Hochschule Mittweida (FH)
University of Applied Sciences

Fachbereich Mathematik/Physik/Informatik

Martin Trinks

Diplomarbeit

Clusterings für Graphen mit stochastisch unabhängig existierenden Kanten

Mittweida, 2009

Erstprüfer: Prof. Dr. Peter Tittmann (Hochschule Mittweida)
Zweitprüfer: Prof. Dr. Eckhard Manthei (Hochschule Mittweida)

Danksagung

Für die Unterstützung und die guten Arbeitsbedingungen während der Erstellung dieser Arbeit bedanke ich mich bei der Hochschule Mittweida und vor allem bei meinem Betreuer Prof. Peter Tittmann recht herzlich.

Für die Möglichkeit über ein Jahr lang an der gestellten Fragestellung arbeiten zu können und dabei auch die eine oder andere Woche über Details nachdenken zu dürfen, bin ich sehr dankbar. Auch, weil sie meinen Entschluss gefördert hat weiter in dieser Richtung tätig zu sein.

Ein großes Dankeschön geht an meinen Eltern, die mir mit ihrer finanziellen Unterstützung nicht nur das Studium, sondern auch die Konzentration darauf ermöglichst haben.

谢谢你们。

Meiner Freundin Stefanie danke ich für die dauerhafte Unterstützung und Motivation. Ich habe die lange Zeit, in der wir gemeinsam unsere (also jeder seine) Diplomarbeiten geschrieben haben, sehr genossen - vielleicht hat es auch gerade deswegen so lange gedauert.

我爱你。

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

1 Einleitung

Der Titel dieser Arbeit verbindet zwei Objekte der Mathematik, die jeder für sich Inhalt zahlreicher Arbeiten sind, deren „Zusammenspiel“ jedoch noch wenig untersucht wurde: *Clusterings* und *Graphen mit stochastisch unabhängig existierenden Kanten*.

Clusterings spielen vor allem beim Umgang mit großen Datenmenge eine entscheidende Rolle. Durch die Entwicklung der Computertechnik haben sich die Schwierigkeiten von der Datenerfassung hin zur Datenauswertung verschoben. Unabhängig davon, aus welchem Anwendungsgebiet die Daten stammen, ist zunächst eine Einteilung der Datensätze in Gruppen sinnvoll. Dabei sollen ähnliche Datensätze in der gleichen Gruppe liegen und sich Datensätze aus verschiedenen Gruppen unterscheiden. Genau das ist es, was man unter einem Clustering versteht.

Graphen mit stochastisch unabhängig existierenden Kanten sind als Modell für Zuverlässigkeitsprobleme von Bedeutung. Eines der ersten Probleme der Graphentheorie war das *Königsberger Brückenproblem*, bei dem ein Weg gesucht wurde, der genau einmal über alle Brücken der Stadt führt und am Ausgangspunkt endet. Geht man dagegen davon aus, dass die Brücken zum Beispiel aufgrund von Bauarbeiten nur mit einer gewissen Wahrscheinlichkeit passierbar sind und fragt, mit welcher Wahrscheinlichkeit es dann möglich ist von einer Flussseite oder Insel zu einer anderen zu gelangen, so wird dies mit einem Graphen mit stochastisch unabhängig existierenden Kanten modelliert.

Dieses Beispiel lässt sich auf viele Anwendungsgebiete, darunter elektrische Systeme und soziale Netzwerke, übertragen. Gemeinsam ist allen eine Menge von Kreuzungen (die Knoten des Graphen), die durch eine Menge von Verbindungen (die Kanten des Graphen) verbunden sind, wobei die Verbindungen nur mit einer gegebenen Wahrscheinlichkeit nutzbar sind.

Ein *Clustering eines Graphen* ist eine Einteilung der Knoten in Gruppen entsprechend der verbindenden Kanten. Für diese Art von Clusterings wurden zahlreiche verschiedene Algorithmen beschrieben, allerdings geht kein Algorithmus auf die besondere Situation ein, dass die Kanten nur mit gewissen Wahrscheinlichkeiten existieren.

Eine Vielzahl von Algorithmen wurde auch für *Graphen mit stochastisch unabhängig existierenden Kanten* entwickelt. Deren Aufgabe beschränkt sich jedoch auf die Ermittlung stochastischer Größen wie der Zusammenhangswahrscheinlichkeit, d. h. der Wahrscheinlichkeit, dass jeder Knoten von jedem anderen Knoten aus erreichbar ist. Aus diesen Ergebnissen werden aber keine Clusterings abgeleitet.

Die vorliegende Arbeit versucht die Lücke zwischen diesen beiden mathematischen Objekten zu verringern und stellt einen Algorithmus zur Bestimmung von Clusterings für Graphen mit stochastisch unabhängig existierenden Kanten vor.

Im zweiten Kapitel werden zunächst die benötigten Grundlagen vorgestellt. Dabei wird neben der Graphentheorie auch auf Clusterings und die Monte-Carlo-Simulation zur näherungsweisen Berechnung von Summen eingegangen.

Im dritten Kapitel wird die Definition eines Graphen mit stochastisch unabhängig existierenden Kanten gegeben. Es werden verschiedene Möglichkeiten zur Berechnung der Zusammenhangswahrscheinlickeit und ein Qualitätsmaß für Clusterings dieser Klasse von Graphen vorgestellt.

Im vierten Kapitel folgt die Beschreibung des entwickelten Clusteringalgorithmus. Dabei werden alle Teilalgorithmen und Berechnungen angegeben, die für einen „Nachbau“ des Algorithmus notwendig sind.

Im fünften Kapitel werden die Eigenschaften des Algorithmus analysiert. Es wird gezeigt, dass der Algorithmus für jedes Clustering eine Schätzung der sogenannten internen Zusammenhangswahrscheinlichkeit liefert.

Im sechsten Kapitel wird der Algorithmus an einem Testgraphen gezeigt und es werden Einflussgrößen des Algorithmus untersucht.

Das siebente Kapitel fasst die Eigenschaften des vorgestellten Qualitätsmaßes und Algorithmus zusammen und nennt Möglichkeiten für weiterführende Untersuchungen.

2 Grundlagen

Im diesem Kapitel werden die verwendeten theoretischen Grundlagen kurz dargelegt. Dabei werden alle im Weiteren verwendeten Begriffe vorgestellt, auf nicht direkt benötigte Eigenschaften und Zusammenhänge wird jedoch nicht eingegangen. Diese können der angegebenen Literatur entnommen werden.

2.1 Graphen

Zur Graphentheorie existiert eine Vielzahl von einführenden Büchern in deutscher Sprache, darunter sind [1] und [26] zu empfehlen. Ersteres behandelt auch die geschichtliche Entwicklung der Graphentheorie aus dem 4-Farben-Problem. In englischer Sprache bietet [5, Seiten 7 - 15] eine gute und kurze Einführung.

2.1.1 Definitionen und Begriffe

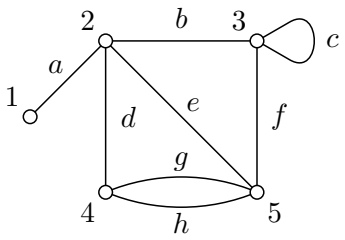
Ein *Graph* $G = (V, E)$ ist ein geordnetes Paar, bestehend aus einer Menge V von *Knoten* und einer Menge E von *Kanten*, wobei jeder Kante zwei, nicht notwendig verschiedene, Knoten zugeordnet werden. Man sagt auch, eine Kante verbindet zwei Knoten miteinander.

Die Anzahl der Knoten wird im Allgemeinen mit n, die Anzahl der Kanten mit m bezeichnet. Die Elemente der Knotenmenge V werden mit $v_1, \ldots, v_n$ und die Elemente der Kantenmenge E werden mit $e_1, \ldots, e_m$ bezeichnet.

Zwei Knoten, die durch eine Kante verbunden sind, heißen *adjazent*. Die einer Kante zugeordneten Knoten werden als die *Endknoten* der Kante bezeichnet. Eine Kante und einer ihrer Endknoten heißen *inzident* zueinander.

In einem *ungerichteten Graphen* ist die Reihenfolge der zugeordneten Knoten unerheblich und man schreibt $e = \{u, v\}$ oder $e = \{v, u\}$, wenn die Kante e die Knoten u und v miteinander verbindet. In einem *gerichteten Graphen* wird jeder Kante zusätzlich eine Richtung zugewiesen, man sagt die Kante e führt von Knoten u zu Knoten v und schreibt $e = (u, v)$.

Zur mathematischen Darstellung bzw. zur internen Darstellung von Graphen in Computerprogrammen werden oft Matrizen benutzt. Am häufigsten wird die *Adjazenzmatrix*



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 1 & 1 & 2 & 0 \end{pmatrix}$$

Abbildung 2.1: Grafische Darstellung und Adjazenzmatrix A eines ungerichteten Graphen $G = (V, E)$ mit 5 Knoten, 8 Kanten, der Knotenmenge $V = \{1, 2, 3, 4, 5\}$, der Kantenmenge $E = \{a, b, c, d, e, f, g, h\}$ und den Kanten $a = \{1, 2\}$, $b = \{2, 3\}$, $c = \{3, 3\}$, $d = \{2, 4\}$, $e = \{2, 5\}$, $f = \{3, 5\}$ sowie $g = h = \{4, 5\}$.

verwendet. Sie ist vom Format $n * n$ und ihr Eintrag an der Kreuzung der i-ten Zeile und j-ten Spalte ist k, wenn es genau k Kanten gibt, die die Knoten v_i und v_j miteinander verbinden (ungerichteter Graph) bzw. die vom Knoten v_i zum Knoten v_j führen (gerichteter Graph).

Zur bildlichen Darstellung von Graphen werden die Knoten meist durch Kreise und die Kanten durch Strecken oder Bögen dargestellt, welche die entsprechenden Endknoten miteinander verbinden. Im Fall von gerichteten Graphen werden die Kanten zusätzlich mit einem Pfeil versehen. Ein Beispiel für die beiden erwähnten Darstellungsmöglichkeiten eines Graphen ist in Abbildung 2.1 zu sehen.

Die Knotenmenge eines Graphen G wird mit $V(G)$ und die Kantenmenge mit $E(G)$ bezeichnet, wobei die Angabe des Parameters entfallen kann, wenn klar ist, um welchen Graphen es sich handelt. Die Bezeichnungen V für die Knotenmenge sowie E für die Kantenmenge stammen von den englischen Begriffen „vertex“ für Knoten und „edge“ für Kante. Für Knoten sind außerdem die Begriffe „Ecke“ und „node“, für Kanten in gerichteten Graphen außerdem „Bogen“ und „arc“ gebräuchlich.

Kanten, deren Endknoten identisch sind, werden *Schlingen* genannt. *Multigraphen* sind Graphen, deren Kantenmenge eine Multimenge ist, d. h. in denen identische Kanten vorhanden sein können. Diese werden als *parallele Kanten* bezeichnet. Ein Graph, der weder Schlingen noch parallele Kanten enthält, wird *schlichter Graph* genannt.

Der *Grad* eines Knotens v wird mit $\deg(v)$ bezeichnet und ist die Anzahl der zu v inzidenten Kanten, wobei Schlingen doppelt gezählt werden. Der *Minimalgrad* $\delta(G)$ eines Graphen G ist der minimale Grad, den ein Knoten v des Graphen G besitzt. Ein Beispiel dazu zeigt Abbildung 2.2.

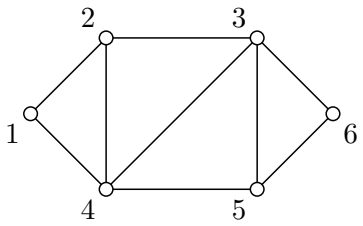


Abbildung 2.2: Ein schlichter, ungerichteter Graph G mit 6 Knoten, 10 Kanten, den Knotengraden $\deg(2) = \deg(3) = \deg(4) = \deg(5) = 3$, $\deg(1) = \deg(6) = 2$ und Minimalgrad $\delta(G) = 2$.

2.1.2 Graphen mit Kantenbewertungen

Zusätzlich zur reinen Struktur eines Graphen können sowohl den Knoten als auch den Kanten eine oder mehrere Bewertungen zugeordnet werden. Man spricht dann von einem *bewerteten Graphen* bzw., wenn dies nicht der Fall ist, von einem *unbewerteten Graphen*. Es werden ausschließlich *Kantenbewertungen* betrachtet, welche durch eine Funktion $\omega : E \to R$ realisiert werden, die jeder Kante $e \in E$ eine Bewertung $\omega(e)$ zuweist. Anstatt der Begriffe „Bewertung“ und „bewertet“ sind in der deutschen Literatur auch oft die Begriffe „Gewicht“ und „gewichtet“ gebräuchlich.

In den meisten Fällen entspricht ein unbewerteter Graph einem bewerteten Graphen, in dem jeder Kante die Bewertung 1 zugeordnet wird. Graphen mit natürlichen Zahlen als Kantenbewertungen können oft als unbewertete Multigraphen dargestellt werden, in denen jede Kante entsprechend ihrer Kantenbewertung mehrfach vorhanden ist, siehe dazu Abbildung 2.3. Dieses Vorgehen lässt sich auch auf Kantenbewertungen mit rationalen Zahlen (Verhältnisse der Kantenbewertungen bleiben erhalten) und reellen Zahlen (können durch rationale Kantenbewertungen beliebig genau approximiert werden) übertragen.

Für das Arbeiten mit bewerteten Graphen ist die Interpretation der jeweiligen Bewertung wesentlich, da von ihr die Eigenschaften der Bewertung abhängen.

Ein Beispiel für eine Eigenschaft einer Kantenbewertung, die von der Interpretation abhängig ist, ist das Verhalten bei einer Parallelreduktion, d. h. zwei parallele Kanten sollen durch eine Kante mit den gleichen Eigenschaften (hinsichtlich der Kantenbewertung) ersetzt werden. Abbildung 2.4 zeigt die Modellierung eines minimalen Netzwerkes aus zwei Servern und zwei Datenleitungen und die unterschiedlichen Ergebnisse für verschiedene Interpretationen der Kantenbewertungen.

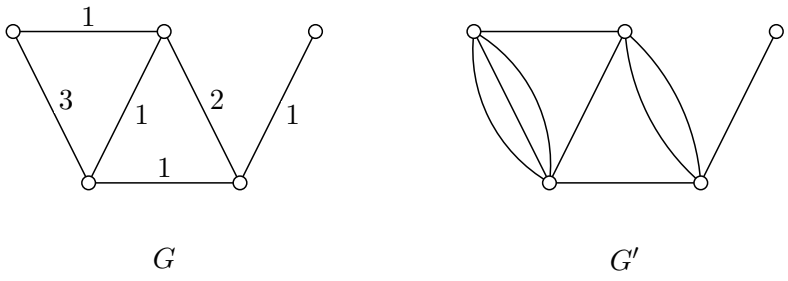


Abbildung 2.3: Die Darstellung eines mit natürlichen Zahlen bewerteten Graphen G als unbewerteten Multigraphen G'. Die Bezeichnung der Kanten des bewerteten Graphen G entspricht der jeweiligen Bewertung.

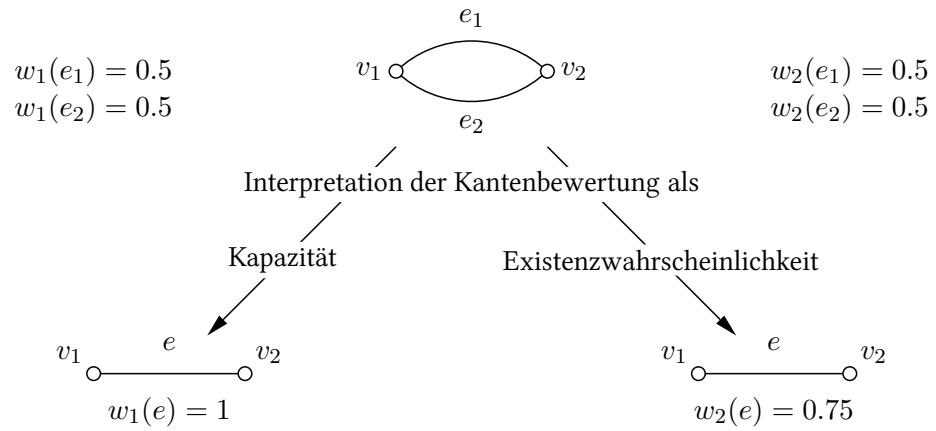


Abbildung 2.4: Gegeben sei der Graph $G = (\{v_1, v_2\}, \{e_1, e_2\})$ als Modell für ein Netzwerk aus zwei Servern (Knoten) und zwei Datenleitungen (Kanten). Die Kantenbewertung w_1 ordne den Kanten die Datenübertragungskapazität in GB/s zu und die Kantenbewertung w_2 ordne den Kanten die Wahrscheinlichkeit zu, dass die Datenleitung funktionstüchtig ist, im Beispiel jeweils 0.5. Berechnet werden sollen die Datenübertragungskapazität und die Wahrscheinlichkeit für eine funktionierende Kommunikation zwischen beiden Servern. Aus Sicht der Graphentheorie handelt es sich um eine Parallelreduktion: zwei parallele Kanten sollen durch eine einzige Kante mit gleichen Eigenschaften ersetzt werden. Die Datenübertragungsraten werden addiert, d. h. die Kantenbewertung w_1 der „gemeinsamen“ Kante e ist 1. Für w_2 wird die Wahrscheinlichkeit berechnet, dass mindestens eine Kante existiert, diese beträgt 0.75.

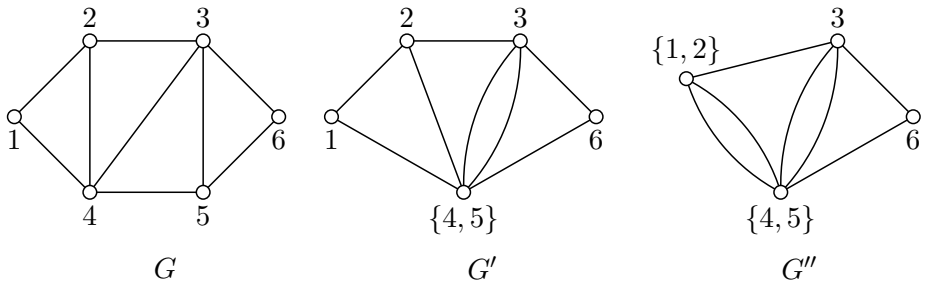


Abbildung 2.5: Drei Graphen G, G' und G'', wobei G' durch Kontraktion der Kante $\{4, 5\}$ im Graphen G hervorgeht, d. h. $G' = G \setminus \{4, 5\}$ und G'' durch Kontraktion der Kante $\{1, 2\}$ im Graphen G' entsteht, d. h. $G'' = G' \setminus \{1, 2\}$

2.1.3 Operationen mit Graphen

Neben den „intuitiv funktionierenden“ Operationen wie dem Einfügen bzw. Entfernen von Knoten (mit Knotengrad 0) und Kanten eines Graphen $G = (V, E)$, gehört die *Kontraktion* zweier Knoten $u, v \in V$ zu den am häufigsten genutzen Operationen. Der erzeugte Graph wird mit $G \setminus \{u, v\}$ bezeichnet und entsteht durch die Zusammenfassung der Knoten u und v in einem einzigen Knoten w, wobei alle zu u oder v inzidenten Kanten durch entsprechende zu w inzidente Kanten ersetzt werden. Existiert eine Kante $e = \{u, v\} \in E$, so spricht man auch von der Kontraktion der Kante e und bezeichnet den entstehenden Graphen mit $G \setminus e$. Bei der Kontraktion entstehende Schlingen werden entfernt, entstehende parallele Kanten können in bewerteten Graphen zu einer Kante zusammengefasst werden, wobei sich die Gewichte addieren. Ein Beispiel für die Kontraktion zweier Kanten in einem Graphen G liefert die Abbildung 2.5.

2.1.4 Zusammenhang und Schnitte

Ein ungerichteter Graph $G = (V, E)$ ist *zusammenhängend*, wenn jeder Knoten von jedem anderen Knoten aus erreichbar ist, d. h. wenn es für jeweils zwei Knoten $u, v \in V$ eine Folge von Kanten $e_1, e_2, \ldots, e_k$ mit den folgenden Eigenschaften gibt:

$$e_j = \{u_j, v_j\} \in E \quad \forall j = 1, \ldots, k, \tag{2.1}$$

$$v_j = u_{j+1} \quad \forall j = 1, \ldots, k-1, \tag{2.2}$$

$$u_1 = u \ \wedge \ v_k = v. \tag{2.3}$$

Ist dies nicht der Fall, so bezeichnet man den Graphen als *nicht zusammenhängend*.

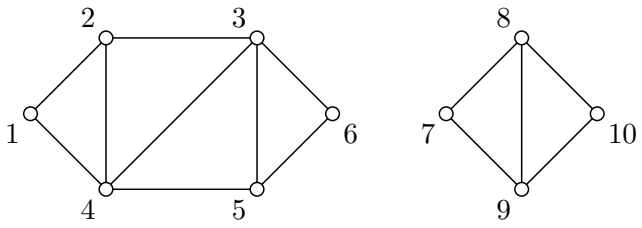


Abbildung 2.6: Ein nicht zusammenhängender Graph G mit 10 Knoten, 14 Kanten und 2 Komponenten. $G' = G[\{1, 2, 3\}]$ ist ein Untergraph, aber keine Komponente, da $G'' = G[\{1, 2, 3, 4, 5, 6\}]$ zusammenhängend ist und $V(G') \subset V(G'')$. Durch das Einfügen einer Kante zwischen zwei Knoten aus verschiedenen Komponenten, zum Beispiel $\{6, 7\}$, entsteht ein zusammenhängender Graph G.

Ein Graph $G' = (V', E')$ ist ein *Untergraph* des Graphen $G = (V, E)$, wenn $V' \subseteq V$ und $E' \subseteq E$. Ein durch die Knotenmenge V' induzierter Untergraph $G[V'] = G' = (V', E')$ ist ein Untergraph von G, dessen Kantenmenge E' aus den Kanten von E besteht, deren beide Endpunkte in V' liegen. Ein durch die Kantenmenge E' induzierter Untergraph $G[E'] = G' = (V, E')$ ist ein Untergraph von G mit der Knotenmenge V des Ausgangsgraphen und der Kantenmenge E'.

Eine *(zusammenhängende) Komponente* $G' = (V', E')$ eines Graphen $G = (V, E)$ ist ein durch die Knotenmenge V' induzierter Untergraph von G, der zusammenhängend und maximal ist. D. h. es gibt keinen zusammenhängenden Untergraphen $G[V'']$ mit $V' \subset V''$. Die *Komponentenanzahl* $c(G)$ eines Graphen G ist die Anzahl der (zusammenhängenden) Komponenten, aus denen der Graph G besteht. Abbildung 2.6 zeigt ein Beispiel für einen Graphen G mit $c(G) = 2$.

