Generierung diskreter Zufallsvariablen und Berechnung der Fréchetdistanz[1]

Karl Bringmann[2]

Abstract: Im ersten Teil dieser Dissertation untersuchen wir das fundamentale Problem der Generierung von Zufallsvariablen mit einer gegebenen diskreten Wahrscheinlichkeitsverteilung. Wir erweitern die klassische Lösung dieses Problems, Walkers Aliasmethode, in verschiedene Richtungen: Wir verbessern ihren Speicherbedarf, lösen den Spezialfall von sortierter Eingabe und untersuchen das Ziehen von natürlichen Verteilungen auf Maschinen mit beschränkter Präzision. Als Anwendung beschleunigen wir die Simulation eines physikalischen Modells.

Der zweite Teil dieser Dissertation gehört zum Gebiet der Geometrie und handelt von Algorithmen für die Fréchetdistanz, einem beliebten Ähnlichkeitsmaß für Kurven, das in quadratischer Zeit berechnet werden kann (bis auf logarithmische Faktoren). Wir zeigen die erste bedingte untere Schranke für dieses Problem: Unter der starken Exponentialzeithypothese ist keine Verbesserung der quadratischen Laufzeit um einen polynomiellen Faktor möglich. Zusätzlich präsentieren wir einen verbesserten Approximationsalgorithmus für realistische Eingabekurven.

1 Generierung diskreter Zufallsvariablen

Das Ziehen einer Zufallszahl mit einer gegebenen Wahrscheinlichkeitsverteilung ist ein fundamentales Problem im Bereich der randomisierten Algorithmen und ist heute wichtiger denn je, da in vielen Wissenschaften Computersimulationen mit Modellen durchgeführt werden, die eine Zufallskomponente enthalten. Wir betrachten dieses Gebiet aus der Perspektive der Algorithmentheorie. Dabei nehmen wir an, dass perfekter uniformer Zufall generiert werden kann, und untersuchen, inwiefern wir davon ausgehend von weiteren Verteilungen ziehen können. Das zentrale Problem im ersten Teil dieser Dissertation [Bri14a] ist das *proportionale Ziehen*. Hierbei sind nicht-negative Zahlen $p_1,\dots,p_n$ gegeben, welche eine Wahrscheinlichkeitsverteilung über $\{1,\dots,n\}$ definieren, indem die Zahl i eine Wahrscheinlichkeit proportional zu p_i hat, d.h. indem i mit Wahrscheinlichkeit $\frac{p_i}{\sum_j p_j}$ gezogen wird. Die Aufgabe ist es, eine Datenstruktur zu bauen, die auf Anfrage eine Zahl von der Eingabeverteilung zieht und zurückgibt. Die klassische Lösung dieses Problems ist die vierzig Jahre alte Aliasmethode von Walker [Wal74], die eine Anfragezeit von $O(1)$ hat, d.h. von der Eingabeverteilung zu ziehen ist in konstanter Zeit möglich, und eine Vorberechnungszeit von $O(n)$ benötigt, d.h. die Datenstruktur wird in Zeit $O(n)$ aufgebaut. Man sieht leicht, dass beide Laufzeiten der Aliasmethode optimal sind, dass also insbesondere keine Datenstruktur eine garantierte Vorberechnungszeit von $o(n)$ auf allen Eingabeverteilungen haben kann, da bei Verteilungen der Form $0,\dots,0,1,0,\dots,0$ das Ziehen von der Verteilung dem Finden der Eins gleichkommt, sodass das Suchen in

[1] Englischer Titel der Dissertation: “Sampling from Discrete Distributions and Computing Fréchet Distances”

[2] ETH Zürich, Institut für Theoretische Informatik, Universitätstrasse 6, 8092 Zürich, karlb@inf.ethz.ch

einem unsortierten Array auf proportionales Ziehen reduziert werden kann. Wir erweitern diese klassische Datenstruktur folgendermaßen in verschiedene Richtungen:

