



TECHNISCHE UNIVERSITÄT CHEMNITZ

Fakultät für Informatik
Professur für Theoretische Informatik

Studienarbeit

Implementierung und Evaluierung eines Algorithmus
zur Exploration sehr großer Graphen

Markus John

Chemnitz, den 13. November 2005

Betreuer: Prof. Dr. Andreas Goerdts

Inhaltsverzeichnis

1	Einleitung	1
2	Das traceroute-Verfahren	2
2.1	Der Internetgraph	2
2.2	Funktionsweise von traceroute	2
2.3	Probleme des Verfahrens	3
2.4	Auswertung der Informationen	4
3	Die zeitbasierte Breitensuche	5
3.1	Gradverteilungen und randomisierte Graphen	5
3.2	Einführung eines Zeitfaktors	11
3.3	Erläuterungen zum Algorithmus	11
3.4	Beispiel	14
4	Implementierung	18
4.1	Konstruktion von Zufallsgraphen	18
4.2	Analyse des Suchbaums	21
5	Auswertung	24
	Literaturverzeichnis	26
	Anhang	27

1 Einleitung

In manchen Situationen ist die vollständige Kenntnisnahme eines Graphen nicht möglich. Gründe dafür können Zeit- oder Speicherplatzmangel sein, doch in einigen Fällen liegt die Ursache im Graphen selbst oder in der Umgebung, in die dieser eingebettet ist. Ein Beispiel dafür ist der sogenannte Internetgraph, auf den im zweiten Kapitel näher eingegangen wird.

Nun stellt sich die Frage, welche Eigenschaften eines Graphen sich trotz lückenhafter Information über diesen ermitteln lassen. Die vorliegende Studienarbeit möchte dies am Beispiel von Gradverteilungen der Knoten untersuchen. Eine Möglichkeit, diese Gradverteilungen zu bestimmen, ohne den Graph vollständig abzulaufen, stellt das Verfahren von traceroute dar, welches das zweite Kapitel behandelt.

Im zweiten Schritt muss untersucht werden, inwieweit die Ergebnisse von traceroute den realen Verhältnissen entsprechen. Dazu wird ein Algorithmus benötigt, der im Prinzip die gleichen Ergebnisse wie das traceroute-Verfahren liefert, sich aber im Gegensatz zu diesem besser durch mathematische Formeln beschreiben lässt. Das dritte Kapitel stellt eine Abänderung der bekannten Breitensuche vor, die dies verwirklicht.

Der Umfang der Arbeit beinhaltet neben den theoretischen Betrachtungen auch die Implementation dieser sogenannten zeitbasierten Breitensuche. Im vierten Kapitel wird auf verschiedene Probleme und Besonderheiten eingegangen, die bei der Implementierung aufgetreten sind. Unter Verwendung dieses Programms soll anschliessend untersucht werden, welche Ergebnisse traceroute auf zufälligen Graphen mit unterschiedlichen Verteilungen der Knotengrade liefern würde, um abschliessend im fünften Kapitel eine Aussage über die Qualität des traceroute-Verfahrens treffen zu können.

2 Das traceroute-Verfahren

Im Bereich der Computernetzwerke gibt es frei verfügbare Hilfsprogramme, die Administratoren im Problemfall dabei unterstützen, Fehlerquellen zu lokalisieren. Unter diesen findet sich auch ein Programm namens traceroute, welches nach Angabe eines Zielpunktes im Netzwerk dessen Erreichbarkeit testet und zusätzlich alle Zwischenpunkte auf dem passierten Weg zum Zielpunkt ausgibt. Mittels traceroute lassen sich sehr einfach Informationen über die Struktur des zugrundeliegenden Netzes sammeln.

2.1 Der Internetgraph

Von allgemeinem Interesse ist der Aufbau des größten aller Computernetzwerke, dem Internet. Da sich das Internet aus logischer Sicht her in mehrere Schichten aufteilt, muss man bei einer Untersuchung der Vernetzungsstruktur jeweils angeben, welche dieser Schichten man gerade betrachtet. Die folgenden Ausführungen beziehen sich immer auf den Internetgraphen auf Routerbene.

Dieser Graph entspricht der Ebene des Internets, auf der IP¹-Pakete verschickt werden. Im OSI² Referenzmodell ist dies die dritte Schicht, die sogenannten Netzwerkschicht. Jeder Teilnehmer des Internets mit einer IP-Adresse wird hierbei durch einen Knoten repräsentiert - ohne weitere Unterscheidung seiner Funktion (Client, Server, Router). Die Kanten des Graphen entsprechen vorhandenen direkten Verbindungen zweier Teilnehmer (d.h. ohne dritten Teilnehmer auf der Verbindungsstrecke). Dabei wird auch die technische Realisierung der Verbindung (ISDN, DSL, Ethernet, ...) ignoriert. Deshalb erhält man nun einen ungewichteten, ungerichteten Graphen, obwohl sich in der Realität die Bandbreiten und Latenzzeiten einzelner Verbindungen zum Teil sehr erheblich voneinander unterscheiden können.

2.2 Funktionsweise von traceroute

Traceroute arbeitet u.a. auf dieser IP-Schicht. Dazu muss man wissen, dass jedes IP-Paket einen TTL³-Wert besitzt. Dieser ist eine ganze Zahl zwischen 0 und 255. Der Absender eines solchen Paketes setzt den Wert - je nach verwendetem Betriebssystem - z.B. auf 64 oder 128. Passiert das Paket nun einen anderen Teilnehmer, der nicht der Empfänger ist, so verringert dieser den Wert um Eins und verschickt es weiter in Richtung Empfänger. Sollte der Wert jedoch im Laufe der Reise 0 erreichen, so wird das Paket weggeworfen und der Absender

¹internet protocol

²open systems interconnection

³time to live

darüber informiert. Hiermit vermeidet man, dass Pakete bei Fehlkonfigurationen endlos im Netz kreisen können und evtl. die Leitungen belasten.

Dieses Verhalten nutzt traceroute aus, um die gewünschten Informationen zu sammeln. Es versendet eine Reihe von Paketen, deren TTL jeweils um Eins größer ist als die des vorherigen Paketes, beginnend mit dem Wert 1. Dadurch gibt es pro Teilnehmer auf dem Weg zum Zielpunkt genau ein Paket, das an dieser Stelle den TTL-Wert 0 besitzt, weggeworfen wird und eine Benachrichtigung des Absenders verursacht. Traceroute wertet diese Benachrichtigungen aus und zeigt alle Zwischenpunkte des Weges inkl. Dauer bis zum Erreichen dieser an. Erreicht ein Paket den vom Benutzer angegebenen Zielpunkt, beendet traceroute seine Arbeit.

2.3 Probleme des Verfahrens

Es gibt nun einige prinzipielle Probleme, weshalb die von traceroute erhaltenen Informationen nicht zu 100% mit den wirklichen Verhältnissen übereinstimmen müssen. So kann man sich z.B. nicht unbedingt darauf verlassen, dass ein System den Absender benachrichtigt, falls es dessen Pakete verwirft. Desweiteren gibt es Teilnehmer, die unter verschiedenen IP-Adressen erreichbar sind, wodurch der Internetgraph zusätzliche Knoten enthalten kann.

Problematischer ist jedoch, dass der scheinbare Weg vom Absender zum Empfänger über die ausgegebenen Zwischenknoten im Netzwerk überhaupt nicht vorhanden sein muss. Ursache dafür ist, dass jeder Zwischenknoten über ein eigenes Paket ermittelt wird. Dieses Paket könnte jedoch aufgrund verschiedener Ursachen einen vollkommen anderen Weg im Netzwerk nehmen als sein Vorgänger.

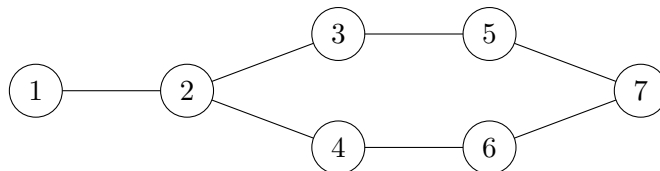


Abbildung 2.1: Beispielnetzwerk

Angenommen, das Netzwerk besitzt den in Abbildung 2.1 dargestellten Aufbau. Weiterhin wird angenommen, dass Teilnehmer 2 Pakete von 1 nach 7 abwechselnd an 3 und 4 weiterleitet. Startet nun ein Benutzer am Rechner 1 das Programm traceroute, um sich die Verbindung zum Rechner 7 anzeigen zu lassen, so erhält er z.B. folgenden Weg: 1 - 2 - 3 - 6 - 7 (das Paket mit der TTL 2 nimmt den oberen Weg, während das danach versendete Paket mit der TTL 3 über den unteren Weg vermittelt wird).

Die einzelnen Schritte von traceroute sind in Tabelle 2.1 aufgelistet. Die erste Spalte enthält dabei den TTL-Wert des gesendeten Paketes, die nächste Spalte den Weg, welcher von diesem Paket zurückgelegt wird und der Endpunkt bezeichnet den Teilnehmer des Netzwerkes, bei dem das Paket verworfen wird. Diese Spalte entspricht ebenfalls der Ausgabe von traceroute und somit dem scheinbaren Weg des Paketes im Netzwerk.

TTL	Pfad	Endpunkt
1	1 → 2	2
2	1 → 2 → 3	3
3	1 → 2 → 4 → 6	6
4	1 → 2 → 3 → 5 → 7	7

Tabelle 2.1: Verlauf von traceroute

Fasst man die Ergebnisse mehrerer traceroute-Durchläufe zusammen, könnte man in diesem Fall den Netzwerkgraph in Abbildung 2.2 erhalten. Dieses Problem lässt sich leider nicht vermeiden, weshalb eine gewisse Skepsis den ermittelten Ergebnissen gegenüber angebracht ist.

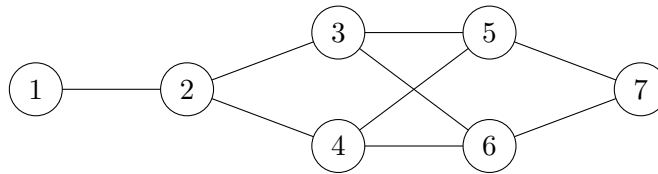


Abbildung 2.2: Fehlerhafter Netzwerkgraph

2.4 Auswertung der Informationen

Trotz der angesprochenen Probleme funktioniert traceroute in der Praxis zuverlässig. Die zurückgelieferten Wege entsprechen häufig nicht nur den realen Verhältnissen, sondern stellen oft sogar die kürzeste Verbindung von Absender und Empfänger im untersuchten Netzwerk dar. Betrachtet man nun, ausgehend von einem festen Absender, mehrere traceroute-Läufe zu verschiedenen Empfängern, so erhält man als Vereinigung dieser kürzesten Wege einen Breitensuchbaum, da von einem ungewichteten Graphen ausgegangen wird. Der Absender der traceroute-Aufrufe übernimmt hierbei die Rolle des Startknotens der Breitensuche und ist damit auch Wurzel des Baumes. Weitere Details zu diesem Verfahren sind in [5] nachzulesen.

Dieser Suchbaum stellt jedoch - je nach Anzahl der verschiedenen Empfänger - nur einen Teil des kompletten Graphen dar. Ein solches Ergebnis erhält man auch, wenn man eine Breitensuche auf einem sehr großen Graphen vorzeitig abbricht und den bis dahin konstruierten Suchbaum als Resultat ansieht. Für die weiteren Betrachtungen ist es also möglich, traceroute ganz zu vernachlässigen und dafür die Ergebnisse der Breitensuche unter verschiedenen Bedingungen zu untersuchen. Aussagen, die für die resultierenden Suchbäume getroffen werden, gelten deshalb mit Einschränkungen (die im jeweiligen Zusammenhang betrachtet werden müssen) auch für die Ergebnisse von traceroute.

3 Die zeitbasierte Breitensuche

Bevor die Breitensuche dazu verwendet werden kann, Aussagen über die Ergebnisse von trace-route zu treffen, sind noch einige Vorbetrachtungen nötig.

In den nachfolgenden Abschnitten geht es um die Eingabemenge der Breitensuche. Dabei werden Graphen betrachtet, deren Knotengrade vorgegebenen Wahrscheinlichkeitsverteilungen entsprechen. Von besonderem Interesse ist nun, wie man diese Graphen mit zufällig festgelegten Kanten erstellt.

Den zweiten Schwerpunkt im Kapitel stellt eine Abänderung des bekannten Algorithmus der Breitensuche wie in [1] beschrieben dar. Die Zielsetzung dort war es, die zu erwartende Gradverteilung im Breitensuchbaum zu berechnen. Mit der ursprünglichen Breitensuche lassen sich aber nur schwer Aussagen z.B. über die Anzahl der abgearbeiteten und entdeckten Knoten zu einem bestimmten Zeitpunkt treffen. Die Einführung eines sogenannten Zeitfaktors verringert diesen Aufwand jedoch erheblich.

Am Ende des Kapitels werden all diese Vorbetrachtungen zu einem vollständigen Algorithmus, basierend auf der ursprünglichen Breitensuche, zusammengefasst, der die Grundlage für die nachfolgende Implementierung darstellt.

3.1 Gradverteilungen und randomisierte Graphen

Definition 3.1 (Verteilung der Grade)

Ein Graph mit n Knoten besitzt die *Gradverteilung* $A = (a_0, a_1, \dots)$, wenn er genau $a_k \cdot n$ Knoten vom Grad k enthält.

Gradverteilungen sind dabei nicht zu verwechseln mit Gradsequenzen $D = (d_0, d_1, \dots, d_{n-1})$, die für jeden Knoten i dessen Grad d_i vorgeben. Am von der Breitensuche zurückgelieferten Ergebnis ist nur die Verteilung der Knotengrade im Suchbaum (im Vergleich zur Verteilung der Knotengrade im Eingabegraph) von Interesse, nicht jedoch der Verlauf der einzelnen Kanten. Es stellt sich also die Frage, ob die Breitensuche, angewandt auf einen Zufallsgraph mit z.B. einer Binomialverteilung der Knotengrade, zu einem Suchbaum mit identischer oder einer von der Ausgangsverteilung unabhängigen Verteilung führt.

Deshalb wird auch kein kompletter Graph $G = (V, E)$ als Eingabe für die Breitensuche verwendet, sondern ausschließlich die Knotenmenge V und die zugehörige Gradsequenz D seien vorgegeben. Diese wird nach dem im Abschnitt 4.1 beschriebenen Algorithmus in Anlehnung an [4] aus der Gradverteilung A generiert.

Es gibt nun zwei Alternativen, wie der Algorithmus von dieser Eingabe ausgehend einen Zufallsgraphen konstruieren kann. Einerseits ist es möglich, den Verlauf der Kanten vollständig

vor der eigentlichen Breitensuche zu bestimmen und diese in der bekannten Form auf den erhaltenen Graph anzusetzen. Eine andere Variante ist, die Kanten erst dann zu ermitteln, wenn sie in der Breitensuche benötigt werden. Dass beide Verfahren dabei äquivalente Ergebnisse liefern, lässt sich durch das Prinzip der verzögerten Entscheidung begründen. Hierfür soll jedoch zuvor geklärt werden, wie man eine zufällige Kantenmenge konstruiert, die der Eingabesequenz genügt.

