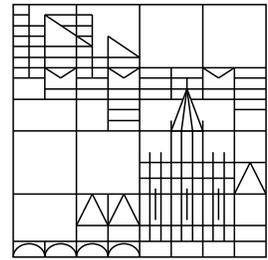


Universität Konstanz



Über das Zeichnen von Graphen

Ulrik Brandes
Dorothea Wagner

Konstanzer Schriften in Mathematik und Informatik

Nr. 85, März 1999

ISSN 1430–3558

Über das Zeichnen von Graphen

Ulrik Brandes

Dorothea Wagner

Universität Konstanz

Fakultät für Mathematik und Informatik

78457 Konstanz

Vorname.Nachname@uni-konstanz.de

Zusammenfassung

Zeichnungen sind nicht nur ein ansprechendes, sondern auch ein sehr effektives Mittel, um die durch einen Graphen repräsentierte Information zu vermitteln. Der geeignete Entwurf solcher Zeichnungen ist jedoch schon bei kleinen Graphen eine schwierige und zeitraubende Arbeit, die nach Automatisierung geradezu ruft. Wir leiten daher eine Formalisierung des Problems her und stellen anhand von Anwendungen aus der Soziologie und dem Verkehrswesen Möglichkeiten des automatischen Zeichnens dar.

1 Einleitung

„How to Draw a Graph“ überschrieb 1963 William T. Tutte einen Aufsatz [43], der heute zu den Klassikern der algorithmischen Graphentheorie gehört. Tutte könnte damit als Begründer des *Graphenzeichnens* bezeichnet werden, wobei sich seine Fragestellung tatsächlich auf das Problem beschränkte, einen planaren Graphen kreuzungsfrei, geradlinig und mit konvexen Flächen in die Ebene einzubetten. Im gleichen Jahr veröffentlichte Donald E. Knuth „Computer Drawn Flowcharts“ in *Communications of the ACM* [31], womit er neben der Anwendungsrelevanz des Graphenzeichnens auch dessen Einordnung als grundlegende Fragestellung der Informatik manifestierte.¹

Als aktuelles Forschungsgebiet der Informatik geht Graphenzeichnen über die Erstellung geometrischer Repräsentationen abstrakter Graphen und Netzwerke hinaus und umfaßt alle Aspekte

¹Es sei bemerkt, daß sich natürlich viel früher schon die Physiker mit diesem Problem beschäftigt haben. Siehe etwa “The Scientific Papers of James Clerk Maxwell”, 1890, Neuauflage 1965, Kapitel XXIV und LXXV [35].

der Visualisierung von Strukturinformation. Die automatische Erzeugung von Graphzeichnungen hat wichtige Anwendungen in Kernbereichen der Informatik wie Compilerbau, Datenbanken, Software-Engineering, VLSI- und Netzwerk-Design und graphischen Benutzerschnittstellen. Anwendungen in anderen Gebieten betreffen vor allem die graphische Datenanalyse (beispielsweise in den Ingenieurwissenschaften, Chemie, Biologie oder Sozial- und Politikwissenschaft) und die Visualisierung zur Informationsvermittlung (beispielsweise durch Schaltpläne, Streckennetze oder Organigramme). Das Feld der für Mathematiker und Informatiker relevanten Aspekte reicht von Graphen- und Ordnungstheorie und Modellierungsfragen über den Entwurf entsprechender Algorithmen bis hin zu Design und Implementation unterstützender Software-Systeme. Entsprechend ist Graphenzeichnen zu einem Thema geworden, das sich stark wachsender Beliebtheit erfreut. Seit 1993 findet jährlich eine internationale Tagung zu *Graph Drawing* statt [17, 40, 4, 34, 16, 47], und ein erstes Buch, das die algorithmischen Aspekte des Graphenzeichnens behandelt, ist 1999 erschienen [18].

Bei der Wiedergabe relationaler Daten sind im wesentlichen die drei in Bild 1 gezeigten Darstellungsformen gebräuchlich, die wir hier an einem wichtigen Beispiel aus der Analyse Sozialer Netzwerke illustrieren. Es handelt sich dabei um die Heiratsbeziehungen florentinischer Adelsfamilien des 15. Jahrhunderts, die bei der Erklärung, warum anstelle der reicheren Strozzi die Medici zu herausragender Bedeutung gelangten, eine wichtige Rolle spielen [36].

Textliche Darstellung. Die vielleicht genaueste und flexibelste, gleichzeitig aber auch am wenigsten intuitive Art, einen Graphen zu vermitteln, ist die textuelle Beschreibung, bestehend aus Listen der Knoten und Kanten, gegebenenfalls erweitert um zusätzliche Erklärungen und Interpretationen der reinen Beziehungsdaten. Diese Darstellung ist rein sequentiell, strukturelle Zusammenhänge sind vergleichsweise schwer erkennbar und schlecht einprägar.

Tabellarische Darstellung. Inzidenz- und Adjazenzmatrizen sind typischerweise kompakter als eine Beschreibung in Textform. Strukturelle Eigenschaften können in beschränktem Umfang sichtbar gemacht werden, aber Zusatzinformationen passen nicht in das unflexible Format. Tabellen eignen sich vor allem zum Austausch von Rohdaten.

Graphische Darstellung. Flexibler als die tabellarische Darstellung und trotzdem kompakt ist die Darstellung von Beziehungsdaten als Diagramm. Die Knoten und Kanten des Graphen werden explizit oder implizit durch graphische Zeichen repräsentiert. Strukturelle Zusammenhänge können in ihrer Gesamtheit vermittelt und durch spezielle Markierungen noch verdeutlicht werden.

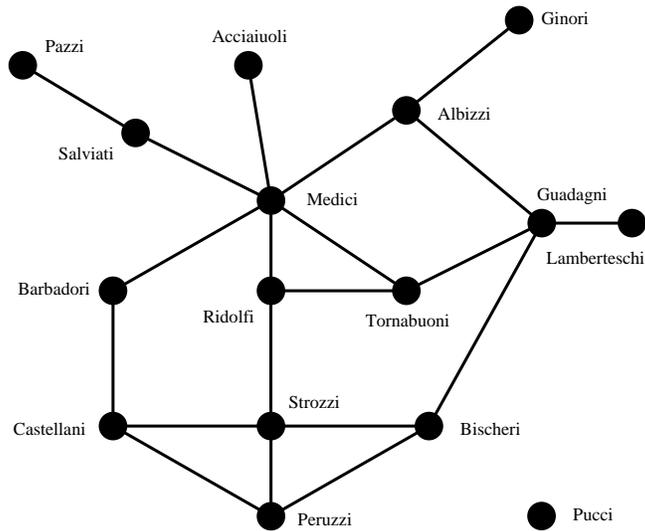
Eine besonders gebräuchliche Form der graphischen Darstellung ist die *geradlinige Repräsentation*, bei der Knoten durch Punkte in der Ebene und Kanten durch gerade Linien dargestellt werden. Beim algorithmisch anspruchsvollsten Problem, dem Layout des Graphen, sind Koordinaten dann nur für die Knotenpunkte anzugeben. Beispiele dieser und anderer graphischer Repräsentationsformen sind in Bild 2 gezeigt.

$$\begin{aligned}
 V &= \{ \text{Acciaiuoli, Albizzi, Barbadori, Bischeri, Castellani, Ginori, Guadagni, Lamberteschi,} \\
 &\quad \text{Medici, Pazzi, Peruzzi, Pucci, Ridolfi, Salviati, Strozzi, Tornabuoni} \} \\
 E &= \{ \{ \text{Acciaiuoli, Medici} \}, \{ \text{Albizzi, Ginori} \}, \{ \text{Albizzi, Guadagni} \}, \{ \text{Albizzi, Medici} \}, \\
 &\quad \{ \text{Barbadori, Castellani} \}, \{ \text{Barbadori, Medici} \}, \{ \text{Bischeri, Lamberteschi} \}, \{ \text{Bischeri, Peruzzi} \}, \\
 &\quad \{ \text{Bischeri, Strozzi} \}, \{ \text{Castellani, Peruzzi} \}, \{ \text{Castellani, Strozzi} \}, \{ \text{Guadagni, Lamberteschi} \}, \\
 &\quad \{ \text{Guadagni, Tornabuoni} \}, \{ \text{Medici, Ridolfi} \}, \{ \text{Medici, Salviati} \}, \{ \text{Medici, Tornabuoni} \}, \\
 &\quad \{ \text{Pazzi, Salviati} \}, \{ \text{Peruzzi, Strozzi} \}, \{ \text{Ridolfi, Strozzi} \}, \{ \text{Ridolfi, Tornabuoni} \} \}
 \end{aligned}$$

(a) Textliche Darstellung

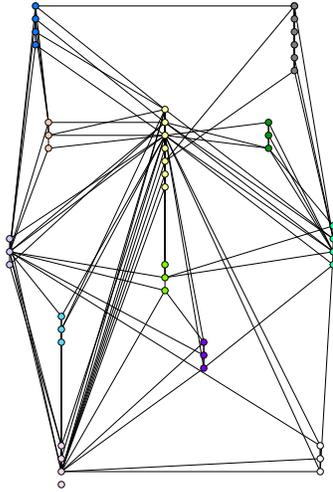
	Acciaiuoli	Albizzi	Barbadori	Bischeri	Castellani	Ginori	Guadagni	Lamberteschi	Medici	Pazzi	Peruzzi	Pucci	Ridolfi	Salviati	Strozzi	Tornabuoni
Acciaiuoli	1
Albizzi	1	1	.	1
Barbadori	.	.	.	1	.	.	.	1
Bischeri	1	.	.	.	1	1	.
Castellani	.	.	1	1	.	.	.	1	.
Ginori	.	1
Guadagni	.	1	.	1	.	.	.	1	1
Lamberteschi	1
Medici	1	1	1	1	1	.	1
Pazzi	1	.	.
Peruzzi	.	.	.	1	1	1	.
Pucci
Ridolfi	1	1	1
Salviati	1	1
Strozzi	.	.	.	1	1	1	.	1
Tornabuoni	1	.	1	.	.	.	1

(b) Tabellarische Darstellung

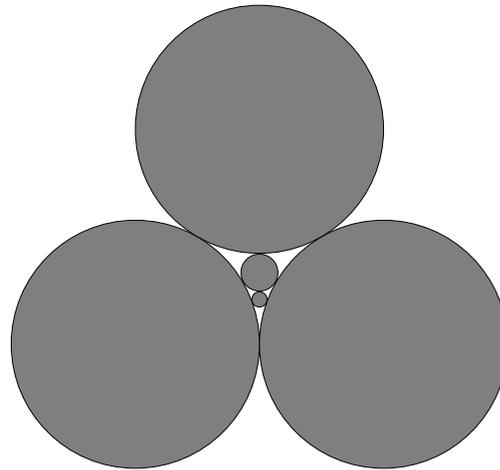


(c) Graphische Darstellung

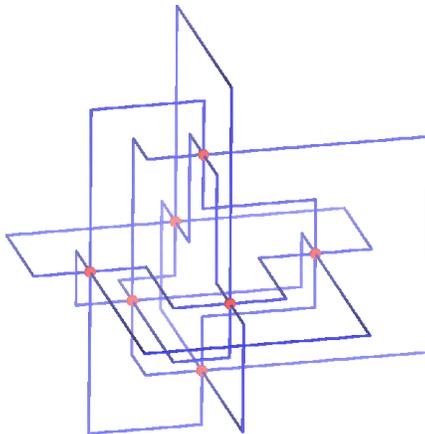
Bild 1: Darstellungsformen für Graphen



(a) Zweidimensionale geradlinige Repräsentation: Knoten sind durch Punkte, Kanten durch gerade Linien dargestellt. Gewinner des *Graph Drawing Contest 1997*, Batajelj/Mrvar, [20].

