SEMINARARBEIT

Gleichverteilte Spannbäume

ausgeführt am

Institut für
Analysis und Scientific Computing
TU Wien

unter der Anleitung von

Dr. Benedikt Stufler

durch

Cederic Demoulin
Matrikelnummer: 11901874
Wiedner Hauptstraße 8-10
1040, Wien

Wien, Februar 2022

Abstract

Der Groundskeeper Algorithmus liefert eine Methode, um, ausgehend von einem zusammenhängenden, ungerichteten, endlichen Graphen, einen Spannbaum zu erzeugen. Ziel dieser Arbeit ist es, diesen Algorithmus zu beschreiben und anschließend zu beweisen, dass die dadurch konstruierten Spannbäume gleichverteilt auf der Menge aller Spannbäume des Graphen sind. Dabei werden grundlegende Erkenntnisse aus der Theorie der Markov-Ketten gezeigt und verwendet. Im zweiten Teil der Arbeit wird die Implementierung des Groundskeeper Algorithmus in Python präsentiert sowie Ergebnisse statistischer Auswertungen von dadurch generierten Bäumen.

Inhaltsverzeichnis

1 Einleitung

Ein Graph, also eine Menge von Knoten, welche mit Kanten verbunden sind, kann für viele Systeme mit denen wir täglich Kontakt haben, stehen. Die Menge der Knoten könnte beispielsweise Haushalte symbolisieren und die Kanten Leitungen, welche die Häuser mit Wasser oder Gas versorgen. Ein Graph könnte aber auch für Computersysteme stehen, die durch Kabel miteinander kommunizieren können. In den meisten Fällen haben diese beiden Szenarien gemeinsam, dass ein Anschluss oder eine Verbindung zum Netzwerk der Wasserleitungen bzw. des Internetanbieters ausreichend ist. Zwei Wasseranschlüsse sowie zwei Internetanschlüsse sind zum einen meistens nicht notwendig und zum anderen eine Frage der Kosten. Wie verbindet man die Haushalte einer Stadt miteinander, wenn die Verbindungen der vorhandenen Straßenarchitektur folgen? Des Weiteren ist zu berücksichtigen, dass nicht jede Verbindung gleich teuer ist. Beim Legen einer Leitung ist beispielsweise die Verlegungstiefe im Erdreich nicht immer gleich. Als Antwort auf diese Frage dient der minimale Spannbaum des Graphen, der für das System steht.

Spannbäume finden darüber hinaus Anwendung in Gebieten wie der Clusteranalyse und der Echt-Zeit Gesichtserkennung. Die allgemeinen Eigenschaften von Spannbäumen sind für die verschiedensten Branchen von Bedeutung, weswegen sie auch in der Mathematik vielstudierte Objekte sind.

Ausgehend von einem bestimmten Graphen, welche Eigenschaften, wie Radius oder Durchmesser, kann man sich von einem Spannbaum erwarten? Um diese Frage zu beantworten ist es von Vorteil, eine Methode zu haben, um einen zufälligen Spannbaum eines Graphen zu erstellen. Diese Methode liefert der Groundskeeper Algorithmus. Ziel dieser Arbeit ist es, diesen Algorithmus zu beschreiben und anschließend zu beweisen, dass die dadurch konstruierten Spannbäume gleichverteilt auf der Menge aller Spannbäume des Graphen sind. Diese Arbeit orientiert sich am Artikel *The random walk construction of uniform spanning trees and uniform labelled trees* von D. J. Aldous.

2 Konstruktion und Beweis

2.1 Konstruktion

Sei im folgenden $G = (V, E)$ ein endlicher zusammenhängender ungerichteter Graph mit Knotenmenge V und Kantenmenge $E \subseteq V \times V$. Mit r_v bezeichnen wir den Grad, die Anzahl der Nachbarn, eines Knotens v aus V. Für zwei Knoten $v, w \in V$ schreiben wir $v \sim w$ falls $(v, w) \in E$, v, w Nachbarn sind. Mit $(X_j; j \geq 0)$ bezeichnen wir einen Randomwalk auf dem Graphen G mit einem zufällig ausgewählten Startknoten X_0. Für einen zufälligen Startknoten v gilt dann

$$X_j = \begin{cases} v & \text{if } j = 0 \\ w \text{ für } w \in \{w \in V : w \sim X_{j-1}\} & \text{if} j > 0 \end{cases}$$

wobei jedes $w \in \{w \in V : w \sim X_{j-1}\}$ gleich wahrscheinlich ist. Das heißt, dass für alle Knoten in V gilt, dass die Wahrscheinlichkeit, dass ein Knoten v aus V der erste Knoten $X_0 = v$ ist, $1/|V|$ ist. Für einen beliebigen Schritt $X_j = v$ mit $j \geq 0$ des Randomwalks und eine Kante (v, w) aus E gilt, dass die Wahrscheinlichkeit, dass der nächste Schritt $X_{j+1} = w$ ist, gleich $1/r_v$ ist. Der Randomwalk terminiert, wenn alle Knoten von V erschlossen wurden. Da G endlich ist, terminiert ein Randomwalk mit Wahrscheinlichkeit 1.

Auf Grundlage dieses Randomwalks kostruieren wir einen Spannbaum des Graphen G und gehen dabei folgendermaßen vor. Wir betrachten die verwendeten Kanten eines Randomwalks, also die Menge

$$\{(u, v) \in E \mid \exists\, i \geq 0 : X_i = u \ \wedge \ X_{i+1} = v\}$$

und entfernen die Kanten, durch die kein neuer Knoten durch den Randomwalk erschlossen wurde. Für eine genauere Beschreibung definieren wir den Zeitpunkt, an welchem ein Knoten das erste Mal entdeckt wurde. Wir bezeichnen diesen Zeitpunkt für jeden Knoten v als T_v, der folgendermaßen definiert ist:

$$T_v = \min\{j \geq 0 : X_j = v\}$$

Da der Randomwalk mit Wahrscheinlichkeit 1 terminiert, sind die T_v wohldefiniert. Wir können nun einen Teilgraph von G definieren, mit der Kantenmenge

$$E' := \{(X_{T_v-1}, X_{T_v})|v \in V \setminus X_0\} \tag{2.1}$$

Wir definieren den Teilgraph $\mathcal{T} = (V, E')$.

2.2 Spannbaum

Um zu zeigen, dass $\mathcal{T}$ ein Spannbaum ist, zeigen wir zunächst, dass $\mathcal{T}$ zusammenhängend ist.

Wir nummerieren dazu die Knoten in V in der Reihenfolge ihrer Entdeckung im Randomwalk und zeigen, dass der Graph, der durch die Knoten $V_N = \{v_1, \ldots, v_N\}$ und die Kantenmenge $E_N = \{(X_{T_v-1}, X_{T_v})|v \in V_N \setminus X_0\}$, $N \leq |V|$, definiert ist, zusammenhängend ist. Wir führen einen Induktionsbeweis über N.

Induktionsanfang: der Graph $V_1 = (\{v_1\}, \emptyset)$ ist als trivialer Graph zusammenhängend.

Induktionsvorraussetzung:
der Graph $G_N = (\{v_1, \ldots, v_N\}, \{(X_{T_v-1}, X_{T_v})|v \in V_N \setminus X_0\})$ mit $N < |V|$ ist zusammenhängend.

