bietet zudem den Vorteil, dass damit erstellte Programme plattformunabhängig sind. Neben den Standard-Java-Bibliotheken wurden die Datenstrukturen für Graphen aus der *JUNG*-Bibliothek¹ (*Java Universal Network/Graph Framework*, Version 2.0) verwendet.

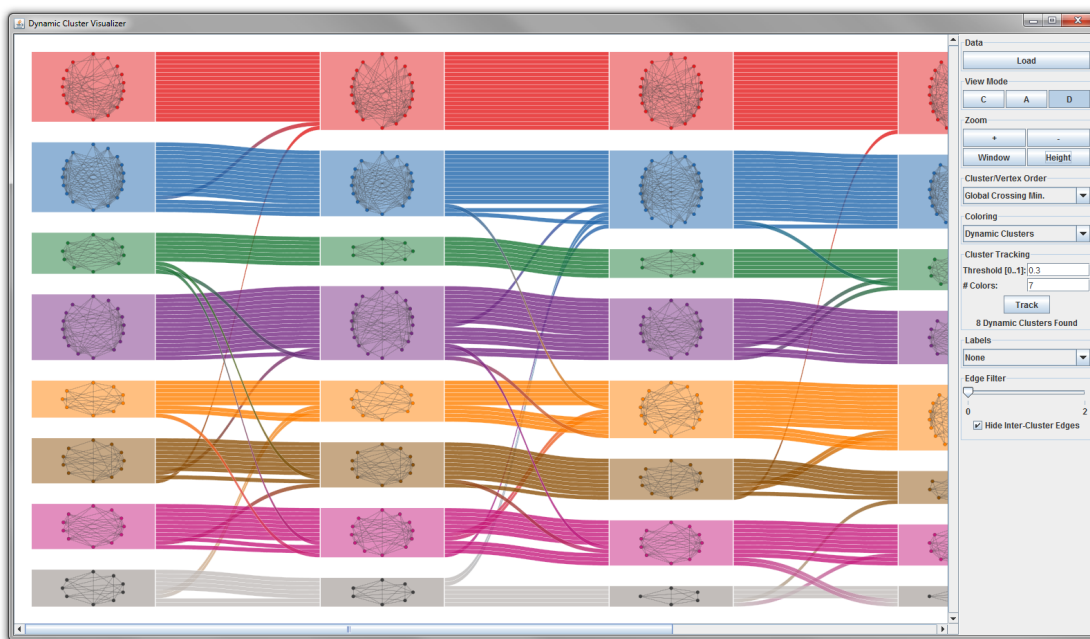


Abbildung 6.1.: Übersicht der entwickelten Anwendung – Im zentralen Bereich wird die Visualisierung dargestellt, am rechten Rand befindet sich eine Menüleiste zur Interaktion.

¹<http://jung.sourceforge.net/>

Hinweis zu den Abbildungen in diesem Kapitel:

Die Abbildungen in diesem Kapitel zeigen die Visualisierung von synthetischen Datensätzen. Die Datensätze wurden mit dem Tool von Green et al. [GDC10] generiert, welches online verfügbar ist². Mit dem Tool kann ein ungerichteter, ungewichteter dynamischer Graph mit eingebetteter Gruppenstruktur erzeugt werden. Die Anzahl der Zeitschritte des dynamischen Graphen, sowie die Eigenschaften der Zeitschritt-Graphen G_t (Anzahl der Knoten, durchschnittlicher Knotengrad, maximale Gruppengröße) können als Parameter übergeben werden. Zudem können Ereignisse in der Gruppenentwicklung spezifiziert werden (z. B. das Aufspalten und Vereinigen von Gruppen).

Für die Abbildungen wurden die beiden folgenden Datensätze verwendet:

Switch 100 Dynamischer Graph, welcher 5 Zeitschritte umfasst. Jeder Zeitschritt-Graph G_t enthält jeweils 100 Knoten in 8 Gruppen. Von G_t auf G_{t+1} wechseln jeweils 10% der Knoten ihre Gruppe.

Merge & Split 100 Dynamischer Graph, welcher 5 Zeitschritte umfasst. Jeder Zeitschritt-Graph G_t enthält jeweils 100 Knoten in 8 Gruppen. Von G_t auf G_{t+1} findet jeweils eine Vereinigung zweier Gruppen und eine Aufspaltung einer Gruppe statt. Zudem wechseln ebenfalls 10% der Knoten ihre Gruppe.

²<http://mlg.ucd.ie/snam>

6.1. Eingabedaten

Um einen Datensatz mit der entwickelten Anwendung zu visualisieren, muss dieser zunächst geladen werden. Als Eingabe erhält die Anwendung den dynamischen Graphen $G = \{G_1, \dots, G_n\}$, welcher das dynamische Netzwerk repräsentiert. Die einzelnen Graphen G_t können ungewichtet oder gewichtet sein; die Anwendung unterstützt jedoch keine gerichteten Graphen. Da kein Clustering-Algorithmus in der Anwendung selbst implementiert wurde, müssen auch die Gruppenstrukturen $P = \{P_1, \dots, P_n\}$ als Eingabe übergeben werden. Als Dateiformate, sowohl für die Graphen, als auch für die Gruppenstrukturen, dienen die Formate der weit verbreiteten Anwendung *Pajek*³ [BM98]. Für jeden Zeitschritt müssen zwei Dateien übergeben werden: Die *.net*-Datei und die *.clu*-Datei. Die *.net*-Datei definiert einen statischen Graphen über eine Knoten- und Kantenliste (siehe Listing 6.1), die *.clu*-Datei partitioniert die Knoten der *.net*-Datei in Gruppen (siehe Listing 6.2). Die Dateien können in der Anwendung über einen Dialog ausgewählt und geladen werden.

Um die Gruppenstrukturen für einen dynamischen Graphen zu berechnen kann z. B. der Infomap-Algorithmus von Rosvall und Bergstrom [RBo8] verwendet werden. Der Infomap-Algorithmus ist ein statischer Clustering-Algorithmus (siehe Abschnitt 3.1). Eine Implementierung in Form eines Kommandozeilen-Tools ist online verfügbar⁴. Das Tool erwartet als Eingabe eine Pajek *.net*-Datei und liefert als Ausgabe eine *.clu*-Datei (siehe Abb. 6.2). Auf diese Weise kann für jeden Graphen G_t des dynamischen Graphen eine Gruppenstruktur P_t bestimmt werden. Dieses Vorgehen entspricht dem lokalen Clustering (siehe Abschnitt 3.2.1). Ursprünglich war geplant einen evolutionären Clustering-Algorithmus (siehe Abschnitt 3.2.2) in die Anwendung zu integrieren, allerdings konnte dies aus Zeitgründen nicht umgesetzt werden. Daher wurde auf ein externes Tool zurückgegriffen.

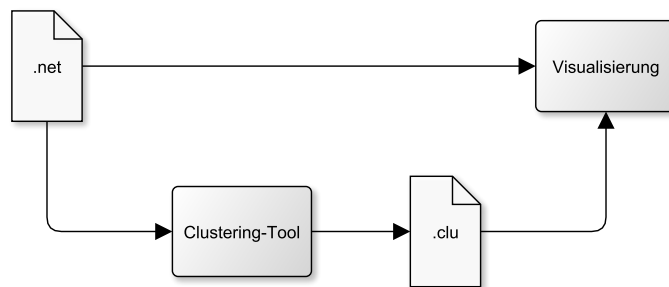


Abbildung 6.2.: Eingabedaten der Visualisierungsanwendung – Für jeden zu visualisierenden Zeitschritt muss der Anwendung eine *.net*-Datei für den Graphen und eine *.clu*-Datei für dessen Gruppenstruktur übergeben werden. Die *.clu*-Datei wird durch die Verwendung eines externen Clustering-Tools gewonnen.

³<http://pajek.imfm.si>

⁴<http://www.tp.umu.se/~rosvall/code.html>

6. Umsetzung

Listing 6.1 Beispiel einer Pajek .net-Datei zur Definition eines ungerichteten, gewichteten Graphen. Die Knoten (engl. *vertices*) werden in der Form $\langle \text{Knoten-ID} \rangle \langle \text{Knotenname} \rangle$ angegeben, die Kanten (engl. *edges*) in der Form $\langle \text{Knoten-ID } 1 \rangle \langle \text{Knoten-ID } 2 \rangle \langle \text{Gewicht} \rangle$.

```
*Vertices 5
1 "Knoten 1"
2 "Knoten 2"
3 "Knoten 3"
4 "Knoten 4"
5 "Knoten 5"
*Edges 4
1 2 1.0
2 3 2.0
3 1 3.0
4 5 1.0
```

Listing 6.2 Beispiel einer Pajek .clu-Datei (passend zu Listing 6.1). In jeder Zeile steht die einem Knoten aus der .net-Datei zugeordnete Partition. Die Zeilennummer - 1 entspricht der Knoten-ID, also z. B. *Knoten 4* \rightarrow *Partition 2*.

```
*Vertices 5
1
1
1
2
2
```

6.2. Ansichten

In der Visualisierung wurden drei Ansichten implementiert. Neben der im Konzept erwähnten Detail-Ansicht und der aggregierten Ansicht wurde zusätzlich die *Cluster-Ansicht* erstellt.

6.2.1. Cluster-Ansicht

Als erstes wurde die Cluster-Ansicht als Basis für die anderen beiden Ansichten implementiert (siehe Abb. 6.3). In der Cluster-Ansicht wird, wie bei den Alluvial Diagrams (Abschnitt 4.3.3), die Entwicklung der Gruppen über die Zeit visualisiert. Die Netzwerkstruktur in den einzelnen Zeitschritten ist nicht sichtbar. Die Gruppen in jedem Zeitschritt werden durch Rechtecke repräsentiert, wobei die Größe der Gruppe (Anzahl der Knoten) auf die Höhe des Rechtecks abgebildet wird. Die Cluster-Ansicht soll als Einstiegspunkt der Analyse dienen, um zunächst die Veränderung der Gruppenstruktur studieren zu können. Anschließend kann in die anderen beiden Ansichten gewechselt werden, um so schrittweise die Detailstufe (engl. *Level of Detail*) zu erhöhen und auch die Struktur des Netzwerks in den einzelnen Zeitschritten in die Analyse mit einzubeziehen. Ein typischer Arbeitsablauf bei der Untersuchung eines Datensatzes ist folgender: Cluster-Ansicht \rightarrow Aggregierte Ansicht \rightarrow Detail-Ansicht.