Ein *Schnitt* $(S, \overline{S})$ eines zusammenhängenden Graphen $G = (V, E)$ ist die Menge aller Kanten mit einem Endpunkt in S und einem Endpunkt in $\overline{S}$, wobei $\{S, \overline{S}\}$ eine Partition der Knotenmenge V ist ($\overline{S} = V \setminus S$). Die Kanten eines Schnittes werden *Schnittkanten* genannt und man sagt, sie *kreuzen* den Schnitt. Ein Schnitt kann durch die Angabe der Partition der Knotenmenge oder durch die Angabe der Schnittkanten eindeutig identifiziert werden.

Die *Kapazität* bzw. der *Wert* eines Schnittes $(S, \overline{S})$ ist die Anzahl der Schnittkanten, d. h. die Anzahl der Kanten mit jeweils genau einen Endknoten in S und einen Endknoten in $\overline{S}$.

Ein *minimaler Schnitt* eines zusammenhängenden Graphen G ist ein Schnitt des Graphen

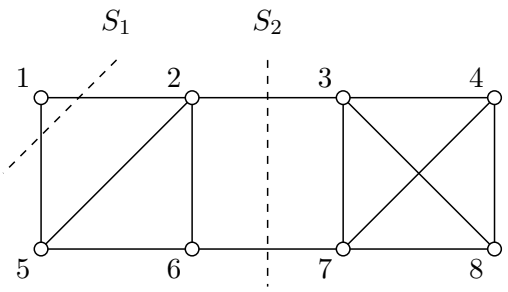


Abbildung 2.7: Ein Graph G mit 8 Knoten, 13 Kanten und mit dem Minimalschnittwert $\lambda(G) = 2$. G besitzt genau die zwei Minimalschnitte

$$S_1 = (\{1\}, \{2, 3, 4, 5, 6, 7, 8\}) \text{ und } S_2 = (\{1, 2, 5, 6\}, \{3, 4, 7, 8\}).$$

Die Schnittkanten von S_1 sind $\{1, 2\}$ und $\{1, 5\}$, die von S_2 sind $\{2, 3\}$ und $\{6, 7\}$.

mit minimalem Wert. Der Wert eines minimalen Schnittes eines Graphen G ist die *Kantenzusammenhangszahl* $\lambda(G)$.

Die Kantenzusammenhangszahl $\lambda(G)$ eines zusammenhängenden Graphen $G = (V, E)$ gibt an, wie viele Kanten aus der Kantenmenge E mindestens entfernt werden müssen, um den Zusammenhang des Graphen zu verlieren. Werden alle Kanten des Schnittes $(S, \overline{S})$ aus der Kantenmenge E entfernt, dann zerfällt der Graph G in die beiden Komponenten $G[S]$ und $G[\overline{S}]$. Ein Beispiel für die minimalen Schnitte eines Graphen zeigt die Abbildung 2.7.

Ein *r-Schnitt* $(S_1, \ldots, S_r)$ eines zusammenhängenden Graphen G ist die Verallgemeinerung eines Schnittes, bei dem die Knotenmenge in r Blöcke $S_1, \ldots, S_r$ partitioniert wird. Der Wert eines r-Schnittes ist die Anzahl der Kanten, die Endpunkte in verschiedenen Blöcken haben. Ein minimaler r-Schnitt ist ein r-Schnitt mit minimalem Wert.

2.2 Karger-Algorithmus zur Bestimmung minimaler Schnitte

Zur Bestimmung von minimalen Schnitten gibt es eine Reihe unterschiedlicher Ansätze und Algorithmen. Eine gute Einführung in dieses Thema bietet Levine in [20], der drei verschiedene Ansätze und jeweils verschiedene Algorithmen sowie Implementierungen erläutert.

An dieser Stelle wird nur der von KARGER in [15] erstmals vorgestellte Algorithmus beschrieben. In der Dissertation von KARGER wird der Algorithmus ebenfalls vorgstellt und auch auf Erweiterungen sowie Details der Implementierung eingegangen [16, Seiten 41 - 70, 213 - 218]. Die wichtigsten Aussagen sind ebenfalls in [18, 19] enthalten.

Der Algorithmus basiert auf der Idee, dass bei der zufälligen Kontraktion einer Kante nur mit geringer Wahrscheinlichkeit eine Kante eines bestimmten minimalen Schnittes kontrahiert wird. Dies folgt aus der Eigenschaft von minimalen Schnitten, dass diese gerade von möglichst wenigen Kanten gekreuzt werden. Andererseits lässt sich zeigen, dass ein bestimmter minimaler Schnitt des Ausgangsgraphen bei der Kontraktion von Kanten genau dann erhalten bleibt, wenn keine Schnittkante kontrahiert wird (Beweis siehe [19, Seite 9]).

Nach der sukzessiven Kontraktion von $n-2$ Kanten verbleibt ein Graph mit zwei Knoten, da sich bei jeder Kontraktion die Anzahl der Knoten um eins verringert. Dessen minimaler Schnitt ist direkt ablesbar: Alle vorhandenen Kanten sind Schnittkanten. Mit einer bestimmten Wahrscheinlichkeit wurde dabei keine Kante eines bestimmten minimalen Schnittes kontrahiert, so dass der minimale Schnitt des entstandenen Graphen mit zwei Knoten dem minimalen Schnitt des Ausgangsgraphen entspricht. Durch Wiederholung der Prozedur lässt sich die Wahrscheinlichkeit, einen minimalen Schnitt zu finden, immer weiter erhöhen.

Im Folgenden wird die Wahrscheinlichkeit dafür bestimmt, dass ein bestimmter minimaler Schnitt bei der sukzessiven Kontraktion von $n-2$ zufällig gewählten Kanten erhalten bleibt. Sowohl die Herleitung als auch die Folgerung für die Anzahl der minimalen Schnitte in einem Graphen mit n Knoten sind auch in [15, Seite 3] und [20, Seite 23] nachzulesen.

Gegeben sei ein unbewerteter, ungerichteter (Multi-)Graph $G = (V, E)$. Es seien $n = |V|$, $m = |E|$ und λ der Wert eines minimalen Schnittes.

Angenommen, es wurden bereits k Kontraktionen durchgeführt, bei denen ein bestimmter minimaler Schnitt erhalten geblieben ist, d. h. keine Kante dieses minimalen Schnittes wurde kontrahiert. Der entstandene Graph sei G_k. Für die Anzahl der Knoten von G_k gilt:

$$n_k = n - k. \tag{2.4}$$

Da G_k weiterhin einen minimalen Schnitt mit Wert λ hat, gilt $\deg(v) \geq \lambda \quad \forall v \in V(G_k)$ (sonst hätte der Schnitt, der die Knotenmenge in $\{v\}$ und $V \setminus \{v\}$ teilt, einen Wert kleiner λ). Für die Anzahl der Kanten nach k Kontraktionen folgt:

$$m_k \geq \lambda\, n_k/2 \tag{2.5}$$

und für die Wahrscheinlichkeit P_k, dass im nächsten Schritt eine Kante ausgewählt wird, die den minimalen Schnitt kreuzt, gilt:

$$P_k = \frac{\lambda}{m_k} \leq \frac{\lambda}{\lambda\, n_k/2} = \frac{2}{n_k} = \frac{2}{n-k}. \tag{2.6}$$

Für die Wahrscheinlichkeit P, dass bei $n-2$ aufeinanderfolgenden Kontraktionen einer zufälligen Kante keine Kante eines bestimmten minimalen Schnittes kontrahiert wird, folgt:

$$P \leq \prod_{k=0}^{n-3} [1 - P_k], \tag{2.7}$$

$$= (1 - \frac{2}{n-1})\,(1 - \frac{2}{n-2}) \cdots (1 - \frac{2}{3}), \tag{2.8}$$

$$= \frac{n-2}{n}\,\frac{n-3}{n-1} \cdots \frac{1}{3}, \tag{2.9}$$

$$= \frac{(n-2)\,(n-1)}{2}, \tag{2.10}$$

$$= \binom{n}{2}^{-1}. \tag{2.11}$$

Daher findet der beschriebene Algorithmus von Karger einen bestimmten minimalen Schnitt in einem Durchlauf mit Wahrscheinlichkeit $P = 1/\binom{n}{2}$. Die Ereignisse, dass der Algorithmus einen bestimmten Schnitt findet, sind disjunkt. Daraus folgt, dass ein Graph mit n Knoten maximal $\binom{n}{2}$ voneinander verschiedene minimale Schnitte haben kann.

Wird der Algorithmus so verändert, dass nur $n - r$ Kantenkontraktionen ausgeführt werden, so verbleibt am Ende ein Graph mit r Knoten. Dessen Kanten entsprechen den Kanten eines r-Schnittes. Auch für diese Variation lässt sich die Wahrscheinlichkeit $P(r)$ abschätzen, dass ein minimaler r-Schnitt gefunden wird. Nach [15, Seite 9] gilt:

$$P(r) \leq r \binom{n}{r-1}^{-1} \binom{n-1}{r-1}^{-1}. \tag{2.12}$$

Anstatt nacheinander Kanten im Graphen $G = (V, E)$ zu kontrahieren, bis nur noch 2 Knoten verbleiben, können auch Kanten aus der Kantenmenge E in den kantenleeren Graphen $G' = (V, \emptyset)$ eingefügt werden, bis der Graph aus nur noch 2 Komponenten besteht. Die Schnittkanten entsprechen in diesem Fall gerade den Kanten mit Endpunkten in verschiedenen Komponenten. Beide Vorgehensweisen basieren auf der gleichen Idee und sind aufeinander abbildbar.

Werden jeweils die gleichen Kanten verwendet, so entsprechen die in einem gemeinsamen Knoten kontrahierten Knoten der ursprünglichen Vorgehensweise gerade den in einer gemeinsamen Komponente befindlichen Knoten der alternativen Ausführung.

2.3 Clusterings

Der Begriff Clustering kommt ursprünglich aus dem Data-Mining („Datenschürfung“) und meint dort die Aufteilung einer Menge von Datensätzen in „natürliche Gruppen“ („decomposition of a set of entities into ‘natural groups’“ [5, Seite 178]). Aus diesem Verständnis heraus hat sich der Begriff Graphenclustering (Clustering eines Graphen) entwickelt, wobei die Knoten entsprechend ihrer Verbindungen (Kanten) untereinander in „natürliche Gruppen“ eingeteilt werden sollen.

Für einen Einstieg in Clusterings von Graphen sei auf das entsprechende Kapitel in [5, Seiten 178 - 215] verwiesen. Einen guten Überblick über verschiedene Clusteringtechniken (auch nicht graphenbasierte) bietet außerdem [12].

Die Bestimmung von Clusterings eines Graphen umfasst zwei Aufgabengebiete: Zum einen die Erstellung von Algorithmen zur Erzeugung von Clusterings und zum anderen die Festlegung von Qualitätsmaßen, um die erzeugten Clusterings zu bewerten.

Obwohl hier an zweiter Stelle erwähnt, gehen die Überlegungen zu den Qualitätsmaßen der Suche nach der algorithmischen Umsetzung voraus. Im Allgemeinen existiert eine bestimmte Aufgabenstellung, die ein Clustering eines Graphen erfordert. Entsprechend der „idealen Form“ eines solchen Clusterings wird ein Qualitätsmaß bestimmt, dass die zu erzielenden Eigenschaften des Clusterings berücksichtigt. Erst nach diesen Festlegungen ist es sinnvoll, sich über mögliche Herangehensweisen zur Erzeugung der Clusterings Gedanken zu machen.

Unter einem *Clustering* C eines Graphen $G = (V, E)$ versteht man eine Partition der Knotenmenge V, d. h. $C = \{C_1, \dots, C_k\}$, wobei man die Mengen C_i *Cluster* nennt und für diese gilt:

$$C_i \neq \emptyset \quad \forall i = 1, \dots, k, \tag{2.13}$$

$$C_i \cap C_j = \emptyset \quad \forall i \neq j, \tag{2.14}$$

$$\bigcup_{i=1}^{k} C_i = V. \tag{2.15}$$

Mit $E(C_i, C_j)$ bezeichnet man die Menge aller Kanten, die jeweils einen Endknoten im Cluster C_i und C_j haben. Die Menge aller Kanten, deren beide Endknoten im Cluster C_i liegen, nennt man *Intraclusterkanten* des Clusters C_i und bezeichnet diese mit $E(C_i) = E(C_i, C_i)$. Die Menge $E(C) = \bigcup_{i=1}^{k} E(C_i)$ aller Kanten, deren Endknoten im gleichen Cluster liegen, nennt man *Intraclusterkanten* des Clusterings C. Die Menge aller Kanten mit Endknoten in zwei verschiedenen Clustern nennt man *Interclusterkanten* von C und bezeichnet diese mit $\overline{E(C)} = E \setminus E(C) = \bigcup_{i \neq j} E(C_i, C_j)$. Die Menge aller möglichen Clusterings von G wird mit $\mathcal{A}(G)$ bezeichnet.

Im Folgenden wird eine Auswahl einfacher Qualitätsmaße von Clusterings beschrieben, denen analog zu [5] das „fundamental paradigm of intra-cluster density versus inter-cluster sparsity“ [5, Seite 178] zugrunde liegt. D. h. es werden Clusterings mit möglichst hoher Dichte innerhalb der Cluster (möglichst viele Intraclusterkanten) und möglichst großer Spärlichkeit zwischen den Clustern (möglichst wenige Interclusterkanten) gesucht. Alternative Herangehensweisen sind ebenfalls in [5, Seite 216 - 292] dargestellt.

Zur Darstellung der grundsätzlichen Ideen wird nur auf Clusterings $C = (C_1, \ldots, C_c)$ für ungerichtete, unbewertete und zusammenhängende Graphen $G = (V, E)$ eingegangen, d. h. die Knotenmenge V wird in genau c Teilmengen aufgeteilt.

Für die Bewertung der Qualität von Clusterings, die nach dem oben erwähnten Paradigma erstellt wurden, werden die zwei Funktionen

$$f(C), g(C) : \mathcal{A}(G) \to \mathbb{R}_0^+. \tag{2.16}$$

genutzt. Die Funktion $f(C)$ misst die Dichte der Intraclusterkanten und $g(C)$ die Spärlichkeit der Interclusterkanten. Beiden Funktionen werden in einer Funktion

$$h(C) = h(f(C), g(C)) \tag{2.17}$$

zusammengefasst, wobei $h(C)$ bezüglich beiden Funktionen monoton steigend sein soll, und im Allgeimeinen größere Werte von $h(C)$ für eine besseres Clustering stehen.

Das einfachste Qualitätsmaß, dass diesem Rahmen entspricht, ist die sogenannte *Abdeckung (coverage)* eines Clusterings C und wird mit $cov(C)$ bezeichnet. Es gilt:

$$cov(C) = h(C) = f(C) = |E(C)|. \tag{2.18}$$

Ein bezüglich der Abdeckung optimales Clustering C' des Graphen $G = (V, E)$ enthält folglich die maximal mögliche Anzahl an Intraclusterkanten. Andererseits ist die Anzahl der Kanten $m = |E|$ konstant, so dass die Anzahl der Interclusterkanten minimiert wird. Daher entspricht ein Clustering $C' = (C'_1, C'_2)$ mit maximaler Abdeckung genau einem minimalen Schnitt $S = (C'_1, C'_2)$, wobei die Interclusterkanten von C die Schnittkanten von S sind. Analog entspricht ein optimales Clustering mit c Clustern einem minimalen c-Schnitt des Graphen. Dadurch kann der Karger-Algorithmus auch als Algorithmus zur Bestimmung von Clusterings bezüglich der Abdeckung interpretiert werden.

Oftmals liefert ein bezüglich der Abdeckung optimales Clustering bzw. ein minimaler Schnitt jedoch keine Einteilung der Knotenmenge in „natürliche Gruppen“, zum Beispiel dann nicht, wenn wie in Abbildung 2.8 nur ein einzelner Knoten (mit minimalem Knotengrad) von den restlichen Knoten getrennt wird.

Um dieses Problem zu umgehen, versucht ein weiteres Qualitätsmaß, der sogenannte *Leitwert (Conductance)* $con(C)$, die „Größe“ der entstehenden Cluster zu berücksichtigen. Für

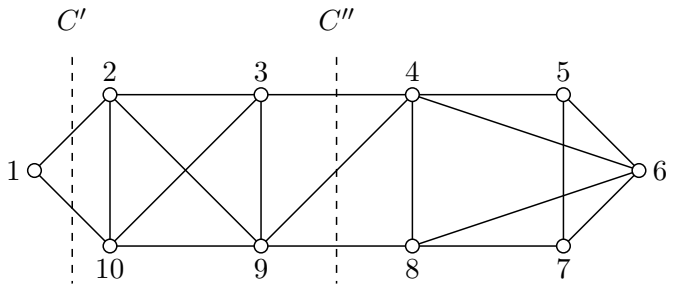


Abbildung 2.8: Clusterings mit 2 Clustern für ein Graphen G mit 10 Knoten. Das bezüglich der Abdeckung optimale Clustering ist $C' = (\{1\}, \{2, \ldots, 10\})$ mit $cov(C) = 2$. Bezüglich des Leitwertes ist zum Beispiel das Clustering $C'' = (\{1, 2, 3, 9, 10\}, \{4, 5, 6, 7, 8\})$ mit $con(C'') = \frac{8}{3}$ optimal.

die beiden Funktionen $f(C)$ und $g(C)$ gilt:

$$f(C) = \min_{i \in \{1,\ldots,c\}} |E(C_i)|, \tag{2.19}$$

$$g(C) = \frac{1}{|\overline{E(C)}|}. \tag{2.20}$$

Die Funktion $h(C)$ setzt sich aus dem Produkt beider Funktionen zusammen, es gilt:

$$con(C) = h(C) = f(C)\, g(C) = \frac{\min_{i \in \{1,\ldots,c\}} |E(C_i)|}{|\overline{E(C)}|}. \tag{2.21}$$

Ein optimales Clustering bezüglich des Leitwertes trennt die beiden Cluster an einem sogenannten „Flaschenhals“, einer Stelle, die im Vergleich zur „Größe“ der Cluster (hier minimale Anzahl von Intraclusterkanten eines Clusters) möglichst schmal ist (möglichst wenige Interclusterkanten hat). Abbildung 2.9 zeigt einen Graphen, der die Wahl des Begriffes „Flaschenhals“ verdeutlicht und Abbildung 2.10 gibt ein Beispiel für einen Graphen, in dem sich das optimale Clustering bezüglich des Leitwertes nicht mit einem „intuitiven“ Clustering deckt.

Die beiden vorgestellten Qualitätsmaße liefern im Allgemeinen kein „intuitives“ Clustering, welches das menschliche Auge „finden“ würde. Dies kann jedoch auch von komplexeren und auf anderen Paradigmen beruhenden Qualitätsmaßen nicht erreicht werden, da das „intuitive“ Clustering auch stark von der grafischen Darstellung des Graphen, und damit nicht allein vom Graphen selbst, abhängig ist.

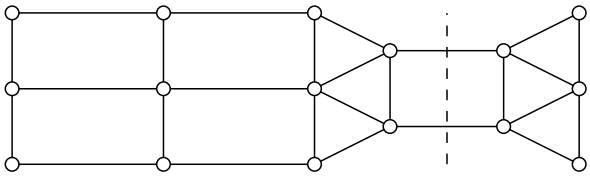


Abbildung 2.9: Ein Graph G in Form einer Flasche. Das optimale Clustering bezüglich des Leitwertes „trennt“ den Graphen am „Flaschenhals“.

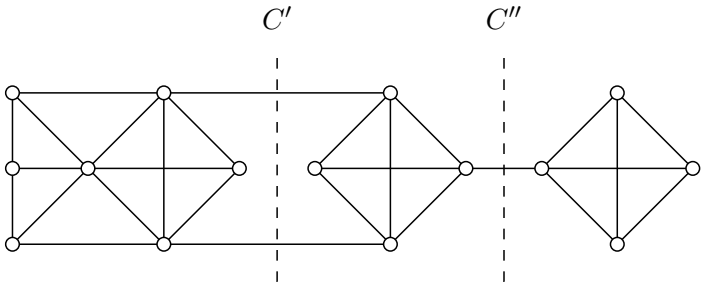


Abbildung 2.10: Clusterings mit 2 Clustern für einen Graphen G mit 15 Knoten. Eingezeichnet ist das bezüglich des Leitwertes optimale Clustering C' mit $con(C') = 6.5$ und das „intuitive“ Clustering C'' mit $con(C'') = 6$.

Gegenüber anderen Qualitätsmaßen haben Abdeckung und Leitwert einen zusätzlichen Schwachpunkt: beide beachten aussschließlich die Einteilung der Kanten in Interclusterkanten und in Intraclusterkanten der einzelnen Cluster. Die Struktur der Kanten innerhalb der Cluster wird dabei nicht beachtet.

Es kann festgehalten werden, dass beide Qualitätsmaße zu Clusterings mit unterschiedlichen Eigenschaften (minimaler Schnitt, minimaler „Flaschenhals“) führen und damit ein gutes Beispiel dafür sind, dass das gewählte Qualitätsmaß entscheidend für die Struktur eines bezüglich dieses Qualitätsmaßes optimalen Clusterings ist. Andererseits gilt, dass die Güte eines Clusterings nur unter Festlegung und Beachtung der zu erzielenden Eigenschaften sinnvoll gemessen werden kann.

2.4 Monte-Carlo-Simulation

Die *Monte-Carlo-Simulation* ist ein stochastisches Verfahren, bei dem durch eine Vielzahl von Zufallsexperimenten näherungsweise Lösungen für Erwartungswerte, Summen und Integrale bestimmt werden.

Einen Überblick über die Ideen der Methode und verschiedene Verfahren geben die Bücher von Glasserman [10], Hammersley und Handscomb [11], Kalos und Whitlock [14] sowie die Arbeiten von Anderson [2] und MacKay [21]. Einen Einblick in die Entstehungsgeschichte der Monte-Carlo-Simulation bietet [23]. Die benötigten Grundlagen aus der Stochastik werden zumeist in den ersten Kapiteln der entsprechenden Arbeiten wiederholt, ansonsten sei auf [22] verwiesen.