Dies ist durch die Verwendung von Knotenkopien zu realisieren, man erstellt also für jeden Knoten mit dem Grad d genau d sogenannte Kopien. Für die folgenden Schritte erhält man als Repräsentation eines Knotens die Menge seiner Kopien. Nun betrachtet man die Menge aller Kopien. Diese Menge enthält eine gerade Anzahl von Elementen, denn die Summe der Knotengrade in einem ungerichteten Graph ist immer gerade (jede Kante erhöht die Grade von jeweils zwei Knoten um den Wert Eins). Teilt man die Kopienmenge in zwei disjunkte, gleichmächtige Mengen auf, so lässt sich eine bijektive Abbildung zwischen diesen angeben. Die gesuchten Kanten entsprechen gerade den Paaren der Bijektion, d.h. wenn Kopie a aus der ersten Teilmenge auf Kopie b der zweiten Teilmenge abgebildet wird, so ergibt sich für den Graph eine Kante zwischen dem Knoten der Kopie a und dem Knoten der Kopie b . Zu jeder Kopie existiert genau eine inzidente Kante, als Ergebnis erhält man damit ein perfektes Matching auf der Menge der Knotenkopien.

An die Aufteilung der Kopien in zwei disjunkte Mengen stellt man nun noch die Anforderung, dass sie gemäß der Gleichverteilung, also uniform zufällig erfolgen sollen. Für jede Kopie a ist es also gleichwahrscheinlich, in der ersten bzw. zweiten Menge enthalten zu sein und ebenso ist es für jede Kopie b der jeweils anderen Menge gleichwahrscheinlich, durch eine Kante mit Kopie a verbunden zu werden. Das Verfahren lässt sich deshalb folgendermaßen vereinfachen: solange noch unbearbeitete Kopien vorhanden sind, wählt man zwei davon zufällig gemäß der Gleichverteilung aus und fügt dem Graph eine Kante zwischen den Knoten der beiden Kopien hinzu. Alternativ werden jeweils die ersten, zweiten, dritten usw. Elemente jeder Menge aufeinander abgebildet. All diese Verfahrensweisen sind im Ergebnis identisch.

Durch die Verwendung der Knotenkopien lassen sich jedoch Schleifen und Mehrfachkanten im Graph nicht ausschliessen, d.h. Kanten von einem Knoten zu sich selbst bzw. mehrere Kanten zwischen zwei Knoten. Ergebnis ist ein uniform randomisierter Multigraph¹, bedingt auf der Sequenz der Knotengrade. Erstellt man den Graph vor der eigentlichen Breitensuche, so könnte man diese Problemfälle zwar vermeiden, jedoch wäre das Ergebnis nicht mehr identisch mit dem des zweiten Verfahrens, welches in der modifizierten Breitensuche zur Anwendung kommen wird.

Nun zurück zum Prinzip der verzögerten Entscheidung. Die Kurzform der Kantenbestimmung („wähle zwei unbearbeitete Kopien zufällig uniform . . .“) verdeutlicht, dass die Kanten jeweils unabhängig voneinander behandelt werden. Sobald durch die Ermittlung zweier Kopien eine Kante festgelegt wurden, fallen diese für alle weiteren Betrachtungen weg. Mehr noch, da jede Kopie genau eine inzidente Kante besitzt, ist es ausreichend, unter allen noch möglichen Kopien jeweils eine weitere gemäß der Gleichverteilung auszuwählen. Man erhält dadurch die benötigte inzidente Kante der ersten Kopie.

Der Zeitpunkt, zu dem dies durchgeführt wird, ist dabei unabhängig von den Zeitpunkten der restlichen Kantenbestimmungen. Es genügt also vollkommen, die Kanten eines Knotens erst

¹Graph mit Schleifen und Mehrfachkanten

zu bestimmen, wenn dieser durch die Breitensuche expandiert werden soll. Im nachfolgenden Abschnitt wird eine weitere Anpassung der Breitensuche vorgestellt, die zusammen mit dieser verspäteten Kantenbestimmung eine einfachere Analyse ermöglicht.

Definition 3.2 (angemessene Verteilung)

Seien $\alpha > 2$ und $C > 0$ konstant. Eine Gradverteilung $A = (a_0, a_1, \dots)$ nennt man *angemessen*, wenn $a_k = 0$ für $k < 3$ und $a_k < C \cdot k^{-\alpha}$ für alle $k \geq 3$ gilt.

Im Folgenden werden nur Graphen mit einer angemessenen Verteilung nach Definition 3.2 betrachtet. Dadurch stellt man sicher, dass der Graph mit hoher Wahrscheinlichkeit zusammenhängend und der Durchschnittsgrad δ endlich ist, wobei δ definiert wird als $\sum_k k a_k$.

Lemma 3.3

Zufällige Graphen gemäß der Gleichverteilung, deren Knoten einen Mindestgrad von Drei besitzen, sind mit hoher Wahrscheinlichkeit zusammenhängend.

Beweis Einen solchen Zufallsgraphen erhält man, wenn zur Ermittlung der Kanten eine zufällige Permutation aller Kopien aufgeschrieben wird. Diese Permutation muss in zwei gleich große Teile aufgeteilt werden, im Folgenden als linke und rechte Hälfte bezeichnet. Nun werden jeweils die ersten, zweiten, dritten usw. Kopien der beiden Teile durch eine Kante verbunden.

Um zu zeigen, dass dieser Graph mit hoher Wahrscheinlichkeit zusammenhängend ist, betrachtet man die Zerlegung der Knotenmenge V , $|V| = n$, in die disjunkten Mengen S und T mit $|S| = k$ sowie $|T| = n - k$. Die Anzahl der Kopien in der Menge S wird als s , die der Kopien in T als t und die Gesamtzahl aller Kopien als c mit $c = s + t$ bezeichnet.

Wenn man davon ausgeht, dass keine Kante zwischen den Mengen S und T verläuft, so müssen jeweils $\frac{s}{2}$ viele Kopien aus S bzw. $\frac{t}{2}$ viele Kopien aus T in der linken und rechten Hälfte der Permutation enthalten sein. Die erste Kopie ist mit der Wahrscheinlichkeit $\frac{s}{c}$ aus S und analog mit $\frac{t}{c}$ aus T . Nach jeder Festlegung einer Kopie verringert sich der Zähler der Wahrscheinlichkeit für weitere Kopien aus dieser Menge um Eins, ebenso der Nenner beider Brüche.

Es gibt $\binom{c/2}{s/2}$ Möglichkeiten, die Kopien aus S und T in der linken Hälfte anzuordnen. Die Wahrscheinlichkeit, dass in dieser Hälfte bei einer zufälligen Permutation $\frac{s}{2}$ viele Kopien aus S enthalten sind, beträgt deshalb

$$\binom{\frac{c}{2}}{\frac{s}{2}} \frac{s \cdot (s-1) \cdot \dots \cdot (\frac{s}{2} + 1) \cdot t \cdot (t-1) \cdot \dots \cdot (\frac{t}{2} + 1)}{c \cdot (c-1) \cdot \dots \cdot (\frac{c}{2} + 1)}.$$

Für jede Position der rechten Hälfte ist durch die Festlegung der linken Hälfte genau vorgeschrieben, aus welcher Menge die Kopie stammen muss, denn sonst könnten Kanten zwischen S und T auftreten. Die Wahrscheinlichkeiten, eine Kopie aus der jeweils geforderten Menge zu erhalten, berechnen sich wie in der linken Hälfte der Permutation.

Als resultierende Wahrscheinlichkeit einer Permutation der Kopienmenge, bei der keine Kante zwischen S und T verläuft, erhält man

$$\begin{aligned} p &= \binom{\frac{c}{2}}{\frac{s}{2}} \frac{s \cdot \dots \cdot (\frac{s}{2} + 1) \cdot t \cdot \dots \cdot (\frac{t}{2} + 1)}{c \cdot \dots \cdot (\frac{c}{2} + 1)} \frac{\frac{s}{2} \cdot \dots \cdot 1 \cdot \frac{t}{2} \cdot \dots \cdot 1}{\frac{c}{2} \cdot \dots \cdot 1} \\ &= \binom{\frac{c}{2}}{\frac{s}{2}} \frac{s! \cdot t!}{c!} = \frac{(\frac{c}{2})! \cdot s! \cdot (c-s)!}{(\frac{s}{2})! \cdot (\frac{c-s}{2})! \cdot c!}. \end{aligned} \tag{3.1}$$

3. Die zeitbasierte Breitensuche

Eine obere Abschätzung von p liefert die Stirling-Formel

$$\sqrt{2\pi n} \left(\frac{n}{e}\right)^n \leq n! \leq \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{\frac{1}{12n}},$$

indem die Fakultäten von 3.1 im Zähler nach oben und im Nenner nach unten abgeschätzt werden.

$$\begin{aligned} p &\leq \frac{\sqrt{\frac{c}{2}} \left(\frac{c}{2e}\right)^{\frac{c}{2}} e^{\frac{1}{6c}} \cdot \sqrt{s} \left(\frac{s}{e}\right)^s e^{\frac{1}{12s}} \cdot \sqrt{c-s} \left(\frac{c-s}{e}\right)^{c-s} e^{\frac{1}{12(c-s)}}}{\sqrt{\frac{s}{2}} \left(\frac{s}{2e}\right)^{\frac{s}{2}} \cdot \sqrt{\frac{c-s}{2}} \left(\frac{c-s}{2e}\right)^{\frac{c-s}{2}} \cdot \sqrt{c} \left(\frac{c}{e}\right)^c} \\ &= \sqrt{2} \cdot e^{\left(\frac{c}{12s(c-s)} + \frac{1}{6c}\right)} \cdot \frac{\left(\frac{c}{2}\right)^{\frac{c}{2}} s^s (c-s)^{c-s}}{\left(\frac{s}{2}\right)^{\frac{s}{2}} \left(\frac{c-s}{2}\right)^{\frac{c-s}{2}} c^c} \\ &= \sqrt{2} \cdot e^{\left(\frac{c}{12s(c-s)} + \frac{1}{6c}\right)} \cdot \frac{\left(\frac{c}{2}\right)^{\frac{s}{2}} \left(\frac{c}{2}\right)^{\frac{c-s}{2}} s^s (c-s)^{c-s}}{\left(\frac{s}{2}\right)^{\frac{s}{2}} \left(\frac{c-s}{2}\right)^{\frac{c-s}{2}} c^s c^{c-s}} \\ &= \sqrt{2} \cdot e^{\left(\frac{c}{12s(c-s)} + \frac{1}{6c}\right)} \cdot \left(\frac{c}{s}\right)^{\frac{s}{2}} \left(\frac{s}{c}\right)^s \left(\frac{c}{c-s}\right)^{\frac{c-s}{2}} \left(\frac{c-s}{c}\right)^{c-s} \\ &= \sqrt{2} \cdot \underbrace{e^{\left(\frac{c}{12s(c-s)} + \frac{1}{6c}\right)}}_{=:f(c,s)} \cdot \underbrace{\left(\frac{s}{c}\right)^{\frac{s}{2}} \left(\frac{c-s}{c}\right)^{\frac{c-s}{2}}}_{=:g(c,s)} \end{aligned} \quad (3.2)$$

Es gilt o.B.d.A. $s \leq \frac{c}{2}$, da der Term 3.2 symmetrisch bezüglich $s = \frac{c}{2}$ ist. Über die partiellen Ableitungen lässt sich zeigen, dass die Funktionen f und g monoton fallend sind und somit die Abschätzung von p insgesamt.

$$\frac{\partial f(c, s)}{\partial c} = -\frac{c^2 + 2(c-s)^2}{12c^2(c-s)^2} \cdot f(c, s) \leq 0$$

$$\frac{\partial f(c, s)}{\partial s} = -\frac{c(c-2s)}{12s^2(c-s)^2} \cdot f(c, s) \leq 0$$

$$\frac{\partial g(c, s)}{\partial c} = \frac{1}{2} \ln\left(\frac{c-s}{c}\right) \cdot g(c, s) \leq 0$$

$$\frac{\partial g(c, s)}{\partial s} = \frac{1}{2} \ln\left(\frac{s}{c-s}\right) \cdot g(c, s) \leq 0$$

Der Ausdruck 3.2 stellt deshalb für $c = 3n$ und $s = 3k$ ($c \geq 3n$ und $s \geq 3k$ gilt wegen gefordertem Mindestgrad) eine obere Schranke für alle p dar, die von der Anzahl der Kopien in S und T unabhängig ist.

$$p \leq \sqrt{2} \cdot e^{\left(\frac{n}{36k(n-k)} + \frac{1}{18n}\right)} \cdot \left(\frac{k}{n}\right)^{\frac{3}{2}k} \left(\frac{n-k}{n}\right)^{\frac{3}{2}(n-k)}$$

Bisher wurde bei den Berechnungen davon ausgegangen, dass die Aufteilung der Knotenmenge V vorgegeben ist. Für ein festes k existieren jedoch $\binom{n}{k}$ viele Möglichkeiten der Aufteilung von V . Die Wahrscheinlichkeit q , dass es eine Aufteilung in zwei Mengen vorgegebener Größe gibt, bei der keine Kante zwischen den Mengen verläuft, ist nach

oben begrenzt durch

$$q \leq \binom{n}{k} \cdot \sqrt{2} \cdot e^{\left(\frac{n}{36k(n-k)} + \frac{1}{18n}\right)} \cdot \left(\frac{k}{n}\right)^{\frac{3}{2}k} \left(\frac{n-k}{n}\right)^{\frac{3}{2}(n-k)}.$$

Der Binomialkoeffizient lässt sich erneut mit Stirling abschätzen und die Schritte, die schon zu 3.2 führen, liefern als Resultat

$$q \leq \frac{1}{\sqrt{\pi}} \sqrt{\frac{n}{k(n-k)}} \cdot e^{\left(\frac{n}{36k(n-k)} + \frac{5}{36n}\right)} \cdot \left(\frac{k}{n}\right)^{\frac{1}{2}k} \left(\frac{n-k}{n}\right)^{\frac{1}{2}(n-k)}.$$

Dieser Ausdruck ist symmetrisch bezüglich $k = \frac{n}{2}$, sodass für die weiteren Betrachtungen o.B.d.A. $k \leq \frac{n}{2}$ gilt. Mit

$$\frac{1}{\sqrt{\pi}} \sqrt{\frac{n}{k(n-k)}} \leq \frac{1}{\sqrt{\pi}} \sqrt{\frac{n}{n-1}} \leq \frac{1}{\sqrt{\pi}} \sqrt{2} \leq 1$$

sowie

$$e^{\left(\frac{n}{36k(n-k)} + \frac{5}{36n}\right)} \leq e^{\left(\frac{n}{36(n-1)} + \frac{5}{36n}\right)} \leq e^{\left(\frac{2}{36} + \frac{5}{36}\right)} \leq e^{\left(\frac{1}{5}\right)}$$

folgt

$$q \leq \sqrt[5]{e} \cdot \left(\frac{k}{n}\right)^{\frac{1}{2}k} \left(\frac{n-k}{n}\right)^{\frac{1}{2}(n-k)}. \quad (3.3)$$