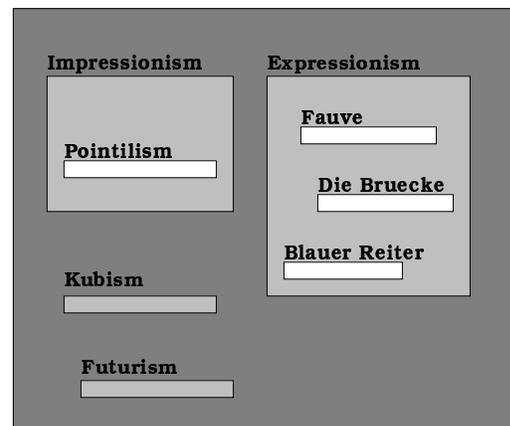


(b) Scheibenrepräsentation: Knoten sind durch Kreise, Kanten implizit durch Berührungen von Kreisen dargestellt. K_5 abzüglich einer Kante; nach [11].



(c) Dreidimensionale orthogonale Repräsentation: Knoten sind durch Punkte, Kanten durch Folgen achsenparalleler Strecken dargestellt. Layout von K_7 mit 2 Knicken pro Kante; nach [49], Betrachterposition aus [46].

Klassische Moderne



(d) Zweidimensionale Inklusionsrepräsentation: Knoten sind durch Flächen, Kanten implizit durch Flächeninklusion dargestellt. Nach [29].

Bild 2: Formen der graphischen Repräsentation abstrakter Graphen. Scheibenrepräsentationen sind nur für planare Graphen, Inklusionsrepräsentationen nur für Bäume möglich.

Die Auswahl einer speziellen graphischen Repräsentation hängt in erster Linie von der Realisierbarkeit (nicht alle Repräsentationsformen sind für alle Graphen realisierbar) und vom Anwendungshintergrund ab. Da beim Entwurf von Layoutalgorithmen üblicherweise Nebenbedingungen und erstrebenswerte Eigenschaften der Darstellung zu beachten sind, ergeben sich im allgemeinen komplexe Optimierungsprobleme, die für die meisten Graphenklassen \mathcal{NP} -schwer sind. Zum Beispiel könnten bei Wahl der geradlinigen Repräsentation die Positionen einiger Knoten eingeschränkt sein. Eine naheliegende Nebenbedingung bei planaren Graphen ist Kreuzungsfreiheit, allgemein erstrebenswerte Eigenschaften eines Layouts sind häufig *uniforme Kantenlängen*, *gleichmäßige Knotenverteilung* und *nicht zu kleine Winkel zwischen benachbarten Kanten*. Die Entscheidung, ob ein gegebener Graph eine geradlinige Darstellung mit Kanten gleicher Länge hat, ist \mathcal{NP} -vollständig [27], ebenso ist die Minimierung der Anzahl an Kreuzungen [26] und die Maximierung des minimalen Winkels [25] \mathcal{NP} -schwer, um nur einige Beispiele zu nennen.

In diesem Aufsatz beschränken wir uns auf die graphische Darstellung relationaler Binärdaten. Um eine einheitliche Beschreibung verschiedener Layoutverfahren zu ermöglichen, entwickeln wir eine allgemeine Formulierung des Layouts als Optimierungsproblem. Wir können natürlich nur einen kleinen Ausschnitt der verfügbaren methodischen Ansätze ansprechen, werden diesen aber an Beispielen aus zwei verschiedenen Anwendungsgebieten, der *Analyse Sozialer Netzwerke* und der *Verkehrsplanung und -optimierung*, vertiefen.

2 Methodische Grundlagen

2.1 Layoutmodelle

Ein Layout eines Graphen besteht in der Festlegung topologischer und geometrischer Eigenschaften der graphischen Repräsentation, insbesondere also von Positionen der die Elemente des Graphen repräsentierenden Zeichen. Die festzulegenden Eigenschaften werden durch entsprechende Variablen ausgedrückt, Nebenbedingungen an das Layout durch die Angabe von Wertebereichen und zulässigen Kombinationen und Optimierungskriterien durch entsprechende Zielfunktionen.

Layoutelemente. Zu einem Graph G mit festgelegter graphischer Repräsentation nennen wir die zugehörigen graphischen Zeichen zusammen mit ihren topologischen oder geometrischen Eigenschaften *Layoutelemente*. Zum Beispiel sind die Layoutelemente einer geradlinigen Repräsentation bereits durch die Knotenpositionen gegeben.

Bezeichne $L = L(G) = \{\lambda_1, \dots, \lambda_l\}$ die Menge der Layoutelemente. Für jedes $\lambda \in L$ gibt es dann eine Menge \mathcal{X}_λ zulässiger Werte der entsprechenden Eigenschaft. Entsprechend legt jeder Vektor $x \in \mathcal{X} = \mathcal{X}_{\lambda_1} \times \dots \times \mathcal{X}_{\lambda_l}$ ein *Layout* fest. Wir bezeichnen mit $x_\Lambda = (x_\lambda)_{\lambda \in \Lambda}$ das Teillayout der Layoutelemente aus $\Lambda \subseteq L$.

Nebenbedingungen. Die Wahl eines bestimmten Wertes für ein Layoutelement kann eigentlich zulässige Werte anderer Layoutelemente verbieten. Um derartige Beschränkungen auszudrücken, bezeichne $R_{\{\lambda_{i_1}, \dots, \lambda_{i_k}\}} \subseteq \mathcal{X}_{\lambda_{i_1}} \times \dots \times \mathcal{X}_{\lambda_{i_k}}$, $1 \leq i_1 < \dots < i_k \leq l$, eine k -stellige Nebenbedingung. $\mathcal{R}_k = \bigcup_{1 \leq i_1 < \dots < i_k \leq l} R_{\{\lambda_{i_1}, \dots, \lambda_{i_k}\}}$ sei die Menge aller k -stelligen Nebenbedingungen, und $\mathcal{R} = \bigcup_{k=1}^l \mathcal{R}_k$ die Menge aller Nebenbedingungen. Dann ist

$$\mathcal{X} \cap \mathcal{R} \stackrel{\text{def}}{=} \{x \in \mathcal{X} : x_\Lambda \in R_\Lambda \text{ for all } R_\Lambda \in \mathcal{R}, \Lambda \subseteq L\}$$

die Menge der *zulässigen Layouts*. Wir bezeichnen Layouts, die eine Nebenbedingung $R \in \mathcal{R}$ oder eine Menge von Nebenbedingungen $\mathcal{R}' \subseteq \mathcal{R}$ erfüllen, mit $\mathcal{X} \cap R$ bzw. $\mathcal{X} \cap \mathcal{R}'$.

Zielfunktionen. Abhängig von der Information, die durch das Layout eines Graphen vermittelt werden soll, sind gewisse Layouteigenschaften besonders wünschenswert. Um die jeweilige Qualität verschiedener Layouts zu messen, wird daher eine Zielfunktion $U : \mathcal{X} \cap \mathcal{R} \rightarrow \mathbb{R}$ definiert. Aus technischen Gründen wird U oft auch auf nichtzulässige Layouts erweitert, d.h. $U : \mathcal{X} \rightarrow \mathbb{R}$. Da es meist einfacher ist, unerwünschte Eigenschaften anzugeben, sehen wir U als eine Kostenfunktion an. Die Zielfunktion ist damit ein Maß für die „Häßlichkeit“ des Layouts und über $\mathcal{X} \cap \mathcal{R}$ zu minimieren.

Im allgemeinen ist es schwierig, „globale“ Kriterien anzugeben, welche die Gesamtqualität des Layouts widerspiegeln. Einfacher ist es, diese durch die Kombination einer Reihe „lokaler“ Kriterien zu messen, in denen die Güte von Konfigurationen kleiner Teilmengen der Layoutelemente bewertet wird. Die wechselseitige Abhängigkeit von Werten der Layoutelemente wird durch einen *Interaktionsgraph* $\mathcal{G} = (L, \mathcal{E})$ modelliert. Die Kantenmenge \mathcal{E} beschreibt ein *Nachbarschaftssystem* $\eta = \bigcup_{\lambda \in L} \eta_\lambda$, in dem die *Nachbarschaft* $\eta_\lambda \subseteq L \setminus \{\lambda\}$ von $\lambda \in L$ gerade die Menge derjenigen Layoutelemente ist, die gemeinsam mit λ in einem lokalen Gütekriterium auftreten. Diese Interaktionen sind symmetrisch, und folglich gilt $\lambda_2 \in \eta_{\lambda_1} \Leftrightarrow \lambda_1 \in \eta_{\lambda_2}$ für alle $\lambda_1, \lambda_2 \in L$. Mithin ist \mathcal{G} ungerichtet. Es sei $\mathcal{C} = \mathcal{C}(\eta)$ eine Menge vollständiger Teilgraphen, sogenannter *Cliquen* in \mathcal{G} . Wir definieren ein *lokales Gütekriterium* als *Interaktionspotential* (oder kurz *Potential*) einer Clique $C \in \mathcal{C}$ durch eine Funktion $U_C : \mathcal{X} \rightarrow \mathbb{R}$, die

$$x_C = y_C \quad \Rightarrow \quad U_C(x) = U_C(y)$$

für alle $x, y \in \mathcal{X}$ erfüllt. Als globale Zielfunktion dient uns die *Energie* $U(x) = \sum_{C \in \mathcal{C}} U_C(x)$ des Layouts x .

Da die Zielfunktion $U : \mathcal{X} \cap \mathcal{R} \rightarrow \mathbb{R}$ die Festlegung der Layoutelemente, die Einschränkungen auf Kombinationen ihrer Werte und die Kriterien für die Güte des Layouts zusammenfaßt, nennen wir sie auch kurz das *Layoutmodell*.