2.4.1 Direct Sampling

Unter *Direct Sampling* versteht man eine Monte-Carlo-Methode, die einen Wert, im vorliegenden Fall eine Summe, durch die Schätzung eines Erwartungswertes als arithmetisches Mittel unabhängiger Realisierungen berechnet. Sie wird in den meisten Büchern zur Monte-Carlo-Simulation beschrieben, oft direkt unter dem Begriff Monte-Carlo-Simulation oder *Monte-Carlo-Integration.*

Es seien

$$S = \sum_{i \in I} a(i) \tag{2.22}$$

eine Summe über einem endlichen Indexbereich I und $h(i)$ eine (Wahrscheinlichkeits-) Funktion mit

$$h : I \to [0, 1], \tag{2.23}$$

$$\sum_{i \in I} h(i) = 1 \text{ und} \tag{2.24}$$

$$a(i) \neq 0 \Rightarrow h(i) > 0. \tag{2.25}$$

Analog zu [7, Seite 29] wird die Zufallsvariable Y betrachtet, die einem Index $i \in I$ den Wert $y(i) = a(i)/h(i)$ zuweist:

$$Y : I \to \mathbb{R}, \tag{2.26}$$

$$i \mapsto y(i) = \frac{a(i)}{h(i)}. \tag{2.27}$$

Es bezeichne $E_h[Y]$ den Erwartungswert der Zufallsgröße Y, wobei $i \in I$ mit Wahrscheinlichkeit $h(i)$ gewählt wird. Dann gilt:

$$E_h[Y] = \sum_{i \in I} [y(i)\ h(i)] = \sum_{i \in I} \left[\frac{a(i)}{h(i)} h(i) \right] = \sum_{i \in I} a(i) = S. \tag{2.28}$$

Die Summe S entspricht dem Erwartungswert $E_h[Y]$ und kann, wie der Erwartungswert selber, durch eine Menge $y^{(1)}, \ldots, y^{(N)}$ von unabhängigen Realisierungen von Y geschätzt werden:

$$S \approx \widehat{S} = \widehat{E_h[Y]} = \overline{Y} = \frac{1}{N} \sum_{j=1}^{N} y^{(j)}. \tag{2.29}$$

Es sei $i^{(1)}, \ldots, i^{(N)}$ eine Menge unabhängiger Realisierungen aus der Menge I mit der Wahrscheinlichkeit $h(i)$. Dann gilt:

$$S \approx \widehat{S} = \widehat{E_h[Y]} = \overline{Y} = \frac{1}{N} \sum_{j=1}^{N} y^{(j)} = \frac{1}{N} \sum_{j=1}^{N} y(i^{(j)}) = \frac{1}{N} \sum_{j=1}^{N} \frac{a(i^{(j)})}{h(i^{(j)})}. \tag{2.30}$$

2.4.2 Importance Sampling

Importance Sampling ist eine Monte-Carlo-Methode, mit der die Varianz der Schätzung entsprechend Direct Samping reduziert werden soll. Dazu wird der Erwartungswert unter einer abweichenden Verteilung der Zufallsvariable geschätzt.

Als Literatur sind die Arbeiten von Geweke [9], Hastings [13], MacKay [21] und Neal [24, 25] zu empfehlen. Neben der Vorstellung der Schätzung mittels Importance Sampling bietet [21] eine leicht verständliche Einführung, [9, 24, 25] gehen außerdem auf die Berechnung der Varianz ein und [13] bezieht sich auf die Nutzung von Markovketten.

Es sei $g(i)$ eine Funktion mit

$$g : I \to [0, 1] \text{ und} \tag{2.31}$$

$$h(i) \neq 0 \Rightarrow g(i) > 0. \tag{2.32}$$

Dann gilt:

$$S = E_h[Y] = \sum_{i \in I} a(i), \tag{2.33}$$

$$= \sum_{i \in I} a(i) \; / \; \sum_{i \in I} h(i), \tag{2.34}$$

$$= \sum_{i \in I} \left[\frac{a(i)}{g(i)} g(i) \right] \; / \; \sum_{i \in I} \left[\frac{h(i)}{g(i)} g(i) \right], \tag{2.35}$$

$$= \sum_{i \in I} \left[\frac{a(i)}{h(i)} \frac{h(i)}{g(i)} g(i) \right] \; / \; \sum_{i \in I} \left[\frac{h(i)}{g(i)} g(i) \right]. \tag{2.36}$$

Analog zu Y sei W eine Zufallsvariable, wobei

$$W : I \to \mathbb{R}, \tag{2.37}$$

$$i \mapsto w(i) = \frac{h(i)}{g(i)}. \tag{2.38}$$

Es folgt:

$$S = E_h[Y] = \sum_{i \in I} [y(i) \; w(i) \; g(i)] \; / \; \sum_{i \in I} [w(i) \; g(i)], \tag{2.39}$$

$$= E_g[YW] \; / \; E_g[W]. \tag{2.40}$$

Es sei $i^{(1)}, \ldots, i^{(N)}$ eine Menge von unabhängigen Realisierungen aus der Menge I mit der Wahrscheinlichkeit proportional zu $g(i)$. Dann kann die Summe S wie folgt geschätzt werden:

$$S \approx \widehat{S} = \widehat{E_h[Y]} = \sum_{j=1}^{N} [y(i^{(j)}) w(i^{(j)})] \; / \; \sum_{j=1}^{N} w(i^{(j)}), \tag{2.41}$$

$$= \sum_{j=1}^{N} \frac{a(i^{(j)})}{g(i^{(j)})} \; / \; \sum_{j=1}^{N} \frac{h(i^{(j)})}{g(i^{(j)})}. \tag{2.42}$$

Geweke zeigt, dass diese Schätzung unter geringen Voraussetzungen konsistent ist (für steigendes N fast sicher gegen S konvergiert) [9, Seiten 1319 - 1320].

Importance Sampling wird oft als ein „Verfahren zur Varianzreduktion“ in der Literatur zur Monte-Carlo-Methode erwähnt und wurde mit diesem Ziel entwickelt. Allerdings können allgemein keine Aussagen darüber getroffen werden, ob die Varianz einer Simulation mit Importance Sampling geringer ist als bei einer Simulation mit Direct Sampling. Die Varianz und die Konvergenzgeschwindigkeit hängen entscheidend von der Wahl der Funktion $g(i)$ ab [9, Seite 1318].

Anderson nennt in [2, Seite 7] vier Eigenschaften, die eine „gute Importance-Sampling-Funktion“ $g(i)$ erfüllen sollte:

1. $g(i) > 0 \quad \forall i : a(i) \neq 0$,
2. $g(i)$ sollte ungefähr proportional zu $h(i)$ sein,
3. es sollte einfach sein, die Elemente $i \in I$ mit Wahrscheinlichkeit $g(i)$ zu wählen,
4. die Werte $g(i)$ sollten einfach berechenbar sein.

Neal stellt in [25, Seiten 13 - 14] die folgende Schätzung der Varianz von $\widehat{S}$ vor:

$$V[\widehat{S}] = \sum_{j=1}^{N} \left[w(i^{(j)}) \, (a(i^{(j)}) - \widehat{S}) \right]^2 \Big/ \left[\sum_{j=1}^{N} w(i^{(j)}) \right]^2 \tag{2.43}$$

Jedoch wird in [9, 21, 24] darauf hingewiesen, dass diese Schätzung der Varianz nicht in jedem Fall eine gute Einschätzung der Genauigkeit von $\widehat{S}$ liefert.

Folgerung 2.1 (Zusammenfassung). *Es sein $S = \sum_{i \in I} a(i)$ eine Summe über einer endlichen Indexmenge I, $i^{(1)}, \ldots, i^{(N)}$ eine Menge von unabhängigen Realisierungen aus der Menge I mit der Wahrscheinlichkeit proportional zu $g(i)$ und für die Funktionen $h(i)$ und $g(i)$ gelte:*

$$h : I \to [0, 1], \tag{2.44}$$

$$\sum_{i \in I} h(i) = 1, \tag{2.45}$$

$$g : I \to [0, 1], \tag{2.46}$$

$$a(i) \neq 0 \Rightarrow h(i) \neq 0 \Rightarrow g(i) \neq 0. \tag{2.47}$$

Dann kann S wie folgt geschätzt werden (Importance Sampling):

$$S \approx \sum_{j=1}^{N} \frac{a(i^{(j)})}{g(i^{(j)})} \Big/ \sum_{j=1}^{N} \frac{h(i^{(j)})}{g(i^{(j)})}. \tag{2.48}$$

Mit $g(i) = h(i)$ folgt (Direct Sampling):

$$S \approx \frac{1}{N} \sum_{j=1}^{N} \frac{a(i^{(j)})}{h(i^{(j)})}. \tag{2.49}$$

3 Graphen mit stochastisch existierenden Kanten

In diesem Kapitel wird eine besondere Klasse von Graphen vorgestellt, die als Graphen mit stochastisch existierenden Kanten bezeichnet werden. Zunächst werden die notwendigen Definitionen und Möglichkeiten zur Beschreibung eingeführt. Im Anschluss wird auf die Zusammenhangswahrscheinlichkeit als die zentrale Größe dieser Klasse von Graphen eingegangen und es werden Algorithmen zu ihrer Berechnung beschrieben. Abschließend wird ein Qualitätsmaß für Clusterings von Graphen mit stochastisch existierenden Kanten angegeben.

3.1 Definitionen und Darstellung

In Abschnitt 2.1 wurden Graphen als feste Strukturen mit einer bestimmten Anzahl von Knoten und Kanten vorgestellt. Oftmals werden aber Graphen benötigt, in denen Knoten oder Kanten nur mit einer gewissen Wahrscheinlichkeit existieren. Diese Arbeit betrachtet ausschließlich Graphen mit stochastisch existierenden Kanten, welche durch die folgende Definition beschrieben werden:

Definition 3.1. *Ein **Graph mit stochastisch existierenden Kanten** $ZG = (V, E, p)$ ist ein Graph $G = (V, E)$ mit der Kantenmenge $E = \{e_1, \ldots, e_{|E|}\}$ zusammen mit einer Funktion $p = p(z) : \{0,1\}^{|E|} \to [0,1]$, wenn gilt:*

1. $\forall z = (z_1, \ldots, z_{|E|}) \in \{0,1\}^{|E|} : p(z)$ *ist die Wahrscheinlichkeit, dass genau die Kanten $e_i \in E$ mit $z_i = 1 \quad \forall i = 1, \ldots, |E|$ existieren,*
2. $\sum_{z \in \{0,1\}^{|E|}} p(z) = 1$.

Im Fall von stochastisch unabhängig existierenden Kanten kann die Existenzwahrscheinlichkeit jeder Kante einzeln angegeben werden. Daraus ergibt sich folgende Definition:

Definition 3.2. *Ein **Graph mit stochastisch unabhängig existierenden Kanten** $ZG = (V, E, p)$ ist ein Graph $G = (V, E)$ zusammen mit einer Funktion $p = p(e) : E \to (0,1)$, wenn die Kanten $e \in E$ unabhängig voneinander mit Wahrscheinlichkeit $p(e)$ existieren.*

Eine weitere Vereinfachung tritt ein, wenn alle Kanten mit der gleichen Wahrscheinlichkeit existieren:

Definition 3.3. *Ein **Graph mit gleichwahrscheinlich unabhängig existierenden Kanten** $ZG = (V, E, p)$ ist ein Graph $G = (V, E)$ zusammen mit einer Zahl $p \in (0, 1)$, wenn alle Kanten $e \in E$ unabhängig voneinander mit Wahrscheinlichkeit p existieren.*

Im Folgenden wird der Begriff *Graph mit stochastisch existierenden Kanten* übergreifend für die in Definition 3.1 und ihren Vereinfachungen (Definitionen 3.2 und 3.3) benannten Fälle genutzt.

Die gängigen Graphenoperationen werden für Graphen mit stochastisch unabhängig existierenden Kanten $ZG = (V, E, p)$ definiert, indem die Graphenoperationen auf den Graphen $G = (V, E)$ angewendet werden und bei einer Veränderung der Kantenmenge der Definitionsbereich der Funktion p angepasst wird. Insbesondere wird definiert:

Definition 3.4. *Für den durch die Kantenmenge E' induzierten Untergraphen $ZG[E']$ gilt:*

$$ZG[E'] = (V, E', p|_{E'}).$$

Definition 3.5. *Für den durch die Knotenmenge V' induzierten Untergraphen $ZG[V']$ gilt:*

$$ZG[V'] = (V', E', p|_{E'}) \text{ mit } E' = \{e = (u, v) \in E \mid u \in V' \ \wedge \ v \in V'\}.$$

Im Fall eines Graphen mit stochastisch unabhängig existierenden Kanten $ZG = (V, E, p)$ wird zur mathematischen Darstellung ebenfalls auf die Graphentheorie zurückgegriffen und ZG als ein bewerteter Graph $G^* = (V^*, E^*)$ mit $V^* = V$, $E^* = E$ und der reellen Kantenbewertung $\omega : E^* \to (0, 1)$ beschrieben, wobei $\omega(e) = p(e)$.

Es wird dabei bewusst zwischen Graphen mit stochastisch unabhängig existierenden Kanten und ihrer Darstellung als Graphen mit reeller Kantenbewertung aus $(0, 1)$ unterschieden, da nicht jeder Graph mit einer reellen Kantenbewertung aus dem Intervall $(0, 1)$ so interpretiert werden kann, dass die Kanten nur stochastisch entsprechend der Kantenbewertung existieren.

Zum einen kann die Kantenbewertung als gewöhnliche reelle Bewertung angesehen werden, bei der durch eine Normierung die Werte auf das Intervall $(0, 1)$ beschränkt sind. Zum anderen ist es möglich, die Kantenbewertung als ein (Wahrscheinlichkeits-)Maß der Ähnlichkeit der Endknoten der Kante zu interpretieren. Für diesen Ansatz beschreiben Aslam, Leblanc und Stein in [3] einen Algorithmus, der zu einem Clustering des Graphen führt.

3.2 Beschreibung als Zufallsexperiment

Aus Sicht der Stochastik beschreibt ein Graph mit stochastisch existierenden Kanten $ZG = (V, E, p)$ ein Zufallsexperiment, das Zufallsexperiment ZG, das sich durch den Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$ beschreiben lässt:

- Ω ist die Ergebnismenge des Zufallsexperimentes und enthält alle möglichen Ergebnisse, die sogenannten Elementarereignisse,
- die σ-Algebra $\mathcal{F}$ ist die Ereignismenge des Zufallsexperimentes und enthält alle Ereignisse, denen eine Wahrscheinlichkeit zugeordnet wird,
- $\mathbb{P}$ ist ein Wahrscheinlichkeitsmaß und ordnet jedem Ereignis $A \in \mathcal{F}$ seine Wahrscheinlichkeit $\mathbb{P}(A)$ zu.

Ein Ausgang bzw. Ergebnis des Zufallsexperimentes ZG ist ein Graph $G' = (V', E')$, der auch als *Ergebnisgraphen* bezeichnet wird. Die Knotenmenge V' entspricht der Knotenmenge V von ZG, die Kantenmenge E' umfasst die Menge der existierenden Kanten und entspricht einer Teilmenge der Kantenmenge E von ZG. Damit ist G' ein Untergraph von $G = (V, E)$.

Folgerung 3.6. *Die Ergebnismenge Ω des Zufallsexperimentes $ZG = (V, E, p)$ umfasst alle Graphen $G' = (V, E')$, deren Kantenmenge eine Teilmenge von E ist:*

$$\Omega = \{G' = (V, E') \mid E' \subseteq E\}. \tag{3.1}$$

Zwei verschiedene Ergebnisgraphen unterscheiden sich genau in ihrer Kantenmenge und können dieser eineindeutig zugeordnet werden. Die Ergebnismenge umfasst genau $2^{|E|}$ Elemente.

Die Kantenmenge eines Ergebnisgraphen G', der Ergebnis eines Graphen mit stochastisch existierenden Kanten $ZG = (V, E, p)$ ist, kann einem binären Vektor zugeordnet werden. Die Dimension des Vektors entspricht dabei der Anzahl der potenziell existierenden Kanten $|E|$ und enthält an der Stelle i den Eintrag 1, falls die Kante $e_i \in E$ im Graphen G' enthalten ist, andernfalls enthält sie den Eintrag 0. Ein solcher Vektor wird als Statusvektor bezeichnet und durch die folgende Definition charakterisiert:

Definition 3.7. *Ein Vektor $z = (z_1, \ldots, z_{|E|}) \in \{0, 1\}^{|E|}$ heißt **Statusvektor** des Graphen $G' = (V', E')$, wenn G' Ergebnisgraph eines Graphen mit stochastisch existierenden Kanten $ZG = (V, E, p)$ ist und gilt:*

$$z_i = \begin{cases} 1 \Leftrightarrow e_i \in E', \\ 0 \Leftrightarrow e_i \notin E'. \end{cases} \tag{3.2}$$

Die Menge aller Statusvektoren wird mit $Z_{ZG} = \{0, 1\}^{|E|}$ bezeichnet.

Definition 3.8. *Die Funktion $z_{ZG}(G')$, die einem Ergebnisgraphen G' des Graphen mit stochastisch existierenden Kanten ZG seinen Statusvektor entsprechend Definition 3.7 zuweist, heißt* ***Statusvektorfunktion****.*

Definition 3.9. *Die Funktion $z_{ZG}^{-1}(z)$, die einem Statusvektor $z = (z_1, \ldots, z_{|E|})$ einen Graphen $G' = (V', E')$ zuweist, so dass $z_{ZG}(G') = z$, heißt* ***inverse Statusvektorfunktion****.*

Wenn klar ist, um welchen Graphen mit stochastisch existierenden Kanten ZG es sich handelt, wird kurz $z = z(G')$ statt $z = z_{ZG}(G')$, $G' = z^{-1}(z)$ statt $G' = z_{ZG}^{-1}(z)$ und Z statt Z_{ZG} geschrieben.

Zur besseren Beschreibung der Ergebnisse des Zufallsexperimentes, das durch einen Graphen mit stochastisch existierenden Kanten ZG beschrieben wird, werden zwei Zufallsvariablen definiert: Zum einen die Zufallsvariable ZG, die dem Ausgang des Zufallsexperimentes seinen Ergebnisgraphen zuordnet, und zum anderen die Zufallsvariable Z, die einem Ergebnisgraphen seinen Statusvektor zuordnet:

Definition 3.10. *Es sei $ZG : \Omega \to \Omega$ die Zufallsvariable, die dem Ausgang des Zufallsexperimentes ZG den Ergebnisgraphen $G' \in \Omega$ zuweist.*

Definition 3.11. *Es sei $ZZ : \Omega \to Z$ die Zufallsvariable, die einem Ergebnisgraphen $G' \in \Omega$ des Zufallsexperimentes $ZG = (V, E, p)$ seinen Statusvektor $z = z_{ZG}(G')$ zuweist.*

Die Definition der Zufallsvariable ZG scheint überflüssig, da sie einen Ergebnisgraphen nur auf sich selbst abbildet. Sie ist allerdings notwendig, um den Ausgang des Zufallsexperimentes ZG einfach und exakt zu beschreiben. Die Bezeichnung ZG wird bewusst in drei verschiedenen Zusammenhängen genutzt, da diese jeweils den gleichen Sachverhalt abbilden und sich aus dem Kontext heraus ergibt, ob es sich um einen Graphen mit stochastisch existierenden Kanten, ein Zufallsexperiment oder eine Zufallsvariable handelt. Die Bezeichnung ZG steht dabei für einen Zufallsgraphen, dessen Eigenschaften in Form des Graphen mit stochastisch existierenden Kanten ZG gegeben sind, dessen Realisierungen durch das Zufallsexperiment ZG bestimmt und dessen Ergebnisse durch die Zufallsvariable ZG beschrieben werden können.

Die Zufallsvariable ZZ kann als eine Funktion der Zufallsvariable ZG aufgefasst werden:

Folgerung 3.12. *Es seien ZG und ZZ die Zufallsvariablen zum Zufallsexperiment ZG. Dann gilt:*

$$ZZ = z(ZG) \tag{3.3}$$

und aufgrund der Eineindeutigkeit von z auch:

$$ZG = z^{-1}(ZZ). \tag{3.4}$$

Der Statusvektor $z = z_{ZG}(G')$ beschreibt den Ergebnisgraphen G' vollständig sowie eineindeutig (die Zuordnung in Definition 3.7 ist eineindeutig) und kann daher an Stelle des Ergebnisgraphen G' betrachtet werden. Analog entspricht die Zufallsvariable ZZ der Zufallsvariable ZG.

Folgerung 3.13. *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch existierenden Kanten, ZG sowie ZZ die entsprechenden Zufallsvariablen, $G' \in \Omega$ ein Ergebnisgraph und $z \in Z$ ein Statusvektor mit $z_{ZG}(G') = z$. Dann gilt für die Wahrscheinlichkeit $\mathbb{P}(ZG = G')$, dass das Zufallsexperiment ZG den Graphen G' erzeugt:*

$$\mathbb{P}(ZG = G') = \mathbb{P}(z(ZG) = z(G')) = \mathbb{P}(ZZ = z). \tag{3.5}$$

Die Folgerung stellt noch einmal heraus, dass anstatt des Zufallsexperimentes ZG und der Ergebnisgraphen $G' \in \Omega$ die Zufallsvariable ZZ und die Statusvektoren $z \in Z$ betrachtet werden können. Es wird daher bei der Bestimmung der entsprechenden Wahrscheinlichkeiten für die in den Definitionen 3.1 bis 3.3 angegebenen Fälle auf die Zufallsvariable ZZ zurückgegriffen.

Satz 3.14. *Für einen Graphen mit stochastisch existierenden Kanten $ZG = (V, E, p)$ gilt:*

$$\mathbb{P}(ZZ = z) = p(z).$$

Satz 3.15. *Für einen Graphen mit stochastisch unabhängig existierenden Kanten $ZG = (V, E, p)$ gilt:*

$$\mathbb{P}(ZZ = z) = \prod_{z_i=1} p(e_i) * \prod_{z_i=0} [1 - p(e_i)].$$

Satz 3.16. *Für einen Graphen mit gleichwahrscheinlich unabhängig existierenden Kanten $ZG = (V, E, p)$ gilt:*

$$\mathbb{P}(ZZ = z) = \prod_{z_i=1} p * \prod_{z_i=0} [1 - p].$$

Beweis der Sätze 3.14 bis 3.16. Satz 3.14 folgt direkt aus der Definition 3.1. Satz 3.15 folgt aus Definition 3.2 und dem Produktsatz für unabhängige Ereignisse. Satz 3.16 folgt aus Satz 3.15 zusammen mit der Vereinfachung gemäß Definition 3.3: $p(e_i) = p \quad \forall i = \{1, \ldots, |E|\}$. □

Im Folgenden werden die Schreibweisen $\mathbb{P}(ZZ = z) = P(z)$ und $\mathbb{P}(ZG = G) = P(G)$ verwendet.

Für die Beschreibung des Zufallsexperimentes $ZG = (V, E, p)$ durch den Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$ folgt:

- $\Omega = \{G' = (V, E') \mid V' = V,\ E' \subseteq E\}$,
- $\mathcal{F} = 2^{\Omega}$ (die Potenzmenge von Ω),
- $\forall A \in \mathcal{F}: \ \mathbb{P}(A) = \sum_{G \in A} \mathbb{P}(G)$.

Die wichtigsten Erkenntnisse aus diesem Abschnitt werden noch einmal in einer Folgerung zusammengefasst:

Folgerung 3.17. *Der Graph mit stochastisch existierenden Kanten $ZG = (V, E, P)$ beschreibt ein Zufallsexperiment, dessen Ergebnis in Form eines Statusvektors $z \in Z = \{0, 1\}^{|E|}$ ausgedrückt werden kann. Die Wahrscheinlichkeit für die Realisierung eines Statusvektors z beträgt $P(z)$.*

3.3 Zusammenhangswahrscheinlichkeit

Graphen mit stochastisch existierenden Kanten entsprechen Zufallsexperimenten, deren Ergebnisse Graphen, die sogenannten Ergebnisgraphen, sind. Daher lassen sich im Allgemeinen keine deterministischen Aussagen über die Struktur und Eigenschaften dieser Graphen machen. Da aber jeder Ergebnisgraph mit einer durch den Graphen mit stochastisch existierenden Kanten festgelegten Wahrscheinlichkeit eintritt, ist die Bestimmung stochastischer Eigenschaften möglich.