Optimaler Speicherbedarf Während die Laufzeitschranken gut verstanden sind, hat der Speicherbedarf von Algorithmen zur Generierung von Zufallsvariablen bisher wenig Aufmerksamkeit erhalten. Dies liegt vor allem daran, dass solche Algorithmen typischerweise auf dem RealRAM-Modell analysiert werden, in dem jede Speicherzelle eine beliebige reelle Zahl speichern kann, also insbesondere beliebig viele Bits an Information. Ein realistischeres Maschinenmodell ist die WordRAM, bei der jede Speicherzelle $w = \Omega(\log n)$ Bits speichert. Wir zeigen, dass Walkers Aliasmethode auf der WordRAM implementiert werden kann und dort einen Speicherbedarf von $n(w + 2\lg n + O(1))$ Bits hat [BL13]. Da das bloße Speichern der Eingabeverteilung $p_1, \dots, p_n$ insgesamt nw Bits benötigt, hat Walkers Aliasmethode eine *Redundanz* von $2n\lg n + O(n)$ Bits. Wir untersuchen ob dieser Speicherbedarf verringert werden kann. Dabei konzentrieren wir uns auf zwei Modelle von Datenstrukturen, die dem Gebiet der "succinct data structures" entlehnt sind: Im systematischen Modell, in dem die Eingabe $p_1, \dots, p_n$ schreibgeschützt ist, präsentieren wir für jedes r eine Datenstruktur mit einer Redundanz von $r + O(w)$, einer erwarteten Anfragezeit von $O(n/r)$ und einer Vorberechnungszeit von $O(n)$. Für $r = n$ verbessert dies die Redundanz der Aliasmethode bereits um einen Faktor $\Omega(\log n)$, obwohl die Aliasmethode nicht systematisch ist. Wir ergänzen diese Datenstruktur um eine untere Schranke, die zeigt, dass unser Kompromiss zwischen Redundanz und Anfragezeit für systematische Datenstrukturen optimal ist. Im nicht-systematischen Modell, in dem die Eingabe geschickter kodiert werden kann als sie einen Wert nach dem anderen zu speichern, veranschaulichen wir einen überraschenden Unterschied zum systematischen Modell: Mit nur einem Bit Redundanz ist es möglich, eine optimale erwartete Anfragezeit von $O(1)$ und Vorberechnungszeit von $O(n)$ zu erhalten. Diese Ergebnisse verbessern nicht nur den Speicherbedarf der klassischen Lösung zur Generierung diskreter Verteilungen, sie liefern zudem die stärkste Separierung zwischen systematischen und nicht-systematischen Datenstrukturen. Außerdem sind unsere Datenstrukturen genauso einfach wie Walkers Aliasmethode und sollten auch praktisch effizient sein.

Eingeschränkte Verteilungen Da die Vorberechnungs- und Anfragezeiten von Walkers Aliasmethode im Sinne der Worst-Case-Analyse optimal sind, untersuchen wir Situationen, in denen zusätzliches Wissen über die Eingabeverteilung vorhanden ist. In erster Linie betrachten wir die Garantie, dass die Eingabeverteilung $p_1, \dots, p_n$ sortiert ist. Wir zeigen, dass in diesem Fall die Vorberechnungszeit auf $O(\log n)$ reduziert werden kann, während die erwartete Anfragezeit nach wie vor $O(1)$ ist. Dies bedeutet, dass unsere Datenstruktur nach dem Ziehen der ersten Zufallsvariablen nur einen Bruchteil der Eingabezahlen gelesen hat. Die Vorberechnungszeit kann auf Kosten der Anfragezeit noch weiter verringert werden. Genauer können wir jede erwartete Anfragezeit $O(t)$ nach einer Vorberechnungszeit von $O(\log_t n)$ erreichen. Insbesondere gibt es eine Datenstruktur mit Vorberechnungs- und erwarteter Anfragezeit $O(\log n / \log\log n)$. Wir ergänzen diesen Kompromiss zwischen Vorberechnungs- und Anfragezeit durch eine passende untere Schranke.

Generierung zufälliger Teilmengen Beim *Teilmengenziehen* sind $p_1, \dots, p_n$ gegeben und wir betrachten n unabhängige Ereignisse, wobei Ereignis i mit Wahrscheinlichkeit p_i auftritt. Die Aufgabe ist es, die Menge der eintretenden Ereignisse zu generieren. Dieses Problem kann als Verallgemeinerung des proportionalen Ziehens gesehen werden, weil wir zeigen, dass jede Datenstruktur für das Teilmengenziehen in eine Datenstruktur für proportionales Ziehen umgewandelt werden kann ohne die asymptotische Laufzeit zu beeinträchtigen [BP12]. Wie für proportionales Ziehen untersuchen wir auch für das Teilmengenziehen sortierte und unsortierte Eingabesequenzen und präsentieren in beiden Fällen neue Datenstrukturen mit einer optimalen Kurve von Kompromissen zwischen Vorberechnungs- und Anfragezeit. Die Situation für das Teilmengenziehen ist komplexer, da die Laufzeit nun nicht mehr nur von der Eingabegröße n sondern auch von der erwarteten Größe μ der gezogenen Teilmenge abhängt. Beispielsweise entwerfen wir eine Datenstruktur für das Teilmengenziehen auf sortierter Eingabe mit Vorberechnungs- und erwarteter Anfragezeit $O(1 + \mu + \frac{\log n}{\log(\log(n)/\mu)})$, allerdings ist dies nur ein Punkt auf einer optimalen Kurve von Kompromissen zwischen Vorberechnungs- und Anfragezeit.