Induktionsschritt: Der Knoten von dem aus v_{N+1} entdeckt wurde, ist der Knoten $X_{T_{v_{N+1}}-1}$. Da dieser Knoten zuvor entdeckt worden sein muss, ist $X_{T_{v_{N+1}}-1}$ in V_N. Da nach der Induktionsvorraussetzung V_N zusammenhängend ist, existiert ein Pfad zwischen v_1 und v_N. Somit können wir diesen Pfad durch die Kante $(X_{T_{v_{N+1}}-1}, X_{T_{v_{N+1}}}) \in E_{N+1}$ erweitern und haben einen Pfad zwischen v_1 und v_{N+1} gefunden. Somit ist also v_1 mit jedem anderen Knoten in V_{N+1} verbunden, wodurch G_{N+1} zusammenhängend ist.

Wir haben also bewiesen, dass für $N \leq |V|$ der Graph G_N zusammenhängend ist. Dadurch ist insbesondere der Graph $G = G_{|V|} =$ zusammenhängend, was wir zeigen wollten.

Um zu zeigen, dass G ein Spannbaum ist, bleibt noch zu zeigen , dass es in G keine Kreise gibt. Dazu zeigen wir zunächst folgende Lemmata:

Lemma 2.2.1. *Ein Graph $G = (V, E)$ mit $|E| < |V|$ hat mindestens ein Blatt, also einen Knoten mit nur einem Nachbarn.*

Beweis. Wir nehmen an, ein Graph $G = (V, E)$ mit $|E| < |V|$ habe kein Blatt. Dann sind alle Knoten von G mindestens vom Grad 2. Summiert man die Grade der Knoten von V auf, zählt man alle Kanten doppelt, somit ergibt sich:

$$2|E| = \sum_{v \in V} r_v \geq 2|V|.$$

Und dadurch

$$|E| \geq |V|$$

was ein Widerspruch zur Annahme $|E| < |V|$ ist. Also hat jeder Graph $G = (V, E)$ mit $|E| < |V|$ mindestens ein Blatt. $\square$

Lemma 2.2.2. *Ein zusammenhängender Graph G mit n Knoten hat mindestens $n - 1$ Kanten.*

Beweis. Für $n \leq 3$ lassen sich alle möglichen Graphen leicht aufzeichnen, um die Aussage zu verifizieren.

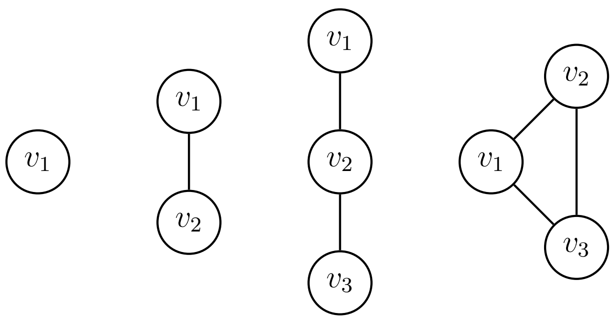

Abbildung 2.1: Alle zusammenhängenden Graphen mit 3 oder weniger Knoten

Seil also nun $n \geq 4$. Um einen Widerspruch zu erzeugen, betrachten wir den Graphen G mit minimaler Knotenanzahl n, dessen Anzahl der Kanten nicht größer als $n - 2$ ist.

Wir entfernen einen Knoten vom Grad 1, welcher aufgrund von Lemma 2.2.1 exisitiert, und dessen zugehörige Kante. Dadurch erhalten wir einen neuen zusammenhängenden Graphen G' mit Knotenanzahl $n - 1$ und weniger als $n - 2$ Kanten.

Dieser Graph ist zusammenhängend und hat mindestens 2 Kanten weniger als Knoten, wodurch der ursprüngliche Graph G nicht der Graph mit dieser Eigenschaft und minimaler Knotenanzahl gewesen sein kann. Somit kann dieser Graph G nicht existieren und ein zusammenhängender Graph G mit n Knoten hat mindestens $n-1$ Kanten.

□

Satz 2.2.3. *Der durch die Kanten in 2.1 definierte Graph ist kreisfrei.*

Beweis. Wir nehmen an es gäbe in dem durch 2.1 definierten Graphen $G = (V, E')$ einen Kreis. Dann können wir eine Kante e aus diesem Kreis entfernen, sodass der $G' = (V, E' \setminus e)$ immer noch zusammenhängend ist. Allerdings gilt $|E'| = |V| - 1$ und somit $|E' \setminus e| = |V| - 2$. Wir haben aber in Lemma 2.2.2 gezeigt, dass ein zusammenhängender Graph mit Knotenzahl $|V|$ mindestens $|V| - 1$ Kanten haben muss. Das ist ein Widerspruch und somit ist G kreisfrei.

□

Damit haben wir gezeigt das der durch 2.1 definierte Graph $\mathcal{T}$ein Spannbaum des originalen Graph G ist.

Jeder Spannbaum von G ist durch diese Konstruktion möglich. Einem bestimmten Spannbaum t könnten mehrere verschiedene Randomwalks zu Grunde liegen. Einer ist aber gerade jener, der durch die Tiefensuche auf dem Baum t bestimmt wird.

2.3 Gleichverteilung

Wir zeigen im Folgenden, dass die durch einen Randomwalk definierten Bäume gleichverteilt sind.

Dazu brauchen wir den Begriff eines stationären stochastischen Prozesses.

Definition 2.3.1 (stochastischer Prozess)**.** [6] Sei $(\Omega, \mathcal{F}, \mathbb{P})$ ein Wahrscheinlichkeitsraum und $(Z, \mathcal{Z})$ Messraum und T eine Indexmenge. Dann heißt eine Familie $X = (X_t)_{t \in T}$ messbarer Abbildungen

$$X_t : \Omega \to Z, t \in T$$

stochastischer Prozess (mit Zustandsraum Z).

Für uns ist ein Randomwalk $(X_j; j \geq 0)$ ein stochastischer Prozess mit dem Raum Ω aller möglichen Randomwalks auf G. Der Zustandsraum Z ist die Menge V der Knoten von G und T die Indexmenge $\mathbb{N}_0$.

Definition 2.3.2 (stationärer stochastischer Prozess)**.** [5] Ein stochastischer Prozess $(X_t)_{t\in T}$ mit der Indexmenge T heißt stationär, wenn die Verteilung von $(X_{s+t})_{t\in T}$ nicht von der Verschiebung $s \in T$ abhängt, also wenn gilt

$$\mathbb{P}_X((X_{s+t)t\in T}) = \mathbb{P}_X((X_t)_{t\in T})$$

für alle $s \in T$

Definition 2.3.3 (Markov-Kette)**.** [8] Ein stochastischer Prozess $(X_t)_{t\in\mathbb{N}_0}$ der nur Werte aus einem höchstens abzählbaren Zustandsraum $Z = \{z_1, \ldots, z_n\}$ annimmt, wird Markov-Kette genannt, wenn gilt:

$$\begin{aligned}\mathbb{P}(x_{t+1} = z_{t+1} | x_t = z_t, x_{t-1} = z_{t-1}, \ldots, x_0 = j_0)\\ = \mathbb{P}(x_{t+1} = z_{t+1} | x_t = z_t)\end{aligned}$$

alle $t \in \mathbb{N}$ und alle $(z_{t+1}, \ldots, z_0) \in Z^{t+2}$. Diese Eigenschaft nennt man auch Gedächtnislosigkeit. Die Größen

$$p_{z,v}(t) = \mathbb{P}(x_{t+1} = v | x_t = z)$$

werden Übergangswahrscheinlichkeiten genannt. Sind diese konstant, so spricht man von stationären Übergangswahrscheinlichkeiten und einer homogenen Markov-Kette. Die Matrix $P(t)$ mit Einträgen $p_{z,v}(t)$ mit $z, v \in V$ ist dann die Übergangsmatrix der Markov-Kette. Da wir nur mit homogenen Markov-Ketten zu tun haben, werden wir P für die Übergangsmatrix schreiben.