Die am häufigsten betrachtete Eigenschaft ist die Zusammenhangswahrscheinlichkeit, d. h. die Wahrscheinlichkeit, dass ein Ergebnisgraph zusammenhängend ist. Die Zusammenhangswahrscheinlichkeit ist damit eine Eigenschaft des Graphen mit stochastisch existierenden Kanten bzw. seiner Repräsentation als Graph mit reeller Kantenbewertung aus dem Intervall $(0, 1)$.

Zur Berechnung der Zusammenhangswahrscheinlichkeit gibt es eine Vielzahl verschiedener Algorithmem und entsprechender Literatur. Einen sehr guten Überblick über verschiedene Methoden bieten Ball, Colbourn und Provan in [4]. Fishman vergleicht in [8] vier Methoden der Berechnung mittels Monte-Carlo-Methoden und geht auch auf die von Easton und Wong in [7] vorgestellte *Sequential Destruction Method* ein, die in leicht abgewandelter Weise in Punkte 3.3.4 vorgestellt wird. Außerdem sei auf die Arbeit von Karger [17] hingewiesen, der für die Berechnung der Zusammenhangswahrscheinlichkeit den in Abschnitt 2.2 vorgestellten Algorithmus zur Bestimmung minimaler Schnitte nutzt.

3.3.1 Definition

Definition 3.18. *Es sei $\mathcal{G}$ die Menge aller Graphen. Die Funktion $\phi_c = \phi_c(G) : \mathcal{G} \to \{0,1\}$ heißt c-**Komponenten-Zusammenhangsfunktion** wenn gilt:*

$$\phi_c(G) = \begin{cases} 1 \Leftrightarrow G \text{ besteht aus maximal } c \text{ Komponenten,} \\ 0 \Leftrightarrow G \text{ besteht aus mehr als } c \text{ Komponenten.} \end{cases} \tag{3.6}$$

Definition 3.19. *Die Funktion $\phi_1 = \phi_1(G) : \mathcal{G} \to \{0,1\}$ heißt **Zusammenhangsfunktion** und es gilt:*

$$\phi_1(G) = \begin{cases} 1 \Leftrightarrow G \text{ ist zusammenhängend,} \\ 0 \Leftrightarrow G \text{ ist nicht zusammenhängend.} \end{cases} \tag{3.7}$$

Definition 3.20. *Der Wert $R_c(ZG) = E[\phi_c(ZG)] = \mathbb{P}(\phi_c(ZG) = 1)$ heißt c-**Komponenten-Zusammenhangswahrscheinlichkeit** des Graphen mit stochastisch existierenden Kanten $ZG = (V, E, p)$.*

Definition 3.21. *Der Wert $R(ZG) = R_1(ZG) = E[\phi_1(ZG)] = \mathbb{P}(\phi_1(ZG) = 1)$ heißt **Zusammenhangswahrscheinlichkeit** des Graphen mit stochastisch existierenden Kanten $ZG = (V, E, p)$.*

Die Zusammenhangswahrscheinlichkeit $R(ZG)$ eines Graphen mit stochastisch existierenden Kanten ZG ist die zentrale Größe für viele Aufgabenstellungen, die als ein solcher Graph modelliert werden. Beispiele sind unter anderem Kommunikationsnetzwerke und Verkehrswege, bei denen die Kreuzungen als Knoten und deren Verbindungen als Kanten dargestellt werden. Die Antwort auf die Frage, mit welcher Wahrscheinlichkeit eine Kommunikation bzw. eine Fortbewegung zwischen allen Kreuzungen möglich ist, liefert gerade die Zusammenhangswahrscheinlichkeit.

Um die Bijektion zwischen der Menge der Ergebnisgraphen und den Statusvektoren eines Zufallsexperimentes ZG einfach nutzen zu können und auch allgemein für reellwertige Funktionen auf Graphen anzuwenden, wird definiert:

Definition 3.22. *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch existierenden Kanten, $G' \in \Omega$ ein Ergebnisgraph des Zufallsexperimentes ZG, $z = z(G')$ der zum Ergebnisgraphen G' gehörende Statusvektor und $\phi : \mathcal{G} \to \mathbb{R}$ eine Funktion, die einem Graphen eine reelle Zahl zuordnet. Dann gilt:*

$$\phi(z) = \phi(z^{-1}(z)) = \phi(G'). \tag{3.8}$$

Sind ZG und ZZ die dem Zufallsexperiment ZG nach den Definitionen 3.10 und 3.11 entsprechenden Zufallsvariablen, so gilt:

$$\phi(ZZ) = \phi(z^{-1}(ZZ)) = \phi(ZG). \tag{3.9}$$

Damit werden insbesondere auch die Zusammenhangsfunktion bzw. die c-Komponenten-Zusammenhangsfunktion für Statusvektoren bzw. die Zufallsvariable ZZ erweitert.

Im Folgenden werden verschiedene Möglichkeiten zur Bestimmung des Erwartungswertes einer Funktion der Zufallsvariablen ZG sowie ZZ und damit auch der c-Komponenten-Zusammenhangswahrscheinlichkeit $R_c(ZG)$ inklusive der Zusammenhangswahrscheinlichkeit $R(ZG)$ vorgestellt. Soweit keine weiteren Forderungen gestellt werden, gelten die Ergebnisse aber allgemein für Funktionen $\phi : \mathcal{G} \to \mathbb{R}$.

3.3.2 Berechnung mit Enumeration

Unter *Enumeration* versteht man in der Mathematik das Aufzählen aller Objekte mit bestimmten Eigenschaften. Im Fall der Berechnung der Zusammenhangswahrscheinlichkeit eines Graphen mit stochastisch existierenden Kanten ist damit die Betrachtung aller möglichen Ergebnisgraphen gemeint.

Entsprechend der Definition des Erwartungswertes für Zufallsvariablen mit diskreten Verteilungen [22, Seite 92] gilt:

$$E[\phi(ZG)] = \sum_{G \in \Omega} [\phi(G)\ \mathbb{P}(ZG = G)] = \sum_{G \in \Omega} [\phi(G)\ P(G)] \tag{3.10}$$

bzw.

$$E[\phi(ZZ)] = \sum_{z \in Z} [\phi(z)\ \mathbb{P}(ZZ = z)] = \sum_{z \in Z} [\phi(z)\ P(z)]. \tag{3.11}$$

Für die Zusammenhangswahrscheinlichkeit gilt entsprechend Definition 3.21:

$$R_c(ZG) = E[\phi_c(ZG)] = \sum_{G \in \Omega} [\phi_c(G)\ \mathbb{P}(ZG = G)] = \sum_{G \in \Omega} [\phi_c(G)\ P(G)] \tag{3.12}$$

$$= E[\phi_c(ZZ)] = \sum_{z \in Z} [\phi_c(z)\ \mathbb{P}(ZZ = z)] = \sum_{z \in Z} [\phi_c(z)\ P(z)]. \tag{3.13}$$

Da die Menge der zu betrachtenden Summanden (entspricht der Anzahl der möglichen Graphen bzw. Statusvektoren) exponentiell wächst, ist die Enumeration nur für sehr kleine Beispiele praktisch durchführbar. Beispielsweise müssen für einen Graphen mit stochastisch existierenden Kanten $ZG = (V, E, p)$ mit zehn Kanten ($|E| = 10$) bereits $2^{10} = 1024$ Summanden ausgewertet werden, da es für jede Kante zwei Möglichkeiten gibt (existieren oder nicht existieren). Mit jeder zusätzlichen Kante verdoppelt sich die Anzahl der Summanden, so dass bei 20 Kanten bereits über eine Million Summanden, bei 30 Kanten über eine Milliarde Summanden für die Enumeration ausgewertet werden müssen.

3.3.3 Berechnung mit Naive Sample

Die Formeln (3.12) bzw. (3.13) bilden die Basis für die Berechnung der Zusammenhangswahrscheinlichkeit mittels *Naive Sample*. Dabei wird die bei der Enumeration betrachtete Summe nur näherungsweise bestimmt.

Entsprechend Formel (2.30) kann die Summe

$$E[\phi(ZZ)] = \sum_{z \in Z} a(z) \text{ mit } a(z) = \phi(z)\ P(z) \tag{3.14}$$

durch die Menge $z^{(1)}, \ldots, z^{(N)}$ unabhängiger Realisierungen der Zufallsgröße ZZ, wobei der Statusvektor $z \in Z = \{0,1\}^{|E|}$ mit Wahrscheinlichkeit $P(z)$ gewählt wird, geschätzt werden:

$$E[\phi(ZZ)] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{a(z^{(i)})}{P(z^{(i)})} = \frac{1}{N} \sum_{i=1}^{N} \phi(z^{(i)}). \tag{3.15}$$

Im Fall der Zusammenhangsfunktion $\phi_1 : \mathcal{G} \to \{0,1\}$ entspricht dies gerade der Bestimmung der Wahrscheinlichkeit $\mathbb{P}(\phi_1(ZG) = 1)$ entsprechend der klassischen Definition der Wahrscheinlichkeit als relative Häufigkeit (Anzahl der Versuche mit $\phi_1(G) = 1$ / Anzahl der Versuche).

3.3.4 Berechnung mit Sequential Construction

Unter *Sequential Construction* versteht man eine Approximation der Zusammenhangswahrscheinlichkeit durch sequenzielles Hinzufügen von Kanten. Dabei werden nicht Graphen oder Statusvektoren, sondern Permutationen von Kanten simuliert und betrachtet.

Eine Permutation der Kanten beschreibt dabei $|E| + 1$ verschiedene Ergebnisgraphen, deren Kantenmenge jeweils aus den ersten j Kanten der Permutation besteht.

Definition 3.23. *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch existierenden Kanten, Π die Menge aller Permutationen der Kantenmenge E und $z(\pi, j)$ eine Funktion, die einer Permutation $\pi \in \Pi$ und einer natürlichen Zahl $j \in \{0, \ldots, |E|\}$ den Statusvektor $z \in Z = \{0,1\}^{|E|}$ zuweist, der gerade dem Ergebnisgraphen mit genau den ersten j Kanten der Permutation π entspricht:*

$$z(\pi, j) = z = (z_1, \ldots, z_{|E|}) \text{ mit } z_i = \begin{cases} 1, & \pi^{-1}(e_i) \leq j, \\ 0, & \pi^{-1}(e_i) > j. \end{cases} \tag{3.16}$$

Dabei ist π^{-1} die inverse Permutation der Permutation π und $\pi^{-1}(e_i)$ die Position der Kante e_i in der Permutation π.

Es wird die Multimenge M betrachtet, die die Statusvektoren für alle verschiedenen Permutationen $\pi \in \Pi$ und alle natürlichen Zahlen $j \in \{0, \ldots, |E|\}$ enthält, d. h.

$$M = \bigcup_{\substack{\pi \in \Pi \\ j \in \{0,\ldots,|E|\}}} \{z(\pi, j)\}. \tag{3.17}$$

Für die Anzahl der Elemente der Multimenge M gilt:

$$|M| = |\Pi| \; |\{0, \ldots, |E|\}| = (|E|)! \; (|E| + 1) = (|E| + 1)!. \tag{3.18}$$

Da es nur $|\{0, 1\}^{|E|}| = 2^{|E|}$ verschiedene Elemente gibt, müssen einige Elemente mehrfach in M enthalten sein.

Es sei $z' = (z'_1, \ldots, z'_{|E|})$ ein Statusvektor mit genau j' Einträgen 1, d. h. $\sum_{i=1}^{|E|} z'_i = j'$. Der Statusvektor z' kann nur als Operand der Vereinigung in Formel (3.17) entstehen, wenn gilt:

- $j = j'$,
- $(\pi_1, \ldots, \pi_{j'})$ ist eine Permutation der Kanten e_i mit $z_i = 1$ und
- $(\pi_{j'+1}, \ldots, \pi_{|E|})$ ist eine Permutation der Kanten e_i mit $z_i = 0$.

Für eine Permutation der j' Kanten mit dem Eintrag 1 gibt es $j'!$ Möglichkeiten, für eine Permutation der $|E| - j'$ Kanten mit dem Eintrag 0 gibt es $(|E| - j')!$ Möglichkeiten. Folglich ist der Statusvektor z' genau $j'! \; (|E| - j')!$ Mal in der Multimenge M enthalten.

Für eine reellwertige Funktion $a(z) : Z \to \mathbb{R}$ gilt daher:

$$\sum_{z \in \{0,1\}^{|E|}} a(z) = \sum_{\substack{\pi \in \Pi \\ j \in \{0,\ldots,|E|\}}} \frac{a(z(\pi, j))}{j! \; (|E| - j)!} = \sum_{\pi \in \Pi} \sum_{j=0}^{|E|} \frac{a(z(\pi, j))}{j! \; (|E| - j)!}. \tag{3.19}$$

Mit $a(z) = \phi(z) \; P(z)$ kann der Erwartungswert $E[\phi(ZZ)]$ mittels Sequential Construction wie folgt exakt berechnet werden:

$$E[\phi(ZZ)] = \sum_{z \in Z} \phi(z) \; P(z) = \sum_{\pi \in \Pi} \sum_{j=0}^{|E|} \frac{\phi(z(\pi, j)) \; P(z(\pi, j))}{j! \; (|E| - j)!}. \tag{3.20}$$

Für die c-Komponenten-Zusammenhangswahrscheinlichkeit folgt:

$$R_c(ZG) = E[\phi_c(ZZ)] = \sum_{z \in Z} \phi_c(z) \; P(z) = \sum_{\pi \in \Pi} \sum_{j=0}^{|E|} \frac{\phi_c(z(\pi, j)) \; P(z(\pi, j))}{j! \; (|E| - j)!}. \tag{3.21}$$

Mit der Schreibweise

$$a(\pi) = \sum_{j=0}^{|E|} \frac{\phi(z(\pi, j))\ P(z(\pi, j))}{j!\ (|E| - j)!}$$

gilt:

$$E[\phi(ZZ)] = \sum_{\pi \in \Pi} a(\pi). \tag{3.22}$$

Definition 3.24. *Es sei ZP eine Zufallsvariable, der für einen Graphen mit stochastisch existierenden Kanten $ZG = (V, E, p)$ eine Permutation $\pi \in \Pi$ der Kantenmenge E mit Wahrscheinlichkeit $p(\pi)$ zugeordnet wird, wobei gilt:*

$$\sum_{\pi \in \Pi} p(\pi) = 1. \tag{3.23}$$

Es sei $\pi^{(1)}, \ldots, \pi^{(N)}$ eine Menge von unabhängigen Realisierungen der Zufallsvariable ZP. Dann lässt sich die Summe (3.22) durch Monte-Carlo-Simulation entsprechend Formel (2.30) schätzen:

$$\begin{aligned}
E[\phi(ZZ)] &= \sum_{\pi \in \Pi} a(\pi), && (3.24)\\
&\approx \frac{1}{N} \sum_{i=1}^{N} \frac{a(\pi^{(i)})}{p(\pi^{(i)})}, && (3.25)\\
&\approx \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{p(\pi^{(i)})} \sum_{j=0}^{|E|} \frac{\phi(z(\pi^{(i)}, j))\ P(z(\pi^{(i)}, j))}{j!\ (|E| - j)!} \right], && (3.26)\\
&\approx \frac{1}{N} \sum_{i=1}^{N} \sum_{j=0}^{|E|} \frac{\phi(z(\pi^{(i)}, j))\ P(z(\pi^{(i)}, j))}{p(\pi^{(i)})\ j!\ (|E| - j)!}. && (3.27)
\end{aligned}$$

Für Funktionen $\tilde{\phi} : \mathcal{G} \to \{0, 1\}$, für die für alle Permutationen π ein eindeutiger Index $k = k(\pi) \in \{0, \ldots, |E| + 1\}$ existiert, so dass

$$\tilde{\phi}(z(\pi, j)) = \begin{cases} 1, & j \geq k, \\ 0, & j < k \end{cases} \tag{3.28}$$

gilt, ist die folgende Vereinfachung möglich: Es sei $k^{(i)} = k(\pi^{(i)})$ der Index entsprechend Formel (3.28) zur Permutation $\pi^{(i)}$, d. h.

$$k^{(i)} = \min_{j \in \{0, \ldots, |E|+1\}} \tilde{\phi}(\pi^{(i)}, j) = 1. \tag{3.29}$$

Die Indexmenge der zweiten Summe kann so angepasst werden, dass nur Summanden ungleich 0 betrachtet werden:

$$E[\tilde{\phi}(ZZ)] = \sum_{\pi\in\Pi}\sum_{j=0}^{|E|}\frac{\tilde{\phi}(z(\pi,j))\ P(z(\pi,j))}{j!\ (|E|-j)!}, \tag{3.30}$$

$$= \sum_{\pi\in\Pi}\sum_{j=k^{(i)}}^{|E|}\frac{P(z(\pi,j))}{j!\ (|E|-j)!}. \tag{3.31}$$

Für die Schätzung gilt entsprechend:

$$E[\tilde{\phi}(ZZ)] \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{|E|}\frac{\tilde{\phi}(z(\pi^{(i)},j))\ P(z(\pi^{(i)},j))}{p(\pi^{(i)})\ j!\ (|E|-j)!}, \tag{3.32}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}\sum_{j=k^{(i)}}^{|E|}\frac{P(z(\pi^{(i)},j))}{p(\pi^{(i)})\ j!\ (|E|-j)!}. \tag{3.33}$$

Für eine Funktion $\tilde{\phi}$ tritt der Fall $k = |E|+1$ genau dann ein, wenn $\tilde{\phi}(z(\pi,j)) = 0 \quad \forall j \in \{0,\ldots,|E|\}$. Daraus folgt, dass $E[\tilde{\phi}(ZZ)] = 0$. Im Fall der Zusammenhangsfunktion ϕ_1 geschieht dies zum Beispiel, wenn die Zusammenhangswahrscheinlichkeit eines Graphen mit stochastisch existierenden Kanten $ZG = (V,E,p)$ untersucht wird, wobei der Graph $G = (V,E)$ nicht zusammenhängend ist. Ist diese Situation ausgeschlossen, so ist $k \in \{0,\ldots,|E|\}$.

Für die Bedingung aus Formel (3.28) sind auch andere Formulierungen möglich:

1. $j < j' \Rightarrow \tilde{\phi}(z(\pi,j)) \leq \tilde{\phi}(z(\pi,j'))$,
2. $\phi((V,E)) = 1 \ \wedge\ E \subseteq E' \Rightarrow \phi((V,E')) = 1$,
3. das Paar (E,U) aus der Kantenmenge E und der Menge von Kantenmengen U,

   $$U = \{E' \mid \phi((V,E')) = 0\},$$

   bilden ein Unabhängigkeitssystem.

Dabei sind die Bedingungen 1 und 2 der Bedingung aus Formel (3.28) äquivalent, Bedingung 3 ist dagegen hinreichend, aber nicht notwendig.

Die c-Komponenten-Zusammenhangsfunktion und die Zusammenhangsfunktion erfüllen diese Bedingungen (ϕ_c erfüllt Bedingung 3 nur für Graphen $G = (V,E)$ mit $|V| > c$), daher können die Vereinfachungen auch zur Berechnung der c-Komponenten-Zusammenhangswahrscheinlichkeit genutzt werden.

Die in den folgenden Abschnitten benötigten Ergebnisse werden in einer Folgerung zusammengefasst:

Folgerung 3.25 (Zusammenfassung). *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch existierenden Kanten, $\pi^{(1)}, \ldots, \pi^{(N)}$ eine Menge von unabhängigen Realisierungen der Zufallsvariable ZP und $k^{(i)}$ der kleinste Index j, für den $\phi_c(z(\pi^{(i)}, j)) = 1$. Dann lässt sich die Zusammenhangswahrscheinlichkeit $R_c(ZG)$ wie folgt exakt berechnen und schätzen:*

$$R_c(ZG) = \sum_{\pi \in \Pi} \sum_{j=0}^{|E|} \frac{\phi_c(z(\pi, j))\ P(z(\pi, j))}{j!\ (|E| - j)!}, \tag{3.34}$$

$$= \sum_{\pi \in \Pi} \sum_{j=k^{(i)}}^{|E|} \frac{P(z(\pi, j))}{j!\ (|E| - j)!}, \tag{3.35}$$

$$\approx \sum_{i=1}^{N} \sum_{j=k^{(i)}}^{|E|} \frac{P(z(\pi^{(i)}, j))}{j!\ (|E| - j)!\ p(\pi^{(i)})}, \tag{3.36}$$

$$\approx \sum_{i=1}^{N} \frac{a(\pi^{(i)})}{p(\pi^{(i)})} \text{ mit } a(\pi^{(i)}) = \sum_{j=k^{(i)}}^{|E|} \frac{P(z(\pi^{(i)}, j))}{j!\ (|E| - j)!}. \tag{3.37}$$

Easton [7, Seiten 30 - 32] und Fishman [8, Seite 148] haben gezeigt, dass die näherungsweisen Berechnung der Zusammenhangswahrscheinlichkeit mit Sequential Construction nach Formel (3.37) eine geringere Varianz hat als die Berechnung mit Naive Sample nach Formel (3.15).

3.4 Qualitätsmaß für Clusterings

Nachdem in Abschnitt 2.3 der Begriff Clustering und zwei einfache Qualitätsmaße für Clusterings vorgestellt wurden, soll im Folgenden ein Qualitätsmaß für Clusterings von Graphen mit stochastisch existierenden Kanten angegeben werden.

Aufgrund der zentralen Bedeutung der Zusammenhangswahrscheinlichkeit bzw. allgemein des Zusammenhangs für Graphen mit stochastisch existierenden Kanten erscheint es sinnvoll „natürliche Gruppen“ im Sinne von zusammenhängenden Komponenten zu interpretieren. Eine „natürliche Gruppe“ ist dann eine Menge von Knoten, die mit hoher Wahrscheinlichkeit zusammenhängend ist und das entsprechende Qualitätsmaß die sogenannte interne Zusammenhangswahrscheinlichkeit.

Mit der internen Zusammenhangswahrscheinlichkeit eines Clusterings ist die Wahrscheinlichkeit gemeint, dass die Knoten der einzelnen Cluster durch Intraclusterkanten zusammenhängen:

Definition 3.26. *Es sei $ZG = (V, E, p)$ ein Graph mit stochastisch existierenden Kanten und $C = (C_1, \ldots, C_c)$ ein Clustering von ZG. Unter der **internen Zusammenhangswahrscheinlichkeit** $R(C)$ des Clusterings C versteht man die Größe:*

$$R(C) = \prod_{i=1}^{c} R(ZG[C_i]). \tag{3.38}$$

Dabei ist $ZG[C_i]$ entsprechend Definition 3.5 der von der Knotenmenge C_i induzierte Untergraph (der Graph bestehend aus den Knoten des Clusters C_i und den Intraclusterkanten $E(C_i)$) und $R(ZG[C_i])$ die Zusammenhangswahrscheinlichkeit des Clusters C_i.