Spezielle Verteilungen Besonders schnelle Methoden zur Generierung von Zufallsvariablen sind für die speziellen Verteilungen bekannt, die in der Wahrscheinlichkeitstheorie allgegenwärtig sind, z.B. Bernoulliverteilung, geometrische Verteilung und Binomialverteilung. Um beispielsweise von der geometrischen Verteilung $\mathrm{Geo}(p)$ zu ziehen reicht es, die einfache Formel $\lceil \frac{\log R}{\log(1-p)} \rceil$ auszuwerten, wobei R eine uniform zufällige reelle Zahl in $(0,1)$ ist. Im RealRAM-Modell, in dem Operationen auf reellen Zahlen in konstanter Zeit möglich sind, kann diese Formel in Zeit $O(1)$ ausgewertet werden. Auf realen Computern wird die Formel allerdings typischerweise mit der üblichen Fließkommapräzision verwendet, sodass sie nicht exakt ist. Daher untersuchen wir die Frage, ob man von den genannten speziellen Verteilungen auf einem Maschinenmodell mit beschränkter Präzision (wie der WordRAM) exakt und effizient ziehen kann [BF13]. Wir beweisen, dass auf der WordRAM eine Zufallsvariable mit geometrischer Verteilung $\mathrm{Geo}(p)$ in erwarteter Laufzeit $O(1 + \log(1/p)/w)$ generiert werden kann. Die ist optimal, da es asymptotisch der erwarteten Anzahl an Speicherzellen für die Ausgabe entspricht. Um diese Laufzeit zu erreichen, müssen wir auf obige einfache Formel verzichten, da es ein bekanntes offenes Problem ist, ob Logarithmen in linearer Zeit berechnet werden können. Zusätzlich präsentieren wir optimale WordRAM-Algorithmen sowohl für die Bernoulli- und Binomialverteilung als auch für Erdős-Rényi-Zufallsgraphen.

Anwendungen Unsere Einsichten zu den bisher untersuchten grundlegenden Problemen der der randomisierten Algorithmik finden eine Anwendung in der Generierung von komplexen, physikalisch motivierten Zufallsstrukturen. Betrachten wir den folgenden exemplarischen Prozess. Das interne diffusionsbegrenzte Wachstum (engl. Internal Diffusion Limited Aggregation, IDLA [MD86]) plaziert Partikel auf dem anfangs leeren Gitter $\mathbb{Z}^2$. In jedem Schritt entsteht ein neues Partikel am Koordinatenursprung und folgt einer zufälligen Irrfahrt, bis es auf einen freien Gitterpunkt trifft, welchen es besetzt. Dieser Prozess modelliert bestimmte chemische und physikalische Phänomene wie Korrosion und das Schmelzen eines Festkörpers um eine Wärmequelle. Dabei bildet sich ungefähr ein Kreis heraus. Dieses Verhalten rigoros zu beweisen ist ein schweres mathematisches Problem, das erst kürzlich gelöst wurde [JLS12]. Aus Sicht der Informatik ist die Generierung der IDLA-Struktur nach Plazieren von n Partikeln ein algorithmisches Problem. Die triviale Simulation des Prozesses benötigt eine erwartete Laufzeit von $\Theta(n^2)$. Wir präsentieren einen Algorithmus mit Laufzeit $O(n \log^2 n)$ und Speicherbedarf $O(\sqrt{n} \log n)$ [BKP+14], was Experimente mit viel größeren IDLA-Strukturen ermöglicht.

2 Berechnung der Fréchetdistanz

Der zweite Teil dieser Dissertation gehört zum Gebiet der algorithmischen Geometrie und behandelt Algorithmen zur Berechnung der Fréchetdistanz, einem beliebten Ähnlichkeitsmaß für Kurven. Intuitiv ist die Fréchetdistanz zweier Kurven P, Q die minimale Länge einer Leine mit der man einen Hund und seinen Halter verbinden kann während sie P beziehungsweise Q ablaufen ohne umzukehren. Zur Definition dieses Maßes sehen wir eine Kurve als eine stetige Abbildung $P\colon [0,1] \to \mathbb{R}^d$ an. Eine *Traversierung* einer Kurve ist eine stetige, monoton wachsende Funktion $\phi\colon [0,1] \to [0,1]$, sodass $P \circ \phi$ eine Reparametrisierung der Kurve P ergibt. Die Fréchetdistanz zweier Kurven P_1, P_2 ist nun definiert als $\min_{\phi_1,\phi_2} \max_{0 \le t \le 1} \|P_1(\phi_1(t)) - P_2(\phi_2(t))\|$, wobei ϕ_1, ϕ_2 über alle Traversierungen laufen. Dabei geben ϕ_1 und ϕ_2 die variablen Geschwindigkeiten an, mit denen der Hund und sein Herrchen die Kurven P_1 beziehungsweise P_2 ablaufen, und die Länge der Leine wird bestimmt durch die maximale Distanz $\max_{0 \le t \le 1} \|P_1(\phi_1(t)) - P_2(\phi_2(t))\|$, die Hund und Herrchen annehmen.