Ordnen wir den Layouts $x \in \mathcal{X}$ Wahrscheinlichkeiten

$$P(X = x) = \frac{1}{Z} e^{-U(x)},$$

zu, wobei $Z = \sum_{y \in \mathcal{X}} e^{-U(y)}$ die normalisierende Konstante sei, so ist die Verteilung P^X eine Gibbs-Verteilung und die Zufallsvariable X ein Gibbs-Zufallsfeld. Wir nennen U , X , und deren Verteilung P^X daher auch *Zufallsfeld-Layoutmodell* für G . Die Wahrscheinlichkeiten $P(X = x)$ hängen nur von der Energie ab, wobei ein Layout niedriger Energie wahrscheinlicher ist als ein Layout hoher Energie. Gibbs-Verteilungen sind in der Thermodynamik und statistischen Mechanik wohlbekannt und werden in zahlreichen Anwendungen genutzt. Ein Überblick mit Schwerpunkt auf Anwendungen in der Bildverarbeitung wird in [48] gegeben. Für die Anwendung beim Graphenzeichnen steht damit eine reichhaltige Theorie zur Verfügung. Die Herleitung eines schlüssigen Ansatzes für das *dynamische Layoutproblem* [7, 9] ist ein Beispiel für die Nützlichkeit dieser Sichtweise. Schließlich lassen sich zahlreiche bekannte Layoutverfahren in einheitlicher Form als Instanzen dieses allgemeineren Modells interpretieren.

2.2 Energieminimierung

Wir haben Graphenlayout als ein Optimierungsproblem formuliert, für das gute Layouts durch Minimierung der Energiefunktion $U(x) = \sum_{C \in \mathcal{C}} U_C(x)$ über alle $x \in \mathcal{X} \cap \mathcal{R}$ erzielt werden. In dieser Allgemeinheit ist das Energieminimierungsproblem \mathcal{NP} -schwer. Es ist sogar schon dann \mathcal{NP} -schwer, wenn für einen Graphen $G = (V, E)$ gilt: $L = V$, $\mathcal{G} = G$, $\mathcal{C} = E$ und $\mathcal{X} = \{0, 1\}^L$. In diesem stark eingeschränkten Fall sind die Layoutelemente gerade die Knoten des Graphen, der Interaktionsgraph ist der Graph G selbst, es gibt keine Nebenbedingungen, Gütekriterien sind ausschließlich auf den Kanten definiert und zulässige Werte sind nur 0 und 1 [8]. Wir wollen daher in diesem Abschnitt kurz einige heuristische Optimierungstechniken für die Minimierung gebräuchlicher Energiefunktionen ansprechen.

Lokale Optimierung. Lokale Kriterien für die Güte eines Layouts sind wegen ihrer Widersprüchlichkeit selten gleichzeitig sicherzustellen. Darüber hinaus ist die tatsächliche Güte eines Layouts und erst recht einer Graphzeichnung schlecht formalisierbar, da es sich meist weniger um einen technischen als vielmehr um einen intuitiven oder sogar subjektiven Gütebegriff handelt. Insofern ist Optimalität meist auch gar nicht unbedingt erforderlich. In vielen Fällen ist lokale Optimalität bereits zufriedenstellend. Diese kann dann durch lokale Suchmethoden wie die iterative Verbesserung eines Layouts durch Änderung der Werte einzelner Layoutelemente erreicht werden.

Die bedingten Wahrscheinlichkeiten $P(X_\lambda = x_\lambda \mid X_{L-\lambda} = x_{L-\lambda})$ beschreiben die Effektivität der Zuordnung des Wertes x_λ zu Layoutelement λ , wenn die Werte aller anderen Layoutelemente $\lambda' \in L - \lambda$ auf $x_{\lambda'}$ festgelegt sind. Man rechnet leicht nach, daß diese Wahrscheinlichkeiten nur von den Nachbarn von λ abhängen. Den optimalen Wert x_λ unter der Voraussetzung, daß alle anderen Werte gegeben sind, erhält man dann durch Maximierung dieser Wahrscheinlichkeit, was äquivalent ist zur Minimierung der *lokalen Energie* $\sum_{C \in \mathcal{C}: \lambda \in C} U_C(x)$ über \mathcal{X}_λ . Wiederholte Minimierung der lokalen Energie für jedes $\lambda \in L$ liefert schließlich ein lokales Minimum der Energie eines Layoutmodells ohne Nebenbedingungen. In der Bildverarbeitung wurde diese Methode erstmals in [3] verwendet. Im Zusammenhang mit Graphenzeichnen verwenden der

Spring Embedder und viele seiner Varianten diese Vorgehensweise [19, 28, 39].

Annealing. Ein allgemeiner Lösungsansatz für große kombinatorische Optimierungsprobleme, mit dem unbefriedigende lokale Minima vermieden werden sollen, ist *Simulated Annealing* [30, 33, 44]. Die Grundidee des Algorithmus ist die iterative Modifikation eines Layoutkandidaten, wobei in jeder Iteration das neue Layout entweder als neuer Kandidat akzeptiert oder verworfen wird. Es wird in jedem Fall akzeptiert, wenn seine Energie kleiner als die Energie des aktuellen Kandidaten ist. Um lokale Minima zu vermeiden, werden jedoch auch Kandidaten mit höherer Energie mit der Wahrscheinlichkeit $e^{-\Delta U/T}$ akzeptiert, wobei ΔU die aktuelle Energiedifferenz und $T > 0$ der Temperaturparameter des Annealing ist. Konvergenz wird durch allmähliche Verringerung von T bewirkt.

Simulated Annealing ist beim Graphenzeichnen beispielsweise in [15] und [14] benutzt worden. Hauptkritikpunkt ist selbstverständlich auch hier die mangelnde Effizienz [5]. Schnellere Algorithmen, die auf Gradientenverfahren beruhen [19, 28, 24, 23, 39] können nur auf geeignete Energiefunktionen angewendet werden, entsprechen aber interessanterweise gerade dem Fall $T = 0$ des Annealings.

2.3 Kräftebasierte Layoutverfahren

Das meistgenannte kräftebasierte Layoutverfahren im Zusammenhang mit Graphenzeichnen ist der von Peter Eades eingeführte *Spring Embedder* [19], wobei jedoch zu bemerken ist, daß ähnliche kräftebasierte Verfahren auch von anderen Autoren beschrieben wurden [35, 43, 21, 37, 32]. Diese Verfahren liefern eine geradlinige Repräsentation allgemeiner ungerichteter Graphen. Ihnen gemeinsam ist die zugrundeliegende Analogie zu physikalischen Modellen. Knoten und Kanten des Graphen werden als physikalische Objekte angesehen, welche wechselseitig Kräfte ausüben, so daß das Gesamtsystem einem Gleichgewichtszustand mit niedriger innerer Energie entgegenstrebt. Während sich Knoten im allgemeinen abstoßen, wirkt zwischen adjazenten Paaren eine zusätzliche Anziehungskraft.

Gleichgewichtszustände eines solchen Kräftesystems scheinen eine Reihe naheliegender Kriterien für ästhetisch ansprechendes Layout gleichzeitig zu erfüllen, so z.B. einheitliche Kantenlängen, Symmetrie und räumliche Gruppierung dichter Teilgraphen. Da die Verfahren einfach zu verstehen und relativ leicht zu programmieren sind, gleichwohl aber zufriedenstellende Ergebnisse liefern [18, S. 304], sind sie sehr verbreitet und werden in vielfältigen Varianten eingesetzt, die sich vor allem in der Definition der Kräfte und in den angewandten Optimierungsheuristiken unterscheiden. Einerseits gibt es Ansätze, die tatsächlich Kräfte einführen, andererseits solche, die den Konfigurationen jeweils eine Potentialenergie zuordnen. Während erstere nur implizit ein Zufallsfeld-Layoutmodell definieren, fügen sich letztere direkt in unsere Modellierung des Layouts ein.

Beim Spring Embedder etwa ist die Zielfunktion nicht explizit angegeben. Es gibt lediglich zwei implizite Kriterien: Knoten, die nicht durch eine Kante verbunden sind, sollen nicht zu nahe bei-

einander liegen, während zwischen miteinander verbundenen Knoten ein gewisser vorgegebener Abstand bestehen soll. Dies sind lokale Kriterien, die sich auf jeweils nur zwei Layoutelemente beziehen und daher leicht nachgebildet werden können. Varianten des Spring Embedders sind [24, 23]. Ferner existieren Erweiterungen auf dreidimensionale geradlinige Repräsentationen [12] und auf Graphen, die sowohl ungerichtete wie gerichtete Kanten enthalten [39]. Ansätze mit explizit angegebenen Zielfunktionen findet man in [28, 15, 42] und in [14] für dreidimensionale Layouts.

Als Beispiel soll uns jedoch das bereits angesprochene Layoutverfahren von Tutte [43] dienen. Zu einem gegebenen Graphen $G = (V, E)$ wird für geradlinige Repräsentation die denkbar einfache Zielfunktion

$$U(x) = \sum_{\{u,v\} \in E} d(x_u, x_v)^2$$

optimiert, wobei $d(x_u, x_v)$ den Euklidischen Abstand der beiden Knotenpositionen bezeichne. Dies entspricht einer Anziehung adjazenter Knoten und führt zu Layouts, die alle Knoten auf denselben Punkt legen. Man beachte jedoch, daß lokale Minima der Energie für alle Knoten v

$$x_v = \frac{1}{d_G(v)} \sum_{u: \{u,v\} \in E} x_u$$

erfüllt. Gibt man nun einige Knotenpositionen fest vor, so ist das zugehörige modifizierte Gleichungssystem für zusammenhängende Graphen immer eindeutig lösbar. Wenn der gegebene Graph sogar dreifach knotenzusammenhängend und planar ist, erhält man unter der verfeinerten Nebenbedingung, daß die Knoten einer Facette seiner eindeutigen planaren Einbettung auf den Ecken eines konvexen Polygons liegen, ein eindeutiges optimales Layout mit folgenden Eigenschaften (siehe Bild 3):

- keine zwei Kanten kreuzen sich
- jeder Knoten liegt im Schwerpunkt seiner Nachbarn
- alle Flächen sind konvex

3 Anwendungen

Aus der reichhaltigen Fülle von Anwendungen des Graphenzeichnens wollen wir in diesem Abschnitt zwei Gebiete beispielhaft herausgreifen, einerseits die *Analyse Sozialer Netzwerke*, andererseits *Verkehrsplanung und -optimierung*. Zwar erscheinen diese Anwendungsbereiche sehr verschieden, in beiden treten jedoch sehr viele typische Fragestellungen und Probleme des Graphen-Layout auf. Bei der Problematik aus der Analyse Sozialer Netzwerke, die wir hier behandeln wollen, geht es tatsächlich um den Entwurf einer geradlinigen Darstellung eines Graphen unter gewissen Nebenbedingungen und Optimalitätskriterien, d.h. sie ist unmittelbar als

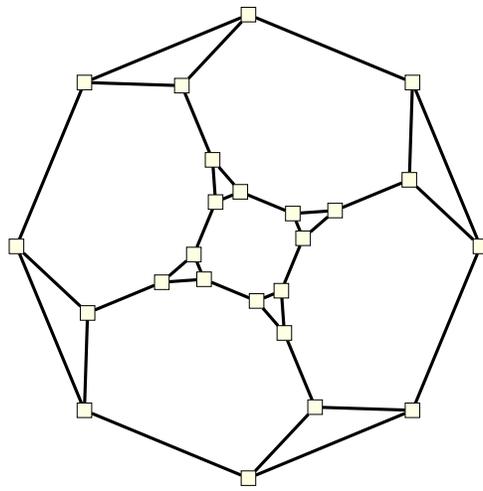


Bild 3: Cube-Connected Cycles

eine Aufgabenstellung des Graphenzeichnens gegeben. Hingegen ist unser Beispiel aus dem Verkehrsbereich nur implizit als Graphlayoutproblem erkennbar. Umso bemerkenswerter ist es, daß in beiden die in Abschnitt 2 eingeführten Methoden des Graphenlayout erfolgreich angewendet werden können, was nicht zuletzt die generelle Anwendbarkeit des Formalismus unterstreicht. Wir werden den Erfolg der gewählten Methode in beiden Fällen an Beispielen, die auf „echten“ Daten beruhen, belegen.