Für die Zusammenhangswahrscheinlichkeit $R(C)$ des Clusterings C werden daher nur die Intraclusterkanten $E(C)$ betrachtet. Der Graph $G = (V, E(C))$ besteht aus genau c Komponenten, da C ein Clustering mit c Clustern ist. Die Wahrscheinlichkeit $R(C)$ kann daher auch als die Wahrscheinlichkeit berechnet werden, dass der durch die Kantenmenge $E(C)$ induzierte Untergraph von ZG, d. h. $ZG[E(C)]$ (Definition 3.4), aus genau c Komponenten besteht:

Folgerung 3.27. *Es sei $ZG = (V, E, p)$ ein Graph mit stochastisch existierenden Kanten und $C = (C_1, \ldots, C_c)$ ein Clustering von ZG. Für die interne Zusammenhangswahrscheinlichkeit $R(C)$ gilt:*

$$R(C) = \prod_{i=1}^{c} R(ZG[C_i]) = R_c(ZG[E(C)]). \tag{3.39}$$

Die interne Zusammenhangswahrscheinlichkeit $R(C)$ kann ähnlich wie die in Abschnitt 2.3 vorgestellte Abdeckung (coverage, $cov(C) = |E|$) als Maß der „Dichte“ der Intraclusterkanten verwendet werden. Anders als bei der Abdeckung wird bei der internen Zusammenhangswahrscheinlichkeit jedoch nicht nur der Status der Kanten als Intra- bzw. Interclusterkanten, sondern auch die Struktur der Intraclusterkanten betrachtet.

Es erscheint außerdem sinnvoll anzunehmen, dass intuitive Clusterings und für konkrete Aufgabenstellungen relevante Clusterings über eine hohe interne Zusammenhangswahrscheinlichkeit verfügen.

Im Gegensatz dazu wird sich das bezüglich der internen Zusammenhangswahrscheinlichkeit optimale Clustering jedoch häufig nicht mit einem intuitiven oder relevanten Clustering decken bzw. eindeutig keine gute Aufteilung in „natürliche Gruppen“ bilden. Stattdessen wird das Clustering aus einer großen Komponente und einer Menge isolierter Knoten bestehen, wie es auch bei Clusterings bezüglich der Abdeckung oft der Fall ist. Ein Beispiel dafür zeigt die Abbildung 3.1.

Im Vergleich zur Abdeckung wird dieses Problem bei der internen Zusammenhangswahrscheinlichkeit zusätzlich dadurch verstärkt, dass isolierte Knoten, die für sich selbst eine Komponente bzw. ein Cluster darstellen, mit Wahrscheinlichkeit 1 zusammenhängend sind.

Ein Ansatz um dies zu vermeiden, wäre die Betrachtung komplexerer Qualitätskriterien analog dem Leitwert (conductance). Wie in Abschnitt 2.3 garantieren aber auch diese keine sinnvolle Einteilung in „natürliche Gruppen“.

Daher wird entsprechend den Bezeichnungen in Abschnitt 2.3 für Clusterings von Graphen mit stochastisch existierenden Kanten das folgende Qualitätsmaß zugrunde gelegt:

$$h(C) = f(C) = R(C). \tag{3.40}$$

Ein hoher Wert dieses Maßes wird als notwendiges, jedoch nicht als hinreichendes Kriterium für ein intuitives bzw. relevantes Clustering angesehen. Ziele sind daher

1. die Bestimmung einer Vielzahl von Clusterings mit hohen internen Zusammenhangswahrscheinlichkeiten,
2. eine erste Bewertung der Clusterings durch ihre interne Zusammenhangswahrscheinlichkeit und
3. die anschließende Bestimmung eines oder mehrerer „optimaler“ Clusterings anhand weiterer Kriterien, die im Bezug zur exakten Aufgabenstellung bzw. zu dem modellierten Sachverhalt stehen.

Da an dieser Stelle nicht auf konkrete Beispiele oder Aufgabenstellungen eingegangen werden soll, wird im folgenden Kapitel ein Algorithmus vorgestellt, der die ersten beiden Punkte erfüllt.

Es sind auch andere Qualitätsmaße für Graphen mit stochastisch existierenden Kanten denkbar. Anstatt für den Zusammenhang der Knoten jedes Clusters nur die Intraclusterkanten zu betrachten, können beispielsweise auch alle Kanten betrachtet werden. D. h. der Zusammenhang zwischen den Knoten eines Clusters kann auch über Knoten anderer Cluster, Interclusterkanten und Intraclusterkanten anderer Cluster zustande kommen.

Außerdem kann an Stelle der Dichte der Intraclusterkanten auch die Spärlichkeit der Interclusterkanten gemessen werden. Ein mögliches Qualitätsmaß ist die Wahrscheinlichkeit, dass kein Paar von Clustern oder aber alle Cluster nicht zusammenhängend sind.

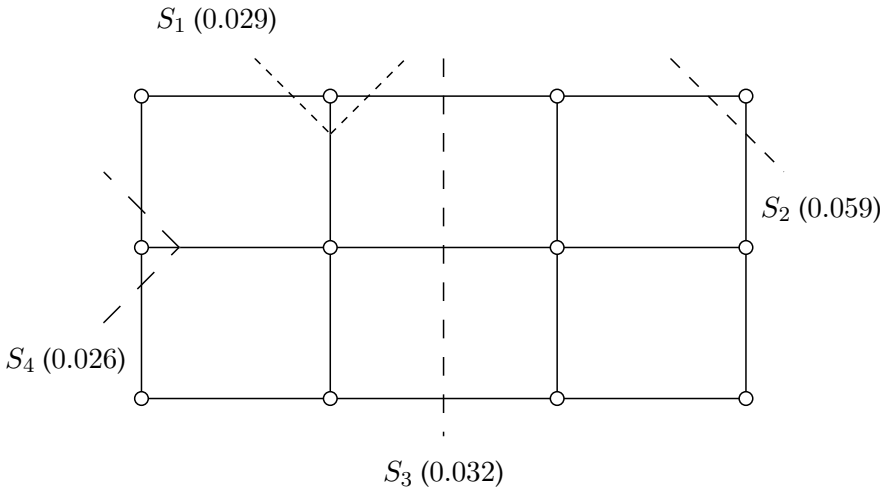


Abbildung 3.1: Ein sogenannter Gittergraph $G_{4,3}$ mit 12 Knoten, wobei alle Kanten mit Wahrscheinlichkeit $p = 0.5$ existieren. Die vier Schnitte S_1 bis S_4 entsprechen vier Clusterings des Graphen mit je zwei Clustern, in Klammern ist jeweils die interne Zusammenhangswahrscheinlichkeit der entsprechenden Clusterings angegeben. Das bezüglich der internen Zusammenhangswahrscheinlichkeit optimale Clustering entspricht dem Schnitt S_2, die interne Zusammenhangswahrscheinlichkeit des „intuitiven“ Clusterings entsprechend Schnitt S_3 ist deutlich geringer.

4 Vorstellung des Clusteringalgorithmus

In diesem Kapitel wird der entwickelte Algorithmus zur Bestimmung und Bewertung von Clusterings für Graphen mit stochastisch unabhängig existierenden Kanten vorgestellt und es werden die zur Implementation notwendigen Funktionen und Berechnungen beschrieben.

Die ersten drei Absätze behandeln nacheinander die einzelnen Schritte der Simulation: die Wahl einer Kantenpermutation, die Bestimmung eines Clusterings und die Berechnung des Schätzwertes sowie des Gewichtes einer Permutation. Im vierten Absatz wird die Schätzung der Zusammenhangswahrscheinlichkeit für die gefundenen Clusterings aus den Ergebnissen der Simulation vorgestellt.

Der Clusteringalgorithmus ist ein Simulationsalgorithmus, der als Eingaben einen Graphen mit stochastisch unabhängig existierenden Kanten $ZG = (V, E, p)$, die Anzahl der Simulationen N und die Anzahl der Cluster c, aus denen die zu bestimmenden Clusterings bestehen sollen, benötigt.

Für die Anzahl der Cluster c soll dabei gelten:

$$|V| \geq c \geq c((V, E)). \tag{4.1}$$

Die erste Ungleichung (Knotenanzahl größer gleich Clusteranzahl) ist notwendig, damit überhaupt ein Cluster erzeugt werden kann, denn im Fall von $|V| < c$ kann mindestens einem Cluster kein Knoten zugordnet werden (Widerspruch mit Formel (2.13)). Im Fall von $|V| = c$ ist keine Berechnung notwendig, da jedem Cluster genau ein Knoten zugordnet werden muss und folglich $R(C) = 1$ gilt.

Die zweite Ungleichung (Clusteranzahl größer gleich Komponentenanzahl) schließt die Erzeugung von Clusterings mit $R(C) = 0$ aus, da im Fall von $c < c((V, E))$ mindestens ein Cluster zwei im Graphen $G = (V, E)$ nicht zusammenhängende Knoten enthält und daher $R(C) = 0$ gilt. Im Fall von $c = c((V, E))$ werden jedem Cluster genau die Knoten einer Komponente zugordnet und der Algorithmus bestimmt mit $R(C)$ das Produkt der Zusammenhangswahrscheinlichkeiten der einzelnen Komponenten.

Jede der N Simulationen ($i = 1, \ldots, N$) besteht aus den folgenden Teilen:

1. Wahl einer Permutation π^i der Kanten,

2. Bestimmung des zur Permutation π^i gehörenden Clusterings C^i sowie der Anzahl der eingefügten Kanten k^i und der entsprechenden Permutation der Intraclusterkanten $\bar{\pi}^i$,
3. Berechnung eines Schätzwertes Y^i und eines Gewichtes w^i für die interne Zusammenhangswahrscheinlichkeit des Clusterings C^i unter der Permutation π^i und
4. Speicherung des Schätzwertes Y^i sowie des Gewichtes w^i für das Clustering C^i.

Es werden N nicht notwendigerweise verschiedene Clusterings $C^1, \ldots, C^N$ mit den entsprechenden Schätzwerten $Y^1, \ldots, Y^N$ und Gewichten $w^1, \ldots, w^N$ bestimmt. Aus der Menge aller Schätzwerte und Gewichte, die zu identischen Clusterings gehören, werden anschließend die Schätzungen der Zusammenhangswahrscheinlichkeiten jedes mindestens einmal erzeugten Clusterings berechnet.

Ausgegeben wird die Menge voneinander verschiedener Clusterings $\mathbf{C} = \{\mathbf{C^1}, \ldots, \mathbf{C^{N'}}\}$ sowie die Menge von Schätzungen der Zusammenhangswahrscheinlichkeit der entsprechenden Clusterings $\mathbf{Y} = \{\mathbf{Y^1}, \ldots, \mathbf{Y^{N'}}\}$, d. h. $R(\mathbf{C^i}) \approx \mathbf{Y^i} \quad \forall i \in 1, \ldots, N'$.

4.1 Wahl einer Kantenpermutation

Die Wahl einer Permutation der Kanten stellt den einzigen stochastischen Schritt des Clusteringalgorithmus dar und ist die Grundlage der Simulation. Dabei werden die Kanten nacheinander proportional zu ihrer Existenzwahrscheinlichkeit ausgewählt und anschließend aus der Menge der wählbaren Kanten entfernt. Algorithmus 4.1 zeigt die notwendigen Schritte.

Algorithmus 4.1 Wahl einer Permutation der Kanten

Eingabe: Kantenmenge E, Funktion $p : E \to (0, 1)$
Ausgabe: Permutation $\pi = (\pi_1, \ldots, \pi_m)$

1: $E' = E$
2: **for** $j = 1$ to $|E|$ **do**
3: wähle eine Kante $e \in E'$ mit Wahrscheinlichkeit $\frac{p(e)}{p(E')}$ mit $p(E') = \sum_{e \in E'} p(e)$
4: $\pi_j = e$
5: $E' = E' \setminus \{e\}$
6: **end for**

Für die Implementierung ist vor allem die Wahl einer Kante $e \in E'$ mit Wahrscheinlichkeit proportional zu ihrer Existenzwahrscheinlichkeit von Bedeutung. Dieses Problem ist in der Literatur als *nonuniform random selection* bekannt und tritt ebenso bei der Bestimmung von minimalen Schnitten in gewichteten Graphen mit dem Algorithmus von KARGER auf. Entsprechend werden verschiedenen Vorgehensweisen in [16, Seiten 213 - 218] und [19, Seiten 11, 16 - 18] ausführlich behandelt.

Eine Alternative ist die Verwendung von *Intervallbäumen*, die unter anderem in [6, Seiten 312 - 317] beschrieben werden. Jeder Kante e_i entspricht dabei ein Knoten b_i des Baumes, der das Intervall $[L(b_i), L(b_i) + p(e_i))$ repräsentiert. Dabei ist $L(b_i)$ die Summe der Existenzwahrscheinlichkeiten aller Kanten aus E', die durch Knoten links des Knotens b_i repräsentiert werden. Zur Wahl einer Kante $e \in E'$ wird eine gleichverteilte Zufallszahl z aus dem Intervall $[0, p(E'))$ generiert und der Knoten b_i bestimmt, in dessen Intervall die Zufallszahl liegt, d. h. $b_i : L(b_i) \leq z < L(b_i) + p(e_i)$. Die entsprechende Kante e_i ist die gewählte Kante für die nächste Position der Permutation.

Um bei der Bestimmung einer Permutation die mehrfache Wahl der gleichen Kante zu vermeiden kann der Intervallbaum nach jeder Wahl einer Kante aktualisiert werden. Dies ist effizient möglich, wenn anstatt der Werte $L(b_i)$ die Summen der Existenzwahrscheinlichkeiten der entsprechenden Kanten des linken Teilbaumes $L'(b_i)$ verwendet werden und bei der Suche in einem rechten Teilbaum des Knotens b_i die Zufallszahl z durch $z - p(e_i) - L'(b_i)$ ersetzt wird. Damit ist nach der Wahl einer Kante nur eine Aktualisierung der entsprechenden Vaterknoten notwendig.

4.2 Bestimmung des Clusterings

Zur Bestimmung des Clusterings mit c Clustern wird auf Grundlage der Kantenpermutation ein Graph mit c Komponenten erzeugt. Anschließend werden die Knoten einer Komponente jeweils einem Cluster zugeordnet.

Ein Graph mit c Komponenten wird analog dem Karger-Algorithmus zur Bestimmung minimaler Schnitte erzeugt, indem man Kanten vom Beginn der Permutation in den kantenleeren Graphen einfügt, bis der gewünschte Zusammenhalt erreicht ist.

Es sei $G_j = (V, \{\pi_1, \ldots, \pi_j\})$ der Graph bestehend aus der Knotenmenge V und den ersten j Kanten der Permutation π. Dann wird der Index k gesucht, so dass der Graph G_k aus c Komponenten besteht, der Graph G_{k-1} jedoch aus $c+1$ Komponenten. D. h. es wird der kleinstmögliche Index k gesucht, so dass G_k aus genau c Komponenten besteht.

Die Cluster des entsprechenden Clusterings $C = (C_1, \ldots, C_c)$ bestehen jeweils aus den Knoten, die sich in einer Komponente von G_k befinden.

Der Wert k kann ausgehend vom kantenleeren Graphen $G_0 = (V, \emptyset)$ bestimmt werden, indem für $j = 1, \ldots, m$ der Graph G_j durch Hinzufügen der Kante π_j zum Graphen G_{j-1} erzeugt wird, bis der Graph $G_j = G_k$ nur noch aus c Komponenten besteht. Die entsprechende Durchführung ist in Algorithmus 4.2 dargestellt.

Für die Implementierung ist die wiederholte Berechnung der Komponentenanzahl des Graphen G' entscheidend. Dabei kann die Tatsache genutzt werden, dass einmal in einer gemeinsamen Komponente befindliche Knoten ihren Zusammenhang behalten. Die Art und Weise dieses Zusammenhangs ist dabei uninteressant, interessant ist nur die Verteilung der

Algorithmus 4.2 Bestimmung der Anzahl einzufügender Kanten

Eingabe: Knotenmenge V, Permutation $\pi = (\pi_1, \ldots, \pi_m)$, Clusteranzahl c
Ausgabe: Anzahl eingefügter Kanten k

1: $G' = (V, E')$
2: $k = 0$
3: **while** G' hat mehr als c Komponenten **do**
4: $\quad k = k + 1$
5: $\quad E' = E' \cup \{\pi_k\}$
6: **end while**

Knoten in den einzelnen Komponenten, d. h. die Verteilung von Elementen in disjunkten Mengen.

Notwendige Operationen einer entsprechenden Datenstruktur sind der Test, ob zwei Endknoten einer Kante in der gleichen Komponente liegen (ob zwei Elemente in der gleiche Menge liegen) und das Einfügen einer Kante (die Vereinigung zweier Mengen). Dies liefert die *Union-Find-Datenstruktur* (*Disjoint Set Data Structure*), die unter anderem in [6, Kapitel 21] beschrieben wird.

Aus der Datenstuktur kann nach dem Einfügen von k Kanten (k Vereinigungen jeweils zweier Mengen) das entsprechende Clustering C direkt abgelesen werden, da die Elemente einer Menge jeweils den Knoten einer Komponente und damit den Knoten eines Clusters entsprechen.

Außerdem soll für die Berechnung der internen Zusammenhangswahrscheinlichkeit die Permutation $\bar{\pi} = (\bar{\pi}_1, \ldots, \bar{\pi}_l)$ der Intraclusterkanten von C bestimmt werden. Die Intraclusterkanten entsprechen den Kanten, die innerhalb einer Komponente von G_k verlaufen, und die Permutation $\bar{\pi}$ der Intraclusterkanten enthält alle Elemente in der gleichen Reihenfolge, wie sie in der Permutation π angeordnet sind.

Aufgrund der Entstehung des Clusterings durch das Einfügen der ersten k Kanten der Permutation π sind diese Kanten in jedem Fall Intraclusterkanten. Daher stimmen die Kanten an den ersten k Positionen der beiden Permutationen überein ($\pi_i = \bar{\pi}_i \quad \forall i = 1, \ldots, k$) und können einfach übernommen werden. Für alle weiteren Kanten muss überprüft werden, ob sich die Endknoten der jeweiligen Kante im gleichen Cluster bzw. in der gleichen Komponente von G_k befinden. Der entsprechende Ablauf ist in Algorithmus 4.3 beschrieben.

Außer der Anzahl eingefügter Kanten k und der Permutation der Intraclusterkanten $\bar{\pi}$ einschließlich der Anzahl der Intraclusterkanten l ist das bestimmte Clustering C selbst von Interesse. Dieses kann zum Beispiel in Form der Intraclusterkanten, der Interclusterkanten oder einer Knotenpartition gespeichert werden. Eine Speicherung der Knotenpartition benötigt dabei am wenigsten Speicherplatz und lässt sich aus der Union-Find-Datenstruktur

Algorithmus 4.3 Bestimmung der Anzahl einzufügender Kanten und der Permutation der Intraclusterkanten

Eingabe: Knotenmenge V, Permutation $\pi = (\pi_1, \ldots, \pi_m)$, Clusteranzahl c
Ausgabe: Anzahl eingefügter Kanten k, Permutation der Intraclusterkanten $\bar{\pi}$

```
1: E' = ∅
2: G' = (V, E')
3: k = 0
4: π̄ = ()
5: while G' hat mehr als c Komponenten do
6:    k = k + 1
7:    E' = E' ∪ {π_k}
8:    π̄_k = π_k
9: end while
10: l = k
11: for j = k + 1 to |E| do
12:    if Endknoten von π_j befinden sich in gleicher Komponente then
13:       l = l + 1
14:       π̄_l = π_j
15:    end if
16: end for
```

auslesen.

4.3 Berechnung der Werte

Ausgehend von einem Graphen mit stochastisch unabhängig existierenden Kanten $ZG = (V, E, p)$, einer Permutation der Intraclusterkanten $\bar{\pi} = (\bar{\pi}_1, \ldots, \bar{\pi}_l)$, der Anzahl der eingefügten Kanten k und der Funktion $p : E \to (0, 1)$ lassen sich die Größen $a(\bar{\pi})$, $p(\bar{\pi}, j)$, $p(\bar{\pi})$ und $q(\bar{\pi}, k)$ berechnen, aus denen im Anschluss der Schätzwert Y und das Gewicht w für die interne Zusammenhangswahrscheinlichkeit bestimmt werden.

Der Wert $p(\bar{\pi}, j)$ entspricht der Wahrscheinlichkeit, dass von den Intraclusterkanten gerade die ersten j Kanten der Permutation $\bar{\pi}$ existieren:

$$p(\bar{\pi}, j) = \prod_{i=1}^{j} p(\bar{\pi}_i) \prod_{i=j+1}^{l} [1 - p(\bar{\pi}_i)]. \tag{4.2}$$

Der Wert $a(\bar{\pi})$ entspricht dem Summanden der Permutation $\bar{\pi}$ entsprechend Sequential

Construction für die Zusammenhangswahrscheinlichkeit (Formel (3.37)):

$$a(\bar{\pi}) = \sum_{j=k}^{l} \frac{p(\bar{\pi}, j)}{j!\ (l-j)!}. \tag{4.3}$$

Der Wert $p(\bar{\pi})$ entspricht der Wahrscheinlichkeit, dass der Algorithmus 4.1 aus der Menge der Intraclusterkanten $E(C)$ gerade die Permutation $\bar{\pi}$ wählt:

$$p(\bar{\pi}) = \prod_{j=1}^{l} \frac{p(\bar{\pi}_j)}{p(E_j)}, \tag{4.4}$$

wobei

$$\begin{aligned} E_1 &= E(C), \\ E_j &= E_{j-1} \setminus \{\bar{\pi}_{j-1}\}, \\ p(E_j) &= \sum_{e \in E_j} p(e). \end{aligned}$$

Der Wert $q(\bar{\pi}, k)$ entspricht der Wahrscheinlichkeit, dass der Algorithmus 4.1 aus der Menge der Kanten E eine Permutation wählt, die entsprechend dem Algorithmus 4.3 zur Permutation der Intraclusterkanten $\bar{\pi}$ führt:

$$q(\bar{\pi}, k) = \prod_{j=1}^{k} \frac{p(\bar{\pi}_j)}{p(E_j)} \prod_{j=k+1}^{l} \frac{p(\bar{\pi}_j)}{p(F_j)}, \tag{4.5}$$

wobei

$$\begin{aligned} E_1 &= E, \\ E_j &= E_{j-1} \setminus \{\bar{\pi}_{j-1}\}, \\ p(E_j) &= \sum_{e \in E_j} p(e), \\ F_{k+1} &= \{\bar{\pi}_{k+1}, \ldots, \bar{\pi}_l\}, \\ F_j &= F_{j-1} \setminus \{\bar{\pi}_{j-1}\}, \\ p(F_j) &= \sum_{e \in F_j} p(e). \end{aligned}$$

Daraus können der Schätzwert Y sowie das Gewicht w für die interne Zusammenhangswahrscheinlichkeit des Clusterings C unter der Permutation $\bar{\pi}$ berechnet werden:

$$Y = \frac{a(\bar{\pi})}{p(\bar{\pi})}, \tag{4.6}$$

$$w = \frac{p(\bar{\pi})}{q(\bar{\pi}, k)}. \tag{4.7}$$

4.4 Schätzung der Zusammenhangswahrscheinlichkeit

Es werden die in der i-ten Simulation ($i = 1, \ldots, N$) berechneten Werte mit Y^i sowie w^i und das dabei gefundene Clustering mit C^i bezeichnet. Dann lässt sich die Zusammenhangswahrscheinlichkeit eines mindestens einmal gefundenen Clusterings $\mathbf{C}$ ($\exists i \in 1, \ldots, N : C^i = \mathbf{C}$) wie folgt schätzen:

$$R(\mathbf{C}) \approx \frac{\sum_{i=1}^{N} [C^i = \mathbf{C}]\, w^i\, Y^i}{\sum_{i=1}^{N} [C^i = \mathbf{C}]\, w^i} = \frac{\sum_{i:C^i=\mathbf{C}} w^i\, Y^i}{\sum_{i:C^i=\mathbf{C}} w^i}. \tag{4.8}$$

Anstatt die Werte Y^i und w^i sowie die Clusterings C^i während der Simulation jeweils zu speichern, bietet es sich an, nur alle voneinander verschiedenen Clusterings $\mathbf{C^1}, \ldots, \mathbf{C^{N'}}$ sowie die dazugehörigen Summen der Zähler ($\sum_{i:C^i=\mathbf{C}} w^i Y^i$) und Nenner ($\sum_{i:C^i=\mathbf{C}} w^i$) zu speichern.