Summiert man die Abschätzung 3.3 für alle Größen k der Menge S auf, so erhält man eine obere Schranke für die Wahrscheinlichkeit, dass der Graph nicht zusammenhängend ist, d.h. dass es eine Zerlegung der Knotenmenge in die Mengen S und T gibt, ohne dass eine Kante zwischen den Mengen verläuft. O.B.d.A. gilt weiterhin $k \leq \frac{n}{2}$, da für $k > \frac{n}{2}$ die Bezeichnungen „ S “ und „ T “ vertauscht werden können.

$$\begin{aligned} \Pr[\text{Graph nicht zshg.}] &\leq \sum_{k=1}^{\frac{n}{2}} \sqrt[5]{e} \cdot \left(\frac{k}{n}\right)^{\frac{1}{2}k} \left(\frac{n-k}{n}\right)^{\frac{1}{2}(n-k)} \\ &= \sqrt[5]{e} \cdot \left(\left(\frac{1}{n}\right)^{\frac{1}{2}} \left(\frac{n-1}{n}\right)^{\frac{1}{2}(n-1)} + \left(\frac{2}{n}\right) \left(\frac{n-2}{n}\right)^{\frac{1}{2}(n-2)} \right) \\ &\quad + \sqrt[5]{e} \cdot \sum_{k=3}^{\frac{n}{2}} \left(\frac{k}{n}\right)^{\frac{1}{2}k} \left(\frac{n-k}{n}\right)^{\frac{1}{2}(n-k)} \\ &\leq O\left(\frac{1}{\sqrt{n}}\right) + O\left(\frac{1}{n}\right) + \sqrt[5]{e} \cdot \sum_{k=3}^{\frac{n}{2}} \left(\frac{k}{n}\right)^{\frac{1}{2}k} \left(\frac{n-k}{n}\right)^{\frac{1}{2}(n-k)} \\ &\leq O\left(\frac{1}{\sqrt{n}}\right) + \sqrt[5]{e} \cdot \frac{n}{2} \left(\frac{3}{n}\right)^{\frac{3}{2}} \left(\frac{n-3}{n}\right)^{\frac{1}{2}(n-3)} \\ &= O\left(\frac{1}{\sqrt{n}}\right) \end{aligned}$$

Daraus ergibt sich die Behauptung

$$\Pr[\text{Graph zshg.}] \geq 1 - O\left(\frac{1}{\sqrt{n}}\right) \xrightarrow{n \rightarrow \infty} 1. \quad \square$$

Lemma 3.4

Der Durchschnittsgrad δ der Knoten in einem Graph mit angemessener Gradverteilung ist endlich.

Beweis Sei $A = (a_0, a_1, \dots)$ die angemessene Gradverteilung des Graphen und $\alpha > 2$ und $C > 0$ Konstanten mit $a_k < C \cdot k^{-\alpha}$ für $k \geq 3$. Für den Durchschnittsgrad δ gilt

$$\begin{aligned} \delta &= \sum_k k \cdot a_k \\ &< \sum_{k=3}^{\infty} k \cdot C \cdot k^{-\alpha} \\ &= C \cdot \sum_{k=3}^{\infty} k^{(1-\alpha)}. \end{aligned}$$

Nach dem Integralkriterium konvergiert diese unendliche Reihe genau dann, wenn das uneigentliche Integral $\int_1^{\infty} x^{(1-\alpha)} dx$ existiert. Das Integral ist dabei zugleich auch obere Abschätzung der Summe.

$$\begin{aligned} \int_1^{\infty} x^{(1-\alpha)} dx &= \lim_{x_0 \rightarrow \infty} \int_1^{x_0} x^{(1-\alpha)} dx \\ &= \lim_{x_0 \rightarrow \infty} \left(\frac{x_0^{(2-\alpha)}}{2-\alpha} - \frac{1^{(2-\alpha)}}{2-\alpha} \right) \\ &= 0 - \frac{1}{2-\alpha} \quad (\text{da } \alpha > 2) \\ &= \frac{1}{\alpha-2} \end{aligned}$$

Hiermit erhält man für den Durchschnittsgrad

$$\delta < \frac{C}{\alpha-2}. \quad \square$$

Wie man schnell einsieht, zählen auch reguläre Graphen mit einem konstanten, von der Knotenanzahl unabhängigen Grad zur Definition 3.2. Sei $g \geq 3$ dieser konstante Grad.

Somit gilt für die Verteilung:

$$a_k = \begin{cases} 0 & k \neq g \\ 1 & k = g \end{cases}.$$

Daraus folgt für die Wahl der Konstanten α und C :

$$\begin{aligned} \alpha &> 2 \\ C &> g^\alpha. \end{aligned}$$

3.2 Einführung eines Zeitfaktors

Ein Vorteil der späten Festlegung der Kanten ist es, die Wahl des Endpunktes einer Kante vom aktuellen Zustand der Breitensuche abhängig zu machen. Angenommen, man verzichtet darauf und möchte die Breitensuche im Hinblick auf die zu erwartende Verteilung der Knotengrade analysieren. Dann bleibt einem kaum eine andere Wahl, als alle Änderungen an der Schlange Q und den Zusammenhang zwischen dem Grad eines Knotens und seiner Position in dieser Schlange zu untersuchen. Dies ist alles andere als trivial und lässt sich vermeiden, wenn man die Breitensuche als einen zeitlich fortschreitenden Algorithmus ansieht, dessen Entscheidungen unter anderem auch vom aktuellen Zeitpunkt abhängen.

Jede Kopie der Knoten bekommt deshalb einen zufälligen, gleichverteilten Index $x \in [0, 1] \subseteq \mathbb{R}$ zugewiesen und wenn für eine Kante ein Endpunkt gesucht ist, wird von allen noch nicht abgearbeiteten Kopien diejenige mit dem größten Index ausgewählt. Dies ist möglich, da sowohl das Matching als auch die Indices gleichverteilt sein sollen und sich somit keine Änderungen im Resultat ergeben. Man kann sich nun vorstellen, dass der Algorithmus in dem Zeitraum $t = 1$ bis $t = 0$ abläuft. Zu einem beliebigen Zeitpunkt t_z gilt, dass alle unentdeckten Knoten einen maximalen Index $x \leq t_z$ und alle Knoten, die entweder entdeckt oder sogar abgearbeitet sind, einen maximalen Index $x > t_z$ besitzen. Unter dem maximalen Index eines Knotens versteht man hier das Maximum der Indices aller zum Knoten zugehörigen Kopien. Die Kopie, deren Index exakt t_z beträgt, wird vom Algorithmus gerade als adjazente für die aktuell in Bearbeitung befindliche Kopie ausgewählt. Die Bezeichnungen der Knoten bzw. Kopien sind in Tabelle 3.1 aufgeführt.

	noch nicht in Q	in Q	nicht mehr in Q
Knoten (vertex)	unentdeckt (untouched)	entdeckt (pending)	abgearbeitet (explored)
Kopien (copy)	unentdeckt (untouched)	entdeckt (enqueued)	abgearbeitet (exposed)

Tabelle 3.1: Knotenbezeichnungen

Es ist nun möglich, für jeden beliebigen Zeitpunkt genau anzugeben, wieviele Knoten bereits abgearbeitet oder entdeckt wurden und wieviele der Knoten bisher unentdeckt geblieben sind, was u.a. in [1] für Beweise genutzt wird. Da diese Arbeit in Bezug darauf entstanden ist, wird auch hier der gleiche Algorithmus verwendet. Als Nachteil muss man dafür in Kauf nehmen, dass die späte Festlegung der Kanten es nicht erlaubt, ohne viel Aufwand Schleifen und Mehrfachkanten zu vermeiden.

3.3 Erläuterungen zum Algorithmus

Der Algorithmus für die abgeänderte Breitensuche ähnelt der bekannten Form der Breitensuche sehr, die sich wie folgt beschreiben lässt. Ausgehend von einem Startknoten werden alle benachbarten Knoten als entdeckt markiert und an eine Schlange angefügt, es sei denn, sie sind schon als entdeckt bzw. abgearbeitet gekennzeichnet. Alle Kanten, die zu bisher unbekanntem Knoten führen, werden in den Suchbaum übernommen. Der somit abgearbeitete

Startknoten wird entsprechend markiert. Solange noch Knoten in der Schlange enthalten sind, entnimmt man den Knoten am Kopf der Schlange und verfährt mit ihm wie mit dem Startknoten. Ergebnis ist ein Baum, der den Graphen aufspannt, wenn dieser zusammenhängend ist, und den Startknoten als Wurzel besitzt.

In der traceroute-simulierenden Breitensuche arbeitet man jedoch mit den Kopien anstelle der Knoten und berücksichtigt den gedachten Zeitfaktor t mittels zufälliger Indices x für die Kopien:

Zeile 2 - 5

Entsprechend der übergebenen Gradsequenz werden für jeden Knoten die zugehörige Anzahl von Kopien erzeugt, diese mit einem Index versehen und als unentdeckt markiert. Einzige Ausnahme bilden die Kopien des Startknotens, da jene – wie der Startknoten in der unveränderten Breitensuche auch – als entdeckt markiert werden müssen, um einen Startpunkt für die Breitensuche festzulegen.

Zeile 6 - 7

Hier findet die Initialisierung der eigentlichen Breitensuche statt: Der Suchbaum enthält nur den Startknoten und die Schlange Q alle Kopien davon.

Zeile 8

Nun erfolgt der Eintritt in die while-Schleife der Breitensuche. Diese wird solange durchlaufen, wie es Knoten in der Schlange gibt, d.h. Knoten, die zwar entdeckt, aber noch nicht abgearbeitet sind.

Zeile 9 - 11

Nächster Schritt ist, für die Kopie am Kopf der Schlange entsprechend dem gedachten Matching eine zufällige Kante zu ermitteln. Wie schon näher dargelegt ist es an diesem Punkt des Algorithmus nun möglich, als Endpunkt für diese gesuchte Kante einfach von allen nicht abgearbeiteten Kopien jene mit dem größten Index zu bestimmen und diese dafür zu verwenden.

Zeile 12 - 14

Falls diese Kopie bisher unentdeckt geblieben ist, so wird der dazugehörige Knoten und die verbindende Kante in den Suchbaum eingefügt und der Knoten als entdeckt markiert. Das wird erreicht, indem alle Kopien des Knotens als entdeckt markiert und an die Schlange angefügt werden. Die mittels Index ausgewählte Kopie bildet eine Ausnahme, denn da sie schon eine inzidente Kante besitzt, kommt sie für weitere nicht mehr in Frage.

Zeile 15 - 16

Aus dem gleichen Grund wird eine Kopie, die bereits entdeckt und somit in Q enthalten ist, daraus entfernt.

Zeile 17

Abschliessend werden beide an der aktuellen Kante beteiligten Kopien als abgearbeitet markiert, denn um die Anforderung der Gradsequenz zu erfüllen, darf jede Kopie nur genau eine benachbarte Kante besitzen.

Algorithmus 1 Fortlaufende (zeitbasierte) Breitensuche auf randomisierten Graphen

Eingabe: randomisierter Multigraph (Knotenmenge V mit Graden d_v); Startknoten $u_s \in V$

Ausgabe: Breitensuchbaum $T = (V_T, E_T)$

```

1: function BFS
2:   for all  $u \in V$  do
3:     Erzeuge  $d_u$  viele Kopien  $u^{(1)}, u^{(2)}, \dots, u^{(d_u)}$ 
4:     Bestimme für jede Kopie  $u^{(i)}$  einen Index (zufällig, uniform)  $x_{u^{(i)}} \in [0, 1] \subseteq \mathbb{R}$ 
5:     Markiere alle Kopien als unentdeckt (ausgenommen Kopien des Startknotens  $u_s$ )
6:    $T = (\{u_s\}, \emptyset)$  // Breitensuchbaum anfangs nur der Startknoten
7:    $Q = (u_s^{(i)} \mid 1 \leq i \leq d_{u_s})$  // Markiere alle Kopien des Startknotens als entdeckt
8:   while  $Q \neq \emptyset$  do
9:     Entnimm die Kopie  $v^{(i)}$  vom Kopf der Schlange  $Q$ 
10:    Bestimme die Kopie mit dem größten Index unter allen nicht abgearbeiteten Kopien
    ohne  $v^{(i)}$ 
11:    Sei diese Kopie  $w^{(j)}$ . Diese wird adjazente Kopie von  $v^{(i)}$ 
12:    if  $w^{(j)}$  ist unentdeckt then //  $w^{(j)} \notin Q$ , d.h. Knoten  $w$  bisher nicht betreten
13:       $T = (V_T \cup \{w\}, E_T \cup \{(v, w)\})$  // Neuer Knoten und Kante zu  $T$  hinzufügen
14:      Füge alle Kopien von  $w$  außer  $w^{(j)}$  an das Ende von  $Q$  an
15:    else //  $w^{(j)} \in Q$ , d.h. Knoten  $w$  schon erreicht
16:      Entferne  $w^{(j)}$  aus  $Q$ 
17:    Markiere  $v^{(i)}$  und  $w^{(j)}$  als abgearbeitet
18:  return  $T$ 

```

3.4 Beispiel

Abbildung 3.1 soll als Grundlage für ein Beispiel der modifizierten Breitensuche dienen. Die Zahlen schräg unter den Knoten geben den Grad des Knotens an. Die Eingabemenge für den Algorithmus ist also

$$V = \{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \eta, \vartheta, \iota, \kappa\}$$

und

$$a_k = \begin{cases} 0 & k \leq 2 \text{ oder } k \geq 6 \\ 0.3 & k = 3 \\ 0.2 & k = 4 \\ 0.5 & k = 5 \end{cases}$$

bzw.

$$d_v = \begin{cases} 3 & v \in \{\zeta, \vartheta, \kappa\} \\ 4 & v \in \{\eta, \iota\} \\ 5 & v \in \{\alpha, \beta, \gamma, \delta, \varepsilon\} \end{cases} .$$

Die Wahl für den Startknoten u_s fällt auf ζ .