3.1 Analyse Sozialer Netzwerke

In Zusammenarbeit mit Wissenschaftlern der Fakultät für Verwaltungswissenschaft der Universität Konstanz beschäftigen wir uns mit Modellen für das Layout von Graphen, die bei der Analyse Sozialer Netzwerke in den Sozialwissenschaften auftreten. Visualisierung als Analysegrundlage ist in diesem Zusammenhang ein sehr junges und weitgehend unerforschtes Gebiet. So ist die Frage der Effektivität und Qualität von Visualisierungen Sozialer Netzwerke noch unklar. Dies legt die Verwendung eines so flexiblen Ansatzes wie des in Abschnitt 2.1 beschrieben nahe, um mit verschiedenen Kriterien experimentieren zu können.

3.1.1 Hintergrund

Bei der Analyse Sozialer Netzwerke geht es um die Beziehungen zwischen sozialen Akteuren. Dies können z.B. affektive, ökonomische, politische oder organisatorische Beziehungen sein. Die Theorie der Sozialen Netzwerke fußt auf der Annahme, daß strukturelle Eigenschaften des Beziehungsgeflechts für die Handlungsweise der Akteure mitbestimmend sind, und geht wesentlich auf [2] und [13] zurück. Das generelle Ziel ist die Beschreibung der Akteursstruktur sowie die Analyse von Beziehungen zwischen Akteuren und von Konstellationen, in denen sich Gruppen von Akteuren befinden. Grundlage der formalen Analyse sind daher die Graphen der

jeweiligen Beziehungen. Dazu werden Werte von Strukturvariablen der Graphen berechnet, mit deren Hilfe Hypothesen darüber belegt werden sollen, wie die Beziehungsstruktur in bestimmte Entscheidungen eingegangen ist oder andere Konsequenzen beeinflußt hat.

Innerhalb des weiten Spektrums der Analysemethoden [38, 45] lassen sich zwei Arten struktureller Analyse unterscheiden. Einerseits die detaillierte Beschreibung, ob und wie verschiedene Akteure eines Netzwerkes miteinander direkt oder indirekt verbunden sind. Andererseits kann die Analyse vornehmlich auf den Vergleich von Akteursprofilen abzielen. Im Detail werden diese Analysen auf drei Ebenen ausgeführt: der Akteursebene, der Gruppenebene und der Ebene des Gesamtnetzwerkes. Auf der Akteursebene interessiert beispielsweise, wie zentral oder periphär ein Akteur bezüglich der Gesamtstruktur ist. Graphparameter wie Zusammenhang, Pfaddistanz oder Knotengrad stellen Operationalisierungen dieses Konzepts dar. Weiterhin interessiert die Existenz von strukturell ausgezeichneten Akteuren wie z.B. Schnittknoten (sogenannten „Brokern“). Auf der Gruppenebene beschäftigt sich die Analyse insbesondere mit der Identifikation von Teilnetzwerken bestimmter Eigenschaften. Operationalisierungen hierzu sind Graphparameter wie dichte Teilgraphen, kleine Separatoren/Schnitte und Durchmesser. Auf Netzwerkebene sind allgemeine Charakteristiken wie die Dichte des Netzwerkes oder der Grad seiner Zentralisierung von Interesse.

Die Visualisierung eines Sozialen Netzwerkes kann unmittelbar auf solchen Maßen aufgebaut werden. Da meist alle drei Ebenen der Analyse gleichzeitig betrachtet werden, sind Visualisierungsmethoden erwünscht, die einen Wechsel zwischen den verschiedenen Ebenen innerhalb desselben Bildes erlauben (vgl. dazu den Abschnitt über Mikro-/Makro-Sichten in [41]). Eine Einführung in die Visualisierung Sozialer Netzwerke wird in [6] gegeben.

3.1.2 Zentralität und Zentralisierung

Einer der wichtigsten Aspekte in der Analyse Sozialer Netzwerke und ein idealer Ausgangspunkt für die experimentelle Erarbeitung von Layoutmodellen ist die Strukturvariable *Zentralität* bzw. *Zentralisierung*. Zentralität wird auf allen drei Ebenen der Analyse (*Akteurszentralität*, *Gruppenzentralität* und *Netzwerkzentralisierung*) betrachtet, und mehrere nicht-äquivalente Operationalisierungen des Zentralitätsbegriffs werden verwendet. So existieren Zentralitäts- und Zentralisierungsmaße gerichteter und ungerichteter Graphen basierend auf Beziehungsaktivität, Nähe (*closeness*) oder auch der Rolle als Mittler (*betweenness*). Wir betrachten hier vor allem Zentralität basierend auf Nähe, ausgedrückt in Pfaddistanzen.

Als Beispiel bestehe Zentralität darin, einen möglichst „direkten Draht“ zu allen anderen Akteuren zu haben. Sei dazu $G = (V, E)$ ein zusammenhängender Graph mit n Knoten. Die *Distanz-Zentralität* (*closeness centrality*) eines Knotens $v \in V$ wird definiert als die inverse Summe der Distanzen zu allen anderen Knoten

$$C_G(v) = \frac{1}{\sum_{w \in V} d_G(v, w)}$$

Hierbei bezeichne $d_G(v, w)$ die Länge eines kürzesten Weges zwischen v und w in G . Für den Vergleich über verschiedene Netzwerke hinweg ist es sinnvoll, diesen Wert zum größtmöglichen Wert innerhalb eines Netzwerkes gleicher Knotenzahl in Beziehung zu setzen. Da das jeweilige Maximum für solche Knoten angenommen wird, die zu allen anderen adjazent sind, wird die zugehörige *normalisierte Distanz-Zentralität* definiert durch

$$C'_C(v) = \frac{n-1}{\sum_{w \in V} d_G(v, w)}$$

Dies ist gerade invers zum Durchschnittsabstand des Knotens von allen andern. Betrachtet man nun die Summe der Differenzen zum größten vorkommenden normalisierten Zentralitätswert und normalisiert wieder mit dem größtmöglichen Wert, der in irgendeinem Graphen angenommen werden kann, so erhält man schließlich die *Distanz-Zentralisierung* des Graphen G . Die größtmögliche Differenz relativer Distanz-Zentralitätswerte tritt bei einem Stern auf, also gilt

$$\begin{aligned} C_C(G) &= \frac{\sum_{v \in V} \left(\max_{u \in V} C'_C(u) - C'_C(v) \right)}{\max_G \sum_{v \in V} \left(\max_{u \in V} C'_C(u) - C'_C(v) \right)} \\ &= \frac{\sum_{v \in V} \left(\max_{u \in V} C'_C(u) - C'_C(v) \right)}{(n-1) \cdot \left(\frac{n-1}{n-1} - \frac{n-1}{1+(n-2) \cdot 2} \right)} \\ &= \frac{\sum_{v \in V} \left(\max_{u \in V} C'_C(u) - C'_C(v) \right)}{(n^2 - 3n + 2)/(2n - 3)} \end{aligned}$$

Ein anderes naheliegendes Maß für Zentralität, das die Beziehungsaktivität eines Akteurs wiedergibt, basiert auf der Anzahl an Verbindungen, die dieser hat. Entsprechend kann man *Gradzentralität* (*degree centrality*) als den Grad des entsprechenden Knotens definieren. Der Anteil von kürzesten Verbindungen zwischen anderen Akteuren, auf denen ein Akteur liegt, definiert seine *Mittler-Zentralität* (*betweenness centrality*). Auch diese beiden Maße können wieder normalisiert und auf das Gesamtnetzwerk erweitert werden. Eine detailliertere Übersicht mit Erläuterungen zu einer Reihe von Zentralitätsmaßen gibt [22].

3.1.3 Zentralitätslayouts

Um in einer Netzwerkvisualisierung die Zentralitäten der einzelnen Akteure sowohl anschaulich als auch präzise darzustellen, wählen wir folgenden Ansatz. Die zulässigen Positionen eines Knotens in unserer geradlinigen Darstellung liegen sämtlich auf einem Kreis um den Bildmittelpunkt, dessen Radius durch die Zentralität des zugehörigen Akteurs bestimmt wird. Ein besonders zentraler Akteur liegt dann nah an der Mitte und umgekehrt.

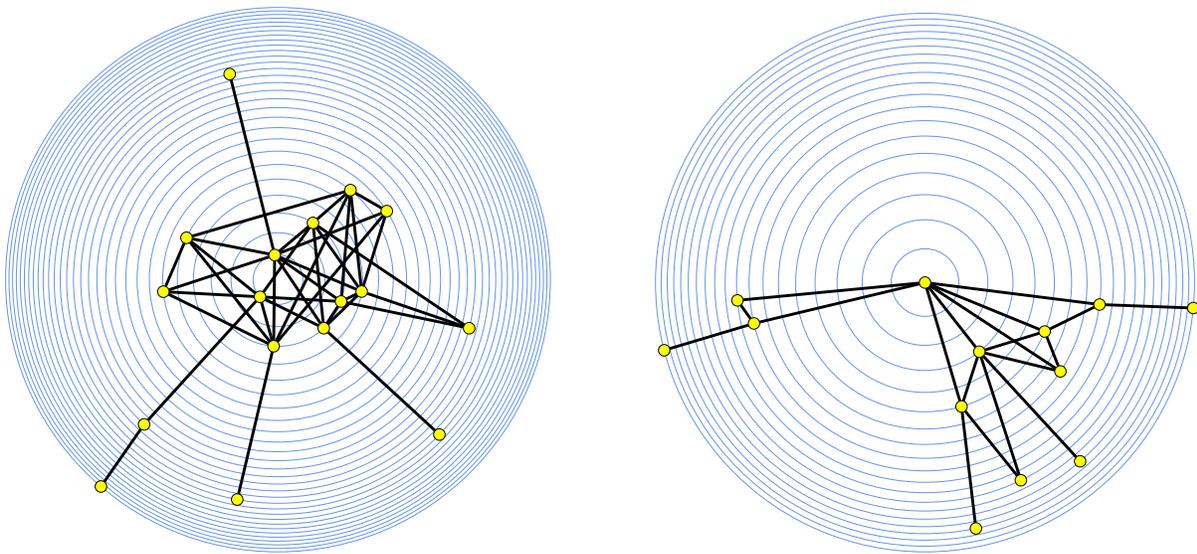


Bild 4: Anonymisierte Netzwerke der „vertraulichen Kommunikation“ lokalpolitisch engagierter Organisationen in zwei nordrhein-westfälischen Kommunen. Die eingezeichneten Kreise zeigen die mögliche Werte der normalisierten Distanz-Zentralitäten