Wir bezeichnen im folgenden die Anzahl der Spannbäume von G mit $N(G)$ und die Menge aller gewurzelten Spannbäume von G mit $\mathcal{S}$. Um zu zeigen, dass die Verteilung der Spannbäume (ohne Wurzel) uniform ist, also

$$\mathbb{P}(\mathcal{T} = t) = \frac{1}{N(G)} = \frac{|V|}{|\mathcal{S}|},$$

mit einem Spannbaum t, betrachten wir zunächst die gewurzelten Spannbäume, die durch einen Randomwalk $(X_j; j \geq 0)$ definiert sind aber die Konstruktion wie in 2.1 zu einem späteren Zeitpunkt m startet. Bezeichne mit T_v^m den Index des ersten Besuches des Knotens v ab dem Index m, also

$$T_v^m = \min\{j \geq m : X_j = v\}.$$

Dann ist

$$S_m = (V, \{(X_{T_v^m - 1}, (X_{T_v^m}) | v \in V \setminus X_m\}) \in \mathcal{S}$$

der Spannbaum mit Wurzel X_m, der durch den Randomwalk $(X_m, X_{m+1}, X_{m+2}, \dots)$ mit $m \geq 0$ definiert wird. Dadurch erhalten wir eine Folge von gewurzelten Spannbäumen $(S_m)_{m \geq 0}$. Im nächsten Schritt betrachten wir einen Randomwalk $(X_j; -\infty < j < \infty)$ auf G, der mit den ganzen Zahlen indexiert ist. Der Randomwalk $(X_j; -\infty < j < \infty)$ induziert dann eine ebenfalls über die ganzen Zahlen indexierte Folge von gewurzelten Spannbäumen $(S_m; -\infty < m < \infty)$. Wir werden uns eine solche Folge von gewurzelten Spannbäumen in Rückwärtszeit

$$(S_m, S_{m-1}, S_{m-2}, \dots), \tag{2.2}$$

welche bei einem Index $m \in \mathbb{Z}$ beginnt, genauer ansehen.

Definition 2.3.4 (stationäre Verteilung). [6] Sei $(X_t)_{t \in T}$ eine Markov-Kette mit Indexmenge T, Zustandsraum Z und Übergangsmatrix P. Eine Verteilung $\boldsymbol{\pi}$ heißt stationär, falls für alle $v \in Z$ gilt:

$$\sum_{z \in Z} \boldsymbol{\pi}(z) p_{z,v} = \boldsymbol{\pi}(v) \tag{2.3}$$

Fasst man $\boldsymbol{\pi}$ als Zeilenvektor auf, so kann man 2.3 auch in der Form

$$\boldsymbol{\pi} P = \boldsymbol{\pi}$$

beschreiben.

Definition 2.3.5 (erreichbar, kommunizierend). [6] Sei $(X_t)_{t \in \mathbb{N}_0}$ eine Markov-Kette, mit Zustandsraum Z, Übergangsmatrix P und zwei Zuständen $i, j \in Z$. Der Zustand j heißt von i aus erreichbar, falls es einen Pfad von i nach j gibt. Das heißt,

$$\exists\, n \geq 1: \quad \mathbb{P}_X(X_{t+n} = j | X_t = i) > 0 \qquad t \in \mathbb{N}_0.$$

Ist i auch von j aus erreichbar, so heißen i und j kommunizierend.

Definition 2.3.6 (irreduzibel). [6] Ist $C \subset Z$ eine Teilmenge des Zustandsraums Z einer Markov-Kette und kommunizieren alle $i, j \in C$ miteinander, so heißt C irreduzibel. Ist Z irreduzibel, so heißt die Markov-Kette irreduzibel.

Lemma 2.3.1. *Sei P die Übergangsmatrix einer irreduziblen Markov-Kette und sei $A = [P - I, \mathbf{1}]$ die Matrix $P - I$ mit einer zusätzlichen letzten Spalte mit nur 1 als Einträgen. Dann gilt, rang$(A) = n$ wobei n die Anzahl der Zustände der Markov-Kette ist.*

Beweis. Da die Zeilen jeder Übergangsmatrix P aufsummiert 1 ergeben, gilt $P\mathbf{1} = \mathbf{1}$ und somit hat die Gleichung $Ax = 0$ die Lösung $(\mathbf{1}, 0)^T$. Sollte $\text{rang}(A) = n$ nicht gelten, so müsste es eine weitere nicht triviale Lösung $(\mathbf{y}, \alpha)^T$ geben, die orthogonal zu $(\mathbf{1}, 0)^T$ ist. Also muss gelten $\sum_i y_i = 0$, wobei aber $\mathbf{y} \neq 0$, da sonst auch $\alpha = 0$ gelten würde, wodurch die weitere Lösung trivial wäre. Wegen $A(\mathbf{y}, \alpha)^T = 0$ gilt $P\mathbf{y} + \alpha\mathbf{1} = \mathbf{y}$. Jeder Eintrag von $\mathbf{y}$ ist also eine Konvexkombination der Einträge von $\mathbf{y}$ plus α. Da die Markov-Kette irreduzibel ist, gibt es einen Zustand k, dessen zugehöriger Eintrag im Vektor $\mathbf{y}$ maximal ist und welcher auf dem Übergangsgraphen der Markov-Kette neben einem Zustand l liegt, dessen zugehöriger Eintrag im Vektor $\mathbf{y}$ geringer ist. Dieses Paar k, l muss existieren, da sonst alle Werte in $\mathbf{y}$ gleich groß wären, wodurch sofort $\mathbf{y} = 0$ folgen würde. Somit ist $p_{kl} \neq 0$ und dadurch $\mathbf{y}_k > \sum_i p_{ki}\mathbf{y}_i$. Da laut Annahme $\mathbf{y}_k = \sum_i p_{ki}\mathbf{y}_i + \alpha$ gelten muss, ist also $\alpha > 0$.

Wählen wir hingegen k als Zustand mit minimalen Wert in $\mathbf{y}$, so erhalten wir $\alpha < 0$ und somit einen Widerspruch.