Abbildung 6.3.: Cluster-Ansicht des Datensatzes Switch 100

Im Vergleich zum Konzept wurden für die Transitionen keine einfachen Linien verwendet, sondern Bézierkurven (siehe Abb. 6.4). Durch deren geschwungene Form werden spitze Winkel vermieden, wodurch das visuelle Nachverfolgen der Entwicklung unterstützt wird. Zudem werden die Transitionen semi-transparent gezeichnet, um diese auch bei Kreuzungen weiterverfolgen zu können.



Abbildung 6.4.: Vergleich der Darstellungen von Transitionen – Links: Gerade Linien als Transitionen. Rechts: Bézierkurven als Transitionen.

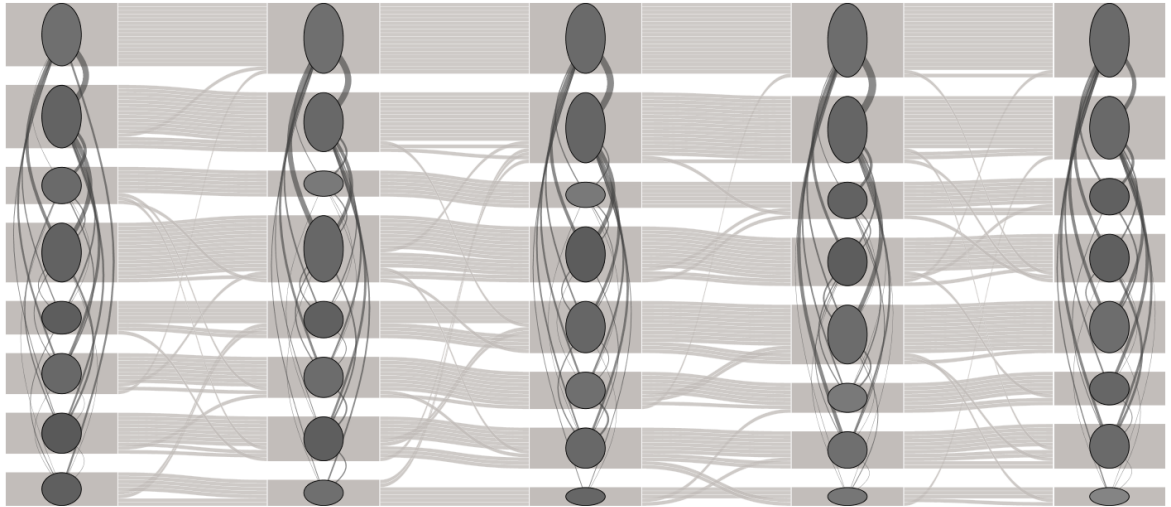


Abbildung 6.5.: Aggregierte Ansicht des Datensatzes Switch 100

6.2.2. Aggregierte Ansicht

Wie im Konzept beschrieben zeigt die aggregierte Ansicht, neben der Entwicklung der Gruppenstruktur, die Verbindungen zwischen Gruppen eines Zeitschritts (siehe Abb. 6.5). Die Gruppen werden als Ellipsen innerhalb der Rechtecke dargestellt. Die *relative Dichte* (engl. *relative density* [For10, S. 11]) einer Gruppe wird auf die Farbe der Ellipse abgebildet. Je höher die relative Dichte, desto gesättigter die Farbe der Ellipse. Im Folgenden wird eine formale Definition der relativen Dichte gegeben [For10]:

Sei C eine Menge von Knoten innerhalb des Graphen G . Der *interne* Grad eines Knotens $v \in C$, k_v^{int} , sei definiert als die Anzahl von Kanten, welche v mit anderen Knoten aus C verbinden. Der *externe* Grad k_v^{ext} wird analog dazu definiert als die Anzahl von Kanten, welche v mit den restlichen Knoten von G verbinden. Ist G ein gewichteter Graph wird anstatt der Anzahl der Kanten deren summiertes Gewicht verwendet. Der interne Grad von C wird definiert als $k_C^{int} = \sum_{v \in C} k_v^{int}$, der externe Grad als $k_C^{ext} = \sum_{v \in C} k_v^{ext}$. Die relative Dichte $\rho(C)$ wird dann definiert als:

$$\rho(C) = \frac{k_C^{int}}{k_C^{int} + k_C^{ext}}$$

Sind die Knoten in C nur untereinander verbunden ($k_C^{ext} = 0$), so ist $\rho(C) = 1$. Gibt es dagegen keine Verbindungen zwischen den Knoten aus C ($k_C^{int} = 0$), so ergibt sich $\rho(C) = 0$. Laut der *schwachen Definition eines Clusters* von Radicchi et al. [RCC⁺04] muss für dessen Knoten C $k_C^{int} > k_C^{ext}$ gelten, woraus folgt $0,5 < \rho(C) \leq 1$. Da die relative Dichte in der Visualisierung nur für Gruppen berechnet wird, welche zuvor von einem Clustering-Algorithmus identifiziert wurden, kann davon ausgegangen werden, dass für alle Gruppen

$C \rho(C) > 0,5$ gilt. In der Visualisierung wird daher der Wertebereich $[0,5;1]$ der relativen Dichte auf den Grauwertbereich (Weiß-Schwarz) abgebildet.

In der aggregierten Ansicht werden lediglich die Kanten zwischen Gruppen dargestellt (Inter-Cluster-Kanten). Wie im Konzept erläutert, werden die Kanten zwischen zwei Gruppen nicht einzeln gezeichnet, sondern zu einer Kante aggregiert. Das Gewicht einer aggregierten Kante ist im ungewichteten Graphen die Summe der Einzelkanten und im gewichteten Graphen die Summe der Gewichte der Einzelkanten. Eine aggregierte Kante wird in der Darstellung durch eine gebogene Linie, genauer eine Spline-Kurve, repräsentiert. Das aggregierte Gewicht wird auf die Dicke der Linie abgebildet. Je größer der Abstand zwischen den beiden Gruppen, welche die aggregierte Kante verbindet, desto stärker ist die Krümmung der Linie. Auf diese Weise sollen Überschneidungen vermieden werden.

6.2.3. Detail-Ansicht

In der Detail-Ansicht wird, zusätzlich zur Cluster-Ansicht, die Struktur des Netzwerks in jedem Zeitschritt als Knoten-Kanten-Diagramm dargestellt (siehe Abb. 6.6). Die Knoten einer Gruppe sind so positioniert, dass sie auf einer Ellipsenbahn liegen (im Englischen als *circular layout* bezeichnet). Auf diese Weise können die Intra-Cluster-Kanten als gerade Linien gezeichnet werden. Wie in der aggregierten Ansicht, werden Inter-Cluster-Kanten als gebogene Linien gezeichnet. Je größer der Abstand zwischen den Knoten einer Inter-Cluster-Kante ist, desto stärker ist die Krümmung der Linie. Auf diese Weise sollen Überschneidungen mit anderen Kanten und den Knoten der Gruppen vermieden werden.

Die Detail-Ansicht ermöglicht das Untersuchen der Verbindungen innerhalb von Gruppen. So können Zusammenhänge zwischen der zeitlichen Entwicklung von Gruppen und ihrer

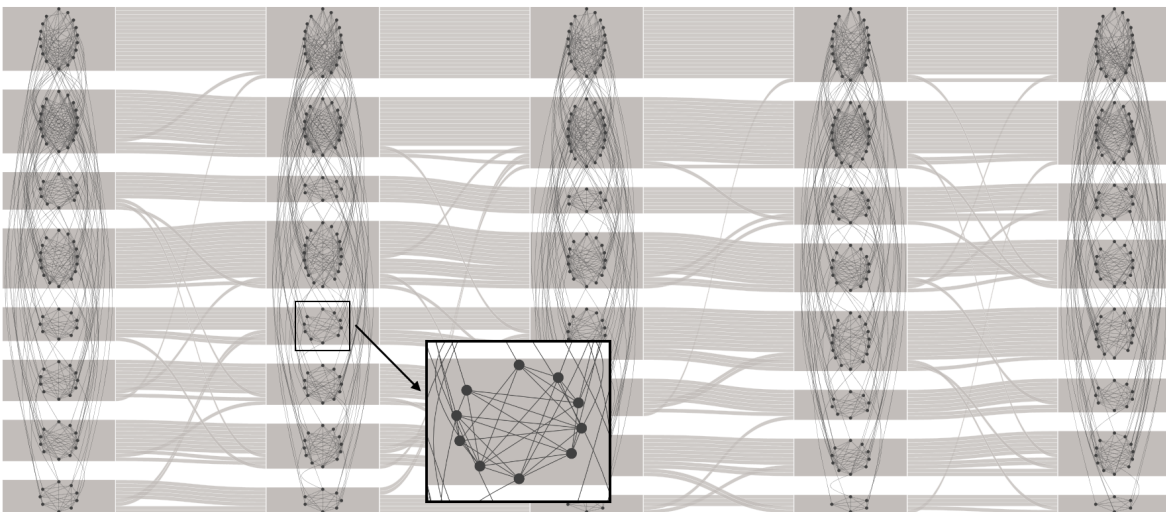


Abbildung 6.6.: Detail-Ansicht des Datensatzes Switch 100

internen Struktur hergestellt werden. Verlassen bspw. mehrere Knoten von einem zum nächsten Zeitschritt ihre Gruppe, so kann die interne Struktur der Gruppe in vorherigen Zeitschritten analysiert werden, um eventuelle Anzeichen für das Ereignis zu identifizieren.

6.3. Interaktion

Die Visualisierung bietet eine Reihe von Interaktionsmöglichkeiten. So kann jede Ansicht beliebig vergrößert werden, um Details in der Darstellung besser erkennen zu können. Vor allem in der Detail-Ansicht ist dies notwendig, um die interne Struktur der Gruppen untersuchen zu können. Wird die Ansicht dabei größer als das Fenster der Anwendung, so kann der angezeigte Bildausschnitt einfach mit gedrückter Maustaste verschoben werden.

In allen Ansichten können Transitionen und Gruppen selektiert werden. In der Detail-Ansicht können zudem einzelne Knoten selektiert werden. Auch das Selektieren mehrerer Elemente gleichzeitig ist möglich. Durch die Selektion einer Gruppe werden alle der Gruppe zugeordneten Knoten im gesamten dynamischen Graphen hervorgehoben (siehe Abb. 6.7). Auf diese Weise wird die vergangene und die zukünftige Entwicklung der Gruppe besser sichtbar. Durch die Selektion einzelner Knoten oder Transitionen werden die betreffenden Knoten ebenfalls im gesamten dynamischen Graphen hervorgehoben. So kann die Gruppenzugehörigkeit eines Knotens über die Zeit einfach erkannt werden.