Das formale Modell enthält als Layoutelemente eines Graphen $G = (V, E)$ wieder nur die Knoten. Die zulässigen Positionen in der Ebene werden jedoch eingeschränkt durch die einstelligen Nebenbedingungen $R_{\{v\}} = \{(x, y) : x^2 + y^2 = C'_C(v) - \max_{z \in V} C'_C(z) + c\}$, $v \in V$, wobei $c = |\{v \in V : C'_C(v) = \max_{z \in V} C'_C(z)\}| - 1$ damit in der Bildmitte nicht mehr als ein Knoten liegt. Die lokalen Gütekriterien entsprechen denen eines typischen kräftebasierten Modells:

- *Knoten/Knoten-Abstoßung*. Für eine gleichmäßige Verteilung der Knoten enthält die Zielfunktion für jedes Knotenpaar $u, v \in V$ das Potential

$$\text{Abstoßung}(x_u, x_v | c_\varrho) = \frac{c_\varrho}{d(x_u, x_v)^2}$$

Die Stärke der Abstoßung kann über den Parameter c_ϱ kontrolliert werden.

- *Knoten/Kanten-Abstoßung*. Damit Kanten nicht durch Knoten verlaufen, wird für jedes Paar einer Kante $\{u, v\} \in E$ und eines nicht-inzidenten Knotens $w \in V$ ein Potential

$$\text{Abstoßung}(u, v, w | c_\varrho) = \frac{c_\varrho}{d(\{u, v\}, w)^2}$$

hinzugefügt. Der kleinste Abstand zwischen einem Punkt auf der Strecke $\overline{x_u, x_v}$ und dem Punkt x_w ist dabei mit $d(\{u, v\}, w)$ bezeichnet.

- *Knoten/Knoten-Anziehung*. Um überlange Kanten zu verhindern, die dann insbesondere nicht radial verlaufen, definieren wir für je zwei adjazente Knoten $u, v \in V$

$$\text{Anziehung}(x_u, x_v | c_\alpha) = \alpha \cdot d(x_u, x_v)^2$$

mit konstantem Parameter c_α .

- *Kantenkreuzungen.* Kantenkreuzungen machen eine Zeichnung unübersichtlich. Wir ergänzen daher für je zwei Kanten $\{u_1, v_1\}, \{u_2, v_2\} \in E$ noch das Potential

$$\text{Kreuzungen}(u_1, v_1; u_2, v_2) = \begin{cases} 1 & \text{falls sich die Strecken } \overline{x_{u_1}, x_{v_1}} \\ & \text{und } \overline{x_{u_2}, x_{v_2}} \text{ schneiden} \\ 0 & \text{sonst} \end{cases}$$

Layouts geringer Energie haben dann gleichmäßig verteilte Knoten, die nicht auf Kanten liegen, und Kanten, die nach Möglichkeit radial und kreuzungsfrei verlaufen. Bild 4 enthält zwei Beispielnetzwerke aus den Datensätzen unserer Kooperationspartner. Deutlich zu erkennen sind die relativen Zentralitäten der Akteure in den einzelnen Netzwerken, die unterschiedliche Struktur der beiden Netzwerke und die Tatsache, daß das Layout zur rechten suboptimal ist. Weitergehende Möglichkeiten bestehen z.B. in der Verwendung der Zentralisierung $C'_C(G)$ als konstante Änderung des Mindestradius.

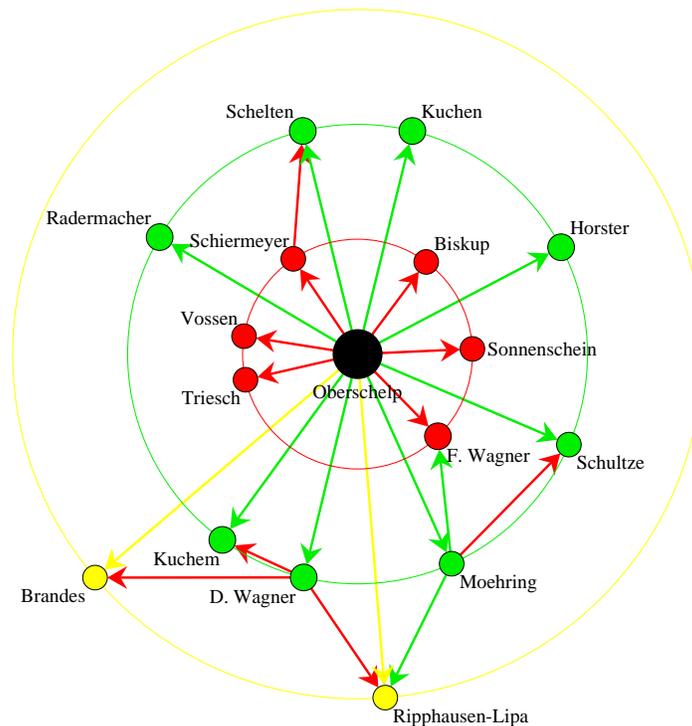


Bild 5: „Wissenschaftlicher Nachwuchs“ von Walter Oberschelp

Ein ähnliches Beispiel zeigt Bild 5. Die Akteure sind Prof. Dr. Walter Oberschelp und sein „wissenschaftlicher Nachwuchs“, wobei dazu solche Personen gerechnet wurden, die bei ihm promoviert haben und sich später habilitierten, oder bei ihm ihre Diplomarbeit geschrieben haben und bei einer bereits qualifizierten Person promovier(t)en.² Die Verteilung der Akteure auf

²Diese Menge wurde aufgrund der zur Verfügung stehenden Datenbasis so gewählt, enthält daher aber nicht ganz zufällig auch die beiden Autoren. Wir entschuldigen uns für beides. Es ist anzunehmen, daß sich bei angemessenerer Modellierung tatsächlich wesentlich mehr Personen als „wissenschaftlicher Nachwuchs“ des zentralsten Akteurs qualifizieren.

Kreise besagt hier (von innen nach außen), daß Walter Oberschelp Erstgutachter der Dissertation, Zweitgutachter der Dissertation oder Betreuer der Diplomarbeit war. Eine gerichtete Kante ist so zu lesen (von dunkel nach hell), daß der Akteur, vom dem sie ausgeht, Erstgutachter, Zweitgutachter oder Betreuer des Akteurs, auf den sie zeigt, war oder ist. Für die Lesbarkeit wurden Kriterien wie in Bild 4 verwendet.

3.2 Analyse von Fahrplandaten

Ein Problem, das in der Kooperation mit *TLC/EVA* (einer Tochter der *Deutsche Bahn AG*) aufgetreten ist, betrifft wiederum die Visualisierung als Hilfsmittel der Datenanalyse. Das grundsätzliche Ziel dieser Kooperation ist die Entwicklung von Methoden der Datenreduktion und Datenvisualisierung für die explorative Analyse großer Mengen von Fahrplandaten der europäischen öffentlichen Verkehrsbetriebe. Es handelt sich größtenteils um Bahndaten, vereinzelt sind jedoch auch Daten über Busse, Fähren oder gar Fußwege enthalten. Die Analyse der Daten betrifft Vollständigkeit, Konsistenz, Veränderungen bei Fahrplanwechseln usw. Sie dient der Qualitätssicherung bei der Fahrplanauskunft, aber auch der internationalen Koordinierung von Fahrplänen und Tarifierungsfragen. Wir werden hier eine Visualisierungsmethode für die Untersuchung eines Teilaspekts der Daten diskutieren. Das zu eigentliche Layoutproblem besteht zwar in der Bestimmung geeigneter Kurven für eine Teilmenge der Kanten, wir werden aber sehen, wie es sich auf die Bestimmung einer geradlinigen Repräsentation eines Hilfsgraphen reduzieren läßt.

Bilder 6 und 7 zeigen einen kleinen Ausschnitt der zugrundeliegenden Daten in ihrem ursprünglichen Format. Typische Datensätze sind von beachtlicher Größe, da schon ein vergleichsweise kleiner Halt wie der deutsche Hauptbahnhof von Konstanz etwa 100 ankommende und abfahrende Züge aufweist. In einem ersten Schritt der Datenanalyse werden die Daten deshalb zu einem Graphen, dem sogenannten *Fahrplangraph*, kondensiert. Trotz des unvermeidlichen Informationsverlusts wird dadurch eine den Daten zugrundeliegende Strukturierung wiedergegeben. Die Knoten dieses Graphen entsprechen allen regulären Halten von Fahrzeugen (typischerweise Bahnhöfen), und sind mit Koordinaten entsprechend der geographischen Lage der Halte gegeben. Der Fahrplangraph enthält eine Kante zwischen einem Knotenpaar, wenn es eine Fahrverbindung (etwa einen Zug) gibt, die von einem der entsprechenden Halte ohne Zwischenhalt zum anderen (oder umgekehrt) verkehrt. Der betrachtete Graph ist also ungerichtet und enthält keine Mehrfachkanten oder Schleifen. Wir werden hier der Einfachheit halber von Bahnhöfen anstelle von Halten sprechen.

Ein wichtiges Element der Datenanalyse ist die Klassifikation der im Graph der Fahrplandaten enthaltenen Kanten in zwei Kategorien: *minimale* Kanten und *transitive* Kanten. Minimale Kanten sind solche, die zu einer direkten Verbindung ohne durchfahrenen Zwischenhalt gehören. Typischerweise werden sie durch regionale Züge, die „an jedem Baum halten“, induziert. Die transitiven Kanten entsprechen Verbindungen, die gewisse Bahnhöfe ohne anzuhalten passieren, und werden typischerweise durch Schnellzüge induziert. Unser Ziel ist nun, eine graphische Darstellung zu entwerfen, die den Fahrplangraph visualisiert und dabei eine Unterscheidung der Kanten entsprechend ihrer jeweiligen Kategorie ermöglicht.

```

(...)
8000880 Radolfzell          -58.5  -510.8  (...)
(...)
8003400 Konstanz           -43.5  -519.8  (...)
8003401 Konstanz-Petersh.  -43.5  -518.2  (...)
8003416 Konstanz-Wollmat   -45.1  -517.5  (...)
(...)
8506131 Kreuzlingen        -40.2  -524.5  (...)
(...)