Somit kann es keine zweite nicht triviale Lösung von $Ax = 0$ geben, wodurch $\text{rang}(A) = n$ gilt. [2] □

Korollar 2.3.1.1. *$dim(\{\pi : \pi P = \pi\}) \leq 1$.*

Beweis. Für ein π mit $\sum_i \pi_i = 1$ und π ist Linkseigenvektor von P muss gelten, $\pi A = (\mathbf{0}, 1)$. Aus $\pi A = (A^T \pi^T)^T$ und $\text{rang}(A) = \text{rang}(A^T)$ folgt dann $\dim(\{xA | x \in \mathbb{R}^n\}) = n$, womit die Abbildung $x \mapsto xA$ injektiv ist. Somit hat $\pi A = (\mathbf{0}, 1)$ höchstens eine Lösung und durch skalieren dieser Lösung erhalten wir den Raum $\{\lambda\pi : \lambda \in \mathbb{R}, \pi A = (\mathbf{0}, 1)\} = \{\pi : \pi P = \pi\}$, womit die Aussage gezeigt ist. [4] □

Definition 2.3.7. [2] Sei $\mathbf{p}(t)$ die Verteilung der Zustände einer Markov-Kette nach t Schritten, $\mathbf{p}(t)_i$ bezeichnet die relative Häufigkeit des Auftretens der Zustands i. Somit gilt klarerweise $\sum_i \mathbf{p}(t)_i$ für jedes t. Bezeichne mit $\mathbf{a}(t)$ die längerfristige Verteilung der Zustände.

$$\mathbf{a}(t) = \frac{1}{t}(\mathbf{p}(0) + \cdots + \mathbf{p}(t-1))$$

Satz 2.3.2 (Fundamentalsatz für Markov-Ketten)**.** *Für eine irreduzible Markov-Kette existiert eine eindeutige Verteilung π die $\pi P = \pi$ erfüllt und für längerfristige Verteilung $a(t)$ gilt,* $\lim_{t\to\infty} a(t) = \pi$

Beweis.

$$\begin{aligned}\mathbf{b}(t) &= \mathbf{a}(t)P - \mathbf{a}(t)\\ &= \frac{1}{t}(\mathbf{p}(1) + \cdots + \mathbf{p}(t)) - \frac{1}{t}(\mathbf{p}(0) + \cdots + \mathbf{p}(t-1))\\ &= \frac{1}{t}(\mathbf{p}(t) - \mathbf{p}(0))\end{aligned}$$

Also gilt $|\mathbf{b}(t)| \leq \frac{2}{t}$ und somit kovergiert $\mathbf{b}(t) = \mathbf{a}(t)P - \mathbf{a}(t)$ gegen $\mathbf{0}$. Dadurch konvergiert $\mathbf{a}(t)$ gegen eine Verteilung π für die $\pi P = \pi$ gilt. Diese Verteilung ist durch 2.3.1.1 eindeutig. □

Wir werden zeigen, dass die Folge in 2.2 eine Markov-Kette ist und dass $(S_m; -\infty < m < \infty)$ ein stationärer stochastischer Prozess ist , wodurch wir über die stationäre Verteilung dieser Markov-Kette, die Verteilung aller gewurzelten Spannbäume erhalten.

Lemma 2.3.3. *Ein gewurzelter Spannbaum S_i aus der Folge 2.2 mit $i > m$ ist vollständig durch (S_{i+1}, X_i) bestimmt.*

Beweis. Dem gewurzelten Spannbaum S_{i+1} liegt der Randomwalk $(X_j; j \geq i+1)$ zugrunde. Beginnen wir nun den Randomwalk bei X_i, anstelle von X_{i+1}, müssen wir eine neue Kante zu unserem Baum hinzufügen und zwar die Kante (X_i, X_{i+1}). Falls diese Kante bereits in S_{i+1} vorhanden war, gilt $S_i = S_{i+1}$, ansonsten ist die Kante, die hinzugefügt wurde als X_i in S_{i+1} entdeckt wurde in S_i nicht mehr vorhanden, da ja X_i der erste Knoten war. Der Rest des Baumes bleibt hingegen unverändert.[3] □

Der ausschlaggebende Punkt ist, dass wir die im zweiten Fall überflüssige Kante eindeutig durch S_{l+1} bestimmen können. Diese Kante ist nämlich die Letzte, vom eindeutigen Weg von X_{i+1} nach X_i, im Baum S_{i+1}, was folgende Grafik illustrieren soll.

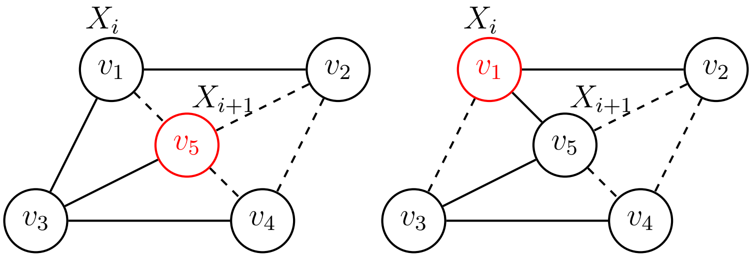

Abbildung 2.2: Beispiel zweier Spannbäume mit Wurzel in rot. Links der Spannbaum S_{i+1} und Rechts S_i. Die Kante (v_3, v_1) wurde als letzte Kante vom Weg von X_{i+1} nach X_i entfernt.

Wir haben somit gezeigt, dass die Folge aus 2.2 gedächtnislos ist. Fassen wir Ω bzw. Z aus 2.3.1 als Menge aller Folgen von gewurzelten Spannbäumen bzw. $\mathcal{S}$ (die Menge aller gewurzelter Spannbäume) auf, dann ist eine Familie $S^m = (S_i)_{i \leq m}$ mit $m \in \mathbb{Z}$ ein stochastischer Prozess. Somit ist jede Folge wie in 2.2 eine Markov-Kette.

Ein Randomwalk ist ebenfalls eine Markov-Kette, da die Übergangswahrscheinlichkeiten nur vom aktuellen Knoten abhängen. Da der Graph G zusammenhängend ist, ist die Markov-Kette die einen Randomwalk beschreibt irreduzibel und somit existiert durch 2.3.2 eine eindeutige stationäre Verteilung.

Lemma 2.3.4. *Die stationäre Verteilung $\boldsymbol{\pi}$ eines Randomwalks auf einem zusammenhängenden, endlichen, ungerichteten Graphen $G = (V, E)$ ist proportional zu dem Grad der Knoten. Genauer:*

$$\boldsymbol{\pi}(v) = \frac{r_v}{2|E|} \tag{2.4}$$

Beweis. Dazu müssen wir zeigen, dass der Vektor $\boldsymbol{\pi}$ ein Linkseigenvektor zum Eigenwert 1 der Übergangsmatrix einer Markov-Kette $(X_j; m \leq j < \infty)$ mit $m \in \mathbb{Z}$ ist. Sei P die Übergangsmatrix, mit Einträgen $p_{v,w} = \mathbb{P}(X_{j+1} = w | X_j = v)$ für $v, w \in V$, dann soll also gelten

$$\boldsymbol{\pi}^T P = \boldsymbol{\pi}^T$$

und somit

$$\sum_v \boldsymbol{\pi}(v) p_{v,w} = \boldsymbol{\pi}(w)$$

für alle $w \in V$. Durch Einsetzen erhalten wir für festes w

$$\sum_v \boldsymbol{\pi}(v) p_{v,w} = \sum_v \frac{r_v}{2|E|} p_{v,w} = \sum_{v \sim w} \frac{r_v}{2|E|} \frac{1}{r_v} = \frac{r_w}{2|E|} = \boldsymbol{\pi}(w)$$

und somit die Behauptung. □

Somit ist in einem Randomwalk $(X_j; -\infty < j < \infty)$ das Auftreten eines bestimmten Knotens nicht von der Zeit abhängig und dadurch zu jedem Zeitpunkt gleich wahrscheinlich. Ein Randomwalk indexiert mit den ganzen Zahlen, ist also ein stationärer stochastischer Prozess. Da ein gewurzelter Spannbaum $S_m,\ m \in \mathbb{Z}$ mit Wahrscheinlichkeit 1 von einer endlichen Folge von Knoten $(X_m, X_{m+1}, \ldots X_{m+n})$ abhängt, ist ein gewurzelter Spannbaum zu jedem Zeitpunkt gleich wahrscheinlich und $(S_m; -\infty < m < \infty)$ ein stationärer stochastischer Prozess.