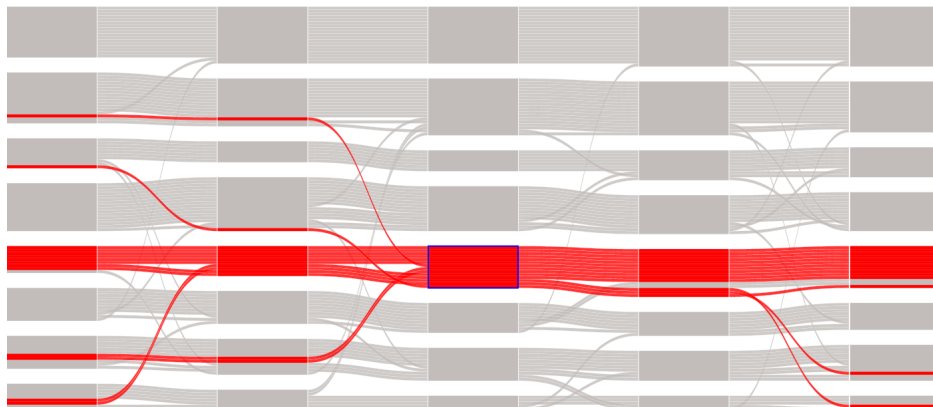


Abbildung 6.7.: Selektion eines Clusters (blau umrandet) in der Cluster-Ansicht (Switch 100 Datensatz)

Der *Kanten-Filter* dient bei der Visualisierung von gewichteten dynamischen Graphen dazu, Kanten mit niedrigem Gewicht auszublenken, um Kanten mit hohem Gewicht besser analysieren zu können. Der Schwellenwert zur Anzeige kann über einen Schieberegler in der Menüleiste eingestellt werden. Kanten mit einem Gewicht unterhalb des Schwellenwertes werden in der Visualisierung nicht gezeichnet (siehe Abb. 6.8). Zudem gibt es die Möglichkeit in der Detail-Ansicht alle Inter-Cluster-Kanten auszublenken, um so den Fokus auf die interne Struktur der Gruppen zu lenken (siehe Abb. 6.10).

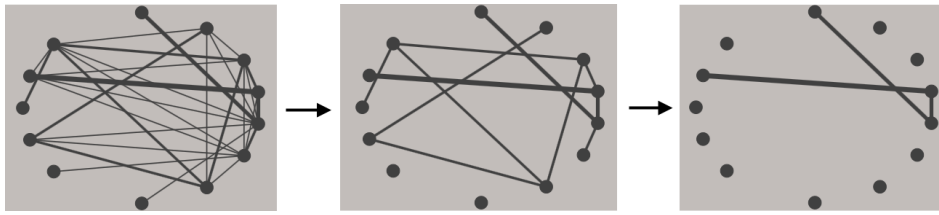


Abbildung 6.8.: Anwendungsbeispiel des Kanten-Filters – Von links nach rechts wird der Schwellenwert zur Anzeige von Kanten erhöht, was dazu führt, dass Kanten mit niedrigem Gewicht immer weiter ausgeblendet werden.

6.4. Einfärbungen

Die im Folgenden beschriebenen Einfärbungen können in allen Ansichten gewählt werden.

6.4.1. Dynamische Gruppen

Um die Entwicklung der Gruppen besser nachvollziehen zu können und dominierende, über die Zeit bestehende Gruppen zu identifizieren, wurde das Verfahren zur Gruppenverfolgung von Greene et al. [GDC10] implementiert (siehe Abschnitt 3.3). Die Grundidee des Verfahrens ist es, die Gruppen aufeinanderfolgender Zeitschritte miteinander zu vergleichen und ähnliche Gruppen in Meta-Gruppen, den dynamischen Gruppen, zusammenzufassen.

Zur Visualisierung der identifizierten dynamischen Gruppen muss in der Anwendung zunächst der Schwellenwert (in Abschnitt 3.3.3 mit θ bezeichnet) eingetragen und die Gruppenverfolgung durchgeführt werden. Zudem kann eingestellt werden, wie viele der gefundenen dynamischen Gruppen farblich hervorgehoben werden sollen. Wurden bspw. 100 dynamische Gruppen identifiziert und sollen nur zehn davon angezeigt werden, so werden die Gruppen mit den zehn größten Signifikanz-Werten eingefärbt (siehe Abschnitt 3.3.4). Bei bis zu sieben dargestellten dynamischen Gruppen werden wahrnehmungspsychologisch klar unterscheidbare Farben verwendet. Die Farbtabelle wurde mithilfe des Tools *ColorBrewer*⁵ erstellt. Bei der Hervorhebung von mehr als sieben Gruppen werden Farbtöne gewählt, welche auf dem Farbkreis des HSV-Farbraums den gleichen Winkelabstand zueinander besitzen. Abbildung 6.9 zeigt die Cluster-Ansicht des Merge & Split 100 Datensatzes mit Einfärbung der dynamischen Gruppen.

⁵<http://colorbrewer.org/>

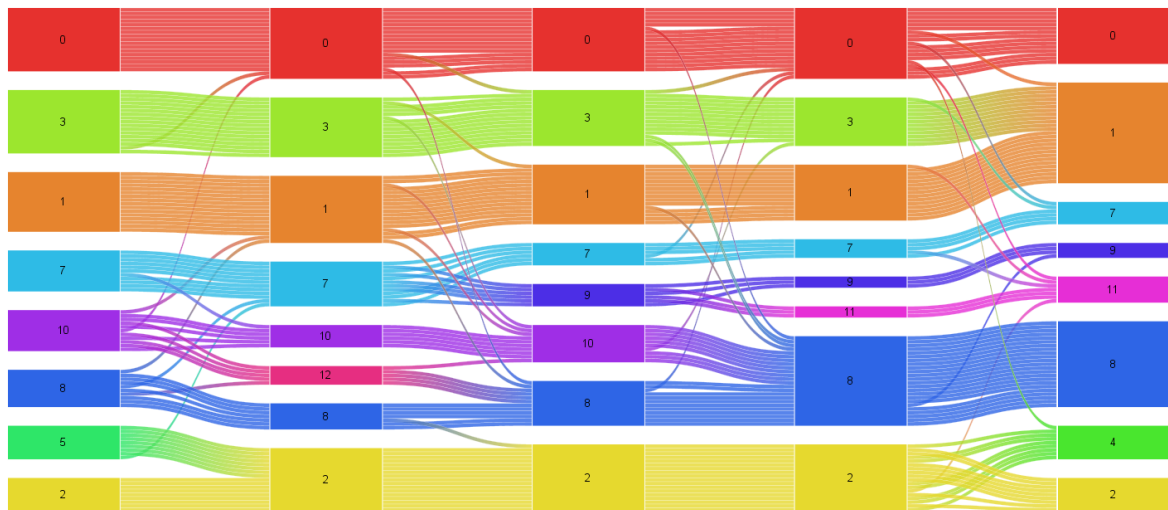


Abbildung 6.9.: Einfärbung der dynamischen Gruppen – Cluster-Ansicht des Merge & Split 100 Datensatzes mit Einfärbung von 13 dynamischen Gruppen. Zur besseren Unterscheidung wird zu jeder Zeitschritt-Gruppe auch die ID der dynamischen Gruppen (0-12) angezeigt. Als Schwellenwert bei der Gruppenverfolgung wurde $\theta = 0,3$ verwendet.

6.4.2. Knoten-Stabilität

Eine Anforderung an die Visualisierung ist, dass konstante Knoten von Knoten, welche oft ihre Gruppenzugehörigkeit ändern, effektiv unterschieden werden können. Um dies zu erreichen verwenden wir die von Takaffoli et al. [TFSZ11] definierte Metrik für die Stabilität eines Knotens v im Bezug auf seine Gruppenzugehörigkeit. Sei $A(v) = \{C_{t_1j_1}, \dots, C_{t_{a_j}j_a}\}$ die Menge der Zeitschritt-Gruppen, denen der Knoten v angehört (nach Zeitschritt sortiert), so ist die Stabilität von v definiert als [TFSZ11]⁶:

$$stability(v) = \frac{1}{n-1} \sum_{i=1}^{|A(v)|-1} sim(C_{t_{ij_i}}, C_{t_{i+1}j_{i+1}})$$

Die Funktion $sim()$ liefert die Jaccard-Ähnlichkeit zweier Mengen (siehe Abschnitt 3.3.2). Knoten, welche oft ihre Gruppenzugehörigkeit wechseln, haben eine Stabilität nahe „0“, wohingegen konstante Knoten eine Stabilität nahe „1“ besitzen. In der Visualisierung wird die Knoten-Stabilität $[0, 1]$ nun auf den Farbbereich Rot-Grün abgebildet (siehe Abb. 6.10).

⁶Takaffoli et al. wählen als Faktor nicht $\frac{1}{n-1}$, sondern $\frac{1}{|A(v)|-1}$

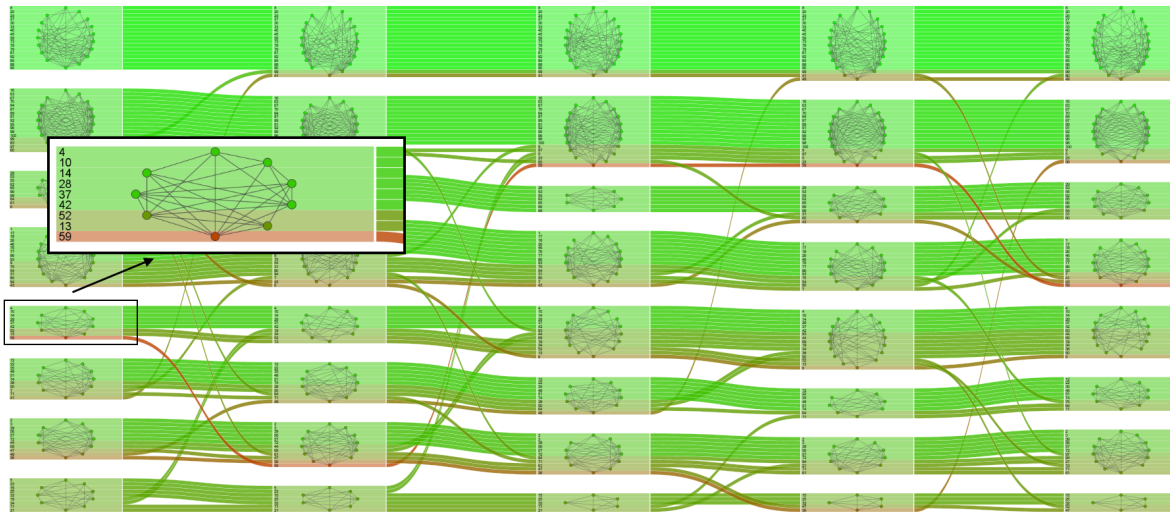


Abbildung 6.10.: Detail-Ansicht des Switch 100 Datensatzes mit Einfärbung der Knoten-Stabilität (Inter-Cluster-Kanten sind ausgeblendet). Grüne Knoten sind im Bezug auf ihre Gruppenzugehörigkeit relativ stabil (z. B. Knoten „4“), rote Knoten wechseln ihre Gruppe des öfteren (z. B. Knoten „59“). (Eine vergrößerte Version dieses Bildes ist im Anhang auf Seite 68 zu finden.)