```

Bild 6: Ausschnitt einer Bahnhofsliste. Jeder Bahnhof hat eine eindeutige Identifikationsnummer. Die Koordinaten sind in Kilometern bezüglich Hannover angegeben

```

*z 05130 85    01
*g SE 8506131 8001790
*A VE 8506131 8001790 000000
*A G 8506131 8001790
8506131 Kreuzlingen          1112  (...)
8003400 Konstanz             1115 1125  (...)
8003401 Konstanz-Petersh.    1127 1128  (...)
8003416 Konstanz-Wollmat     1130 1130  (...)
8004997 Reichenau(Baden)     1132 1133  (...)
8002683 Hegne                1135 1135  (...)
8000496 Allensbach           1138 1138  (...)
8003872 Markelfingen         1143 1143  (...)
8000880 Radolfzell           1147 1149  (...)
8001059 Böhlingen-Rickelsh.  1152 1152  (...)
8000073 Singen(Hohentwiel)   1158 1200  (...)
8004107 Mühlhausen(b Engen)  1206 1206  (...)
8006321 Welschingen-Neuhaus. 1209 1209  (...)
8001790 Engen                1212      (...)

```

Bild 7: Fahrplanausschnitt eines Zuges. Es werden diejenigen Bahnhöfe mit Ankunfts- und Abfahrtszeiten aufgeführt, an denen der Zug hält

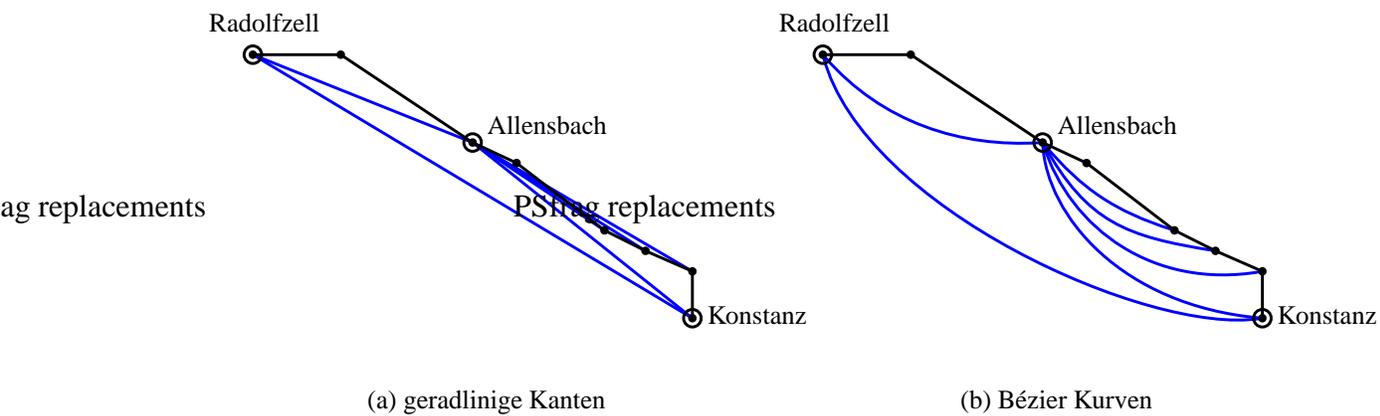


Bild 8: Verschiedene Repräsentationen transitiver Kanten

Die Klassifikation der Kanten kann natürlich leicht durch verschiedene Farben angezeigt werden. Eine naheliegende Visualisierung des Graphen besteht dann in der Positionierung der Knoten entsprechend ihrer Koordinaten und der Wiedergabe der Kanten als gerade Linien. Bild 8(a) macht jedoch schon an einem kleinen Beispiel deutlich, daß Kantenüberlappungen und kleine Winkel zwischen Kanten die Lesbarkeit einer solchen Visualisierung sehr stark beeinträchtigen. Um die geographischen Gegebenheiten zu erhalten ist ein Verschieben der Knoten nicht angebracht. Während minimale Kanten weiterhin geradlinig dargestellt werden können, weil sie typischerweise Schienenstrecken entsprechen, erscheint es günstig, transitive Kanten in geeigneter Weise als Kurven darzustellen. Hierzu bieten sich, wie in Bild 8(b) gezeigt, Bézier-Kurven an. Im folgenden formulieren wir dieses Visualisierungsproblem im Rahmen des in Abschnitt 2 eingeführten Formalismus als ein Graphlayoutproblem.

3.2.1 Layout transitiver Kanten als Kurven

Wir definieren ein Zufallsfeld-Layoutmodell für das Layout eines Fahrplangraphen $G = (V, E)$. Die Knoten von G identifizieren wir mit deren Positionen, die durch die geographische Lage der entsprechenden Bahnhöfe festgelegt sind. Die minimalen Kanten und sehr lange transitive Kanten werden als gerade Linien repräsentiert.³ Die Layoutelemente, welche positioniert werden müssen, um den Verlauf der Bézier-Kurven zu bestimmen, sind deren Stützpunkte. Sei daher $\check{E}_{\tau_1} \subseteq E$ die Menge der transitiven Kanten, deren Länge unter einem vorgegebenen Schwellwert τ_1 liegt. Die Menge der Layoutelemente $L = \{b_u(e), b_v(e) \mid e = \{u, v\} \in \check{E}_{\tau_1}\}$ besteht dann aus den jeweils zwei Stützpunkten der Kanten in \check{E}_{τ_1} . Falls zwei Stützpunkte zu derselben Kante gehören, nennen wir sie *Partner*. Den *Anker* v des Stützpunkts $b_v(e) \in L$ bezeichnen wir mit $a_{b_v(e)}$. Für alle Stützpunkte liegt die *Initialposition* auf der geraden Linie durch die Endpunkte ihrer Kanten mit gleichem Abstand von Partner und Anker.

³Bild 12 macht deutlich, warum es nicht sinnvoll ist, alle transitiven Kanten als Bézier-Kurven zu repräsentieren.

Die einem Stützpunkt b zugeordnete Position wird in Abhängigkeit von Partner, Anker, allen Stützpunkten mit demselben Anker oder mit Initialposition in unmittelbarer Nähe von b , und allen Bahnhöfen in der Nähe der Initialposition von b bestimmt. Sei $\{u, v\} \in \check{E}_{\tau_1}$ eine transitive Kante und $b \in L$ ein Stützpunkt von $\{u, v\}$. Zu zwei gegebenen Parametern ϵ_1 und ϵ_2 betrachten wir die Ellipse, deren Hauptachse durch u und v geht, und deren Radien $\epsilon_1 \cdot \frac{d(u,v)}{2}$ und $\epsilon_2 \cdot \frac{d(u,v)}{2}$ sind.⁴ Dann bezeichne \mathcal{E}_b die Menge aller Bahnhöfe und Stützpunkte außer b , deren Position bzw. Initialposition innerhalb dieser Ellipse liegen. Da die Nachbarschaft eines Layoutelements aus allen Layoutelementen besteht, die einen Einfluß auf dessen Positionierung haben, ist η_b die Vereinigung von $\mathcal{E}_b \cap L$ mit der Menge aller Stützpunkte mit demselben Anker wie b und der Menge aller Stützpunkte b' , für die $b \in \mathcal{E}_{b'}$ (da Interaktionen symmetrisch sind). Die folgenden erwünschte Eigenschaften werden durch Interaktionspotentiale modelliert.

- *Abstand zu Bahnhöfen.* Für jeden Stützpunkt $b \in L$ einer Kante $\{u, v\} \in \check{E}_{\tau_1}$ wird ein Abstoßungspotential

$$\sum_{s \in \mathcal{E}_b \cap V} \text{Abstoßung}(x_b, s \mid (\varrho_1 \cdot \lambda_b)^4)$$

definiert, wobei $\lambda_b = \frac{d(u,v)}{3}$ und ϱ_1 konstant. Diese Potentiale bewirken einen ausreichenden Abstand zu Bahnhöfen in der Umgebung b und können durch Wahl von ϱ_1 kontrolliert werden. Ein kombiniertes Abstoßungs- und Anziehungspotential

$$\begin{aligned} \text{Abstand}(x_b, a_b \mid (\lambda_1 \cdot \lambda_b)^4) &= \text{Abstoßung}(x_b, a_b \mid (\lambda_1 \cdot \lambda_b)^4) \\ &\quad + \text{Anziehung}(x_b, a_b \mid 1) \end{aligned}$$

mit λ_1 konstant, hält b genügend nahe an seinem Anker a_b .

- *Abstand von Stützpunkten in unmittelbarer Nähe.* Wie bei nahegelegenen Bahnhöfen sollte auch ein Stützpunkt $b_1 \in L$ nicht zu nahe an einem anderen Stützpunkt $b_2 \in \eta_{b_1}$ positioniert werden. Falls b_1 weder Partner von b_2 ist, noch an b_2 gebunden ist (Bündelung wird später definiert), definieren wir daher Potentiale

$$\text{Abstoßung}(x_{b_1}, x_{b_2} \mid \varrho_2^4 \cdot \min\{\lambda_{b_1}^4, \lambda_{b_2}^4\}).$$

Der gewünschte Abstand zwischen Partnern b_1 und b_2 ist identisch mit dem gewünschten Abstand von deren jeweiligem Anker

$$\text{Abstand}(x_{b_1}, x_{b_2} \mid (\lambda_1 \cdot \lambda_{b_1})^4).$$

- *Bündelung.* Es ist wünschenswert, daß Stützpunkte $b_1, b_2 \in L$, die denselben Anker haben, auch auf derselben Seite eines Pfades minimaler Kanten, der durch den Anker geht, liegen. Daher *bündeln* wir diese, falls λ_{b_1} nicht zu sehr von λ_{b_2} abweicht, d.h. falls $\frac{1}{\tau_2} < \frac{\lambda_{b_1}}{\lambda_{b_2}} < \tau_2$, für einen Schwellwert $\tau_2 \geq 1$. Dazu definieren wir Potentiale

$$\beta \cdot \text{Abstand}(x_{b_1}, x_{b_2} \mid \lambda_2^4 \cdot (\lambda_{b_1}^4 + \lambda_{b_2}^4)/2),$$

wobei λ_2 ein Dehnungsfaktor für die Länge der Anbindungskanten ist, und β die Wichtigkeit der Anbindung in Relation zu den anderen Potentialen kontrolliert.

⁴Für unsere Beispiele wurden $\epsilon_1 = 1.1$ und $\epsilon_2 = 0.5$ gewählt.