Um die stationäre Verteilung der Markov-Kette von gewurzelten Spannbäumen in Rückwärtszeit wie in 2.2 zu ermitteln, wollen wir die Übergangsmatrix und deswegen die Übergangswahrscheinlichkeiten

$$\mathbb{P}(S_m = u | S_{m+1} = t)$$

betrachten, also die Wahrscheinlichkeiten, dass ein gewurzelter Spannbaum u auftritt bedingt durch den Nachfolger t. Dazu sind die Übergangswahrscheinlichkeiten eines Randomwalks $(X_j; -\infty < j < \infty)$ in Rückwärtszeit von Bedeutung

Lemma 2.3.5. *Für einen Randomwalk $(X_j; -\infty < j < \infty)$, beliebiges $m \in \mathbb{Z}$, $v, w \in V$ mit $v \sim w$ gilt*

$$\mathbb{P}(X_{m-1} = w | X_m = v) = \frac{1}{r_v}$$

Beweis. Für den Beweis nutzen wir, dass für beliebiges $v \in V$ und einen beliebigen Schritt $m \in \mathbb{Z}$ im Randomwalk $(X_j; -\infty < j < \infty)$ gilt,

$$\mathbb{P}(X_m = v) = \boldsymbol{\pi}(v),$$

mit $\boldsymbol{\pi}$, der stationären Verteilung des Randomwalks aus 2.3.4. Dann erhalten wir nach Definition der bedingten Wahrscheinlichkeit für $w \sim v$

$$\begin{aligned}
\mathbb{P}(X_{m-1} = w | X_m = v) &= \frac{\mathbb{P}(X_{m-1} = w, X_m = v)}{\mathbb{P}(X_m = v)} \\
&= \frac{\boldsymbol{\pi}(w)\mathbb{P}(X_m = v | X_{m-1} = w)}{\boldsymbol{\pi}(v)} \\
&= \frac{\frac{r_w}{2|E|}\frac{1}{r_w}}{\frac{r_v}{2|E|}} \\
&= \frac{1}{r_v}
\end{aligned}$$

und somit die Aussage. □

Bezeichne für einen Baum t, den Grad seiner Wurzel mit $r(t)$. Bedingt durch einen gewurzelten Spannbaum $S_i = u$ mit $i \leq m$ der Folge 2.2 von gewurzelten Spannbäumen, betrachten wir jetzt die Wahrscheinlichkeit für einen gewurzelten Spannbaum S_{i-1}. Dem Baum u liegt der Randomwalk $(X_j; j \geq i)$ zugrunde. Für den Vorgängerknoten X_{i-1} kommen dadurch $r(u)$ Knoten in Frage, wobei durch 2.3.5 jeder Knoten davon gleich wahrscheinlich ist. Somit gibt es auch $r(u)$ Bäume aus $\mathcal{S}$ die für S_{i-1} in Frage kommen und die alle gleich wahrscheinlich sind. Bezeichne die Menge dieser Bäume mit $\mathcal{D}(u)$. Dann gilt für festes $u \in \mathcal{S}$

$$\mathbb{P}(S_{i-1} = t | S_i = u) = \begin{cases} \frac{1}{r(u)} & \text{falls } t \in \mathcal{D}(u) \\ 0 & \text{sonst} \end{cases} \tag{2.5}$$

Gehen wir in 2.5 allerdings von festem $S_{i-1} = t \in \mathcal{S}$ aus, so gibt es in der Markov-Kette 2.2 der gewurzelten Spannbäume, $r(t)$ Nachfolger S_i von t, für die die Gleichung gilt. Die Menge dieser Bäume bezeichnen wir mit $\mathcal{C}(t)$

Mit 2.5 können wir die Übergangsmatrix der Markov-Kette (S_m, S_{m-1}, S_{m-2}) für $i \leq m$ aufstellen:

$$P = \begin{pmatrix} \mathbb{P}(S_{i-1} = t_1 | S_i = t_1) & \mathbb{P}(S_{i-1} = t_2 | S_i = t_1) & \mathbb{P}(S_{i-1} = t_3 | S_i = t_1) & \dots \\ \mathbb{P}(S_{i-1} = t_1 | S_i = t_2) & \mathbb{P}(S_{i-1} = t_2 | S_i = t_2) & \ddots & \\ \mathbb{P}(S_{i-1} = t_1 | S_i = t_3) & \ddots & & \\ \vdots & & & \end{pmatrix}$$

Diese Matrix hat aufgrund von

$$\begin{aligned} \sum_{t \in \mathcal{S}} r(t) \mathbb{P}(S_m = t' | S_{m+1} = t) &= \sum_{t \in \mathcal{C}(t')} r(t) \mathbb{P}(S_m = t' | S_{m+1} = t) \\ &= \sum_{t \in \mathcal{C}(t')} r(t) \frac{1}{r(t)} \\ &= r(t') \end{aligned}$$

den Linkseigenvektor $(r(t))_{t \in S}$ zum Eingenwert 1.[1]

Um die stationäre Wahrscheinlichkeit der Markov-Kette der gewurzelten Spannbäume zu erhalten, müssen wir diesen Vektor noch normieren.

$$\sum_{t\in\mathcal{S}} r(t) = N(G)\sum_{v\in V} r_v = 2N(G)|E|$$

Somit ist der Vektor

$$\frac{1}{2N(G)|E|}(r(t)_{t\in\mathcal{S}})$$

die gesuchte und wegen Satz 2.3.2 eindeutige stationäre Verteilung $\boldsymbol{\pi}$ der Markov-Kette.

Da $(S_m; -\infty < m < \infty)$ ein stationärer Prozess ist, ist die Verteilung $\boldsymbol{\pi}$ genau der Vektor der Wahrscheinlichkeiten

$$(\mathbb{P}(S_m = t))_{t\in\mathcal{S}}$$

wodurch dann

$$\mathbb{P}(S_m = t) = \frac{r(t)}{2|E||N(G)|}$$

für einen gewurzelten Spannbaum t gilt.

Somit hängt die Wahrscheinlichkeit des Auftretens eines Spannbaumes nur vom Grad seiner Wurzel hab.

Kehren wir nun wieder zu dem ursprünglichen Spannbaum $\mathcal{T}$, der durch den Randomwalk $(X_j; j \geq 0)$ definiert wird, zurück.

Ist die Verteilung des Startknotens X_0 die der stationären Verteilung 2.4 des Randomwalks über die ganzen Zahlen, so können wir auch die stationäre Wahrscheinlichkeit des Spannbaums anwenden. So ist dann

$$\mathbb{P}(\mathcal{T} = t) = \frac{r(t)}{2|E||N(G)|}$$

für einen gewurzelten Baum t. Wenn wir diese Wahrscheinlichkeit mit der Bedingung eines bestimmten Startknotens $X_0 = w$ versehen, wobei $w \in V$ beliebig ist, dann ist jeder Spannbaum mit w als Wurzel gleich wahrscheinlich, da diese Bäume alle den selben Grad der Wurzel haben. Da wir jeden Spannbaum von G mit jeder Wurzel w auffassen können, sind die entstehenden Bäume ohne Wurzel gleichverteilt. Ist nun X_0 bzw. w uniform, so ist immer noch jeder Spannbaum gleich wahrscheinlich.