6.5. Minimierung der Kreuzungen von Transitionen

Um die Visualisierung gut analysieren zu können, ist es wichtig, dass zwischen den Zeitschritten möglichst wenige Kreuzungen von Transitionen existieren. Dies ermöglicht ein effizientes Nachvollziehen der Gruppenentwicklungen. Die Gruppen und deren Knoten müssen folglich in jedem Zeitschritt vertikal so angeordnet werden, dass dies erreicht wird (im Folgenden als *lineare Ordnung* bezeichnet).

Die Minimierung der Kreuzungen von Transitionen in dieser Visualisierung ist ähnlich zu einem anderen Layout-Problem, der *Kreuzungsminimierung bei bipartiten Graphen* (engl. *bipartite graph crossing minimization*, kurz BGCM). Verfahren zur Lösung des BGCM-Problems (z. B. [CEKSo7]) könnten allerdings nur die Transitionen zwischen jeweils zwei Zeitschritten minimieren und somit keine globale Minimierung für den kompletten dynamischen Graphen erzeugen. Aus diesem Grund wurden zwei eigene, heuristische Verfahren entwickelt. Die Verfahren liefern keine optimale Lösung, aber das visuelle Ergebnis ist deutlich besser als bei naiven Anordnungen (bspw. die Gruppen vertikal nach ihrer Größe anzuordnen, wie in [OMB⁺07] und [RTJ⁺11]).

6.5.1. Globale Minimierung

Ziel des Verfahrens ist es, eine lineare Ordnung zu bestimmen, welche über alle Zeitschritte hinweg möglichst wenige Kreuzungen von Transitionen zur Folge hat. Vor allem die durch

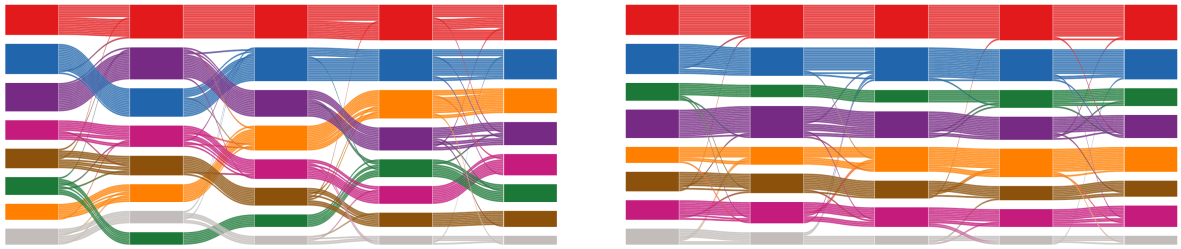


Abbildung 6.11.: Globale Minimierung der Kreuzungen von Transitionen (Switch 100 Datensatz) – Links: Die Gruppen werden in jeden Zeitschritt einfach nach ihrer Größe geordnet. Rechts: Die Gruppen und Knoten werden nach dem Verfahren zur globalen Minimierung der Kreuzungen angeordnet. In beiden Abbildungen sind die identifizierten dynamischen Gruppen hervorgehoben.

die Gruppenverfolgung identifizierten dynamischen Gruppen sollen so dargestellt werden, dass ihre Entwicklung gut nachvollziehbar ist.

Im ersten Schritt des Verfahrens werden dazu die Gruppen in jedem Zeitschritt entsprechend der Signifikanz ihrer zugehörigen dynamischen Gruppen angeordnet. Formal ausgedrückt bedeutet das: Die Gruppen $\{C_{t1}, \dots, C_{tk_t}\}$ im Zeitschritt t werden entsprechend dem Wert $\text{sig}(dmax(C_{ij}))$ absteigend angeordnet. Die Funktion $dmax(C_{ij})$ gibt die dynamische Gruppe D zurück, für welche gilt $C_{ij} \in D$. Falls C_{ij} teil mehrerer dynamischer Gruppen ist (vgl. C_{21} in Abb. 3.2), gibt $dmax(C_{ij})$ die dynamische Gruppe mit der höchsten Signifikanz zurück. Auf diese Weise werden die signifikanten dynamischen Gruppen im oberen Teil der Visualisierung platziert und Zeitschritt-Gruppen, welche zur selben dynamischen Gruppe gehören, befinden sich auf ähnlicher vertikaler Höhe.

Im zweiten Schritt werden die Knoten in jeder Zeitschritt-Gruppe entsprechend ihrer Stabilität (siehe Abschnitt 6.4.2) angeordnet. So werden alle konstanten Knoten einer Gruppe nah beieinander platziert. Ebenso werden Knoten, welche häufig ihre Gruppenzugehörigkeit ändern, nebeneinander positioniert. Diese Knotenanordnung ermöglicht eine gute Unterscheidung zwischen stabilen und instabilen Knoten.

Abbildung 6.11 zeigt das Ergebnis der globalen Minimierung beim Switch 100 Datensatz und vergleicht es mit der naiven Anordnung nach der Gruppengröße. Auch bei Abbildung 6.10 wurde die globale Minimierung durchgeführt. Es lässt sich hier gut erkennen, dass die Knoten in jeder Gruppe entsprechend ihrer Stabilität sortiert sind.

6.5.2. Lokale Minimierung

Unterscheiden sich die Gruppenstrukturen der einzelnen Zeitschritte stark, so liefert das Verfahren zur globalen Minimierung oftmals kein zufriedenstellendes Ergebnis (vgl. Abb. 6.13b). Aus diesem Grund wurde ein weiteres Verfahren entwickelt, welches vorrangig versucht

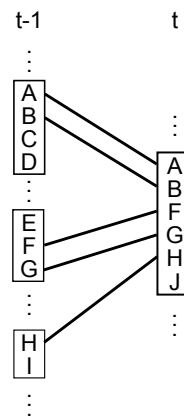
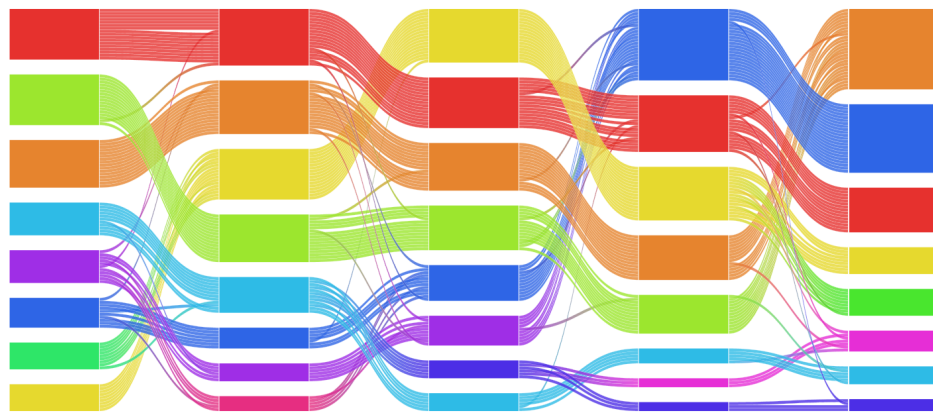


Abbildung 6.12.: Knotenordnung bei der lokalen Minimierung – Die Ordnung der Knoten in Zeitschritt t wird bestimmt durch deren vertikale Position (d. h. Ordnung) in Zeitschritt $t - 1$. Knoten J kommt in Zeitschritt t hinzu und wird darum ganz zuletzt einsortiert.

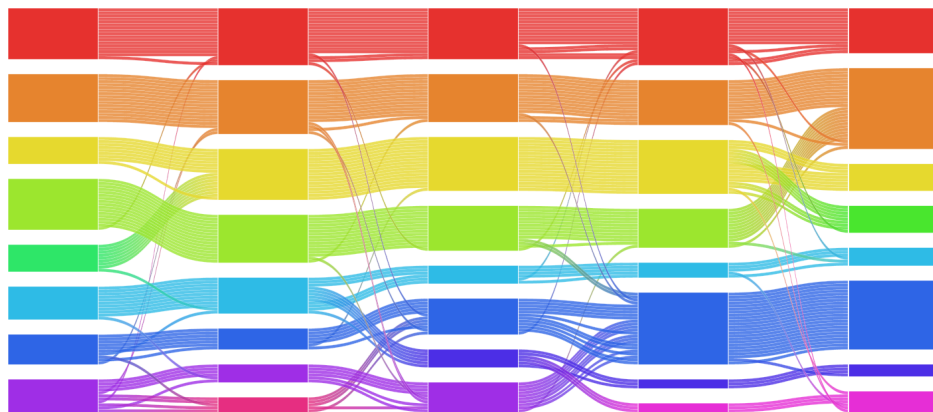
die Anzahl der Transitions-Kreuzungen zwischen jeweils zwei benachbarten Zeitschritten zu minimieren. Zunächst wird der Algorithmus zur Bestimmung der Gruppenordnung erläutert. Ziel des Algorithmus ist, ähnliche Gruppen benachbarter Zeitschritte auf ähnlicher vertikaler Höhe anzuordnen. Der Algorithmus liefert für jeden Zeitschritt t des dynamischen Graphen eine geordnete Liste der Gruppen L_t .