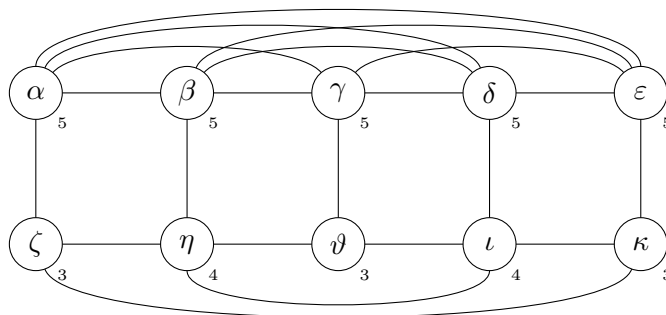


Abbildung 3.1: Beispielgraph

Erster Schritt des Algorithmus ist die Erstellung der Kopien und Ermittlung der Indices für diese. Das Ergebnis ist in Abbildung 3.2 dargestellt. Dann wird der Breitensuchbaum T und die Schlange Q initialisiert:

$$\begin{aligned} T &= (V_T, E_T) = (\{\zeta\}, \emptyset) \\ Q &= (\zeta^{(1)}, \zeta^{(2)}, \zeta^{(3)}) \end{aligned}$$

3. Die zeitbasierte Breitensuche

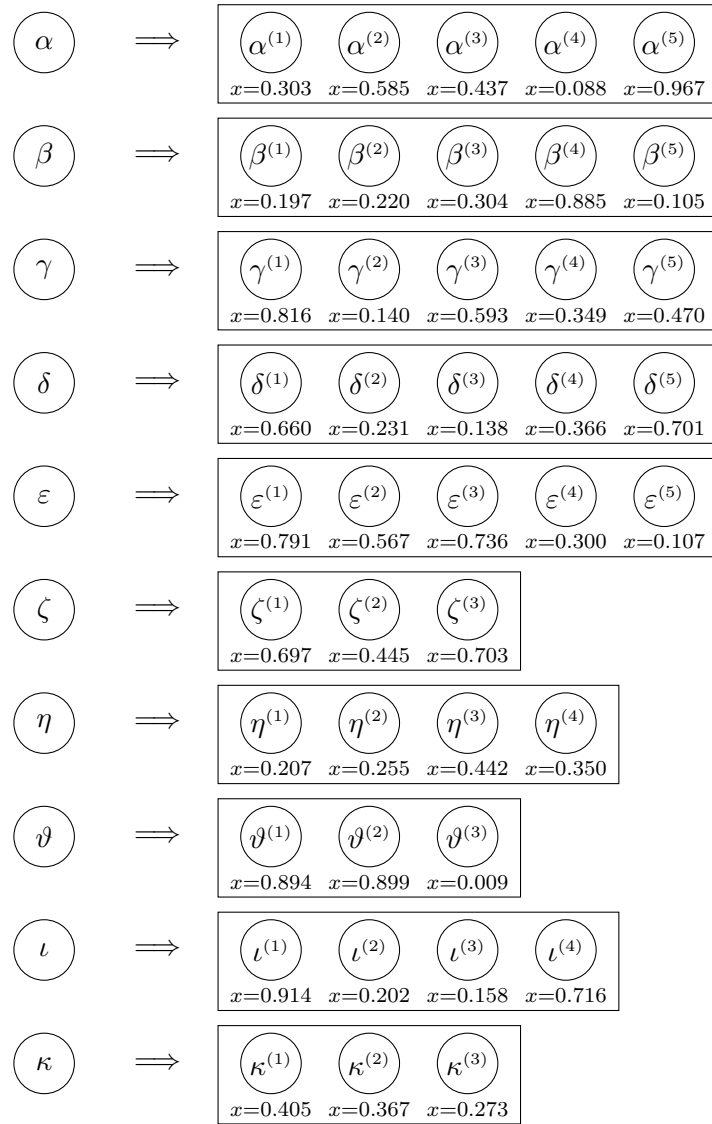


Abbildung 3.2: Kopien der Knoten mit Indices

3. Die zeitbasierte Breitensuche

Die Arbeitsweise der while-Schleife sowie die Änderungen an der Schlange Q und dem Breitensuchbaum T sind (in gekürzter Form) in Tabelle 3.2 aufgelistet. Dabei ist der Zusammenhang zwischen Zeitpunkt t und dem Index der Kopie $w^{(j)}$, die als Endpunkt der gerade betrachteten Kante ausgewählt wird, erkennbar.

	t	$v^{(i)}$	$w^{(j)}$	Q	V_T	E_T
0	1.000			$(\zeta^{(1)}, \zeta^{(2)}, \zeta^{(3)})$	$\{\zeta\}$	\emptyset
1	0.967	$\zeta^{(1)}$	$\alpha^{(5)}$	$(\zeta^{(2)}, \zeta^{(3)}, \alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)}, \alpha^{(4)})$	$\{\zeta, \alpha\}$	$\{\{\zeta, \alpha\}\}$
2	0.914	$\zeta^{(2)}$	$\iota^{(1)}$	$(\zeta^{(3)}, \alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)}, \alpha^{(4)}, \iota^{(2)}, \iota^{(3)}, \iota^{(4)})$	$\{\zeta, \alpha, \iota\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}\}$
3	0.899	$\zeta^{(3)}$	$\vartheta^{(2)}$	$(\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)}, \alpha^{(4)}, \iota^{(2)}, \iota^{(3)}, \iota^{(4)}, \vartheta^{(1)}, \vartheta^{(3)})$	$\{\zeta, \alpha, \iota, \vartheta\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}, \{\zeta, \vartheta\}\}$
4	0.894	$\alpha^{(1)}$	$\vartheta^{(1)}$	$(\alpha^{(2)}, \alpha^{(3)}, \alpha^{(4)}, \iota^{(2)}, \iota^{(3)}, \iota^{(4)}, \vartheta^{(3)})$	$\{\zeta, \alpha, \iota, \vartheta\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}, \{\zeta, \vartheta\}\}$
...
18	0.366	$\varepsilon^{(5)}$	$\delta^{(4)}$	$(\delta^{(2)}, \delta^{(3)}, \eta^{(1)}, \eta^{(2)}, \eta^{(4)}, \kappa^{(3)})$	$\{\zeta, \alpha, \iota, \vartheta, \beta, \gamma, \varepsilon, \delta, \eta, \kappa\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}, \{\zeta, \vartheta\}, \{\alpha, \beta\}, \{\alpha, \gamma\}, \{\alpha, \varepsilon\}, \{\vartheta, \delta\}, \{\gamma, \eta\}, \{\gamma, \kappa\}\}$
19	0.350	$\delta^{(2)}$	$\eta^{(4)}$	$(\delta^{(3)}, \eta^{(1)}, \eta^{(2)}, \kappa^{(3)})$	$\{\zeta, \alpha, \iota, \vartheta, \beta, \gamma, \varepsilon, \delta, \eta, \kappa\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}, \{\zeta, \vartheta\}, \{\alpha, \beta\}, \{\alpha, \gamma\}, \{\alpha, \varepsilon\}, \{\vartheta, \delta\}, \{\gamma, \eta\}, \{\gamma, \kappa\}\}$
20	0.273	$\delta^{(3)}$	$\kappa^{(3)}$	$(\eta^{(1)}, \eta^{(2)})$	$\{\zeta, \alpha, \iota, \vartheta, \beta, \gamma, \varepsilon, \delta, \eta, \kappa\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}, \{\zeta, \vartheta\}, \{\alpha, \beta\}, \{\alpha, \gamma\}, \{\alpha, \varepsilon\}, \{\vartheta, \delta\}, \{\gamma, \eta\}, \{\gamma, \kappa\}\}$
21	0.255	$\eta^{(1)}$	$\eta^{(2)}$	$()$	$\{\zeta, \alpha, \iota, \vartheta, \beta, \gamma, \varepsilon, \delta, \eta, \kappa\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}, \{\zeta, \vartheta\}, \{\alpha, \beta\}, \{\alpha, \gamma\}, \{\alpha, \varepsilon\}, \{\vartheta, \delta\}, \{\gamma, \eta\}, \{\gamma, \kappa\}\}$
22	0.000			$()$	$\{\zeta, \alpha, \iota, \vartheta, \beta, \gamma, \varepsilon, \delta, \eta, \kappa\}$	$\{\{\zeta, \alpha\}, \{\zeta, \iota\}, \{\zeta, \vartheta\}, \{\alpha, \beta\}, \{\alpha, \gamma\}, \{\alpha, \varepsilon\}, \{\vartheta, \delta\}, \{\gamma, \eta\}, \{\gamma, \kappa\}\}$

Tabelle 3.2: Ablauf der Breitensuche

3. Die zeitbasierte Breitensuche

Am resultierenden Suchbaum (Abbildung 3.3) ist deutlich zu erkennen, dass das Ergebnis der Breitensuche nicht zwangsläufig die dem Graph zugrundeliegende Verteilung der Knotengrade widerspiegelt. Von der im Ursprungsgraph vorhandenen Anzahl von Knoten hohen Grades (≥ 4) bleibt im Breitensuchbaum nur ein einziger, während die restlichen Knoten recht niedrige Grade besitzen.

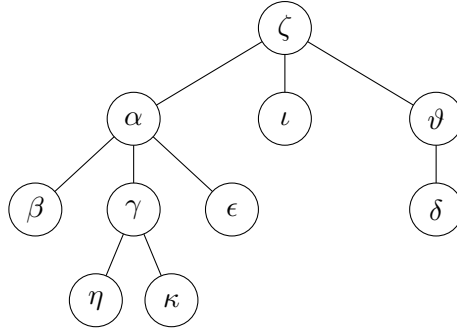


Abbildung 3.3: Breitensuchbaum

Dies lässt Rückschlüsse darauf zu, dass auch die Ergebnisse des traceroute-Verfahrens zur Ermittlung des Internetgraphen bzw. dessen Gradverteilung nicht zwingend den realen Verhältnissen entsprechen müssen.

4 Implementierung

4.1 Konstruktion von Zufallsgraphen

Erster Schritt für eine Implementierung ist die Erstellung geeigneter Zufallsgraphen. Dazu muss das Programm eine Wahl zwischen verschiedenen Typen von Verteilungen der Knotengrade ermöglichen. Anschließend wird sichergestellt, dass die generierte Verteilung sowohl dem vom Benutzer gewählten Typ als auch den Anforderungen der Breitensuche (siehe Definition 3.1 und 3.2) entspricht.

Wichtig dafür wäre noch zu bemerken, dass die relative Anzahl von Knoten mit Grad k , a_k , gleich der Wahrscheinlichkeit ist, dass ein zufällig gemäß der Gleichverteilung gewählter Knoten den Grad k besitzt. Die zufällige Gradsequenz erhält man deshalb in zwei Schritten:

- (a) Ermittlung einer angemessenen Gradverteilung entsprechend den Benutzervorgaben
- (b) Bestimmung der zufälligen Knotengrade nach folgendem Ablauf:
 1. Wähle einen noch nicht betrachteten Knoten aus.
 2. Bestimme eine Zufallszahl r , uniform verteilt, $0 < r \leq 1$.
 3. Ermittle das kleinste k_r , für das gilt

$$r \leq \sum_{k=0}^{k_r} a_k . \quad (4.1)$$

Dadurch wird jeder Knotengrad k nach seiner Wahrscheinlichkeit a_k oft ausgewählt.

4. Lege den Knotengrad auf k_r fest.

Der zweite Schritt ist dabei vollkommen unabhängig vom Typ der gewählten Verteilung und lässt sich für alle Eingaben verwenden. Die Aufgabe des ersten Schrittes hingegen ist das Normieren der gewählten Verteilung. Aus Definition 3.1 folgt nämlich

$$\sum_k a_k = 1 . \quad (4.2)$$

Die folgenden Abschnitte zeigen, wie man diese Gleichheit durch geeignete Wahl der Konstanten in den Verteilungen sicherstellen kann. Dabei bezeichnet n weiterhin die Knotenanzahl des Graphen.

Konstante Verteilungen

In diese Kategorie fallen alle vom Benutzer fest vorgegebenen Verteilungen, d.h. für alle Knotengrade k ist die relative Anzahl der Knoten a_k vorgegeben. Theoretisch ist es möglich, auch für diese Verteilungen die Knotengrade zufällig zu bestimmen. Dies wird jedoch im Programm nicht realisiert, sondern die Knotengrade werden gemäß ihrer relativen Häufigkeit auf die vorhandenen Knoten verteilt. Das ermöglicht, die Breitensuche auf einer exakt definierten Verteilung laufen zu lassen. Bei einer zufälligen Bestimmung der Knotengrade nach Schritt (b) hingegen könnten unerwünschte Abweichungen von der vorgegebenen Verteilung auftreten.

Ein Sonderfall dieser Verteilungen sind reguläre Graphen, bei denen die a_k für alle k bis auf eine Ausnahme Null sind.

Binomial- & Poisson-Verteilungen

Die Graphen mit binomial verteilten Knotengraden zählen zu den einfachsten Typen randomisierter Graphen. Die Entscheidung, ob eine bestimmte Kante im Graphen enthalten ist oder nicht, ist dabei unabhängig von allen anderen Kanten. Die Wahrscheinlichkeit p für eine Kante ist bei allen möglichen Kanten des Graphen identisch. Der Durchschnittsgrad z des Graphen entspricht der zu erwartenden Anzahl ausgehender Kanten eines Knotens und ergibt sich aus dem Produkt der Wahrscheinlichkeit p und der Anzahl möglicher benachbarter Knoten, d.h. für einen festen Durchschnittsgrad erhalten wir als Kantenwahrscheinlichkeit

$$p = \frac{z}{n-1} \rightarrow \frac{z}{n} \quad (n \gg 1).$$

Die gesuchte Wahrscheinlichkeit a_k für einen Grad k berechnet sich nach Binomialverteilung

$$a_k = C \binom{n}{k} p^k (1-p)^{n-k} \quad C > 0,$$

wobei C in diesem Fall zur Normierung der Verteilung benötigt wird. Die Verwendung von n anstelle von $n-1$ für den maximalen Grad eines Knotens führt nur zu vernachlässigbar geringen Abweichungen der Wahrscheinlichkeiten, die ohnehin durch die Konstante C ausgeglichen werden. Mit Gleichung 4.2 und Definition 3.2 erhält man

$$\begin{aligned} 1 &= \sum_{k=3}^{n-1} C \binom{n}{k} p^k (1-p)^{n-k} \\ C &= \frac{1}{\sum_{k=3}^{n-1} \binom{n}{k} p^k (1-p)^{n-k}} \\ &= \frac{1}{\sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} - \sum_{k=0}^2 \binom{n}{k} p^k (1-p)^{n-k} - \binom{n}{n} p^n (1-p)^{n-n}} \\ &= \frac{1}{1 - \sum_{k=0}^2 \binom{n}{k} p^k (1-p)^{n-k} - p^n}. \end{aligned}$$

Es ist ersichtlich, dass Binomialverteilungen – und auch die folgende Poisson-Verteilung – in der unveränderten Form keine Normierung benötigen ($\sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} = (p + (1-p))^n = 1$), aber durch die Bedingung, dass alle Knotengrade größer als Zwei sein müssen, eine Normierung erforderlich wird.

Betrachtet man sehr große Graphen ($n \rightarrow \infty, p \rightarrow 0, np = z$), so geht die Binomialverteilung in die Poisson-Verteilung über

$$a_k = C \frac{z^k e^{-z}}{k!},$$

$$C = \frac{1}{1 - e^{-z} \left(1 + z + \frac{z^2}{2} + \frac{z^n}{n!}\right)}.$$

Geometrische Verteilungen

Geometrische Verteilungen besitzen einen Parameter p mit $0 < p < 1$ und sind ein Spezialfall der sogenannten negativen Binomialverteilung. Die zugehörigen Wahrscheinlichkeiten lassen sich wie folgt ermitteln

$$a_k = C(1-p)p^k \quad C > 0.$$

Die Berechnung der Summe (Gleichung 4.2) zur Bestimmung der Normierungskonstante C erfolgt mit Hilfe der geometrischen Folge

$$\begin{aligned} \sum_{k=0}^n a_k &= C(1-p) \sum_{k=0}^n p^k \\ &= C(1-p) \frac{p^{n+1} - 1}{p - 1} \\ &= C(1 - p^{n+1}) \\ \sum_{k=3}^{n-1} a_k &= \sum_{k=0}^n a_k - \sum_{k=0}^2 a_k - C(1-p)p^n \\ &= C(1 - p^{n+1}) - C(1 - p^3) - C(1-p)p^n \\ &= C(p^3 - p^{n+1} - (1-p)p^n) \\ &= C(p^3 - p^n) \end{aligned}$$

und somit

$$C = \frac{1}{p^3 - p^n}.$$

power-law Verteilungen

Eine power-law Verteilung liegt vor, wenn der Anteil a_k der Knoten mit dem Grad k proportional zu $k^{-\alpha}$ bzw. zu $k^{-\alpha} e^{-k/\beta}$ ist. Obwohl die zweite Form im Gegensatz zur ersten eine

Normalisierung der Verteilung auch für $\alpha \leq 1$ ermöglicht, wird im weiteren Verlauf nur die einfachere Form der power-law Verteilung betrachtet. Man erhält deshalb

$$a_k = Ck^{-\alpha} \quad C > 0, \alpha > 1 .$$

Setzt man dies in 4.2 ein, ergibt sich

$$C = \frac{1}{\sum_{k=3}^{n-1} k^{-\alpha}} .$$

Der Wert C konvergiert für ein festes α und n genügend groß, wobei einige Werte von C (für $n \rightarrow \infty$) in Tabelle 4.1 aufgelistet sind.