Dafür wird ein assoziativer Container AC verwendet, der als Schlüssel (eindeutigen Bezeichner) das Clustering C zum Beispiel in Form der Knotenpartition nutzt und die Werte in einem Vektor $v(C) = (v_1(C), v_2(C))$ speichert. Dabei entspricht v_1 der Zählersumme und v_2 der Nennersumme.

Am Ende jeder Simulation wird geprüft, ob der Container das Clustering C^i bereits enthält. Wenn ja, werden v_1 und v_2 aktualisiert. Wenn nein, wird ein neuer Eintrag mit dem Schlüssel C^i und den entsprechenden Werten angelegt.

Der Algorithmus 4.4 fasst die einzelnen Schritte zur Bestimmung von Clusterings für einen Graphen mit stochastisch unabhängig existierenden Kanten $ZG = (V, E, p)$ zusammen.

Die erzeugten Clusterings liegen nach Abschluss der Simulation als Schlüssel in dem Container vor, die entsprechende Schätzung der Zusammenhangswahrscheinlichkeit des Clusterings C lässt sich aus den entsprechenden Werten bestimmen:

$$R(C) = \frac{v_1(C)}{v_2(C)}. \tag{4.9}$$

Als Datenstruktur für den assoziativen Container ist in der Programmiersprache C++ der Container *map* geeignet, der Bestandteil der *Standard Template Library* (*STL*) ist. Dieser Container nutzt einen *binären Baum* [6, Seiten 255 - 273], alternativ ist die Verwendung von *Hashtabellen* [6, Seiten 221 - 253] und *B-Bäumen* [6, Seiten 439 - 459] möglich.

Algorithmus 4.4 Überblick Clusteringalgorithmus

Eingabe: Graph mit stochastisch unabhängig existierenden Kanten ZG, Clusteranzahl c
Ausgabe: assoziativer Container AC

1: $AC = ()$
2: **for** $i = 1$ to N **do**
3: wähle π^i {Algorithmus 4.1}
4: bestimme C^i, $\bar{\pi}^i$, k^i {Algorithmus 4.3}
5: berechne Y^i, w^i {Formeln (4.2) bis (4.7)}
6: **if** AC enthält Schlüssel C^i **then**
7: $v_1(C^i) = v_1(C^i) + w^i\ Y^i$
8: $v_2(C^i) = v_2(C^i) + w^i$
9: **else**
10: erstelle neuen Eintrag in AC mit Schlüssel C^i
11: $v_1(C^i) = w^i\ Y^i$
12: $v_2(C^i) = w^i$
13: **end if**
14: **end for**

5 Analyse des Clusteringalgorithmus

An dieser Stelle wird der in Kapitel 4 beschriebene Algorithmus zur Bestimmung von Clusterings für Graphen mit stochastisch unabhängig existierenden Kanten analysiert. Dabei soll gezeigt werden, dass die Schätzung der Zusammenhangswahrscheinlichkeiten konsistent ist.

5.1 Analyse des Zufallsexperimentes ZE

Zunächst wird das Zufallsexperiment ZE betrachtet, dass in Algorithmus 5.1 beschrieben wird und eine geordnete Auswahl von a Kanten aus der Kantenmenge E_1 bestimmt. Als Spezialfälle sind mit $a = 1$ die Wahl einer einzelnen Kante bzw. mit $a = |E_1|$ die Wahl einer Permutation der Kantenmenge E_1 möglich.

Algorithmus 5.1 Zufallsexperiment ZE: Wahl einer geordneten Auswahl von Kanten

Eingabe: Kantenmenge E_1, Funktion $p : E_1 \to (0, 1)$
Ausgabe: geordnete Auswahl $\pi = (\pi_1, \ldots, \pi_a)$

1: $E' = E_1$
2: **for** $j = 1$ to a **do**
3: wähle ein Element $e \in E'$ mit Wahrscheinlichkeit $\frac{p(e)}{p(E')}$ mit $p(E') = \sum_{e \in E'} p(e)$
4: $\pi_j = e$
5: $E' = E' \setminus \{e\}$
6: **end for**

Das Zufallsexperiment ist eine Verallgemeinerung des in Algorithmus 4.1 vorgestellten Verfahrens zur Wahl einer Permutation der Kantenmenge E und entspricht diesem mit $E_1 = E$ sowie $a = |E_1| = |E|$.

Im Folgenden werden die Wahrscheinlichkeiten betrachtet, dass das Zufallsexperiment ZE eine einzelne Kante, eine geordnete Auswahl, eine Permutation und Permutationen mit bestimmten Eigenschaften erzeugt.

Die in Algorithmus 5.1 gestellte Forderung einer Funktion $p : E_1 \to (0, 1)$ ist nicht notwendig. Alle Ergebnisse in diesem Abschnitt gelten allgemein für Funktionen p mit $p(e) > 0 \quad \forall e \in E_1$ und damit für positive Kantenbewertungen. Zur Verkürzung wird in den folgenden Sätzen ein Graph mit stochastisch unabhängig existierenden Kanten $G =$

(V, E, p) gefordert, obwohl eine Kantenmenge mit einer positiven Bewertungsfunktion ausreichend wäre.

Es wird zunächst die Wahrscheinlichkeit betrachtet, dass das Zufallsexperiment ZE in einem beliebigen Durchlauf $j \in 1, \ldots, a$ eine bestimmte Kante e wählt:

Satz 5.1 (einmalige Auswahl durch ZE). *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch unabhängig existierenden Kanten und $E' \subseteq E$ die Menge der wählbaren Kanten. Für die Wahrscheinlichkeit, dass das Zufallsexperiment ZE in einem beliebigen Durchlauf die Kante $e \in E'$ wählt, gilt:*

$$P(\{e\}) = \frac{p(e)}{p(E')} \tag{5.1}$$

mit

$$p(E') = \sum_{e \in E'} p(e). \tag{5.2}$$

Beweis. Der Satz folgt direkt aus Algorithmus 5.1. In Zeile 1 wird die Menge E' durch die Ausgangsmenge E_1 initialisiert und während des Algorithmus durch Entfernung der gewählten Kante in Zeile 5 aktualisiert, d. h. E' enthält bei der Wahl einer Kante in Zeile 3 die Menge der wählbaren Kanten. Die Wahrscheinlichkeit für die Wahl einer Kante $e \in E'$ entspricht gerade der in Zeile 3 angegebenen Wahrscheinlichkeit. □

Satz 5.2 (geordnete Auswahl durch ZE). *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch unabhängig existierenden Kanten und $E_1 = \{\pi_1, \ldots, \pi_b\} \subseteq E$ eine Menge von Kanten. Für die Wahrscheinlichkeit $p(\pi)$, dass das Zufallsexperiment ZE die geordnete Auswahl $\pi = (\pi_1, \ldots, \pi_a)$ mit $1 \leq a \leq b$ aus der Menge E_1 bestimmt, gilt:*

$$P(\{\pi\}) = p(\pi) = \prod_{j=1}^{a} \frac{p(\pi_j)}{p(E_j)}, \tag{5.3}$$

wobei

$$E_1 = \{\pi_1, \ldots, \pi_b\},$$

$$E_j = E_{j-1} \setminus \{\pi_{j-1}\} = \{\pi_j, \ldots, \pi_b\} \quad \forall 1 < j \leq a,$$

$$p(E_j) = \sum_{e \in E_j} p(e).$$

Beweis. Entsprechend Satz 5.1 wird für jede Auswahl einer Kante π_j durch ZE eine Kante e aus der Menge der wählbaren Kanten E' mit der Wahrscheinlichkeit $\frac{p(e)}{P(E')}$ mit $P(E') = \sum_{e \in E'} p(e)$ gewählt und die Kante e daraufhin aus der Menge der wählbaren Kanten entfernt.

Es ist E_1 die Menge aller in der ersten Auswahl wählbaren Kanten und E_j die Menge aller in der j-ten Auswahl wählbaren Kanten, da E_j aus der Menge der in der vorausgegangenen Auswahl wählbaren Kanten E_{j-1} durch Entfernung der in der vorausgegangenen Auswahl gewählten Kante π_{j-1} entsteht.

Die Menge E_j enthält genau die nach $j-1$ vorangegangenen Auswahlen für die Positionen $1, \ldots, j-1$ wählbaren Kanten. Für die Wahrscheinlichkeit die Kante $e \in E_j$ in der j-ten Auswahl zu wählen, gilt:

$$P(\{\pi_j = e\}) = \frac{p(e)}{p(E_j)}.$$

Für die Wahrscheinlichkeit, die geordnete Auswahl π durch Auswahl der Kante π_1 in der ersten Auswahl, der Kante π_2 in der zweiten Auswahl, …, der Kante π_a in der a-ten Auswahl zu wählen, folgt:

$$\begin{aligned} P(\{\pi\}) &= \frac{p(\pi_1)}{p(E_1)} \frac{p(\pi_2)}{p(E_2)} \cdots \frac{p(\pi_a)}{p(E_a)} \\ &= \prod_{j=1}^{a} \frac{p(\pi_j)}{p(E_j)} \end{aligned}$$

□

Satz 5.3 (Permutation durch ZE). *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch unabhängig existierenden Kanten und $E_1 = \{\pi_1, \ldots, \pi_b\} \subseteq E$ eine Menge von Kanten. Für die Wahrscheinlichkeit $p(\pi)$, dass das Zufallsexperiment ZE die Permutation $\pi = (\pi_1, \ldots, \pi_b)$ aus der Menge E_1 erzeugt, gilt:*

$$P(\{\pi\}) = p(\pi) = \prod_{j=1}^{b} \frac{p(\pi_j)}{p(E_j)}, \tag{5.4}$$

wobei

$$\begin{aligned} &E_1 = \{\pi_1, \ldots, \pi_b\}, \\ &E_j = E_{j-1} \setminus \{\pi_{j-1}\} = \{\pi_j, \ldots, \pi_b\} \quad \forall 1 < j \leq b, \\ &p(E_j) = \sum_{e \in E_j} p(e). \end{aligned}$$

Beweis. Die Aussage folgt direkt aus Satz 5.2 mit $a = b$, da eine geordnete Auswahl von b Kanten aus einer Menge von b Kanten gerade einer Permutation der Kanten entspricht. □

Folgerung 5.4. *Bei der Wahl einer Permutation aller Kanten des Graphen mit stochastisch unabhängig existierenden Kanten $G = (V, E, p)$ gilt Satz 5.3 mit $b = |E| = m$ und $E_1 = E$.*

Folgerung 5.5. *Bei der Wahl einer Permutation aller Intraclusterkantenindizes des Clusterings C gilt Satz 5.3 mit $b = |E(C)| = l$ und $E_1 = E(C)$.*

Definition 5.6 (eingeschränkte geordnete Auswahl / Permutation). *Es seien M sowie A Kantenmengen und π eine geordnete Auswahl / Permutation der Menge M. Unter $\pi' = \pi|_A$, der auf A **eingeschränkten geordneten Auswahl / Permutation** π, versteht man die geordnete Auswahl / Permutation $\pi' = (\pi'_1, \ldots, \pi'_a)$ mit*

$$\pi'_i \in A \quad \forall 1 \leq i \leq a, \tag{5.5}$$

$$\pi^{-1}(\pi'_1) < \cdots < \pi^{-1}(\pi'_a). \tag{5.6}$$

Dabei ist π^{-1} die inverse Permutation zur Permutation π und $\pi^{-1}(\pi'_i)$ die Position der Kante π'_i in der Permutation π. Die Formel (5.6) *fordert daher, dass alle Kanten der geordneten Auswahl / Permutation π' in der geordneten Auswahl / Permutation π' in der gleichen Reihenfolge vorkommen wie in der geordneten Auswahl / Permutation π.*

Satz 5.7 (eingeschränkte Permutation durch ZE). *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch unabhängig existierenden Kanten, $A \subseteq E$ und $B \subseteq E$ disjunkte Kantenmengen mit $A \cup B = M$ und $|A| = u$, $|B| = v$, $|M| = u + v = w$. Weiterhin seien $\pi' = (\pi'_1, \ldots, \pi'_u)$ eine Permutation der Menge A und $\pi = (\pi_1, \ldots, \pi_w)$ eine Permutation der Menge M. Die Wahrscheinlichkeit, dass das Zufallsexperiment ZE eine Permutation π der Menge M erzeugt, so dass $\pi|_A = \pi'$, entspricht der Wahrscheinlichkeit, dass das Zufallsexperiment ZE die Permutation π' der Menge A erzeugt, d. h.:*

$$P(\{\pi \mid \pi|_A = \pi'\}) = P(\{\pi'\}) = p(\pi') = \prod_{i=1}^{u} \frac{p(\pi'_i)}{p(A'_i)}, \tag{5.7}$$

wobei

$$A'_1 = A,$$

$$A'_i = A'_{i-1} \setminus \{\pi'_{i-1}\} = \{\pi'_i, \ldots, \pi'_u\} \quad \forall 1 < i \leq u,$$

$$p(A'_i) = \sum_{e \in A'_i} p(e).$$

Beweis. Es seien $M_j \subseteq M$ $(1 \leq j \leq w)$ die Mengen der bei der Wahl der Permutation π von M in der j-ten Auswahl noch wählbaren Kanten, d. h. $M_1 = M$ und $M_j = M_{j-1} \setminus \pi_{j-1} \quad \forall 1 < j \leq w$.

Es seien A_j bzw. B_j $(1 \leq j \leq w)$ die Teilmengen der Kantenmengen A bzw. B mit den bei der Wahl der Permutation π von M in der j-ten Auswahl noch wählbaren Kanten, d. h. $A_j = M_j \cap A$ und $B_j = M_j \cap B$. Es gilt:

$$A_j \cup B_j = (M_j \cap A) \cup (M_j \cap B) = M_j \cap (A \cup B) = M_j \cap M = M_j$$

und

$$p(M_j) = p(A_j \cup B_j) = \sum_{e \in A_j \cup B_j} p(e) = \sum_{e \in A_j} p(e) + \sum_{e \in B_j} p(e) = p(A_j) + p(B_j).$$

Es wird das zweistufige Zufallsexperiment ZE^* betrachtet, das bei der Auswahl der j-ten Kante zunächst mit Wahrscheinlichkeit $\frac{p(A_j)}{p(A_j)+p(B_j)}$ bzw. $\frac{p(B_j)}{p(A_j)+p(B_j)}$ die Entscheidung für eine der Mengen A_j bzw. B_j trifft und anschließend eine Kante $a \in A_j$ bzw. $b \in B_j$ mit Wahrscheinlichkeit $\frac{p(a)}{p(A_j)}$ bzw. $\frac{p(b)}{p(B_j)}$ wählt.

Für die Wahrscheinlichkeit, dass bei der Wahl einer Permutation π der Menge M durch ZE^* in der j-ten Auswahl eine Kante $a \in A_j$ bzw. $b \in B_j$ gewählt wird, gilt:

$$P(\pi_j = a) = \frac{p(A_j)}{p(A_j) + p(B_j)} \frac{p(a)}{p(A_j)} = \frac{p(a)}{p(A_j) + p(B_j)} = \frac{p(a)}{p(M_j)},$$

$$P(\pi_j = b) = \frac{p(B_j)}{p(A_j) + p(B_j)} \frac{p(b)}{p(B_j)} = \frac{p(b)}{p(A_j) + p(B_j)} = \frac{p(b)}{p(M_j)}.$$

Mit Satz 5.1 folgt, dass die Wahrscheinlichkeiten für die Wahl einer Kante $e \in M_j = A_j \cup B_j$ durch die Zufallsexperimente ZE und ZE^* identisch sind. Daher kann ZE^* anstatt von ZE betrachten werden, ohne dass sich die Wahrscheinlichkeiten für die Wahl einer bestimmten Permutation ändern und es gilt:

$$P_{ZE}(\{\pi \mid \pi|_A = \pi'\}) = P_{ZE^*}(\{\pi \mid \pi|_A = \pi'\}).$$

Die Wahl einer Permutation π durch ZE^*, so dass $\pi|_A = \pi'$, hängt nur von der Wahl der Kanten ab, die aus den Mengen A_j gewählt werden. Da während einer Wahl der Permutation π alle Kanten der Menge A gewählt werden, bildet die geordnete Auswahl der Kanten, die aus den Mengen A_j gewählt werden, eine Permutation der Menge A. Diese Permutation ist $\pi' = (\pi'_1, \ldots, \pi'_u)$.

Die Wahrscheinlichkeit, die Kante $a \in A'_i$ bei der i-ten Wahl einer Kante aus den Mengen A_j zu wählen $(\pi'_i = a)$, beträgt $\frac{p(a)}{p(A'_i)}$, wobei A'_i die Menge der noch wählbaren Kanten bei

der i-ten Wahl einer Kante aus den Mengen A_j ist, d. h. $A'_1 = A$, $A'_i = A'_{i-1} \backslash \pi'_{i-1} \quad \forall 1 < i \leq u$ und $p(A'_i) = \sum_{e \in A'_i} p(e)$.

Für die Wahrscheinlichkeit, durch ZE^* eine Permutation π' bei der Auswahl der Kanten aus den Mengen A_j zu wählen, und damit für die Wahrscheinlichkeit, eine Permutation π mit $\pi|_A = \pi'$ durch ZE^* bzw. ZE zu wählen, folgt:

$$P(\{\pi \mid \pi|_A = \pi'\}) = \prod_{i=1}^{u} \frac{p(\pi'_i)}{p(A'_i)}.$$

Diese Wahrscheilichkeit entspricht nach Satz 5.3 gerade der Wahrscheinlichkeit für die Wahl einer Permutation π' aus der Menge $A'_1 = A$. □

Definition 5.8 (k-ähnliche Permutation). *Es seien $\pi = (\pi_1, \ldots, \pi_m)$ eine Permutation der Kantenmenge E, $\bar{\pi} = (\bar{\pi}_1, \ldots, \bar{\pi}_l)$ eine Permutation der Kantenmenge $E' \subseteq E$ und $k \in \mathbb{N}, 0 < k \leq l \leq m$. Es ist die Permutation $\bar{\pi}$ der Permutation π k-ähnlich bzw. es gilt $\pi \cong_k \bar{\pi}$ g.d.w.*

1. *die ersten k Positionen von π mit $\bar{\pi}$ übereinstimmen, d. h. $\bar{\pi}_i = \pi_i \quad \forall 1 \leq i \leq k$ und*
2. *die weiteren Kanten von $\bar{\pi}$ in der gleichen Reihenfolge in π vorkommen, d. h.*

$$(\pi_{k+1}, \ldots, \pi_m)|_{\{\bar{\pi}_{k+1}, \ldots, \bar{\pi}_l\}} = (\bar{\pi}_{k+1}, \ldots, \bar{\pi}_l).$$

Satz 5.9. *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch unabhängig existierenden Kanten, $\bar{\pi} = (\bar{\pi}_1, \ldots, \bar{\pi}_l)$ eine Permutation der Kanten $E(C) = \{\bar{\pi}_1, \ldots, \bar{\pi}_l\} \subseteq E$, $\pi = (\pi_1, \ldots, \pi_m)$ eine Permutation aller Kanten E von ZG und $k \in \mathbb{N}, 0 < k \leq l \leq m$. Für die Wahrscheinlichkeit $q(\bar{\pi}, k)$, dass das Zufallsexperiment ZE eine Permutation π aller Kanten E von ZG erzeugt, so dass $\pi \cong_k \bar{\pi}$, gilt:*

$$P(\{\pi \mid \pi \cong_k \bar{\pi}\}) = q(\bar{\pi}, k) = \prod_{j=1}^{k} \frac{p(\bar{\pi}_j)}{p(E_j)} \prod_{j=k+1}^{l} \frac{p(\bar{\pi}_j)}{p(F_j)}, \tag{5.8}$$

wobei

$$\begin{aligned}
&E_1 = E = \{\pi_1, \ldots, \pi_m\},\\
&E_j = E_{j-1} \setminus \{\bar{\pi}_{j-1}\} = E \setminus \{\bar{\pi}_1, \ldots, \bar{\pi}_{j-1}\} \quad \forall 1 < j \leq k+1,\\
&p(E_j) = \sum_{e \in E_j} p(e),\\
&F_{k+1} = \{\bar{\pi}_{k+1}, \ldots, \bar{\pi}_l\},\\
&F_j = F_{j-1} \setminus \{\bar{\pi}_{j-1}\} = \{\bar{\pi}_j, \ldots, \bar{\pi}_l\} \quad \forall k+1 < j \leq m,\\
&p(F_j) = \sum_{f \in F_j} p(f).
\end{aligned}$$

Beweis. Es wird das Zufallsexperiment ZE als zweistufiges Zufallsexperiment ZE' betrachtet, indem zunächst eine geordnete Auswahl $(\pi_1, \ldots, \pi_k)$ aus der Menge E getroffen wird (ZE'_1), anschließend eine Permutation $(\pi_{k+1}, \ldots, \pi_m)$ der verbleibenden Kanten gewählt wird (ZE'_2) und beide zu einer Permutation $\pi = (\pi_1, \ldots, \pi_k, \pi_{k+1}, \ldots, \pi_m)$ zusammengesetzt werden.

Bei der Wahl der Permutation π durch das Zufallsexperiment ZE bzw. ZE' sind die Mengen der wählbaren Kanten E_j und damit auch die Wahrscheinlichkeiten für die Wahl einer bestimmten Kante π_j in der j-ten Auswahl identisch. D. h. beide Zufallsexperimente wählen die Permutation π mit der gleichen Wahrscheinlichkeit $P(\pi) = P_{ZE}(\pi) = P_{ZE'}(\pi)$.