1. Bestimme L_1 , indem die Gruppen von P_1 entsprechend ihrer Größe geordnet werden.
2. Für jeden Zeitschritt $t > 1$:
 - a) Für jede Gruppe $C_{tj} \in P_t$:
 - i. Finde zu C_{tj} die Gruppe C_{t-1k} aus dem vorherigen Zeitschritt, für welche $\text{sim}(C_{tj}, C_{t-1k})$ maximal wird.
 - ii. Falls eine Gruppe C_{t-1k} mit $\text{sim}(C_{tj}, C_{t-1k}) > 0$ gefunden wurde, füge C_{tj} der Liste von Nachfolgern von C_{t-1k} hinzu ($\text{successors}(C_{t-1k})$).
 - iii. Andernfalls füge C_{tj} der Liste neuer Gruppen im Zeitschritt t hinzu ($\text{new}(t)$).
 - b) Iteriere über die geordnete Liste L_{t-1} und betrachte jeweils eine Gruppe C_{t-1k} daraus:
 - i. Füge die Gruppe(n) $\text{successors}(C_{t-1k})$ in L_t ein.
 - c) Füge die Gruppe(n) $\text{new}(t)$ in L_t ein.

Die Knoten einer Gruppe im Zeitschritt t werden entsprechend ihrer vertikalen Position (d. h. Ordnung) im vorherigen Zeitschritt angeordnet (siehe Abb. 6.12). Abbildung 6.13 zeigt einen Vergleich der Verfahren zur Minimierung der Transitions-Kreuzungen.



(a) Keine Minimierung (Gruppen nach Größe geordnet)



(b) Globale Minimierung



(c) Lokale Minimierung

Abbildung 6.13.: Vergleich der Verfahren zur Minimierung der Transitions-Kreuzungen am Beispiel des Merge & Split 100 Datensatzes

7. Fallstudien

In diesem Kapitel werden zwei reale Netzwerk-Datensätze mit der entwickelten Visualisierung analysiert.

7.1. Hypertext 2009

Der erste Datensatz wurde im Rahmen der *ACM Hypertext 2009*¹ Konferenz erhoben und ist online verfügbar². Etwa 110 Teilnehmer (ca. 75% aller Teilnehmer) erklärten sich bereit, während der Konferenz RFID-Einheiten zu tragen, mit deren Hilfe persönliche Kontakte zwischen den Teilnehmern registriert werden konnten. Die Konferenz erstreckte sich über drei Tage. Die RFID-Einheiten meldeten einen *Kontakt* an ein zentrales System, wenn sich zwei Personen räumlich nah beieinander aufhielten und sich mit dem Oberkörper zugewandt waren. Im Intervall von jeweils 20 Sekunden prüfte das System, welche Kontakte momentan aktiv waren und protokollierte diese. Für jeden Kontakt wurden die IDs der beteiligten Personen und das Zeitintervall gespeichert. Der Datensatz wurde in der Vergangenheit bereits von Isella et al. [ISB⁺11] untersucht.

Für die Visualisierung mussten die Daten aggregiert werden, um eine überschaubare Anzahl an Zeitschritten zu erhalten. Zunächst musste hierfür eine Zeitschrittweite festgelegt werden, wobei mit mehreren Schrittweiten experimentiert wurde. Letztendlich stellt sich eine Schrittweite von 6 Stunden als brauchbar heraus, da hier die Gruppenstrukturen über die Zeit zumindest teilweise konstant waren, was eine Analyse ermöglichte. Bei der Aggregation wurden in jedem Zeitschritt alle Kontakte zwischen zwei Personen zu einem Kontakt mit der summierten Dauer zusammengefasst. Jeder Zeitschritt kann dann als ungerichteter, gewichteter Graph modelliert werden, indem die Teilnehmer durch Knoten und die Kontakte durch Kanten repräsentiert werden. Die Kantengewichte repräsentieren die Kontaktdauer im Zeitschritt. Der Datensatz wurde so in einen dynamischen Graphen überführt. Für jeden Zeitschritt-Graphen wurde anschließend durch den Infomap-Algorithmus [RB08] eine Gruppenstruktur bestimmt.

Abbildung 7.2 zeigt die Cluster-Ansicht der resultierenden Visualisierung. Durch den Algorithmus zur Gruppenverfolgung konnten einige dynamische Gruppen identifiziert werden (eingefärbt). Etwa die Hälfte aller Teilnehmer ist Teil einer solchen, stabileren Gruppe. Diese

¹<http://www.ht2009.org>

²<http://www.sociopatterns.org>

7. Fallstudien

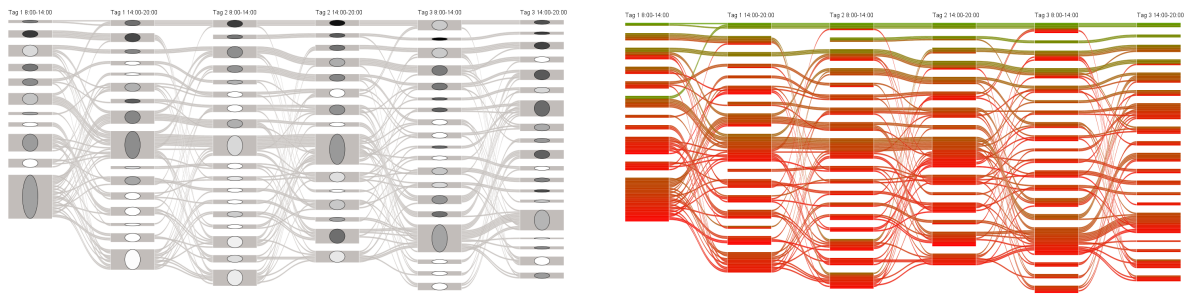


Abbildung 7.1.: Hypertext 2009 – Links: Aggregierte Ansicht des Datensatzes (ohne Inter-Cluster-Kanten). Der Großteil der Gruppen, vor allem in der unteren Hälfte, besitzt nur einen niedrigen inneren Zusammenhalt. Rechts: Cluster-Ansicht mit Einfärbung der Knoten-Stabilität. Nur wenige Knoten im oberen Bereich der Ansicht sind im Bezug auf ihre Gruppenzugehörigkeit stabil (grün). Der Großteil der Knoten wechselt häufig die Gruppe (rot).

Teilnehmer haben im Verlauf der Konferenz vorwiegend einen Kontakt zu denselben 2-6 anderen Teilnehmern unterhalten. Die dynamischen Gruppen besitzen meist einen konstanten „Kern“ von Personen, welche über die gesamte Zeit Teil der Gruppe sind (z. B. Kasten 2 in Abb. 7.2). Wechselt man in die Detail-Ansicht, so zeigt sich, dass die Elemente des Kerns meist dicht untereinander verbunden sind, wohingegen temporäre Knoten der Gruppe oft nur einzelne Verbindungen zum Kern aufweisen (siehe Kasten 1 & 3 in Abb. 7.2).

Trotz des großen Aggregationsintervalls von 6 Stunden ist eine hohe Fluktuation in der Gruppenzugehörigkeit der Knoten erkennbar (siehe Abb. 7.1). Dies liegt sicher an der Natur der Daten: Während einer wissenschaftlichen Konferenz tauschen sich die Teilnehmer meist mit vielen anderen Teilnehmern aus und knüpfen neue Kontakte. Daher war eine konstante Gruppenstruktur über die Zeit nicht zu erwarten. Durch die Verwendung eines evolutionären Clustering-Algorithmus könnten eventuell stabilere Gruppenstrukturen identifiziert werden.

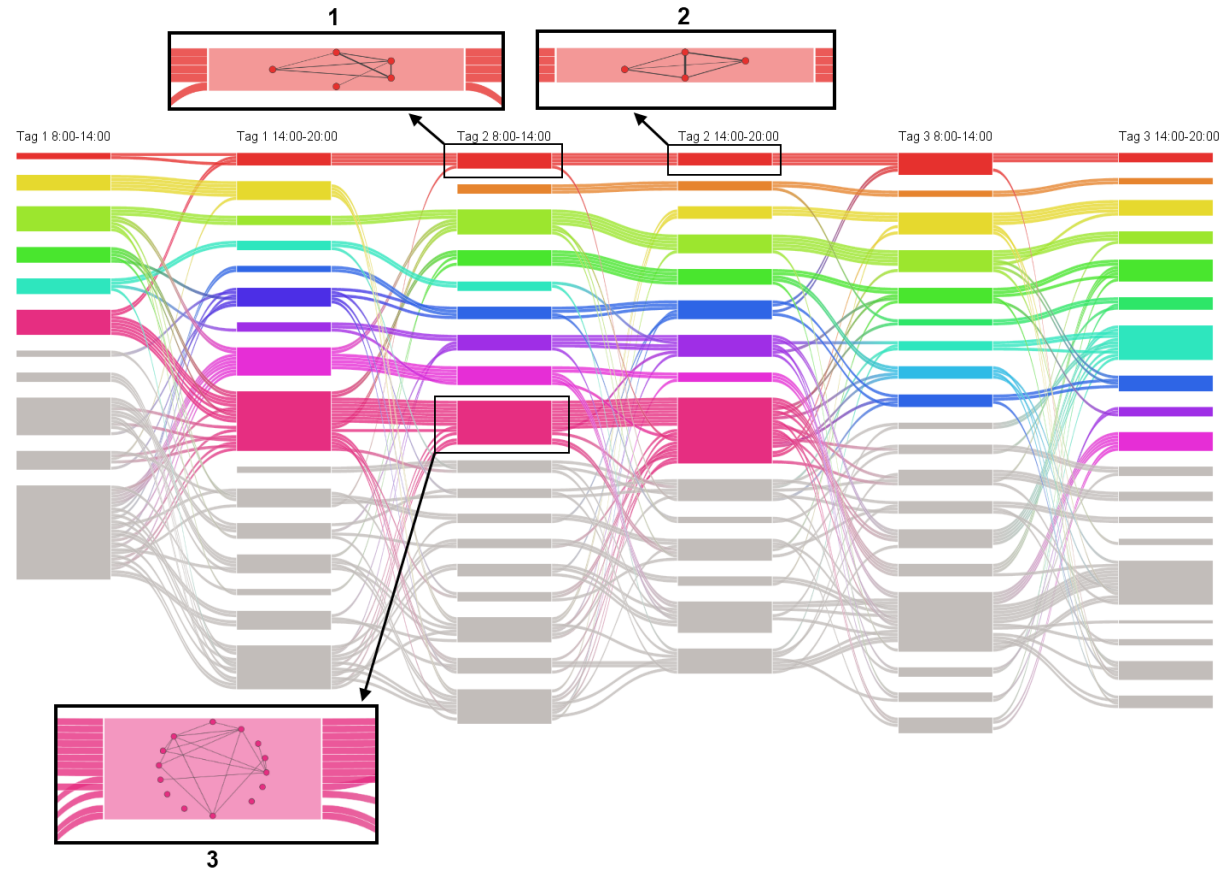


Abbildung 7.2.: Hypertext 2009 – Cluster-Ansicht mit Einfärbung der dynamische Gruppen (globale Minimierung der Kreuzungen). Als Schwellenwert für die Gruppenverfolgung wurde $\theta = 0,25$ gewählt. So wurden 27 dynamische Gruppen identifiziert, von welchen die 14 signifikantesten farblich hervorgehoben sind. Die vergrößerten Ausschnitte zeigen die Detail-Ansicht ausgewählter Gruppen. Bei Ausschnitt 3 ist der Kanten-Filter mit einem Schwellenwert von 1 Minute aktiv, d. h. es werden nur Verbindung mit einer Dauer von mindestens eine Minute angezeigt.