■

3 Implementierung

Im folgenden Abschnitt, werden Skripte präsentiert, welche verwendet wurden, um die Bäume zu untersuchen, die durch die Implementierung des Algorithmus mit einem bestimmten Graphen entstanden sind. Die Skripten wurden verwendet, um Spannbäume zu generieren und anschließend für Eigenschaften wie Durchmesser, Anzahl der Knoten vom Grad k, Statistiken zu erstellen.

3.1 Generieren von Bäumen

Für die Implementierung in Python bietet sich die Library networkx an. Sie umfasst grundlegende Funktionen, die im Umgang mit Graphen essentiell sind. Zunächst werden wir Funktionen betrachten, die verwendet wurden, um Spannbäume zu generieren. Des Weiteren werden wir darstellen, wie diese Funktionen im Zusammenhang mit dem in 2.1 beschriebenen Algorithmus stehen.

3.1.1 Erstellen eines Randomwalks

Folgender Pythoncode wurde benutzt, um einen Randomwalk auf dem Graphen zu erstellen, der alle Knoten abdeckt.

```
1
2  def random_walk(graph):
3      random_node = random.choice(list(graph))
4
5      steps = 10**6
6
7      randomwalk = [random_node]
8
9      while len(list(dict.fromkeys(randomwalk))) \
10                                 != len(list(graph)):
11
12         i = 0
13         for i in range(steps):
14
15             neighbor_list = list(\
16                                  graph.neighbors(\
17                                  random_node))
18
19             random_node = random.choice(neighbor_list)
20             randomwalk.append(random_node)
21
22     return(randomwalk)
```

Abbildung 3.1: Funktion zum Erstellen eines Randomwalks auf einem Graphen. Der Randomwalk deckt alle Knoten des Graphen ab.

Nach der Übergabe des Graphen an die Funktion wird in Zeile 3 ein Startknoten ausgewählt, der unserem X_0 entspricht. Anschließend definieren wir die Anzahl der Schritte, die auf dem Graphen gegangen werden sollen, bevor kontrolliert wird, ob schon alle Knoten besucht wurden. Dies dient dazu, die Laufzeit der Funktion, gegenüber dem Kontrollieren nach jeder Iteration, drastisch zu minimieren. Die Abfolge der Knoten wird als Liste gespeichert. Um den Randomwalk laufen zu lassen, wird in der for-Schleife ein zufälliger Nachbarknoten des aktuellen Knotens ausgewählt und zur Liste hinzugefügt. Die äußere while-Schleife kontrolliert alle 600 000 Schritte, ob schon alle Knoten besucht wurden.

3.1.2 Erstellen eines Spannbaumes

Der folgende Code wurde verwendet, um auf Grundlage eines zuvor erstellten Randomwalks mit der Funktion aus 3.1.1, einen Spannbaum zu erstellen.

```
1
2  def create_tree(randomwalk):
3      T = nx.Graph()
4
5      T_geordnete_knoten = list(dict.fromkeys(randomwalk))
6
7      for node in T_geordnete_knoten[1:]:
8
9          i = randomwalk.index(node)
10         T.add_edge(randomwalk[i-1],node)
11
12     return(T)
```

Abbildung 3.2: Funktion zum Erstellen eines Spannbaumes durch einen Randomwalk

Nach Übergabe eines Randomwalks, der alle Knoten eines Graphen abdeckt, und der Initierung eines Baumes, erstellen wir in Zeile 5 eine Liste der Knoten in der Reihenfolge ihrer Entdeckung. Anschließend werden in der for-Schleife, die über die Knoten (bis auf den Startknoten) in Reihenfolge ihrer Entdeckung iteriert, die Kanten nach und nach hinzugefügt. Die Variable i entspricht hierbei genau

$$T_v = \min\{j \geq 0 : X_j = v\}.$$

da `index()` den Zeitpunkt der Entdeckung zurückgibt. Somit können wir die Kante

$$(X_{T_v - 1}, X_{T_v}) = \texttt{(randomwalk[i-1],node)}$$

zur Kantenmenge von T hinzufügen.

3.2 Statistiken anhand eines Beispiels

Als Beispielgraph wurde folgender Graph mit 9835 Knoten und 21856 Kanten gewählt. Der Radius dieses Graphen beträgt 25 und der Durchmesser 50.

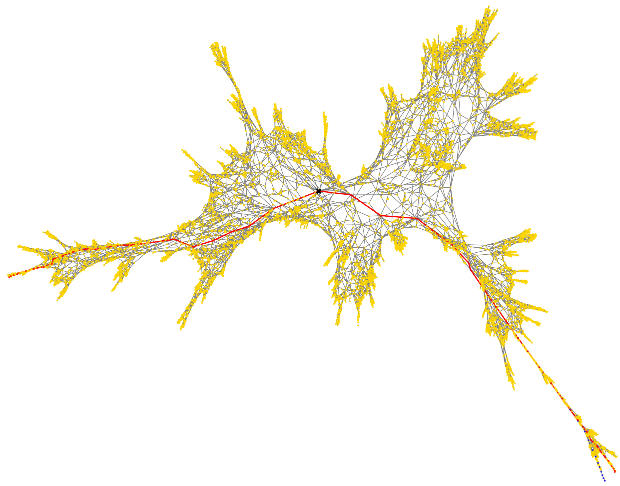

Abbildung 3.3: Beispielgraph in networkx geplottet. Zentrum des Graphen in schwarz in der Mitte und der Pfad des Durchmessers in rot

Ein Spannbaum, der mit den obigen Skripten erzeugt wurde, könnte dann folgendermaßen aussehen.

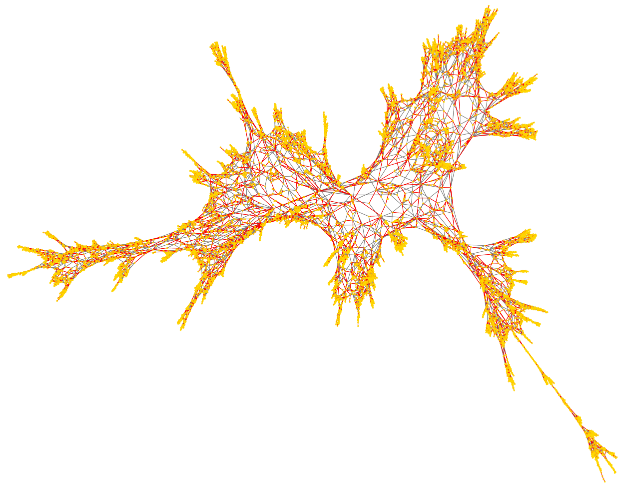

Abbildung 3.4: zufälliger Spannbaum des Beispielgraphen mit Kanten in rot eingezeichnet