Alt und Godau führten dieses Maß 1991 in der algorithmischen Geometrie ein [AG95, God91]. Für Polygonzüge P_1 und P_2 mit n beziehungsweise m Eckpunkten, $n \ge m$, entwarfen sie einen Algorithmus mit Laufzeit $O(nm \log(nm))$. Seit Alt und Godaus wegweisender Arbeit ist die Fréchetdistanz zu einem fruchtbaren Teilgebiet der algorithmischen Geometrie geworden, in der viele Varianten und Verallgemeinerungen untersucht werden (siehe z.B. [AB10, DHPW12, CCdVE+10, BBW09]). Als natürliches Maß für die Ähnlichkeit zweier Kurven hat die Fréchetdistanz viele Anwendungen gefunden, beispielsweise in der Verifikation von Unterschriften (siehe z.B. [MP99]), dem Abgleichen von Karten und Trackingdaten (siehe z.B. [BPSW05]) und der Analyse sich bewegender Objekte (siehe z.B. [BBG+11]).

Quadratische Zeitkomplexität? In den letzten Jahren wurden verbesserte Algorithmen für mehrere Varianten der Fréchetdistanz gefunden. Agarwal et al. [AAKS13] zeigten, dass eine diskrete Variante der Fréchetdistanz in subquadratischer Zeit $O\big(nm \frac{\log\log n}{\log n}\big)$ berechnet werden kann. Buchin et al. [BBMM14] entwarfen einen Algorithmus für die klassische Fréchetdistanz mit einer Laufzeit von $O(n^2 \sqrt{\log n} (\log\log n)^{3/2})$ auf der Real RAM und $O(n^2 (\log\log n)^2)$ auf der Word RAM. Diese Resultate berühren allerdings nicht die offene

Frage, ob es einen *stark subquadratischen*[3] Algorithmus für die Fréchetdistanz gibt, also einen Algorithmus mit Laufzeit $O(n^{2-\delta})$ für eine Konstante $\delta > 0$.

Die einzige bekannte untere Schranke zeigt, dass das Berechnen der Fréchetdistanz Zeit $\Omega(n \log n)$ benötigt (im algebraischen Entscheidungsbaummodell [BBK$^+$07]), dies ähnelt dem bekannten Fakt, dass vergleichsbasiertes Sortieren Zeit $\Omega(n \log n)$ benötigt. Die typische Art und Weise (bedingte) quadratische untere Schranken für geometrische Probleme zu zeigen ist 3SUM-Härte [GO95]. Tatsächlich stellte Helmut Alt die Vermutung auf, die Fréchetdistanz sei 3SUM-hart [BBMM14], allerdings ist diese Vermutung nach wie vor offen. Statt eine Verbindung zwischen der Fréchetdistanz und 3SUM zu zeigen, betrachten wir die starke Exponentialzeithypothese.

Starke Exponentialzeithypothese Die starke Exponentialzeithypothese (engl. Strong Exponential Time Hypothesis) wurde von Impagliazzo, Paturi und Zane [IPZ01, IP01] eingeführt und postuliert, dass es keine Algorithmen für das Erfüllbarkeitsproblem gibt, die viel schneller sind als die erschöpfende Suche. Dies erlaubt bedingte untere Schranken für weitere Probleme zu beweisen. Um die starke Exponentialzeithypothese zu definieren, erinnern wir an das k-SAT-Problem: Die Eingabe ist eine Formel ϕ mit N Variablen und M Klauseln, die ODER-Verknüpfungen von k negierten oder unnegierten Variablen sind. Die Aufgabe ist zu entscheiden, ob es eine Wahrheitswertbelegung der Variablen gibt, die alle Klauseln von ϕ erfüllt.

Starke Exponentialzeithypothese: *Es gibt kein $\delta > 0$, sodass für alle* k *das* k-SAT-*Problem einen $O((2-\delta)^N)$-Algorithmus hat.*

Wir erinnern daran, dass eine erschöpfende Suche zur Lösung des Erfüllbarkeitsproblems Zeit $O^*(2^N)$ benötigt. Die schnellsten Algorithmen für k-SAT sind Varianten des bekannten PPSZ-Algorithmus [PPSZ05] und haben eine Laufzeit, die nur etwas schneller ist als eine erschöpfende Suche, nämlich von der Form $O(2^{(1-c/k)N})$ für eine Konstante $c > 0$. Insbesondere ist also kein Algorithmus für k-SAT mit Laufzeit $O(1.99^N)$ bekannt. Daher ist SETH eine vernünftige Annahme, die eine Barriere formalisiert, die mit heutigen Methoden undurchdringbar scheint. Es ist bekannt, dass es die starke Exponentialzeithypothese sogar erlaubt, bedingte untere Schranken für Polynomialzeitprobleme zu zeigen, z.B. für k-Dominating Set [PW10] und den Durchmesser von dünnbesetzten Graphen [RVW13]. Wir zeigen eine ähnliche untere Schranke für die Fréchetdistanz.