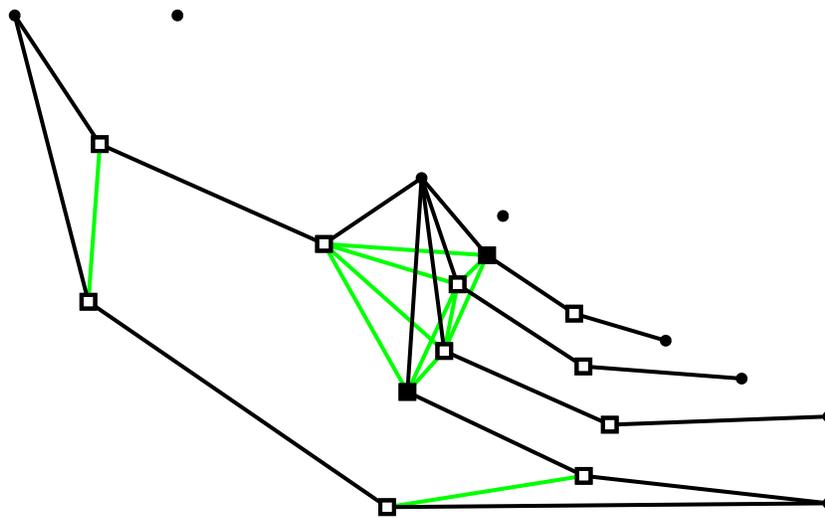


Bild 9: Induzierter Hilfsgraph für das Beispiel aus Bild 8(b). Die durch schwarze Vierecke gekennzeichneten Layoutelemente sind nicht gebündelt, da sich die Abstände zu ihrem Anker zu sehr unterscheiden (Schwellwert τ_2)

Insgesamt besteht die Zielfunktion also ausschließlich aus Abstoßungs- und Anziehungspotentialen, die ein Layoutmodell für einen imaginären Hilfsgraphen definieren: die Bahnhöfe entsprechen Knoten mit festgelegter Position, wohingegen die Stützpunkte zu positionierenden Knoten entsprechen. Kanten gibt es im Hilfsgraph jeweils zwischen einem Stützpunkt und seinem Anker, zwischen Partnern und zwischen gebündelten Stützpunkten nicht zu verschiedener Ziellängen. Bild 9 zeigt einen solchen Hilfsgraphen.

Die hier beschriebene Zielfunktion wurde aufgrund von experimentellen Studien mit unterschiedlichen Potentialen und Parametern aufgestellt. Zu Beginn haben wir eine einfache Kombination von Abstoßungen von Bahnhöfen und Anziehungen und Abstoßungen von Partnern und Ankern betrachtet. Zunächst haben wir Splines als Repräsentation der transitiven Kanten verwendet, weil uns diese besonders gut kontrollierbar erschienen. Immerhin verlaufen sie tatsächlich durch ihre Stützpunkte. Allerdings zeigte sich schnell, daß die Kurven zwischen den mittleren Stützpunkten zu weit in die Layoutregion hineinreichten. Die durchaus vielversprechenden ersten Resultate unserer Experimente mit Bézier-Kurven haben uns veranlaßt, ausführlichere experimentelle Studien zur Aufstellung ausgefeilterer Zielfunktionen und einer sorgfältigeren Wahl des Parametervektors $\theta = (\varrho_1, \varrho_2, \lambda_1, \lambda_2, \beta, \tau_1, \tau_2)$ vorzunehmen (siehe dazu auch [10]). Größere Fahrplangraphen sind in den Bildern 10, 11 und 12 gezeigt.

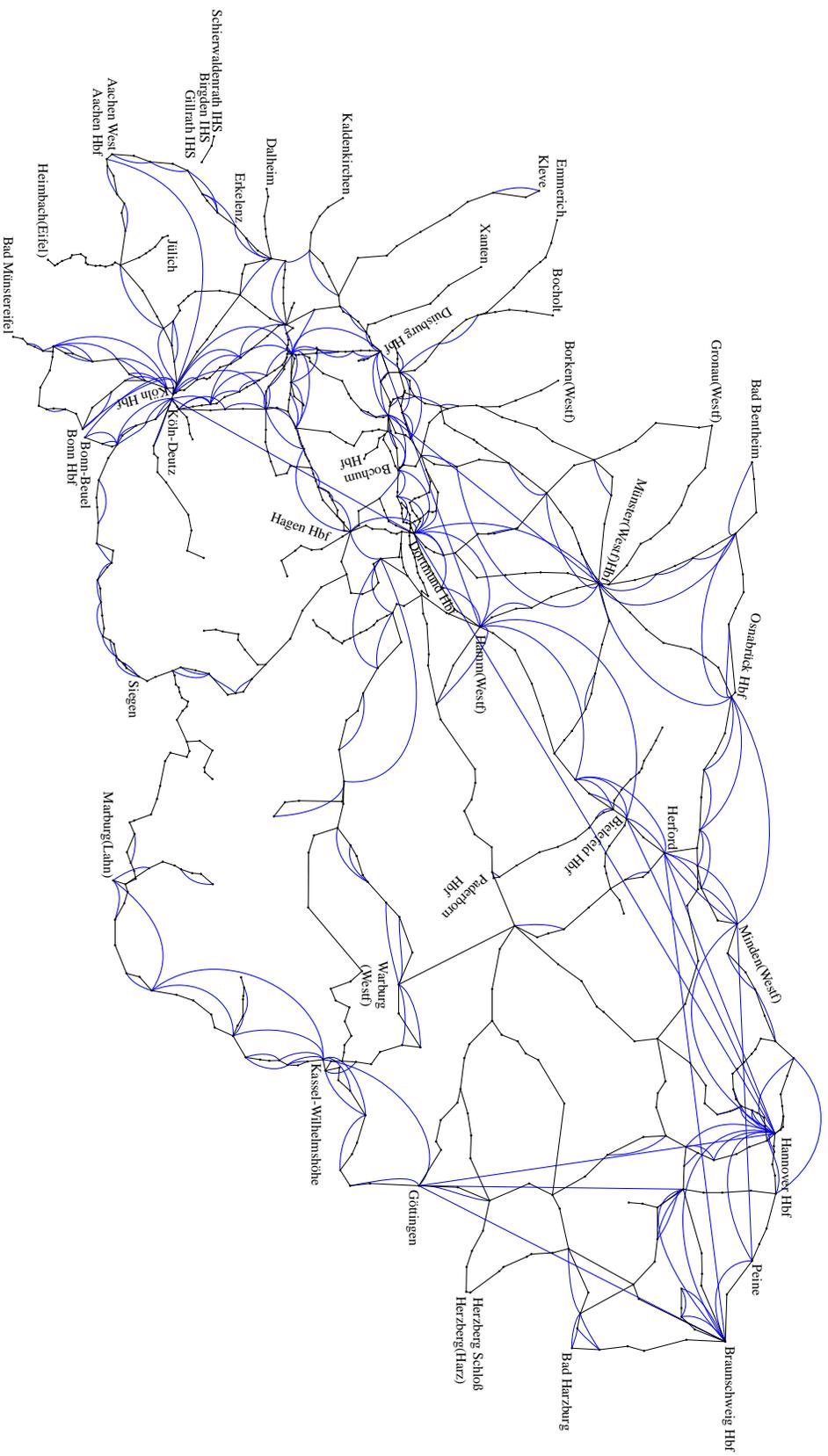


Bild 10: Fahrplangraph des Sommerfahrplans 1997 der Deutschen Bahn, begrenzt auf eine fast beliebig gewählte Region Deutschlands. 882 Knoten, 1 199 Kanten (davon 260 transitiv). $\theta \equiv (0.7, 0.3, 0.7, 0.5, 0.4, 70, 3)$

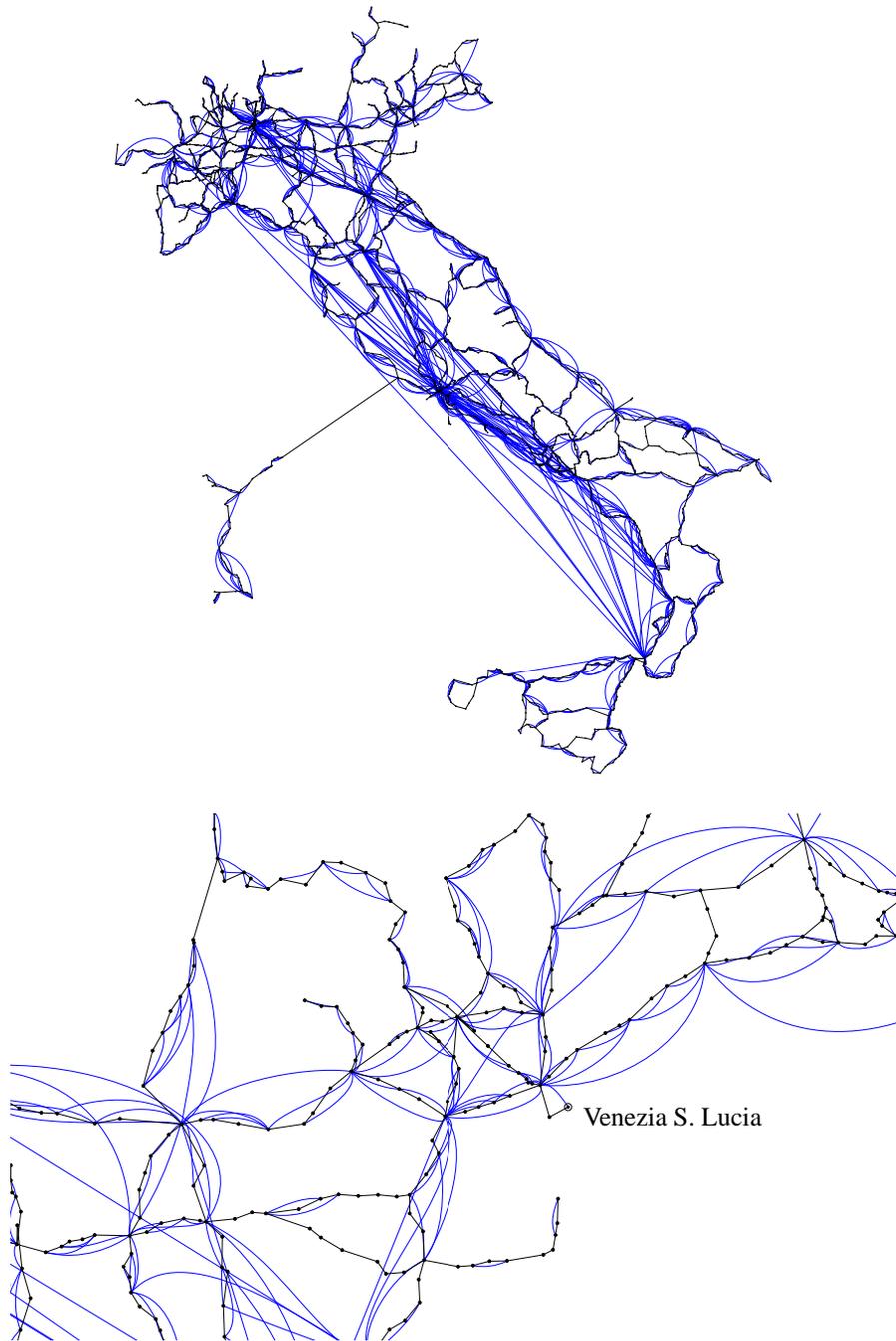


Bild 11: Zug- und Fährverbindungen in Italien und ein vergrößerter Ausschnitt. 2 386 Knoten, 4 370 Kanten (davon 1 849 transitiv), $\theta = (0.7, 0.3, 0.7, 0.5, 0.4, 100, 3)$