α	2.0	2.5	3.0	5.0	10.0
C	2.5321	6.0713	12.9774	176.1260	55512.6033

Tabelle 4.1: power-law Konstante C

4.2 Analyse des Suchbaums

Nachdem die Breitensuche wie in Kapitel 3 beschrieben auf einem solchen Zufallsgraphen durchgeführt wurde, muss im nächsten Schritt ermittelt werden, wie gut die Gradverteilung des Suchbaums durch einen der bekannten Typen – Poisson-Verteilung, geometrische Verteilung und power-law Verteilung – beschrieben werden kann. In Bezug auf eine noch festzulegende Fehlergröße gilt es also, zu jedem Verteilungstyp den optimalen Parameter zu bestimmen und diese Verteilungen dann miteinander zu vergleichen.

Sehr viele Knoten im Suchbaum sind Blätter, die per Definition den Grad Eins besitzen und deshalb kaum Informationen zur Gradverteilung des ursprünglichen Graphen beitragen. Es liegt nahe, diese bei der Analyse des Suchbaums zu ignorieren, um den Einfluss der relevanten inneren Knoten nicht unnötig abzuschwächen. Eine ähnliche Argumentation kann man ebenfalls für Knoten anbringen, die im Suchbaum sehr tief und damit in der Nähe der Blätter vorzufinden sind. Viele ihrer Nachbarknoten sind schon entdeckt und die Kanten werden von der Breitensuche deshalb ignoriert. Offensichtlich sollte die Tiefe, bis in welche der Baum betrachtet wird, beschränkt werden.

Diese Beschränkung lässt sich schon in der Breitensuche realisieren, indem man die bisherige Höhe des Suchbaums mitführt und bei Erreichen einer gewissen Grenze die Breitensuche vorzeitig abbricht. Angenommen, für die Analyse sollen nur Knoten betrachtet werden, die in einer maximalen Tiefe von t_{cut} im Suchbaum anzutreffen sind. Die Breitensuche darf dann erst verlassen werden, wenn alle Knoten der Ebene $t_{cut} + 1$ erreicht wurden. Somit erhält man für die Blätter des beschnittenen Baumes (Knoten der Ebene t_{cut} im vollständigen Suchbaum) korrekte Werte für die Knotengrade.

Als Vergleichswert der einzelnen Verteilungen kommt eine quadratische Fehlersumme zum Einsatz. Dazu wird für den (beschnittenen) Suchbaum die exakte Gradverteilung, d.h. die

relativen Häufigkeiten a_k der Knotengrade k , bestimmt. Die Differenz dieser Häufigkeit und der relativen Häufigkeit des Grades in der zu testenden Verteilung wird quadriert und über alle Grade aufsummiert. Sei T der Suchbaum mit der Gradverteilung $A^T = (a_0^T, a_1^T, \dots)$ und $A^P = (a_0^P, a_1^P, \dots)$ die Verteilung, die geprüft werden soll. Die quadratische Fehlersumme e beträgt somit

$$e_{AP} = \sum_k (a_k^T - a_k^P)^2 . \quad (4.3)$$

Je größer der Wert von e ist, desto weniger entspricht die zu testende Verteilung der Verteilung des Suchbaums. Für $e = 0$ wird die Verteilung des Suchbaums hingegen vollständig durch den Typ und den Parameter der Prüfverteilung beschrieben. Dabei ist zu beachten, dass die Fehlersummen verschiedener Suchbäume nur bedingt miteinander vergleichbar sind, da der Wert der Fehlersumme auch von der Gradverteilung des Suchbaums abhängt.

Algorithmus 2 beschreibt die durchzuführende Analyse im Einzelnen. Ausgehend von einem Intervall werden zwei Werte links und rechts vom Mittelwert berechnet (Zeile 4/5). Diese dienen als Parameter für Prüfverteilungen (Zeile 6), deren quadratische Fehlersummen bzgl. des Suchbaums nach Gleichung 4.3 bestimmt werden (Zeile 7). Erreicht eine dieser Fehlersummen ein neues Minimum, so wird sowohl die Fehlersumme als auch der zugehörige Parameter abgespeichert (Zeile 8-11). Das Intervall wird anschliessend auf 60% seiner Größe verkleinert, indem der schlechtere (im Sinne der Fehlersummen) Prüfparameter die ihm näher liegende Intervallgrenze ablöst (Zeile 12-15). Sollte der Fall auftreten, dass das bisherige Minimum ausserhalb des neuen Intervalls liegt, so ruft sich die Funktion rekursiv selbst auf. Dabei testet sie zusätzlich zu dem neuen Intervall eines, in dem das bisherige Minimum enthalten ist (Zeile 16-25). Liegt das bisherige Minimum hingegen im neuen Intervall, so werden alle Schritte wiederholt, diesmal jedoch auf dem verkleinerten Intervall. Unterschreitet die Intervallgröße einen vorgegebenen Wert (Zeile 26), so endet die Schleife und das gefundene Minimum und der zugehörige Parameter werden zurückgeliefert (Zeile 27).

Der beschriebene Algorithmus kann nicht garantieren, dass das gefundene Minimum auch globales Minimum ist. Im direkten Vergleich mit einem zweiten Verfahren, welches pro Intervall sehr viel mehr Prüfverteilungen bestimmt, versagte er jedoch kein einziges Mal. Grund dafür ist der rekursive Aufruf mit einem weiteren Intervall, welches das gespeicherte Minimum enthält. Dies gestaltet den Algorithmus sehr viel stabiler gegenüber lokalen Minima der Fehlerfunktion.

Um nun herauszufinden, welchem Verteilungstyp der Suchbaum entspricht, genügt es, den Algorithmus einmal pro Typ mit einem ausreichend großen Startintervall aufzurufen und die zurückgegebenen Fehlersummen miteinander zu vergleichen. Für Verteilungen wie z.B. die geometrische, deren Parameter ein Wahrscheinlichkeitswert ist, bietet sich offensichtlich das Intervall $[0, 1]$ an.

Algorithmus 2 Test eines Verteilungstyps bzgl. des Suchbaums

Eingabe: Intervallgrenzen l und r , Genauigkeit $prec$ und Typ der Prüfverteilung dt

Ausgabe: Parameter p^* aus dem Intervall mit minimaler quadratischer Fehlersumme e^*

```

1: function TESTDEGREEDISTRIBUTION( $l, r, prec, dt$ )
2:    $e^* = +\infty$  // Minimum anfangs unbekannt
3:   repeat
4:      $lp = 0.6 \cdot l + 0.4 \cdot r$  // linken (Prüf-)Parameter berechnen
5:      $rp = 0.4 \cdot l + 0.6 \cdot r$  // ebenso den rechten
6:     Bestimme die Gradverteilungen  $ld$  und  $rd$ ; jeweils vom Typ  $dt$  und mit den Para-
metern  $lp$  bzw.  $rp$ 
7:     Berechne zugehörige Fehlersummen  $e_{ld}$  und  $e_{rd}$  nach Gleichung 4.3
8:     if  $e_{ld} \leq e^*$  then // linker Parameter liefert neue minimale Fehlersumme
9:        $p^* = lp; e^* = e_{ld}$  // dann inkl. dieser merken
10:    if  $e_{rd} \leq e^*$  then // analog für rechten Parameter
11:       $p^* = rp; e^* = e_{rd}$ 
12:    if  $e_{ld} > e_{rd}$  then // rechter Parameter besser als linker
13:       $l = lp$  // dann Intervall links abschneiden
14:    else
15:       $r = rp$  // ansonsten Intervall rechts abschneiden
16:    if  $p^* < l$  then // Optimum fällt links aus dem Intervall
17:      Aufruf von TESTDEGREEDISTRIBUTION( $p^*, l, prec, dt$ ) liefert  $(p_1^*, e_1^*)$ 
18:      Aufruf von TESTDEGREEDISTRIBUTION( $l, r, prec, dt$ ) liefert  $(p_2^*, e_2^*)$ 
19:      Bestimme  $k \in \{1, 2\}$  mit  $e_k^* = \min(e_1^*, e_2^*)$  // dann beide Intervalle testen
20:      return  $(p_k^*, e_k^*)$  // und Optimum zurückgeben
21:    if  $r < p^*$  then // Optimum fällt rechts aus dem Intervall
22:      Aufruf von TESTDEGREEDISTRIBUTION( $l, r, prec, dt$ ) liefert  $(p_1^*, e_1^*)$ 
23:      Aufruf von TESTDEGREEDISTRIBUTION( $r, p^*, prec, dt$ ) liefert  $(p_2^*, e_2^*)$ 
24:      Bestimme  $k \in \{1, 2\}$  mit  $e_k^* = \min(e_1^*, e_2^*)$  // dann analog fortfahren
25:      return  $(p_k^*, e_k^*)$  // und Optimum zurückgeben
26:  until  $(r - l) \leq prec$  // wiederholen, bis Intervall gewünschte Größe besitzt
27:  return  $(p^*, e^*)$ 

```

5 Auswertung

Die in den vorangegangenen Kapiteln besprochenen Algorithmen wurden in der Sprache Java implementiert und dienen als Grundlage für das Programm „tracerouteBFS“, welches im Rahmen dieser Arbeit entstanden ist. Es bietet die Möglichkeit, die zeitbasierte Breitensuche auf beliebigen Zufallsgraphen zu testen und dafür auch selbst Zufallsgraphen nach Benutzerangaben zu erstellen.

Um zu überprüfen, inwieweit man mit Hilfe der zeitbasierten Breitensuche Aussagen über die Verteilung des zugrundeliegenden Graphen treffen kann, wurden umfangreiche Simulationen durchgeführt. Dabei wurden hauptsächlich Graphen, deren Knotengrade einer Poisson-Verteilung entsprechen, betrachtet. Die kleinsten Graphen bestanden aus 1.000 Knoten und ca. 2.500 Kanten, während die größten mit 400.000 Knoten und rund 3.000.000 Kanten sehr viel umfangreicher waren als die in [5, 2] untersuchten Ausschnitte des Internetgraphen bzw. die in [3] verwendeten Graphen. Als Erwartungswerte der Durchschnittsgrade wurden mit $z = 5$, $z = 10$ und $z = 15$ Größen verwendet, die in der Nähe des für den Internetgraphen vermuteten Durchschnittsgrads liegen. Die Ergebnisse sind auszugsweise als Tabellen im Anhang der Arbeit zu finden.

Bricht die Breitensuche nicht vorzeitig ab, sodass alle Knoten des Graphen das Resultat beeinflussen, dann bestätigt sich das Ergebnis von [1]. Bis auf einen Fall findet sich immer eine power-law Verteilung als günstigste Variante den Suchbaum zu beschreiben. Die Ausnahme tritt auf, wenn der Ursprungsgraph schon eine power-law Verteilung besitzt und zwar mit dem Exponent $\alpha = 3.0$. In diesen Fällen ist eine geometrische Verteilung günstiger, jedoch unterscheiden sich die Fehlersummen der beiden Verteilungen nur gering. Es ist anzunehmen, dass sich dieses Verhalten für noch größere Exponenten fortsetzt, da sich power-law und geometrische Verteilung zumindest im Bereich kleinerer Knotengrade ähnlich verhalten.

Ein vollkommen anderes Bild ergibt sich jedoch, wenn man ausschliesslich einen Teil des Breitensuchbaums betrachtet. Enthält dieser nur noch einen gewissen Anteil aller Knoten, dann stellt sich die geometrische Verteilung als bessere Annäherung der Gradverteilung heraus. Verringert man die Anzahl der Knoten im Suchbaum immer weiter, so erfolgt letztendlich ein weiterer Wechsel und man erhält die Poisson-Verteilung als Optimum. Deren Parameter entspricht bei einem bestimmten Verhältnis von Suchbaumknoten zu Graphknoten übrigens recht exakt dem Parameter der ursprünglichen Verteilung. Beschneidet man den Suchbaum immer weiter, so bleibt die Poisson-Verteilung als Optimum, nur wird deren Parameter überschätzt.

In den Simulationen stellte sich heraus, dass die optimale Tiefe, in der der Suchbaum abgeschnitten werden sollte, von der Knotenanzahl n und dem Durchschnittsgrad z abhängig ist. Je größer die Knotenanzahl oder der Durchschnittsgrad der Poisson-Verteilung ist, desto kleiner muss der betrachtete Ausschnitt des Suchbaums sein. Ansonsten kann nicht sichergestellt werden, dass der ermittelte Parameter oder auch nur die Verteilung mit der dem Graphen zugrundeliegenden übereinstimmt.

Ein weiteres Problem zeigt sich, wenn man anstelle von Graphen mit einer Poisson-Verteilung jene mit einer power-law Verteilung untersucht. Die Ergebnisse bei diesen Graphen sind sehr unterschiedlich und noch stärker vom Parameter der Verteilung, d.h. dem Exponenten α abhängig. Für den kleinsten untersuchten Exponenten $\alpha = 2.0$ erhält man anfangs zwar eine power-law Verteilung, deren Exponent wird jedoch sehr viel größer eingeschätzt als es wirklich der Fall ist. Verkleinert man nun den untersuchten Ausschnitt des Graphen, so erreicht man eine Sprungstelle (für 100.000 Knoten liegt diese bei $3.80 \leq df \leq 3.85$, zur Erläuterung s. Anhang) und der ermittelte Exponent schwankt dort um den realen Wert. Die Größe des betrachteten Suchbaums ist für diese spezielle Verteilung also sehr viel bedeutsamer als es bei den bisherigen Verteilungen zu beobachten war.

Erhöht sich der Exponent der power-law Verteilung des Graphen auf $\alpha = 2.5$ oder $\alpha = 3.0$, erhält man das von der Poisson-Verteilung bekannte Ergebnis. Der vollständige Suchbaum liefert power-law, durch Abschneiden wird daraus die geometrische Verteilung und schränkt man den betrachteten Ausschnitt immer weiter ein, so bekommt man eine Poisson-Verteilung als Resultat. Ausgehend von einem solchen Verlauf der Analyse ist es also nicht möglich zu sagen, welche Verteilung der ursprüngliche Graph besitzt.