Hauptresultat Unser Hauptresultat des zweiten Teils dieser Dissertation liefert ein starkes Indiz dafür, dass die Fréchetdistanz keine stark subquadratischen Algorithmen hat.

Wir beweisen, dass es unter der starken Exponentialzeithypothese für kein $\delta > 0$ einen Algorithmus mit Laufzeit $O(n^{2-\delta})$ für die Fréchetdistanz gibt [Bri14b].

[3] Wir nutzen die Bezeichnung *stark subquadratisch* um zwischen einer polynomiellen Verbesserung von der Form $O(n^{2-\delta})$ und logarithmischen Verbesserungen wie dem $O(n^2 \log\log n / \log n)$-Algorithmus von Agarwal et al. [AAKS13] zu unterscheiden.

Da die starke Exponentialzeithypothese eine glaubhafte Annahme ist, kann man es durch dieses Resultat als unwahrscheinlich ansehen, dass die Fréchetdistanz stark subquadratische Algorithmen besitzt. Insbesondere würde ein stark subquadratischer Algorithmus für die Fréchetdistanz nicht nur verbesserte Algorithmen für das Erfüllbarkeitsproblem liefern, sondern auch für weitere Probleme wie HittingSet, SetSplitting und NAE-SAT über die Reduktionen in [CDL$^+$12]. Alternativ, im Geiste von Pătraşcu und Williams [PW10], kann man unser Resultat auch als möglichen Angriff auf das Erfüllbarkeitsproblem ansehen, da nun schnellere Algorithmen für die Fréchetdistanz einen möglichen Weg zu verbesserten Erfüllbarkeitsalgorithmen darstellen. Auf jeden Fall sollte man sich bewusst sein, dass schnellere Algorithmen für die Fréchetdistanz zu finden mindestens so schwer ist wie ein Durchbruch für das Erfüllbarkeitsproblem, was unmöglich sein könnte.

Wir weisen darauf hin, dass unsere unteren Schranken (soweit nicht anders angegeben) für Kurven in der Euklidischen Ebene gelten, und damit auch für Kurven im $\mathbb{R}^d$ für jedes $d \geq 2$.

Beweisskizze des Hauptresultats Um unser Hauptresultat zu zeigen, entwerfen wir eine Reduktion von k-SAT auf die Fréchetdistanz. Gegeben eine k-SAT-Instanz φ, partitionieren wir ihre Variablen in zwei gleich große Mengen V_1, V_2. Um eine erfüllende Belegung von φ zu finden, müssen wir Belegungen b_1 von V_1 und b_2 von V_2 wählen, sodass jede Klausel von b_1 oder b_2 erfüllt wird. Wir konstruieren zwei Kurven P_1, P_2, wobei P_k für die Wahl von b_k verantwortlich ist. Die Kurve P_k besteht aus je einem *Belegungsgadget* für jede Belegung von V_k. Belegungsgadgets bestehen aus je einem *Klauselgadget* für jede Klausel von φ, das kodiert, ob die Klausel von der Belegung erfüllt wird. Weiterhin stellen wir sicher, dass die Belegungsgadgets von Belegungen b_1 von V_1 und b_2 von V_2 genau dann eine Fréchetdistanz kleiner gleich 1 haben, wenn (b_1, b_2) eine erfüllende Belegung von φ darstellt. In P_1 und P_2 verknüpfen wir diese Belegungsgadgets mit weiteren Kurvenabschnitten um ein logisches ODER zu simulieren, welches erzwingt dass zwei Belegungsgadgets parallel abgelaufen werden. Wenn φ nicht erfüllbar ist, dann hat jedes Paar von Belegungsgadgets Fréchetdistanz größer als 1, und somit haben auch P_1, P_2 eine Fréchetdistanz größer als 1. Wenn andererseits eine erfüllende Belegung (b_1, b_2) für φ existiert, dann stellen wir sicher, dass P_1 und P_2 abgelaufen werden können während im Prinzip nur die Belegungsgadgets zu b_1, b_2 parallel abgelaufen werden, was eine Fréchetdistanz kleiner gleich 1 liefert.