Es sei A das Ereignis, dass das Zufallsexperiment ZE' eine Permutation $\pi = (\pi_1, \ldots, \pi_m)$ aller Kanten E von ZG erzeugt, so dass die Bedingung 1 aus Definition 5.8 erfüllt ist. B sei das Ereignis, dass das Zufallsexperiment ZE' eine Permutation $\pi = (\pi_1, \ldots, \pi_m)$ aller Kanten E von ZG erzeugt, so dass die Bedingung 2 aus Definition 5.8 erfüllt ist. Es gilt:

$$q(\bar{\pi}, k) = P(A \cup B)$$

und nach dem *Multiplikationssatz für Ereignisse* [22, Seite 120]:

$$q(\bar{\pi}, k) = P(A \cup B) = P(A)\ P(B/A).$$

Die Bedingung 1 ist nur von der Wahl der ersten k Kanten abhängig, d. h. das Ereignis A tritt genau dann ein, wenn das Zufallsexperiment ZE'_1 die geordnete Auswahl $(\bar{\pi}_1, \ldots, \bar{\pi}_k)$ aus der Menge E bestimmt. Für die Wahrscheinlichkeit $P(A)$ gilt daher nach Satz 5.2:

$$P(A) = \prod_{j=1}^{k} \frac{p(\bar{\pi}_j)}{p(E_j)}.$$

Ist das Ereignis A eingetreten, dann wurde die geordnete Auswahl $(\bar{\pi}_1, \ldots, \bar{\pi}_k)$ gewählt und die Menge der noch wählbaren Kanten entspricht der Menge E_{k+1}.

Ist das Ereignis A eingetreten, dann ist die Bedingung 2 nur von der Wahl der letzten $m-k$ Kanten abhängig, d. h. das Ereignis B/A tritt genau dann ein, wenn das Zufallsexperiment ZE'_2 eine Permutation $(\pi_{k+1}, \ldots, \pi_m)$ aus der Menge E_{k+1} bestimmt, so dass

$$(\pi_{k+1}, \ldots, \pi_m)|_{\{\bar{\pi}_{k+1}, \ldots, \bar{\pi}_l\}} = (\bar{\pi}_{k+1}, \ldots, \bar{\pi}_l).$$

Für die Wahrscheinlichkeit $P(B/A)$ gilt daher nach Satz 5.9:

$$P(B/A) = \prod_{j=k+1}^{l} \frac{p(\bar{\pi}_j)}{p(F_j)}.$$

Es folgt:

$$q(\bar{\pi}, k) = P(A)\ P(B/A) = \prod_{j=1}^{k} \frac{p(\bar{\pi}_j)}{p(E_j)} \prod_{j=k+1}^{l} \frac{p(\bar{\pi}_j)}{p(F_j)}.$$

□

Die Ergebnisse aus der Betrachtung des Zufallsexperimentes ZE werden in zwei Folgerungen zusammengefasst. Zum einen wird die Wahrscheinlichkeit benötigt, mit der eine Permutation der Intraclusterkanten gewählt werden würde. Zum anderen wird die Wahrscheinlichkeit benötigt, mit der eine Permutation der Intraclusterkanten indirekt (durch die Wahl einer Permutation der Kanten und die Bestimmung der Permutation der Intraclusterkanten entsprechend Algorithmus 4.3) gewählt wird.

Folgerung 5.10. *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch unabhängig existierenden Kanten, C ein Clustering des Graphen ZG mit den Intraclusterkanten $E(C) = \{\bar{\pi}_1, \ldots, \bar{\pi}_l\}$. Die Wahrscheinlichkeit, dass das Zufallsexperiment ZE bzw. der Algorithmus 4.1 die Permutation $\bar{\pi}$ der Intraclusterkanten $E(C)$ wählt, beträgt*

$$P(\{\bar{\pi}\}) = p(\bar{\pi}) = \prod_{j=1}^{l} \frac{p(\bar{\pi}_j)}{p(E_j)}, \tag{5.9}$$

wobei

$$E_1 = E(C) = \{\bar{\pi}_1, \ldots, \bar{\pi}_l\},$$

$$E_j = E_{j-1} \setminus \{\bar{\pi}_{j-1}\} = \{\bar{\pi}_j, \ldots, \bar{\pi}_l\} \quad \forall 1 < j \leq a,$$

$$p(E_j) = \sum_{e \in E_j} p(e)$$

und entspricht damit der Formel (4.4).

Folgerung 5.11. *Es seien $ZG = (V, E, p)$ ein Graph mit stochastisch unabhängig existierenden Kanten, C ein Clustering des Graphen ZG mit c Clustern und den Intraclusterkanten $E(C) = \{\pi_1, \ldots, \pi_l\}$. Die Wahrscheinlichkeit, dass das Zufallsexperiment ZE bzw. der Algorithmus 4.1 eine Permutation π der Kantenmenge E wählt, so dass entsprechend Algorithmus 4.3 das Clustering C sowie die Permutation $\bar{\pi}$ der Intraclusterkanten durch das Einfügen der ersten k Kanten entsteht, beträgt*

$$P(\{\pi \mid \pi \cong_k \bar{\pi}\}) = q(\bar{\pi}, k) = \prod_{j=1}^{k} \frac{p(\bar{\pi}_j)}{p(E_j)} \prod_{j=k+1}^{l} \frac{p(\bar{\pi}_j)}{p(F_j)}, \tag{5.10}$$

wobei

$$E_1 = E,$$

$$E_j = E_{j-1} \setminus \{\bar{\pi}_{j-1}\},$$

$$p(E_j) = \sum_{e \in E_j} p(e),$$

$$F_{k+1} = \{\bar{\pi}_{k+1}, \ldots, \bar{\pi}_l\},$$

$$F_j = F_{j-1} \setminus \{\bar{\pi}_{j-1}\} = \{\bar{\pi}_j, \ldots, \bar{\pi}_l\} \quad \forall k+1 < j \leq m,$$

$$p(F_j) = \sum_{f \in F_j} p(f)$$

und entspricht damit der Formel (4.5).

5.2 Analyse der Schätzung der Zusammenhangswahrscheinlichkeit

Zunächst wird betrachtet, wie die interne Zusammenhangswahrscheinlichkeit eines Clusterings durch Sequential Construction bestimmt werden kann.

Eine Schätzung für die interne Zusammenhangswahrscheinlichkeit wird immer dann berechnet, wenn das entsprechende Clustering „gefunden“, d. h. durch Algorithmus 4.2 bestimmt wurde.

Die interne Zusammenhangswahrscheinlichkeit $R(C)$ des Clusterings $C = \{C_1, \ldots, C_c\}$ entspricht nach Definition 3.26 der Wahrscheinlichkeit, dass die Knoten der einzelnen Cluster jeweils durch Intraclusterkanten verbunden sind und nach Folgerung 3.27 der Wahrscheinlichkeit, dass der durch die Intraclusterkanten induzierte Untergraph aus genau c Komponenten besteht:

$$R(C) = \prod_{i=1}^{l} R(ZG[C_i]) = R_c(ZG[E(C)]). \tag{5.11}$$

Für die exakte Berechnung der c-Komponenten-Zusammenhangswahrscheinlichkeit von

$ZG[E(C)]$ gilt entsprechend den Formeln (3.34) und (3.34):

$$R(C) = R_c(ZG[E(C)]) = \sum_{\bar{\pi}\in\bar{\Pi}} \sum_{j=0}^{l} \frac{\phi_c(z(\bar{\pi}, j))\ P(z(\bar{\pi}, j))}{j!\ (l-j)!}, \tag{5.12}$$

$$= \sum_{\bar{\pi}\in\bar{\Pi}} \sum_{j=k(\bar{\pi})}^{l} \frac{P(z(\bar{\pi}, j))}{j!\ (l-j)!}, \tag{5.13}$$

$$= \sum_{\bar{\pi}\in\bar{\Pi}} a(\bar{\pi}) \text{ mit } a(\bar{\pi}) = \sum_{j=k(\bar{\pi})}^{l} \frac{P(z(\bar{\pi}, j))}{j!\ (l-j)!}. \tag{5.14}$$

Dabei ist $\bar{\Pi}$ die Menge aller Permutationen der Intraclusterkanten $E(C)$, $z(\bar{\pi}, j)$ bezieht sich auf $ZG[E(C)]$ und $k(\bar{\pi})$ ist der kleinste Index j, für den $\phi_c(z(\bar{\pi}, j)) = 1$.

Wie bereits in Punkt 3.3.4 beschrieben, lässt sich diese Summe durch Monte-Carlo-Simulation näherungsweise bestimmen. Es seien $\bar{\pi}^{(1)}, \ldots, \bar{\pi}^{(N)}$ eine Menge zufällig und unabhängig mit Verteilungsfunktion $p(\bar{\pi})$ gewählter Permutationen der Intraclusterkanten $E(C)$, so dass $\sum_{\bar{\pi}\in\bar{\Pi}} p(\bar{\pi}) = 1$ und $p(\bar{\pi}) > 0 \quad \forall \bar{\pi} \in \bar{\Pi}$. Dann lässt sich die Zusammenhangswahrscheinlichkeit des Clusterings C entsprechend der Formeln (3.36) und (3.37) wie folgt schätzen:

$$R(C) = R_c(ZG[E(C)]) \approx \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{p(\bar{\pi}^i)} \sum_{j=0}^{l} \frac{\phi_c(z(\bar{\pi}, j))\ P(z(\bar{\pi}^i, j))}{j!(l-j)!} \right], \tag{5.15}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{p(\bar{\pi}^i)} \sum_{j=k(\bar{\pi}^{(i)})}^{l} \frac{P(z(\bar{\pi}^i, j))}{j!(l-j)!} \right], \tag{5.16}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \frac{a(\bar{\pi}^i)}{p(\bar{\pi}^i)} \text{ mit } a(\bar{\pi}^{(i)}) = \sum_{j=k(\bar{\pi}^{(i)})}^{l} \frac{P(z(\bar{\pi}^{(i)}, j))}{j!\ (l-j)!}. \tag{5.17}$$

Für die Bestimmung der internen Zusammenhangswahrscheinlichkeit eines Clusterings C im Rahmen des Clusteringalgorithmus ist diese Schätzung jedoch nicht anwendbar, da die Permutation der Intraclusterkanten $\bar{\pi}^{(i)}$ nicht direkt durch das Zufallsexperiment ZE bestimmt wird, sondern aus einer Permutation aller Kanten abgeleitet wird. D. h. heißt die Permutation $\bar{\pi}^{(i)}$ wird in einem Durchgang, der das Clustering C erzeugt, nicht mit Wahrscheinlichkeit $p(\bar{\pi}^{(i)})$ gemäß Folgerung 5.10, sondern mit Wahrscheinlichkeit $q(\bar{\pi}^{(i)}, k(\bar{\pi}^{(i)}))$ (Folgerung 5.11) erzeugt.

Andererseits gilt zwar $\sum_{\bar{\pi}\in\bar{\Pi}} p(\bar{\pi}^{(i)}) = 1$, aber nicht $\sum_{\bar{\pi}\in\bar{\Pi}} q(\bar{\pi}^{(i)}, k(\bar{\pi}^{(i)})) = 1$, daher ist eine direkte Schätzung der Zusammenhangswahrscheinlichkeit $R(C)$ bzw. der Summe

$\sum_{\bar{\pi}\in\bar{\Pi}} a(\bar{\pi})$ mit

$$R(C) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{a(\bar{\pi}^i)}{q(\bar{\pi}^i, k(\bar{\pi}^i))} \tag{5.18}$$

nicht möglich.

Somit ist der „Umweg“ über die Formeln des Importance Sampling notwendig. Nach Folgerung 2.1 gilt für eine Schätzung von $R(C)$:

$$R(C) = R_c(ZG[E(C)]) \approx \frac{\sum_{i=1}^{N}\frac{a(\bar{\pi}^i)}{q(\bar{\pi}^i, k(\bar{\pi}^i))}}{\sum_{i=1}^{N}\frac{p(\bar{\pi}^i)}{q(\bar{\pi}^i, k(\bar{\pi}^i))}}, \tag{5.19}$$

$$\approx \sum_{i=1}^{N}\frac{a(\bar{\pi}^i)}{q(\bar{\pi}^i, k(\bar{\pi}^i))} \Big/ \sum_{i=1}^{N}\frac{p(\bar{\pi}^i)}{q(\bar{\pi}^i, k(\bar{\pi}^i))}. \tag{5.20}$$

Dies entspricht gerade den Formeln (4.6) bis (4.8).

Folgerung 5.12. *Der in Kapitel 4 beschriebene Algorithmus bestimmt für alle Clusterings eine Schätzung der internen Zusammenhangswahrscheinlichkeit mittels Sequential Construction und Importance Sampling. Diese Schätzung ist konsistent.*

5.3 Komplexität

Um eine Einschätzung der Laufzeit des Algorithmus für verschiede Graphen zu erhalten, wird die Komplexität des Algorithmus betrachtet.

Die Komplexität ist von den verwendeten externen Algorithmen und Datenstrukturen abhängig, die zur Implementierung der vier Schritte einer Simulation entsprechend Kapitel 4 verwendet werden:

1. Wahl einer Permutation der Kanten,
2. Bestimmung des Clusterings (sowie der Anzahl der eingefügten Kanten und der Permutation der Intraclusterkanten),
3. Berechnung der Werte (des Schätzwertes, des Gewichtes sowie der dafür benötigten Werte) und
4. Speicherung der Werte (des Schätzwertes und des Gewichtes).

Im Folgenden wird zunächst die Komplexität der einzelnen Schritte in Abhängigkeit der Anzahl der Kanten $|E| = m$ betrachtet.

Zu 1: Wahl einer Permutation der Kanten: Sowohl die von KARGER vorgestellte Implementierung als auch die in Kapitel 4.1 vorgestellte Alternative mit Hilfe von Intervallbäumen haben eine Komplexität von $\mathcal{O}(m \log m)$.

Zu 2: Bestimmung des Clusterings: Bei Verwendung der in Kapitel 4.2 angegebenen Datenstruktur ergibt sich für alle praktischen Fälle (weniger als 10^{80} Kanten) eine lineare Komplexität von $\mathcal{O}(m)$.

Zu 3: Berechnung der Werte: Die Berechnung der Werte ist bei einer rekursiven Berechnung von $p(\bar{\pi}, j)$ durch

$$p(\bar{\pi}, j) = p(\bar{\pi}, j-1) \frac{p(\bar{\pi}_j)}{1 - p(\bar{\pi}_{j-1})} \tag{5.21}$$

mit einer Komplexität von $\mathcal{O}(m)$ möglich.

Zu 4: Speicherung der Werte: Die Komplexität für die Speicherung der Werte entspricht der Komplexität, das entsprechende Clustering zu finden oder festzustellen, dass für das Clustering noch keine Werte gespeichert wurden und ein neues Clustering einzufügen. Die Komplexität beträgt bei der Verwendung eines binären oder B-Baumes $\mathcal{O}(\log x)$, wobei x die Anzahl der zu speichernden Clusterings ist. Bei der Verwendung von Hashtabellen beträgt die Komplexität im Mittel sogar nur $\mathcal{O}(1)$. Da die Clusterings auch in Form eines (binären) Statusvektors der Intra- oder Interclusterkanten gespeichert werden können, gibt es maximal $x \leq 2^m$ verschiedene Clusterings (die Verzweigungen des Binärbaumes entsprechen jeweils der Tatsache, dass eine bestimmte Kante enthalten ist oder nicht), wobei ein bestimmtes Clustering mit einer Komplexität von $\mathcal{O}(m)$ gefunden werden kann.

Daraus folgt, dass die Laufzeit einer Simulation von der Wahl einer Permutation der Kanten dominiert wird und eine Laufzeitkomplexität von $\mathcal{O}(m \log m)$ hat. Da die Anzahl der Kanten quadratisch mit der Anzahl der Knoten $|V| = n$ steigen kann, beträgt die Komplexität einer Simulation in Abhängigkeit von der Anzahl der Knoten $\mathcal{O}(n^2 \log n)$.

Entsprechend besitzt der Algorithmus mit N Simulationen für einen Graphen mit stochastisch unabhängig existierenden Kanten mit m Kanten eine Laufzeitkomplexität von $\mathcal{O}(N\ m \log m)$.

Obwohl die Anzahl der möglichen Partitionen der Knotenmenge mit steigender Anzahl der Knoten exponentiell wächst, kann in jeder Simulation höchstens ein Clustering gefunden werden, für welches neuer Speicherplatz benötigt wird. Dieser Speicherplatz wird von der Speicherung der Knotenpartition dominiert, d. h. der Algorithmus mit N Simulationen für einen Graphen mit stochastisch unabhängig existierenden Kanten mit n Knoten besitzt eine Speicherplatzkomplexität von $\mathcal{O}(N\ n)$.

Ist die Anzahl der Simulationen jedoch linear abhängig von der Anzahl der möglichen Clusterings (ein konstanter Anteil aller Clusterings soll untersucht werden), so wächst der Speicherplatz im Allgemeinen exponentiell.

6 Praktische Ergebnisse

In diesem Kapitel werden einige praktische Ergebnisse des Clusteringalgorithmus beispielhaft vorgestellt. Es wird die Funktionsweise des Algorithmus anhand der einzelnen Schritte einer Simulation gezeigt. Weiterhin werden die Einflüsss der Anzahl der Simulationen und der Größe des Graphen untersucht.

6.1 Testgraph und Funktionsweise des Algorithmus

Zunächst werden die einzelnen in Kapitel 4 vorgestellten Schritte einer Simulation des Clusteringalgorithmus anhand der Bestimmung eines Clusterings mit 3 Clustern für den Graphen mit stochastisch unabhängig existierenden Kanten S erläutert.

Der Graph S mit 9 Knoten und 18 Kanten entspricht dem von Tittmann in [27, Seite 2] beschriebenen sozialen Netzwerk und ist in Abbildung 6.1 zu sehen. Die Existenzwahrscheinlichkeiten wurden anhand der dort vorgestellten Formel berechnet und auf zwei Dezimalstellen gerundet.

Im ersten Schritt jeder Simulation (Abschnitt 4.1) wird als Ausgangspunkt eine Permutation der Kanten des Graphen S entsprechend Algorithmus 4.1 zufällig gewählt, diese sei

$$\begin{aligned}\pi = (&\{2,7\},\{1,8\},\{4,5\},\{2,3\},\{6,9\},\{3,5\},\{6,8\},\{1,7\},\{2,5\},\\ &\{1,3\},\{1,6\},\{4,7\},\{1,2\},\{8,9\},\{1,5\},\{2,4\},\{5,7\},\{1,4\}).\end{aligned}$$

Die Permutation π wird nach Satz 5.3 mit der Wahrscheinlichkeit $p(\pi) = 7.62 * 10^{-16}$ gewählt.

Im zweiten Schritt (Abschnitt 4.2) wird das Clustering C sowie die Anzahl der einzufügenden Kanten von Beginn der Permutation und die Permutation der Intraclusterkanten bestimmt.

Von der Permutation π müssen die ersten 6 Kanten ($k = 6$) eingefügt werden, damit der entstehende Graph aus genau $c = 3$ Komponenten besteht. Dies entspricht dem Clustering

$$C = \{\{1,8\},\{2,3,4,5,7\},\{6,9\}\}.$$

Die entsprechende Permutation der Intraclusterkanten ist

$$\bar{\pi} = (\{2,7\},\{1,8\},\{4,5\},\{2,3\},\{6,9\},\{3,5\},\{2,5\},\{4,7\},\{2,4\},\{5,7\}).$$

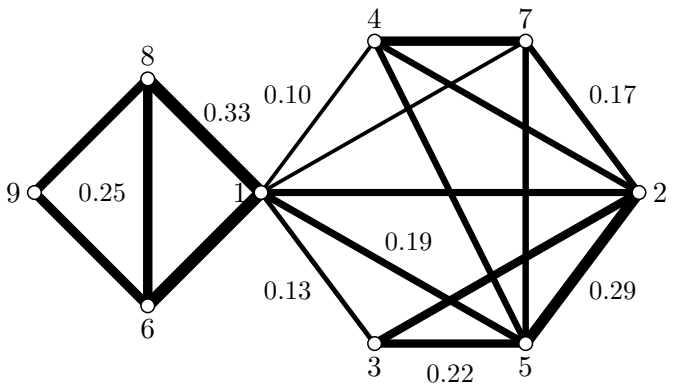


Abbildung 6.1: Der Graph mit stochastisch unabhängig existierenden Kanten S mit

$$p(\{1,4\}) = p(\{1,7\}) = 0.1,\ p(\{1,3\}) = 0.13,$$

$$p(\{2,4\}) = p(\{2,7\}) = p(\{4,5\}) = p(\{5,7\}) = 0.17,$$
$$p(\{1,2\}) = p(\{1,5\}) = 0.19,\ p(\{2,3\}) = p(\{3,5\}) = 0.22,$$
$$p(\{4,7\}) = p(\{6,8\}) = p(\{6,9\}) = p(\{8,9\}) = 0.25,$$
$$p(\{2,5\}) = 0.29 \text{ und } p(\{1,6\}) = p(\{1,8\}) = 0.33.$$

Die Beschriftung der Knoten entspricht den Knotennummern, die Beschriftung der Kanten den Existenzwahrscheinlichkeiten. Die Stärke der Kanten ist proportional zu den Existenzwahrscheinlichkeiten.

Die Abbildung 6.2 verdeutlicht dies anhand der eingefügten Kanten und der Permutation der Intraclusterkanten.

Im dritten Schritt (Abschnitt 4.3) werden die benötigten Werte für die Permutation der Intraclusterkanten $\bar{\pi}$ berechnet. (Die Werte sind jeweils gerundet.)

Für den Wert $a(\bar{\pi})$ gilt nach Formel (4.3):

$$a(\bar{\pi}) = 3.10 * 10^{-9}.$$

Für den Wert $p(\bar{\pi})$ gilt nach Formel (4.4):

$$p(\bar{\pi}) = 4.28 * 10^{-7}.$$

Für den Wert $q(\bar{\pi}, 6)$ gilt nach Formel (4.5):

$$q(\bar{\pi}, 6) = 7.59 * 10^{-9}.$$

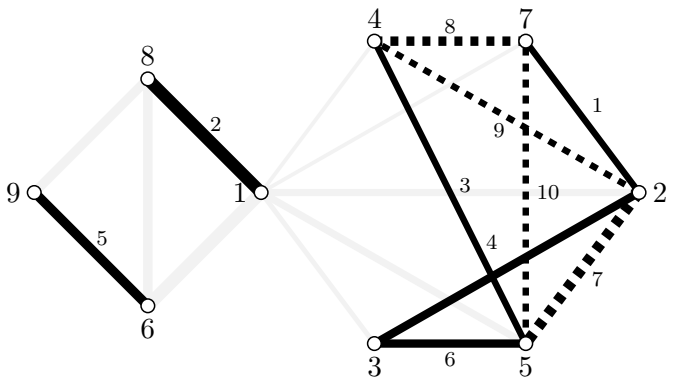


Abbildung 6.2: Die Knotenmenge V des Graphen mit stochastisch unabhängig existierenden Kanten ZG bildet gemeinsam mit den ersten 6 Kanten der Permutation π einen Graphen mit 3 Komponenten, der ein Clustering C mit 3 Clustern darstellt. Die Beschriftung der Kanten entspricht der Position der Kanten in der Permutation der Intraclusterkanten. Die eingefügten Kanten sind schwarz, die weiteren Intraclusterkanten schwarz gestrichelt und die Interclusterkanten grau gezeichnet.