Einzig der Fall einer power-law Verteilung mit $\alpha = 2.0$ konnte somit recht genau mit dem beschriebenen Verfahren der Analyse erkannt werden. Alle anderen getesteten Verteilungen überschneiden sich in den Resultaten und lassen sich deshalb nicht voneinander trennen oder über- bzw. unterschätzen den Parameter der gesuchten Verteilung zum Teil erheblich. Kennt man zumindest die Knotenanzahl des Graphen, so lässt sich in einzelnen Fällen durch Vergleiche mit Simulationen verschiedener Verteilungen und gleicher Knotenanzahl die gesuchte Verteilung finden.

Die Ergebnisse zeigen, dass die zeitbasierte Breitensuche und damit auch das traceroute-Verfahren es in der bisherigen Form nicht ermöglichen, zuverlässige Aussagen über die Gradverteilung eines Graphen zu treffen. Durch ein vorzeitiges Abbrechen der Breitensuche bzw. Ignorieren bestimmter Suchbaumknoten lassen sich zusätzliche Erkenntnisse sammeln. Ausgehend von einem vollkommen unbekanntem Graphen reichen jedoch auch diese nicht aus, die zugrundeliegende Gradverteilung hinreichend genau zu bestimmen.

Literaturverzeichnis

- [1] Dimitris Achlioptas, Aaron Clauset, David Kempe und Cristopher Moore: *On the Bias of Traceroute Sampling*, 2005.
- [2] S. N. Dorogovtsev und J. F. F. Mendes: *Evolution of Networks*, 2003.
- [3] Anukool Lakhina, John W. Byers, Mark Crovella und Peng Xie: *Sampling Biases in IP Topology Measurements*, 2003.
- [4] M. E. J. Newman, S. H. Strogatz und D. J. Watts: *Random graphs with arbitrary degree distributions and their applications*, 2001.
- [5] Jean-Jacques Pansiot und Dominique Grad: *On Routes and Multicast Trees in the Internet*, 1998.

Anhang

In den nachfolgenden Tabellen sind die Ergebnisse zahlreicher Simulationsläufe der zeitbasierten Breitensuche auf unterschiedlichen Zufallsgraphen dargestellt. Die an das Programm „tracerouteBFS“ übergebenen Parameter sind dabei jeweils in der linken, oberen Ecke eingetragen: der Verteilungstyp des Graphen, die Anzahl der Knoten n sowie ein Parameter, der die Gradverteilung näher spezifiziert. Bei den Poisson-Verteilungen entspricht dieser Parameter dem Erwartungswert des Durchschnittsgrads z , für power-law Verteilungen ist es der Exponent α .

In der ersten Spalte mit der Überschrift „BFS-Tiefe“ ist der sogenannte Beschneidungsfaktor df aufgelistet. Über diesen wird festgelegt, in welcher Tiefe die Breitensuche vorzeitig abgebrochen werden soll (vgl. Abschnitt 4.2). Dabei berechnet sich die Tiefe nach $t_{cut} = \lfloor \log n/df \rfloor$. Bei der anschließenden Analyse des Suchbaums werden nur Knoten mit einer Tiefe $t < t_{cut}$ beachtet, Knoten mit $t = t_{cut}$ sind zwar im Suchbaum enthalten, werden aber ignoriert und dienen nur zur Berechnung der Grade höher gelegener Knoten. Trifft die Breitensuche auf einen Knoten in der Tiefe $t = t_{cut} + 1$, so wird sie an dieser Stelle beendet.

Die Ergebnisse eines Simulationslaufs sind jeweils in einem Kasten mit sieben Werten zusammengefasst. Der linke Wert (Spaltenüberschrift „Suchbaum“) entspricht der Anzahl relevanter Knoten im Suchbaum, d.h. die Anzahl von Knoten, deren Grad in die Analyse eingeht. Die sechs Werte rechts daneben (Spaltenüberschrift „Verteilung“) listen der Reihe nach die gefundenen optimalen Parameter der Poisson-, geometrischen und power-law Verteilung mit den zugehörigen quadratischen Fehlersummen auf. Die fett markierten Werte gehören dabei zu der Verteilung mit der kleinsten Fehlersumme in diesem Durchlauf.

poisson,100.000,5			Durchlauf				
BFS-Tiefe			1. Suchbaum Verteilung	2. Suchbaum Verteilung	3. Suchbaum Verteilung	4. Suchbaum Verteilung	5. Suchbaum Verteilung
1,00	Poisson	Parameter	0,90324	0,90034	0,89953	0,89518	0,90034
		Fehler	0,01595	0,01632	0,01650	0,01638	0,01627
	Geometric	Parameter	100.000	100.000	100.000	100.000	100.000
		Fehler	0,41249	0,41130	0,41094	0,40910	0,41130
	Power-Law	Parameter	0,00612	0,00639	0,00652	0,00651	0,00636
		Fehler	1,91900	1,92348	1,92348	1,92348	1,92348
			0,00180	0,00178	0,00177	0,00170	0,00177
2,00	Poisson	Parameter	5,42495	5,72034	5,61489	5,87444	5,77460
		Fehler	0,00370	0,00335	0,00421	0,00361	0,00363
	Geometric	Parameter	5.996	2.976	3.677	2.425	2.400
		Fehler	0,88449	0,89206	0,89035	0,89475	0,89424
	Power-Law	Parameter	0,07091	0,06989	0,07211	0,06999	0,06835
		Fehler	0,98173	0,93694	0,94141	0,93022	0,93022
		0,12017	0,11584	0,11911	0,11480	0,11344	
2,50	Poisson	Parameter	5,92927	5,97719	5,88570	5,99179	5,95086
		Fehler	0,00502	0,00572	0,00635	0,00431	0,00465
	Geometric	Parameter	1.187	1.051	520	577	524
		Fehler	0,89682	0,89775	0,89638	0,89810	0,89638
	Power-Law	Parameter	0,07606	0,07666	0,07469	0,06650	0,07420
		Fehler	0,87746	0,87299	0,82944	0,83267	0,83267
		0,11930	0,11922	0,11585	0,10708	0,11547	
3,00	Poisson	Parameter	6,01452	5,90880	5,95819	6,20885	6,05333
		Fehler	0,00650	0,00775	0,00563	0,00803	0,00727
	Geometric	Parameter	115	144	150	261	217
		Fehler	0,89696	0,89717	0,89898	0,90150	0,89839
	Power-Law	Parameter	0,07765	0,07216	0,05797	0,08357	0,07343
		Fehler	0,68636	0,70502	0,70502	0,76549	0,75803
		0,11259	0,10800	0,09261	0,12100	0,11087	

poisson,100.000,5			Durchlauf				
BFS-Tiefe			6. Suchbaum Verteilung	7. Suchbaum Verteilung	8. Suchbaum Verteilung	9. Suchbaum Verteilung	10. Suchbaum Verteilung
1,00	Poisson	Parameter	0,89729	0,89953	0,90155	0,89300	0,90106
		Fehler	0,01624	0,01651	0,01609	0,01683	0,01608
	Geometric	Parameter	100.000	100.000	100.000	100.000	100.000
		Fehler	0,41050	0,41085	0,41173	0,40934	0,41139
	Power-Law	Parameter	0,00638	0,00652	0,00623	0,00678	0,00624
		Fehler	1,92348	1,92348	1,92348	1,92348	1,92348
		0,00173	0,00176	0,00177	0,00174	0,00177	
2,00	Poisson	Parameter	5,57825	5,47202	5,52244	5,81676	5,71743
		Fehler	0,00422	0,00349	0,00324	0,00400	0,00324
	Geometric	Parameter	4.215	5.232	4.784	2.702	2.981
		Fehler	0,88893	0,88656	0,88728	0,89380	0,89181
	Power-Law	Parameter	0,07263	0,06973	0,06875	0,07303	0,06926
		Fehler	0,95709	0,95709	0,95709	0,93022	0,93694
		0,12034	0,11831	0,11682	0,11852	0,11532	
2,50	Poisson	Parameter	5,84672	5,88498	5,99253	5,89410	5,82281
		Fehler	0,00491	0,00426	0,00394	0,00583	0,00496
	Geometric	Parameter	1.340	2.054	708	675	1.666
		Fehler	0,89542	0,89585	0,89775	0,89638	0,89440
	Power-Law	Parameter	0,07343	0,07294	0,06986	0,07784	0,07468
		Fehler	0,88418	0,91778	0,85731	0,85731	0,91778
		0,11716	0,11731	0,11120	0,11992	0,11931	
3,00	Poisson	Parameter	6,51346	5,94795	5,88622	6,22175	6,29906
		Fehler	0,00631	0,01155	0,01159	0,00788	0,00808
	Geometric	Parameter	114	158	176	210	197
		Fehler	0,90491	0,89717	0,89717	0,90017	0,90266
	Power-Law	Parameter	0,06227	0,08830	0,08979	0,07531	0,07568
		Fehler	0,64995	0,70502	0,73115	0,73787	0,73115
		0,09368	0,12526	0,12738	0,11193	0,11124	

poisson,100.000,10			Durchlauf						
			1.	2.	3.	4.	5.		
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung		
1,00	Poisson	Parameter	0,50296	0,49870	0,50430	0,50430	0,50296		
		Fehler	0,01494	0,01471	0,01518	0,01507	0,01484		
		Geometric	Parameter	100.000	100.000	100.000	100.000	100.000	
1,00	Geometric	Parameter	0,24849	0,24651	0,24893	0,24893	0,24858		
		Fehler	0,01117	0,01101	0,01133	0,01124	0,01106		
		Power-Law	Parameter	2,37694	2,38141	2,37694	2,37694	2,37694	
1,00	Power-Law	Fehler	0,00199	0,00198	0,00203	0,00200	0,00194		
		2,00	Poisson	Parameter	3,17422	1,23518	1,77051	0,68798	1,53970
				Fehler	0,01668	0,02577	0,02467	0,01994	0,02605
Geometric	Parameter			27.842	53.650	42.487	79.344	46.411	
2,00	Geometric	Parameter	0,80456	0,53043	0,65475	0,33198	0,60935		
		Fehler	0,00411	0,00823	0,00250	0,01266	0,00473		
		Power-Law	Parameter	1,22978	1,66980	1,47364	2,08605	1,55036	
2,00	Power-Law	Fehler	0,03815	0,00216	0,00986	0,00107	0,00585		
		2,50	Poisson	Parameter	6,50830	6,25099	7,77387	6,91743	7,41870
				Fehler	0,00310	0,00387	0,00063	0,00219	0,00126
Geometric	Parameter			11.835	12.645	7.603	10.483	8.970	
2,50	Geometric	Parameter	0,90465	0,90112	0,92160	0,91098	0,91657		
		Fehler	0,03308	0,03144	0,04433	0,03687	0,03982		
		Power-Law	Parameter	0,98173	0,98496	0,91778	0,95709	0,93022	
2,50	Power-Law	Fehler	0,07233	0,07155	0,07936	0,07493	0,07623		
		3,00	Poisson	Parameter	10,72034	10,42865	10,61524	10,62008	10,47071
				Fehler	0,00153	0,00084	0,00066	0,00111	0,00024
Geometric	Parameter			668	1.040	1.164	888	1.335	
3,00	Geometric	Parameter	0,94397	0,94233	0,94345	0,94310	0,94233		
		Fehler	0,05903	0,05324	0,05659	0,05296	0,05295		
		Power-Law	Parameter	0,70502	0,73787	0,75803	0,73115	0,76549	
3,00	Power-Law	Fehler	0,08266	0,07825	0,08144	0,07726	0,07834		
		poisson,100.000,10			Durchlauf				
		BFS-Tiefe			6.	7.	8.	9.	10.
1,00	Poisson	Parameter	0,50296	0,50465	0,50094	0,50296	0,50247		
		Fehler	0,01468	0,01482	0,01506	0,01492	0,01477		
		Geometric	Parameter	100.000	100.000	100.000	100.000	100.000	
1,00	Geometric	Parameter	0,24822	0,24960	0,24770	0,24849	0,24822		
		Fehler	0,01093	0,01102	0,01126	0,01113	0,01101		
		Power-Law	Parameter	2,38141	2,37694	2,38141	2,37694	2,38141	
1,00	Power-Law	Fehler	0,00190	0,00190	0,00203	0,00197	0,00193		
		2,00	Poisson	Parameter	1,19129	1,08523	1,23591	0,73256	2,76581
				Fehler	0,02544	0,02508	0,02590	0,02086	0,01883
Geometric	Parameter			54.554	58.278	53.435	75.853	31.039	
2,00	Geometric	Parameter	0,51779	0,48419	0,53067	0,35107	0,77617		
		Fehler	0,00872	0,01003	0,00828	0,01268	0,00178		
		Power-Law	Parameter	1,69344	1,75104	1,66980	2,03844	1,28254	
2,00	Power-Law	Fehler	0,00180	0,00110	0,00212	0,00088	0,03042		
		2,50	Poisson	Parameter	5,87928	8,38366	6,54318	6,44749	6,04897
				Fehler	0,00524	0,00018	0,00280	0,00325	0,00455
Geometric	Parameter			13.948	6.294	11.646	11.909	13.527	
2,50	Geometric	Parameter	0,89454	0,92688	0,90560	0,90429	0,89696		
		Fehler	0,02804	0,04716	0,03372	0,03308	0,02909		
		Power-Law	Parameter	1,00860	0,88418	0,98173	0,98173	0,99740	
2,50	Power-Law	Fehler	0,06907	0,08025	0,07289	0,07259	0,06973		
		3,00	Poisson	Parameter	10,25101	10,56979	10,52445	10,55119	10,35469
				Fehler	0,00056	0,00187	0,00094	0,00113	0,00084
Geometric	Parameter			1.624	1.018	1.294	644	832	
3,00	Geometric	Parameter	0,94119	0,94336	0,94233	0,94345	0,94233		
		Fehler	0,05501	0,06120	0,05373	0,05647	0,05465		
		Power-Law	Parameter	0,77760	0,73787	0,76549	0,70502	0,73115	
3,00	Power-Law	Fehler	0,08123	0,08598	0,07903	0,08023	0,07925		

poisson,100.000,15			Durchlauf				
			1.	2.	3.	4.	5.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,34318	0,34542	0,34623	0,34744	0,34542
		Fehler	0,01086	0,01086	0,01076	0,01073	0,01084
		Geometric	100.000	100.000	100.000	100.000	100.000
1,00	Geometric	Parameter	0,17217	0,17284	0,17300	0,17379	0,17284
		Fehler	0,00925	0,00924	0,00914	0,00910	0,00922
		Power-Law	2,77056	2,76733	2,76733	2,76733	2,76733
1,00	Power-Law	Parameter	0,00323	0,00320	0,00315	0,00310	0,00319
		Fehler					
2,00	Poisson	Parameter	0,34744	0,34542	0,34542	0,34623	0,34695
		Fehler	0,01087	0,01074	0,01084	0,01080	0,01092
		Geometric	99.714	99.952	99.654	99.810	99.603
2,00	Geometric	Parameter	0,17410	0,17284	0,17284	0,17300	0,17379
		Fehler	0,00924	0,00913	0,00922	0,00918	0,00928
		Power-Law	2,76733	2,76733	2,76733	2,76733	2,76733
2,00	Power-Law	Parameter	0,00318	0,00315	0,00319	0,00317	0,00320
		Fehler					
2,50	Poisson	Parameter	1,00108	0,94305	1,09329	0,96510	1,10880
		Fehler	0,03083	0,02909	0,03233	0,03032	0,03187
		Geometric	45.779	47.881	43.012	46.713	43.043
2,50	Geometric	Parameter	0,46071	0,43851	0,49322	0,44762	0,49744
		Fehler	0,01566	0,01552	0,01491	0,01596	0,01443
		Power-Law	1,77891	1,80578	1,72131	1,79906	1,72131
2,50	Power-Law	Parameter	0,00063	0,00068	0,00059	0,00069	0,00061
		Fehler					
3,00	Poisson	Parameter	12,38210	11,69248	12,46173	12,42296	12,50465
		Fehler	0,00055	0,00141	0,00073	0,00059	0,00048
		Geometric	3.719	4.615	3.592	3.594	3.436
3,00	Geometric	Parameter	0,95201	0,94901	0,95236	0,95236	0,95296
		Fehler	0,04442	0,03975	0,04205	0,04435	0,04605
		Power-Law	0,77760	0,81252	0,77760	0,77760	0,77760
3,00	Power-Law	Parameter	0,06693	0,06353	0,06423	0,06662	0,06806
		Fehler					