Da die Kurven P_1, P_2 ein Belegungsgadget für jede Belegung von einer Hälfte der Variablen haben und da wir sicherstellen, dass die Größe von Belegungsgadgets polynomiell in der Anzahl an Klauseln M ist, haben die konstruierten Kurven $n = O^*(2^{N/2})$ Eckpunkte. Also würde jeder Algorithmus für die Fréchetdistanz mit Laufzeit $O(n^{2-\delta})$ einen Algorithmus für k-SAT mit Laufzeit $O^*(2^{(1-\delta/2)N})$ liefern, was der starken Exponentialzeithypothese widerspricht.

Erweiterungen Wir erweitern unser Hauptresultat in zwei wichtige Richtungen: Wir zeigen Approximationshärte und liefern scharfe untere Schranken für den Fall, dass eine

Kurve viel weniger Eckpunkte hat als die andere, $m \ll n$. Um unser Resultat ausdrücken zu können, formalisieren wir zuerst was es bedeutet, dass eine Aussage "eingeschränkt auf $m \approx n^\gamma$ für alle γ gilt". Eine Aussage gelte für *jede polynomielle Einschränkung von* $n^{\gamma_0} \leq m \leq n^{\gamma_1}$, falls für alle $\delta > 0$ und $\gamma_0 + \delta \leq \gamma \leq \gamma_1 - \delta$ die Aussage eingeschränkt auf $n^{\gamma-\delta} \leq m \leq n^{\gamma+\delta}$ gilt.

Wir beweisen, dass es unter der starken Exponentialzeithypothese für kein $\delta > 0$ *eine 1.001-Approximation für die Fréchetdistanz mit Laufzeit* $O((nm)^{1-\delta})$ *gibt. Dies gilt für jede polynomielle Einschränkung von* $1 \leq m \leq n$ *[Bri14b].*

Realistische Eingabekurven Wie wir gesehen haben, stellt die quadratische Zeitkomplexität anscheinend eine Barriere für die Fréchetdistanz dar. Da diese Laufzeit für viele Anwendungen zu groß ist, gibt es einige Ansätze, diese Barriere wenigstens für realistische Instanzen zu durchbrechen. Zu diesem Zweck wurden mehrere eingeschränkte Klassen von Kurven untersucht, siehe Rückgratkurven [AHPK+06], κ-beschränkte und κ-geradlinige Kurven [AKW04] und Kurven von ϕ-geringer Dichte [DHPW12]. Das gängigste Modell realistischer Kurven sind die sogenannten *c-gepackten Kurven*. Eine Kurve π in $\mathbb{R}^d$ ist c-gepackt, wenn für jeden Punkt $z \in \mathbb{R}^d$ und jeden Radius $r > 0$ die Gesamtlänge von π im Ball $B(z,r)$ höchstens cr ist, wobei $B(z,r)$ der Ball mit Radius r und Mittelpunkt z ist. Dieses Modell ist aus praktischer Sicht gut begründet. Driemel et al. [DHPW12] führten c-gepackte Kurven ein und entwarfen eine $(1+\varepsilon)$-Approximation für die Fréchetdistanz auf c-gepackten Kurven im euklidischen Raum $\mathbb{R}^d$, $d \geq 2$, mit Laufzeit $O(cn/\varepsilon + cn \log n)$. Darüber hinaus sind bereits viele Varianten der Fréchetdistanz auf diesem Modell realistischer Kurven untersucht worden [CDG+11, HPR11, DHP13, GS13].

Für kleine c und $1/\varepsilon$ läuft der Algorithmus von Driemel et al. in nahezu linearer Zeit. Allerdings ist es nicht klar, ob die Abhängigkeit von c und ε optimal ist, wenn c und $1/\varepsilon$ mit n wachsen können. Beispielsweise benötigt dieser Algorithmus selbst für $c = O(1)$ quadratische Laufzeit um eine $(1+\frac{1}{n})$-Approximation zu berechnen - in ungefähr der gleichen Zeit kann man stattdessen den exakten Algorithmus von Alt und Godau ausführen, der keine Annahmen an die Eingabe stellt. Wir liefern ein starkes Indiz dafür, dass der Algorithmus von Driemel et al. für jede Konstante $0 < \varepsilon \leq 0.001$ eine optimale Abhängigkeit von c und n hat.

Wir beweisen, dass es unter der starken Exponentialzeithypothese für kein $\delta > 0$ *eine* $(1+\varepsilon)$*-Approximation für die Fréchetdistanz auf c-gepackten Kurven mit Laufzeit* $O((cn)^{1-\delta})$ *gibt. Dies gilt für jede polynomielle Einschränkung von* $1 \leq c \leq n$ *[Bri14b].*

Da wir diese Aussage für alle polynomiellen Einschränkungen beweisen, schließen wir beispielsweise eine 1.001-Approximation mit Laufzeit $O(c^2 + n)$ aus. Bezüglich der Abhängigkeit von ε zeigen wir eine bedingte untere Schranke in jeder Dimension $d \geq 5$. Diese weicht von der Laufzeit von Driemel et al. nur um einen Faktor $\sqrt{\varepsilon}$ ab.