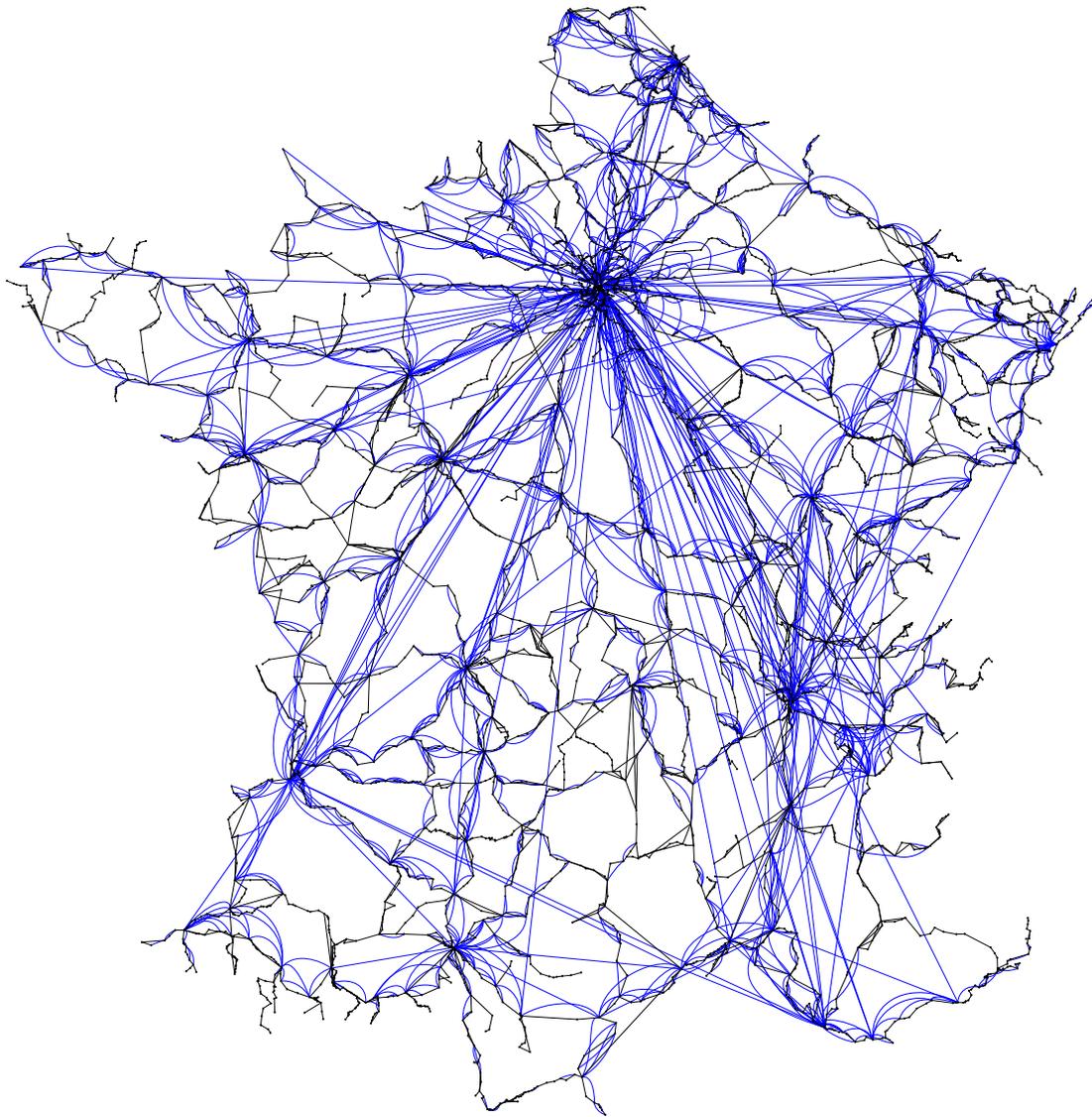


Bild 12: Zugverbindungen in Frankreich. 4 551 Knoten, 7 793 Kanten (davon 2 408 transitiv),
 $\theta = (0.7, 0.3, 0.7, 0.5, 0.4, 100, 3)$

Literatur

- [1] E. H. L. Aarts und J. H. M. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, 1989.
- [2] S. D. Berkowitz. *An Introduction to Structural Analysis: The Network Approach to Social Research*. Butterworths, 1982.
- [3] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.
- [4] F. J. Brandenburg (ed.). *Proceedings of the 3rd International Symposium on Graph Drawing (GD '95), Lecture Notes in Computer Science*, Nr. 1027. Springer, 1996.
- [5] F. J. Brandenburg, M. Himsolt und C. Rohrer. An experimental comparison of force-directed and randomized graph drawing algorithms. In [4], Seiten 76–87.
- [6] U. Brandes, P. Kenis, J. Raab, V. Schneider und D. Wagner. Explorations into the visualization of policy networks. Erscheint in *Journal of Theoretical Politics*, 11(1), 1999.
- [7] U. Brandes und D. Wagner. A Bayesian paradigm for dynamic graph layout. In [16], Seiten 236–247.
- [8] U. Brandes und D. Wagner. Random field models for graph layout. *Konstanzer Schriften in Mathematik und Informatik*, Nr. 33. Universität Konstanz, April 1997.
- [9] U. Brandes und D. Wagner. Dynamic grid embedding with few bends and changes. In K.-Y. Chwa und O. H. Ibarra (eds.), *Proceedings of the 9th Annual International Symposium on Algorithms and Computation (ISAAC '98), Lecture Notes in Computer Science*, Nr. 1533, Seiten 89–98. Springer, 1998.
- [10] U. Brandes und D. Wagner. Using graph layout to visualize train interconnection data. Erscheint in [47].
- [11] G. R. Brightwell und E. R. Scheinerman. Representations of planar graphs. *SIAM Journal on Discrete Mathematics*, 6(2):214–229, Mai 1993.
- [12] I. Bruß und A. Frick. Fast interactive 3-D graph visualization. In [4], Seiten 99–110.
- [13] R. S. Burt. *Toward a Structural Theory of Action: Network Models of Social Structure, Perception, and Action*. Academic Press, 1982.
- [14] I. F. Cruz und J. P. Twarog. 3D graph drawing with simulated annealing. In [4], Seiten 162–165.
- [15] R. Davidson und D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996.
- [16] G. Di Battista (ed.). *Proceedings of the 5th International Symposium on Graph Drawing (GD '97), Lecture Notes in Computer Science*, Nr. 1353. Springer, 1997.

- [17] G. Di Battista, P. Eades, H. de Fraysseix, P. Rosenstiehl und R. Tamassia, (eds.). *Proceedings of the ALCOM International Workshop on Graph Drawing (GD '93)*, 1993. ftp.cs.brown.edu/pub/gd94/gd-92-93/gd93-v2.ps.Z.
- [18] G. Di Battista, P. Eades, R. Tamassia und I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
- [19] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.
- [20] P. Eades, J. Marks und S. C. North. Graph-drawing contest report. In [16], Seiten 438–445.
- [21] C. J. Fisk, D. L. Caskey und L. E. West. ACCEL: Automated circuit card etching layout. *Proceedings of the IEEE*, 55(11):1971–1982, November 1967.
- [22] L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.
- [23] A. Frick, A. Ludwig und H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In [40], Seiten 388–403.
- [24] T. M. J. Fruchterman und E. M. Reingold. Graph-drawing by force-directed placement. *Software—Practice and Experience*, 21(11):1129–1164, 1991.
- [25] A. Garg. On drawing angle graphs. In [40], Seiten 84–95.
- [26] M. R. Garey und D. S. Johnson. Crossing Number is \mathcal{NP} -Complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983.
- [27] D. S. Johnson. The \mathcal{NP} -completeness column: An ongoing guide. *Journal of Algorithms*, 3:89–99, 1982.
- [28] T. Kamada und S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1989.
- [29] T. Kamps, J. Kleinz und J. Read. Constraint-based spring-model algorithms for graph layout. In [4], Seiten 349–360.
- [30] S. Kirkpatrick, C. D. Gelatt und M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, Mai 1983.
- [31] Donald E. Knuth. Computer Drawn Flowcharts. *Communications of the ACM*, 6(9):555–563, 1963
- [32] J. A. McDonald und J. Pedersen. Geometric abstractions for constrained optimization of layouts. In A. Buja und P. A. Tukey (eds.), *Computing and Graphics in Statistics, The IMA Volumes in Mathematics and its Applications*, Nr. 36, pages 95–105. Springer, 1991.
- [33] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller und E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, Juni 1953.

- [34] S. C. North (ed.). *Proceedings of the 4th International Symposium on Graph Drawing (GD '96)*, *Lecture Notes in Computer Science*, Nr. 1190. Springer, 1996.
- [35] W. D. Niven (ed.). *The Scientific Papers of James Clark Maxwell*, Dover Publications, New York, 1965.
- [36] J. F. Padgett und C. K. Ansell. Robust action and the rise of the Medici, 1400-1434. *American Journal of Sociology* 98:1259–1319, 1993.
- [37] N. R. Quinn und M. A. Breuer. A force directed component placement procedure for printed circuit boards. *IEEE Transactions on Circuits and Systems*, 26(6):377–388, 1979.
- [38] J. Scott. *Social Network Analysis: A Handbook*. Sage Publications, 1991.
- [39] K. Sugiyama und K. Misue. A simple and unified method for drawing graphs: Magnetic-spring algorithm. In [40], Seiten 364–375.
- [40] R. Tamassia und I. G. Tollis (eds.). *Proceedings of the 2nd International Symposium on Graph Drawing (GD '94)*, *Lecture Notes in Computer Science*, Nr. 894. Springer, 1995.
- [41] E. R. Tufte. *Envisioning Information*. Graphics Press, 1990.
- [42] D. Tunkelang. A practical approach to drawing undirected graphs. Technical Report CMU-CS-94-161, School of Computer Science, Carnegie Mellon University, Juni 1994.
- [43] W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society, Third Series*, 13:743–768, 1963.
- [44] P. J. M. van Laarhoven und E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Reidel, 1988.
- [45] S. Wasserman und K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- [46] R. Webber. *Finding the Best Viewpoint for Three-Dimensional Graph Drawings*. PhD thesis, University of Newcastle, 1998.
- [47] S. H. Whitesides (ed.). *Proceedings of the 6th International Symposium on Graph Drawing (GD '98)*, *Lecture Notes in Computer Science*, Nr. 1547. Springer, 1998.
- [48] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods, Applications of Mathematics*, Nr. 27. Springer, 1995.
- [49] D. Wood. On higher-dimensional orthogonal graph drawing. Technical Report 96/286, Department of Computer Science, Monash University, November 1996.