Für die Werte Y und w gilt nach den Formeln (4.6) und (4.7):

$$Y = 0.00722 \text{ und}$$
$$w = 56.5.$$

Diese Werte werden im vierten Schritt gespeichert und nach der Durchführung aller Simulationen wird daraus eine Schätzung der internen Zusammenhangswahrscheinlichkeit des Clusterings C nach Formel (4.8) berechnet. Im Fall von nur einer Simulation, in der das Clustering C betrachtet wird, hat das Gewicht w keine Auswirkung und es gilt:

$$R(C) \approx 0.00722.$$

Der exakten Wert beträgt:

$$R(C) = 0.0032338.$$

6.2 Optimale Clusterings des Testgraphen

Das bezüglich der internen Zusammenhangswahrscheinlichkeit optimale Clustering mit 3 Clustern des Graphen mit stochastisch unabhängig existierenden Kanten S ist das Clustering

$$C' = \{\{1, 2, 4, 5, 6, 7, 8\}, \{3\}, \{9\}\}.$$

Die interne Zusammenhangswahrscheinlichkeit des Clusterings C' beträgt

$$R(C') = 0.0125038.$$

Dabei tritt das bereits in Abschnitt 3.4 erwähnte Problem auf, dass optimale Clusterings bezüglich der internen Zusammenhangswahrscheinlichkeit oft nur aus einem großen Cluster und einer Anzahl isolierter Knoten bestehen.

Das Clustering mit der größten internen Zusammenhangswahrscheinlichkeit und nur einem isolierten Knoten ist das Clustering

$$C'' = \{\{1, 6, 8, 9\}, \{2, 4, 5, 7\}, \{3\}\}.$$

Mit einer internen Zusammenhangswahrscheinlichkeit von

$$R(C'') = 0.00997551$$

belegt das Clustering C'' die 5. Position in der Reihenfolge der Clusterings nach der Größe der internen Zusammenhangswahrscheinlichkeit.

Das Clustering mit der größten internen Zusammenhangswahrscheinlichkeit und ohne einem isolierten Knoten ist das Clustering

$$C''' = \{\{1, 2, 3, 5\}, \{4, 7\}, \{6, 8, 9\}.$$

Mit einer internen Zusammenhangswahrscheinlichkeit von

$$R(C''') = 0.00340143$$

belegt es die 38. Position in der Reihenfolge der Clusterings nach der Größe der internen Zusammenhangswahrscheinlichkeit.

Die Clusterings C', C'' und C''' sind in Abbildung 6.3 abgebildet.

Auf den folgenden Plätzen 39 bis 42 in der Reihenfolge der Clusterings nach der internen Zusammmenhangswahrscheinlichkeit folgen 4 Clusterings mit der gleichen internen Zusammenhangswahrscheinlichkeit, wovon eines das Clustering C ist, welches im vorangegangenen Abschnitt 6.1 vorgestellt wurde.

6.3 Einflussgrößen des Algorithmus

Im Folgenden werden die beiden wichtigsten Parameter des Algorithmus auf ihren Einfluss hin untersucht:

1. die Anzahl der Simulationen (auf die Genauigkeit der Schätzung) und

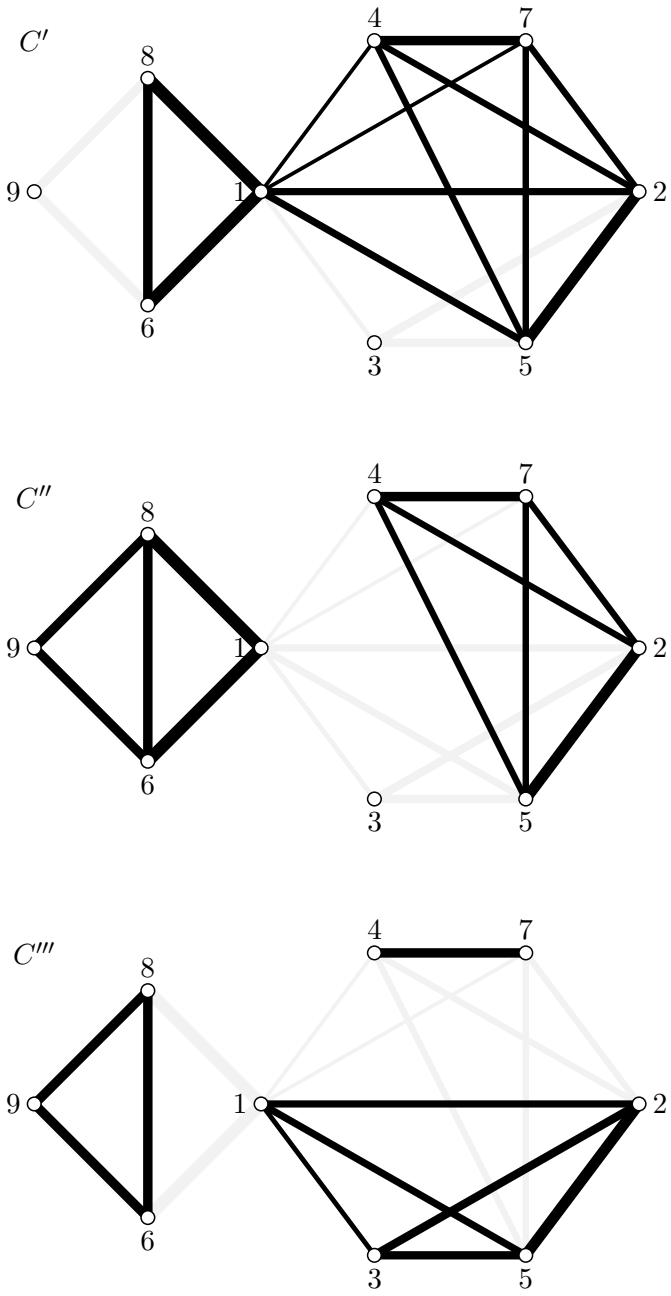


Abbildung 6.3: Die Clusterings C', C'' und C''' des Graphen mit stochastisch unabhängig existierenden Kanten S.

2. die Größe des Graphen (auf die Laufzeit).

Als Testgraph dient der in Abschnitt 6.1 vorgestellte und in Abschnitt 6.2 untersuchte Graph mit stochastisch unabhängig existierenden Kanten S.

Für beide Untersuchungen werden die Mittelwerte der folgenden Größen zur Einschätzung des gesamten Algorithmus herangezogen:

- N': Anzahl der erzeugten verschiedenen Clusterings und
- t: benötige Zeit in Sekunden.

Zur Einschätzung der Genauigkeit wird das in Abschnitt 6.1 „gefundene“ Clustering C betrachtet. Dabei werden die Mittelwerte der folgenden Größen betrachtet:

- C': Anzahl der Simulationen, die das Clustering C erzeugt haben,
- V: Varianz der Schätzung für $R(C)$,
- A_E: Abweichung der Schätzung für $R(C)$ vom exakten Wert und
- A_P: Abweichung der Position des Clusterings C in der Reihenfolge aller Clusterings nach der geschätzten internen Zusammenhangswahrscheinlichkeit vom exakten Wert. (Der „exakte Wert“ umfasst für das Clustering C den Bereich der Positionen 39 bis 42, da es 3 weitere Clusterings mit gleicher interner Zusammenhangswahrscheinlichkeit gibt.)

Die entsprechenden Mittelwerte werden mit $\bar{N}'$, $\bar{t}$, $\bar{C}'$, $\bar{A}_E$, $\bar{V}$ und $\bar{A}_P$ bezeichnet und jeweils aus den Ergebnissen von 10 Durchführungen mit den gleichen Parametern berechnet.

6.3.1 Anzahl der Simulationen

Zunächst wird der Einfluss der Anzahl der Simulationen auf die Genauigkeit der Schätzung der internen Zusammenhangswahrscheinlichkeit untersucht. Es werden Clusterings mit 3 Clustern für den Graphen S bestimmt, die Anzahl der Simulationen variiert von 10^3 bis 10^9. Die Ergebnisse der Mittelwerte von jeweils 10 Durchführungen sind in Tabelle 6.1 aufgeführt.

Die Werte lassen für das gewählte Beispiel folgende Rückschlüsse zu:

1. Die Schätzung konvergiert, d. h. mit steigender Anzahl der Simulationen nimmt die Abweichung der Schätzung vom wahren Wert ab.
2. Bereits eine geringe Anzahl an Simulationen reicht aus, um jedes Clustering zu betrachten und die Clusterings richtig einzuordnen.
3. Der lineare Zusammenhang zwischen der Anzahl der Simulationen und der Laufzeit wird bestätigt.

N	$\bar{N}'$	$\bar{t}$	$\bar{C}'$	$\bar{V}$	$\bar{A}_E$	$\bar{A}_P$
1000	157.30	0.01	6.30	$1.14E-05$	0.9094731	26.60
10000	235.20	0.12	63.50	$6.72E-07$	0.2344832	10.90
100000	246.00	1.19	623.60	$1.74E-07$	0.0996104	3.50
1000000	246.00	12.02	6384.40	$2.13E-08$	0.0314462	1.20
10000000	246.00	119.76	63694.70	$2.14E-09$	0.0129308	0.70
100000000	246.00	1143.39	636736.50	$2.11E-10$	0.0033446	0.80
1000000000	246.00	11444.10	6365495.00	$2.11E-11$	0.0009045	0.00

Tabelle 6.1: Einfluss der Anzahl der Simulationen auf die Ergebnisse der Schätzung der internen Zusammenhangswahrscheinlichkeit des Clusterings C bei der Bestimmung von Clusterings mit 3 Clustern für den Graphen S.

Die Punkte 1 und 3 belegen somit vor allem die theoretischen Aussagen aus Kapitel 5 und zeigen, dass eine entsprechende Implementierung möglich ist.

Zur Konvergenz im vorliegenden Beispiel kann gesagt werden, dass sich bei einer Verzehnfachung der Anzahl der Simulationen die Abweichung der Schätzung auf ein Drittel reduziert.

Der Punkt 2 hat vor allem praktische Bedeutung, wenn nur eine Einschätzung der Clusterings untereinander notwendig ist. In diesem Fall wird keine exakte Schätzung der internen Zusammenhangswahrscheinlichkeit, sondern nur eine gute Schätzung der Position der Clusterings in ihrer Reihenfolge nach der Größe der internen Zusammenhangswahrscheinlichkeit benötigt.

6.3.2 Größe des Graphen

Um den Einfluss der Größe des Graphen (der Anzahl der Kanten) auf die Laufzeit des Algorithmus zu untersuchen, werden Clusterings mit 4 Clustern für die vollständigen Graphen K_n mit $n \in \{5, \ldots, 15\}$ bestimmt, wobei jede Kante mit Wahrscheinlichkeit $p = 0.5$ existiert. Die Ergebnisse der Mittelwerte von jeweils 10 Durchführungen mit 10^6 Simulationen sind in Tabelle 6.2 aufgeführt.

Diese Daten entsprechen dem in Abschitt 5.3 beschriebenen Zusammenhang zwischen der Anzahl der Kanten und der Laufzeit des Algorithmus.

Ebenfalls erkennbar ist das zunächst exponentielle Wachstum der Anzahl der verschiedenen betrachteten Clusterings. Dieses Verhalten hält an, solange die Anzahl der Simulationen deutlich größer ist als die Anzahl der möglichen Clusterings. Im Anschluss wächst die Anzahl der möglichen Clusterings weiterhin exponentiell, das Wachstum der betrachteten Clusterings nimmt dagegen immer mehr ab. Damit sinkt der Anteil der „gefundenen“ Clusterings an den möglichen Clusterings deutlich.

n	m	$S_{n,4}$	$\bar{N}'$	$\bar{t}$
5	10	10	10.00	3.23
6	15	65	65.00	4.60
7	21	350	350.00	6.42
8	28	1701	1701.00	8.85
9	36	7770	7769.40	11.75
10	45	34105	31707.80	15.31
11	55	145750	86071.00	19.49
12	66	611501	148786.20	24.01
13	78	2532530	207699.70	29.34
14	91	10391745	254943.00	35.22
15	105	42355950	285901.90	41.04
16	120	171798901	303863.90	47.67
17	136	694337290	313103.00	55.33
18	153	2798806985	317753.80	63.94

Tabelle 6.2: Einfluss der Anzahl der Kanten auf die Laufzeit des Algorithmus bei der Bestimmung von Clusterings mit 3 Clustern für den vollständigen Graphen $K_n = (V, E)$ mit $|E| = \frac{n\,(n-1)}{2} = m$ und $p = 0.5$. Außerdem ist die Anzahl der möglichen Clusterings $S_{n,4}$ angegeben. (Sie entspricht der Stirlingzahl zweiter Art $S_{n,k}$ mit $k = c = 4$.)

7 Zusammenfassung

Zielstellung der Arbeit war die Entwicklung eines Algorithmus zur Bestimmung von Clusterings für Graphen mit stochastisch unabhängig existierenden Kanten und eines notwendigen Qualitätsmaßes für diese Clusterings. Das Qualitätsmaß wurde in Kapitel 3 und der Clusteringalgorithmus in Kapitel 4 vorgestellt. Eine theoretische Analyse des Algorithmus folgte in Kapitel 5 und eine Auswertung der praktischen Ergebnisse in Kapitel 6.

In den folgenden beiden Abschnitten werden die wichtigsten Eigenschaften des Qualitätsmaßes und des Algorithmus noch einmal zusammengefasst. Schließlich wird ein Ausblick auf offene Probleme und mögliche Variationen des Algorithmus gegeben.

7.1 Eigenschaften des Qualitätsmaßes

Als Qualitätsmaß für Clusterings von Graphen mit stochastisch unabhängig existierenden Kanten wurde die interne Zusammenhangswahrscheinlichkeit eines Clusterings vorgestellt. Dieses Qualitätsmaß ist eng verbunden mit der zentralen Größe dieser Klasse von Graphen, der Zusammenhangswahrscheinlichkeit, und ist unabhängig von einer speziellen Aufgabenstellung sowie einer speziellen Interpretation für Knoten und Kanten.

Eine hohe interne Zusammenhangswahrscheinlichkeit erscheit notwendig, jedoch nocht hinreichend für „gute“ Clusterings. Daraus wird die Forderung an den Algorithmus abgeleitet, nicht das optimale Clustering bezüglich des Qualitätsmaßes zu suchen, sondern eine Vielzahl von Clusterings zu bestimmen und deren interne Zusammenhangswahrscheinlichkeiten zu schätzen.

7.2 Eigenschaften des Algorithmus

Es wurde ein Algorithmus beschrieben, der eine Vielzahl von Clusterings eines Graphen mit stochastisch unabhängig existierenden Kanten bestimmt und eine Schätzung des eingeführten Qualitätsmaßes, der internen Zusammenhangswahrscheinlichkeit, für jedes „gefundene“ Clustering berechnet.

Die Bestimmung der Clusterings erfolgt dabei analog dem Algorithmus von Karger zur Bestimmung minimaler Schnitte. Daraus folgt, dass auch die in Abschnitt 2.2 vorgestellten

Abschätzungen für das Finden eines Schnittes mit minimaler Summe der Existenzwahrscheinlichkeiten der Interclusterkanten gelten.

Die interne Zusammenhangswahrscheinlichkeit der erzeugten Clusterings wird mit Importance Sampling und Sequential Construction bestimmt. Die berechnete Schätzung ist konsistent, d. h. sie konvergiert mit einer steigenden Anzahl an Simulationen gegen den wahren Wert.

Der Vorteil gegenüber dem Ansatz, zunächst eine Menge von potentiellen Clusterings zu bestimmen und anschließend deren interne Zusammenhangswahrscheinlichkeit zu bestimmen, liegt in der zweifachen Nutzung der erzeugten Kantenpermutation. Sowohl für die Bestimmung der Clusterings als auch für die Berechnung ihrer internen Zusammenhangswahrscheinlichkeit wird die gleiche Kantenpermutation als Ausgangspunkt genutzt.

Die Laufzeit des gesamten Algorithmus wird von der Wahl einer Kantenpermutation dominiert, daher verändert diese „doppelte“ Nutzung zwar nicht die Komplexität, ermöglich aber praktisch eine Halbierung der Laufzeit.

Ein weiterer Vorteil ist die genauere Schätzung „interessanterer“ Clusterings. Clusterings, die häufig im ersten Teil des Algorithmus erzeugt werden, haben eine geringe Summe der Existenzwahrscheinlichkeiten der Interclusterkanten und verfügen daher eventuell über eine hohe interne Zusammenhangswahrscheinlichkeit. Dadurch dass diese Clusterings häufiger erzeugt werden, wird auch ihre interne Zusammenhangswahrscheinlichkeit besser geschätzt.

7.3 Ausblick

Mit dem Clusteringalgorithmus für Graphen mit stochastisch unabhängig existierenden Kanten ist ein Rahmen entstanden, der den Algorithmus von Karger, Sequential Construction und Importance Sampling miteinander verbindet. Dieser Rahmen ermöglicht es, eine Kantenpermutation zu wählen und daraus ein Clustering sowie eine Schätzung für eine stochastische Eigenschaft des Clusterings abzuleiten.

Entsprechend kann jeder Teil des Algorithmus abgewandelt werden, um die Laufzeit und den Speicherplatzbedarf zu verringern, die Genauigkeit der Schätzung zu verbessern oder andere Qualitätsmaße zu berechnen. Beispielsweise sind die folgenden Variationen denkbar:

- Variation des Algorithmus zur Bestimmung der Kantenpermutation (Verwendung von $-\log(p(e_i))$ anstatt der Gewichte $p(e_i)$, „Verweigerung“ von Kanten, die zu isolierten Knoten führen),
- Verwendung von Markov-Ketten anstatt der Wahl unabhängiger Permutationen der Kanten.

Außerdem ist eine bessere Verifizierung der Ergebnisse notwendig:

- Abschätzung der Genauigkeit der Schätzung,
- Vergleich der Ergebnisse mit anderen Clusteringalgorithmen (vor allem mit dem Algorithmus aus [3]),
- Untersuchung empirischer Graphen und Bewertung der Ergebnisse unter Einbeziehung der Interpretationen.

Weiterhin sind die beschriebenen Berechnungen nur für kleine Graphen praktisch durchführbar. Für praktisch relevante Daten sollte daher eine

- Anpassung der Berechnungen für große Graphen oder
- Erstellung einfacherer Qualitätsmaße

erfolgen. Darüber hinaus sollten die Möglichkeiten für eine

- Optimierung für Graphen mit gleichwahrscheinlich existierenden Kanten und
- Erweiterung für Graphen mit stochastisch existierenden Kanten

untersucht werden.

Literaturverzeichnis

[1] Aigner, Martin: *Graphentheorie*. Teubner Studienbücher: Mathematik. Teubner, 1984.

[2] Anderson, Eric C.: *Monte Carlo Methods and Importance Sampling*. 1999.

[3] Aslam, Javed, Alain Leblanc und Clifford Stein: *A new approach to clustering*. In: Naher, Stefan und Dorothea Wagner (Herausgeber): *Algorithm Engineering: 4th International Workshop*, Band 1982 der Reihe *Lecture Notes in Computer Science*, Seiten 74–86. Springer, 2001.

[4] Ball, Michael O., Charles J. Colbourn und J. Scott Provan: *Network Reliability*. Handbooks in Operations Research and Management, 7:673–762, 1995.

[5] Brandes, Ulrik und Thomas Erlebach (Herausgeber): *Network Analysis*. Lecture Notes in Computer Science. Springer, 2005.

[6] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest und Clifford Stein: *Algorithmen - eine Einführung*. Oldenbourg, 2004.

[7] Easton, Malcolm C. und C. K. Wong: *Sequential Destruction Method for Monte Carlo Evaluation of System Reliability*. IEEE Transactions on Reliability, R-29(1):27–32, 1980.

[8] Fishman, George S.: *A Comparison of Four Monte Carlo Methods for Estimating the Probability of s-t Connectedness*. IEEE Transactions on Reliability, R-35(2):145–155, 1986.

[9] Geweke, John: *Bayesian Inference in Econometric Models Using Monte Carlo Integration*. Econometrica, 57(6):1317–1339, 1989.

[10] Glasserman, Paul: *Monte Carlo Methods in Financial Engineering*. Nummer 53 in *Stochastic Modelling and Applied Probability*. Springer, 2003.

[11] Hammersley, J. M. und D. C. Handscomb: *Monte Carlo Methods*. Methuen & Co., London, 1964.

[12] Han, Eui-Hong: *An Introduction to Cluster Analysis for Data Mining*. 2000.

[13] Hastings, W. Keith: *Monte Carlo sampling methods using Markov chains and their applications*. Biometrika, 57(1):97–109, 1970.

[14] Kalos, Mavin H. und Paula A. Whitlock: *Monte Carlo Methods.* WILEY-VCH, 2008.

[15] Karger, David R.: *Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm.* In: *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, Seiten 21–30, 1993.

[16] Karger, David R.: *Random Sampling in Graph Optimization Problems.* Doktorarbeit, Stanford University, 1995.

[17] Karger, David R.: *A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem.* SIAM Review, 43(3):499–522, 2001.

[18] Karger, David R. und Clifford Stein: *An $\widetilde{O}(n^2)$ algorithm for minimum cuts.* In: *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, Seiten 757–765, 1993.

[19] Karger, David R. und Clifford Stein: *A new approach to the minimum cut problem.* Journal of the ACM, 43(4):601–640, 1996.

[20] Levine, Matthew S.: *Experimental Study of Minimum Cut Algorithms.* Masterarbeit, Massachusetts Institute of Technology, 1997.

[21] MacKay, David J. C.: *Introduction to Monte Carlo Methods.* Learning in Graphical Models, Seiten 175–204, 1998.

[22] Meintrup, David und Stefan schäffler: *Stochastik.* Statistik und ihre Anwendugen. Springer, 2005.

[23] Metropolis, Nicholas: *The Beginning of the Monte Carlo Method.* Los Alamos Science, 15:125–130, 1987.

[24] Neal, Radford M.: *Probabilistic Inference Using Markov Chain Monte Carlo Methods.* Technischer Bericht CRG-TR-91-1, University of Toronto, 1993. University of Toronto, Department of Computer Science.

[25] Neal, Radford M.: *Annealed Importance Sampling.* Statistics and Computing, 11(2):125–139, 2001.

[26] Tittmann, Peter: *Graphentheorie.* Mathematik-Studienhilfen. Hanser, 2003.

[27] Tittmann, Peter: *Stochastische soziale Netzwerke.* 2009.