Um die Grade der Knoten, den Durchmesser und den Radius der Bäume sowie die Höhe der Bäume mit Wurzel als Startknoten auszurechnen, wurde folgendes Skript verwendet.

```
1  #!/usr/bin/env python3
2
3  import networkx as nx
4  import numpy as np
5  from networkx.readwrite.graphml import write_graphml_lxml
6  import sys
7
8  if len(sys.argv) > 1:
9      n = int(sys.argv[1])
10
11 else:
12     print("Keine Anzahl neuer Bäume gegeben...")
13
14     n = int(input("n = "))
15
16 f = open("heights.txt","r+")
17
18 already_calculated = len(f.read().split(",")) - 1
19
20 for i in range( already_calculated + 1,\
21                 already_calculated+n+1):
22
23     tree = nx.read_graphml("Trees/tree"+str(i))
24
25     f.write(","+str(nx.eccentricity(tree,list(tree)[0])))
26
27     f.flush()
28
29 f.close()
```

Abbildung 3.5: Skript zum Berechnen von Höhen von gewurzelten Bäumen

Dieses Commandlineskript dient dazu, die Höhe bezüglich des Startknotens einer eingelesenen Anzahl an Bäumen zu berechnen. Die Höhen wurden in einer .txt Datei ”heights.txt”gespeichert und neuberechnete Höhen wurden in der Datei angehängt. Da die zuvor in 3.1 erstellte Bäume in dem Ordner ”Treesäls .graphml File abgespeichert wurden, werden aus diesem Ordner die Bäume eingelesen und anschließend ihre Höhe bestimmt.

Die Skripte um Durchmesser bzw. Radius eines Baumes zu bestimmen sind ident, mit der Ausnahme, dass in Zeile 24 nicht die Funktion `nx.eccentricity()` sondern die Funktionen `nx.diameter` bzw. `nx.radius` verwendet wurden und die ausgerechneten Höhen in einem anderen Textfile gespeichert wurden. Die Anzahl der Knoten mit bestimmten Grad wurde ebenso berechnet, mit dem Unterschied, dass die verschiedenen Anzahlen als Listen in einem Textfile gespeichert wurden.

Es folgen Plots, welche die Eigenschaften Höhe, Radius, Durchmesser und Grad der Knoten von generierten Spannbäumen beschreiben.

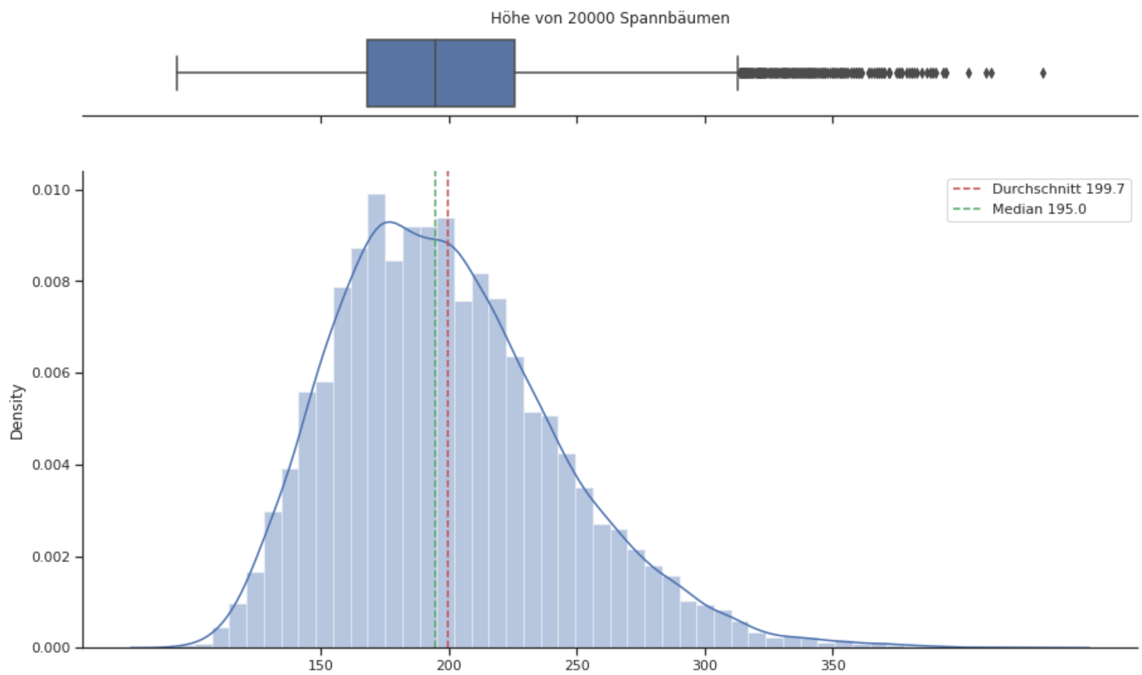

Abbildung 3.6: Höhe von 20 000 durch Randomwalks erzeugten Spannbäumen

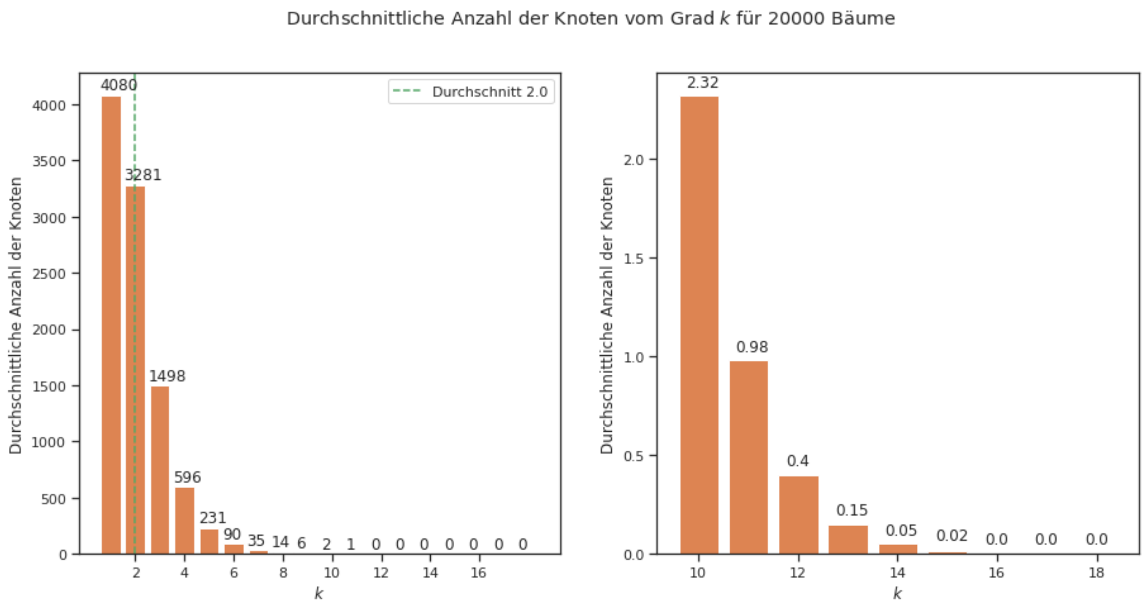

Abbildung 3.7: Durchschnittliche Anzahl der Knoten nach Grad von 20 000 Spannbäumen