poisson,100.000,15			Durchlauf				
			6.	7.	8.	9.	10.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,34318	0,34695	0,34318	0,34542	0,34744
		Fehler	0,01084	0,01091	0,01080	0,01095	0,01091
		Geometric	100.000	100.000	100.000	100.000	100.000
1,00	Geometric	Parameter	0,17249	0,17370	0,17173	0,17249	0,17410
		Fehler	0,00923	0,00926	0,00920	0,00933	0,00926
		Power-Law	2,77056	2,76733	2,77056	2,76733	2,76733
1,00	Power-Law	Parameter	0,00323	0,00319	0,00321	0,00326	0,00317
		Fehler					
2,00	Poisson	Parameter	0,34187	0,35179	0,34187	0,34542	0,34744
		Fehler	0,01075	0,01096	0,01087	0,01080	0,01082
		Geometric	99.886	99.364	99.956	99.931	99.795
2,00	Geometric	Parameter	0,17130	0,17542	0,17121	0,17249	0,17379
		Fehler	0,00917	0,00928	0,00928	0,00919	0,00919
		Power-Law	2,77056	2,74269	2,77056	2,76733	2,76733
2,00	Power-Law	Parameter	0,00322	0,00314	0,00328	0,00318	0,00315
		Fehler					
2,50	Poisson	Parameter	1,42008	1,01621	1,30752	1,02056	1,40871
		Fehler	0,03271	0,03053	0,03395	0,02994	0,03437
		Geometric	36.997	45.335	38.503	45.549	36.932
2,50	Geometric	Parameter	0,58861	0,46561	0,55988	0,46612	0,58809
		Fehler	0,01002	0,01528	0,01231	0,01476	0,01097
		Power-Law	1,56902	1,77891	1,60187	1,77891	1,56902
2,50	Power-Law	Parameter	0,00202	0,00061	0,00122	0,00054	0,00185
		Fehler					
3,00	Poisson	Parameter	12,78113	12,37992	13,66645	12,20277	12,86849
		Fehler	0,00063	0,00028	0,00027	0,00076	0,00044
		Geometric	3.320	3.546	2.300	3.932	3.137
3,00	Geometric	Parameter	0,95331	0,95227	0,95684	0,95135	0,95407
		Fehler	0,04416	0,04590	0,04717	0,04340	0,04531
		Power-Law	0,77760	0,77760	0,73787	0,79834	0,77221
3,00	Power-Law	Parameter	0,06599	0,06825	0,06714	0,06617	0,06673
		Fehler					

poisson,400.000,5			Durchlauf				
BFS-Tiefe			1.	2.	3.	4.	5.
			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,90155	0,90034	0,90155	0,89729	0,90155
		Fehler	0,01616	0,01629	0,01610	0,01642	0,01633
	Geometric	Parameter	400.000	400.000	400.000	400.000	400.000
		Fehler	0,00628	0,00638	0,00624	0,00648	0,00638
	Power-Law	Parameter	1,92348	1,92348	1,92348	1,92348	1,92348
		Fehler	0,00177	0,00177	0,00178	0,00175	0,00179
2,00	Poisson	Parameter	5,25101	5,60500	4,78065	4,80622	5,02741
		Fehler	0,00330	0,00353	0,00260	0,00283	0,00318
	Geometric	Parameter	29.787	17.132	49.317	47.462	39.372
		Fehler	0,88080	0,88925	0,86694	0,86810	0,87460
	Power-Law	Parameter	0,06800	0,07045	0,06163	0,06298	0,06623
		Fehler	1,01606	0,99740	1,06907	1,06907	1,04219
2,50	Poisson	Parameter	5,76363	5,89711	5,94571	5,92846	5,88622
		Fehler	0,00360	0,00352	0,00366	0,00397	0,00389
	Geometric	Parameter	9.715	4.037	4.609	4.468	5.364
		Fehler	0,89306	0,89630	0,89717	0,89691	0,89594
	Power-Law	Parameter	0,07076	0,07020	0,07127	0,07220	0,07188
		Fehler	0,98173	0,93694	0,94141	0,93694	0,94141
3,00	Poisson	Parameter	5,93580	6,00761	6,13198	5,97464	6,00277
		Fehler	0,00410	0,00408	0,00605	0,00483	0,00407
	Geometric	Parameter	814	866	998	986	842
		Fehler	0,89717	0,89744	0,90096	0,89802	0,89820
	Power-Law	Parameter	0,07231	0,07100	0,07664	0,07317	0,07234
		Fehler	0,85731	0,86400	0,86400	0,86400	0,85731
			0,11436	0,11303	0,11805	0,11540	0,11414

poisson,400.000,5			Durchlauf				
BFS-Tiefe			6.	7.	8.	9.	10.
			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,90289	0,90034	0,89729	0,90324	0,90155
		Fehler	0,01626	0,01626	0,01651	0,01605	0,01611
	Geometric	Parameter	400.000	400.000	400.000	400.000	400.000
		Fehler	0,04182	0,04121	0,041059	0,041217	0,041139
	Power-Law	Parameter	0,00633	0,00635	0,00653	0,00619	0,00625
		Fehler	1,92348	1,92348	1,92348	1,91900	1,92348
			0,00179	0,00177	0,00176	0,00180	0,00177
2,00	Poisson	Parameter	5,51384	4,93023	5,54173	4,85665	5,07019
		Fehler	0,00362	0,00286	0,00341	0,00283	0,00279
	Geometric	Parameter	19.143	43.076	18.433	45.698	36.962
		Fehler	0,88762	0,87170	0,88806	0,86942	0,87564
	Power-Law	Parameter	0,07003	0,06398	0,06966	0,06364	0,06488
		Fehler	1,00188	1,04891	0,99740	1,04891	1,04219
			0,11958	0,11781	0,11904	0,11806	0,11776
2,50	Poisson	Parameter	5,91502	5,93282	5,90880	5,97598	5,93652
		Fehler	0,00386	0,00367	0,00371	0,00411	0,00347
	Geometric	Parameter	5.673	2.966	4.337	2.239	2.375
		Fehler	0,89620	0,89691	0,89672	0,89810	0,89691
	Power-Law	Parameter	0,07215	0,07054	0,06970	0,07203	0,06862
		Fehler	0,94141	0,93022	0,93694	0,91778	0,93022
			0,11794	0,11512	0,11488	0,11577	0,11286
3,00	Poisson	Parameter	6,04127	6,14229	6,00060	6,19202	5,97029
		Fehler	0,00381	0,00568	0,00416	0,00502	0,00434
	Geometric	Parameter	690	367	924	580	593
		Fehler	0,89863	0,90112	0,89802	0,90179	0,89672
	Power-Law	Parameter	0,07004	0,07099	0,07275	0,07607	0,07154
		Fehler	0,83267	0,79834	0,86400	0,82944	0,83267
			0,11112	0,10927	0,11485	0,11589	0,11304

poisson,400.000,10			Durchlauf				
			1.	2.	3.	4.	5.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,50175	0,49870	0,50094	0,50247	0,50296
		Fehler	0,01484	0,01494	0,01494	0,01492	0,01480
	Geometric	Parameter	400.000	400.000	400.000	400.000	400.000
		Fehler	0,24770	0,24683	0,24762	0,24815	0,24822
	Power-Law	Parameter	2,38141	2,38141	2,38141	2,38141	2,38141
		Fehler	0,00197	0,00203	0,00201	0,00198	0,00194
2,00	Poisson	Parameter	0,52728	0,70752	0,67455	0,66165	0,56482
		Fehler	0,01565	0,02039	0,01975	0,01932	0,01683
	Geometric	Parameter	385.289	311.211	321.120	326.072	365.797
		Fehler	0,25965	0,34035	0,32640	0,32058	0,27744
	Power-Law	Parameter	2,32418	2,06531	2,11069	2,12636	2,26944
		Fehler	0,00186	0,00099	0,00115	0,00119	0,00167
2,50	Poisson	Parameter	3,86528	5,19430	3,99723	3,56692	2,31989
		Fehler	0,01319	0,00719	0,01255	0,01463	0,02168
	Geometric	Parameter	93.274	67.327	90.271	100.739	141.265
		Fehler	0,83925	0,88002	0,84488	0,82569	0,73353
	Power-Law	Parameter	0,00975	0,02196	0,01127	0,00714	0,00060
		Fehler	1,17504	1,07653	1,15812	1,20291	1,36166
3,00	Poisson	Parameter	10,31467	8,67202	9,97984	10,15794	9,33279
		Fehler	0,00054	0,00012	0,00031	0,00026	0,00005
	Geometric	Parameter	5.306	21.271	8.410	7.211	14.824
		Fehler	0,94181	0,92973	0,93965	0,94076	0,93499
	Power-Law	Parameter	0,05711	0,04905	0,05491	0,05419	0,05214
		Fehler	0,83267	0,93022	0,85731	0,85731	0,88418
			0,08451	0,08221	0,08362	0,08230	0,08317

poisson,400.000,10			Durchlauf				
			6.	7.	8.	9.	10.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,50094	0,50296	0,50175	0,50175	0,50094
		Fehler	0,01491	0,01485	0,01496	0,01486	0,01490
	Geometric	Parameter	400.000	400.000	400.000	400.000	400.000
		Fehler	0,24762	0,24867	0,24770	0,24797	0,24728
	Power-Law	Parameter	2,38141	2,37694	2,38141	2,38141	2,38141
		Fehler	0,00199	0,00194	0,00200	0,00198	0,00200
2,00	Poisson	Parameter	0,59285	0,52800	0,52728	0,52800	0,81676
		Fehler	0,01769	0,01565	0,01567	0,01591	0,02226
	Geometric	Parameter	352.010	385.695	385.356	383.361	281.563
		Fehler	0,29051	0,26038	0,25981	0,26052	0,38599
	Power-Law	Parameter	2,21760	2,32418	2,32418	2,32418	1,95840
		Fehler	0,00155	0,00184	0,00186	0,00191	0,00063
2,50	Poisson	Parameter	4,17464	3,42379	3,69006	4,69823	4,92587
		Fehler	0,01164	0,01502	0,01388	0,00919	0,00841
	Geometric	Parameter	86.195	104.123	97.367	76.225	72.166
		Fehler	0,85143	0,81891	0,83199	0,86743	0,87336
	Power-Law	Parameter	0,01300	0,00633	0,00858	0,01758	0,01912
		Fehler	1,14394	1,20291	1,17827	1,10938	1,08325
3,00	Poisson	Parameter	9,37298	8,95873	9,73628	9,46020	9,82047
		Fehler	0,00011	0,00004	0,00009	0,00010	0,00010
	Geometric	Parameter	14.016	18.277	11.134	12.714	10.074
		Fehler	0,93575	0,93222	0,93783	0,93628	0,93877
	Power-Law	Parameter	0,05264	0,05078	0,05353	0,05297	0,05344
		Fehler	0,88418	0,91778	0,87746	0,88418	0,87299
			0,08336	0,08302	0,08318	0,08336	0,08267

poisson,400.000,15			Durchlauf				
			1.	2.	3.	4.	5.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,34542	0,34542	0,34623	0,34318	0,34695
		Fehler	0,01082	0,01084	0,01086	0,01095	0,01083
	Geometric	Parameter	400.000	400.000	400.000	400.000	400.000
		Fehler	0,17328	0,17249	0,17344	0,17249	0,17379
Power-Law	Parameter	2,76733	2,76733	2,76733	2,76733	2,76733	
	Fehler	0,00317	0,00321	0,00318	0,00327	0,00315	
2,00	Poisson	Parameter	0,34542	0,34542	0,34542	0,34542	0,34542
		Fehler	0,01090	0,01089	0,01082	0,01088	0,01091
	Geometric	Parameter	400.000	400.000	399.999	400.000	400.000
		Fehler	0,17270	0,17275	0,17275	0,17328	0,17270
Power-Law	Parameter	0,00927	0,00927	0,00920	0,00926	0,00929	
	Fehler	2,76733	2,76733	2,76733	2,76733	2,76733	
			0,00322	0,00322	0,00318	0,00320	0,00324
2,50	Poisson	Parameter	0,47500	0,46041	0,42271	0,41472	0,54901
		Fehler	0,01606	0,01577	0,01419	0,01388	0,01913
	Geometric	Parameter	310.966	317.536	339.641	345.094	277.523
		Fehler	0,23564	0,22899	0,21033	0,20675	0,27098
Power-Law	Parameter	0,01261	0,01250	0,01154	0,01134	0,01429	
	Fehler	2,42173	2,44860	2,55109	2,55781	2,27267	
			0,00302	0,00316	0,00326	0,00326	0,00276
3,00	Poisson	Parameter	5,65995	6,64800	6,97944	8,56087	8,85729
		Fehler	0,01820	0,01467	0,01352	0,00840	0,00763
	Geometric	Parameter	56.678	48.432	46.059	35.561	33.793
		Fehler	0,89889	0,91230	0,91633	0,93024	0,93237
Power-Law	Parameter	0,01252	0,01695	0,01868	0,02511	0,02637	
	Fehler	1,04219	1,00188	0,98496	0,94141	0,93694	
			0,04788	0,05147	0,05275	0,05632	0,05701