Wir beweisen, dass es unter der starken Exponentialzeithypothese für kein $\delta > 0$ *eine* $(1+\varepsilon)$*-Approximation für die Fréchetdistanz auf c-gepackten Kurven in* $\mathbb{R}^d$*,* $d \geq 5$*, mit*

Laufzeit $O(\min\{cn/\sqrt{\varepsilon}, n^2\}^{1-\delta})$ *gibt. Dies gilt für genügend kleines* $\varepsilon > 0$ *und für jede polynomielle Einschränkung von* $1 \leq c \leq n$ *und* $\varepsilon \leq 1$ *[Bri14b].*

Dies lässt noch immer eine Lücke zwischen dem besten bekannten Algorithmus und unserer bedingten unteren Schranke. Wir schließen diese Lücke, indem wir einen schnelleren Algorithmus mit Laufzeit $O(cn\log^2(1/\varepsilon)/\sqrt{\varepsilon} + cn\log n)$ präsentieren [BK14]. Dieser Algorithmus hat eine optimale Abhängigkeit von n, c und ε (abgesehen von Faktoren geringerer Ordnung, unter der starken Exponentialzeithypothese und in hohen Dimensionen). Die Idee des verbesserten Algorithmus basiert auf Eigenschaften, die es unmöglich machen, bessere bedingte untere Schranken zu zeigen. Dies zeigt, dass bedingte untere Schranken nicht nur die Schwere von Problemen aufzeigen, sondern auch algorithmische Verbesserungen inspirieren können, indem sie lösbare Klassen von Instanzen nahelegen. Die optimale Laufzeit in den Dimensionen $d = 2, 3, 4$ zu bestimmen bleibt ein schweres offenes Problem.

Literaturverzeichnis

[AAKS13] Pankaj Agarwal, Rinat Ben Avraham, Haim Kaplan und Micha Sharir. Computing the Discrete Fréchet Distance in Subquadratic Time. In *Proc. 24th ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*, Seiten 156–167, 2013.

[AB10] Helmut Alt und Maike Buchin. Can we compute the similarity between surfaces? *Discrete & Computational Geometry*, 43(1):78–99, 2010.

[AG95] Helmut Alt und Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(1-2):78–99, 1995.

[AHPK+06] Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang und Carola Wenk. Fréchet distance for curves, revisited. In *Proc. 14th Annual European Symposium on Algorithms (ESA'06)*, Jgg. 4168 of *LNCS*, Seiten 52–63. Springer, 2006.

[AKW04] Helmut Alt, Christian Knauer und Carola Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, 2004.

[BBG+11] Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Maarten Löffler und Jun Luo. Detecting commuting patterns by clustering subtrajectories. *International Journal of Computational Geometry & Applications*, 21(3):253–282, 2011.

[BBK+07] Kevin Buchin, Maike Buchin, Christian Knauer, Günter Rote und Carola Wenk. How difficult is it to walk the dog? In *Proc. 23rd European Workshop on Computational Geometry (EWCG'07)*, Seiten 170–173, 2007.

[BBMM14] Kevin Buchin, Maike Buchin, Wouter Meulemans und Wolfgang Mulzer. Four Soviets Walk the Dog - with an Application to Alt's Conjecture. In *Proc. 25th ACM-SIAM Symposium on Discrete Algorithms (SODA'14)*, Seiten 1399–1413, 2014.

[BBW09] Kevin Buchin, Maike Buchin und Yusu Wang. Exact algorithms for partial curve matching via the Fréchet distance. In *Proc. 20th ACM-SIAM Symposium on Discrete Algorithms (SODA'09)*, Seiten 645–654, 2009.

[BF13] Karl Bringmann und Tobias Friedrich. Exact and efficient generation of geometric random variates and random graphs. In *Proc. 40th International Colloquium on Automata, Languages, and Programming (ICALP'13)*, Jgg. 7965 of *LNCS*, Seiten 267–278. Springer, 2013.

[BK14] Karl Bringmann und Marvin Künnemann. Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds, 2014. Eingereicht, http://arxiv.org/abs/1408.1340.

[BKP+14] Karl Bringmann, Fabian Kuhn, Konstantinos Panagiotou, Ueli Peter und Henning Thomas. Internal DLA: Efficient Simulation of a Physical Growth Model. In *Proc. 41th International Colloquium on Automata, Languages, and Programming (ICALP'14)*, Jgg. 8572 of *LNCS*, Seiten 247–258. Springer, 2014.

[BL13] Karl Bringmann und Kasper Green Larsen. Succinct Sampling from Discrete Distributions. In *Proc. 45th Annual ACM Symposium on Symposium on Theory of Computing (STOC'13)*, Seiten 775–782. ACM, 2013.