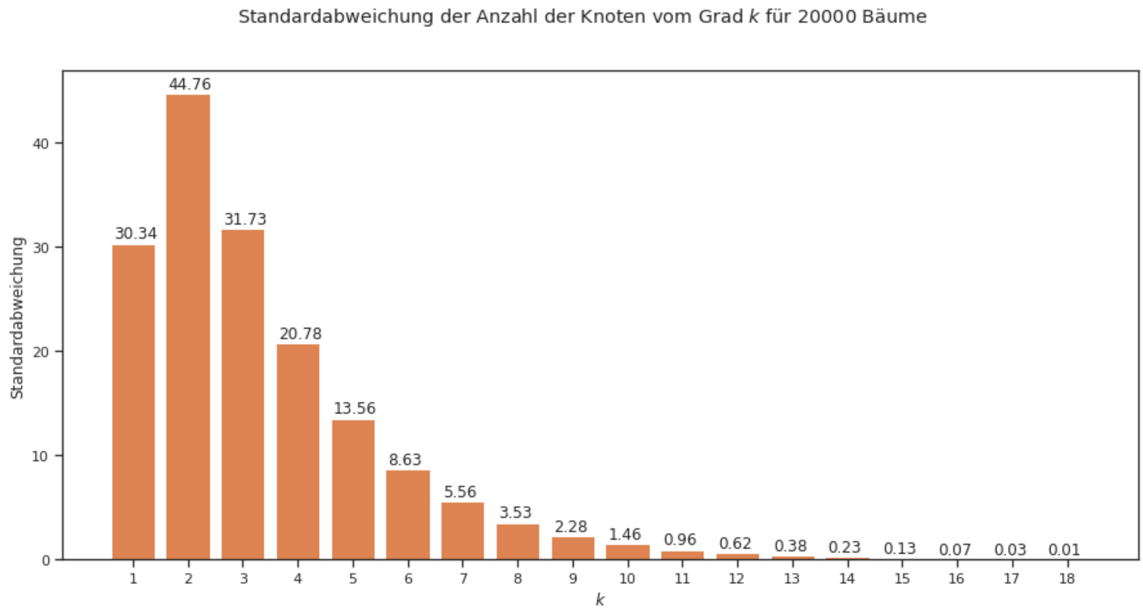

Abbildung 3.8: Standardabweichung der Anzahl der Knoten nach Grad von 20 000 Spannbäumen

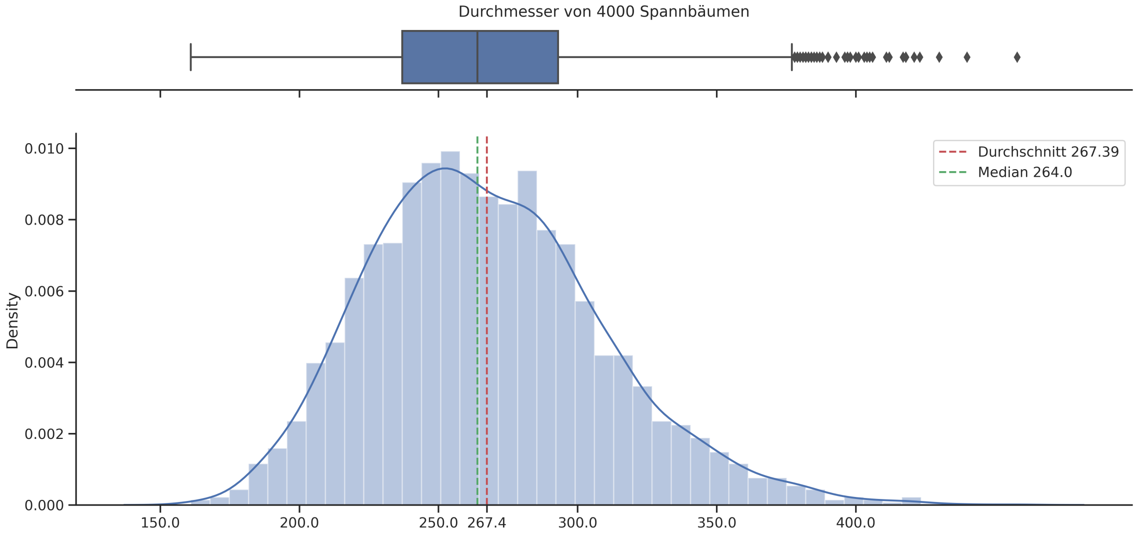

Abbildung 3.9: Durchmesser von 4 000 durch Randomwalks erzeugten Spannbäumen des Beispielgraphen.

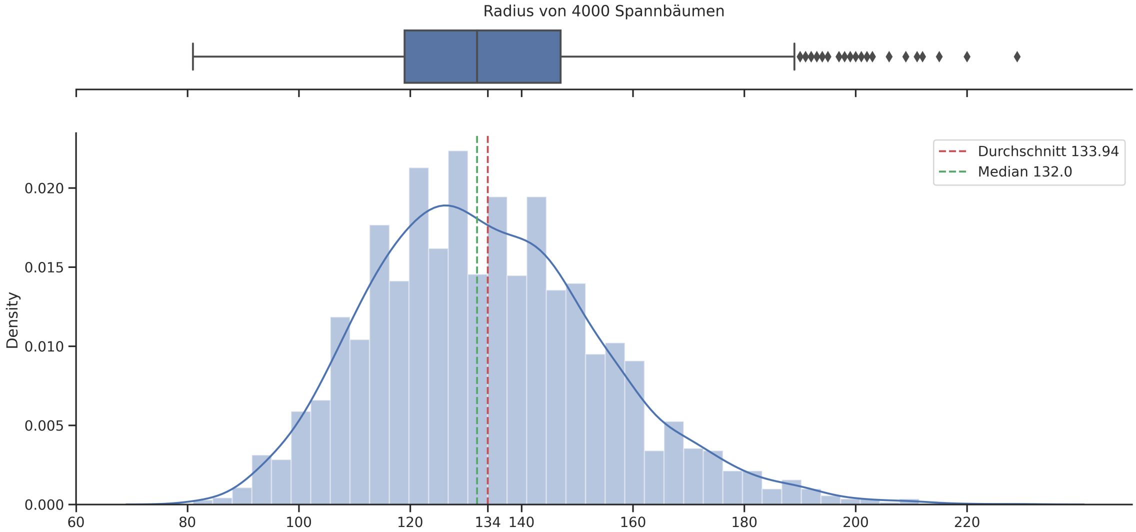

Abbildung 3.10: Radius von 4 000 durch Randomwalks erzeugten Spannbäumen

Literaturverzeichnis

[1] D. J. Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal on Discrete Mathematics*, 3(4):450–465, 1990.

[2] A. Blum, J. Hopcroft, and R. Kannan. *Random Walks and Markov Chains*, page 62–108. Cambridge University Press, 2020.

[3] W. Dana. The groundskeeper's algorithm works. http://www-personal.umich.edu/~willdana/Electrees%20Notes.pdf, 2019. [Online; accessed 25-February-2022].

[4] F. Eisenbrand. Randomized algorithms. https://www.epfl.ch/labs/disopt/wp-content/uploads/2018/09/lec7.pdf, 2014. [Online; accessed 25-February-2022].

[5] A. Klenke. *Wahrscheinlichkeitstheorie*. Springer Spektrum, 2013.

[6] D. Meintrup and S. Schäffler. *Stochastik Theorie und Anwendungen*. Springer, 2005.

[7] R. Pemantle. Uniform random spanning trees, 2004.

[8] N. Privault. *Understanding markov chains examples and applications*. Springer Singapore, 2018.

Abbildungsverzeichnis