7. Fallstudien

1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
		Welt				Welt				Welt			
Europa				Europa				Europa				Europa	
Afrika		Afrika		Afrika		Afrika		Afrika		Afrika		Afrika	
Asien				Asien				Asien				Asien	
	Nord-amerika			Nord-amerika		Nord-amerika		Nord-amerika		Nord-amerika	Nord-amerika		
	Süd-amerika		Süd-amerika		Süd-amerika		Süd-amerika		Süd-amerika			Süd-amerika	

Tabelle 7.1.: Übersicht der FIFA-Meisterschaften für Fußball-Nationalmannschaften.

7.2. Fußball-Begegnungen

Der zweite Datensatz beschreibt alle Begegnungen zwischen Fußball-Nationalmannschaften in den Jahren 1992-2005. Enthalten sind nicht nur die Begegnungen in offiziellen Turnieren, sondern auch Freundschaftsspiele. Die Begegnungen können als dynamischer gewichteter Graph modelliert werden, wobei die Spiele pro Jahr zusammengefasst werden. Die Knoten repräsentieren Mannschaften und die Kanten repräsentieren die Begegnungen zwischen zwei Mannschaften im jeweiligen Jahr. Das Gewicht einer Kante entspricht die Anzahl der Begegnungen beider Mannschaften.

Um die Visualisierung besser verstehen und überprüfen zu können, werden im Folgenden einige Hintergrundinformationen zur Struktur des internationalen Fußballs gegeben. Organisiert wird der internationale Fußball durch den Weltfußballverband *FIFA*³, welcher unter anderem die Fußball-Weltmeisterschaft ausrichtet. Die FIFA gliedert sich in sechs Kontinentalverbände: Europa, Afrika, Asien, Nord- und Mittelamerika, Südamerika und Ozeanien. Jeder Kontinentalverband organisiert in regelmäßigen Abständen ein Turnier zwischen den Nationalmannschaften seines Kontinents (z. B. organisiert der europäische Kontinentalverband die Europameisterschaft). Tabelle 7.1 zeigt, in welchen Jahren internationalen Turniere im Zeitraum von 1992 bis 2005 stattgefunden haben. Allen Turnieren geht eine Qualifikationsphase voraus, durch welche die Teilnehmer ermittelt werden.

Abbildung 7.5 zeigt die Visualisierung des Datensatzes mit der Cluster-Ansicht. Auffällig ist, dass die grüne dynamische Gruppe genau jeden zweiten Zeitschritt auftritt. Vergleicht man die Jahre des Auftretens mit Tabelle 7.1, so korrelieren diese mit den Jahren der Welt- bzw. Europameisterschaften. Untersucht man die einzelnen Elemente der grünen Gruppen, so ist darin meist der Großteil der WM- bzw. EM-Teilnehmer enthalten. So enthält bspw. die grüne Gruppe im Jahr 1996 alle 16 Teilnehmer der EM und die Gruppe im Jahr 1998 zumindest 24/32 Teilnehmer der WM. Die Gruppen enthalten nicht exakt die Turniermannschaften, da der Datensatz auch Freundschaftsspiele beinhaltet. Die Visualisierung zeigt, dass es eine große Überschneidung der Mannschaften von WM- und EM-Gruppen gibt. Eine EM-Gruppe zerfällt stets im nächsten Zeitschritt in eine Reihe kleinerer Gruppen und vereint sich anschließend wieder in der WM-Gruppe (siehe Abb. 7.3).

Die blaue dynamische Gruppe besteht zum Großteil aus südamerikanischen Mannschaften. Sie vereint sich ebenfalls immer mit der WM-Gruppe (1994, 1998 und 2002) und spal-

³<http://www.fifa.com/>

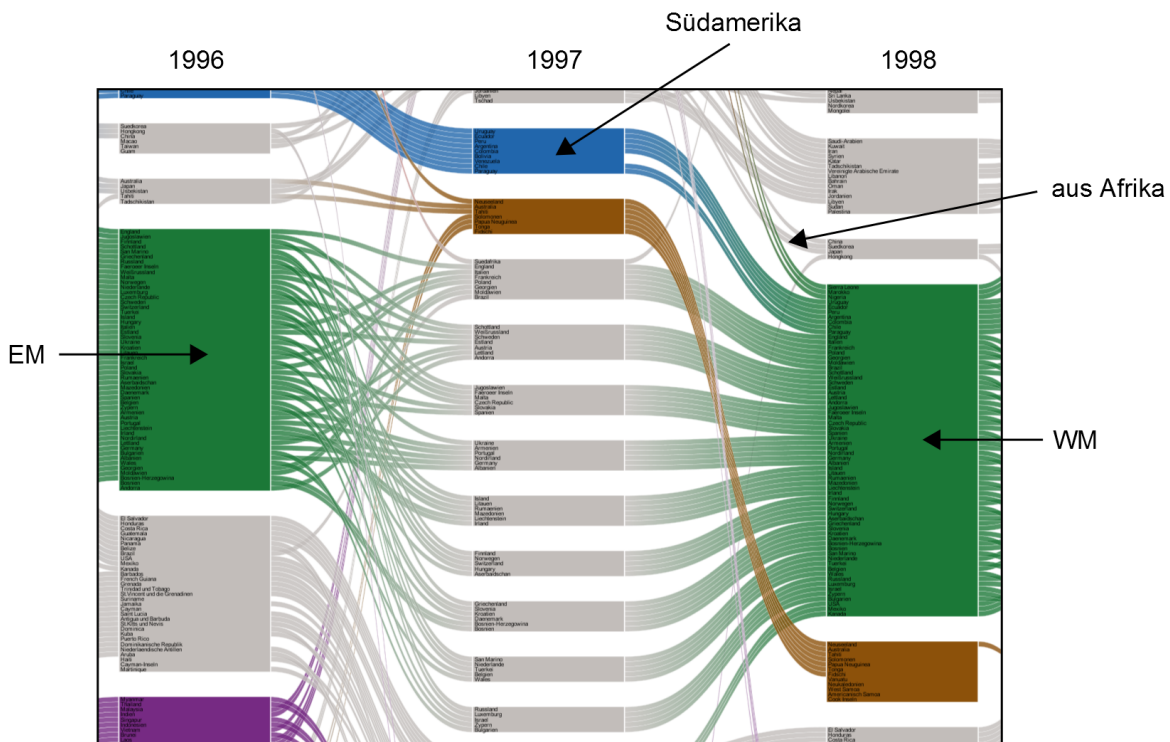


Abbildung 7.3.: Fussball-Datensatz – Vergrößerung des in Abbildung 7.5 markierten Ausschnitts. Die EM-Gruppe von 1996 spaltet sich 1997 auf und vereinigt sich 1998 wieder in der WM-Gruppe. Zur WM-Gruppe kommen ebenfalls Mannschaften aus Südamerika und Afrika hinzu, welche zum Großteil aus wirklichen Teilnehmern des WM-Turniers bestehen.

tet sich im darauffolgenden Zeitschritt wieder ab. Auch diese Beobachtung ist erklärbar. Südamerikanische Mannschaften schneiden bei Weltmeisterschaften meist gut ab, sind also lange im Turnier und absolvieren viele Spiele. In den WM-Jahren sind sie darum stets Teil der WM-Gruppen. In den Jahren zwischen Weltmeisterschaften spielen die südamerikanischen Mannschaften meist nur gegeneinander oder gegen Mannschaften aus Nord- und Mittelamerika.

In Abbildung 7.5 fällt zudem auf, dass die rote dynamische Gruppe über den gesamten Zeitraum relativ stabil ist. Die Gruppe spaltet sich einige Male auf (1994, 1998 und 2002), vereinigt sich aber jeweils kurz darauf wieder. Es findet kaum ein Wechsel von Elementen zu anderen Gruppen statt. Noch deutlicher zeigt sich dies, wenn eine rote Zeitschritt-Gruppe selektiert wird (siehe Abb. 7.4). Bei genauerer Analyse zeigt sich, dass die roten Gruppen ausschließlich aus afrikanische Nationalmannschaften bestehen. Die Visualisierung legt den Schluss nahe, dass afrikanische Mannschaften häufig gegeneinander spielen, aber nur selten gegen Nationen anderer Kontinente. Dies ist plausibel, da die Afrikameisterschaft im sehr kurzem Abstand von jeweils zwei Jahren stattfindet (siehe Tabelle 7.1). Nach Abschluss einer

7. Fallstudien

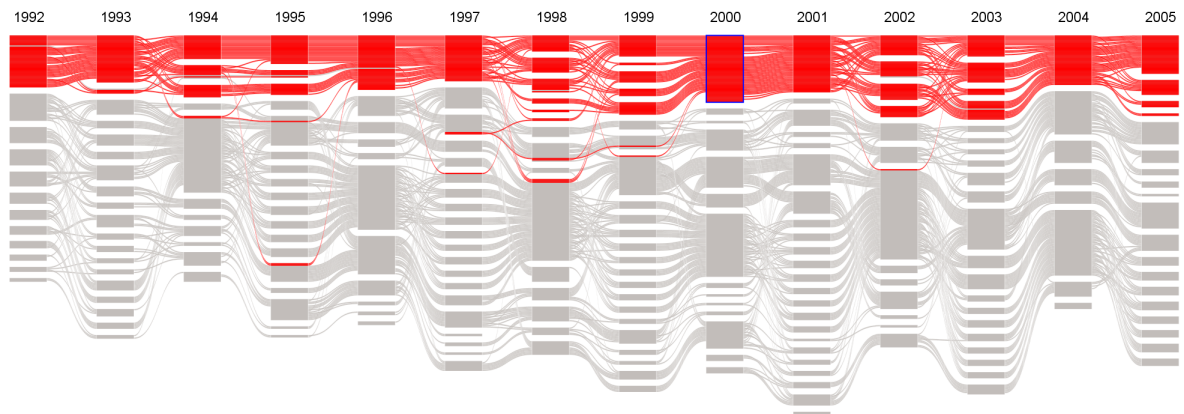


Abbildung 7.4.: Fussball-Datensatz – Selektion der Afrika-Gruppe im Jahr 2000 (blau umrandet). Durch die Hervorhebung wird deutlich sichtbar, dass die Gruppe trotz Aufteilungen und Vereinigungen über die Zeit sehr stabil ist und es nur sehr wenige Wechsel von Elementen zu anderen Gruppen gibt.