[BP12] Karl Bringmann und Konstantinos Panagiotou. Efficient Sampling Methods for Discrete Distributions. In *Proc. 39th International Colloquium on Automata, Languages, and Programming (ICALP'12)*, Jgg. 7391 of *LNCS*, Seiten 133–144. Springer, 2012.

[BPSW05] Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas und Carola Wenk. On map-matching vehicle tracking data. In *Proc. 31st International Conference on Very Large Data Bases (VLDB'05)*, Seiten 853–864, 2005.

[Bri14a] Karl Bringmann. Sampling from Discrete Distributions and Computing Fréchet Distances, 2014. Dissertation. http://scidok.sulb.uni-saarland.de/volltexte/2015/5988/.

[Bri14b] Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *Proc. 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS'14)*, Seiten 661–670, 2014.

[CCdVE+10] Erin Wolf Chambers, Éric Colin de Verdière, Jeff Erickson, Sylvain Lazard, Francis Lazarus und Shripad Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry*, 43(3):295–311, 2010.

[CDG+11] Daniel Chen, Anne Driemel, Leonidas J. Guibas, Andy Nguyen und Carola Wenk. Approximate Map Matching with respect to the Fréchet Distance. In *Proc. 13th Workshop on Algorithm Engineering and Experiments (ALENEX'11)*, Seiten 75–83, 2011.

[CDL+12] Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh und Magnus Wahlström. On Problems as Hard as CNF-SAT. In *Proc. 27th IEEE Conference on Computational Complexity (CCC'12)*, Seiten 74–84, 2012.

[DHP13] Anne Driemel und Sariel Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.

[DHPW12] Anne Driemel, Sariel Har-Peled und Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discrete & Computational Geometry*, 48(1):94–127, 2012.

[GO95] Anka Gajentaan und Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry: Theory and Applications*, 5(3):165–185, 1995.

[God91] Michael Godau. A natural metric for curves - computing the distance for polygonal chains and approximation algorithms. In *Proc. 8th Symposium on Theoretical Aspects of Computer Science (STACS'91)*, Jgg. 480 of *LNCS*, Seiten 127–136. Springer, 1991.

[GS13] Joachim Gudmundsson und Michiel Smid. Fréchet Queries in Geometric Trees. In *Proc. 21st Annual European Symposium on Algorithms (ESA'13)*, Jgg. 8125 of *LNCS*, Seiten 565–576. Springer, 2013.

[HPR11] Sariel Har-Peled und Benjamin Raichel. The Fréchet Distance Revisited and Extended. In *Proc. 27th Annual Symposium on Computational Geometry (SoCG'11)*, Seiten 448–457, 2011.

[IP01] Russell Impagliazzo und Ramamohan Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi und Francis Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[JLS12] David Jerison, Lionel Levine und Scott Sheffield. Logarithmic fluctuations for internal DLA. *Journal of the American Mathematical Society*, 25:271–301, 2012.

[MD86] Paul Meakin und J. M. Deutch. The formation of surfaces by diffusion limited annihilation. *The Journal of Chemical Physics*, 85:2320, 1986.

[MP99] Mario E. Munich und Pietro Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proc. 7th IEEE International Conference on Computer Vision*, Jgg. 1, Seiten 108–115, 1999.

[PPSZ05] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks und Francis Zane. An improved exponential-time algorithm for k-SAT. *Journal of the ACM*, 52(3):337–364, 2005.

[PW10] Mihai Pătraşcu und Ryan Williams. On the possibility of faster SAT algorithms. In *Proc. 21st ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, Seiten 1065–1075, 2010.

[RVW13] Liam Roditty und Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proc. 45th Annual ACM Symposium on Symposium on Theory of Computing (STOC'13)*, Seiten 515–524, 2013.

[Wal74] A. J. Walker. New Fast Method for Generating Discrete Random Numbers with Arbitrary Distributions. *Electronic Letters*, 10(8):127–128, 1974.

Karl Bringmann studierte sowohl Mathematik als auch Informatik an der Universität des Saarlandes und erhielt seine MSc-Abschlüsse im Jahr 2011. Von 2011 bis 2014 war er Doktorand am Max-Planck-Institut für Informatik, Saarbrücken, und an der Universität des Saarlandes. Seit September 2014 ist er Postdoktorand am Institut für Theoretische Informatik der ETH Zürich. Die zentralen Gebiete seiner Arbeit sind randomisierte Algorithmen, algorithmische Geometrie und untere Schranken. Während seiner akademischen Laufbahn wurde er durch ein Stipendium der Studienstiftung des deutschen Volkes, ein Google European Doctoral Fellowship, sowie ein ETH Zurich Postdoctoral Fellowship gefördert.