poisson,400.000,15			Durchlauf				
			6.	7.	8.	9.	10.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter	0,34542	0,34542	0,34542	0,34542	0,34318
		Fehler	0,01081	0,01080	0,01081	0,01089	0,01087
	Geometric	Parameter	400.000	400.000	400.000	400.000	400.000
		Fehler	0,17249	0,17284	0,17249	0,17270	0,17217
Power-Law	Parameter	0,00920	0,00918	0,00920	0,00927	0,00926	
	Fehler	2,77056	2,76733	2,76733	2,76733	2,77056	
			0,00320	0,00317	0,00319	0,00323	0,00323
2,00	Poisson	Parameter	0,34695	0,34542	0,34623	0,34542	0,34542
		Fehler	0,01081	0,01085	0,01084	0,01094	0,01089
	Geometric	Parameter	400.000	399.999	400.000	399.996	399.984
		Fehler	0,17352	0,17284	0,17344	0,17275	0,17284
Power-Law	Parameter	0,00918	0,00923	0,00921	0,00931	0,00926	
	Fehler	2,76733	2,76733	2,76733	2,76733	2,76733	
			0,00315	0,00319	0,00317	0,00324	0,00321
2,50	Poisson	Parameter	0,40255	0,42664	0,47572	0,37176	0,39336
		Fehler	0,01327	0,01432	0,01637	0,01191	0,01287
	Geometric	Parameter	353.308	337.100	309.602	377.001	359.126
		Fehler	0,20074	0,21264	0,23642	0,18569	0,19709
Power-Law	Parameter	0,01091	0,01161	0,01285	0,00998	0,01063	
	Fehler	2,59140	2,52347	2,42173	2,66978	2,61504	
			0,00323	0,00323	0,00310	0,00322	0,00321
3,00	Poisson	Parameter	6,01331	7,97101	6,56558	7,15448	10,69089
		Fehler	0,01735	0,01025	0,01504	0,01286	0,00256
	Geometric	Parameter	53.626	38.930	48.729	44.536	23.035
		Fehler	0,90389	0,92585	0,91185	0,91830	0,94419
Power-Law	Parameter	0,01380	0,02291	0,01676	0,01949	0,03587	
	Fehler	1,04219	0,95709	1,00188	0,98496	0,88418	
			0,04896	0,05521	0,05137	0,05325	0,06267

powerlaw,100.000,2,0			Durchlauf				
BFS-Tiefe			1. Suchbaum Verteilung	2. Suchbaum Verteilung	3. Suchbaum Verteilung	4. Suchbaum Verteilung	5. Suchbaum Verteilung
1,00	Poisson	Parameter	0,07243	0,07243	0,07413	0,07413	0,07243
		Fehler	0,00022	0,00023	0,00025	0,00028	0,00023
	Geometric	Parameter	100.000	100.000	100.000	100.000	100.000
		Fehler	0,03591	0,03631	0,03709	0,03709	0,03658
	Power-Law	Parameter		0,00021	0,00022	0,00023	0,00026
		Fehler		4,87299	4,85731	4,82944	4,82944
			0,00010	0,00011	0,00012	0,00013	0,00011
3,75	Poisson	Parameter	0,07534	0,07614	0,08703	0,08267	0,08187
		Fehler	0,00022	0,00025	0,00034	0,00029	0,00027
	Geometric	Parameter	98.904	98.671	99.760	99.437	99.984
		Fehler	0,03762	0,03796	0,04330	0,04102	0,04058
	Power-Law	Parameter		0,00021	0,00032	0,00027	0,00025
		Fehler		4,81252	4,79834	4,61221	4,69084
			0,00010	0,00011	0,00015	0,00012	0,00012
3,80	Poisson	Parameter	0,08775	0,10046	0,07614	0,08703	0,08267
		Fehler	0,00032	0,00042	0,00025	0,00037	0,00030
	Geometric	Parameter	80.129	79.699	99.134	92.054	93.395
		Fehler	0,04383	0,05020	0,03796	0,04351	0,04102
	Power-Law	Parameter		0,00030	0,00024	0,00035	0,00028
		Fehler		4,60549	4,40485	4,79834	4,60549
			0,00013	0,00016	0,00012	0,00013	
3,85	Poisson	Parameter	0,59551	0,37920	1,98451	0,34187	0,57906
		Fehler	0,00788	0,00414	0,05925	0,00337	0,00874
	Geometric	Parameter	12.262	18.389	1.400	22.595	11.358
		Fehler	0,28345	0,18569	0,74146	0,16769	0,27804
	Power-Law	Parameter		0,00462	0,00297	0,01661	0,00248
		Fehler		2,27267	2,74269	1,32733	2,86656
			0,00015	0,00021	0,00100	0,00020	0,00014

powerlaw,100.000,2,0			Durchlauf				
BFS-Tiefe			6. Suchbaum Verteilung	7. Suchbaum Verteilung	8. Suchbaum Verteilung	9. Suchbaum Verteilung	10. Suchbaum Verteilung
1,00	Poisson	Parameter	0,07534	0,07614	0,07461	0,07243	0,07243
		Fehler	0,00026	0,00025	0,00026	0,00024	0,00021
	Geometric	Parameter	100.000	100.000	100.000	100.000	100.000
		Fehler	0,03776	0,03855	0,03718	0,03631	0,03586
	Power-Law	Parameter		0,00025	0,00024	0,00025	0,00022
		Fehler		4,79834	4,77760	4,81700	4,85731
			0,00012	0,00011	0,00012	0,00011	0,00009
3,75	Poisson	Parameter	0,08703	0,08267	0,12435	0,07461	0,07614
		Fehler	0,00033	0,00029	0,00066	0,00023	0,00025
	Geometric	Parameter	96.961	98.036	63.508	99.617	95.875
		Fehler	0,04330	0,04093	0,06197	0,03718	0,03855
	Power-Law	Parameter		0,00031	0,00027	0,00059	0,00022
		Fehler		4,61221	4,69756	4,13699	4,82944
			0,00014	0,00013	0,00022	0,00010	0,00011
3,80	Poisson	Parameter	0,07413	0,11256	0,07969	0,07461	0,08703
		Fehler	0,00024	0,00051	0,00028	0,00025	0,00034
	Geometric	Parameter	99.981	76.155	98.779	99.817	99.666
		Fehler	0,03665	0,05597	0,03980	0,03752	0,04325
	Power-Law	Parameter		0,00022	0,00046	0,00026	0,00024
		Fehler		4,83267	4,26301	4,73115	4,81252
			0,00011	0,00018	0,00013	0,00012	0,00015
3,85	Poisson	Parameter	2,05205	0,72531	0,51021	1,37995	2,24673
		Fehler	0,05961	0,01189	0,00671	0,04009	0,06963
	Geometric	Parameter	1.261	8.969	13.645	3.270	826
		Fehler	0,74694	0,33983	0,24683	0,58630	0,78915
	Power-Law	Parameter		0,01713	0,00643	0,00428	0,01540
		Fehler		1,29600	2,08605	2,42496	1,55036
			0,00055	0,00013	0,00013	0,00032	0,00063

powerlaw,100.000,2,5			Durchlauf				
			1.	2.	3.	4.	5.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter Fehler	0,40850 0,00306	0,40290 0,00301	0,40981 0,00315	0,39771 0,00288	0,39081 0,00298
	Geometric	Parameter Fehler	100.000 0,19842	100.000 0,19586	100.000 0,19910	100.000 0,19353	100.000 0,19037
		Parameter Fehler	0,00190 2,68222 0,00003	0,00187 2,71582 0,00002	0,00197 2,70338 0,00003	0,00180 2,72254 0,00002	0,00189 2,73600 0,00001
2,50	Poisson	Parameter Fehler	1,02693 0,00623	0,86849 0,00519	0,51809 0,00363	1,30994 0,00743	0,56191 0,00407
	Geometric	Parameter Fehler	40.341 0,44266 0,00118	49.953 0,38696 0,00131	79.531 0,24797 0,00183	32.511 0,52716 0,00114	75.393 0,26631 0,00193
		Parameter Fehler	1,85182 0,00591	1,97051 0,00375	2,44860 0,00031	1,66980 0,01098	2,37022 0,00045
3,00	Poisson	Parameter Fehler	2,48225 0,01537	2,97101 0,02270	2,19179 0,01207	1,74901 0,01008	2,48225 0,01736
	Geometric	Parameter Fehler	8.156 0,75643 0,00667	3.926 0,81542 0,00873	14.143 0,71114 0,00435	16.907 0,63336 0,00246	7.643 0,75866 0,00602
		Parameter Fehler	1,28254 0,03003	1,16260 0,03045	1,36166 0,02670	1,48931 0,01904	1,27582 0,02721
3,50	Poisson	Parameter Fehler	2,86501 0,02064	2,58188 0,01656	1,97016 0,01095	0,95819 0,00538	1,65422 0,00915
	Geometric	Parameter Fehler	5.214 0,80164 0,00663	8.846 0,76882 0,00691	14.250 0,67555 0,00394	43.662 0,41804 0,00117	21.448 0,61163 0,00175
		Parameter Fehler	1,20291 0,02829	1,27582 0,03045	1,42213 0,02400	1,90333 0,00527	1,53469 0,01743

powerlaw,100.000,2,5			Durchlauf				
			6.	7.	8.	9.	10.
BFS-Tiefe			Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung	Suchbaum Verteilung
1,00	Poisson	Parameter Fehler	0,40908 0,00327	0,37176 0,00257	0,42005 0,00331	0,39081 0,00299	0,40850 0,00303
	Geometric	Parameter Fehler	100.000 0,19910	100.000 0,18109	100.000 0,20343	100.000 0,19011	100.000 0,19836
		Parameter Fehler	0,00206 2,66978 0,00003	0,00166 2,79420 0,00002	0,00205 2,65859 0,00002	0,00191 2,74269 0,00001	0,00187 2,68222 0,00003
2,50	Poisson	Parameter Fehler	0,39699 0,00274	0,55272 0,00370	0,44510 0,00333	0,41634 0,00299	1,08039 0,00630
	Geometric	Parameter Fehler	94.886 0,19231	75.561 0,26205	88.525 0,21514	93.536 0,20161	40.186 0,45968
		Parameter Fehler	0,00171 2,72701 0,00003	0,00176 2,38141 0,00052	0,00194 2,59812 0,00006	0,00181 2,66978 0,00004	1,80578 0,00700
3,00	Poisson	Parameter Fehler	0,65135 0,00403	0,86420 0,00489	2,84181 0,01801	2,08622 0,01290	2,24939 0,01326
	Geometric	Parameter Fehler	62.596 0,30284	49.259 0,38530 0,00124	6.734 0,79690 0,00727	13.203 0,69716 0,00370	10.938 0,72256 0,00526
		Parameter Fehler	0,00150 2,23834 0,00128	1,97797 0,00128 0,00393	1,20291 0,03090	1,38240 0,02321	1,34300 0,02737
3,50	Poisson	Parameter Fehler	2,69089 0,01816	1,52939 0,00788	3,85995 0,02778	2,44445 0,01572	1,80145 0,00954
	Geometric	Parameter Fehler	6.395 0,78349 0,00806	24.229 0,58276 0,00205	1.136 0,88134 0,01289	9.656 0,75107 0,00557	17.355 0,64217 0,00249
		Parameter Fehler	1,22978 0,03092	1,56902 0,01664	0,98173 0,03176	1,29600 0,02780	1,46916 0,02037

powerlaw,100.000,3.0			Durchlauf				
BFS-Tiefe			1. Suchbaum Verteilung	2. Suchbaum Verteilung	3. Suchbaum Verteilung	4. Suchbaum Verteilung	5. Suchbaum Verteilung
1,00	Poisson	Parameter	0,74546	0,76533	0,74667	0,75704	0,75496
		Fehler	0,00529	0,00537	0,00534	0,00507	0,00501
	Geometric	Parameter	100.000 0,34190	100.000 0,34922	100.000 0,34251	100.000 0,34620	100.000 0,34489
		Fehler	0,00153	0,00146	0,00154	0,00135	0,00136
	Power-Law	Parameter	2,11069	2,08605	2,11069	2,11069	2,11069
		Fehler	0,00195	0,00221	0,00195	0,00227	0,00220
1,50	Poisson	Parameter	0,75061	0,75656	0,75061	0,74619	0,75583
		Fehler	0,00539	0,00549	0,00547	0,00531	0,00524
	Geometric	Parameter	99.893 0,34415	99.999 0,34626	99.761 0,34422	99.999 0,34204	100.000 0,34560
		Fehler	0,00155	0,00157	0,00159	0,00155	0,00145
	Power-Law	Parameter	2,11069	2,11069	2,11069	2,11069	2,11069
		Fehler	0,00198	0,00201	0,00195	0,00193	0,00212
2,00	Poisson	Parameter	1,11202	2,52110	2,82329	1,73472	0,93580
		Fehler	0,00351	0,00510	0,00704	0,00253	0,00472
	Geometric	Parameter	73.481 0,46442	24.107 0,74059	17.180 0,77636	46.194 0,61541	86.338 0,40963
		Fehler	0,00038	0,02230	0,02617	0,00569	0,00062
	Power-Law	Parameter	1,80578	1,29600	1,22978	1,51005	1,93020
		Fehler	0,01124	0,06893	0,07164	0,03433	0,00553
2,50	Poisson	Parameter	2,98525	3,45458	4,18190	3,90453	3,04504
		Fehler	0,00812	0,01359	0,02892	0,02596	0,00885
	Geometric	Parameter	14.211 0,79273	6.321 0,83586	823 0,87993	3.022 0,86587	12.813 0,79926
		Fehler	0,02945	0,03868	0,05687	0,05478	0,03091
	Power-Law	Parameter	1,20291	1,10938	0,93022	1,00860	1,17827
		Fehler	0,07560	0,08218	0,09265	0,09537	0,07706

powerlaw,100.000,3.0			Durchlauf				
BFS-Tiefe			6. Suchbaum Verteilung	7. Suchbaum Verteilung	8. Suchbaum Verteilung	9. Suchbaum Verteilung	10. Suchbaum Verteilung
1,00	Poisson	Parameter	0,74837	0,74111	0,75496	0,77016	0,76139
		Fehler	0,00522	0,00529	0,00543	0,00540	0,00529
	Geometric	Parameter	100.000 0,34318	100.000 0,34015	100.000 0,34546	100.000 0,35111	100.000 0,34765
		Fehler	0,00147	0,00154	0,00156	0,00147	0,00146
	Power-Law	Parameter	2,11069	2,11069	2,11069	2,08605	2,08605
		Fehler	0,00203	0,00191	0,00199	0,00223	0,00215
1,50	Poisson	Parameter	0,75327	0,76097	0,75103	0,75656	0,75327
		Fehler	0,00542	0,00514	0,00508	0,00495	0,00508
	Geometric	Parameter	99.988 0,34493	99.986 0,34781	100.000 0,34328	99.850 0,34560	99.749 0,34458
		Fehler	0,00155	0,00137	0,00141	0,00131	0,00138
	Power-Law	Parameter	2,11069	2,08605	2,11069	2,11069	2,11069
		Fehler	0,00201	0,00226	0,00211	0,00228	0,00216
2,00	Poisson	Parameter	1,32929	2,68919	2,70752	1,35974	1,99457
		Fehler	0,00286	0,00602	0,00650	0,00283	0,00274
	Geometric	Parameter	60.575 0,52416	20.094 0,76100	19.751 0,76358	61.023 0,53155	36.191 0,66301
		Fehler	0,00146	0,02430	0,02546	0,00158	0,00997
	Power-Law	Parameter	1,68100	1,24222	1,24222	1,66980	1,42885
		Fehler	0,01893	0,06993	0,07097	0,01980	0,04499
2,50	Poisson	Parameter	3,57992	4,16883	3,37729	3,86366	4,20543
		Fehler	0,01466	0,03700	0,01316	0,01979	0,03772
	Geometric	Parameter	6.017 0,84430	397 0,88002	7.520 0,82861	2.438 0,86383	787 0,87971
		Fehler	0,03963	0,06968	0,03691	0,04664	0,07090
	Power-Law	Parameter	1,07653	0,86400	1,11781	1,00860	0,91778
		Fehler	0,08266	0,10298	0,08143	0,08582	0,10731