Meisterschaft beginnt direkt die Qualifikationsphase für das nächste Turnier. Die Beteiligung afrikanischer Mannschaften bei Weltmeisterschaften ist meist gering (bspw. waren bei der WM 1994 von 24 Mannschaften nur drei vom afrikanischen Kontinent). Zudem scheiden diese bei Weltmeisterschaften meist früh aus. Es gibt auch nur wenige Freundschaftsspiele zwischen afrikanischen Mannschaften und Mannschaften anderer Kontinente. Dies alles stützt die Beobachtung der wenigen Wechsel zu anderen Gruppen.

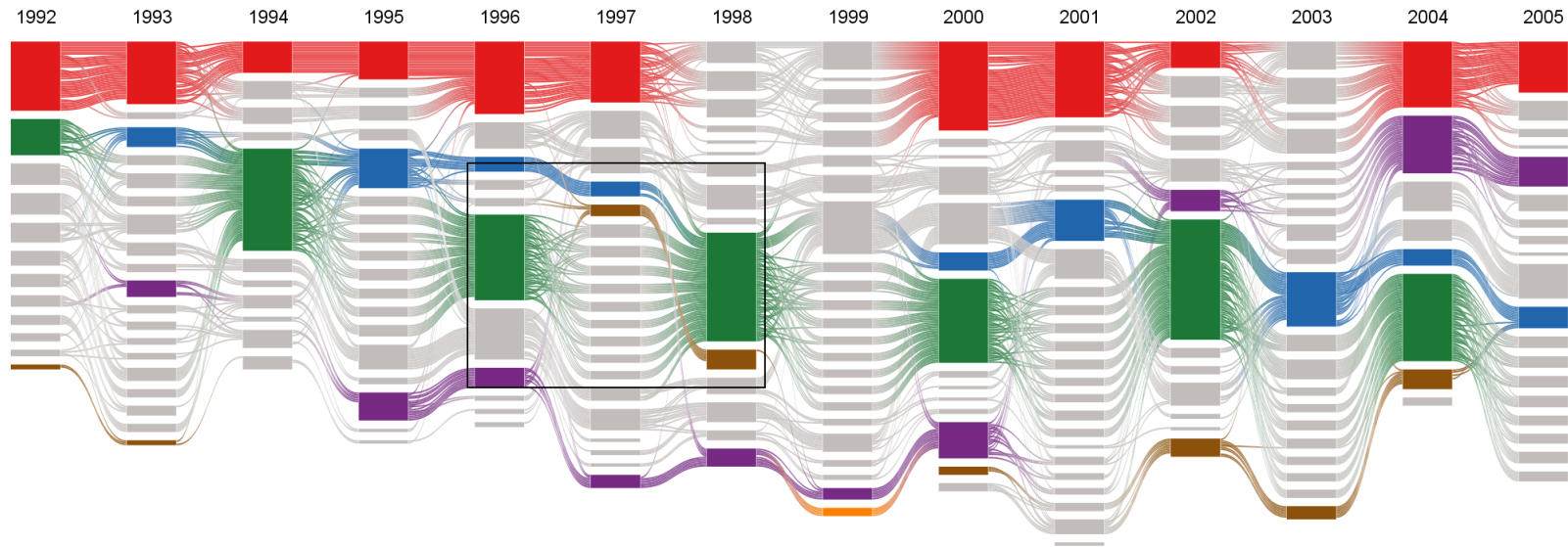


Abbildung 7.5.: Fussball-Datensatz – Cluster-Ansicht mit Einfärbung der dynamischen Gruppen (lokale Minimierung der Kreuzungen). Als Schwellenwert für die Gruppenverfolgung wurde $\theta = 0,3$ verwendet. Auf diese Weise wurden 60 dynamische Gruppen identifiziert, von welchen die sechs signifikantesten eingefärbt sind. Diese sechs Gruppen können den Kontinentalverbänden der FIFA zugeordnet werden: Rot → Afrika, Grün → Welt/Europa, Blau → Südamerika, Lila/Orange → Asien, Braun → Ozeanien. Der schwarz umrandete Ausschnitt wird in Abbildung 7.3 vergrößert dargestellt.

8. Zusammenfassung und Ausblick

Ziel dieser Arbeit war die Entwicklung einer Visualisierung von Gruppen (Clustern) in dynamischen Netzwerken. Die Visualisierung sollte es ermöglichen, die zeitliche Evolution der Gruppen nachvollziehen zu können. Zudem sollte auch die Struktur des Netzwerks zu jedem Zeitpunkt sichtbar sein.

Zunächst wurden bestehende Visualisierungskonzepte für Netzwerke bzw. Gruppen untersucht und bewertet. Es zeigte sich, dass keines der untersuchten Konzepte die in dieser Arbeit formulierten Anforderungen vollständig erfüllt. Daher wurde auf Basis bestehender Ansätze ein eigenes Visualisierungskonzept entwickelt. Grundidee des Konzeptes ist es, die Darstellung der Gruppenentwicklung und der Netzwerkstruktur in eine statische Ansicht zu integrieren. Auf diese Weise können Zusammenhänge zwischen der Gruppenentwicklung und der Netzwerkstruktur hergestellt werden. Das Konzept beinhaltet mehrere Ansichten mit jeweils unterschiedlichem Detailgrad.

Das entwickelte Konzept wurde anschließend in einer Java-Anwendung umgesetzt. Es wurden mehrere Interaktionsmöglichkeiten implementiert, um die Analyse der Gruppenentwicklung zu unterstützen. So können einzelne Elemente sowie Gruppen durch Selektion global hervorgehoben werden. Auf diese Weise kann z. B. die Gruppenzugehörigkeit von Elementen über die Zeit einfach verfolgt werden. Zudem können dominierende Gruppen, welche über die Zeit hinweg bestehen, identifiziert und hervorgehoben werden. Elemente, welche im Bezug auf ihre Gruppenzugehörigkeit stabil sind, können visuell von häufig wechselnden Elementen unterschieden werden. Um eine möglichst klare Darstellung zu erreichen, wurden zwei Verfahren zur Minimierung von Linienkreuzungen in der Ansicht implementiert. Zuletzt wurde die Visualisierung zur Analyse zweier realer Netzwerke eingesetzt.

Ausblick

Im Folgenden werden einige mögliche Erweiterungen der Anwendung aufgelistet, welche ihre Nutzbarkeit weiter steigern könnten.

Verwendung eines evolutionären Clustering-Verfahrens: Zur Bestimmung der Gruppenstrukturen in den Zeitschritten wird momentan der statische Infomap-Algorithmus verwendet (siehe Abschnitt 6.1). Besser wäre die Verwendung eines evolutionären Clustering-Verfahrens (siehe Abschnitt 3.2.2), da dieses zufällige Schwankungen in den Zeitschritten kompensiert und damit stabilere Gruppenstrukturen identifiziert.

Weitere Layout-Algorithmen für die Darstellung der Zeitschritt-Graphen: Die Knoten jeder Gruppe in der Detail-Ansicht werden momentan einfach auf einer Ellipsenbahn angeordnet (*circular layout*). Diese Positionierung ist nicht optimal, um die interne Struktur einer Gruppe gut erkennen zu können. Besser geeignet wären Layout-Algorithmen wie das *force-directed placement* [FR91], die eng miteinander verbundene Knoten nah beieinander anordnen. Bestehende Layout-Algorithmen können allerdings nicht unverändert in die Visualisierung integriert werden, da hier die vertikale Position der Knoten fest vorgegeben ist. Es ist zu untersuchen, welche Layout-Algorithmen auch mit dieser Beschränkung sinnvoll verwendet werden könnten.

Unterstützung gerichteter Graphen: Die Anwendung kann bis jetzt nur ungerichtete dynamische Graphen visualisieren. Für gerichtete Graphen muss die Darstellung angepasst werden, da nun zwischen zwei Knoten nicht maximal eine Kante, sondern zwei Kanten verlaufen können. Um dennoch eine gute Wahrnehmbarkeit zu erreichen, könnte z. B. in der aggregierten Ansicht der Ansatz der *TimeArcTrees* (siehe Abschnitt 4.2.2) mit den separat verlaufenden *upward* und *downward edges* verwendet werden.

A. Anhang

Es folgt eine vergrößerte Version von Abbildung 6.10 auf Seite 53 zur besseren Lesbarkeit. Die Abbildung zeigt die Visualisierung des Switch 100 Datensatzes mit Einfärbung der Knoten-Stabilität. Knoten, welche häufig ihre Gruppenzugehörigkeit ändern (rot), können in der Abbildung gut von stabilen Knoten (grün) unterschieden werden. Gut erkennbar ist zudem, wie die Knoten in jeder Gruppe durch das Verfahren zur globalen Minimierung der Kreuzungen entsprechend ihrer Stabilität geordnet sind.

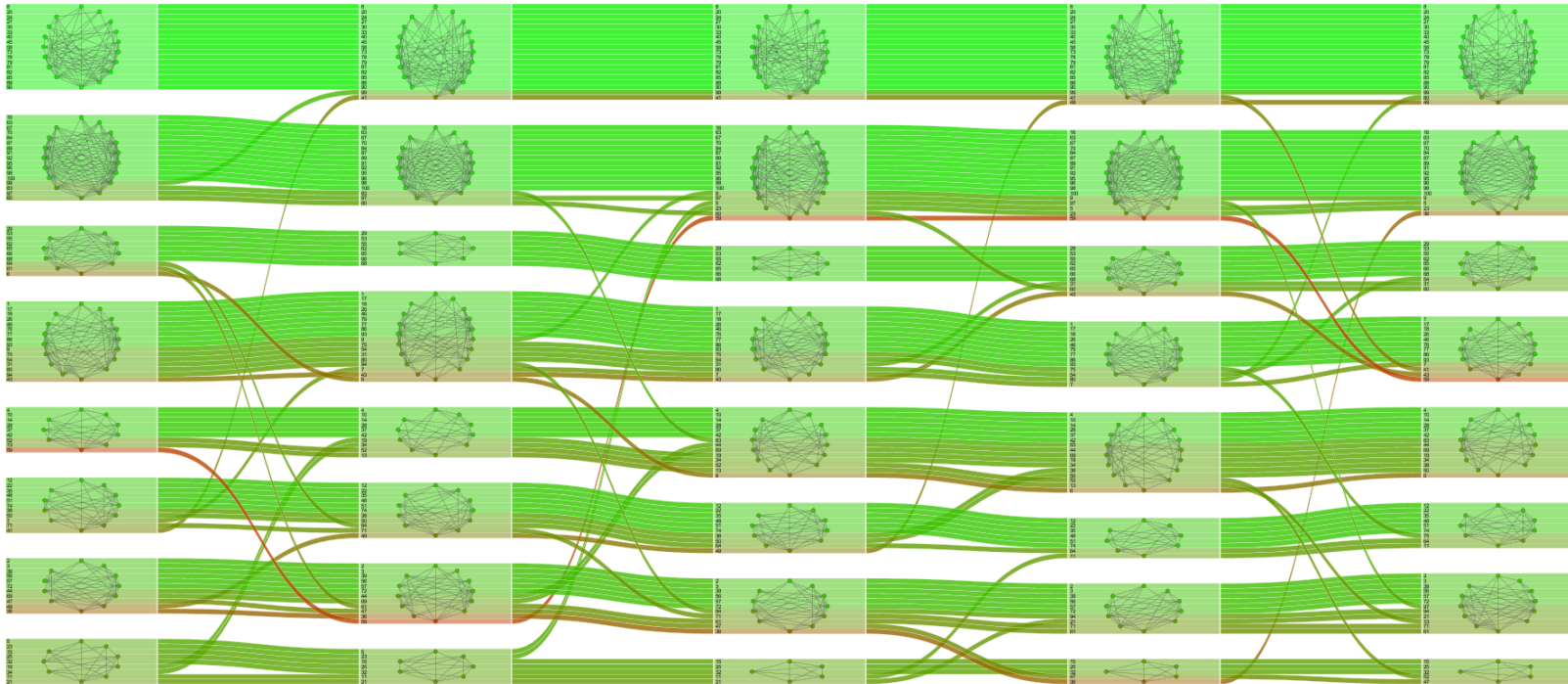


Abbildung A.1.: Detail-Ansicht des Switch 100 Datensatzes mit Einfärbung der Knoten-Stabilität (Inter-Cluster-Kanten sind ausgeblendet). Grüne Knoten sind im Bezug auf ihre Gruppenzugehörigkeit stabil, rote Knoten wechseln ihre Gruppe des öfteren. Zur Bestimmung der linearen Ordnung wurde das Verfahren zur globalen Minimierung verwendet.

Literaturverzeichnis

- [BBV⁺12] F. Beck, M. Burch, C. Vehlow, S. Diehl, D. Weiskopf. Rapid Serial Visual Presentation in Dynamic Graph Visualization. In IEEE, Herausgeber, *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2012)*, Band 2012, S. 1–8. 2012. (Zitiert auf Seite 24)
- [BC03] U. Brandes, S. Corman. Visual Unrolling of Network Evolution and the Analysis of Dynamic Discourse. *Information Visualization*, 2(1):40–50, 2003. (Zitiert auf den Seiten 7 und 26)
- [BETT98] G. Battista, P. Eades, R. Tamassia, I. Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998. (Zitiert auf Seite 23)
- [BGLL08] V. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008. (Zitiert auf Seite 17)
- [BHKLo6] L. Backstrom, D. Huttenlocher, J. Kleinberg, X. Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 44–54. ACM, 2006. (Zitiert auf Seite 18)
- [BLM⁺06] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D. Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006. (Zitiert auf den Seiten 7, 13 und 16)
- [BM98] V. Batagelj, A. Mrvar. Pajek - Program for large network analysis. *Connections*, 21(2):47–57, 1998. (Zitiert auf Seite 45)
- [BVB⁺11] M. Burch, C. Vehlow, F. Beck, S. Diehl, D. Weiskopf. Parallel edge splatting for scalable dynamic graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2344–2353, 2011. (Zitiert auf Seite 24)
- [CEKS07] O. Cakiroglu, C. Erten, . Karatas, M. Sözdinler. Crossing minimization in weighted bipartite graphs. *Experimental Algorithms*, S. 122–135, 2007. (Zitiert auf Seite 53)
- [CKTo6] D. Chakrabarti, R. Kumar, A. Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 554–560. ACM, 2006. (Zitiert auf den Seiten 18 und 19)

- [CSZ⁺07] Y. Chi, X. Song, D. Zhou, K. Hino, B. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 153–162. ACM, 2007. (Zitiert auf Seite 19)
- [Don00] S. van Dongen. Graph clustering by flow simulation. 2000. (Zitiert auf Seite 27)
- [ELLS11] B. Everitt, S. Landau, M. Leese, D. Stahl. *Cluster Analysis*. John Wiley & Sons, 5 Auflage, 2011. (Zitiert auf Seite 17)
- [ER10] D. Edler, M. Rosvall. The Map Generator software package, 2010. <http://www.mapequation.org>. (Zitiert auf den Seiten 7, 33 und 35)
- [For10] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010. (Zitiert auf den Seiten 14, 17, 18 und 48)
- [FR91] T. Fruchterman, E. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991. (Zitiert auf den Seiten 23 und 66)
- [Fre96] L. Freeman. Some antecedents of social network analysis. *Connections*, 19(1):39–42, 1996. (Zitiert auf Seite 16)
- [FT04] Y. Frishman, A. Tal. Dynamic drawing of clustered graphs. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, S. 191–198. IEEE, 2004. (Zitiert auf den Seiten 7 und 24)
- [GBD09] M. Greilich, M. Burch, S. Diehl. Visualizing the evolution of compound digraphs with TimeArcTrees. In *Computer Graphics Forum*, 3, S. 975–982. Wiley Online Library, 2009. (Zitiert auf den Seiten 7 und 25)
- [GDC10] D. Greene, D. Doyle, P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, S. 176–183. IEEE, 2010. (Zitiert auf den Seiten 7, 14, 15, 19, 20, 21, 41, 44 und 51)
- [GN02] M. Girvan, M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. (Zitiert auf Seite 17)
- [HMM00] I. Herman, G. Melançon, M. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000. (Zitiert auf Seite 23)
- [ID90] A. Inselberg, B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization'90*, S. 361–378. IEEE Computer Society Press, 1990. (Zitiert auf Seite 35)
- [ISB⁺11] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J. Pinton, W. Van den Broeck. What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180, 2011. (Zitiert auf Seite 57)

- [Jac12] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912. (Zitiert auf Seite 20)
- [KH09] M. Kim, J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proceedings of the VLDB Endowment*, 2(1):622–633, 2009. (Zitiert auf den Seiten 14 und 19)
- [LCZ⁺08] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, B. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th international conference on World Wide Web*, S. 685–694. ACM, 2008. (Zitiert auf den Seiten 14 und 19)
- [LF09] A. Lancichinetti, S. Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5):056117, 2009. (Zitiert auf Seite 17)
- [LSP⁺10] A. Lex, M. Streit, C. Partl, K. Kashofer, D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1027–1035, 2010. (Zitiert auf den Seiten 7, 29, 30, 31 und 32)
- [MMBd05] J. Moody, D. McFarland, S. Bender-deMoll. Dynamic network visualization. *American Journal of Sociology*, 110(4):1206–1241, 2005. (Zitiert auf Seite 23)
- [NG04] M. Newman, M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004. (Zitiert auf den Seiten 7, 9, 14 und 15)
- [OMB⁺07] M. Ogawa, K. Ma, C. Bird, P. Devanbu, A. Gourley. Visualizing social interaction in open source software projects. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, S. 25–32. IEEE, 2007. (Zitiert auf den Seiten 7, 25, 27, 28, 34 und 53)
- [PBV07] G. Palla, A. Barabasi, T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007. (Zitiert auf den Seiten 7, 10 und 15)
- [PDFV05] G. Palla, I. Derényi, I. Farkas, T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005. (Zitiert auf den Seiten 7, 17 und 18)
- [PHG06] H. Purchase, E. Hoggan, C. Görg. How important is the Mental map?: an empirical investigation of a dynamic graph layout algorithm. In *Proceedings of the 14th international conference on Graph drawing*, S. 184–195. Springer-Verlag, 2006. (Zitiert auf Seite 23)
- [POM09] M. Porter, J. Onnela, P. Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009. (Zitiert auf den Seiten 7 und 10)
- [RBo8] M. Rosvall, C. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008. (Zitiert auf den Seiten 17, 45 und 57)

- [RB10] M. Rosvall, C. Bergstrom. Mapping change in large networks. *PLoS one*, 5(1):e8694, 2010. (Zitiert auf den Seiten 7, 33 und 34)
- [RCC⁺04] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004. (Zitiert auf Seite 48)
- [Rom04] C. Romburg. *Cluster Analysis For Researchers*. Lulu Press, 2004. (Zitiert auf Seite 17)
- [RTJ⁺11] K. Reda, C. Tantipathananandh, A. Johnson, J. Leigh, T. Berger-Wolf. Visualizing the Evolution of Community Structures in Dynamic Social Networks. In *Computer Graphics Forum*, 3, S. 1061–1070. Wiley Online Library, 2011. (Zitiert auf den Seiten 7, 16, 35, 36 und 53)
- [TBW09] C. Tantipathananandh, T. Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 827–836. ACM, 2009. (Zitiert auf Seite 35)
- [TFSZ11] M. Takaffoli, J. Fagnan, F. Sangi, O. Zaiane. Tracking changes in dynamic information networks. In *Computational Aspects of Social Networks (CASoN), 2011 International Conference on*, S. 94–101. IEEE, 2011. (Zitiert auf den Seiten 14, 19 und 52)
- [Tuf90] E. Tufte. *Envisioning information*. Graphics Press, 1990. (Zitiert auf Seite 25)
- [VLKS⁺11] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J. Fekete, D. Fellner. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. In *Computer graphics forum*, 6, S. 1719–1749. Wiley Online Library, 2011. (Zitiert auf den Seiten 23 und 24)

Alle URLs wurden zuletzt am 18. 12. 2012 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift