

Online-Optimierung der Rundreise auf der Kreislinie mit Informationsvorlauf

Matthias Tinkl

Angaben zur Veröffentlichung / Publication details:

Tinkl, Matthias. 2011. "Online-Optimierung der Rundreise auf der Kreislinie mit Informationsvorlauf." Augsburg: Universität Augsburg.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:

Deutsches Urheberrecht

Weitere Informationen finden Sie unter: / For more information see:

<https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/>



ONLINE-OPTIMIERUNG
DER
RUNDREISE
AUF DER
KREISLINIE
MIT
INFORMATIONSVORLAUF

Dissertation
zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Augsburg

vorgelegt von
Dipl.-Math. oec. Matthias Tinkl

Augsburg, Februar 2011

Erstgutachter: Prof. Dr. Karl Heinz Borgwardt
Zweitgutachter: Prof. Dr. Dieter Jungnickel

Tag der mündlichen Prüfung:

02. Februar 2011

Danksagung

Mein Dank gilt besonders Prof. Karl-Heinz Borgwardt für die Betreuung während der Bearbeitung. Sowie Prof. Dieter Jungnickel und Prof. Dirk Hachenberger, als auch dem Rest des Lehrstuhls „Diskrete Mathematik, Optimierung und Operations Research“ am mathematischen Institut der Universität Augsburg für die sehr interessante Zusammenarbeit und viele anregende Diskussionen.

Schließlich gilt mein Dank meinen Eltern, ohne deren tolle Unterstützung ich gar nicht erst so weit gekommen wäre.

Vorwort

Diese Arbeit ist aufgeteilt in drei Teile. Zuerst werden die Grundlagen der Online-Optimierung vorgestellt, darunter fallen die gängigsten Problemstellungen mit den für diese bisher erreichten kompetitiven Faktoren (der Quotient zwischen dem Wert der Lösung der Offline-Bearbeitung mit vollständiger Information und der Lösung des Online-Algorithmus mit unvollständiger Information), sowie den zur Berechnung des kompetitiven Faktors verwendeten Algorithmus zur Online-Bearbeitung.

Der zweite Teil stellt ein Konzept zur Verbesserung der Situation der Online-Bearbeitung durch die Erhöhung der Information für Entscheidungen vor und erläutert dessen Anwendungsmöglichkeiten.

Im dritten und letzten Teil schließlich soll diese Verbesserung an Hand einer Rundreise auf der Kreislinie genauer untersucht werden.

Im anschließenden Anhang finden sich noch Erläuterungen zu den für die Arbeit implementierten Programmen. Hierbei sei insbesondere auf die jeweiligen Parameter-Dateien hingewiesen, welche sofern vorhanden am Anfang des zum zugehörigen Programms gehörenden Abschnitts umfangreicher erörtert werden. Mit diesen Dateien können die Programme relativ einfach und unkompliziert für die jeweiligen Erfordernisse eingestellt werden.

Inhaltsverzeichnis

I. Grundlagen zur Online-Optimierung	1
1. Einführung in die Online-Optimierung	5
1.1. Grundsätzliches zur Online-Optimierung	6
1.2. Online Optimierung und Stochastik	9
1.2.1. Stochastische Optimierung	9
1.2.2. Stochastische Analyse	10
2. Online-Optimierungsprobleme	11
2.1. Alltägliche Optimierungsprobleme	11
2.2. Spezielle und größere Optimierungsprobleme	13
3. Einführung in die Kompetitive Analyse	19
3.1. Grundlagen	19
3.2. Techniken	23
3.2.1. Potenzial-Funktionen zum Finden einer Obergrenze	23
3.2.2. Sequenzen zum Finden einer Untergrenze	24
3.3. Entscheidungen mit Zufall	25
3.3.1. Algorithmen mit randomisierten Entscheidungen	25
3.3.2. Yao's Prinzip	26
4. Kompetitive Faktoren zu den Optimierungsproblemen	29
4.1. Zusätzliche kleine Hilfsmittel	29
4.2. Kaufentscheidung	30
4.2.1. Bahncard-Problem	30
4.2.2. Ski-Fahrer-Problem	35
4.3. Punktsuche	37
4.4. Pack-Problem	41
4.5. Verwaltung von Speicherseiten	44
4.6. Rundreisen	46
4.6.1. Rundreise auf der positiven Halbgeraden	46
4.6.2. Dial-A-Ride	52

II. Informationsvorlauf in der Online-Optimierung	59
5. Grundlegendes zum Informationsvorlauf	63
5.1. Einführung	63
5.2. Arten des Informationsvorlaufs	64
5.2.1. Informationsvorlauf als Zeiteinheit	64
5.2.2. Informationsvorlauf nach Reihenfolge	65
5.2.3. Informationsvorlauf nach einer Bearbeitungsgröße	66
5.3. Literaturstellen zum Informationsvorlauf und dessen Analyse	67
6. Verwendung des Informationsvorlaufs	69
6.1. Informationsvorlauf als Mittel zur Vorausschau	69
6.2. Informationsvorlauf als Mittel zur Auswahl	70
6.3. Kombination der beiden Verwendungsmöglichkeiten	73
III. Rundreise auf der Kreislinie	75
7. Einführung	79
7.1. Grundlegende Definitionen	79
7.2. Beispiel für eine Sequenz von Anfragen	80
7.3. Kompetitive Analyse	83
7.4. Modellierung als gemischtganzzahliges lineares Problem	91
7.5. Mögliche Erweiterungen für die Rundreise auf der Kreislinie	93
8. Gerichtete Rundreise auf der Kreislinie	95
8.1. Grundlagen zur gerichteten Rundreise auf der Kreislinie	96
8.2. Simulation einer gerichteten Rundreise auf der Kreislinie	98
8.3. Empirische Analyse	101
8.3.1. Durchschnittliche Schrittweite	104
8.3.2. Modifizierte durchschnittliche Schrittweite	106
8.3.3. Schrittweiten für einzelne Werte des Informationsvorlaufs	108
8.3.4. Schrittweiten der einzelnen Schritte unter Veränderung des Informationsvorlaufs	111
8.4. Erste Schrittweiten	113
8.4.1. Erste Schrittweite	114
8.4.2. Zweite Schrittweite	116
8.5. Allgemeine Schrittweite	119
8.5.1. Berechnung der Schrittweiten für einen Informationsvorlauf von zwei Anfragen	121
8.5.2. Formel zur allgemeinen Berechnung der Schrittweiten	131

9. Verwendung von Informationsvorlauf bei der Rundreise auf der Kreislinie	137
9.1. Algorithmus	137
9.2. Simulation einer Rundreise auf der Kreislinie mit Richtungswechsel . . .	138
9.3. Empirische Analyse	141
9.3.1. Durchschnittliche Schrittweite	141
9.3.2. Modifizierte durchschnittliche Schrittweite	143
9.3.3. Schrittweiten für einzelne Werte des Informationsvorlaufs	145
9.3.4. Schrittweiten der einzelnen Schritte unter Veränderung des Informationsvorlaufs	148
9.4. Erste Schrittweiten	149
9.4.1. Erste Schrittweite	149
9.4.2. Zweite Schrittweite	152
9.5. Überlegungen zur allgemeinen Schrittweite	163
10. Diskretisierung der Problemstellungen	175
10.1. Aufbau der diskretisierten Problemstellung	175
10.2. Erwartungswerte erste Schrittweite	177
10.2.1. Gerichtete Rundreise auf der Kreislinie	178
10.2.2. Normale Rundreise auf der Kreislinie	179
10.3. Allgemeine Schrittweite	181
10.3.1. Ein kurzes Beispiel zur Berechnung	185
10.3.2. Implementierung in Maple	188
10.3.3. Ergebnisse mit Maple	189
10.4. Fazit	192
11. Ausblick auf die Rundreise auf der Oberfläche einer Kugel	193
11.1. Problemstellung	193
11.2. Erzeugung von gleichverteilten Zufallspunkten auf der Oberfläche der Kugel	195
11.3. Durchführung einer Rundreise	198
 IV. Anhang	 203
A. Implementierung der Programme	205
A.1. Programm zur Simulation einer Rundreise auf der Kreislinie	205
A.1.1. Das Steuerprogramm	205
A.1.2. Das Paket tools	207
A.2. Programme zur Berechnung und Sammlung der empirischen Ergebnisse .	207
A.2.1. Das Programm zur Erzeugung der Sequenzen der Anfragen	208
A.2.2. Das Programm zur Lösung der Instanzen der Problemstellung mit verschiedenen Werten für den Informationsvorlauf	208

A.2.3. Das Programm zum Einsammeln der bei der Auswertung gewonnenen Daten	210
A.3. Bibliotheken zur diskreten Variante der Rundreise auf der Kreislinie . . .	212
A.3.1. Die Bibliothek zur Erzeugung der wesentlichen Bestandteile . . .	212
A.3.2. Die Bibliothek mit den Methoden zur Berechnung der erwarteten Schrittweiten	213
B. Empirische Daten	215
B.1. Daten für die gerichtete Rundreise auf der Kreislinie	215
B.2. Daten für die Rundreise auf der Kreislinie mit Richtungswechsel	223
Lebenslauf	239

Abbildungsverzeichnis

4.2.1. Verschiedene Verteilungen der Phasen des Bahncard-Problems	32
4.3.1. Beispiel für den Algorithmus zur Punktsuche	38
7.1.1. Koordinatensystem zur Rundreise auf der Kreislinie	79
7.2.1. Einfaches Beispiel für die Rundreise auf der Kreislinie	81
7.2.2. Route der Online-Bearbeitung	82
7.2.3. Route der Offline-Bearbeitung	82
7.3.1. Verschiedene Reiserouten auf der Kreislinie	83
8.2.1. Beispiel für die gerichtete Rundreise	99
8.3.1. Empirische Werte für die durchschnittliche Schrittweite der gerichteten Rundreise auf der Kreislinie bei unterschiedlichem Informationsvorlauf . .	105
8.3.2. Bereinigte empirische Werte für die durchschnittliche Schrittweite zur gerichteten Rundreise auf der Kreislinie	107
8.3.3. Die Schrittweiten der ersten Schritte für einen Informationsvorlauf von 2, 4, 29 und 69 bei der gerichteten Rundreise auf der Kreislinie	109
8.3.4. Die Schrittweiten der letzten Schritte für einen Informationsvorlauf von 2, 4, 29 und 69 bei der gerichteten Rundreise auf der Kreislinie	110
8.3.5. Schrittweiten der Schritte 1, 10 und 70, sowie 95 unter Variation des Informationsvorlaufs für die gerichtete Rundreise auf der Kreislinie	111
8.4.1. Empirische und berechnete erwartete Schrittweite für den ersten Schritt bei der gerichteten Rundreise auf der Kreislinie	116
8.4.2. Empirische und berechnete erwartete Schrittweite für den zweiten Schritt bei der gerichteten Rundreise auf der Kreislinie	118
8.5.1. Darstellung der Bearbeitungsreihenfolgen bis zum Schritt n für einen Informationsvorlauf von zwei Anfragen	123
8.5.2. Schematische Darstellung zur Struktur bei der rekursiven Berechnung der Schrittweite mit einem Informationsvorlauf von zwei Anfragen	125
9.2.1. Beispiel für eine Rundreise auf der Kreislinie mit Richtungswechsel	138
9.3.1. Empirische Werte für die durchschnittliche Schrittweite bei unterschiedlichem Informationsvorlauf zur Rundreise auf der Kreislinie mit Richtungswechsel	142
9.3.2. Bereinigte empirische Werte für die durchschnittliche Schrittweite zur Rundreise auf der Kreislinie mit Richtungswechsel	144

9.3.3.	Die Schrittweiten der ersten dreihundert Schritte für einen Informationsvorlauf von 4, 29, 69 und 129 bei der Rundreise auf der Kreislinie mit Richtungswechsel	146
9.3.4.	Die Schrittweiten der letzten dreihundert Schritte für einen Informationsvorlauf von 4, 29, 69 und 129 bei der Rundreise auf der Kreislinie mit Richtungswechsel	147
9.3.5.	Schrittweiten der Schritte 1, 10 sowie 70 unter Variation des Informationsvorlaufs für die Rundreise auf der Kreislinie mit Richtungswechsel	148
9.4.1.	Empirische und berechnete erwartete Schrittweite für den ersten Schritt der Rundreise auf der Kreislinie mit Richtungswechsel	151
9.4.2.	Zulässige Bereiche für die alten Anfragen bei der Rundreise auf der Kreislinie mit Richtungswechsel	152
9.4.3.	Dichtefunktionen für die möglichen Distanzen zum Reisepunkt einer verbleibenden Anfrage bei der Rundreise auf der Kreislinie mit Richtungswechsel nach dem ersten Schritt und einer Bewegung um weniger als einen viertel Kreis	153
9.4.4.	Dichtefunktionen für die möglichen Distanzen zu einer verbleibenden Anfrage nach dem ersten Schritt und einer Bewegung um mehr als einen viertel Kreis	154
9.4.5.	Empirische und berechnete erwartete Schrittweite für den zweiten Schritt bei der Rundreise auf der Kreislinie mit Richtungswechsel	164
9.5.1.	Distanz-Bereiche zu verbleibenden Anfragen bei der Rundreise auf der Kreislinie mit Richtungswechsel	166
10.3.1.	Ausschnitt des Graphen der Übergänge zu Beispiel 10.3.5	187
11.1.1.	Darstellung der Oberfläche der Kugel	194
11.1.2.	Berechnung der Distanz zweier Punkte	195
11.2.1.	Veranschaulichung der Verteilungen für zufällige Punkte auf der Oberfläche einer Kugel	196

Tabellenverzeichnis

7.2.1. Daten der Anfragen zu Beispiel 7.2.1	81
8.3.1. Standarddarstellung der Ergebnisse zur Regressionsanalyse	102
8.3.2. Ergebnis zur nichtlinearen Regression der durchschnittlichen Schrittweite für die gerichtete Rundreise auf der Kreislinie	104
8.3.3. Ergebnisse der verschiedenen Modelle zur durchschnittlichen Schrittweite für die gerichtete Rundreise auf der Kreislinie	106
8.3.4. Ergebnis zur nichtlinearen Regression der bereinigten durchschnittlichen Schrittweite für die gerichtete Rundreise auf der Kreislinie	107
8.3.5. Ergebnisse der verschiedenen Modelle für die bereinigten Daten zur gerichteten Rundreise auf der Kreislinie	108
8.4.1. Ergebnisse zur nichtlinearen Regression der ersten Schrittweite bei der gerichteten Rundreise auf der Kreislinie	115
8.4.2. Ergebnisse zur nichtlinearen Regression der zweiten Schrittweite bei der gerichteten Rundreise auf der Kreislinie	119
9.3.1. Ergebnisse zur nichtlinearen Regression der durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel	143
9.3.2. Ergebnisse der verschiedenen Modelle zur Rundreise auf der Kreislinie mit Richtungswechsel	143
9.3.3. Ergebnis zur nichtlinearen Regression der bereinigten durchschnittlichen Schrittweite für die Rundreise auf der Kreislinie mit Richtungswechsel	144
9.3.4. Ergebnisse der verschiedenen Modelle für die Rundreise auf der Kreislinie mit Richtungswechsel	145
9.4.1. Ergebnisse zur nichtlinearen Regression der ersten Schrittweite für die Rundreise auf der Kreislinie mit Richtungswechsel	152
9.4.2. Ergebnisse zur nichtlinearen Regression der zweiten Schrittweite für die Rundreise auf der Kreislinie mit Richtungswechsel	164
10.3.1. Mögliche Situationen der Entfernungen für die Reisepunkte der Anfragen aus dem Informationsvorlauf bei der gerichteten Rundreise	186
10.3.2. Werte für einen Informationsvorlauf von zwei Anfragen bei der diskreten gerichteten Rundreise auf der Kreislinie	190
10.3.3. Werte für einen Informationsvorlauf von drei Anfragen bei der diskreten gerichteten Rundreise auf der Kreislinie	190

10.3.4. Werte für einen Informationsvorlauf von vier Anfragen bei der diskreten gerichteten Rundreise auf der Kreislinie	190
10.3.5. Werte für einen Informationsvorlauf von zwei Anfragen bei der diskreten Rundreise auf der Kreislinie mit Richtungswechsel	191
10.3.6. Werte für einen Informationsvorlauf von drei Anfragen bei der diskreten Rundreise auf der Kreislinie mit Richtungswechsel	191
11.3.1. Ergebnisse der Regressionsanalyse für die betrachteten Sphären	201
B.1.1. Empirische Daten zur durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie	219
B.1.2. Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie	222
B.2.1. Empirische Daten zur durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel	226
B.2.2. Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel	230

Auflistung der Dateien

- 8.1. Beispielhafte Originalausgabe von R 103
- A.1. RundReise_Param.log 209
- A.2. Collect_Param.txt 211

Teil I.

Grundlagen zur Online-Optimierung

Übersicht

Dieser Teil besteht aus einer Zusammenfassung zur allgemeinen Online-Optimierung. Im Kapitel 1 wird ein grober Überblick zu umfassenderen Literaturstellen gegeben und anschließend die Grundbegriffe der normalen Online-Optimierung eingeführt und erklärt. Danach wird in Kapitel 2 eine Übersicht über die in der gängigen Fachliteratur verwendeten Optimierungsprobleme gegeben.

Das Kapitel 3 stellt im Anschluss daran die theoretischen Grundlagen für eine Methode zur Analyse der Online-Algorithmen vor und geht auch auf die dabei verwendeten grundlegenden Techniken ein. Abschnitt 3.3 bringt randomisierte Entscheidungen und erläutert das Zusammenspiel zwischen Aufgabensteller und Bearbeiter.

Im anschließenden Kapitel 4 wird zu den in Kapitel 2 vorgestellten Problemstellungen eine Übersicht auf die in der Literatur verwendeten Algorithmen und deren Abschneiden gegenüber der optimalen Offline-Lösung gegeben. Dabei wird bei mehreren vorhandenen Algorithmen für ein Optimierungsproblem nur derjenige mit dem besten Abschneiden oder einem besonderen Vorgehen vorgestellt.

1. Einführung in die Online-Optimierung

Die Betrachtung von Optimierungsproblemen mit zum Startzeitpunkt noch weitgehend unbekannten Daten wird in der heutigen Zeit immer wichtiger. Mit den unbekannten Daten sind hier die konkreten Ausprägungen und die Anzahl der zu betrachtenden Objekte gemeint. So kennt zum Beispiel eine Fahrstuhlsteuerung zu einem Zeitpunkt t nur die Anfragen der Personen, welche bis dahin in dem Fahrstuhl eingestiegen und noch nicht wieder ausgestiegen sind. Dagegen ist jedoch unbekannt, wie viele weitere Personen in nächster Zeit zusteigen wollen und in welche Stockwerke diese transportiert werden möchten.

Die Steuerung hat zudem zu dem Zeitpunkt, in welchem neue Personen den Knopf für das Herbeirufen des Fahrstuhls drücken, noch keine vollständige Information über deren Ziele. Das heißt, dass zwar grundsätzlich nach dem Betätigen des Rufknopfes die voraussichtliche Fahrtrichtung bekannt ist, aber noch nicht das genaue Stockwerk. Bei minimaler Ausstattung des Fahrstuhls liegt eventuell sogar noch nicht einmal die Information über die gewünschte Fahrtrichtung vor.

Erst nach dem Einsteigen der Wartenden, wenn der Fahrstuhl in deren Stockwerk Halt gemacht hat, und den folgenden Betätigungen der entsprechenden Knöpfe für deren Ziele, kann die Fahrstuhlsteuerung dann entscheiden, wie die neuen Fahrtziele eingebaut werden. Eine Analyse dieser Problemstellung findet sich zum Beispiel in [FR06], dort wird allerdings nicht nur das einfache Problem mit einem Fahrstuhl behandelt, sondern die verknüpfte Steuerung mehrerer Fahrstühle miteinander.

Da diese Art von Fragestellung nicht nur bei Aufzügen auftritt, sondern in vielen Feldern der Dienstleistungs- und Produktionsprozesse, entstand daraus im Laufe der Zeit ein eigener neuer Zweig der Optimierung, nämlich die Online-Optimierung.

Für einen ersten Eindruck über die Thematik sei auf folgende Literaturstellen verwiesen:

- [Bor92] ist eine erste kurze und prägnante Einführung von Borodin.
- [BE98], dies ist ein von Borodin und Yaniv später geschriebenes Buch, welches schon eine größere Übersicht auf das Thema bietet.

Neuere Zusammenfassungen finden sich zum Beispiel bei:

- [KR02], ein Vorlesungsskript von Rambau und Krumke an der Technischen Universität in Berlin.

- [Hag05], ein Skript von Hagerup zur Vorlesung Online-Algorithmen an der Universität Augsburg.
- [Kru02], die Habilitationsschrift von Krumke fasst die damaligen neusten Ergebnisse aus Berlin zusammen.

In diesen Schriften werden meist mehrere Problemstellungen der Online-Optimierung mit deren kompetitiven Kennzahlen, beziehungsweise die dazu bisher gefundenen Ober- und Unterschranken, vorgestellt. Neuere Methoden zum Vergleich für Algorithmen zur Online-Optimierung als die kompetitive Analyse finden sich unter anderem in folgenden Literaturstellen:

- [Hei05], ein Vergleich mit der erwarteten Performance.
- [BB01], hier wird als neue Maßzahl das Verhältnis der Kosten im „Worst-Case“ für Sequenzen von Anfragen gleicher Länge eingeführt.
- [Alb03], in dieser Arbeit von Albers werden zur Analyse auf eine eingeschränkte Menge aufbauende Sequenzen von Anfragen verwendet, etwas umfassender ist ein von ihr verfasstes Skript [Alb96], welches während eines Aufenthalts an der Universität von Aarhus im Rahmen einer Vorlesung über Online-Algorithmen entstanden ist.

1.1. Grundsätzliches zur Online-Optimierung

In diesem Abschnitt werden erst einmal die generellen Begriffe der Online-Optimierung vorgestellt.

Definition 1.1.1 (Online-Optimierung).

Bei der *Online-Optimierung* geht es um Optimierungsprobleme bei denen zu bestimmten Anfragen eine Entscheidung getroffen werden muss. Der Unterschied zur normalen Optimierung besteht hierbei darin, dass nicht von Anfang an alle Anfragen, zusammengefasst in einer Sequenz, bekannt sind, sondern diese erst im Laufe der Bearbeitung bekannt werden. Dabei müssen schon von Anfang an, trotz des nur unvollständigen Überblicks über die Sequenz, mit einer meist auch noch unbekannten Anzahl von Anfragen, Entscheidungen getroffen werden.

Zudem unterliegen die Anfragen in der Sequenz jeweils einem Ordnungsmerkmal M , welches über die Erfüllung gewisser Bedingungen, etwa dem Erreichen bestimmter Zeitpunkte, oder dem Fortschritt der Bearbeitung, regelt, wann die dazu gehörige Anfrage zur Bekanntgabe freigegeben wird.

Ein Algorithmus ist ein *Online-Algorithmus*, wenn er die Problemstellung einer Online-Optimierung so angeht, dass er immer für die schon bekannten Anfragen eine Möglichkeit zur Lösung zur Verfügung stellt und soweit nötig, auch schon verwirklicht. Diese

Lösung wird dann bei der Bekanntgabe neuer Anfragen entweder angepasst, oder es wird später für diese neuen Daten ein eigener Lösungsvorschlag erzeugt. Beide Möglichkeiten können auch in Variationen und kombiniert auftreten.

Die älteste Messlatte für die Qualität von Online-Optimierung ist die Qualität der Offline-Optimierung. Anpassungen und neuere Methodiken für Vergleichsmöglichkeiten zwischen Online-Algorithmen finden sich unter anderem in den oben angegebenen Literaturstellen.

Definition 1.1.2 (Offline-Optimierung).

Ein *Offline-Problem* ist eine Problemstellung mit vollständiger Kenntnis über alle zu bearbeitenden Anfragen.

Anders als der Online-Algorithmus, hat die *Offline-Bearbeitung* also zur Lösung der Problemstellung von Anfang an das Wissen über die komplette Sequenz der Anfragen für die Planung zur Verfügung und kann somit eine optimale Lösung des Problems unter Einbeziehung aller Informationen berechnen. Meist ergibt sich das Vorgehen der Offline-Bearbeitung dabei entweder durch ein sogenanntes Orakel, welches eine minimale Lösung liefert, oder durch eine vollständige Enumeration aller infrage kommender Lösungen.

Dabei dürfen die Anfragen jedoch erst dann bearbeitet werden, wenn das zugehörige Ordnungsmerkmal erfüllt ist, ansonsten ist ein Vergleich zwischen einem Online-Algorithmus und der Offline-Bearbeitung zwecklos.

Allgemein ausgedrückt bedeutet dies, dass die Offline-Bearbeitung Anfragen erst dann bearbeiten darf, wenn auch der Online-Algorithmus davon Kenntnis haben könnte und diese Kenntnis zur Bearbeitung der Anfrage nutzen kann. Da das Konzept des Ordnungsmerkmals wegen der vielfältigen Möglichkeiten an dieser Stelle ziemlich abstrakt aufgebaut ist, sei auf zwei mögliche beispielhafte Umsetzungen des Ordnungsmerkmals und deren Verwendung hingewiesen:

- Bei Problemstellungen, welche innerhalb eines zeitlichen Rahmens eine Sortierung der Anfragen nach zeitlichen Gesichtspunkten verlangen, darf auf jeden Fall keine Anfrage vor dem Zeitpunkt ihrer Bekanntgabe erledigt werden.
- Bei Problemstellungen, in welchen jede Anfrage sofort bearbeitet werden muss und somit nur die Reihenfolge der Bekanntgabe der Anfragen eine Rolle spielt, darf keine der Anfragen vor ihrem Vorgänger bearbeitet werden, dies wäre sonst ein signifikanter Verstoß gegen die Prämisse der sofortigen Bearbeitung.

Ein bei der Online-Bearbeitung auftretendes Problem ist zum Beispiel folgendes:

Es seien bei einem Traveling-Salesman-Problem am Anfang ein Startpunkt und k weitere Reisepunkte bekannt. Damit ist aber nicht garantiert, dass nicht noch in nächster Zeit, oder auch irgendwann viel später, zusätzliche Reisepunkte auftreten können. Ebenso ist die genaue Anzahl der Reisepunkte die insgesamt auftreten werden unbekannt. Das Hinzufügen neuer Punkte kann sich dabei durchaus auch rückwirkend auf

die Güte früherer Lösungen auswirken, dies kann beispielsweise anhand von Beispiel 7.2.1 nachvollzogen werden.

Ohne eine explizite Information darüber, dass die gerade bekanntgegebene Anfrage die letzte Anfragen aus der Sequenz ist, ist die Bestimmung des Endes einer Sequenz generell nicht machbar. Daher besteht für einen Online-Algorithmus bei der Bearbeitung eines Online-Problems nicht die Möglichkeit auf die Bekanntgabe aller Anfragen zu warten. Aus diesem Grund wird der Aufbau eines Online-Algorithmus grundsätzlich folgendermaßen gehandhabt:

Algorithmus 1.1.3 (Grundsätzliches Vorgehen bei der Online-Optimierung).

Das generelle Vorgehen bei der Optimierung eines Online-Problems gliedert sich wie folgt:

- (i) Lösung des Problems mit den bisher unbearbeiteten bekannten Anfragen und Start der Bearbeitung dieses Lösungsvorschlags.*
- (ii) Bei der Bekanntgabe von neuen Anfragen wird entweder direkt auf diese reagiert, indem die aktuelle Bearbeitung unterbrochen wird und ein neuer Lösungsvorschlag gesucht wird, oder sie werden erst einmal ignoriert.*
- (iii) Wenn ein vorher gefundener Lösungsvorschlag komplett abgearbeitet wurde und noch nicht bearbeitete Daten vorliegen, wird wieder Schritt (i) gestartet. Ansonsten wird solange gewartet, bis neue Anfragen ankommen, und erst danach wieder mit Schritt (i) weitergemacht.*

Innerhalb des in Algorithmus 1.1.3 beschriebenen Vorgehens gilt es einige Eigenheiten der Online-Bearbeitung zu beachten:

Bemerkung 1.1.4 (Eigenheiten der Online-Optimierung).

Die Eigenheiten der Online-Optimierung sind:

- (i) Die Vereinigung von separaten Lösungen für einzelne Teilprobleme muss nicht automatisch zu einer optimalen Lösung führen.
- (ii) Die Gesamtzahl der zufälligen Anfragen mit ihren Ausprägungen ist meist a priori nicht bekannt.

Teil (i) der Bemerkung 1.1.4 ist dafür verantwortlich, dass die Online-Bearbeitung fast immer schlechtere Ergebnisse liefert, da sie im Normalfall nur mit Ausschnitten des gesamten Problems arbeiten kann und somit nicht den meist notwendigen Gesamtüberblick hat.

Teil (ii) über die Unkenntnis des Zeitpunktes der Beendigung der Bekanntgabe neuer Daten kann theoretisch dazu führen, dass die Bearbeitung mit einem Online-Algorithmus endlos weitergeht, auch wenn die Offline-Bearbeitung wegen der Kenntnis des Endes zu einem früheren Zeitpunkt schon abgeschlossen wurde.

Um zum Beispiel diese endlose Bearbeitung bei einer Rundreise zu umgehen, kann bei deren Bearbeitung durch einen Online-Algorithmus innerhalb von Algorithmus 1.1.3 jede Lösung für die bekannten, aber noch nicht bearbeiteten, Anfragen als eine eigene Rundreise mit dem Startpunkt als Endpunkt ausgearbeitet werden.

Dies bedeutet dann, dass beim Warten auf weitere neu Anfragen als aktueller Standort wieder der Startpunkt dient und die bis dahin gefahrene Rundreise als abgeschlossen angesehen wird. Die Zeitmessung der Rundreise wird folglich nur bis zur Abarbeitung der letzten Anfrage und der danach folgenden Rückfahrt zum Startpunkt gemessen. Eine Wartezeit kann sich somit nur dann ergeben, wenn nach der Rückkehr zum Startpunkt noch eine weitere, zum Zeitpunkt der Rückkehr noch unbekannte, Anfrage bearbeitet werden muss. Das bedeutet im Umkehrschluss aber auch, dass bei einem Warten an einem anderen Punkt als den Startpunkt diese Wartezeit immer mit zählt.

Das Zurückfahren zum Startpunkt kann jedoch bei bestimmten Fällen (zum Beispiel immer dann, wenn der letzte angefahrene Punkt näher an den neuen Punkten liegt als der Startpunkt) durchaus auch in einer Verschlechterung des gesamten Ergebnisses resultieren.

1.2. Online Optimierung und Stochastik

In diesem Abschnitt soll die stochastische Optimierung kurz vorgestellt und auf die mögliche Verwendung der Stochastik für die Online-Optimierung hingewiesen werden.

1.2.1. Stochastische Optimierung

Die stochastische Optimierung verwendet, anders als die klassische Optimierung nicht nur einzelne Instanzen eines Optimierungsproblems, sondern betrachtet mehrere mögliche Szenarien mit deren einzelnen Eintrittswahrscheinlichkeiten. Unter Einbeziehung dieser Eintrittswahrscheinlichkeiten sollen dann bestimmte Kriterien (zum Beispiel Erwartungswerte) optimal gestaltet werden.

Dabei wird das zugrunde liegende Problem nicht für jedes einzelne Szenario allein optimiert, sondern die Gesamtheit aller Szenarien erfasst. Konkret bedeutet dies, dass in die Modellierung der Problemstellung alle möglichen Szenarien eingebaut werden, und danach angestrebt wird, eine gewünschte Zielsetzung über alle Szenarien zu optimieren.

Das bedeutet insbesondere, dass bei einem Modell in welchem eine direkte Reaktion auf bestimmte später eintretende Ereignisse nicht vorgesehen ist, es nicht für jedes einzelne Szenario eigene Variablen gibt, sondern nur Variablen die unabhängig von dem dann konkret eintretenden Szenario sind. Damit muss die Belegung der Variablen so ausgewählt werden, dass egal was passiert die gewünschten Anforderungen erfüllt werden.

Eine der gewünschten Zielsetzungen kann beispielsweise die Optimierung des Erwartungswertes sein oder etwa die Optimierung des Ergebnisses des schlechtesten Szenarios bei Verwendung der gefundenen Belegung der Variablen.

Das führt dazu, dass bei der stochastischen Optimierung entweder erst einmal vor der eigentlichen Analyse über mögliche Szenarien und deren Ausprägungen, sowie eventuell deren Eintrittswahrscheinlichkeiten, nachgedacht werden muss, oder es wird eine schon vorher festgelegte Menge von Szenarien, – diese kann zum Beispiel durch den Auftraggeber vorgegeben sein, – verwendet.

Insgesamt gesehen beschäftigt sich die stochastische Optimierung mit der Optimierung einer vorgegebenen Problemstellung unter der Einbeziehung einer Wahrscheinlichkeitsverteilung für die möglichen Szenarien. Das heißt die vorgegebene Wahrscheinlichkeitsverteilung geht direkt in die Entscheidungen ein. Insbesondere ist deshalb das Ergebnis der Bearbeitung einer Instanz der Problemstellung davon abhängig, welches der möglichen Szenarien dann tatsächlich eintritt.

1.2.2. Stochastische Analyse

Innerhalb der Online-Optimierung gibt es einige Bereiche in welchen eine Anwendung der Stochastik Sinn macht. So kann zum Beispiel mithilfe der Stochastik das Verhalten und die Qualität (Average-Case-Analyse) eines Algorithmus unter der Annahme einer bestimmten Verteilung für die einzelnen Anfragen der gesamten Sequenz untersucht werden. Innerhalb dieser Untersuchung wird nun nicht das Abschneiden des Algorithmus für eine einzige Instanz, sei es jetzt im besten oder schlechtesten Fall, betrachtet, sondern das sich aus dem Erwartungswert über die zufällige Verteilung ergebene durchschnittliche Abschneiden.

Insbesondere gilt: Wenn der Algorithmus grundsätzlich wie in Definition 3.3.1 mit randomisierten Entscheidungen sein Verhalten beeinflusst, so muss dieser definitiv mithilfe einer stochastischen Analyse untersucht werden.

Insgesamt ergibt sich durch die Analyse mit dem Average-Case eine eher stimmige Aussage über das Verhalten des Algorithmus. Dabei baut diese Aussage nicht nur auf einem Extremum, sei es nun in optimaler oder schlechtester Ausprägung, auf, sondern diese wurde über die gesamte Problemstellung, allerdings unter Verwendung einer Annahme über die Wahrscheinlichkeitsverteilung der Anfragen, getroffen.

Die stochastische Analyse beschäftigt sich also mit dem Abschneiden eines Algorithmus unter der Annahme, dass die möglichen Instanzen der zugehörigen Problemstellung nach einer vorgegebenen Wahrscheinlichkeitsverteilung auftreten. Anders als bei der stochastischen Optimierung wird hierbei im Normalfall die Verteilung der Instanzen nicht bei der Durchführung des Algorithmus berücksichtigt, das heißt, dass das Ergebnis der Bearbeitung einer Instanz der Problemstellung bei der Anwendung eines deterministischen Algorithmus eindeutig feststeht.

Die Analysen in den beiden Kapiteln 8 und 9 bestehen aus einer stochastischen Auswertung des zugehörigen Algorithmus über eine bestimmte vorgegebene Zufallsverteilung der Anfragen in der Sequenz.

2. Online-Optimierungsprobleme

Dieses Kapitel bietet einen Überblick über verschiedene Arten von Online-Optimierungsproblemen. Dazu wird zuerst in Abschnitt 2.1 auf alltägliche einfache Optimierungsprobleme eingegangen und in Abschnitt 2.2 werden schließlich weitere speziellere und auch komplexere Optimierungsprobleme angesprochen.

2.1. Alltägliche Optimierungsprobleme

Ein Optimierungsproblem, welches jedem immer wieder begegnet, ist die Entscheidung ab welchem Zeitpunkt oder bei welchen Gegebenheiten es sinnvoll ist, einen bestimmten Gegenstand oder auch eine bestimmte Leistung zu kaufen, statt immer wieder auszuleihen oder weiterhin zu mieten .

Optimierungsproblem 2.1.1 (Kaufentscheidung).

Bei dem Problem der *Kaufentscheidung* stellt sich für den Interessenten die Frage, entweder für viel Geld einen bestimmten Gegenstand gleich zu kaufen, oder diesen erst einmal für weniger Geld nur zu mieten und eventuell dann doch später noch zu kaufen. Vor seiner Entscheidung muss der Interessent abwägen, ob er den Gegenstand überhaupt kaufen will und wenn ja unter welchen Umständen, wie zum Beispiel ab einer vorher festgelegten Anzahl an Mietzahlungen.

Eine weitere Variante ist die Entscheidung über den Einkauf einer bestimmten Möglichkeit (zum Beispiel eine Kundenkarte oder ähnliches) zur Verbesserung der Konditionen für den Kauf von Dienstleistungen.

Online bedeutet hierbei, dass die Anzahl der Tage, oder auch Stunden, an welchen der Gegenstand oder die Leistung benötigt wird und/oder deren Verteilung im vorgegebenen Zeitrahmen (dieser muss nicht unbedingt endlich sein) unbekannt ist.

Anwendungen von Optimierungsproblem 2.1.1 sind zum Beispiel die Entscheidung für den Kauf einer Bahncard wie in [KR02, Abschnitt 3.2], beziehungsweise das Abwägen zwischen dem Kaufen und oder dem Mieten von Ski wie in [KR02, Unterabschnitt 2.2.1] und [Hag05, Kapitel 3].

Eine ähnlich schöne einfache Problemstellung ist die Suche nach einem bestimmten Objekt, ohne dessen Standort genau zu kennen. Allerdings wird die Betrachtung der Suche immer komplexer, je mehr Freiheit dem Suchenden bei der Auswahl der Suchwege gelassen wird.

Optimierungsproblem 2.1.2 (Punktsuche).

Das Problem der *Punktsuche* liegt darin, in einem vorgegeben Raum einen bestimmten vordefinierten Punkt mit unbekannten Koordinaten zu finden.

Krumke und Rambau verwenden in [KR02, Kapitel 2] als Beispiel für eine Punktsuche die Suche nach einem geparkten Auto innerhalb einer Straße. Eine allgemeinere einführende Behandlung dieser Thematik findet sich in dem Artikel von Baeza-Yates, Culberson und Rawlins [BCR93].

Ein weiteres oft anzutreffendes Problem im Alltag ist das Packen von Reisetaschen oder Einkaufstaschen, bei welchen auch immer wieder zusätzliche Gegenstände berücksichtigt werden müssen, ohne dass der Inhalt einer Tasche nachträglich verändert werden kann.

Optimierungsproblem 2.1.3 (Packen).

Beim *Packen* geht es darum, Gegenstände in möglichst wenige Kisten zu verpacken. Bei der Online-Variante müssen die Gegenstände direkt nach der Reihenfolge, in welcher sie ankommen, verpackt werden. Ein Umräumen eines schon verpackten Gegenstandes in eine andere Kiste ist meist nicht erlaubt.

Diese Aufgabenstellung kann mit verschiedenen Zielsetzungen betrachtet werden. Das einfache Packen („Bin Packing“) findet sich in [KR02, Kapitel 2] als Übungsaufgabe mit dem Ziel der Minimierung der Anzahl der verwendeten Kisten. Eine dazu gehörende Problemstellung ist ein erweitertes Packen („Color Bin Packing“) aus [Kru02, Kapitel 8] oder auch [KdPRS01], welches die Gegenstände in Gruppen unterteilt und die Kisten möglichst homogen füllen will. Eine Anwendung dazu findet sich zum Beispiel in [KR02, Kapitel 13].

Ein Optimierungsproblem, welchem die meisten Menschen unbewusst bei der Arbeit am Computer begegnen, ist die in der Software-Programmierung wichtige effektive Verwaltung von Hauptspeicher.

Optimierungsproblem 2.1.4 (Verwaltung von Speicherseiten).

Unter einer *Verwaltung von Speicherseiten*, auch „Paging“ genannt, wird die Allokation von Speicherseiten in einem begrenzten Hauptspeicher mit Kapazität K verstanden. Dabei existieren grundsätzlich zwei Arten von Speicher:

- der Hauptspeicher mit einer schnellen Zugriffszeit.
- der normale Speicher mit wesentlich langsamerer Zugriffszeit.

Zur Vereinfachung wird angenommen, dass alle Speicherseiten die gleiche Größe haben.

Jede Anfrage bezeichnet nun eine für sich notwendige Seite des Speichers. Wenn diese Seite zum Zeitpunkt der Anfrage im Hauptspeicher geladen ist, so kann die Anfrage ohne Aufwand bearbeitet werden. Sollte diese Seite allerdings nicht im Hauptspeicher geladen sein, so muss diese aus dem langsamen Speicher nachgeladen werden und eine der aktuell im Hauptspeicher befindlichen Seiten aus diesem gelöscht werden. Ziel der Aufgabenstellung ist es, die Anzahl der Nachlade-Operationen zu minimieren.

Behandelt wird diese Problemstellung unter graphentheoretischen Aspekten in [Alb99, Kapitel 2], sowie in [KR02, Kapitel 4] als auch in [Hag05, Kapitel 7].

2.2. Spezielle und größere Optimierungsprobleme

Um einen tieferen Eindruck über die Anwendbarkeit und Relevanz der Online-Optimierung zu erhalten, werden in diesem Abschnitt verschiedene Beispiele für komplexere Problemstellungen der Online-Optimierung dargestellt.

Optimierungsproblem 2.2.1 (Rundreise-Problem).

In einem *Rundreise-Problem*, auch bekannt als „*travelling repairman problem*“ (TRP) oder „*travelling salesman problem*“ (TSP), ist eine Menge von nicht notwendigerweise disjunkten Punkten P innerhalb eines metrischen Raums gegeben. Oft verwendete metrische Räume sind:

- die positive Halbgerade $\mathbb{R}^+$,
- die Zahlengerade $\mathbb{R}$,
- der gesamte $\mathbb{R}^n$,
- ein Graph mit gewichteten Kanten (das Gewicht einer Kante gibt die Entfernung zwischen den beiden inzidenten Knoten an),
- als abstrakte Grundlage ein allgemeiner metrischer Raum.

Zusätzlich zu den normalen Punkten in P gibt es noch einen besonderen Punkt 0. Die dabei verwendete Aufgabenstellung ist es, eine derartige Route im vorgegebenen metrischen Raum zu finden, so dass jeder der Punkte aus P mindestens einmal besucht wird. Der Punkt 0 soll dabei sowohl Start- als auch Endpunkt der Route sein.

Es sei hier aber auch gleich die Anmerkung angebracht, dass eine Rundreise in der Literatur nicht immer notwendigerweise so definiert wird, dass wieder zum Startpunkt zurückgekehrt werden muss. Diese kann auch so definiert werden, dass allein das Besuchen aller Punkte aus P schon ausreichend ist. In diesem besonderen Fall ist nach dem Besuch des letzten Punktes nicht mehr die Rückfahrt zum Startpunkt vorgesehen, sondern es wird bei diesem Punkt verweilt. Als weitere zusätzliche Variante gibt es zudem noch die Möglichkeit zwar einen Endpunkt festzulegen, für diesen aber nicht den Startpunkt zu verwenden.

Die Zielsetzungen der beiden angegebenen speziellen Problemarten bei der Aufstellung der Route sind:

- Bei einem „*travelling salesman problem*“ wird grundsätzlich während der Planung der Rundreise mehr drauf geachtet, dass die insgesamt verstreichende Zeit bis zur

endgültigen Rückkehr zum Ausgangspunkt, also dem Abschluss der Route, minimiert wird. Dabei wird weniger berücksichtigt, wann eine Anfrage bearbeitet wird, sondern nur darauf geachtet, dass diese bearbeitet wird. Die Einsatzplanung eines Vertreters mit Haustürgeschäften ist ein typisches Beispiel für diese Zielsetzung.

- Beim „travelling repairman problem“ dagegen tritt genau dieser Aspekt mehr in den Vordergrund. Hierbei wird nicht nur auf die Minimierung der gesamten Zeit geachtet, sondern darauf die Summe der Wartezeiten der einzelnen Anfragen bis zur deren Bearbeitung zu minimieren. Die Wartezeit entspricht dabei der zeitlichen Differenz zwischen der Bekanntgabe der Anfrage und deren Bearbeitung. Ein Standardbeispiel für diese Art der Problemstellung ist die Planung des Tagesablaufs eines Wartungstechnikers ohne dessen Einsatz das defekte Gerät nicht mehr genutzt werden kann.

Bei der Online-Behandlung dieser Problemstellungen werden die einzelnen Punkte erst im Laufe der Zeit bekannt und deren Anzahl ist nicht vorher festgelegt.

Die Rundreise-Problemstellung findet sich zum Beispiel bei der Betrachtung des Tagesablaufs eines Servicemanns wie in [KdPPS03] oder der Arbeitsabläufe der „Gelben Engel“ (ADAC-Straßenwacht) wie in [KRT03]. In seiner Habilitationsschrift [Kru02, Kapitel 3] betrachtet Krumke insbesondere die Fragestellungen auf der Zahlengeraden und auf deren positiver Halbgeraden.

Bemerkung 2.2.2 (Mögliche Erweiterungen von Optimierungsproblem 2.2.1).

Das Optimierungsproblem 2.2.1 kann zu weiteren Optimierungsproblemen erweitert werden:

- Beim „traveling-repairman-problem“ ist es möglich, die jeweiligen Wartezeiten der Anfragen mit jeweils individuellen Gewichten zu versehen.
- Das „*latency online dial-a-ride problem*“ ist eine Erweiterung bei welcher nicht nur einzelnen Punkte zu besuchen sind, sondern stattdessen von den einzelnen Punkten Objekte zu anderen Punkten transportiert werden müssen. Dabei kann entweder immer nur ein Objekt auf einmal oder es können bis zu C Objekte mitgenommen werden.
- Das k -Server Problem erweitert auf mehrere Reisende (hier dann Server genannt), welche jeweils getrennt voneinander verschiedene Routen bearbeiten.

Anwendungen zu dial-a-ride finden sich in [AKR00] und [HKR00], sowie in [KR02, Kapitel 9], als auch in [Kru02, Kapitel 4] mit mehreren verschiedenen Zielsetzungen insbesondere zur maximalen und durchschnittlichen Wartezeit. In [Kru02, Kapitel 5] wird das Optimierungsproblem 2.2.1 mit allgemeinen metrischen Räumen behandelt. Eine Behandlung des k -Server Problems findet sich sowohl in [Alb96, Kapitel 4] als auch in [KR02, Kapitel 8]. Betrachtungen zum Traveling-Repairman-Problem finden sich zum Beispiel in [KdPPS03].

Korollar 2.2.3.

Das Optimierungsproblem 2.1.4 ist ein Spezialfall des k -Server Problems.

Beweis zu Korollar 2.2.3.

Aus dem Optimierungsproblem 2.1.4 ergibt sich das k -Server Problem, wenn mit k Servern gearbeitet wird und der metrische Raum so definiert ist, dass zwischen allen Paaren von Punkten immer eine Distanz von 1 gegeben ist. Dann entspricht jede Speicherseite aus Optimierungsproblem 2.1.4 einem der Punkte im metrischen Raum und die aktuelle Belegung des Hauptspeichers ist gegeben durch die Standorte der k Server. $\square$

Wenn im Optimierungsproblem 2.2.1 statt auf dem ganzen metrischen Raum nur auf bestimmten Punkten und dazu passenden vorgegebenen Verbindungen gearbeitet werden darf, ergeben sich die Transport-Probleme.

Optimierungsproblem 2.2.4 (Transport-Problem).

Anders als beim Optimierungsproblem 2.2.1 ist beim *Transport-Problem* das Netzwerk zum Transportieren der Güter schon fest vorgegeben. Es kann also nicht mehr ein beliebiger Weg zwischen zwei Punkten verwendet werden, sondern um von einem Punkt zu einem anderen zu gelangen, müssen fest vorgegebene Verbindungen benutzt werden.

Wie in der Erweiterung „dial-a-ride“ vom Optimierungsproblem 2.2.1 sind hier nicht mehr nur einfache Punkte zum Anfahren gegeben, sondern immer Aufträge zum Transport von einem Punkt zu einem anderen Punkt. Dabei können bei einer Kapazität von $C \geq 1$ falls möglich auch mehrere Aufträge auf einmal bearbeitet werden. Die bei der Online-Betrachtung nicht festgelegten Daten sind die Aufträge zum Transport beziehungsweise im Extremfall auch noch jeweils der Zielpunkt zu welchem das Gut transportiert werden muss.

Beispiele für das Optimierungsproblem 2.2.4 sind die Verwaltung eines Lagers und die Steuerung von Logistik-Aufträgen wie in [GKR01]. Eine weitere aus dem Bereich der Informatik kommende Problemklasse ist die folgende Problemstellung:

Optimierungsproblem 2.2.5 (Listen-Aktualisierung).

In der *Listen-Aktualisierung* geht es darum, eine unsortierte Liste von Werten so zu verwalten, dass der Aufwand bei Zugriffen für eine Folge von in der Liste enthaltenen Werten minimal wird. Die Kosten eines Zugriffs auf ein Element entsprechen dabei der Position dieses Elements in der Liste.

Dabei sind die folgendem zusätzlichen Aktionen des Verwalters der Liste erlaubt:

- (i) Die kostenlose Positionierung des aktuell nachgefragten Elements auf eine niedrigere und damit frühere Position als die aktuelle innerhalb der Liste.
- (ii) Der Tausch zweier benachbarter Elemente mit Kosten 1.

Bemerkung 2.2.6 (Einfügen und Löschen innerhalb der Liste).

Das Einfügen und Löschen von Werten in die Liste kann durch Erweiterungen der Aktionen bei Zugriffen behandelt werden:

- Beim Löschen erfolgt ein normaler Zugriff mit dessen Kosten und danach wird der Wert falls er vorhanden ist gelöscht.
- Um festzustellen, ob ein einzufügender Wert tatsächlich neu oder schon in der Liste enthalten ist, wird zuerst ein Zugriff auf diesen Wert ausgeführt. Wenn dieser Zugriff fehlschlägt, wird der neue Wert fiktiv am Ende der Liste mit gesamten Kosten $1 + |Liste|$ angehängt und ein Zugriff mit den davon ausgelösten Aktionen darauf simuliert und die daraus resultierenden Änderungen durchgeführt.

Da die Liste spätestens nach einer größeren Anzahl von Aktionen keiner vorher gegebenen Sortierung mehr genügt, sind die Auswirkungen eines Einfügens eines neuen Elements an einer bestimmten fest vorgegebenen Position in der Liste nicht direkt absehbar. Als Konsequenz daraus müsste dafür eine weitere Aktion vorgegeben werden, welche dann insbesondere auch Bezug auf die aktuelle Sortierung der Liste nehmen müsste. Diese aktuelle Sortierung der Liste ist jedoch nur noch durch die Wirkungen der Aktionen des Verwalters gegeben, mit der oben angegebenen Simulation des Zugriffs wird gewährleistet, dass die neue Anfrage nach diesen Regeln einsortiert wird.

Aus diesen Gründen wird eine gezielte Einfügung eines Wertes an einer bestimmten Stelle als unzulässig betrachtet.

In [Alb96, Kapitel 5] und in [KR02, Abschnitt 3.1] sowie [Hag05, Kapitel 4] wird auf diese Problemstellung eingegangen.

Eine in der Telekommunikation oft auftretende Aufgabenstellung ist das Verteilen von Bandbreiten innerhalb eines Netzwerks auf die einzelnen Anfragen.

Optimierungsproblem 2.2.7 (Netzwerk Routing).

Beim *Netzwerk Routing* („virtual circuit routing“) geht es zum Beispiel um Anfragen zur Datenübertragung zwischen je zwei Punkten mit einer gewissen Systemlast oder auch Bandbreite. Aufgabe des Verwalters ist es innerhalb der vorhandenen Kapazitäten zu bleiben und dabei für jede Anfrage wenn möglich eine Verbindung zwischen den zwei gewünschten Punkten zu finden und diese Verbindung mit der Anfrage zu belegen. Dabei zu beachtende mögliche Zielsetzungen sind zum Beispiel:

- Die Minimierung der maximalen Last der vorhandenen Kanten im Netzwerk ohne Ablehnung von Anfragen (Die Kapazität des Netzwerks sei dabei groß genug gewählt.).
- Die Maximierung der Summe der Bandbreiten, genannt der *Durchsatz* der angenommenen Anfragen unter Einhaltung vorgegebener Kapazitäten. Diese Zielsetzung wird in der Literatur auch als „admission control“ bezeichnet.

Die Online noch unbekannten Daten sind in diesem Fall die Anfragen mit ihren Daten für den Transport.

Krumke und Rambau behandeln das Optimierungsproblem 2.2.7 ausführlich in [KR02, Kapitel 6].

Eine in der Produktion von Gütern auftretende Problemstellung ist die Erstellung von Reihenfolgen zur Bearbeitung verschiedener Güter auf einer Maschine.

Optimierungsproblem 2.2.8 (Metrische Tasksysteme).

Ein metrisches Tasksystem besteht aus einer Maschine mit einer endlichen Zustandsmenge M . Diese Maschine soll nun eine Folge von Aufträgen in der ankommenden Reihenfolge erledigen. Jeder dieser Aufträge verursacht abhängig vom momentanen Zustand der Maschine einen bestimmten Aufwand.

Zielsetzung der Optimierung ist es nun, den gesamten entstehenden Aufwand zu minimieren. Um dieser Zielsetzung entsprechen zu können, kann die Maschine vor der Bearbeitung eines Auftrags ihren eigenen Zustand wechseln, allerdings führt ein Wechsel zwischen zwei Zuständen jeweils zu einem von den beiden Zuständen abhängigen bestimmten zusätzlichen Aufwand, welcher in den Gesamtaufwand eingerechnet werden muss.

Der unbekannte Aspekt bei der Online-Optimierung ist hier die Folge der Aufträge.

Diese Problemstellung ist sehr allgemein und wird in [KR02, Kapitel 7] relativ knapp vorgestellt.

Eine größere Übersicht über die Algorithmen zur Online-Optimierung bietet zum Beispiel das Buch [FW98] von Fiat und Woeginger, eine kritische Auseinandersetzung bietet der Artikel [Alb03] von Albers.

3. Einführung in die Kompetitive Analyse

Um innerhalb einer bestimmten Problemklasse die Ergebnisse eines Online-Algorithmus im Vergleich zu den Ergebnissen eines Offline-Algorithmus, also bei Vorlage der vollständigen Informationen von Anfang an, betrachten zu können, wird die kompetitive Analyse¹ verwendet.

Die kompetitive Analyse betrachtet, über alle Instanzen einer Problemstellung hinweg, für jede dieser Instanzen den Quotienten zwischen dem Wert der Lösung des zu beurteilenden Algorithmus und dem Wert der Lösung der Offline-Bearbeitung. Der Artikel [Bor92] von Borodin bietet eine kurze und prägnante Einführung in die Problemstellung der kompetitiven Analyse, eine größere Übersicht bietet das später von Borodin zusammen mit Yaniv geschriebene Buch [BE98].

Der erste Teil dieses Kapitels geht auf die Grundlagen der kompetitiven Analyse ein. Im zweiten Abschnitt werden grundlegende Techniken zur Berechnung des kompetitiven Faktors vorgestellt. Danach kommt noch ein kurzer Einblick in die Bearbeitung von Online-Problemen mit zufälligen Algorithmen.

3.1. Grundlagen

Für die weiteren Betrachtungen innerhalb dieses Abschnitts sei als Ziel der Optimierung immer die Minimierung einer Zielfunktion vorgegeben.

Definition 3.1.1 (Kompetitive Analyse).

Die *kompetitive Analyse* bestimmt ein Maß für die Güte der Lösungen aller Instanzen einer Problemstellung durch den Online-Algorithmus gegenüber der jeweiligen Lösung der gleichen Instanz durch die Offline-Bearbeitung. Dabei wird als Maß für eine Problemstellung, bei einer zu minimierenden Zielfunktion, der größte Quotient für eine der Instanzen der Problemstellung zwischen dem errechneten Wert des Online-Algorithmus (Alg) und dem dagegen optimalen Wert der Offline-Bearbeitung (Off) verwendet.

Der *kompetitive Faktor* c ergibt sich als die kleinste Zahl, für die folgende Eigenschaft über alle Instanzen I einer Problemstellung mit Minimierung einer Zielfunktion gilt:

$$\text{Kosten}_{Alg}(I) \leq c \cdot \text{Kosten}_{Off}(I) + \alpha$$

¹Nach [Gro95] wurde die kompetitive Analyse 1985 zum ersten Mal in [ST85] von Sleator und Tarjan eingeführt.

Dabei ist α eine konstante nicht negative Zahl. Wenn für einen Algorithmus $\alpha = 0$ gilt, so heißt dieser *strikt c -kompetitiv* und c ist der *strikt kompetitive Faktor*. Falls $\alpha > 0$ gilt, so ist der Algorithmus nur *c -kompetitiv*.

Da der Online-Algorithmus ebenfalls zur Lösung des Offline-Problems verwendet werden kann, gilt bei einer Minimierung:

$$\text{Kosten}_{\text{Off}}(I) \leq \text{Kosten}_{\text{Alg}}(I).$$

Somit gilt grundsätzlich immer $c \geq 1$

Definition 3.1.2 (Optimaler Online-Algorithmus).

Ein Online-Algorithmus wird als ein *optimaler Online-Algorithmus* bezeichnet, wenn dessen kompetitiver Faktor c unter allen anderen Algorithmen seiner Art nachweisbar der kleinste mögliche kompetitive Faktor ist. Das heißt es muss nachgewiesen werden, dass es für jeden anderen Algorithmus eine Sequenz gibt, für welche dieser minimal den kompetitiven Faktor c erreichen kann.

Bemerkung 3.1.3 (Verbot der frühzeitigen Abarbeitung).

Das in Definition 1.1.2 getroffene Verbot der frühzeitigen Abarbeitung von Anfragen σ_i , – das heißt, dass Anfragen erst dann bearbeitet werden dürfen, wenn das in der Sequenz dazu mitgegebene Ordnungsmerkmal erfüllt ist, – sorgt für eine gerechtere Angleichung der beiden gegensätzlichen Positionen. Ohne diese Einschränkung könnte der kompetitive Faktor so ziemlich jedes Online-Algorithmus gegen unendlich geschickt werden. Bei einer Rundreise bestünde zum Beispiel die ganze Sequenz der Anfragen nur aus einer Anfrage, deren Bekanntgabe zum Zeitpunkt $t \rightarrow \infty$, also in äußerst ferner Zukunft, erfolgt. Da die Offline-Bearbeitung die Anfrage von Anfang an kennt, könnte sie diese sofort bearbeiten, der Online-Algorithmus hingegen müsste bis zu deren Bekanntgabe warten und würde somit unendlich lange brauchen.

Allerdings ergeben sich durch den Vergleich mithilfe der kompetitiven Analyse zwischen dem Online-Algorithmus und der Offline-Bearbeitung trotzdem gewisse Schwierigkeiten bei der Beurteilung eines Algorithmus.

Bemerkung 3.1.4 (Problematik der kompetitiven Analyse).

Aus der Definition der kompetitiven Analyse nach Definition 3.1.1 ergeben sich folgende Schwachstellen:

- Die Betrachtung über alle Instanzen der Problemstellung hinweg (unendlich viele), dabei enthalten sind natürlich auch Instanzen mit unendlich vielen Anfragen, führt meist zu einer ziemlich abstrakten Betrachtungsweise. Dies liegt vor allem daran, dass auch äußerst seltene Randfälle, egal wie unwahrscheinlich deren Auftreten ist, voll berücksichtigt werden müssen.
- Bei vollständiger Information für die Offline-Bearbeitung hat der Online-Algorithmus teilweise wegen seiner unvollständigen Information kaum eine Möglichkeit für

eine wirklich gute kompetitive Kennzahl. Um diesen Effekt etwas abzuschwächen, wird die Information der Offline-Bearbeitung manchmal eingeschränkt.

- Die Festlegung der Kennzahl auf den „Worst-Case“ bedeutet, dass schon eine große Abweichung von der besten Lösung bei einer einzigen Instanz, und sei diese auch noch so unwahrscheinlich, zu einer wesentlich schlechteren Kennzahl führt. Dabei wird allerdings überhaupt nicht berücksichtigt, wie gut der Algorithmus auf den restlichen Instanzen der Problemstellung arbeitet.

Eine Möglichkeit das Problem mit dem manchmal massiven Informationsvorsprung und dessen Ausnutzung durch die Offline-Bearbeitung anzugehen, ist eine Einschränkung der Verwendbarkeit dieser Informationen. In [Kru02, Kapitel 3] oder auch [BKdPS00] wird zum Beispiel dazu die so genannte faire Offline-Bearbeitung definiert:

Definition 3.1.5 (Faire Offline-Bearbeitung).

Eine *faire Offline-Bearbeitung* unterliegt in ihren Möglichkeiten gewissen Beschränkungen, welche verhindern sollen, dass diese die Kenntnis der zusätzlichen Information in einer Weise nutzt, welche dem Online-Algorithmus nicht zur Verfügung stehen.

So gibt es bei einem Problem einer Rundreise (siehe Optimierungsproblem 2.2.1) in der Ebene zum Beispiel nach [Kru02] und [BKdPS00] die Einschränkung, dass sich die faire Offline-Bearbeitung zu keinem Zeitpunkt t außerhalb der konvexen Hülle der Reisepunkte befinden darf, welche bis dahin bekannt gegeben worden sind. Die faire Offline-Bearbeitung darf also nicht, schon im Vorgriff auf einen späteren Reisepunkt, das durch die konvexe Hülle der bereits bekannten Reisepunkte definierte Gebiet verlassen. Allerdings kann diese immer noch die frühe Kenntnis der zukünftigen Reisepunkte ausnutzen und vor deren eigentlicher Bekanntgabe, also vor deren Erscheinen, innerhalb des über die konvexe Hülle definierten Gebietes zu einer günstiger gelegenen Position fahren.

Da der Online-Algorithmus eigentlich, außer in den trivialen Fällen immer mit unvollständigen Wissen über die Sequenz der Anfragen auskommen muss und einmal getroffene Entscheidungen entweder gar nicht oder nur bedingt geändert werden können, gibt es verschiedene Ansätze zur grundsätzlichen Vorgehensweise bei der Abarbeitung der eingehenden und noch nicht bearbeiteten Anfragen.

Definition 3.1.6 (Verschiedene Komponenten von Verhaltensstrategien für Online-Algorithmen).

Für einen Online-Algorithmus gibt es folgende unterschiedlichen Strategien für sein Vorgehen während der Bearbeitung einer Sequenz von Anfragen.

Replan: Bei Anwendung der *Replan*-Strategie reagiert der Algorithmus auf jede neue Anfrage unmittelbar, indem die neue Information verarbeitet wird und für alle noch nicht bearbeiteten Anfragen ein neuer Vorschlag zur Bearbeitung berechnet wird.

Ignore: Das Gegenteil der Replan-Strategie ist die *Ignore*-Strategie, diese verwendet nacheinander Lösungsvorschläge für Teilsequenzen der gesamten Sequenz von Anfragen. Je nach Situation werden bei dieser Strategie neue Anfragen anders behandelt.

Solange der im Moment aktuelle Vorschlag, zur Bearbeitung einer Teilsequenz, noch nicht abgeschlossen worden ist, werden alle neuen Anfragen grundsätzlich nicht beachtet.

Eine erneute Berechnung eines Vorschlags zur Bearbeitung der zurückgestellten Anfragen findet erst dann wieder statt, wenn die Durchführung des aktuellen Vorschlags zur Bearbeitung endgültig beendet ist. Der Ausdruck Vorschlag zur Bearbeitung bezieht sich dabei auf eine durch den Algorithmus für eine Teilmenge der Anfragen ausgearbeitete Lösungsmöglichkeit.

Warten: Da nicht alle Anfragen sofort bekannt werden, sondern sich erst im Zeitablauf ergeben, kann es immer vorkommen, dass die Online-Bearbeitung einen Zustand verlässt, obwohl dieser als Ausgangslage für die Bearbeitung der demnächst bekannt werdenden Anfragen äußerst günstig gewesen wäre.

Um die Auswirkungen solcher Wechsel des Zustands abzuschwächen wurde die Strategie des *Wartens* eingeführt. Dabei besteht kein Zwang mehr sofort nach der Bearbeitung der aktuell ausgewählten Anfrage zur Bearbeitung der nächsten ausgewählten Anfrage überzugehen, sondern es können einige Zeiteinheiten im aktuellen Zustand abgewartet werden. Damit ist also eine sofortige Fortführung der Bearbeitung nicht mehr zwingend notwendig. Eine Anwendung dieser Strategie ist verbunden mit der Hoffnung, durch das Warten in dem jetzigen Zustand vielleicht für in nächster Zeit bekannt werdende Anfragen eine bessere Ausgangslage zu behalten.

Diese Strategie kann so, beim späteren Eintreffen von geeigneten Anfragen, selbst dann einen Sinn machen, wenn das Warten, etwa wie bei der Aufgabenstellung einer Minimierung der gesamten Bearbeitungszeit, an und für sich eigentlich kontraproduktiv ist.

Allerdings muss bei der Anwendung dieser Strategie darauf geachtet werden, dass nicht zu lange oder sogar endlos in einem Zustand verblieben wird. Weil bei der Online-Bearbeitung im Normalfall auch nicht bekannt ist, wie viele Anfragen insgesamt bearbeitet werden müssen, kann eine zu lange Wartezeit, insbesondere wenn keine geeigneten Anfragen eintreffen, auch dazu führen, dass jede Vergleichbarkeit mit den erreichten Werten der Offline-Bearbeitung verloren geht.

Natürlich sind auch Kombinationen und Varianten der einzelnen Strategien zugelassen.

3.2. Techniken

Ein kompetitiver Faktor soll durch den Vergleich der Kosten des Online-Algorithmus mit denen der Offline-Bearbeitung auf allen Instanzen einer Problemstellung, wie die in Kapitel 2 vorgestellten Optimierungsprobleme, gewonnen werden. Da der direkte, enumerative Beweis allerdings meist zu einer unendlichen Beweisfolge führen würde, wird nun mit verschiedenen Techniken versucht, den dazu eigentlich notwendigen Aufwand zu reduzieren.

Grundsätzlich gibt es dafür mehrere verschiedene Methoden, in den folgenden Abschnitten soll eine Übersicht über einen Teil dieser Methoden gegeben werden.

Die einfachste Methode ist es wie in Abschnitt 7.3 den Kosten-Quotienten zwischen der Lösung des Online-Algorithmus und der Lösung der Offline-Bearbeitung aufzustellen und nach oben abzuschätzen. Diese Methode ist allerdings nicht sonderlich elegant, sondern eher nur grobes Handwerkszeug, da dabei sowohl Nenner als auch Zähler meist isoliert betrachtet werden und deren Zusammenhang, gemeint ist hier die Auswirkungen der Sequenzen von Anfragen auf die Kosten beider Algorithmen, unbeachtet bleibt.

3.2.1. Potenzial-Funktionen zum Finden einer Oberschranke

Eine wichtige Technik, um Oberschranken für den kompetitiven Faktor eines Online-Algorithmus zu erhalten, sind die Potenzial-Funktionen, welche mit der amortisierten Analyse verbunden sind. Explizit definiert werden diese zum Beispiel in [Alb96, Abschnitt 3.1] und in [KR02, Kapitel 3].

Definition 3.2.1 (Potenzial-Funktionen).

Eine *Potenzial-Funktion* Φ ordnet der aktuellen Situation des Online-Algorithmus eine nicht negative Zahl Φ_i zu. Diese Zahl soll im Idealfall das Potenzial des aktuellen Zustandes, zum Beispiel beim Optimierungsproblem 2.2.5 die vorliegende Sortierung der Liste, des Online-Algorithmus gegenüber dem Zustand der Offline-Bearbeitung widerspiegeln. Aus der Summe der Änderung des Wertes der Potenzial-Funktion nach einer Aktion für eine Anfrage σ_i und den tatsächlich auftretenden Kosten des Algorithmus o_i lassen sich die *amortisierten Kosten* a_i berechnen:

$$a_i = o_i + \Phi_i - \Phi_{i-1} \quad (3.2.1)$$

Wird nach der Aktion das Potenzial besser ($\Phi_i - \Phi_{i-1} \geq 0$), so überschätzt a_i die echten Kosten, bei einer Verschlechterung werden die Kosten unterschätzt.

Die tatsächlichen Gesamtkosten ergeben sich dann zu:

$$\sum_{i=1}^n o_i = \sum_{i=1}^n [a_i - (\Phi_i - \Phi_{i-1})] = \sum_{i=1}^n a_i + (\Phi_0 - \Phi_n) \leq \sum_{i=1}^n a_i + \Phi_0$$

Können nun die folgenden Aussagen

$$a_i \leq c \cdot \text{Off}(\sigma_i) \quad \forall i = 1, \dots, n \quad (3.2.2a)$$

$$\Phi_0 \text{ ist unabhängig von der gesamten Sequenz } \sigma \quad (3.2.2b)$$

für alle möglichen Anfragen σ_i und aktuelle Zustände des Online-Algorithmus nachgewiesen werden, so ist damit ein kompetitiver Faktor von c bewiesen. Mit der zusätzlichen Anforderung $\Phi_0 = 0$ ist der Algorithmus sogar strikt c -kompetitiv.

Bemerkung 3.2.2 (Einschränkung in der Anwendbarkeit der Potenzial-Funktion aus Definition 3.2.1).

Die Betrachtungsweise mit Potenzialfunktionen ist nur dann anwendbar, wenn jede Anfrage in dem Moment bearbeitet wird, in welchem sie bekannt wird. Das heißt die Anfrage σ_{i+1} wird dem Algorithmus erst nach der Festlegung und normalerweise auch erst nach der Ausführung seiner Aktion für σ_i übergeben. Insbesondere müssen dabei Entscheidungen zur Bearbeitung einer Anfrage unabhängig von den nachfolgenden Anfragen sein, dürfen also nicht wieder geändert werden.

3.2.2. Sequenzen zum Finden einer Unterschranke

Um erste Eindrücke vom kompetitiven Faktor eines Online-Algorithmus zu erhalten werden oft erst einmal verschiedene Instanzen der Problemstellung erzeugt und für diese Instanzen der Quotient zwischen den vom Online-Algorithmus benötigten Kosten und den Kosten der Offline-Bearbeitung berechnet.

Manchmal ergeben sich bei diesen ersten Versuchen schon gewisse Eindrücke davon wie eine Sequenz aufgebaut sein muss um den Aufwand der Online-Bearbeitung im Vergleich zu einer besseren Offline-Bearbeitung in die Höhe zu treiben. Aus jeder dieser Instanzen der Problemstellung lässt sich aber auch eine erste Schätzung für eine Unterschranke des kompetitiven Faktors herleiten. Da dieser nämlich für alle Instanzen eines Optimierungsproblems gelten muss, kann er nicht kleiner sein als der Quotient zwischen den Online- und den Offline-Kosten einer beliebigen Instanz der Problemstellung. Aus der Findung einer Sequenz mit „Worst-Case“-Verhalten lässt sich folgendes Konzept zur Bestimmung des kompetitiven Faktors herleiten:

Definition 3.2.3 (Böswilliger Gegner).

Unter dem Konzept *Böswilliger Gegner* wird die Erzeugung einer derartigen Sequenz σ zur Eingabe verstanden, so dass das Ergebnis von deren Bearbeitung für den Online-Algorithmus immer so schlecht wie möglich ist.

Dieses Konzept findet sich unter anderem in [KR02, Abschnitt 2.2].

3.3. Entscheidungen mit Zufall

Da bei der Bearbeitung eines Online-Optimierungsproblems mit einem deterministischen Algorithmus in der kompetitiven Analyse zum Beispiel durch die Anwendung von Definition 3.2.3 immer genau die Schwachstellen dieses Algorithmus zum Erhöhen des kompetitiven Faktors ausgenutzt werden, stellt sich die Frage, wie die Auswirkungen dieses Vorgehens abgeschwächt werden können.

Das Finden der möglichst schlechten Sequenzen von Anfragen ist durch die genau festgelegte Vorgehensweise eines Online-Algorithmus begünstigt, und daher ist ein möglicher Ansatzpunkt dieses festgelegte Verhalten zu ändern. Diese Veränderung kann dadurch erreicht werden, dass in die Algorithmen ein randomisiertes Element eingebaut wird.

Der nächste Unterabschnitt definiert nun diese Algorithmen ganz allgemein und in dem darauf folgenden wird mithilfe eines Prinzips aus der Spieltheorie eine erste Unterschranke für den kompetitiven Faktor für Online-Algorithmen mit zufälligen Entscheidungen hergeleitet.

3.3.1. Algorithmen mit randomisierten Entscheidungen

Um das Durchschauen eines Algorithmus für einen Gegenspieler (hier der Erzeuger der Anfrage-Sequenzen) schwieriger zu gestalten, wird bei vorher festgelegten Entscheidungen eine zufällige Komponente eingebaut, dadurch entstehen die randomisierten Algorithmen.

Definition 3.3.1 (Randomisierte Algorithmen).

Ein Algorithmus ist ein *randomisierter Algorithmus*, wenn zur Laufzeit des Algorithmus oder vor dessen Bearbeitung der ersten Anfrage dessen Verhalten bei der Eingabe einer Sequenz von Anfragen nicht eindeutig berechenbar ist.

Dieser Einfluss des Zufalls kann sich bei jeder einzelnen Entscheidung auswirken oder er legt nur vor der Bearbeitung der ersten Anfrage das weitere Verhalten gegenüber der restlichen Sequenz fest. Dabei ist eine anfängliche zufällige Entscheidung zwischen mehreren Online-Algorithmen ausdrücklich eingeschlossen.

Algorithmen ohne den Einfluss von randomisierten Entscheidungen werden *deterministische Algorithmen* genannt.

Bemerkung 3.3.2.

Innerhalb dieser Arbeit geht es, sofern nicht speziell auf den Anteil des Zufalls verwiesen wird, immer nur um deterministische Algorithmen.

Da nun bei den Algorithmen nach Definition 3.3.1 durch deren spezielle Konstruktion das Ergebnis für die Bearbeitung einer Menge von Anfragen vor dem Feststehen des Ausgangs der randomisierten Entscheidungen nicht mehr exakt bestimmt werden kann, muss die Festlegung des kompetitiven Faktors aus Definition 3.1.1 dementsprechend angepasst werden.

Definition 3.3.3 (Kompetitiver Faktor für randomisierte Algorithmen).

Um bei der Analyse der Algorithmen aus Definition 3.3.1 die randomisierten Entscheidungen berücksichtigen zu können, wird die Definition des *kompetitiven Faktors* um den Erwartungswert $\mathbb{E}$ über die Wahrscheinlichkeitsverteilung P des Online-Algorithmus erweitert:

$$\mathbb{E}_p[\text{Kosten}_{Alg}(I)] \leq c \cdot \mathbb{E}_p[\text{Kosten}_{Off}(I)] + \alpha = c \cdot \text{Kosten}_{Off}(I) + \alpha$$

Alles weitere aus der Definition 3.1.1 lässt sich ohne zusätzliche Veränderung übernehmen.

Durch die Einwirkung des Zufalls ergibt sich nun allerdings auch eine veränderte Situation beim möglichen Wissen des Gegners, welcher die Sequenz der Anfrage erzeugt und die Offline-Bearbeitung durchführt. Dieses Wissen muss nun genauer spezifiziert werden.

Dazu gibt es in den meisten Literaturstellen drei Hauptarten von Gegnern welche hier nur kurz vorgestellt werden sollen.

Definition 3.3.4 (Gegner bei randomisierten Algorithmen).

In [KR02, Abschnitt 2.3] und in [Alb96, Abschnitt 2.1] finden sich folgende Konzepte für die verschiedenen Ausprägungen von Gegnern für randomisierte Algorithmen.

Blind: Der *blinde Gegner* („Oblivious Adversary“) muss die Sequenz der Anfragen vor Beginn der Bearbeitung komplett aufgestellt haben und kann daher nicht auf Erkenntnisse aus dem Verhalten des Online-Algorithmus zugreifen.

Online Adaptiv: Der *online adaptive Gegner* („Adaptive Online Adversary“) darf den Online-Algorithmus beobachten und kann aufbauend auf dessen Verhalten neue Anfragen erzeugen. Allerdings muss dieser Gegner ebenfalls die Sequenz von Anfragen online bearbeiten, und zwar ohne Kenntnis der momentanen sowie der zukünftigen Entscheidungen des Online-Algorithmus.

Offline Adaptiv: Der *offline adaptive Gegner* („Adaptive Offline Adversary“) entspricht dem online adaptivem Gegner, außer dass er die gesamte Sequenz der Anfragen offline bearbeiten darf.

3.3.2. Yao's Prinzip

Mit Yao's Prinzip wird die Anpassung einer Erkenntnis aus der Spieltheorie zu Zwei-Personen-Nullsummenspielen mit gemischten Strategien in die Online-Optimierung beschrieben. Da eine umfassende Einbeziehung von theoretischen Aussagen und Definitionen der Spieltheorie innerhalb der Thematik dieser Arbeit nicht angebracht ist, sei hier auf [Bor01, Kapitel 30 und 31] für einen größeren Überblick über die theoretischen Grundlagen zu Zwei-Personen-Nullsummenspielen verwiesen.

Wichtig für die in diesem Unterabschnitt vorzustellende Technik ist nur die Tatsache, dass bei Zwei-Personen-Nullsummenspielen mit gemischten Strategien die Spielwerte der beiden Spieler bei Verwendung von MinMax-Strategien übereinstimmen. Konkret ergibt sich als mathematische Formel:

Lemma 3.3.5 (Spielwert für MinMax bei Zwei-Personen-Nullsummenspielen mit gemischten Strategien).

Nach [Bor01, Satz 30.8] gilt für ein Zwei-Personen-Nullsummenspiel mit gemischten Strategien und deren Wahrscheinlichkeitsvektoren p und q sowie der Auszahlungsmatrix M :

$$\max_p \min_i p^T M e_i = \min_q \max_j e_j^T M q$$

Daraus leitet sich dann nach [KR02, Abschnitt 5.2] das *Prinzip von Yao* durch eine Abschätzung der linken Seite nach unten ab:

Lemma 3.3.6 (Yao's Prinzip).

$$p^T \min_i M e_i \leq \min_q \max_j e_j^T M q$$

Angewandt auf die Online-Optimierung mit zufälligen Algorithmen ergibt sich nach [Alb96, Abschnitt 3.2] und [KR02, Abschnitt 5.2] folgende Aussage:

Satz 3.3.7 (Yao's Prinzip für Online-Optimierung).

Falls eine Wahrscheinlichkeitsverteilung P für mehrere Sequenzen von Anfragen σ^i eines Online-Optimierungsproblems existiert, so dass für alle deterministischen Algorithmen folgende Abschätzung gilt,

$$\mathbb{E}_p[\text{Kosten}_{\text{Alg}}(\sigma^i)] \geq \delta \cdot \mathbb{E}_p[\text{Kosten}_{\text{Off}}(\sigma^i)]$$

dann gilt $c \geq \delta$ für alle kompetitiven Faktoren c , welche von zufälligen Algorithmen gegen den blinden Gegner erreicht werden können.

Beweis zu Satz 3.3.7.

Sei $\{\sigma^x \mid x \in \mathcal{X}\}$ die Menge aller möglichen Sequenzen von Anfragen und $\{\text{Alg}_y \mid y \in \mathcal{Y}\}$ die Menge aller deterministischen Algorithmen. Um nun den Satz 3.3.7 zu beweisen, muss nur gezeigt werden, dass für jede Wahrscheinlichkeitsverteilung Y über die Menge der deterministischen Algorithmen (analog ist X eine Wahrscheinlichkeitsverteilung über die Menge der Sequenzen von Anfragen) der kompetitive Faktor δ für mindestens eine Sequenz gilt. Dazu soll nachgewiesen werden, dass

$$\inf_Y \sup_{x \in \mathcal{X}} \{\mathbb{E}[\text{Kosten}_{\text{Alg}_y}(\sigma^x)] - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x)\}$$

größer gleich Null ist.

$$\begin{aligned}
& \inf_Y \sup_{x \in \mathcal{X}} \left\{ \mathbb{E}_Y[\text{Kosten}_{Alg_y}(\sigma^x)] - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x) \right\} \\
&= \inf_Y \sup_{x \in \mathcal{X}} \left\{ \int_{\mathcal{Y}} \text{Kosten}_{Alg_y}(\sigma^x) - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x) dY \right\} \\
&= \inf_Y \sup_X \left\{ \int_{\mathcal{X}} \int_{\mathcal{Y}} \text{Kosten}_{Alg_y}(\sigma^x) - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x) dY dX \right\} \\
&\geq \sup_X \inf_Y \left\{ \int_{\mathcal{X}} \int_{\mathcal{Y}} \text{Kosten}_{Alg_y}(\sigma^x) - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x) dY dX \right\} \quad (3.3.1)
\end{aligned}$$

Mit Fubini ergibt sich aus (3.3.1):

$$\begin{aligned}
& \sup_X \inf_Y \left\{ \int_{\mathcal{Y}} \int_{\mathcal{X}} \text{Kosten}_{Alg_y}(\sigma^x) - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x) dX dY \right\} \\
&= \sup_X \inf_{y \in \mathcal{Y}} \left\{ \int_{\mathcal{X}} \text{Kosten}_{Alg_y}(\sigma^x) - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x) dX \right\} \\
&= \sup_X \left\{ \inf_{y \in \mathcal{Y}} \left\{ \mathbb{E}_X[\text{Kosten}_{Alg_y}(\sigma^x)] dX \right\} - \delta \cdot \text{Kosten}_{\text{Off}}(\sigma^x) \right\} \geq 0 \quad (3.3.2)
\end{aligned}$$

Die letzte Ungleichung aus (3.3.2) mit dem größer gleich Null ergibt sich aus der in der Voraussetzung des Satzes genannten Abschätzung für alle deterministischen Algorithmen welche die Wahrscheinlichkeitsverteilung P für die Sequenzen der Anfragen erfüllen muss.

Die Umwandlungen der Extremaloperatoren wie $\inf_{y \in \mathcal{Y}} \leftrightarrow \inf_Y \int_{\mathcal{Y}}$ ergibt sich aus der Erweiterung von der Grundmenge in deren konvexen Hülle und der umgekehrten Reduktion dieser konvexen Hülle wieder auf die Grundmenge zurück. $\square$

4. Kompetitive Faktoren zu den Optimierungsproblemen

Dieses Kapitel führt zu einigen der Optimierungsprobleme aus Kapitel 2 deterministische Algorithmen mit den besten kompetitiven Faktoren ein und behandelt, sofern möglich, den Nachweis der Optimalität der Algorithmen. Dabei ist jeweils die verwendete Literatur angegeben, welche für die Definition des Algorithmus und den Beweis der Optimalität als Quelle gedient hat.

4.1. Zusätzliche kleine Hilfsmittel

Um die Beweise in den anderen Abschnitten nicht zu sehr zu überfrachten, werden die mathematischen Hilfsaussagen, welche nicht direkt mit der Online-Optimierung zu tun haben, in diesen Abschnitt gesammelt.

Korollar 4.1.1 (Quotienten-Abschätzung mit gleichen Anteilen).

Für alle Zahlen a, b, x mit $x \geq 0$ und $b > 0$ gilt unter der Voraussetzung $a \geq b$:

$$\frac{a+x}{b+x} \leq \frac{a}{b} \quad (4.1.1)$$

Mit der Voraussetzung $a \leq b$ ergibt sich die analoge Aussage:

$$\frac{a+x}{b+x} \geq \frac{a}{b}$$

Beweis zu Korollar 4.1.1.

Die Voraussetzungen $a \geq b$ und $x \geq 0$ ergeben:

$$x \cdot a \geq x \cdot b \quad (4.1.2)$$

Nach der Addition von $a \cdot b$ zu beiden Seiten von (4.1.2) ergibt sich:

$$x \cdot a + a \cdot b \geq x \cdot b + a \cdot b \iff a \cdot (x+b) \geq b \cdot (x+a) \quad (4.1.3)$$

Da sowohl $b > 0$ und damit auch $x+b > 0$ ist, kann (4.1.3) durch die beiden Ausdrücke geteilt werden und dies liefert das gewünschte Ergebnis von (4.1.1) $\square$

Korollar 4.1.2 (Quotienten-Abschätzung mit gleichen Anteilen und verschiedenen Gewichten).

Für alle Zahlen a, δ, x, z mit $a + \delta \cdot x > 0$ und $\delta \in [0, 1]$ gilt unter der Voraussetzung $z \geq x$:

$$\frac{a + x}{a + \delta \cdot x} \leq \frac{a + z}{a + \delta \cdot z} \quad (4.1.4)$$

Beweis zu Korollar 4.1.2.

Mit $\delta \leq 1$ gilt

$$a \cdot \delta \leq a \quad (4.1.5)$$

Wird nun zu beiden Seiten von (4.1.5) der Ausdruck $z - x \geq 0$ multipliziert, so ergibt sich:

$$a \cdot \delta \cdot (z - x) \leq a \cdot (z - x) \iff a \cdot \delta \cdot z - a \cdot \delta \cdot x \leq a \cdot z - a \cdot x \quad (4.1.6)$$

Die Addition von a^2 und $\delta \cdot x \cdot z$ zu beiden Seiten der Ungleichung aus (4.1.6), sowie eine so gestaltete Umstrukturierung, dass beide Seiten nur Summanden beinhalten, liefert folgende Ungleichung:

$$a^2 + a \cdot \delta \cdot z + a \cdot x + \delta \cdot x \cdot z \leq a^2 + a \cdot \delta \cdot x + a \cdot z + \delta \cdot x \cdot z \quad (4.1.7)$$

Durch eine Zusammenfassung der jeweiligen Seiten aus (4.1.7) kommt es zu der folgenden Form

$$(a + x) \cdot (a + \delta \cdot z) \leq (a + z) \cdot (a + \delta \cdot x) \quad (4.1.8)$$

Da $a + \delta \cdot x > 0$ vorausgesetzt ist und damit wegen $z \geq x$ auch $a + \delta \cdot z > 0$ gilt, entspricht dies der obigen Formel (4.1.4) und die Aussage ist bewiesen. $\square$

4.2. Kaufentscheidung

Das Optimierungsproblem 2.1.1 ist eine Problemstellung, welche in der Literatur zum Beispiel in Form des Bahncard-Problems oder der Entscheidung eines Ski-Fahrers ab wie vielen Tagen Schnee sich der Kauf eigener Ski rechnet gerne für einführende Beispiele verwendet wird.

Für beide Probleme kann relativ schnell ein Online-Algorithmus mit bestmöglichen kompetitiven Faktor gefunden werden.

4.2.1. Bahncard-Problem

Im Bahncard-Problem geht es darum, ob und wann sich ein Reisender eine Bahncard mit Ermäßigung auf alle zukünftigen Zugfahrten kaufen soll. Der Preis der Bahncard sei dabei B Geldeinheiten und der Faktor der Ermäßigung sei λ , dass heißt eine Fahrkarte, welche normal p Geldeinheiten kosten würde, kostet nach dem Kauf der Bahncard nur

noch $(1 - \lambda)p$ Geldeinheiten. Die Gültigkeit der Bahncard sei dabei auf die Zeitdauer T , gemessen ab dem Zeitpunkt des Kaufes, beschränkt. Der betrachtete Zeitraum ist unabhängig von der Zeitdauer und kann theoretisch auch unendlich lang sein, das heißt insbesondere auch, dass es vorteilhaft sein kann, innerhalb des betrachteten Zeitraums eventuell nicht nur eine Bahncard zu kaufen, sondern mehrere nacheinander.

Für die deutsche Bahn ergeben sich im Jahr 2009 zum Beispiel für die klassische Bahncard folgende Daten: $B = 225\text{€}$ und $\lambda = \frac{1}{2}$ sowie $T = 1$ Jahr.

Der Reisende muss nun – vor dem Kauf der Bahncard – abschätzen, ob er danach noch so viele Fahrkarten benötigt, dass die Summe seiner Ersparnisse größer als die Kosten der Bahncard sind, also ob $\sum_z p_z \lambda \geq B$ gilt.

Algorithmus 4.2.1 (Bahncard-Algorithmus aus [KR02, BCSUM]).

Der Reisende kauft zum Zeitpunkt t eine Bahncard, wenn er noch keine besitzt und die möglich gewordenen Einsparungen von Kosten im Zeitraum $[\max\{t - T, 0\}, t]$ die Kosten der Bahncard überstiegen hätten.

Satz 4.2.2 (Algorithmus 4.2.1 ist optimal).

Der Algorithmus 4.2.1 hat einen strikten kompetitiven Faktor von $1 + \lambda$ und ist damit ein optimaler Online-Algorithmus. Insbesondere hat dieser für die deutsche Bahncard einen strikt kompetitiven Faktor von $\frac{3}{2}$.

Beweis zu Satz 4.2.2 nach [KR02, Satz 3.10 und Satz 3.12].

Der Beweis ist aufgegliedert in zwei Teile: Teil (i) weist den kompetitiven Faktor von $1 + \lambda$ nach und Teil (ii) definiert eine derartige Folge von Bahnfahrten, so dass jeder deterministische Algorithmus um $1 + \lambda$ erhöhte Kosten gegenüber der Offline-Bearbeitung hat.

- (i) Bei der Betrachtung des kompetitiven Faktors von Algorithmus 4.2.1 wird die gesamte Zeitspanne, innerhalb welcher der Reisende Bahn fährt, in mehrere verschiedene Zeiträume aufgeteilt.

Seien dazu $\tau_1, \dots, \tau_k$ die Zeitpunkte, zu welchen die Offline-Bearbeitung eine Bahncard kauft, zur einfacheren Darstellung sei τ_{k+1} der Zeitpunkt der letzten Bahnfahrt. Das heißt die Offline-Bearbeitung fährt in den Zeitintervallen $[\tau_i, \tau_i + T)$ zu den ermäßigten Preisen und somit liegen in diesen Zeitintervallen Bahnfahrten mit Kosten, deren Potential zur Einsparung höher ist als die Kosten B der Bahncard (sonst würde sich diese für die Offline-Bearbeitung nicht rechnen).

Da der Algorithmus 4.2.1 eine Bahncard genau dann kauft, wenn sich diese in den letzten T Zeiteinheiten bezahlt gemacht hätte und er zu diesem Zeitpunkt noch keine besitzt, wird auch dieser k mal die Bahncard kaufen.

In den Intervallen $[0, \tau_1)$ und $[\tau_i + T, \tau_{i+1})$ für $i = 1, \dots, k$ wird von der Offline-Bearbeitung der volle Preis gezahlt, dies führt dazu, dass innerhalb dieser Zeitintervalle der Algorithmus 4.2.1 nicht mehr bezahlt als die Offline-Bearbeitung.

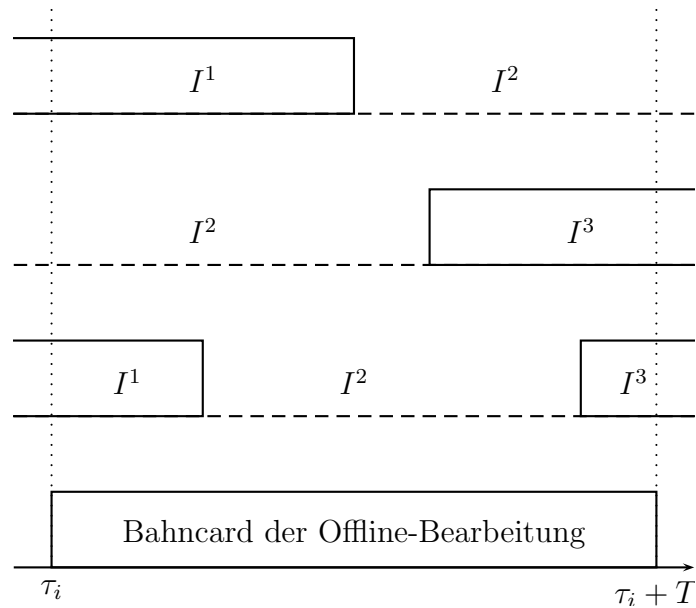


Abbildung 4.2.1.: Verschiedene Verteilungen der Phasen des Bahncard-Problems

Innerhalb der Zeiträume $[\tau_i, \tau_i + T)$ ergeben sich drei verschiedene Phasen: Die erste Phase I^1 und die letzte Phase I^3 sind Zeiträume, in welchen auch der Algorithmus 4.2.1 eine Bahncard besitzt, und während der mittleren Phase I^2 fährt dieser ohne Bahncard. Allerdings kann es auch vorkommen, dass bis zu zwei der Phasen wegfallen, so kann der Algorithmus 4.2.1 zum Beispiel vor dem Zeitpunkt τ_1 keine Bahncard besitzen und daher fällt dort zumindest die erste Phase weg. Eine schematische Darstellung der Phasen findet sich in Abbildung 4.2.1, dort ist in der untersten Ebene der Zeitraum eingezeichnet, in welchem die Offline-Bearbeitung eine Bahncard besitzt. Auf den darüber liegenden drei Ebenen sind drei der vier unterschiedlichen Verteilungen der Phasen eingezeichnet:

- (a) Der Online-Algorithmus besitzt zum Zeitpunkt τ_i schon eine Bahncard und kauft später noch einmal eine. Das dazugehörige Schema befindet sich in der zweiten Ebene von unten.
- (b) Bei der nächsten Ebene kauft der Online-Algorithmus innerhalb von $[\tau_i, \tau_i + T)$ eine Bahncard und besaß vorher noch keine Bahncard.
- (c) Die oberste Ebene stellt schließlich den Fall dar, dass der Online-Algorithmus zwar schon zum Zeitpunkt τ_i eine Bahncard besitzt, danach aber während $[\tau_i, \tau_i + T)$ keine mehr nachkauft.
- (d) Im Fall, dass beide gleichzeitig kaufen, ergibt sich für den Online-Algorithmus die gleiche Struktur wie für die Offline-Bearbeitung.

Die gesamten in einer Phase I entstehenden Kosten, ohne die Verwendung einer

Bahncard, werden als $P(I)$ zusammengefasst.

Also ergeben sich für die Offline-Bearbeitung über den ganzen Zeitraum Kosten von K_{Off} mit

$$K_{\text{Off}} = kB + \sum_{s=1}^k (1 - \lambda)P([\tau_s, \tau_s + T)) + \sum_{s=1}^k P([\tau_s + T, \tau_{s+1})) + P([0, \tau_1)) \quad (4.2.1)$$

und für den Algorithmus 4.2.1 ergibt sich folgende Schranke für die Kosten K_{Alg} des Algorithmus 4.2.1:

$$K_{\text{Alg}} \leq kB + \sum_{s=1}^k \{ (1 - \lambda)[P(I_s^1) + P(I_s^3)] + P(I_s^2) + P([\tau_s + T, \tau_{s+1})) \} + P([0, \tau_1)) \quad (4.2.2)$$

Dabei gilt die Beziehung $I_s^1 \cup I_s^2 \cup I_s^3 = [\tau_s, \tau_s + T)$ und $P(Z)$ ist die Summe der normalen Kosten der im Zeitraum Z auftretenden Bahnfahrten.

Der kompetitive Faktor kann durch den Quotienten zwischen (4.2.2) und (4.2.1) abgeschätzt werden zu:

$$c \leq \frac{kB + \sum_{s=1}^k \{ (1 - \lambda)[P(I_s^1) + P(I_s^3)] + P(I_s^2) + P([\tau_s + T, \tau_{s+1})) \} + P([0, \tau_1))}{kB + \sum_{s=1}^k \{ (1 - \lambda)P([\tau_s, \tau_s + T)) + P([\tau_s + T, \tau_{s+1})) \} + P([0, \tau_1))} \quad (4.2.3)$$

Eine Umsortierung von (4.2.3) und die Abschätzung des Quotienten unter Anwendung von Korollar 4.1.1 durch die Reduktion des gemeinsamen Anteils

$$\sum_{s=1}^k \{ (1 - \lambda)[P(I_s^1) + P(I_s^3)] + P([\tau_s + T, \tau_{s+1})) \} + P([0, \tau_1))$$

liefert die folgende Beziehung:

$$c \leq \frac{kB + \sum_{s=1}^k P(I_s^2)}{kB + \sum_{s=1}^k (1 - \lambda)P(I_s^2)} \quad (4.2.4)$$

Da die Länge des Zeitintervalls I_s^2 durch $|I_s^2| \leq T$ beschränkt ist und weil der Algorithmus 4.2.1 in I_s^2 keine Bahncard besitzt, gilt:

$$\lambda P(I_s^2) \leq B \iff P(I_s^2) \leq \frac{B}{\lambda}$$

Somit kann mithilfe von Korollar 4.1.2 für (4.2.4) folgende weitere Abschätzung

gefunden werden:

$$\begin{aligned} c &\leq \frac{kB + k \cdot \frac{B}{\lambda}}{kB + k(1-\lambda) \cdot \frac{B}{\lambda}} = \frac{B + \frac{B}{\lambda}}{B + (1-\lambda) \cdot \frac{B}{\lambda}} \\ &= \frac{\lambda B + B}{\lambda} \cdot \frac{\lambda}{\lambda B + (1-\lambda)B} = \frac{\lambda B + B}{B} = 1 + \lambda \end{aligned} \quad (4.2.5)$$

Damit ist Teil (i) abgeschlossen.

- (ii) Sei $\sigma = \{(r_1 \ t_1), \dots, (r_n \ t_n)\}$ eine Folge von Bahnfahrten mit relativ kleinen Kosten $\varepsilon > 0$. Die Anzahl der Fahrten n sei so gewählt, dass der Online-Algorithmus zum Zeitpunkt $t_n < T$ zum ersten Mal eine Bahncard kauft.

Das heißt für den Online-Algorithmus entstehen, da er die Bahncard nur einmal nutzen kann, Kosten K_{Alg} von

$$K_{Alg} = \sum_{i=1}^{n-1} \varepsilon + B + (1-\lambda)\varepsilon = n\varepsilon + B - \lambda\varepsilon. \quad (4.2.6)$$

Die Offline-Bearbeitung hat, abhängig davon, ob sich eine Bahncard rechnet oder nicht, die folgenden Kosten:

$$K_{Off} = \begin{cases} \sum_{i=1}^n \varepsilon & \text{falls } \lambda \sum_{i=1}^n \varepsilon \leq B \\ (1-\lambda) \sum_{i=1}^n \varepsilon + B & \text{falls } \lambda \sum_{i=1}^n \varepsilon > B \end{cases} \quad (4.2.7)$$

Also ergibt sich als Unterschranke für den kompetitive Faktor c der Quotient aus (4.2.7) und (4.2.6) zu:

$$c \geq \frac{K_{Alg}}{K_{Off}} = \begin{cases} \frac{n\varepsilon + B - \lambda\varepsilon}{n\varepsilon} & \text{falls } \lambda n\varepsilon \leq B \\ \frac{n\varepsilon + B - \lambda\varepsilon}{(1-\lambda)n\varepsilon + B} & \text{falls } \lambda n\varepsilon > B \end{cases} \quad (4.2.8)$$

Falls $\lambda n\varepsilon \leq B$ gilt, ergibt sich aus (4.2.8) folgende Unterschranke:

$$c \geq \frac{n\varepsilon - \lambda\varepsilon + n\lambda\varepsilon}{n\varepsilon} \geq 1 + \lambda - \frac{\lambda}{n}$$

Da für $\varepsilon \rightarrow 0$ die Anzahl der Fahrkarten bis zum Kauf der Bahncard durch den Algorithmus gegen unendlich geht, ergibt sich daraus im Grenzwert $c \geq 1 + \lambda$.

Für den Fall, dass sich eine Bahncard rechnet, also $B < \lambda n\varepsilon$ gilt, ergibt sich aus (4.2.8) die Unterschranke zu:

$$c \geq \frac{(1-\lambda)n\varepsilon + B - \lambda\varepsilon + \lambda n\varepsilon}{(1-\lambda)n\varepsilon + B} = 1 + \frac{\lambda n\varepsilon - \lambda\varepsilon}{n\varepsilon - \lambda n\varepsilon + B} \quad (4.2.9)$$

Durch die Umformung der Bedingung zu $B - \lambda n\varepsilon < 0$ lässt sich der Nenner von (4.2.9) mit $n\varepsilon$ nach oben abschätzen und es folgt:

$$c \geq 1 + \frac{\lambda n\varepsilon - \lambda\varepsilon}{n\varepsilon} = 1 + \lambda - \frac{\lambda}{n}$$

Mit der gleichen Argumentation über den Grenzwert für ε gegen Null mit unendlich vielen Fahrten, wie schon im anderen Fall, ergibt sich die gewünschte Schranke. $\square$

4.2.2. Ski-Fahrer-Problem

Das Ski-Fahrer-Problem ist ein Spezialfall des Problems aus Unterabschnitt 4.2.1, bei dieser Aufgabenstellung muss sich ein Ski-Fahrer entscheiden, wann er eigene Ski kauft und wie lange er welche mieten will.

Für den zu erlangenden Faktor der Ermäßigung gilt bei diesem Problem $\lambda = 1$. Zudem gibt es, anders als beim allgemeinen Bahncard-Problem einen festgelegten konstanten Preis p (mit $B \bmod p = 0$) für die einzelnen Leistungen, welcher wesentlich kleiner ist als die für den Kauf der Ski zu zahlende Geldmenge B .

Der vorher festgelegte konstante Preis führt für diesen Spezialfall zu einem besseren strikt kompetitiven Faktor als der erwartete Faktor $1 + \lambda = 2$. Zusätzlich vereinfacht die Annahme, dass einmal gekaufte Ski grundsätzlich eine unendliche Gültigkeitsdauer $T = \infty$ besitzen, die Analyse, da damit die Betrachtung über mehrere Zeitperioden hinfällig wird.

Algorithmus 4.2.3 (Ski-Algorithmus nach [KR02, Unterabschnitt 2.2.1]).

Der Ski-Algorithmus kauft an einem Tag zum Skifahren seine eigenen Ski genau dann, wenn er mit diesem Tag genau n Tage gefahren ist und $np \geq B$ gilt.

Satz 4.2.4 (Algorithmus 4.2.3 ist optimal).

Der Algorithmus 4.2.3 ist strikt $2 - \frac{p}{B}$ kompetitiv und es kann keinen besseren deterministischen Algorithmus geben.

Beweis zu Satz 4.2.4 nach [KR02, Unterabschnitt 2.2.1].

Auch dieser Beweis gliedert sich wieder in zwei Teile:

- (i) Dieser Teil befasst sich mit dem Beweis des kompetitiven Faktors.

Es gibt zwei verschiedene Szenarien für die Summe der Kosten (hierbei ist n die Anzahl der Tage an denen Ski gefahren wird):

- (a) $np < B$
- (b) $np \geq B$

Die Offline-Bearbeitung hat, egal welches Szenario eintritt, immer folgende Kostenstruktur:

$$K_{\text{Off}} = \min\{B, np\} \quad (4.2.10)$$

Der Algorithmus 4.2.3 hat je nach Szenario unterschiedliche Kosten K_{Alg} :

$$K_{\text{Alg}} = \begin{cases} np & \text{bei Szenario (i)(a)} \\ 2B - p & \text{bei Szenario (i)(b)} \end{cases} \quad (4.2.11)$$

Eine Unterschranke für den kompetitiven Faktor ergibt sich dann aus dem Quotienten von (4.2.11) und (4.2.10) zu:

$$c \geq \frac{K_{\text{Alg}}}{K_{\text{Off}}} = \begin{cases} \frac{np}{np} = 1 & \text{bei Szenario (i)(a)} \\ \frac{2B-p}{B} = 2 - \frac{p}{B} & \text{bei Szenario (i)(b)} \end{cases}$$

Damit gilt $c \geq 2 - \frac{p}{B}$ und die Behauptung ist bewiesen.

- (ii) Für den Beweis der Optimalität genügt es zu zeigen, dass $2 - \frac{p}{B}$ eine allgemein gültige Unterschranke für den strikt kompetitiven Faktor aller deterministischen Algorithmen ist.

Dazu wird ein derartiges Szenario aufgebaut, so dass jeder deterministische Algorithmus mindestens so hohe Kosten K_{Alg} hat, dass $K_{\text{Alg}} \geq (2 - \frac{p}{B})K_{\text{Off}}$ erfüllt ist.

Das Szenario wird so gestaltet, dass es gerade genügend Tage, mit der Möglichkeit Ski zu fahren, bis zum Kauf der Ski durch den Algorithmus gibt. Nach dem Kauf wird das Skifahren unmöglich. Sei n die Anzahl der Tage an denen Ski gefahren werden kann. Für den Algorithmus entstehen immer folgende Kosten K_{Alg} :

$$K_{\text{Alg}} = (n - 1)p + B \quad (4.2.12)$$

Die Offline-Bearbeitung hat wieder nur Kosten K_{Off} mit:

$$K_{\text{Off}} = \min\{B, np\} \quad (4.2.13)$$

Der Quotient aus (4.2.12) und (4.2.13) liefert die folgende Unterschranke für den kompetitiven Faktor c :

$$c \geq \frac{(n - 1)p + B}{\min\{B, np\}} \quad (4.2.14)$$

Würde sich nun der Kauf von eigenen Ski rechnen, also wenn $np \geq B$ ist, dann wird aus (4.2.14)

$$c \geq \frac{(n - 1)p + B}{B} \geq \frac{B - p + B}{B} = 2 - \frac{p}{B}.$$

Falls sich der Kauf nicht rechnet kommt es mit $B > np$ aus (4.2.14) zu

$$c \geq \frac{(n-1)p + B}{np} \geq 1 + \frac{B-p}{np} \geq 1 + \frac{B-p}{B} \geq 2 - \frac{p}{B}. \quad \square$$

4.3. Punktsuche

Beim Problem der Punktsuche aus Optimierungsproblem 2.1.2 geht es darum, einen bestimmten Punkt mit unbekannter Position in einem vorgegebenen Raum zu finden. Da schon im zweidimensionalen Raum das Problem der Laufwege alles relativ unübersichtlich macht, wird hier nur (wie in [KR02]) der eindimensionale Raum betrachtet. Im eindimensionalen Raum gestaltet sich das Optimierungsproblem wie folgt:

Es wird im Ursprungspunkt gestartet und nach einem bestimmten unbekannten Zielpunkt $a \in \mathbb{R}$ mit $|a| \geq 1$ gesucht, welcher erst als gefunden gilt, wenn er das erste Mal berührt wird. Für Standorte a mit $|a| < 1$ macht die kompetitive Analyse keinen Sinn, da dann a sehr nahe bei Null gewählt werden kann und somit der kompetitive Faktor immer gegen unendlich geht.

Für diese Aufgabenstellung gibt es einen schönen einfachen Algorithmus:

Algorithmus 4.3.1 (Punktsuche-Algorithmus nach [KR02, Übung 2.6]).

Der Suchende wählt eine positive Zahl $\alpha > 1$ und setzt den Zähler i auf 1. Nun geht er zuerst α Einheiten in positiver Richtung, und anschließend zum Ursprung zurück.

Jedes mal wenn der Suchende wieder zum Ursprung kommt zählt er den Wert von i um eins nach oben und läuft weitere α^i Einheiten in seiner aktuellen Laufrichtung. Ist er dann bei $(-1)^{i+1} \cdot \alpha^i$ angekommen, so kehrt er um und läuft wieder zum Ursprung.

Zur Verdeutlichung des Vorgehens bei der Bearbeitung der Punktsuche unter Verwendung von Algorithmus 4.3.1 zeigt Abbildung 4.3.1 eine kleine Skizze mit $\alpha = 2$ und $a = 3$. Dabei sind die mit w_i bezeichneten Pfeile die einzelnen Streckenabschnitte, w_1 und w_3 sind die Bewegungen in Vorwärtsrichtung um α und α^2 , sowie w_2 und w_4 die zugehörigen Bewegungen in Rückwärtsrichtung. Dabei entspricht eine Bewegung in Vorwärtsrichtung einer Bewegung vom Ausgangspunkt hinweg zu einem Teil des eindimensionalen Raums der noch nicht besucht wurde, analog verläuft dann eine Bewegung in Rückwärtsrichtung genau anders herum, entspricht also einer Rückkehr zum Ausgangspunkt. Mit der Bewegung in Vorwärtsrichtung w_5 , welche eigentlich um $\alpha^3 = 8$ Einheiten nach rechts geht, wird der gesuchte Punkt a endlich erreicht.

Satz 4.3.2 (Kompetitiver Faktor für Algorithmus 4.3.1).

Der Algorithmus 4.3.1 hat einen strikt kompetitiven Faktor von mindestens neun, und dieser kann nur durch die Vorgabe von $\alpha = 2$ tatsächlich erreicht werden.

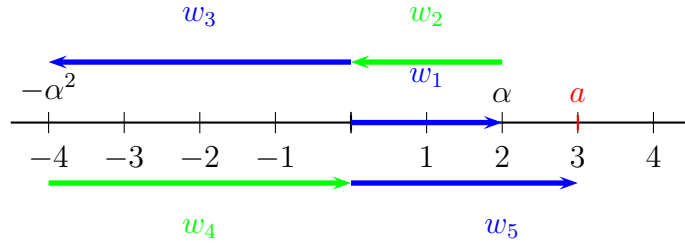


Abbildung 4.3.1.: Beispiel für den Algorithmus zur Punktssuche

Beweis zu Satz 4.3.2 durch eine Modifikation der Lösung zu [KR02, Übung 2.6].

Sei $\varepsilon > 0$ und k die erste natürliche Zahl mit

$$\alpha^k + \varepsilon \leq |a| \leq \alpha^{k+2} \text{ sowie } (-1)^{k+1} = \frac{a}{|a|}.$$

Die Kosten K_{Alg} des Algorithmus 4.3.1 und der Offline-Bearbeitung K_{Off} ergeben sich dann zu:

$$K_{Off} = |a| \qquad K_{Alg} = |a| + 2 \sum_{i=1}^{k+1} \alpha^i \quad (4.3.1)$$

Durch die Anwendung der Summenformel wird aus den Kosten des Algorithmus:

$$K_{Alg} = |a| + 2 \cdot \frac{1 - \alpha^{k+2}}{1 - \alpha} - 2 \quad (4.3.2)$$

Mit der Eigenschaft $\alpha^k + \varepsilon \leq |a|$ kann (4.3.2) abgeschätzt werden zu:

$$\begin{aligned} K_{Alg} &= |a| + 2 \cdot \frac{\alpha^{k+2} - 1}{\alpha - 1} - 2 \leq |a| + 2 \cdot \frac{\alpha^2 |a| - 1}{\alpha - 1} - 2 \\ &= \frac{|a|(\alpha - 1) + 2\alpha^2 |a| - 2 - 2(\alpha - 1)}{\alpha - 1} = \frac{|a|[2\alpha^2 + \alpha - 1] - 2\alpha}{\alpha - 1} \end{aligned} \quad (4.3.3)$$

Die erste Ableitung von (4.3.3) nach α ist

$$\begin{aligned} &\frac{|a|[(4\alpha + 1)(\alpha - 1) - (2\alpha^2 + \alpha - 1)]}{(\alpha - 1)^2} - \frac{2(\alpha - 1) - 2\alpha}{(\alpha - 1)^2} \\ &= \frac{|a|[4\alpha^2 - 3\alpha - 1 - 2\alpha^2 - \alpha + 1] + 2}{(\alpha - 1)^2} = \frac{|a|[2\alpha^2 - 4\alpha] + 2}{(\alpha - 1)^2}. \end{aligned} \quad (4.3.4)$$

Die quadratische Lösungsformel liefert als Nullstellen

$$\alpha_{1,2} = \frac{4|a| \pm \sqrt{16|a|^2 - 16|a|}}{4|a|}$$

Da $\alpha > 1$ gilt, ist $\alpha_1 = 1 + \sqrt{1 - \frac{1}{|a|}}$ der einzige relevante Extrempunkt. Die zweite Ableitung von (4.3.3) nach α ist

$$\begin{aligned} & \frac{|a| [(4\alpha - 4)(\alpha - 1)^2 - 2(\alpha - 1)(2\alpha^2 - 4\alpha)] - 4(\alpha - 1)}{(\alpha - 1)^4} \\ &= \frac{|a| [(4\alpha - 4)(\alpha^2 - 2\alpha + 1) - 2(2\alpha^3 - 4\alpha^2 - 2\alpha^2 + 4\alpha)] - 4(\alpha - 1)}{(\alpha - 1)^4} \\ &= \frac{|a| [4\alpha^3 - 8\alpha^2 + 4\alpha - 4\alpha^2 + 8\alpha - 4 - 4\alpha^3 + 12\alpha^2 - 8\alpha] - 4(\alpha - 1)}{(\alpha - 1)^4} \\ &= \frac{4|a|(\alpha - 1) - 4(\alpha - 1)}{(\alpha - 1)^4} = \frac{4|a| - 4}{(\alpha - 1)^3}. \quad (4.3.5) \end{aligned}$$

Da die zweite Ableitung (4.3.5) wegen $\alpha_1 > 1$ und $a \geq 1$ positiv ist, liegt bei α_1 ein Minimum. Aus der Darstellung von α_1 folgt außerdem $\alpha_1 \leq 2$, dieser Wert wird im Grenzwert für $|a|$ gegen unendlich tatsächlich angenommen. Das bedeutet, da bei der Bestimmung des kompetitiven Faktors alle Instanzen berücksichtigt werden müssen, also auch der Grenzfall, dass a im Unendlichen liegt, dass um auch hierfür minimale Kosten zu erhalten die anfängliche Wegstrecke auf zwei gesetzt werden muss. Nach dem Einsetzen der Oberschranke $\alpha = 2$ in (4.3.3) ergibt sich folgende Abschätzung für die Kosten des Algorithmus 4.3.1:

$$K_{Alg} \leq |a| \cdot \frac{2 \cdot 4 + 2 - 1}{2 - 1} - \frac{2 \cdot 2}{2 - 1} = 9|a| - 4$$

Mit (4.3.1) ergibt sich schließlich der strikt kompetitive Faktor von neun. $\square$

Nachdem damit der minimale kompetitive Faktor von neun für den Algorithmus 4.3.1 mit $\alpha = 2$ nachgewiesen wurde, stellt sich natürlich die Frage ob die Möglichkeit besteht, dass es einen Algorithmus mit einem besseren strikt kompetitiven Faktor gibt. Krumke und Rambau verweisen dazu in ihrem Skript [KR02] auf einen Beweis von Baeza-Yates in [BCR93, Kapitel 2]. In diesem geht es um die Suche nach einem unbekannten Punkt auf einer Linie, dies entspricht genau der oben in Optimierungsproblem 2.1.2 definierten Punktsuche.

Satz 4.3.3 (Optimaler kompetitiver Faktor).

Jeder deterministische Algorithmus hat für das Optimierungsproblem 2.1.2 einen kompetitiven Faktor c , welcher nicht kleiner als neun ist.

Diese Aussage findet sich wie oben erwähnt in [BCR93, Kapitel 2], da der darin enthaltene Beweis etwas umständlich ist, soll hier eine alternative Möglichkeit des Beweises nach dem Artikel [HIKL99, Unterabschnitt 2.1] dargestellt werden. Dabei werden die Bezeichnungen der Punktsuche übernommen.

Beweis zu Satz 4.3.3.

Zuerst einmal wird mit $f(i)$ die Schrittweite im Schritt i bezeichnet. Somit gilt mit der Terminologie von Abbildung 4.3.1:

$$|w_{2i}| = |w_{2i-1}| = f(i) \equiv \alpha^i$$

Um bei der Suche Erfolg zu haben, muss zusätzlich noch $f(i) > f(i-2)$ für alle Schritte $i > 2$ gelten, damit nicht mehrmals am gleichen Punkt umgedreht wird. Durch diese Anforderung wird eine echte Fortbewegung erzwungen.

Sei k nun derjenige Schritt in Richtung des gesuchten Punktes a , für welchen

$$f(k) + \varepsilon \leq |a| \leq f(k+2) \quad (4.3.6)$$

mit $\varepsilon > 0$ gilt. Damit ergibt sich für den kompetitiven Faktor c :

$$c \geq \frac{2 \sum_{j=1}^{k+1} f(j) + |a|}{|a|} \quad (4.3.7)$$

Nach Korollar 4.1.1 entsteht aus (4.3.7) im schlimmsten Fall:

$$c \geq \frac{2 \sum_{j=1}^{k+1} f(j) + f(k) + \varepsilon}{f(k) + \varepsilon} = 1 + 2 \cdot \frac{\sum_{j=1}^{k+1} f(j)}{f(k) + \varepsilon} \quad (4.3.8)$$

Nach Umformung von (4.3.8) entsteht:

$$2 \sum_{j=1}^{k+1} f(j) + f(k) + \varepsilon \leq c(f(k) + \varepsilon) \quad (4.3.9)$$

Die Abschätzung in (4.3.9) muss für alle Werte k und ε gelten und somit auch im Grenzwert ε gegen Null. Wird dazu auch noch

$$\delta = \frac{c-3}{2} \quad (4.3.10)$$

gesetzt, so ergibt sich (4.3.9) zu:

$$f(k+1) \leq \delta f(k) - \sum_{j=1}^{k-1} f(j) \quad (4.3.11)$$

Wird die Abschätzung aus (4.3.11) mehrmals (hier $t \leq k - 1$ mal) auf sich selbst angewendet, so ergibt sich folgende rekursive Formel:

$$f(k+1) \leq a_t f(k-t) - b_t \sum_{j=1}^{k-1-t} f(j) \quad (4.3.12)$$

Dabei sind die Folgen a_t und b_t wie folgt rekursiv definiert:

$$\begin{aligned} a_0 &= \delta & a_{t+1} &= a_t \delta - b_t \\ b_0 &= 1 & b_{t+1} &= a_t + b_t \end{aligned}$$

Zu dieser Rekursionsformel gehört die Übergangsmatrix $T = \begin{pmatrix} \delta & -1 \\ 1 & 1 \end{pmatrix}$, mit welcher $\begin{pmatrix} a_{t+1} \\ b_{t+1} \end{pmatrix} = T \begin{pmatrix} a_t \\ b_t \end{pmatrix}$ gilt. Das charakteristische Polynom dazu ist:

$$\begin{aligned} \det(T - xE) &= (\delta - x)(1 - x) + 1 = \delta - x\delta - x + x^2 + 1 \\ &= x^2 - (\delta + 1)x + (\delta + 1) \end{aligned} \quad (4.3.13)$$

Als Lösungen des charakteristischen Polynoms aus (4.3.13) können über die quadratische Lösungsformel die beiden folgenden Werte ermittelt werden:

$$x_{1,2} = \frac{1}{2} \left(\delta + 1 \pm \sqrt{(\delta + 1)(\delta - 3)} \right) \quad (4.3.14)$$

Da die Wurzel aus (4.3.14) nur dann eine reelle Lösung erhält, wenn δ größer gleich drei ist, kann damit über die Auflösung von (4.3.10) mit $\delta = 3$ nach

$$c = 2\delta + 3 = 9$$

ein minimaler kompetitiver Faktor echt kleiner als neun ausgeschlossen werden. Somit hat sich auch der Algorithmus 4.3.1 als optimaler Online-Algorithmus erwiesen. $\square$

4.4. Pack-Problem

Beim Problem des Packens, also Optimierungsproblem 2.1.3, geht es um das Verpacken von nacheinander eintreffenden Gegenständen in Kisten, dabei darf der $(i+1)$ -te Gegenstand für die Online-Bearbeitung erst betrachtet werden, wenn der Gegenstand i schon gepackt worden ist.

Die einfachste Problemstellung ist das Verpacken von n Gegenständen, mit dem jeweiligen Platzbedarf r_i , in Kisten mit fester Kapazität B . Dabei ist es das Ziel der

Aufgabenstellung die Anzahl k der verwendeten Kisten zu minimieren. Ein einfacher Online-Algorithmus für diese Aufgabenstellung ist FirstFit.

Algorithmus 4.4.1 (FirstFit).

Der FirstFit Algorithmus ordnet die Kisten nach einer Reihenfolge an und legt jeden ankommenden Gegenstand in jeweils die erste Kiste in welche er noch hineinpasst.

Die Aussagen zur Optimalität des kompetitiven Faktors und die dazugehörigen Beweise finden sich in [KR02, Aufgabe 2.4 und Teil C].

Satz 4.4.2 (Kompetitiver Faktor für Algorithmus 4.4.1).

Algorithmus 4.4.1 ist für das einfache Pack-Problem strikt kompetitiv mit dem Faktor zwei.

Beweis zu Satz 4.4.2.

Ohne Beschränkung der Allgemeinheit gelte $B = 1$ und $r_i \leq 1$ für alle Gegenstände i . Sei weiterhin k_{Alg} die Anzahl der vom Algorithmus 4.4.1 verwendeten Kisten und seien davon t Kisten nur mit maximal 50 % gefüllt, sowie $\sum_{i=1}^n r_i$ der gesamte Platzbedarf aller Gegenstände. Der gesamte Platzbedarf ergibt eine Unterschranke für die minimale Anzahl an Kisten, also

$$k_{\text{Off}} \geq \sum_{i=1}^n r_i$$

und damit gilt:

$$c \leq \frac{k_{Alg}}{k_{\text{Off}}} \leq \frac{k_{Alg}}{\sum_{i=1}^n r_i} \quad (4.4.1)$$

Daraus ergeben sich drei verschiedene Situationen:

- (i) Der Fall $t \geq 2$ kann nicht auftreten, da sonst die zweite dieser Kisten mit einem Grad der Füllung von maximal 50 % problemlos in die erste dieser Kisten umgefüllt werden könnte. Der Algorithmus 4.4.1 legt die Gegenstände allerdings immer in die erste Kiste in welche diese noch rein passen, und somit kann nicht mehr als eine Kiste halb leer sein.
- (ii) Falls jede der k_{Alg} Kisten schon zu mehr als die Hälfte gefüllt ist, so ergibt sich $k_{Alg} \leq 2 \sum_{i=1}^n r_i$ und damit aus (4.4.1) $c \leq 2$.
- (iii) Falls eine der Kisten nur einen Grad der Füllung von $\alpha \leq \frac{1}{2}$ hat, so müssen wegen der Vorgehensweise des Algorithmus 4.4.1 die anderen $(k_{Alg} - 1)$ Kisten mindestens einen Grad der Füllung von $1 - \alpha$ haben. Dadurch lässt sich der gesamte Platzbedarf wie folgt nach unten abschätzen:

$$\sum_{i=1}^n r_i \geq (k_{Alg} - 1)(1 - \alpha) + \alpha$$

Daraus ergibt sich wiederum eine Oberschranke für k_{Alg} :

$$k_{Alg} - 1 \leq \frac{\sum_{i=1}^n r_i - \alpha}{(1 - \alpha)} \quad \Rightarrow \quad k_{Alg} \leq 1 + \frac{\sum_{i=1}^n r_i - \alpha}{(1 - \alpha)} \quad (4.4.2)$$

Das Maximum für (4.4.2) auf dem Intervall $\alpha \in [0, \frac{1}{2}]$ liegt wegen

$$\sum_{i=1}^n r_i \geq (k_{Alg} - 1)\alpha$$

bei $\alpha = \frac{1}{2}$ und somit ergibt sich wieder $k_{Alg} \leq 2 \sum_{i=1}^n r_i$.

Damit ist der Algorithmus 4.4.1 strikt kompetitiv für den Faktor zwei. $\square$

Nach der speziellen Betrachtung des Algorithmus nun noch eine allgemeine Unterschranke für alle deterministischen Algorithmen.

Lemma 4.4.3 (Unterschranke für den kompetitiven Faktor).

Kein deterministischer Algorithmus kann einen besseren kompetitiven Faktor als $\frac{4}{3}$ für die obige Problemstellung erreichen.

Beweis zu Lemma 4.4.3.

Der Beweis läuft über einen Widerspruch.

Sei also ein c -kompetitiver Algorithmus mit $c < \frac{4}{3}$ gegeben und σ eine Folge von n Gegenständen der Größe $\frac{1}{2} - \varepsilon$ und von n Gegenständen mit Größe $\frac{1}{2} + \varepsilon$. Die Offline-Bearbeitung braucht also nur n Kisten.

Sei m die Anzahl der nach den ersten n Gegenständen verbrauchten Kisten und ℓ die restliche Anzahl von Kisten, welche danach noch vom Algorithmus geöffnet werden. Somit gilt dann also:

$$k_{Alg} = m + \ell$$

Durch die Wahl der Gegenstände und der Sequenz sind nun in den ersten m Kisten jeweils maximal zwei Gegenstände und in den restlichen ℓ Kisten jeweils nur einer. Daraus folgt $2m + \ell \geq 2n$ und damit $\ell \geq 2(n - m)$ also gilt:

$$k_{Alg} \geq 2n - m \quad (4.4.3)$$

Nach der Annahme vom Anfang ist der Algorithmus c -kompetitiv und somit gilt für die isolierte Bearbeitung der ersten n Gegenstände dieser Faktor ebenfalls. Also ergibt sich:

$$m \leq c \cdot \frac{n}{2} \leq \frac{4}{3} \cdot \frac{n}{2} = \frac{2}{3}n \quad (4.4.4)$$

Die Abschätzung für m aus (4.4.4) eingesetzt in (4.4.3) liefert schließlich einen Wider-

spruch zur obigen Annahme von $c < \frac{4}{3}$:

$$k_{Alg} \geq 2n - \frac{2}{3}n = \frac{4}{3}n$$

□

4.5. Verwaltung von Speicherseiten

Bei der Verwaltung der Speicherseiten nach Optimierungsproblem 2.1.3 geht es um die Verwaltung von Speicherinhalten in einem Computer mit zwei verschiedenen Arten von Speicher. Ein normaler Computer zum Beispiel besitzt einen kleinen aber schnellen Arbeitsspeicher (RAM) und einen wesentlich größeren, dafür aber langsameren, Hauptspeicher (die Festplatte).

Zur Vereinfachung des Modells wird angenommen, dass immer nur ganze Speicherseiten abgerufen werden, welche alle die gleiche Größe haben und im Hauptspeicher dauerhaft abgelegt sind. Sowohl Hauptspeicher als auch Arbeitsspeicher sind in Einheiten unterteilt welche jeweils eine Speicherseite aufnehmen können. Im Gegensatz zum Hauptspeicher ist der Inhalt des Arbeitsspeicher variabel und die Einheiten des Arbeitsspeichers können jederzeit einzeln mit Seiten aus dem Hauptspeicher neu geladen werden. Diese Lade-Operation zwischen den beiden Arten des Speichers ist der arbeitsaufwendige Teil der Speicherverwaltung.

Der Computer erhält, während mit ihm gearbeitet wird Anforderungen für bestimmte Dateninhalte in seinem Speicher. Befinden sich diese Daten aktuell in seinem Arbeitsspeicher, so kann er diese gleich verwenden, sind diese allerdings nur im Hauptspeicher vorhanden, so müssen diese erst einmal in den Arbeitsspeicher geladen werden, bevor sie verarbeitet werden können. Aufgabe der Speicherverwaltung ist es nun die Anzahl der notwendigen Ladevorgänge zu minimieren, indem sie die aktuell vorhandenen Seiten im Arbeitsspeicher so aufbaut, dass bei der Bearbeitung der danach ankommenden Anfragen für Daten möglichst wenige Zugriffe auf den Hauptspeicher Anwendung finden.

Bei der Online-Optimierung geht es nun wie in Optimierungsproblem 2.1.4 beschrieben, darum, ohne Kenntnis der zukünftigen Anfragen, die Daten im Arbeitsspeicher so zu verwalten, dass die Anzahl der Lade-Operationen aus dem Hauptspeicher möglichst minimal wird. Immer wenn eine solche Lade-Operation auftritt wird von einem Seitenfehler gesprochen.

Algorithmus 4.5.1 (Least-Recently-Used).

Wenn ein Seitenfehler auftritt, dann wird die Seite im Arbeitsspeicher gelöscht, deren letzte Anforderung am weitesten zurück liegt.

Für diesen Algorithmus lässt sich folgende Aussage beweisen:

Satz 4.5.2 (Algorithmus 4.5.1 ist optimal).

Algorithmus 4.5.1 hat einen strikt kompetitiven Faktor von k , wobei k die Größe des Arbeitsspeichers ist. Zudem existiert kein besserer deterministischer Algorithmus.

Beweis zu Satz 4.5.2 nach [KR02, Satz 4.6 und Satz 4.9].

Wie in dem vorherigen Abschnitt gestaltet sich der Beweis wieder in zwei verschiedene Teile. Dabei sei ohne Beschränkung der Allgemeinheit der Arbeitsspeicher am Anfang schon voll belegt.

- (i) Zu zeigen ist, dass jeder deterministische Algorithmus mindestens einen kompetitiven Faktor von k hat.

Sei $\{p_1, \dots, p_{k+1}\}$ eine Menge von Speicherseiten und $p_1, \dots, p_k$ der anfängliche Zustand des Arbeitsspeichers. Wenn die Sequenz der Anfragen so konstruiert wird, dass nach der Anfrage für p_{k+1} in den nächsten $k - 1$ Anfragen immer die Seite angefragt wird, welche von dem Algorithmus im vorherigen Schritt gelöscht wurde, so hat dieser Algorithmus sicher k Seitenfehler. Bei der deterministischen Offline-Bearbeitung kann unter Kenntnis der nächsten k Anfragen genau die Seite der anfänglichen Belegung aus dem Hauptspeicher gelöscht werden, welche nicht in diesen Anfragen enthalten ist. Dann entsteht nur ein Seitenfehler bei der Bearbeitung der Anfrage p_{k+1} und damit ist der kompetitive Faktor immer größer gleich k .

- (ii) Algorithmus 4.5.1 ist strikt k -kompetitiv.

Sei σ eine Sequenz von Anfragen für Speicherseiten, mit Phasen $P(0), P(1), \dots$, so dass für Algorithmus 4.5.1 während $P(0)$ maximal k Seitenfehler und während der anderen Phasen immer genau k Seitenfehler auftreten. Diese Aufteilung kann für jede mögliche Sequenz getroffen werden. Dazu muss diese nur dem Algorithmus 4.5.1 übergeben werden, und anschließend, nach der Bearbeitung, können ausgehend von der letzten Anfrage und den bei der Bearbeitung auftretenden Seitenfehlern, die Anfragen den einzelnen Phasen zugeordnet werden. Wenn es nun gelingt eine Sequenz zu finden, für welche die Offline-Bearbeitung nur einen Seitenfehler pro Phase benötigt, dann ist das gewünschte Ergebnis erreicht.

Die dazu wichtige Aussage ist:

Jede Phase $P(i)$ der Sequenz mit $i > 0$ beinhaltet k Speicherseiten welche verschieden sind zu der Seite p der letzten Anfrage der vorherigen Phase $P(i - 1)$. Damit hat auch die Offline-Bearbeitung mindestens einen Seitenfehler pro Phase, da außer p nur $k - 1$ andere Seiten zu Beginn der Phase $P(i)$ im Arbeitsspeicher sein können.

Wenn die k Seitenfehler des Algorithmus 4.5.1 von k verschiedenen Seiten verursacht werden ist die Aussage trivialerweise erfüllt. Aber was passiert, wenn eine Seite zweimal einen Seitenfehler innerhalb einer Phase verursacht. Damit dies geschehen kann, muss eine Seite q einmal in den Arbeitsspeicher geladen und danach wieder gelöscht werden. Da der Algorithmus 4.5.1 eine Speicherseite aus dem Arbeitsspeicher nur dann löscht, wenn ihre letzte Anforderung am weitesten zurück liegt, müssen zwischen dem Laden und dem Löschen einer Speicherseite Anfragen

für k weitere verschiedene Speicherseiten liegen. Also liegen in $P(i)$ Anfragen zu $k + 1$ verschiedenen Speicherseiten und davon sind mindestens k Seiten ungleich p .

Der Fall, dass der Algorithmus 4.5.1 in $P(i)$ einen durch p verursachten Seitenfehler beinhaltet, führt über genau die gleiche Argumentation (nur das erste Laden von p tritt nicht innerhalb von $P(i)$ auf) zu dem gewünschten Ergebnis.

Somit ist der Beweis zu Algorithmus 4.5.1 abgeschlossen. $\square$

Bemerkung 4.5.3 (Start mit leerem Arbeitsspeicher).

Der Fall eines am Anfang leeren Arbeitsspeichers kann auf die obige Beweisführung zurückgeführt werden, da bei leerem Arbeitsspeicher für die ersten k verschiedenen Seiten immer k Seitenfehler auftreten und sich der kompetitive Faktor mithilfe von Korollar 4.1.1 auf den des Falles mit einem vollen Arbeitsspeicher überführen lässt.

4.6. Rundreisen

Vom Rundreise-Problem, wie es in Optimierungsproblem 2.2.1 allgemein beschrieben wird, existieren mehrere Ausprägungen. Zum Beispiel befasst sich das Kapitel 7 mit der Fragestellung einer Rundreise auf der Kreislinie des Einheitskreises. Weitere Anwendungen sind unter anderem die Suche nach Rundreisen innerhalb eines vorgegebenen Netzwerkes, hier ist die Arbeit eines Dienstleisters mit Bereitschaftsdienst ein klassisches Beispiel, oder auch innerhalb eines bestimmten vorgegebenen Raumes oder Gebiets ohne Beschränkungen bei der Menge der Verbindungen. Da die Betrachtung solcher Probleme mit der kompetitiven Analyse schnell unübersichtlich wird, beschränkt sich die Literatur eher auf die *relativ einfachen* Fälle wie zum Beispiel mit der Halbgeraden $\mathbb{R}_0^+$ oder der Geraden $\mathbb{R}$ als zugrunde liegendem Gebiet.

Bei der Rundreise geht es darum, eine solche Route zu finden, dass die Reisepunkte aller Anfragen nach deren Bekanntgabe besucht werden. Unterscheidungen der einzelnen Fragestellungen ergeben sich nun in der Grundmenge für die Orte und der Festlegung der zugelassenen möglichen Verbindungen. Ohne Beschränkung der Allgemeinheit beträgt bei den vorgestellten Beispielen die verwendete Reisegeschwindigkeit eine Entfernung von Eins pro Zeiteinheit.

4.6.1. Rundreise auf der positiven Halbgeraden

Unter der Rundreise auf der positiven Halbgeraden wird eine Aufgabenstellung verstanden, in welcher sich alle für Anfragen mögliche Orte auf der positiven Halbgeraden befinden mit dem Nullpunkt als Ausgangspunkt.

Unter diesen Bedingungen ist es möglich, einen relativ einfachen Algorithmus zu formulieren, welcher seine Vorgehensweise direkt auf den gewünschten kompetitiven Faktor aufstützt:

Algorithmus 4.6.1 („Wait Smartly“ nach [KR02, Kapitel 3]).

Sei $s(t)$ die momentane Position des Algorithmus zum Zeitpunkt t und α der gewünschte strikt kompetitive Faktor. Der Algorithmus unterscheidet zwischen zwei Fällen innerhalb der ihm vorliegenden Anfragen $\sigma_{\leq t} = \{\sigma_1, \dots, \sigma_{m(t)}\}$ mit $\sigma_i = (t_i \ s_i)$ und $t_i \leq t$ für alle $i \in \{1, \dots, m(t)\}$:

- (i) Es gibt noch eine nicht bearbeitete Anfrage σ_i mit $s_i > s(t)$.
- (ii) Es gibt nur noch nicht bearbeitete Anfragen σ_i mit $s_i \leq s(t)$.

Beim Eintreten von Fall (i) bewegt er sich solange auf der Halbgeraden nach rechts, bis es keine solche Anfrage mehr gibt.

Wenn der andere Fall (ii) auftritt, dann berechnet der Algorithmus erst die Lösung der Offline-Bearbeitung für die Sequenz der Anfragen $\sigma_{\leq t}$. Mit dieser Lösung berechnet er die für ihn noch mögliche Wartezeit W im Punkt $s(t)$ ohne seinen ihm vorgegebenen strikt kompetitiven Faktor α zu verletzen. Die mögliche Wartezeit ergibt sich dann zu:

$$W = \alpha T_{\text{Off}} - s(t) - t$$

Solange W echt positiv ist, wird beim Punkt $s(t)$ genau W Zeiteinheiten gewartet, wenn aber W gleich Null gilt, dann ist die Wartezeit vorüber und es geht zum Ausgangspunkt zurück.

Das Warten wird allerdings sofort unterbrochen, sobald eine neue Anfrage bekannt wird und daraufhin werden wieder die beiden Fälle untersucht und derjenige, welcher von beiden eintritt, entscheidet dann über das weitere Vorgehen.

Es ist aus der Konstruktion des Algorithmus 4.6.1 klar ersichtlich, dass dieser, **solange die Wartezeit nie negativ wird**, einen kompetitiven Faktor von α garantiert. Allerdings wird bei der praktischen Anwendung eine effiziente und schnelle Möglichkeit für die Berechnung der Offline-Lösungen benötigt.

Satz 4.6.2 (Kompetitiver Faktor für Algorithmus 4.6.1).

Für Werte α mit $\alpha \geq \frac{1}{4}(1 + \sqrt{17})$ ist der Algorithmus 4.6.1 tatsächlich genau strikt α -kompetitiv. Zudem kann allgemein kein besserer strikt kompetitiver Faktor als

$$c = \frac{1}{4}(1 + \sqrt{17})$$

für deterministische Algorithmen gefunden werden.

Beweis zu Satz 4.6.2 nach [Kru02, Theorem 3.13 und Theorem 3.10].

Es sind folgende Aussagen zu beweisen:

- (i) $W \geq 0$ für alle $\alpha \geq \frac{1}{4}(1 + \sqrt{17})$.
- (ii) $c \geq \frac{1}{4}(1 + \sqrt{17})$ für alle deterministischen Algorithmen.

Zu (ii): Sei $\sigma_1 = (0 \ 1)$ die erste Anfrage und T der Zeitpunkt, zu welchem der Algorithmus wieder zum Nullpunkt zurückgekehrt ist. Damit der Algorithmus c -kompetitiv sein kann muss $T \leq cT_{\text{Off}} = 2c$ gelten. Zum Zeitpunkt T werde eine zweite Anfrage $\sigma_2 = (T \ 1)$ bekannt. Dann braucht der Algorithmus mindestens $T_{\text{Alg}} \geq T + 2$ Zeit zur Bewältigung der ersten beiden Anfragen, dagegen werden von der Offline-Bearbeitung nur $T + 1$ Zeiteinheiten benötigt. Daraus ergibt sich zusammen mit der Tatsache $T \leq 2c$ und dem Korollar 4.1.1:

$$c \geq \frac{T_{\text{Alg}}}{T_{\text{Off}}} \geq \frac{T + 2}{T + 1} \stackrel{\text{nach Korollar 4.1.1}}{\geq} \frac{2c + 2}{2c + 1} = 1 + \frac{1}{2c + 1} \quad (4.6.1)$$

Wird nun (4.6.1) mit dem Hauptnenner multipliziert und alles auf eine Seite gebracht ergibt sich folgendes:

$$2c^2 + c \geq 2c + 1 + 1 \quad \Rightarrow \quad 2c^2 - c - 2 \geq 0 \quad (4.6.2)$$

Die Auflösung der quadratischen Gleichung aus (4.6.2) liefert

$$c = \frac{1}{4}(1 + \sqrt{17})$$

als einzige zulässige Lösung. Damit ist also gezeigt, dass jeder deterministische Algorithmus mindestens einen kompetitiven Faktor von $\frac{1}{4}(1 + \sqrt{17})$ besitzt.

Zu (i): Der Beweis, dass W im Algorithmus 4.6.1 für $\alpha \geq \frac{1}{4}(1 + \sqrt{17})$ nicht negativ ist, erfolgt per Induktion über die Anzahl m der Anfragen.

Induktionsanfang ($m = 0$):

Mit keiner Anfrage ist $W \geq 0$ trivialerweise erfüllt.

Induktionsbehauptung ($m - 1$):

Für $m - 1$ Anfragen seien die Wartezeiten immer nicht negativ.

Induktionsschritt ($m - 1 \rightarrow m$):

Sei $\sigma = \{\sigma_1, \dots, \sigma_m\}$ eine Sequenz von Anfragen und t_m der Zeitpunkt der letzten Anfrage, sowie $\sigma_m = (t_m \ x)$ die Anfrage aus $\sigma_{=t_m}$ mit der größten Entfernung zum Ausgangspunkt. Falls $t_m = 0$ ist, so vereinfacht sich die Problemstellung im wesentlichen auf den Induktionsanfang mit der Anfrage σ_m als einzig relevante Anfrage, alle anderen werden dann im Zuge der Bearbeitung dieser Anfrage auch erledigt.

Sei also ohne Beschränkung der Allgemeinheit $t_m > 0$, weiterhin seien $s(t_m)$ und $s^*(t_m)$ die aktuellen Positionen des Algorithmus 4.6.1 und der Offline-Bearbeitung zum Zeitpunkt t_m . Ferner sei f die Koordinate der größten von Algorithmus 4.6.1

noch nicht bearbeiteten Anfrage in der Sequenz $\sigma_{<t_m}$ und F die allgemein größte Koordinate in $\sigma_{<t_m}$. Also gilt nach der Definition $f \leq F$.

Für die Lage von x innerhalb der positiven Halbgeraden gibt es nun drei Fälle: Entweder ist x kleiner gleich f oder x liegt zwischen f und F oder aber x ist größer gleich F .

- (a) Der erste Fall mit $x \leq f$ bewirkt für den Algorithmus 4.6.1 keine Veränderung gegenüber der Sequenz $\sigma_1, \dots, \sigma_{m-1}$ da dieser noch die Anfrage mit $f \geq x$ bearbeiten muss. Nach der Induktionsbehauptung sind die Wartezeiten für die Sequenz mit $m - 1$ Elementen positiv und da sich die Gesamtdauer der Offline-Bearbeitung durch die Hinzunahme einer Anfrage nicht verringern kann, bleibt diese Eigenschaft der Wartezeiten erhalten.
- (b) Der zweite Fall mit $f \leq x < F$ gestaltet sich etwas schwieriger. Falls allerdings $s(t_m)$ größer als x ist greift wieder die gleiche Argumentation wie im oberen Fall, da sich für den Algorithmus 4.6.1 nichts verändert und die Gesamtdauer der Offline-Bearbeitung auch wieder höchstens größer werden kann. Sei also $x > s(t_m)$, dann muss sich der Algorithmus 4.6.1 von $s(t_m)$ nach x bewegen und kommt dort zum Zeitpunkt $t_m + d(s(t_m), x)$ an. Dabei entspricht $d(z, y)$ der Entfernung zwischen den beiden Punkten z und y , es gilt also

$$d(z, y) = |z - y|.$$

Die Wartezeit W ergibt sich dann zu

$$W = \alpha T_{\text{Off}}(\sigma) - [t_m + d(s(t_m), x) + x]$$

Mit der Tatsache, dass die Offline-Bearbeitung auch mindestens $t_m + x$ Zeiteinheiten benötigt bevor diese fertig werden kann, ergibt sich eine Unterschranke für die Wartezeit von:

$$W \geq (\alpha - 1)T_{\text{Off}}(\sigma) - d(s(t_m), x) \quad (4.6.3)$$

Sei t' der Zeitpunkt zu dem der Algorithmus 4.6.1 die Anfrage für F bearbeitet hat, da $F > f$ in diesem Fall vorausgesetzt ist, gilt:

$$t_m \geq t' + d(s(t_m), F) \quad T_{\text{Off}}(\sigma_{\leq t'}) \geq 2F \quad (4.6.4)$$

Die Induktionsbehauptung angewandt auf t' liefert

$$t' + F = \alpha T_{\text{Off}}(\sigma_{\leq t'}) \quad (4.6.5)$$

Aus (4.6.4) und (4.6.5) folgt:

$$t_m \geq \alpha T_{\text{Off}}(\sigma_{\leq t'}) - F + d(s(t_m), F) \geq (2\alpha - 1)F + d(s(t_m), F) \quad (4.6.6)$$

Mit $d(s(t_m), x) \leq d(s(t_m), F) \leq F$ ergibt sich aus (4.6.6) weiterhin:

$$t_m \geq (2\alpha - 1)d(s(t_m), x) + d(s(t_m), x) = 2\alpha d(s(t_m), x) \quad (4.6.7)$$

Da $T_{\text{Off}}(\sigma)$ auf jeden Fall größer gleich $t_m + x$ ist, folgt daraus mit (4.6.7)

$$T_{\text{Off}}(\sigma) \geq 2\alpha d(s(t_m), x) + x \geq (2\alpha + 1)d(s(t_m), x). \quad (4.6.8)$$

Die Abschätzung aus (4.6.8) eingesetzt in (4.6.3) liefert

$$\begin{aligned} W &\geq (\alpha - 1)(2\alpha + 1)d(s(t_m), x) - d(s(t_m), x) = (2\alpha^2 - \alpha - 2)d(s(t_m), x) \\ &= \left(\frac{1}{4}(9 + \sqrt{17}) - \frac{1}{4}(1 + \sqrt{17}) - 2 \right) d(s(t_m), x) = 0 \end{aligned} \quad (4.6.9)$$

- (c) Der dritte Fall mit $x \geq F$ gestaltet sich ähnlich wie der zweite aber mit ein paar Unterschieden:

Wenn $x \geq F$ gilt, so wird sich der Algorithmus 4.6.1 nach dem Zeitpunkt t_m in einer Bewegung nach rechts zu x befinden und dort folgende Wartezeit berechnen:

$$W^* = \alpha T_{\text{Off}}(\sigma_{\leq t_m}) - (t_m + d(s(t_m), x) + x) \quad (4.6.10)$$

Zu zeigen bleibt also nur noch, dass W^* nicht negativ ist. Da die Offline-Bearbeitung fair ist, gilt für deren optimale Lösung zur Sequenz der Anfragen $\sigma_{\leq t_m}$:

$$T_{\text{Off}}(\sigma_{\leq t_m}) \geq t_m + d(s^*(t_m), x) + x \geq t_m + d(F, x) + x \quad (4.6.11)$$

Diese Abschätzung aus (4.6.11) in die Berechnung der Wartezeit aus (4.6.10) eingesetzt, führt zu

$$\begin{aligned} W^* &= (\alpha - 1)T_{\text{Off}}(\sigma_{\leq t_m}) + T_{\text{Off}}(\sigma_{\leq t_m}) - (t_m + d(s(t_m), x) + x) \\ &\geq (\alpha - 1)T_{\text{Off}}(\sigma_{\leq t_m}) + d(F, x) - d(s(t_m), x) \\ &= (\alpha - 1)T_{\text{Off}}(\sigma_{\leq t_m}) - d(s(t_m), F). \end{aligned} \quad (4.6.12)$$

Sei dann wieder t' der Zeitpunkt der Rückkehr des Algorithmus 4.6.1 von F mit der Sequenz von Anfragen $\sigma_{< t_m}$ als Grundlage, allerdings können hier nun zwei unterschiedliche Situationen eintreten.

- es gelte $t' \leq t_m$. Dies entspricht dem zweiten Fall und aus dessen Induktionsbehauptung folgt daher:

$$t' + F = \alpha T_{\text{Off}}(\sigma_{\leq t'}) \quad (4.6.13a)$$

$$t_m \geq t' + d(F, s(t_m)) \quad (4.6.13b)$$

$$T_{\text{Off}}(\sigma_{\leq t'}) \geq 2F \quad (4.6.13c)$$

Unter Verwendung der drei Abschätzungen aus (4.6.13) ergibt sich für t_m die folgende Abschätzung:

$$\begin{aligned} t &\stackrel{(4.6.13b)}{\geq} t' + d(s(t_m), F) \stackrel{(4.6.13a)}{=} \alpha T_{\text{Off}}(\sigma_{\leq t'}) - F + d(s(t_m), F) \\ &\stackrel{(4.6.13c)}{\geq} 2\alpha F - F + d(s(t_m), F) = (2\alpha - 1)F + d(s(t_m), F) \end{aligned} \quad (4.6.14)$$

Mit der Abschätzung für das Ergebnis der Offline-Bearbeitung aus (4.6.11) und dem Ersetzen von t_m durch (4.6.14) sowie mit $F \geq d(s(t_m), F)$ ergibt sich:

$$\begin{aligned} T_{\text{Off}}(\sigma_{\leq t_m}) &\geq 2\alpha d(s(t_m), F) + d(F, x) + x \\ &\geq (2\alpha + 1)d(s(t_m), F) \end{aligned} \quad (4.6.15)$$

Eingesetzt in die Berechnung der Wartezeit aus (4.6.12) liefert (4.6.15) das gewünschte Ergebnis, welches auch schon in (4.6.9), also im zweiten Fall, erreicht wurde:

$$\begin{aligned} W^* &\geq (\alpha - 1)(2\alpha + 1)d(s(t_m), F) - d(s(t_m), F) \\ &= (2\alpha^2 - \alpha - 2)d(s(t_m), F) = 0 \end{aligned}$$

- Gilt allerdings $t' > t_m$, dann folgt sofort, dass

$$t' \geq t_m + d(s(t_m), F) \quad (4.6.16)$$

gilt. Eine Umformulierung der ersten Abschätzung zur Gesamtdauer der Offline-Bearbeitung aus (4.6.11) ergibt:

$$\begin{aligned} T_{\text{Off}}(\sigma_{\leq t_m}) &\geq t_m + d(s^*(t_m), x) + x \\ &= t_m + d(s^*(t_m), F) + F + 2d(F, x) \end{aligned} \quad (4.6.17)$$

Mit der Tatsache, dass

$$T_{\text{Off}}(\sigma_{< t_m}) \leq t_m + d(s^*(t_m), F) + F \quad (4.6.18)$$

ist, kann ein direkter Bezug zwischen den beiden Abschätzungen zu der Gesamtdauer der Offline-Bearbeitung aus (4.6.17) und (4.6.18) hergestellt werden:

$$T_{\text{Off}}(\sigma_{\leq t_m}) \geq T_{\text{Off}}(\sigma_{< t_m}) + 2d(F, x) \quad (4.6.19)$$

Die Abschätzung (4.6.19) eingesetzt in die Wartezeit aus (4.6.10) bewirkt:

$$\begin{aligned} W^* &\geq \alpha(T_{\text{Off}}(\sigma_{<t_m}) + 2d(F, x)) - (x + t_m + d(s(t_m), x)) \\ &= \alpha T_{\text{Off}}(\sigma_{<t_m}) + \alpha 2d(F, x) - [F + 2d(F, x) + t_m + d(s(t_m), F)] \\ &\geq \alpha T_{\text{Off}}(\sigma_{<t_m}) - (t' + F) + (\alpha - 1)2d(F, x) \quad (4.6.20) \end{aligned}$$

Die letzte Ungleichung ergibt sich dabei aus (4.6.16).

Da der erste Teil $\alpha T_{\text{Off}}(\sigma_{<t_m}) - (t' + F)$ der letzten Abschätzung aus (4.6.20) nach der Induktionsbehauptung nicht negativ ist und der zweite Teil $(\alpha - 1)2d(F, x)$ für alle $\alpha \geq 1$ ebenfalls nicht negativ ist, ist damit auch die neue Wartezeit nicht negativ.

Damit ist der Beweis dieses Satzes abgeschlossen. □

4.6.2. Dial-A-Ride

Beim in der Bemerkung 2.2.2 erwähnten dial-a-ride-Problem geht es darum, von einem Ausgangspunkt aus, Transportaufträge mit Abholpunkten und Lieferpunkten abzuarbeiten und danach zum Ausgangspunkt zurückzukehren. Damit fällt diese Problemstellung unter die in Optimierungsproblem 2.2.4 zusammengefassten Transportprobleme.

Zusätzlich ist beim Transport meistens noch eine gewisse Kapazitätsbeschränkung des Transportmittels zu beachten, das heißt, dass immer nur eine bestimmte Anzahl von Aufträgen auf einmal transportiert werden kann. Ein Abbruch eines einmal begonnenen Auftrags ist in der Regel nicht möglich, das Transportmittel darf meist nur im jeweiligen Lieferpunkt entladen werden. Im weiteren Verlauf dieses Unterabschnitts ist außerdem die Kapazität des Transportmittels so ausgelegt, dass immer genau ein Auftrag auf einmal bearbeitet werden kann.

Bei der Online-Bearbeitung ist die Sequenz der Transportaufträge unbekannt, einzig die bis zum aktuellen Zeitpunkt eingegangenen Aufträge sind bekannt. Um diese bei der Online-Bearbeitung besser in den Griff zu kriegen, können zum Beispiel innerhalb der Sequenz mehrere kleinere Rundreisen zur Abarbeitung von Aufträgen gebildet werden.

Unter den hier gegebenen Vorgaben, besteht die Rundreise aus einer Sortierung der aktuell zu bearbeitenden Anfragen, wobei die Rundreise ausgehend vom Ausgangspunkt den Abholpunkt der in dieser Sortierung ersten Anfrage besucht und sich von diesem direkt zum Lieferpunkt dieser Anfrage aufmacht. Danach werden sukzessive nacheinander der Abholpunkt, sowie anschließend der jeweilige dazugehörige Lieferpunkt, der nächsten Anfrage der Rundreise in der vorgegebenen Sortierung angefahren. Minimiert werden soll schließlich die zwischen der Bekanntgabe des ersten Auftrages und der endgültigen Rückkehr zum Ausgangspunkt verstrichene Zeit.

Da das dial-a-ride-Problem eine Erweiterung des allgemeine Rundreise-Problems (siehe Optimierungsproblem 2.2.1) ist, kann aus der Aussage in Satz 7.3.9 zum kompetitiven

Faktor des Rundreise-Problem auf dem Einheitskreis oder auch aus [AFL⁺01] eine Unterschranke für den kompetitiven Faktor des dial-a-ride-Problems getroffen werden:

Lemma 4.6.3 (Optimaler kompetitiver Faktor).

Jeder Online-Algorithmus für das Problem dial-a-ride hat, über die komplette Problemstellung der Rundreise betrachtet, mindestens einen kompetitiven Faktor von zwei.

Beweis zu Lemma 4.6.3.

Bei einer Gleichsetzung des Abholpunktes und des Lieferpunktes bei jeder Anfrage im dial-a-ride-Problem ergibt sich ein Rundreise-Problem. Also ist jedes Rundreise-Problem ein dial-a-ride-Problem, bei welchem in allen Anfragen die beiden relevanten Punkte übereinstimmen.

Da in dieser Arbeit für einen Spezialfall des Rundreise-Problems nach Satz 7.3.9 ein minimaler kompetitiver Faktor von mindestens zwei nachgewiesen wurde, kann auch die größere Problemklasse der dial-a-ride-Probleme, bei einer Betrachtung über alle möglichen Grundmengen, keinen besseren kompetitiven Faktor haben. $\square$

Definition 4.6.4 (Zustände der Bearbeitung eines Algorithmus für dial-a-ride-Probleme).

Während der Online-Bearbeitung eines dial-a-ride-Problems gibt es drei Zustände in welchen sich ein Algorithmus gerade befinden kann:

Abgearbeitet: Beim Zustand *Abgearbeitet* befindet sich der Algorithmus im ursprünglichen Ausgangspunkt und alle aktuell bekannten Anfragen wurden bereits bearbeitet.

Aktivität: Während des Zustands *Aktivität* bearbeitet der Algorithmus aktuell gerade eine Route.

Warteschleife: Während der *Warteschleife* ist der Algorithmus wieder im ursprünglichen Ausgangspunkt, und es gibt unbearbeitete Anfragen. Allerdings werden diese unbearbeiteten Anfragen bis zu einem vorher bestimmten Ende der Warteschleife nicht bearbeitet.

Krumke stellt in seiner Habilitationsschrift [Kru02, Abschnitt 2.6] den Algorithmus „Smart-Start“ als optimalen Algorithmus vor. Anders als beim Algorithmus 4.6.1 ist Smart-Start nicht darauf ausgelegt direkt einen vorgegebenen strikt kompetitiven Faktor c einzuhalten, sondern es gibt einen vorgegebenen Wartefaktor – es wäre vielleicht angebrachter hier von einem Größenfaktor zu sprechen – θ . Mit diesem Faktor wird festgelegt, ab welcher Größenordnung der Zeitdauer für die neue Rundreise gegenüber der bis dahin vergangenen Zeit eine Wartephase eingelegt werden soll.

Das bedeutet konkret, dass falls die aktuelle Zeit plus der Dauer für die neue Rundreise größer als die aktuelle Zeit mal den Wartefaktor ist, dann ist die neue Rundreise

im Vergleich zur bisherigen Rundreise zu lang. Dadurch würde sich eine ungünstige Bearbeitung, die durch diese neue Rundreise eventuell herbeigeführt würde im Vergleich zur bisherigen Rundreise zu stark auswirken.

Als Resultat wird daraufhin erst einmal eine Zeit lang abgewartet und weitere Anfragen gesammelt, bis die Wartezeit, welche sich aus der Differenz zwischen der aktuellen Zeit multipliziert mit dem Faktor $\theta - 1$ und dem Aufwand der geplanten, dann aber doch nicht gleich gestarteten, neuen Rundreise ergibt, verstrichen ist. Die damit verbundene Hoffnung ist, wie schon in Definition 3.1.6 dargelegt, dass sich durch die während dem Warten noch neu ankommenden Anfragen die Effizienz der neuen Rundreise verbessert.

Algorithmus 4.6.5 („Smart-Start“ aus [KR02, Abschnitt 9.2]).

Der Algorithmus behandelt neue Anfragen abhängig davon, in welchem der drei Zustände aus Definition 4.6.4 er sich in diesem Moment befindet.

- *Falls die alten Anfragen schon alle abgearbeitet wurden, so muss entschieden werden, ob entweder gewartet wird oder sofort mit der weiteren Bearbeitung begonnen wird.*
- *Falls momentan eine Rundreise bearbeitet wird, so werden neue Anfragen solange ignoriert, bis diese Rundreise fertig bearbeitet ist.*
- *Falls gerade eine Wartephase ist, so wird bis zu deren Ende ausgeharrt, und die neuen Anfragen erst einmal ignoriert.*

Sowohl nach jeder abgeschlossenen Bearbeitung einer Rundreise, als auch nach dem Ende einer Wartephase muss anhand der Daten aller bis dahin noch nicht bearbeiteten aber schon bekannten Anfragen eine Entscheidung getroffen werden, ob gleich mit der Bearbeitung einer neuen Rundreise begonnen wird oder eine Wartephase begonnen wird.

Bemerkung 4.6.6 (Entscheidung zwischen Warteschleife und Start einer Rundreise).

Bei der Entscheidung zwischen Warteschleife und sofortiger Bearbeitung berechnet der Algorithmus 4.6.5 zuerst durch einen Algorithmus zur Approximation mit Güte ρ eine Rundreise S für die noch nicht bearbeiteten Anfragen mit gesamten Zeitaufwand $\ell(S)$. Dabei hat der Algorithmus zur Approximation genau dann die Güte $\rho \geq 1$, wenn die Länge der durch ihn geplanten Rundreise nicht größer ist als ρ mal die Länge der optimalen Rundreise.

Mit diesem Zeitaufwand $\ell(S)$ wird dann der erste Zeitpunkt t' ausgewählt, zu welchem er mit der Bearbeitung der noch offenen Anfragen beginnen kann, so dass maximal die durch den vordefinierten Wartefaktor θ mögliche Wartezeit abgewartet wird. Das bedeutet, dass t' derjenige erste Zeitpunkt mit $t' \geq t$ ist, für welchen

$$t' + \ell(S) \leq \theta t'$$

gilt. Das führt dazu, dass für t' sowohl $t' \geq \frac{\ell(S)}{\theta-1}$ als auch $t' \geq t$ gelten muss, diese beiden Abschätzungen zusammen mit der Tatsache, dass t' der erste Zeitpunkt nach t ist der

diese erfüllt, ergeben für die Berechnung von t' :

$$t' = \max \left\{ \frac{\ell(S)}{\theta - 1}, t \right\} \quad (4.6.21)$$

Durch (4.6.21) wird aufgezeigt, dass der Algorithmus 4.6.5 nur dann in die Warteschleife geht, wenn der durch die neu geplante Rundreise S neu hinzu kommende Zeitaufwand $\ell(S)$ größer als der bereits zum aktuellen Zeitpunkt benötigte Zeitaufwand mal den um eins verringerten Wartefaktor θ ist.

Falls $t' = t$ gilt, so beginnt der Algorithmus 4.6.5 unmittelbar mit der Bearbeitung der vorher gefundenen Rundreise für S . Im zweiten Fall mit $t' > t$ geht er bis zum Zeitpunkt t' in die Warteschleife.

Satz 4.6.7 (Kompetitiver Faktor zu Algorithmus 4.6.5).

Der Algorithmus 4.6.5 ist strikt $c = \max\{\theta, \rho \cdot (1 + \frac{1}{\theta-1})\}$, $\frac{\theta}{2} + \rho$ kompetitiv.

Als direkte Folgerung ergibt sich aus Satz 4.6.7 für $\theta = \frac{1}{2}(1 + \sqrt{1 + 8\rho})$ ein minimaler kompetitiver Faktor von

$$\begin{aligned} c &= \max \left\{ \frac{1}{2}(1 + \sqrt{1 + 8\rho}), \rho \left(1 + \frac{2}{(1 + \sqrt{1 + 8\rho}) - 2} \right), \frac{1}{4}(1 + \sqrt{1 + 8\rho}) + \rho \right\} \\ &\stackrel{\text{mit } \rho \geq 1}{=} \max \left\{ \rho \left(1 + \frac{2}{(1 + \sqrt{1 + 8\rho}) - 2} \right), \frac{1}{4}(1 + \sqrt{1 + 8\rho}) + \rho \right\} \\ &= \frac{1}{4}(1 + \sqrt{1 + 8\rho}) + \rho. \quad (4.6.22) \end{aligned}$$

Wenn zur Lösung der jeweiligen Rundreise für die aktuell bekannten Anfragen ein exaktes Verfahren, also $\rho = 1$ verwendet wird, ergibt sich aus (4.6.22) ein kompetitiver Faktor von zwei. Damit ist der Algorithmus 4.6.5 schon der optimale Online-Algorithmus für das dial-a-ride-Problem.

Beweis zu Satz 4.6.7 (dem Beweis zu [KR02, Satz 9.4] folgend).

Die induktiv vorgehende Analyse des Algorithmus 4.6.5 bezieht sich ebenfalls auf die in Definition 4.6.4 definierten Zustände der Bearbeitung. Sei dazu t_m der Zeitpunkt der letzten Anfrage aus einer Sequenz von Anfragen σ , die Menge der Anfragen zum Zeitpunkt t_m wird durch $\sigma_{=t_m}$ gekennzeichnet.

Fall 1: Der Algorithmus 4.6.5 habe bereits alle anderen Anfragen abgearbeitet. Dann wird zuerst mit dem Algorithmus zur Approximation für $\sigma_{=t_m}$ eine Rundreise S mit Länge $\ell(S)$ berechnet, für welche durch die Güte ρ der Approximation folgende Abschätzung der Länge gilt:

$$\ell(S) \leq \rho T_{\text{Off}}(t_m, 0, \sigma_{=t_m})$$

Dabei wird mit $T_{\text{Off}}(t_m, 0, \sigma_{=t_m})$ das beste mögliche Ergebnis der Offline-Bearbeitung für die Anfragen aus $\sigma_{=t_m}$ zum Zeitpunkt t_m mit Startpunkt 0 bezeichnet.

Nach der Berechnung der Rundreise S wird der Zeitpunkt t' gesucht, für welchen

$$t' = \min\{t \geq t_m : t + \ell(S) \leq \theta t\}$$

gilt.

- Falls nun $t' = t_m$ gilt, so steht fest, dass die Bearbeitung der ganzen Sequenz σ durch den Algorithmus 4.6.5 nicht länger als

$$T_{\text{Smart}}(\sigma) \leq \theta t_m \leq \theta T_{\text{Off}}(\sigma)$$

dauern wird.

- Gilt allerdings $t' > t_m$ so wird das ganze etwas komplizierter. Wegen der Güte der Approximation ρ gilt für die Länge der Rundreise S :

$$T_{\text{Off}}(\sigma) \geq T_{\text{Off}}(\sigma_{=t_m}) \geq \frac{\ell(S)}{\rho} = \frac{\theta - 1}{\rho} t' \quad (4.6.23)$$

Wird nun (4.6.23) nach t' aufgelöst und die dadurch entstehende Abschätzung in $T_{\text{Smart}}(\sigma) = t' + \ell(S) \leq \theta \cdot t'$ eingesetzt so wird der erste Teil des kompetitiven Faktors aus Satz 4.6.7 erreicht:

$$T_{\text{Smart}}(\sigma) \leq \frac{\theta}{\theta - 1} \rho T_{\text{Off}}(\sigma) = \rho \left(1 + \frac{1}{\theta - 1}\right) T_{\text{Off}}(\sigma)$$

Fall 2: Wenn sich der Algorithmus 4.6.5 zum Zeitpunkt t_m in der Warteschleife befindet, so verbleibt er solange in diesem Zustand, bis der Zeitpunkt t' mit

$$t' = \min\{t \geq t_m : t + \ell(S) \leq \theta t\}$$

erreicht wird, dabei ist diesmal S die durch die Approximation gefundene Rundreise für alle zum Zeitpunkt t_m noch nicht bearbeiteten Anfragen. Ein analoges Vorgehen wie bei der kompletten Bearbeitung aller vorherigen Anfragen, liefert dann das gewünschte Ergebnis.

Fall 3: Falls der Algorithmus 4.6.5 zum Zeitpunkt t_m bereits eine Rundreise S bearbeitet, so beendet er zuerst diese und plant danach für die noch nicht bearbeiteten Anfragen eine neue Rundreise. Trifft er dabei die Entscheidung in die Warteschleife zu gehen, so tritt wieder einer der beiden anderen Fälle ein. Ansonsten wird eine neue Rundreise S' für die zu diesem Zeitpunkt noch nicht bearbeiteten Anfragen begonnen.

Sei also t_S der Startzeitpunkt für die aktuelle Rundreise S und $\sigma_{t>t_S}$ die Menge der während der Bearbeitung von S angekommenen Anfragen. Die von S' zu bearbeitenden Anfragen entsprechen dann den in $\sigma_{t>t_S}$ enthaltenen Anfragen. Also gilt für die Bearbeitungszeit T_{Smart} des Algorithmus 4.6.5 folgende Beziehung:

$$T_{Smart}(\sigma) = t_S + \ell(S) + \ell(S') \quad (4.6.24)$$

Durch den Aufbau des Algorithmus 4.6.5 muss dann auch

$$t_S + \ell(S) \leq \theta t_S \quad (4.6.25)$$

gelten. Sei σ_f die erste Anfrage aus $\sigma_{t>t_S}$, welche bei der Offline-Bearbeitung bearbeitet wird, und α_f deren zugehöriger Abholpunkt. Daraus folgt dann:

$$T_{Off}(\sigma) \geq t_S + T_{Off}(t_m, \alpha_f, \sigma_{t>t_S}) \geq t_S + d(0, \alpha_f) \quad (4.6.26)$$

Dann gilt wegen der Güte der Approximation ρ für die Berechnung der Rundreisen

$$\ell(S') \leq \rho [d(0, \alpha_f) + T_{Off}(t_m, \alpha_f, \sigma_{t>t_S})] \stackrel{(4.6.26)}{\leq} \rho [d(0, \alpha_f) + T_{Off}(\sigma) - t_S]. \quad (4.6.27)$$

Werden nun die beiden Formeln (4.6.27) und (4.6.25) in (4.6.24) eingesetzt, so ergibt sich folgende Kette von Abschätzungen:

$$\begin{aligned} T_{Smart}(\sigma) &\stackrel{(4.6.24), (4.6.25)}{\leq} \ell(S') + \theta t_S \stackrel{(4.6.27)}{\leq} (\theta - \rho)t_S + \rho d(0, \alpha_f) + \rho T_{Off}(\sigma) \\ &\stackrel{(4.6.26)}{\leq} \theta T_{Off}(\sigma) + (2\rho - \theta)d(0, \alpha_f) \leq \begin{cases} \theta T_{Off}(\sigma) + (2\rho - \theta) \cdot \frac{T_{Off}(\sigma)}{2} & \text{für } \theta \leq 2\rho \\ \theta T_{Off}(\sigma) & \text{für } \theta > 2\rho \end{cases} \\ &\leq \max \left\{ \frac{\theta}{2} + \rho, \theta \right\} T_{Off}(\sigma) \end{aligned}$$

Damit ist die Aussage des Satzes bewiesen. $\square$

Teil II.

Informationsvorlauf in der Online- Optimierung

Überblick

Innerhalb dieses Teils wird das Konzept Informationsvorlauf vorgestellt und auf sich daraus ergebende Effekte hingewiesen.

Das Kapitel 5 definiert die für das Verständnis des Informationsvorlaufs notwendigen Begriffe und die verschiedenen möglichen Konzepte.

Kapitel 6 geht schließlich auf die unterschiedlichen Wege und Möglichkeiten ein, den im vorherigen Kapitel definierten Informationsvorlauf tatsächlich zu verwenden.

5. Grundlegendes zum Informationsvorlauf

Grundsätzlich kennt bei der Online-Optimierung der zugehörige Online-Algorithmus bei der Bearbeitung der Sequenz aus Anfragen immer nur die Ausprägungen der bisher bearbeiteten Anfragen, sowie der aktuellen Anfrage. In einigen Problemstellungen kommt es zudem auch vor, dass der Online-Algorithmus zeitweise zusätzlich einige wenige weitere Anfragen mit deren Ausprägungen kennt.

Aufgrund dieser gegenüber der Offline-Bearbeitung, welche immer die gesamte Sequenz der Anfragen kennt, ziemlich spärlichen Informationsmenge, unterscheidet sich die vom Online-Algorithmus gefundene Lösung und der zugehörige Wert der Zielfunktion meist deutlich von dem der Offline-Bearbeitung.

Um diesen Nachteil etwas auszugleichen, soll nun mit dem Informationsvorlauf eine systematische Möglichkeit vorgestellt werden, wie der Online-Algorithmus zu mehr Informationen, und damit auch zu besseren Möglichkeiten bei der Bearbeitung kommen kann.

Theoretisch ist es grundsätzlich auch möglich einen Informationsvorlauf etwa dadurch zu simulieren, dass die weitere Fortführung der Online-Bearbeitung nach dem Abschluss der Bearbeitung einer Anfrage solange hinauszögert wird, bis wieder Informationen über eine festgelegte Anzahl von Anfragen erhalten wurden.

Da allerdings bei der Online-Optimierung meist auch die Größe der Sequenz der Anfragen nicht bekannt ist und das Erreichen des Endes der Sequenz nicht unbedingt aufgezeigt wird, benötigt diese Methode eine Abschlussregel. Diese Abschlussregel muss festlegen unter welchen Umständen davon ausgegangen werden soll, dass das Ende der Sequenz der Anfragen erreicht wurde und welche Konsequenzen dies bei der Planung der Bearbeitung der aufgeschobenen Anfragen nach sich zieht.

5.1. Einführung

Um das in der Einleitung des Kapitels erwähnte Defizit bei der für die Planung der Bearbeitung bereitstehenden Information zu verringern, erhält der Online-Algorithmus einen so genannten Informationsvorlauf. Darunter ist im Grundsatz zu verstehen, dass der Algorithmus bei der Bearbeitung der aktuell eingetroffenen Anfrage nicht nur die Informationen über die bisher schon bearbeiteten Anfragen besitzt, sondern zusätzliche Informationen erhält:

- Entweder indem er über die Informationen hinaus, die er bei der eigentlichen Art und Weise der Problemstellung zur Verfügung hätte, weitere Kenntnis von zukünftigen Anfragen erhält,
- oder indem ihm gewissermaßen eine Garantie über die Größe der bei der Planung der Bearbeitung zur Verfügung stehenden Information gegeben wird.

5.2. Arten des Informationsvorlaufs

In diesen Abschnitt sollen die einzelnen verschiedenen Arten der Aufstellung eines Informationsvorlaufs vorgestellt werden. In Unterabschnitt 5.2.1 geht es um einen zeitlichen Ablauf, in Unterabschnitt 5.2.2 um eine Darstellung über die Reihenfolge. Ein etwas komplexeres Konzept findet sich schließlich in Unterabschnitt 5.2.3. Dabei wird bei dem jeweiligen Konzepten auch gleich auf eventuell vorhandene Schwächen beziehungsweise Schwierigkeiten bei der Analyse verwiesen.

5.2.1. Informationsvorlauf als Zeiteinheit

Die erste und eigentlich nächstliegende Möglichkeit für einen Informationsvorlauf ist die Definition eines Informationsvorlaufs über Zeiteinheiten, das heißt, dass dem Algorithmus als zusätzliche Information alle Anfragen eines bestimmten Zeitraums mitgeteilt werden.

Da in diesem Fall die Steuerung der Anfragen und deren Informationen über eine zeitliche Ordnung vorgenommen wird, ergibt sich folgende Definition zur Bekanntgabe der einzelnen Anfragen:

Definition 5.2.1 (Bekanntgabe der Anfragen bei Zeiteinheiten).

Das in Definition 1.1.1 angesprochene Ordnungsmerkmal M besteht in diesem Fall für jede Anfrage σ_i in einem Zeitpunkt t_i . Sobald bei der Bearbeitung der Problemstellung der Zeitpunkt t erreicht wird, werden alle Anfragen, deren Ordnungsmerkmal M dem Zeitpunkt t entspricht, dem Algorithmus bekannt gegeben.

Bei dieser Version des Informationsvorlaufs über einen vorgegebenen Zeitraum kommt es jedoch unter dem Aspekt einer Analyse zu einigen Nebeneffekten:

- Um einen zeitlichen Informationsvorlauf nutzen zu können, muss zusätzlich eine Verteilung für das zeitliche Auftreten der Anfragen definiert oder durch Untersuchungen gefunden werden. Dies führt dazu, dass bei einer theoretischen Analyse wie zum Beispiel der Average-Case-Analyse aus Unterabschnitt 1.2.2 nicht nur der Erwartungswert über die Verteilung der Anfragen genommen werden muss, sondern auch noch der zeitliche Ablauf dieser Anfragen zufällig ist. Ein und dieselbe Sequenz von Anfragen kann dann, je nachdem welche zeitlichen Abstände zwischen den einzelnen Anfragen liegen, zu sehr unterschiedlichen Ergebnissen führen.

- Durch die zeitliche Verteilung gibt es insbesondere auch keine Gewissheit über die tatsächlich durch den Informationsvorlauf erhaltenen Informationen. So kann es zum Beispiel bei einer zusätzlichen Information über einen bestimmten Zeitraum vorkommen, dass während diesem einmal nur drei Anfragen erkannt werden beziehungsweise bei etwas mehr Glück sogar zehn oder mehr Anfragen sichtbar sind, im schlimmsten Fall ist in diesem Zeitraum sogar überhaupt keine Anfrage enthalten.
- Bei der empirischen Untersuchung der Ergebnisse hingegen macht sich diese zufällige Verteilung der Zeitpunkte des Auftretens nicht wesentlich bemerkbar, außer in der für die Relevanz der Ergebnisse zu erhöhenden Anzahl der Sequenzen, die untersucht werden sollten.

5.2.2. Informationsvorlauf nach Reihenfolge

Bei dieser Art des Informationsvorlaufs wird dem Algorithmus die Garantie gegeben, dass bei der Planung des nächsten Schrittes immer eine konstante Anzahl an Anfragen mit ihren Ausprägungen als Information zur Verfügung steht. Dies führt dann dazu, dass sofort mit der Bearbeitung einer Anfrage gleich die nächste Anfrage aus der Sequenz bekannt gegeben werden muss oder falls keine weitere Anfrage mehr vorhanden ist, angezeigt wird, dass das Ende der Sequenz der Anfragen erreicht wurde. Wenn diese Art des Informationsvorlaufs angewandt wird, so muss die Definition zur Bekanntgabe der Anfrage den Umständen entsprechend wie folgt angepasst werden:

Definition 5.2.2 (Bekanntgabe der Anfragen bei Reihenfolge).

Das in Definition 1.1.1 angesprochene Ordnungsmerkmal M besteht in diesem Fall für jede Anfrage σ_i ganz einfach aus deren Position i in der Sequenz der Anfragen. Die Anfrage σ_i wird dabei dann bekannt gegeben, wenn nach der Bekanntgabe der davor liegenden Anfrage σ_{i-1} eine Anfrage bearbeitet wurde. Das kann einerseits durch die direkte Bearbeitung von σ_{i-1} ausgelöst werden oder durch die Bearbeitung einer schon vorher bekannten Anfrage, welche bis dahin noch nicht bearbeitet worden ist.

Bei dieser Methodik wird quasi von allen zeitlichen oder sonstigen anderen Abläufen abstrahiert hin zu einer Behandlung der Anfragen aus der Sequenz rein nach der Reihenfolge in der diese in der Sequenz angeordnet sind. Durch diese Abstraktion ist es bei dieser Art des Informationsvorlaufs nicht ganz so einfach, passende praktische Beispiele zu finden. Dieses Konzept bietet sich allerdings natürlicherweise immer dann an, wenn tatsächlich eine Anfrage nach der anderen auftaucht oder wenn sich bei der Modellierung des Optimierungsproblems herausstellt, dass aufgrund von vorgegebenen Voraussetzungen nur eine maximale Anzahl von Anfragen für die Planung oder Auswahl zur Verfügung gestellt werden kann.

5.2.3. Informationsvorlauf nach einer Bearbeitungsgröße

Als weitere Variante für einen Informationsvorlaufs gibt es noch die Möglichkeit eine Voraussicht über eine vorher festgelegte Bearbeitungsgröße zu definieren. Als Informationsvorlauf werden dann die Informationen über so viele Anfragen verwendet, bis deren Summe über die Ausprägungen der festgelegten Bearbeitungsgröße erstmals einen vorher festgelegten maximalen Wert überschreitet. Das könnte zum Beispiel bei einem Transportband dessen Länge und damit die Kapazität für die Ablage der zu bearbeiteten Objekte sein, der Informationsvorlauf würde dann Informationen zu so vielen Objekten liefern, bis das Band überfüllt ist. Natürlich kann das gleiche auch für die Bearbeitungszeit eines Auftrags oder für ähnliche andere Merkmale definiert werden.

Definition 5.2.3 (Bekanntgabe der Anfragen bei Bearbeitungsgröße).

Das in Definition 1.1.1 angesprochene Ordnungsmerkmal M besteht hier für jede Anfrage σ_i darin, ob diese mit den bereits bekannten, aber noch nicht bearbeiteten Anfragen die vor dieser in der Sequenz der Anfragen liegen, in der Summe den maximalen Wert unterschreitet oder zum ersten Mal überschreitet. Für diese Art des Informationsvorlaufs ist daher für die Anordnung der Anfragen in der Sequenz schon eine vorgegebene von der ausgewählten Bearbeitungsgröße unabhängige Sortierung notwendig.

Dabei gilt hier, dass immer dann neue Anfragen bekannt gegeben werden, wenn durch die Bearbeitung einer bekannten Anfrage der maximale Wert der Summe über die Ausprägungen der noch nicht bearbeiteten Anfragen in der ausgewählten Bearbeitungsgröße wieder unterschritten wird. Danach werden solange neue Anfragen bekannt gegeben, bis die Summe diesen maximalen Wert wieder überschreitet.

Für diese Form des Informationsvorlaufs ist es notwendig, dass die verwendete Bearbeitungsgröße der jeweiligen Anfragen zufällig bestimmt wird, falls stattdessen nämlich alle Anfragen in dieser Bearbeitungsgröße den gleichen Wert besitzen, dann entspricht dies, wie in Unterabschnitt 5.2.2, einem Informationsvorlauf mit einer festen Anzahl von bekannten Anfragen. Allerdings ergeben sich auch hier wie in Unterabschnitt 5.2.1 einige nicht kalkulierbaren Aspekte:

- Die Anzahl der Anfragen, welche benötigt werden um den maximalen Wert zu überschreiten, ist natürlicher Weise abhängig von der zufällig erzeugten Sequenz der Anfragen und damit nicht als fester Wert kalkulierbar.
- Je nachdem welche Anfrage aus einem so festgelegten Informationsvorlauf ausgewählt wird, und wie groß die Ausprägung der verwendeten Größe der nächsten noch nicht bekannten Anfrage ist, ergibt sich eventuell in der Folge ein ganz unterschiedlicher neuer Informationsvorlauf. Damit ist dann die Vergleichbarkeit verschiedener Algorithmen mit unterschiedlichen Reihenfolgen der Bearbeitung auf der gleichen Sequenz von Anfragen nicht mehr so einfach gegeben, da in einzelnen Phasen der Bearbeitung die jeweils gesamten vorhandenen Informationen zweier

Algorithmen zur gleichen Zeit nicht identisch sein müssen. Falls etwa der eine zuletzt eine Anfrage mit einem großen Wert in der Größe ausgewählt hat und der andere eine mit einem ziemlich kleinen Wert, dann wird es vorkommen, dass der erste mehrere neue Anfragen zur Kenntnis bekommt, der zweite hingegen jedoch nur eine einzelne neue Anfrage erhält.

5.3. Literaturstellen zum Informationsvorlauf und dessen Analyse

Zur Verwendung von Informationsvorlauf (im englischen „lookahead“) in der bisherigen Fachliteratur hier eine sicherlich nicht vollständige Liste von Beispielen:

- So untersuchte Albers in der Arbeit [Alb93] den Einfluss des Informationsvorlaufs auf das Paging-Problem und in [Alb98] die Auswirkungen auf die Problemstellung der Aktualisierung von Listen. Bei beiden Problemen wurde dabei zwischen schwachen und starken Informationsvorlauf unterschieden. Der schwache Informationsvorlauf bestand aus einer Vorausschau über eine bestimmte Anzahl von Anfragen, der starke hingegen konnte auf so viele Anfragen zurückgreifen, bis er eine bestimmte Anzahl von verschiedenen Anfragen zur Verfügung hat. Als Ergebnisse wurden einige verbesserte Ober- und Unterschranken erreicht.
- In [AAL05] untersuchen Allulli, Ausiello und Laura die Auswirkungen auf ein Traveling-Salesman-Problem. Dabei betrachten sie einerseits einen Informationsvorlauf, der abhängig von der Zeit ist, und andererseits auch einen Informationsvorlauf mit einer festen Anzahl von Anfragen. Beim klassischen Traveling-Salesman-Problem ergaben sich keine Verbesserungen auf die kompetitive Analyse, dagegen erreichten sie bei der Variante ohne eine Rückkehr zum Ausgangspunkt sowie der Rundreise auf einer Linie mit fester Endzeit eine Verbesserung.
- Eine weitere differenziertere Untersuchung des Paging-Problems mit Informationsvorlauf findet sich von Breslauer in [Bre96]. In dieser Arbeit wird untersucht, wie sich ein Informationsvorlauf über die zukünftigen Anfragen und im Gegensatz dazu eine Vergrößerung des Zwischenspeichers auswirken. Die daraus gewonnene Erkenntnis ist, dass beide Erweiterungen den gleichen Effekt liefern und somit austauschbar sind.
- Grove schließlich untersucht mit [Gro95] den Einfluss von zusätzlicher Information bei der Problemstellung Binpacking. Dabei gewährt er einen Blick auf die Zukunft mit so vielen Anfragen, bis diese vorher vorgegebene Kosten überschreiten.
- In [Dam02] zeigt Damaschke ein Ergebnis für die Rundreise auf der Halbachse wobei er als Informationsvorlauf nur das Ende der Sequenz der Anfragen verwendet.

Mit diesem eigentlich minimalen zusätzlichen Wissen kann eine Verringerung des kompetitiven Faktors von $\frac{3}{2}$ auf $\frac{4}{3}$ erreicht werden, was den Einfluss der Kenntnis über den Zeitpunkt des Endes hervorhebt.

6. Verwendung des Informationsvorlaufs

Ein Informationsvorlauf lässt sich prinzipiell auf mehrere Arten verwenden:

- (i) Bei der Planung nach der Bekanntgabe der aktuellen Anfrage wird die zusätzliche Information zwar eingebunden, findet aber keine direkte Verwendung, das heißt die Bearbeitung der Anfragen erfolgt in der unveränderten Reihenfolge ihrer Bekanntgabe. Eine genauere Behandlung dieser Verwendung findet sich in Abschnitt 6.1
- (ii) Die zusätzlichen bekannten Informationen können komplett ebenfalls direkt bearbeitet werden, das heißt, die Anfragen müssen nicht zwingend nach der Reihenfolge ihrer Bekanntgabe bearbeitet werden. Diese Möglichkeit der Verwendung wird in Abschnitt 6.2 ausführlicher behandelt.
- (iii) Abschnitt 6.3 schließlich beschreibt eine aus den anderen beiden Arten kombinierte Verwendung des Informationsvorlaufs.

Für die weiteren Betrachtungen in diesem Kapitel ist der Informationsvorlauf, wie in Unterabschnitt 5.2.2 vorgestellt, strikt auf die Reihenfolge der Anfragen beschränkt. Das bedeutet, dass dem Online-Algorithmus zur Verbesserung seiner Informationslage grundsätzlich immer die nach der Reihenfolge in der Sequenz nächsten t Anfragen mit ihren Ausprägungen bekannt sind.

6.1. Informationsvorlauf als Mittel zur Vorausschau

Die Verwendung des Informationsvorlaufs als Vorausschau ist die einfachste Form der Verbesserung des Informationsstands des Online-Algorithmus.

Definition 6.1.1 (Informationsvorlauf zur Vorausschau).

Ein Informationsvorlauf der Größe $p \geq 0$ zur Vorausschau, bezeichnet als *Vorausschau-information*, bedeutet, dass der Online-Algorithmus bei der Planung zur Art der Bearbeitung der nächsten Anfrage σ_i grundsätzlich die Informationen von p zusätzlichen Anfragen zur Verfügung hat.

Allerdings darf dabei nur σ_i , im Normalfall ist das diejenige Anfrage mit dem kleinsten Index aus der Menge der zwar bekannt gegebenen aber noch nicht bearbeiteten Anfragen, sofort bearbeitet werden.

Dafür hat der Online-Algorithmus bei Problemstellungen mit dieser Art des Informationsvorlaufs grundsätzlich mehrere Möglichkeiten wie eine Bearbeitung durchgeführt werden kann. Das heißt, dass dann mit den Informationen aus der Vorausschauinformation die Bearbeitung der aktuellen Anfrage so gesteuert werden kann, dass deren Bearbeitung und die Bearbeitung der nachfolgenden Anfragen aus der Vorausschauinformation ein möglichst gutes Ergebnis ergibt.

Bei einer Vorausschauinformation von $p = 0$ Anfragen muss die Sequenz der Anfragen in der Reihenfolge der Anfragen abgearbeitet werden ohne bei der Bearbeitung der jeweiligen Anfrage zusätzliche Informationen zu weiteren zukünftigen Anfragen zur Verfügung zu haben.

Der in Definition 6.1.1 festgelegte Informationsvorlauf benötigt zur richtigen Entfaltung also die Möglichkeit, bei der Bearbeitung einer Anfrage eine Auswahl über die Art und Weise der Bearbeitung treffen zu können. Eine der Problemstellungen, die ein solches Verhalten möglich machen, ist zum Beispiel das Optimierungsproblem 2.1.3, das Packen von Gegenständen. Bei der Verpackung von Gegenständen geht es nämlich grundsätzlich darum ob ein Gegenstand eingepackt werden soll oder wohin dieser gepackt werden soll. Um diese Entscheidungen besser treffen zu können, ist es natürlich hilfreich die nächsten noch nicht zur Bearbeitung freigegebenen Gegenstände aus der Sequenz zu kennen.

Diese Verbesserung der Entscheidungsgrundlage macht sich im Normalfall insbesondere dann deutlich bemerkbar, wenn nicht über das Einpacken oder nicht Einpacken entschieden werden muss, sondern mehrere Behälter zum Einpacken der Gegenstände zur Auswahl stehen.

Durch die Beschränkung der Verwendung des so definierten Informationsvorlaufs gibt es eine Unterscheidung innerhalb der Einordnung der Anfragen in den Zeitpunkt der Bekanntgabe und den Zeitpunkt, zu welchem die Anfrage bearbeitet werden muss.

Definition 6.1.2 (Erster Bearbeitungspunkt bei Vorausschauinformation).

Falls ein Informationsvorlauf zur Verwendung als Vorausschauinformation bereitgestellt wird, dann entsprechen sich die zu den jeweiligen Anfragen gehörenden Merkmale der Bekanntgabe und der Freigabe zur Bearbeitung nicht mehr.

Der *erste Bearbeitungspunkt* einer Anfrage σ_i ergibt sich bei einem solchen Informationsvorlauf daher so, dass diese Anfrage erst nach der Bearbeitung aller Vorgänger σ_j mit $j < i$ für die Bearbeitung freigegeben wird.

Dagegen ist die Anfrage σ_i schon p Anfragen vorher das erste Mal bei der Planung der Bearbeitung der aktuellen Anfrage $\sigma_{\max\{i-p, 1\}}$ bekannt.

6.2. Informationsvorlauf als Mittel zur Auswahl

Nach der Definition des Informationsvorlaufs als Hilfe bei der Planung zur Bearbeitung der aktuellen Anfrage in Definition 6.1.1 soll in diesem Abschnitt ein Informationsvorlauf mit einer aktiveren Wirkung vorgestellt werden.

Definition 6.2.1 (Informationsvorlauf zur Bearbeitung).

Ein Informationsvorlauf zur Bearbeitung, bezeichnet als *Bearbeitungsinformation* der Größe $k \geq 1$ bedeutet, dass der Online-Algorithmus zum Zeitpunkt t grundsätzlich k Anfragen, welche schon zur Bearbeitung freigegeben sind, für die Auswahl der als nächstes zu bearbeitenden Anfrage zur Verfügung hat.

Dabei besteht bei dieser Art des Informationsvorlaufs im Allgemeinen die einzige Auswahlmöglichkeit des Online-Algorithmus darin, dass dieser zwar auswählen darf, welche der Anfragen aus der Bearbeitungsinformation bearbeitet werden soll, allerdings gibt es dann im Normalfall nur eine Möglichkeit wie diese Anfrage bearbeitet werden kann.

Eine Konstellation mit mehreren Möglichkeiten zur Bearbeitung der einzelnen Anfragen ist jedoch nicht ausdrücklich ausgeschlossen.

Insbesondere entspricht hierbei eine Bearbeitungsinformation von einer Anfrage genau der direkten Bearbeitung der Anfragen aus der Sequenz der Anfragen nach deren Reihenfolge in der Sequenz. Ein typisches Beispiel für eine solche Zielsetzung und der Art der Verwendung des Informationsvorlaufs ist eine Rundreise, bei dieser gibt es in der Regel zu jeder Anfrage nur eine einzige Möglichkeit zur Bearbeitung, nämlich den direkten Weg vom aktuellen Standort zum Reisepunkt der Anfrage.

Anders als bei der Vorausschauinformation kann bei der Bearbeitungsinformation direkt auf die Anfragen aus dem Informationsvorlauf zugegriffen werden. Das führt dazu, dass jede Anfrage sobald sie für den Online-Algorithmus sichtbar, also bekannt wird, sofort bearbeitet werden kann, damit wird die in Definition 6.1.2 vorgenommene Unterscheidung zwischen dem ersten Bearbeitungspunkt und der Bekanntgabe hinfällig.

Definition 6.2.2 (Erster Bearbeitungspunkt bei Bearbeitungsinformation).

Bei der Verwendung des Informationsvorlaufs als Bearbeitungsinformation stimmen der *erste Bearbeitungspunkt* und die Bekanntgabe der Anfrage überein.

Das heißt insbesondere, dass die Anfrage σ_i genau dann bekannt und zur Bearbeitung freigegeben wird, wenn der Online-Algorithmus die Planung für die Auswahl der Anfrage im Schritt $(i - k)_+ + 1$ vornimmt. Dabei ist k die Größe der Bearbeitungsinformation.

Nachdem sich aus dem in Definition 6.2.1 definierten Informationsvorlauf für jede Anfrage schon eine frühere Verfügbarkeit ergibt, muss darüber nachgedacht werden, wie die Bearbeitungsinformation am Anfang beim Start des Algorithmus initialisiert werden soll.

Methoden beim Start

Diese hier nun vorgenommenen Überlegungen sind meist insbesondere bei der zusätzlichen Verwendung von Wartezeiten interessant, das heißt dann, wenn es von Interesse ist, dass die Anfragen entweder besonders schnell oder innerhalb eines bestimmten vorgegebenen Zeitrahmens bearbeitet werden sollen.

Die *Wartezeit* ergibt sich dabei als die Differenz zwischen dem Schritt während dessen Planung die Anfrage bekannt gegeben wird und dem Schritt in welchem die Anfrage dann konkret bearbeitet wird.

Definition 6.2.3 (Startmethoden).

Beim Start eines Online-Algorithmus mit einer Bearbeitungsinformation von k Anfragen gibt es verschiedene Ausprägungen:

Feststart: Bei einem *Feststart* ist die Anfrage σ_1 die erste Anfrage und noch keine weitere Bearbeitungsinformation vorhanden. Der Algorithmus startet somit nur mit den Informationen zu der Anfrage σ_1 .

Bei dieser Methode muss eine Möglichkeit gegeben sein, dass auch mal keine Anfrage bearbeitet wird, bis insgesamt k noch nicht bearbeitete Anfragen vorhanden sind.

Fliegender Start: Beim *fliegenden Start* sind zu Beginn schon außer σ_1 zusätzlich $k - 1$ frühere, noch nicht bearbeitete Anfragen aus einer vorherigen Bearbeitung des Online-Algorithmus bekannt.

Also kann der Algorithmus am Anfang auf die Informationen von k Anfragen $\{\sigma_{\pi(2-k)}, \dots, \sigma_{\pi(0)}, \sigma_1\}$ zurückgreifen. Die injektive Abbildung π ordnet dabei die zusätzlichen früheren Anfragen nach ihrer Reihenfolge in der vorherigen Sequenz der Anfragen, $\pi(2 - k)$ kennzeichnet also die älteste dieser Anfragen.

Obwohl hierbei mit früheren Anfragen gearbeitet werden muss, erfolgt der Start der Beobachtung und damit auch der Bewertung des Algorithmus erst ab der Bekanntgabe der ersten Anfrage σ_1 aus der Sequenz der Anfragen und den danach getroffenen Entscheidungen.

Start mit voll nutzbarer Information (Vollstart): Mit *Vollstart* ist gemeint, dass mit Beginn auch schon die ersten k Anfragen bearbeitet werden können. Dabei werden diese direkt aus der Sequenz der Anfragen σ entnommen und der Startpunkt auf k gesetzt.

Die Konstellation beim Start gestaltet sich dann als $\{\sigma_1, \dots, \sigma_k\}$. Dies entspricht im Endeffekt einem Feststart mit einer Nichtbearbeitung der ersten $k - 1$ Anfragen.

Natürlich sollte bei der Betrachtung der verschiedenen Startmethoden klar sein, dass bei der Verwendung einer unendlichen Sequenz von Anfragen die unterschiedlichen Auswirkungen der einzelnen Startmethoden auf das insgesamt Ergebnis mit zunehmender Anzahl der schon bearbeiteten Anfragen verschwinden und sich die Bearbeitung auf eine identische Struktur einpendelt.

Erweiterungen zur Bearbeitungsinformation

Da es bei einer Bearbeitung einer Sequenz von Anfragen mit einer Bearbeitungsinformation zu Verschiebungen in der Reihenfolge der Bearbeitung der einzelnen Anfragen

kommen kann und im Allgemeinen auch kommen wird, ergibt sich dabei der eventuell nicht so ganz erwünschte Effekt einer Verschiebung der Bearbeitung einer Anfrage bis zum Ende der Sequenz.

Um diesen Effekt zu vermeiden, ist es möglich, dazu als neues Konzept eine maximale Verzögerung einzuführen.

Definition 6.2.4 (maximale Verzögerung).

Die erste Möglichkeit zur Bearbeitung einer Anfrage σ_i ist bei einer Bearbeitungsinformation von k Anfragen während der Auswahl der Anfrage in Schritt $((i - k)_+ + 1)$. Damit ergibt sich bei einer *maximalen Verzögerung* von ℓ , dass die Anfrage spätestens im Schritt $(i - k + 1 + \ell)$ bearbeitet werden muss.

Damit eine maximale Verzögerung überhaupt einen Sinn ergibt, muss eine größere Bearbeitungsinformation als eine Anfrage gegeben sein. Folglich muss also $k > 1$ gelten und außerdem muss unbedingt $\ell \geq k - 1$ gelten, damit überhaupt eine zulässige Bearbeitung möglich ist.

Bei vorgegebener maximaler Verzögerung muss allerdings geklärt werden, ob die sich durch das erstmalige Auffüllen der Bearbeitungsinformation ergebenden Verzögerungen als tatsächliche Verzögerungen gewertet werden oder nicht.

6.3. Kombination der beiden Verwendungsmöglichkeiten

In diesem Abschnitt soll nun zum Abschluss von Kapitel 6 eine Kombination der in Abschnitt 6.1 und Abschnitt 6.2 eingeführten Arten der Verwendung vorgestellt werden. Der Informationsvorlauf besteht dementsprechend aus zwei unterschiedlichen Teilen.

Definition 6.3.1 (Kombinierter Informationsvorlauf).

Ein *kombinierter Informationsvorlauf* der Größe $k + p$ bedeutet, dass der Online-Algorithmus bei der Auswahl der nächsten Anfrage σ_i grundsätzlich Informationen zu $k + p$ Anfragen für die Planung zur Verfügung hat.

Der kombinierte Informationsvorlauf teilt sich dabei in zwei Abschnitte auf, die jeweils nach Definition 6.1.1 und Definition 6.2.1 aufgebaut sind. Analog zu diesen Definitionen besteht dieser kombinierte Informationsvorlauf aus einer Bearbeitungsinformation von k Anfragen und einer Vorausschauinformation von p Anfragen.

Sei die gerade eben zur Bearbeitung freigegebene Anfrage, also die Anfrage mit dem größten Index aus den Anfragen welche schon bearbeitet werden dürfen, mit σ_i bezeichnet. Dann ist der zweite Abschnitt des Informationsvorlaufs, also die Vorausschauinformation, die Menge $\{\sigma_{i+1}, \dots, \sigma_{i+p}\}$. Der erste Abschnitt, die Bearbeitungsinformation, besteht analog soweit möglich aus $k - 1$ noch nicht bearbeiteten Anfragen aus der Menge $\{\sigma_j \mid j < i\}$, sowie der Anfrage σ_i . Die Anfragen sind dabei aufsteigend nach ihrer Reihenfolge in der Sequenz der Anfragen sortiert.

Durch die Kombination der beiden Arten der Verwendung des Informationsvorlaufs, ergeben sich auch bei der Bekanntgabe und dem ersten Bearbeitungspunkt der Anfragen Unterschiede.

Definition 6.3.2 (Erster Bearbeitungspunkt bei kombiniertem Informationsvorlauf). Da beim kombinierten Informationsvorlauf eine Bearbeitungsinformation von k Anfragen gegeben ist, liegt der *erste Bearbeitungspunkt* der Anfrage σ_i schon bei der Auswahl der im Schritt $((i - k)_+ + 1)$ zu bearbeitenden Anfrage.

Bedingt durch die Vorausschauinformation von p Anfragen, diene diese Anfrage σ_i dann aber schon bei p vorherigen Schritten als zusätzliche Information bei der Planung der Auswahl der nächsten Anfrage und deren Bearbeitung. Das heißt, dass die Anfrage σ_i bereits im Schritt $((i - k - p)_+ + 1)$ dem Online-Algorithmus bekannt gegeben wird.

Teil III.

Rundreise auf der Kreislinie

Übersicht

In diesem Teil wird die Wirkung eines Informationsvorlaufs auf die Problemstellung der Rundreise auf der Kreislinie untersucht.

Kapitel 7 erklärt die Aufgabenstellung etwas ausführlicher. Hierbei wird unter anderem eine kompetitive Analyse durchgeführt und das zur Problemstellung gehörende gemischt-ganzzahlige, lineare Modell vorgestellt.

In Kapitel 8 findet sich die Untersuchung der Wirkung des Informationsvorlaufs auf eine vereinfachte Variante der Problemstellung.

Die Auswirkung auf die allgemeinere Problemstellung wird anschließend in Kapitel 9 untersucht.

Kapitel 10 zeigt Ergebnisse aus empirischen Untersuchungen zu der diskreten Variante der Rundreise auf der Kreislinie.

Das Kapitel 11 liefert schließlich einen kurzen Ausblick zur Rundreise auf der Oberfläche der Kugel.

7. Einführung

Als Hauptproblem dieses Kapitels soll eine besondere Art von Traveling-Salesman-Optimierungsproblem Anwendung finden. Zuerst einmal wird die Problemstellung definiert und an Hand eines Beispiels etwas mehr veranschaulicht. Danach wird eine grundsätzliche kompetitive Analyse durchgeführt und eine Formulierung der Aufgabenstellung als gemischtganzzahliges lineares Modell vorgestellt.

7.1. Grundlegende Definitionen

Die formelle Definition des Hauptproblems für diesen Teil lautet wie folgt:

Optimierungsproblem 7.1.1 (Rundreise auf der Kreislinie).

Die *Rundreise auf der Kreislinie* – in dieser allgemeinen Form auch als *Rundreise auf der Kreislinie mit Richtungswechsel* bezeichnet – ist als ein auf der Linie eines Kreises beschränktes Traveling-Salesman-Problem mit Start- und Zielpunkt v_0 , sowie weiteren Reisepunkten v_r definiert. Dabei wird v_0 auch als der *Ausgangspunkt* bezeichnet.

Da v_0 nicht nur der Startpunkt, sondern auch der Zielpunkt ist, muss der Reisende am Ende seiner Reise, wenn alle Anfragen mit ihren zugehörigen Reisepunkten abgearbeitet wurden, wieder zum Ausgangspunkt zurückkehren.

Zur Bestimmung der Lage der einzelnen Reisepunkte wird ein Koordinatensystem verwendet, welches sich am Winkel des Kreisabschnittes des einzelnen Reisepunktes zum Ausgangspunkt v_0 mit dessen festgesetzter Winkelkoordinate 0 orientiert. Der relevante

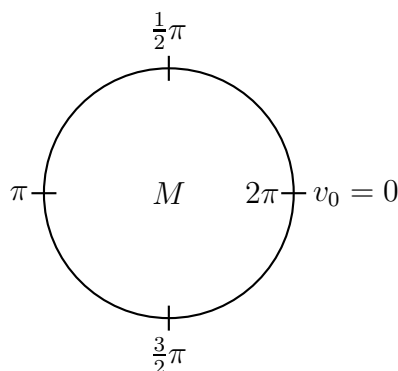


Abbildung 7.1.1.: Koordinatensystem zur Rundreise auf der Kreislinie

Kreisabschnitt ergibt sich als der Abschnitt des Kreises, welcher als Weg ausgehend vom

Ausgangspunkt bis zum Reisepunkt mit Laufrichtung gegen den Uhrzeigersinn verläuft. Als räumliche Position des Ausgangspunktes v_0 wird in den Beispielen der rechte Schnittpunkt der waagrechten Geraden durch den Kreismittelpunkt M mit der Kreislinie verwendet.

Weiterhin sei die Reisegeschwindigkeit des Reisenden konstant und als eine Längeneinheit pro Zeiteinheit festgesetzt.

Bei den späteren Berechnungen ist der Radius des Kreises wegen der Verwendung der Winkelkoordinaten unerheblich und wird daher auf eins gesetzt.

Nach der Einführung von Optimierungsproblem 7.1.1 nun noch einige dazugehörige notwendige Begriffe.

Definition 7.1.2 (Begriffsbildungen für die Rundreise auf der Kreislinie).

Weitere zur Beschreibung der Aufgabenstellung notwendige Begriffe sind:

- Die gesamte *Sequenz der Anfragen* einer Rundreise wird mit σ bezeichnet und besteht grundsätzlich aus Einträgen $\sigma_i = (v_i \ t_i)$ für einzelne *Anfragen*. Dabei bezeichnet $t_i \geq 0$ den Zeitpunkt der Bekanntgabe der Anfrage und $v_i \in [0, 2\pi]$ die Winkelkoordinate der Anfrage, welche als *Reisepunkt* bezeichnet wird.

Ein Reisepunkt gibt dabei an, welche Position der Kreislinie erreicht werden muss, damit die zugehörige Anfrage als bearbeitet gilt.

- Eine Anfrage σ_i gilt als *erfüllt*, wenn ihr Reisepunkt v_i nach dem Zeitpunkt t_i angefahren wird.
- Die *Menge der Anfragen* bis zu einer bestimmten Zeit t wird mit $\sigma_{\leq t}$ bezeichnet und besteht aus allen Anfragen σ_i mit $t_i \leq t$.
- Eine *Reiseroute* R für eine Sequenz $\sigma = \sigma_1, \dots, \sigma_k$ ist die Bildung einer Reihenfolge, in welcher diese Anfragen angefahren werden sollen. Eine Rundreise auf der Kreislinie besteht dann aus den Anfragen σ und einer Rundreise $R(\sigma)$, welche alle Reisepunkte der Anfragen besucht, hierbei keine Anfrage vor ihrer Bekanntgabe erfüllt, und danach wieder zum Ausgangspunkt v_0 zurückkehrt. Dabei müssen die Reisepunkte $v_1, \dots, v_k$ nicht notwendigerweise paarweise verschieden sein, das heißt ein Reisepunkt darf auch mehrmals angefahren werden.

Die Bekanntgabe der Anfrage orientiert sich damit, zumindest in diesem Kapitel, an den in Definition 5.2.1 verwendeten Zeiteinheiten.

7.2. Beispiel für eine Sequenz von Anfragen

Um einen ersten Einblick in die Problematik der Online-Optimierung der Rundreise auf der Kreislinie zu erhalten, wird nun ein kleines Beispiel betrachtet.

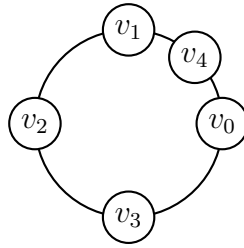


Abbildung 7.2.1.: Einfaches Beispiel für die Rundreise auf der Kreislinie

Beispiel 7.2.1 (Sequenz aus Anfragen für das Optimierungsproblem 7.1.1).

Abbildung 7.2.1 zeigt ein einfaches Beispiel für ein Rundreise-Problem auf der Kreislinie mit Start- und Zielpunkt v_0 und den Daten der Anfragen aus Tabelle 7.2.1. Der Radius des verwendeten Kreises sei als 1 gegeben, das heißt, es wird der Einheitskreis verwendet.

σ_i	1	2	3	4
v_i	$\frac{1}{2}\pi$	π	$\frac{3}{2}\pi$	$\frac{1}{4}\pi$
t_i	0	0	0	π

Tabelle 7.2.1.: Reisedaten zu Beispiel 7.2.1

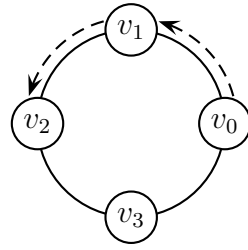
Die damit verbundene Aufgabe ist es nun, eine minimale Wegstrecke so auszuwählen, dass alle Anfragen $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ erfüllt werden und die Rundreise danach so schnell wie möglich wieder bei v_0 ankommt.

Bei den hier in Tabelle 7.2.1 gegebenen Anfragen ist zum Beispiel für die Sequenz $\sigma_{t \leq 0} = \{\sigma_1, \sigma_2, \sigma_3\}$ die Rundreise mit den Stationen

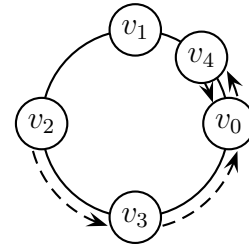
$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_0 \quad (7.2.1)$$

eine der zwei optimalen Lösungen. Da diese Rundreise einem einmaligen Umfahren des Einheitskreises entspricht, ist dafür eine Reisestrecke von 2π notwendig. Mit der oben im Optimierungsproblem 7.1.1 getroffenen Aussage über die Reisegeschwindigkeit wird daher für diese Rundreise eine Zeit von 2π benötigt. Weil eine schnellere Bearbeitung der Anfragen $\{\sigma_1, \sigma_2, \sigma_3\}$ nicht möglich ist, ist somit der Quotient zwischen der Lösung des Online-Algorithmus und der Lösung der Offline-Bearbeitung gleich eins. Doch dieses Ergebnis gilt bei der hier vorgegebenen Instanz nur für die Teilsequenzen $\sigma_{t \leq j}$ mit $j < \pi$.

Bei der Betrachtung der gesamten Sequenz $\sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ hingegen, wird bei einer Verwendung der in (7.2.1) angegebenen Route zum Zeitpunkt π bei Erreichen des Punktes v_2 und der damit verbundenen Erfüllung der Anfrage σ_2 die zusätzliche Anfrage σ_4 mit dem Reisepunkt v_4 bekannt. Dabei liegt v_4 wie in Abbildung 7.2.2 gezeigt auf



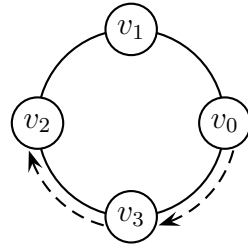
(a) Online-Route bis zum Zeitpunkt π



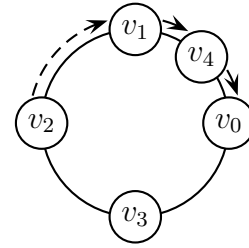
(b) Online-Route nach Bekanntgabe von v_4

Abbildung 7.2.2.: Route der Online-Bearbeitung

der Strecke $v_0 \rightarrow v_1$. Damit ergibt sich nun für die Online-Optimierung mit der schon gefahrenen Route $v_0 \rightarrow v_1 \rightarrow v_2$ insgesamt eine größere Fahrtstrecke, nämlich $2\pi + \frac{1}{2}\pi$, da der Punkt v_4 noch einmal angefahren werden muss. Somit kann die Online-Rundreise also erst nach $\frac{5}{2}\pi$ Zeiteinheiten beendet werden.



(a) Offline-Route bis zum Zeitpunkt π



(b) Offline-Route nach Bekanntgabe von v_4

Abbildung 7.2.3.: Route der Offline-Bearbeitung

Bei vorheriger Kenntnis dieser Anfrage führt die optimale Reiseroute zu v_2 nicht über $v_0 \rightarrow v_1 \rightarrow v_2$, sondern wie in Abbildung 7.2.3 links zu sehen über $v_0 \rightarrow v_3 \rightarrow v_2$ und danach über v_1 und v_4 wieder zurück nach v_0 mit einer Gesamtstrecke von 2π . Das bedeutet also, dass sich an der Lösung der Offline-Bearbeitung durch die Hinzunahme von v_4 nichts ändert, nur dass noch zusätzlich in v_4 Halt gemacht werden muss und die zweite Lösung mit der umgekehrten Laufrichtung nicht mehr optimal ist.

Der relative Fehler dieser Instanz der Problemstellung ergibt sich damit zu

$$\Delta_{\text{relativ}} = \frac{\frac{5}{2}\pi - 2\pi}{2\pi} = \frac{1}{4} = 25 \text{ Prozent}$$

Dies entspricht einem Quotienten von $\frac{5}{4}$ zwischen der Streckenlänge der Lösung des Algorithmus und der Streckenlänge der Lösung der Offline-Bearbeitung für diese Instanz der Problemstellung. Damit ist schon einmal sicher, dass der kompetitive Faktor für diese Problemstellung bei Verwendung von deterministischen Algorithmen größer gleich $\frac{5}{4}$ ist.

7.3. Kompetitive Analyse

In diesem Abschnitt soll das Optimierungsproblem 7.1.1, also die Rundreise auf der Kreislinie, mithilfe der kompetitiven Analyse untersucht werden.

Da die einzelnen Reisepunkte bei diesem Rundreise-Problem auf der Kreislinie liegen und diese keine Endpunkte besitzt, gibt es zwischen je zwei Punkten v_i und v_j zwei verschiedene Reiserouten R_1 und R_2 , welche eine disjunkte Zerlegung der Kreislinie darstellen. In dem in Abbildung 7.3.1 dargestellten Beispiel sind für die zwei Punkte v_i

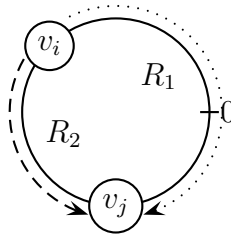


Abbildung 7.3.1.: Verschiedene Reiserouten auf der Kreislinie

und v_j beide Routen mit jeweils verschiedenen Linien eingezeichnet. Die kleinere der beiden Routen R_2 , das ist die gestrichelte Linie in der Abbildung 7.3.1, hat hier als Streckenlänge $L(R_2)$ den Wert $(v_j - v_i)$ und die größere, als gepunktete Linie eingezeichnet, hat, da wegen der durch die beiden Routen gegebenen disjunkten Zerlegung des Kreises

$$L(R_1) + L(R_2) = 2\pi$$

gelten muss, die Länge $2\pi - (v_j - v_i)$. Um nun zu vermeiden, dass bei der Angabe der minimalen Distanz zwischen je zwei Punkten immer der gesamte Ausdruck

$$\min \{|v_i - v_j|, 2\pi - |v_i - v_j|\}$$

verwendet werden muss, wird dazu eine Hilfsfunktion definiert.

Definition 7.3.1 (Distanzfunktion).

Für die Berechnung der minimalen Distanz zwischen je zwei Punkten v_i und v_j auf der Kreislinie ohne eine Beschränkung der Laufrichtung wird folgende *Distanzfunktion*

eingeführt:

$$\text{dist}(v_i, v_j) := \min \{|v_i - v_j|, 2\pi - |v_i - v_j|\}$$

Zur Vereinfachung kann $\text{dist}(v_i)$ für die Distanz des Punktes v_i zum Ausgangspunkt v_0 verwendet werden.

Da bei der Rundreise auf der Kreislinie immer nur die direkte Distanz zwischen zwei in der Reiseroute aufeinander folgenden Reisepunkten von Interesse ist, reicht die in Definition 7.3.1 zur Berechnung der Distanz definierte Funktion aus. Ein Umweg über die längere Route zwischen zwei Reisepunkten ist nämlich nur dann sinnvoll, wenn auf dieser Route ein weiterer Reisepunkt bearbeitet werden könnte. In diesem Fall würden dann aber bei einer effektiven Planung der Reiseroute die beiden betrachteten Reisepunkte nicht mehr aufeinander folgen.

Definition 7.3.2 (Streckenlänge zwischen zwei Reisepunkten innerhalb einer Route).

Die *Streckenlänge* $L_R(v_i, v_j)$ zwischen zwei Reisepunkten v_i und v_j mit $j > i$ innerhalb der Route R ist im Fall von $j = i + 1$ durch den Wert der Distanzfunktion ausgewertet für die beiden Punkte, also als $\text{dist}(v_i, v_j)$ gegeben. Für $j > i + 1$ berechnet sich die Streckenlänge innerhalb der Route durch die Summe über die dazwischen liegenden Einzelabschnitte $\text{dist}(v_t, v_{t+1})$ mit t von i bis $j - 1$. Allgemein gilt immer:

$$L_R(v_i, v_j) = \sum_{t=i}^{j-1} \text{dist}(v_t, v_{t+1}) \geq \text{dist}(v_i, v_j) \quad j \geq i$$

Dabei ist $(v_0, v_1, \dots, v_{n+1} = v_0)$ die Auflistung der Reisepunkte zur Route R nach der Reihenfolge in der die Reisepunkte angefahren werden.

Weil in dieser Arbeit eine Geschwindigkeit von einer Zeiteinheit pro Längeneinheit betrachtet wird, lässt sich die Fahrtzeit zwischen zwei Reisepunkten innerhalb einer Route R ebenfalls durch die Funktion L_R ausdrücken. Die gesamte Streckenlänge einer Route R berechnet sich über $L_R(v_0, v_{n+1})$ mit $v_{n+1} = v_0$.

Bevor nun nach diesen Einführungen die ersten Resultate gezeigt werden, werden zwei der Strategien aus Definition 3.1.6 als Algorithmen ausformuliert.

Algorithmus 7.3.3 (Einfacher Replan-Algorithmus).

Der einfache Replan-Algorithmus verwendet die Replan-Strategie aus Definition 3.1.6, das bedeutet, dass der Algorithmus bei Bekanntgabe einer neuen Anfrage immer eine Neuplanung seines weiteren Vorgehens vornimmt.

Idealerweise wird dabei ausgehend vom aktuellen Standort ω ein optimaler Weg zum Ausgangspunkt v_0 gesucht, welcher die Reisepunkte aller bisher bekannten aber noch nicht bearbeiteten Anfragen σ' einschließlich der neu bekannt gegebenen Anfragen besucht. Hierbei ist ein mehrmaliger Besuch verschiedener Punkte, zum Beispiel des Ausgangspunktes, nicht ausgeschlossen.

Algorithmus 7.3.4 (Einfacher Ignore-Algorithmus).

Aufbauend auf der Ignore-Strategie von Definition 3.1.6 entspricht das Vorgehen des einfachen Ignore-Algorithmus einer gewissen Sturheit.

Der Algorithmus plant anfangs mit den ersten bekannten Anfragen eine erste Rundreise durch deren Reisepunkte mit Rückkehr zum Ausgangspunkt. Während der Bearbeitung dieser Rundreise ignoriert er alle weiteren neu bekannt werdenden Anfragen bis zu dem Zeitpunkt der Rückkehr zum Ausgangspunkt.

Bei diesem angekommen wird für alle bis dahin ignorierten Anfragen eine neue, idealerweise optimale, Rundreise mit Rückkehr zum Ausgangspunkt geplant und diese wieder ohne Unterbrechung bis zu deren Ende durchgeführt.

Das erste Resultat der Analyse ist eine Oberschranke für die Dauer einer Rundreise auf einem Kreis.

Lemma 7.3.5 (Maximale Reisedauer der Rundreise auf der Kreislinie).

Sei eine Sequenz von Anfragen $\sigma_i = (v_i \ t_i)$ mit $i = \{1, \dots, n\}$ und $t_{i-1} \leq t_i$ sowie $v_i \in [0, 2\pi]$ zum Rundreise-Problem auf der Kreislinie gegeben. Dann gelten folgende Aussagen für die maximale Reisedauer $T \geq t_n$:

- (i) *Die maximale Dauer T_{Off} einer Rundreise auf der Kreislinie bei der Offline-Bearbeitung ist:*

$$T_{\text{Off}} \leq t_n + 2\pi$$

- (ii) *Die maximale Dauer T_{Replan} einer Rundreise auf der Kreislinie bei der Online-Bearbeitung mit Verwendung des Algorithmus 7.3.3 und ohne Warten beträgt:*

$$T_{\text{Replan}} \leq t_n + 3\pi$$

Beweis zu Lemma 7.3.5.

Sei $\sigma_{=t_n}$ die Sequenz der Anfragen, welche zum Zeitpunkt t_n bekannt werden, sowie ω der aktuelle Standort des Algorithmus zum Zeitpunkt t_n . Außerdem bezeichne

$$T(\{\sigma_{j_1}, \dots, \sigma_{j_\ell}\}) \leq 2\pi$$

die Streckenlänge, und damit auch die Bearbeitungszeit, der optimalen Rundreise ausgehend vom Startpunkt v_0 für die Reisepunkte der Anfragen $\{\sigma_{j_1}, \dots, \sigma_{j_\ell}\}$ unabhängig vom Zeitpunkt der Bekanntgabe der jeweiligen Anfrage.

Zu (i): Da bei der Offline-Bearbeitung die vollständige Information für die Problemstellung der Rundreise auf der Kreislinie vorliegt, kann theoretisch mit dem Beginn der Reise gewartet werden bis der letzte Reisepunkt bekannt ist, und dann erst die eigentliche Fahrt, welche maximal einmal um den Kreis führt, starten. Also ergibt sich eine maximale Reisedauer von $t_n + 2\pi$.

Bei einer Verwendung von auf dem gesamten Kreis äquidistanten Reisepunkten mit Bekanntgabe zum Zeitpunkt t_n wird diese Schranke im Grenzwert für eine unendliche Anzahl von Reisepunkten angenommen. Der günstigste Standort zu diesem Zeitpunkt ist der Ausgangspunkt v_0 , da dann nur eine Strecke von 2π zur Bearbeitung der Anfragen und der anschließenden Rückkehr zum Ausgangspunkt benötigt wird. Wird ein anderer Standort $\bar{\omega}$ verwendet, so muss, nach der Abarbeitung der Anfragen mit einem Aufwand von 2π , noch die Strecke von $\text{dist}(\bar{\omega})$ zum Ausgangspunkt zurückgelegt werden.

Zu (ii): Um von ω zum Ausgangspunkt zu gelangen, muss eine Strecke von $\text{dist}(\omega)$ zurückgelegt werden. Beginnend beim Ausgangspunkt sind in maximal $T(\sigma_{=t_n} \cup \sigma_{Rest})$ Zeiteinheiten alle Reisepunkte der Anfragen aus $\sigma_{=t_n}$ und der Anfragen $\sigma_{Rest} \subseteq \sigma_{<t_n}$, welche zu dem Zeitpunkt t_n noch nicht von dem Algorithmus 7.3.3 bearbeitet worden sind, erreichbar. Also gilt

$$T_{Replan} \leq t_n + T(\sigma_{=t_n} \cup \sigma_{Rest}) + \text{dist}(\omega) \quad (7.3.1)$$

Die rechte Seite der Ungleichung (7.3.1) wird für $\omega = \pi$ und $T(\sigma_{=t_n} \cup \sigma_{Rest}) = 2\pi$ maximal und damit ergibt sich die obige Schranke, welche ebenfalls durch auf dem ganzen Kreis verteilte äquidistante Reisepunkte in der Betrachtung des Grenzwertes mit unendlich vielen Anfragen erreicht werden kann. $\square$

Bemerkung 7.3.6 (Lemma 7.3.5 Teil (ii) mit Ignore).

Bei Anwendung des Algorithmus 7.3.4 verschlechtert sich die Oberschranke für die gesamte Reisezeit auf $T_{Ignore} \leq t_n + 4\pi$.

Auch diese Oberschranke ist scharf, dazu müssen nur unmittelbar nach dem Start einer neuen Route mit Streckenlänge 2π neue Anfragen bekannt gegeben werden, welche zur Bearbeitung wieder eine Route mit Länge 2π benötigen. Da Algorithmus 7.3.4 alle nach der Planung bekannt gewordenen Anfragen nicht beachtet, werden bei einer strengen Auslegung dieser Strategie die neuen Anfragen selbst dann nicht bearbeitet, wenn der Algorithmus bei der Abarbeitung seiner vorher geplanten Rundreise deren Reisepunkt noch besuchen wird.

Allerdings ist bei der Problemstellung der Rundreise auf der Kreislinie die reine Anwendung von Algorithmus 7.3.4 im Normalfall ineffizient, zumindest sollten immer alle Anfragen als bearbeitet gesetzt werden, deren Reisepunkt nach der Bekanntgabe bei der Fahrt zum Reisepunkt einer anderen Anfrage angefahren wurde.

Korollar 7.3.7 (Erste Unterschranke für den kompetitiven Faktor des Rundreise-Problems auf der Kreislinie).

Die Problemstellung der Rundreise hat bei Verwendung des Algorithmus 7.3.3, wie im Beispiel 7.2.1 gesehen, einen strikten kompetitiven Faktor c welcher durch $\frac{5}{4}$ nach unten beschränkt ist.

Das einfache Beispiel zur Veranschaulichung der Problemstellung liefert einen Hinweis darauf, wie eine Sequenz aufgebaut sein kann um den kompetitiven Faktor eines deterministischen Online-Algorithmus größer werden zu lassen.

Beispiel 7.3.8 (Erweiterung von Beispiel 7.2.1).

Gegeben sei die Sequenz $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_{3n-1}\}$ mit:

- $v_i = \frac{i}{n}\pi$ und $t_i = 0$ für alle $i \leq 2n - 1$
- Wenn die Summe des aktuellen Zeitpunktes $\tilde{t}$ und der schon gefahrenen Strecke $\text{dist}(\tilde{\omega})$ zum momentanen Standort $\tilde{\omega}$ des Online-Algorithmus 2π ergibt, also bei

$$\tilde{t} + \text{dist}(v_0, \tilde{\omega}) = 2\pi,$$

werden n neue Anfragen $\sigma_{2n}, \dots, \sigma_{3n-1}$ mit Abstand $\frac{\text{dist}(\tilde{\omega})}{n}$ auf der vom Online-Algorithmus bereits gefahrenen Strecke platziert.

Satz 7.3.9 (Zweite Unterschranke für den kompetitiven Faktor des Rundreise-Problems auf der Kreislinie).

Der strikte kompetitive Faktor von Beispiel 7.3.8 für alle deterministischen Algorithmen ist zwei und somit gilt allgemein $c \geq 2$ für die strikt kompetitiven Faktoren c der deterministischen Algorithmen für das Optimierungsproblem 7.1.1 der Rundreise auf der Kreislinie.

Beweis zu Satz 7.3.9.

In Beispiel 7.3.8 ergibt sich für die Offline-Bearbeitung eine Gesamtzeit für die Rundreise von $T_{\text{Off}} = 2\pi$. Dabei muss die Fahrt um den Kreis nur so beginnen, dass zum Zeitpunkt $\tilde{t}$ der Streckenabschnitt mit den Anfragen $\sigma_{2n}, \dots, \sigma_{3n-1}$ noch nicht abgefahren ist.

Dagegen benötigt jeder deterministische Online-Algorithmus, wegen der kontraproduktiven Festlegung der neuen Anfragen $\sigma_{2n}, \dots, \sigma_{3n-1}$ für das gleiche Problem eine Reisezeit von:

$$T_{\text{Alg}} = 4\pi - 2 \cdot \frac{\text{dist}(\tilde{\omega})}{n}$$

Die Reisezeit ergibt sich dabei aus folgenden Komponenten:

- Aus der aktuellen Zeit $\tilde{t}$.
- Reisezeit für die restlichen noch nicht bearbeiteten Anfragen aus der Sequenz $\{\sigma_1, \dots, \sigma_{2n-1}\}$, also $2\pi - \text{dist}(\tilde{\omega})$
- Reisezeit für die Anfragen zum Zeitpunkt $\tilde{t}$ von $2 \left(\text{dist}(\tilde{\omega}) - \frac{\text{dist}(\tilde{\omega})}{n} \right)$

Damit ergibt sich für die gesamte Reisezeit T_{Alg} der Online-Bearbeitung:

$$\begin{aligned} T_{Alg} &= \tilde{t} + (2\pi - \text{dist}(\tilde{\omega}, 0)) + 2 \left(\text{dist}(\tilde{\omega}) - \frac{\text{dist}(\tilde{\omega})}{n} \right) \\ &= 2\pi + \tilde{t} + \text{dist}(\tilde{\omega}) - 2 \cdot \frac{\text{dist}(\tilde{\omega})}{n} \stackrel{\langle \tilde{t} + \text{dist}(\tilde{\omega}) = 2\pi \rangle}{=} 4\pi - 2 \cdot \frac{\text{dist}(\tilde{\omega})}{n} \end{aligned}$$

Diese beiden Reisezeiten liefern somit einen strikten kompetitiven Faktor c_n mit:

$$c_n = \frac{(4\pi - 2 \cdot \frac{\text{dist}(\tilde{\omega})}{n})}{2\pi} = 2 - \frac{\text{dist}(\tilde{\omega})}{n\pi}$$

Bei der Betrachtung des Grenzwertes $\lim_{n \rightarrow \infty} c_n$ ergibt sich eine Unterschranke von zwei für den strikten kompetitiven Faktor des Grundproblems. $\square$

Diese Unterschranke von zwei für den strikten kompetitiven Faktor der Problemstellung der Rundreise aus dem Optimierungsproblem 7.1.1 lässt sich sogar als bestmöglicher kompetitiver Faktor zeigen.

Ausiello, Feuerstein und Leonardi beweisen in [AFL⁺95, Theorem 4.1] für den allgemeinen Fall einer Rundreise in einem metrischen Raum, dass der von ihnen definierte Algorithmus „PAH“ einen strikt kompetitiven Faktor von zwei besitzt. Dieser Algorithmus ist auch für die Problemstellung der Rundreise auf der Kreislinie anwendbar, so dass sich für die Rundreise auf der Kreislinie ein strikt kompetitiver Faktor von zwei ergibt. Der dazu notwendige Algorithmus ist wie folgt aufgebaut:

Algorithmus 7.3.10 (PAH nach [AFL⁺95, Kapitel 4]).

Sei t_n der aktuelle Zeitpunkt und ω der aktuelle Standort. Das Vorgehen des Algorithmus gliedert sich dann wie folgt:

- (i) *Wenn der Algorithmus den Ausgangspunkt v_0 erreicht, also $\omega = v_0$ gilt, so wird für alle aktuell unbearbeiteten Anfragen eine optimale Rundreise zur Abarbeitung der zugehörigen Reisepunkte mit anschließender Rückkehr zum Ausgangspunkt erstellt und mit dieser begonnen.*
- (ii) *Bei der Bekanntgabe einer neuen Anfrage σ' mit Reisepunkt v' ändert der Algorithmus je nachdem wie v' zu ω positioniert ist sein Vorgehen:*
 - (a) *Falls $\text{dist}(v') > \text{dist}(\omega)$ gilt, die neue Anfrage also weiter weg von Ausgangspunkt positioniert ist als der aktuelle Standort, dann verwirft der Algorithmus seine aktuelle Rundreise und begibt sich auf direkten Weg zurück zum Ausgangspunkt v_0 .*
 - (b) *Im anderen Fall $\text{dist}(v') \leq \text{dist}(\omega)$, wird die Anfrage so lange ignoriert, bis der Algorithmus wieder den Ausgangspunkt v_0 erreicht.*

Dabei steht die Abkürzung PAH für „Plan-At-Home“, also für einen Algorithmus der seine Planungen immer im Ausgangspunkt durchführt.

Zuerst noch einige Feststellungen die bei dem Beweis des strikt kompetitiven Faktors hilfreich sind:

Korollar 7.3.11.

Weil selbst die Offline-Bearbeitung auf jeden Fall eine Bearbeitungszeit benötigt, welche nicht kleiner ist als die Streckenlänge einer optimal gewählten Rundreise mit Ausgangspunkt v_0 über alle Reisepunkte zu den Anfragen einer möglichen Teilmenge σ' der Sequenz der Anfragen, gilt unabhängig von den Zeitpunkten der Bekanntgaben auf jeden Fall schon einmal:

$$T(\sigma') \leq T_{\text{Off}} \text{ mit } \sigma' \subseteq \sigma$$

Insbesondere muss wegen des allgemeinen Startes bei v_0 außerdem $\text{dist}(\omega) \leq t_n$ gelten.

Satz 7.3.12 (Kompetitiver Faktor für Algorithmus 7.3.10).

Algorithmus 7.3.10 hat für die Problemstellung der Rundreise auf der Kreislinie einen strikt kompetitiven Faktor von zwei.

Beweis zu Satz 7.3.12, analog zu [AFL⁺95].

Zum Zeitpunkt t_n werde die Anfrage σ' mit Reisepunkt v' bekannt gegeben, dadurch gilt insbesondere auch gleich

$$T_{\text{Off}} \geq t_n + \text{dist}(v'). \quad (7.3.2)$$

Der aktuelle Standort des Algorithmus 7.3.10 sei die Position ω , für die aktuelle Situation gibt es dann drei verschiedene Möglichkeiten:

- Es gilt $\omega = v_0$:

In diesem Fall benötigt der Algorithmus 7.3.10 für die Bearbeitung der gesamten Sequenz der Anfragen maximal noch eine Zeit von $T(\sigma_{\text{Rest}})$, wobei hier σ_{Rest} die Menge der Anfragen ist, welche noch nicht von diesem bearbeitet wurden. Damit ergibt sich insgesamt also

$$T_{\text{Alg}} \leq t_n + T(\sigma_{\text{Rest}}) \stackrel{\text{mit Korollar 7.3.11}}{\leq} 2T_{\text{Off}},$$

das heißt in diesem Fall ein strikt kompetitiver Faktor von zwei.

- Für $\text{dist}(v') > \text{dist}(\omega)$ folgt:

Bei dieser Konstellation kehrt der Algorithmus 7.3.10 nach Schritt (ii)(a) zum Ursprung zurück. Dabei macht es von der Systematik keinen Unterschied, ob er sich schon auf dem Rückweg befindet oder gerade bei der Bearbeitung einer Rundreise ist. Wegen der Rückkehr zum Ursprung ist der Aufwand des Algorithmus 7.3.10 durch

$$T_{\text{Alg}} \leq t_n + \text{dist}(\omega) + T(\sigma_{\text{Rest}}) \leq t_n + \text{dist}(v') + T(\sigma_{\text{Rest}}) \stackrel{\text{mit Korollar 7.3.11}}{\leq} 2T_{\text{Off}}$$

nach oben beschränkt und somit auch hier zwei als der gewünschte strikt kompetitive Faktor nachgewiesen.

- Im letzten Fall von $\text{dist}(v') \leq \text{dist}(\omega)$ greift Schritt (ii)(b):

Falls sich der Algorithmus 7.3.10 zu diesem Zeitpunkt schon wegen Schritt (ii)(a) auf dem Rückweg zum Ausgangspunkt befindet, so gilt immer noch die bei dem Abbruch getroffene Abschätzung.

Sei also R die aktuelle Rundreise der Algorithmus 7.3.10 im Moment folgt, sowie σ_I die Menge der Anfragen die von diesem seit dem Start dieser Rundreise bei der Bearbeitung ignoriert wurden. Natürlich gehört auch die neue Anfrage σ' zu dieser Menge.

Weiterhin sei t_s der Zeitpunkt zu welchem die Offline-Bearbeitung zum ersten Mal eine der Anfragen σ_s aus σ_I bearbeitet und P der kürzeste Weg von dem Reisepunkt v_s dieser Anfrage σ_s zum Ausgangspunkt v_0 , so dass dabei alle Reisepunkte der Anfragen aus σ_I angefahren werden. Die gesamte Bearbeitungszeit der Offline-Bearbeitung lässt sich dann über

$$T_{\text{Off}} \geq t_s + |P| \quad (7.3.3)$$

nach unten abschätzen.

Für die maximale Endzeit der optimalen Rundreise R mit gesamter Länge $L(R)$ des Algorithmus 7.3.10 mit den Anfragen σ_R kann dann zum Zeitpunkt t_s folgende Oberschranke aufgestellt werden:

$$T_{\text{Alg}} \leq t_s + L(R) - \text{dist}(v_s) = t_s + T(\sigma_R) - \text{dist}(v_s) \quad (7.3.4)$$

Diese Oberschranke ergibt sich dadurch, dass der Algorithmus um die Anfrage s zum Zeitpunkt t_s ignorieren zu können auf dieser Rundreise schon eine Entfernung von $\text{dist}(v_s)$ zum Ausgangspunkt zurückgelegt haben muss. Aus der Oberschranke (7.3.4) zum Zeitpunkt t_s kann, durch das Hinzufügen einer optimalen Bearbeitungszeit für eine Rundreise durch die Reisepunkte der ignorierten Anfragen σ_I ,

$$T_{\text{Alg}} \leq t_s + T(\sigma_R) - \text{dist}(v_s) + T(\sigma_I) \quad (7.3.5)$$

eine Oberschranke für die gesamte Bearbeitungszeit des Algorithmus 7.3.10 gewonnen werden. Da außerdem

$$T(\sigma_I) \leq \text{dist}(v_s) + |P| \quad (7.3.6)$$

gilt, kann (7.3.5) fortgesetzt werden zu:

$$\begin{aligned}
 T_{Alg} &\leq t_s + T(\sigma_R) - \text{dist}(v_s) + T(\sigma_I) \\
 &\stackrel{\text{mit (7.3.6)}}{\leq} t_s + T(\sigma_R) - \text{dist}(v_s) + \text{dist}(v_s) + |P| \\
 &= t_s + T(\sigma_R) + |P| \stackrel{\text{mit (7.3.3) und Korollar 7.3.11}}{\leq} 2T_{\text{Off}}
 \end{aligned}$$

Das heißt auch in diesem Fall wird der gewünschte strikt kompetitive Faktor erreicht. $\square$

7.4. Modellierung als gemischtganzzahliges lineares Problem

Das Optimierungsproblem 7.1.1 lässt sich als ein gemischtganzzahliges lineares Problem beschreiben. Dessen Restriktionen gestalten sich wie folgt:

Lemma 7.4.1 (Gemischtganzzahliges lineares Problem für die Lösung der Offline-Bearbeitung der Problemstellung der Rundreise auf der Kreislinie mit Richtungswechsel).

Das folgende gemischtganzzahlige lineare Optimierungsproblem modelliert die Problemstellung der Rundreise auf der Kreislinie mit Richtungswechsel unter Einhaltung der für die jeweiligen Anfragen σ_a vorgegebenen Zeitpunkte t_a für deren Bekanntgabe:

$$\min \quad T \tag{7.4.1a}$$

unter:

$$A := \{1, \dots, n\}, \quad N := \{1, \dots, n\} \tag{7.4.1b}$$

$$\sum_{m \in N} w_{a,m} = 1 \quad \forall a \in A \tag{7.4.1c}$$

$$\sum_{a \in A} w_{a,m} = 1 \quad \forall m \in N \tag{7.4.1d}$$

$$z_m \geq \sum_{a \in A} t_a w_{a,m} \quad \forall m \in N \tag{7.4.1e}$$

$$z_1 \geq \sum_{a \in A} \text{dist}(v_a) \cdot w_{a,1} \tag{7.4.1f}$$

$$z_{m+1} \geq z_m + (w_{a_1,m} + w_{a_2,m+1} - 1) \cdot \text{dist}(a_1, a_2) \quad \forall m < n, a_1 \neq a_2 \in A \quad (7.4.1g)$$

$$T \geq z_n + \sum_{a \in A} \text{dist}(v_a) \cdot w_{a,n} \quad (7.4.1h)$$

$$0 \leq z_m \leq 2\pi \cdot n, \quad \forall m \in N \quad (7.4.1i)$$

$$w_{a,m} \in \{0, 1\} \quad \forall (a, m) \in A \times N \quad (7.4.1j)$$

Bemerkung 7.4.2 (Erläuterungen zu der Modellierung aus Lemma 7.4.1).

Da das lineare gemischtganzzahlige Modell relativ groß ist und komplexere Restriktionen enthält, sollen diese hier ausführlicher beschrieben werden.

(7.4.1b): Alle Anfragen sind in der Menge A enthalten. Zu jeder einzelnen Anfrage a gehört ein Zeitpunkt der Bekanntgabe t_a und ein Reisepunkt v_a . Die Menge N dient nur zur Unterscheidung ob ein Index als Anfrage oder als Nummerierung gedacht ist.

(7.4.1i): Jede reelle Variable z_m ist die Ankunftszeit am m -ten Reisepunkt innerhalb der optimalen Reiseroute.

(7.4.1j): Die binäre Variable $w_{a,m}$ bestimmt ob die Anfrage a das m -te Reiseziel ist ($w_{a,m} = 1$) oder nicht ($w_{a,m} = 0$).

(7.4.1a): Das zu minimierende Hauptziel ist die Gesamtzeit T der Rundreise.

(7.4.1c): Pro Auftrag wird genau eine Nummer vergeben.

(7.4.1d): Jeder Nummer wird genau ein Auftrag zugeordnet.

(7.4.1e): Diese Restriktionen garantieren, dass jede Anfrage erst nach deren Bekanntgabe zum Zeitpunkt t_a erfüllt werden kann.

(7.4.1f): Zusätzlich zu (7.4.1e) wird verlangt, dass das erste Reiseziel erst nach der Bewältigung des kürzesten Weges vom Ausgangspunkt zu diesem erfüllt wird.

(7.4.1g): Die Differenz der Ankunftszeiten zweier aufeinander folgender Reisepunkte in der Rundreise muss mindestens größer gleich der Zeit zur Bewältigung der Strecke auf der Kreislinie zwischen den beiden Reisepunkten sein.

(7.4.1h): Die Gesamtreise endet nach Erreichen des letzten Reisepunktes zum Zeitpunkt z_n plus der Zeit für die Rückkehr zum Ausgangspunkt.

7.5. Mögliche Erweiterungen für die Rundreise auf der Kreislinie

In diesem Abschnitt soll es um Möglichkeiten zur Erweiterung der Problemstellung Rundreise auf der Kreislinie aus Optimierungsproblem 7.1.1 gehen. Diese Erweiterungen stammen alle aus der historisch gewachsenen Entwicklung der allgemeinen Rundreise-Probleme und sollen hier nur kurz dargestellt werden:

(i) **Einbau von Transportaufträgen:**

Das bedeutet, dass eine Anfrage nicht mehr schon dann als bearbeitet gelten kann, wenn nach ihrer Bekanntgabe der zugehörige Reisepunkt besucht wurde, sondern hier gibt es jetzt noch einen sogenannten Zielpunkt, zu welchem ein vom Reisepunkt mitzunehmender Gegenstand, oder auch eine Person, geliefert werden muss.

Eine zusätzliche Ausbaustufe dieser Problematik ergibt sich durch eine Begrenzung der Kapazität der Mitnahme, wobei noch unterschieden werden kann, ob ein einmal aufgenommener Gegenstand entweder nur noch an dem dazu gehörenden Zielpunkt abgeladen werden darf oder auch einfach während des Transports an beliebiger anderer Stelle zwischengelagert werden kann.

Die Online-Optimierung von Transportproblemen, in der Fachliteratur als „Dial-a-Ride“-Probleme bezeichnet, findet sich zum Beispiel in [AKR00] oder auch in [HKRW01].

(ii) **Einbau eines Verfallsdatums, also einer maximalen Verzögerung:**

Das heißt für jede Anfrage existiert nicht nur ein Zeitpunkt, ab welchem diese überhaupt erst bearbeitet werden darf, sondern auch noch ein Verfallszeitpunkt, bis zu welchem die Anfrage bearbeitet werden sollte. Wurde eine Anfrage zum Zeitpunkt des Verfallsdatums noch nicht bearbeitet, so entfällt sie ganz.

Dadurch wird entweder eine Veränderung der Zielfunktion, wie zum Beispiel durch den Einbau einer Straffunktion für nicht bearbeitete Anfragen, oder eine Modifikation der Problemstellung, etwa mit der neuen Zielsetzung in einer bestimmten Zeit möglichst viele Anfragen zu bearbeiten, oder eine erzwungene Ausführung notwendig.

Bei einer Anwendung von Anfragen mit zu den Reisepunkten unterschiedlichen Zielpunkten, kann ebenfalls auch noch eine maximale Ladezeit der Gegenstände, also eine Beschränkung der Zeit die zwischen der Abfahrt beim Reisepunkt der zugehörigen Anfrage und der Ankunft beim Zielpunkt vergangen sein darf, eingebaut werden.

Eine Änderung der Problemstellung durch eine Unterbrechung der Kreislinie an einer Stelle womit ein komplettes Umfahren der Kreislinie unmöglich wird, führt auf die schon

in der Literatur behandelte Problematik der Rundreise auf der Zahlengeraden (diese findet sich zum Beispiel bei [Kru02, Kapitel 3]).

In den nächsten beiden Kapiteln wird untersucht, wie sich die Verwendung von Informationsvorlauf nach Definition 6.2.1 auf die Bearbeitung der Problemstellung aus Optimierungsproblem 7.1.1, also für Rundreisen auf der Kreislinie, auswirkt. Dabei wird dann nicht mehr wie in diesem Kapitel mit beliebigen Zeitpunkten der Bekanntgabe gearbeitet, sondern es gilt, dass immer eine konstante Anzahl von Anfragen bekannt ist. Dies führt dazu, dass sobald eine Anfrage bearbeitet wurde, die nächste neue Anfrage bekannt gegeben werden muss und damit ist eine beliebige Festlegung der Zeitpunkte der Bekanntgaben der Anfragen nicht mehr möglich.

8. Gerichtete Rundreise auf der Kreislinie

In diesem Kapitel geht es um eine Vereinfachung der Rundreise auf der Kreislinie mit Richtungswechsel aus Optimierungsproblem 7.1.1. In der eigentlichen Problemstellung ist es möglich, während der Bearbeitung der Anfragen die Richtung der Fahrt auf der Kreislinie zu ändern. Durch diese Möglichkeit der Richtungsänderung ergeben sich bestimmte Eigenheiten bei der Berechnung der Erwartungswerte:

- Die möglichen Schrittweiten zwischen dem aktuellen Standort und einer der Anfragen, welche nicht gleich zum Zeitpunkt ihrer Bekanntgabe bearbeitet wurden, beinhalten mehrere Fallunterscheidungen, wobei nicht immer jeder der unterschiedlichen Fälle vorhanden sein muss (dies ist in Abbildung 9.5.1 auf Seite 166 anschaulich zu sehen).
- Bei der Betrachtung der Schrittweite im Schritt t muss die ganze Historie der vorherigen Schritte eingearbeitet werden. Dies führt nicht nur dazu, dass einerseits die verschiedenen möglichen Bearbeitungsreihenfolgen unterschieden werden müssen, sondern andererseits auch noch bei einer vorgegebenen Reihenfolge der Bearbeitung in mehrere Situationen mit unterschiedlichen Laufrichtungen und verschiedenen Abständen zur letzten ausgewählten Anfrage unterschieden werden muss.

Diese allgemeinere Problematik wird in Kapitel 9 genauer behandelt. Alles im allen ergeben sich bei der Weiterführung einer schon vorgegebenen Reise über die Reisepunkte von t Anfragen mit einer nach Definition 6.2.1 definierten Bearbeitungsinformation von k Anfragen $2k$ verschiedene Möglichkeiten den nächsten Schritt zu unternehmen. Dabei ergibt sich der Vorfaktor von zwei daraus, dass einerseits in der bisherigen Richtung weitergefahren werden kann und andererseits auch in der umgekehrten Richtung.

Bis einschließlich zum Schritt t ergeben sich so insgesamt $(2k)^t$ unterschiedliche mögliche Bearbeitungsreihenfolgen. Noch nicht eingerechnet sind dabei die Unterscheidungen, welche sich innerhalb der Berechnungen aus der unterschiedlichen Struktur der Positionen der Reisepunkte der Anfragen gegenüber dem aktuellen Standort ergeben.

Aufgrund dieser Anzahl an Möglichkeiten, aber vor allem auch wegen der sich aus den Fallunterscheidungen bei den Wahrscheinlichkeiten zu den möglichen Schrittweiten zwischen dem Standort und dem Reisepunkt einer noch nicht bearbeiteten Anfrage ergebenden Problematik wird sich dieses Kapitel auf eine eingeschränkte Problemstellung konzentrieren.

Für das weitere Vorgehen sind noch die folgenden grundlegenden Aussagen notwendig:

- Der Informationsvorlauf bezieht sich auf den in Abschnitt 6.2 definierten Informationsvorlauf, also auf die Bearbeitungsinformation (siehe Definition 6.2.1), und orientiert sich wie in Unterabschnitt 5.2.2 definiert an der Reihenfolge der Anfragen innerhalb der Sequenz. Das heißt es sind bei jeder Entscheidungsfindung des Algorithmus k Anfragen bekannt, welche alle bearbeitet werden können.
- Die für die Analyse zu verwendende Startmethode ist der in Definition 6.2.3 eingeführte Vollstart, eine maximale Verzögerung wird dabei nicht vorgegeben.

8.1. Grundlagen zur gerichteten Rundreise auf der Kreislinie

Aus den vorher in der Einführung des Kapitels aufgeführten Gründen soll nun eine vereinfachte Variante der Rundreise auf der Kreislinie eingeführt werden. Da die Probleme bei der Rundreise auf der Kreislinie mit Richtungswechsel dadurch entstehen, dass während der Bearbeitung der Sequenz von Anfragen die Laufrichtung geändert werden kann, wird diese Möglichkeit innerhalb dieses Kapitels verboten und somit entsteht die folgende neue Problemstellung:

Optimierungsproblem 8.1.1 (Gerichtete Rundreise auf der Kreislinie).

Die *gerichtete Rundreise* auf der Kreislinie unterscheidet sich von der Rundreise auf der Kreislinie mit Richtungswechsel aus Optimierungsproblem 7.1.1 nur dadurch, dass im Gegensatz zu dieser nur eine Laufrichtung entlang der Kreislinie erlaubt ist. Ohne Beschränkung der Allgemeinheit wird dabei die Bewegung im Uhrzeigersinn verboten.

Im weiteren Verlauf werden in dieser Arbeit die Begriffe gerichtete Rundreise auf der Kreislinie für das Optimierungsproblem 8.1.1 und Rundreise auf der Kreislinie mit Richtungswechsel für das Optimierungsproblem 7.1.1 verwendet.

Durch die in Optimierungsproblem 8.1.1 festgelegte Einschränkung der Laufrichtung ist bei einer Optimierung ohne Wartezeiten die einzige sinnvolle Möglichkeit zur Auswahl der nächsten Anfrage für die weitere Route das Vorgehen nach der Greedy-Methode. Damit ergibt sich der Online-Algorithmus zu:

Algorithmus 8.1.2 (Greedy für die gerichtete Rundreise auf der Kreislinie).

- (i) Die Rundreise beginnt und endet in v_0 , dem Punkt des Kreises mit der Winkelkoordinate 0.
- (ii) Bei der Entscheidung welches Ziel als nächstes angefahren wird, ist immer ein Informationsvorlauf von k Anfragen gegeben (außer in der Sequenz sind nicht mehr genügend Anfragen vorhanden). Aus den zur Verfügung stehenden Anfragen wird

diejenige gewählt, deren Reisepunkt dem aktuellen Standpunkt in der Laufrichtung am nächsten liegt und diese bearbeitet.

Nach der Bearbeitung wird die Information zu einer neuen Anfrage bekannt gegeben und mit diesem Schritt von neuem begonnen.

Nach der Einschränkung der Laufrichtung muss natürlich auch die in Definition 7.3.1 definierte Distanzfunktion angepasst werden.

Definition 8.1.3 (Distanzfunktion für die gerichtete Rundreise auf der Kreislinie).

Die *Distanzfunktion* vom Punkt v_i zum Punkt v_j ergibt sich unter der Verwendung des Koordinatensystems aus Abbildung 7.1.1 und mit Bewegungsrichtung entlang der Messung der Winkelkoordinate zu:

$$\text{dist}(v_i, v_j) := \begin{cases} v_j - v_i & \text{falls } v_j \geq v_i \\ 2\pi - v_j + v_i & \text{falls } v_j < v_i \end{cases}$$

Bemerkung 8.1.4 (Belegung des anfänglichen Informationsvorlaufs).

Algorithmus 8.1.2 wie auch der später definierte Algorithmus 9.1.1 gehören zu der Sorte von Algorithmen, welche bei der Online-Bearbeitung von Rundreisen immer nur das nächste Ziel aus dem Informationsvorlauf auswählen, ohne für den restlichen Informationsvorlauf eine vorläufige Planung zu erstellen. Solange dann für die Problemstellung keine maximale Verzögerung, wie in Definition 6.2.4 angesprochen, eingeführt wurde, so ist die Art und Weise der ersten Belegung des Informationsvorlaufs nicht relevant. Es wird nämlich immer nur aus der Gesamtheit der Anfragen im Informationsvorlauf eine der möglichen Anfragen ausgewählt, wobei deren Zeitpunkt der Bekanntgabe unerheblich ist. Die restlichen $k - 1$ Anfragen aus dem Informationsvorlauf werden bis zur Auswahl der nächsten Anfrage nicht weiter beachtet.

Wird allerdings als Grundlage für einen Algorithmus zur Bearbeitung eine Heuristik mit kompletter Planung, also ein Verfahren welches neue Punkte immer in eine vorher vollständig erstellte Rundreise einbaut (ein Überblick zu einigen der infrage kommenden Verfahren für Rundreisen kann in [Bor01, Abschnitt 26.2] gefunden werden), verwendet, so wird für die Anfragen aus dem ersten Informationsvorlauf eine fiktive Rundreise erstellt. Aufbauend auf dieser ersten Rundreise würde der Algorithmus dann die neu zum Informationsvorlauf hinzukommenden Anfragen sukzessive einbauen ohne den bis zu deren Bekanntgabe schon realisierten Teil der Rundreise zu verändern. Das bedeutet, dass bei der schon vorhandenen Rundreise die neue Anfrage nicht zwischen zwei benachbarte Anfragen, welche beide nicht mehr im Informationsvorlauf enthalten sind, eingebaut werden darf.

Bei dieser Art der Einarbeitung der neu bekannt gewordenen Anfragen macht die Methodik der ersten Belegung des ersten Informationsvorlaufs dann, auch ohne eine maximale Verzögerung, doch einen Unterschied:

- Bei einer sukzessiven Bekanntgabe werden die Anfragen eine nach der anderen, in der nicht veränderbaren Reihenfolge aus der Sequenz der Anfragen, in die anfängliche Rundreise der vorherigen Anfragen eingefügt.

Damit wird insbesondere bei der Einfügung immer nur die Lage einer Anfrage gegenüber der bereits geplanten Rundreise betrachtet.

- Werden dagegen alle Anfragen gleichzeitig bekannt, so hat der Algorithmus die zusätzliche Möglichkeit unter den noch nicht eingeplanten Anfragen des ersten Informationsvorlaufs eine für die Einfügung in die bis dahin geplante Rundreise auszuwählen.

Somit kann dieser bei der Einfügung die Lage aller noch nicht eingeplanter Anfragen aus dem Informationsvorlauf gegenüber der bereits geplanten Rundreise betrachten. Diese so gewonnene Rundreise sollte in den meisten Fällen kürzer sein als die sukzessiv gewonnene Rundreise, da durch die Auswahl der Anfrage grundsätzlich ein größerer Spielraum für die Planung besteht.

Alternativ kann auch einfach die optimale Lösung für eine Rundreise durch die Anfragen des Informationsvorlaufs als erste fiktive Rundreise verwendet werden, dies hätte zusätzlich den positiven Effekt, dass sich unabhängig von der Entstehung des ersten Informationsvorlaufs immer das gleiche Ergebnis oder zumindest eine von der Dauer der Reise identische Rundreise ergibt.

8.2. Simulation einer gerichteten Rundreise auf der Kreislinie

Um die Problemstellung etwas deutlicher zu erläutern wird nun erst einmal ein Beispiel durchgeführt.

Beispiel 8.2.1 (Beispiel zur gerichteten Rundreise auf der Kreislinie).

Gegeben sei eine Sequenz von zwanzig Anfragen und eine Bearbeitungsinformation von fünf Anfragen, das heißt es sind immer die Reisepunkte von fünf bisher noch nicht bearbeiteten Anfragen zur Bearbeitung verfügbar. Das schematische Vorgehen bei der Online-Bearbeitung unter Verwendung des Algorithmus 8.1.2 wird nun anhand der nachfolgenden Graphiken dargestellt.

Allgemein gelten für die Graphiken folgende Konventionen:

- Reisepunkte von noch nicht bekannt gegebenen Anfragen werden als Punkte auf der äußeren Kreislinie dargestellt.
- Der große Punkt kennzeichnet den aktuellen Standort.

- Dunkle Punkte auf der inneren Kreislinie sind die Reisepunkte der Anfragen aus dem Informationsvorlauf, dabei wird der Reisepunkt der neu hinzugekommenen Anfrage immer durch einen schraffierten Punkt auf dieser Kreislinie dargestellt.
- Die gestrichelte Linie entspricht der Wegstrecke, welche in diesem Schritt zurückgelegt wird, die Bewegung erfolgt entgegen dem Uhrzeigersinn.
- Reisepunkte von schon bearbeiteten Anfragen werden als helle Punkte auf der inneren Kreislinie dargestellt. Neben jedem schon bearbeiteten Punkt steht ebenfalls die Position der zugehörigen Anfrage in der entstehenden Rundreise.

Da der Ablauf des Algorithmus durch die Einschränkung auf eine Laufrichtung weitgehend selbst erklärend ist, – es wird einfach immer die entgegen dem Uhrzeigersinn nächstgelegene Anfrage ausgewählt, – wird zwischen den Graphiken auf weitergehende Erklärungen verzichtet.

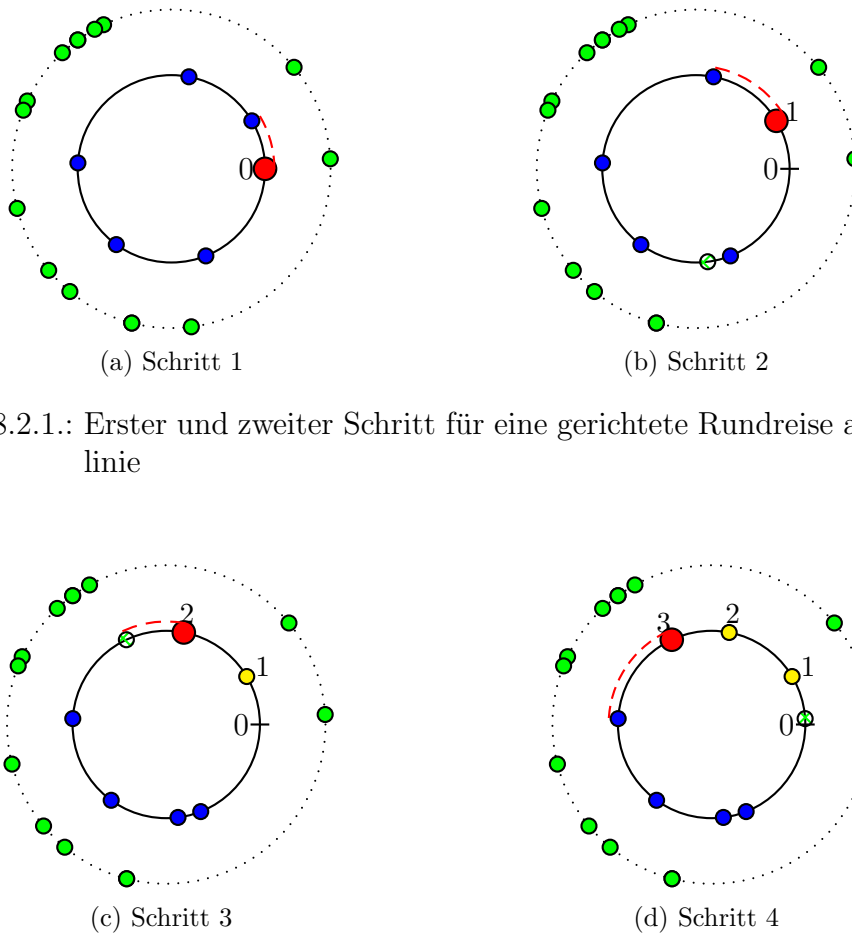
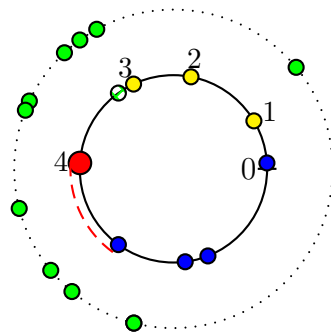
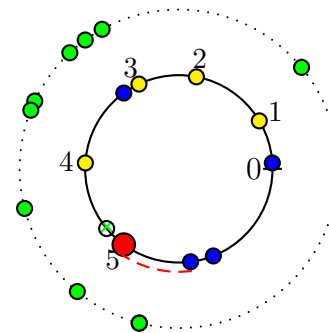


Abbildung 8.2.1.: Erster und zweiter Schritt für eine gerichtete Rundreise auf der Kreislinie

Abbildung 8.2.1 (Fortsetzung): Dritter und vierter Schritt für eine gerichtete Rundreise auf der Kreislinie

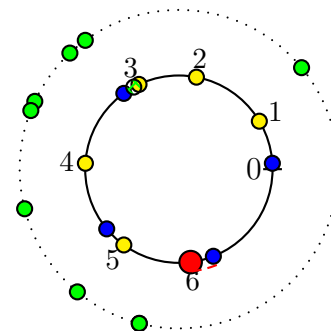


(e) Schritt 5

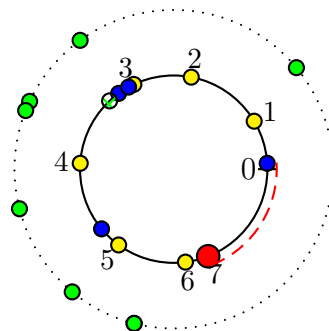


(f) Schritt 6

Abbildung 8.2.1 (Fortsetzung): Fünfter und sechster Schritt für eine gerichtete Rundreise auf der Kreislinie

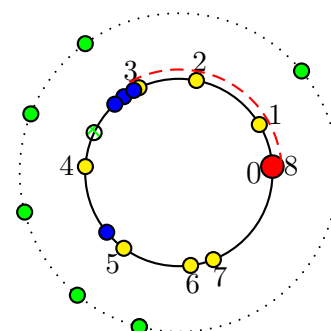


(g) Schritt 7

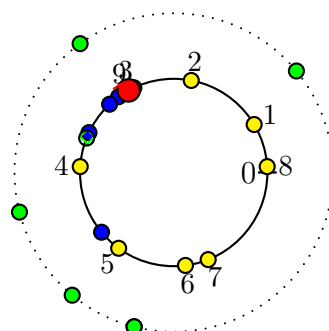


(h) Schritt 8

Abbildung 8.2.1 (Fortsetzung): Siebter und achter Schritt für eine gerichtete Rundreise auf der Kreislinie



(i) Schritt 9



(j) Schritt 10

Abbildung 8.2.1 (Fortsetzung): Neunter und zehnter Schritt für eine gerichtete Rundreise auf der Kreislinie

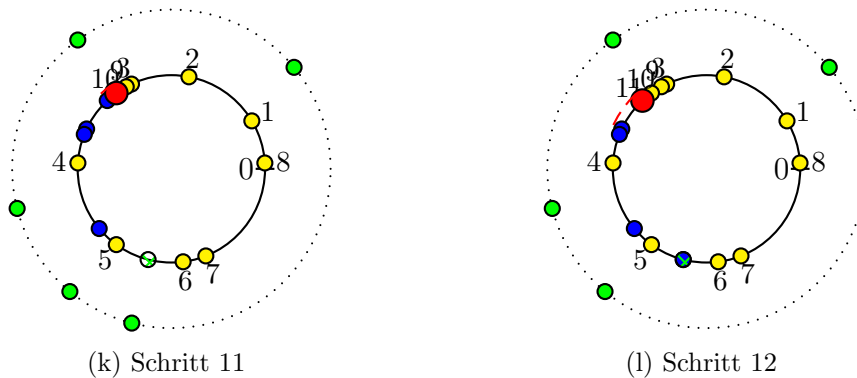


Abbildung 8.2.1 (Fortsetzung): Elfter und zwölfter Schritt für eine gerichtete Rundreise auf der Kreislinie

8.3. Empirische Analyse

Zuerst einmal wird die Bearbeitung der Problemstellung aus Optimierungsproblem 8.1.1, also der gerichteten Rundreise auf der Kreislinie, mit dem Algorithmus 8.1.2 einer empirischen Auswertung über die durchschnittliche Schrittweite unterzogen.

Grundlagen zur Auswertung

Für die empirische Untersuchung werden Schrittweiten betrachtet, welche aus der Auswertung von 10.000 Dateien mit jeweils einer Sequenz von 50.000 Anfragen unter Verwendung von unterschiedlichen Werten für den Informationsvorlauf k , wobei k zwischen zwei und dreihundert liegt, resultieren. Gerechnet wird mit einem Dual-Core-Prozessor sowie 4 GB RAM unter Verwendung von Kubuntu in der 64-Bit Variante. Für die Auswertung wurden die in Abschnitt A.2 beschriebenen Programme verwendet. Die Reisepunkte der Anfragen liegen dabei jeweils unabhängig gleichverteilt in dem Intervall $[0, 2]$.

Zur besseren Verständlichkeit der Zahlenwerte und für eine einfachere Darstellung wird bei allen empirischen Auswertungen der Umfang der Kreislinie auf zwei Einheiten skaliert. Somit sind also alle in diesem Abschnitt verwendeten Zahlenwerte zu den einzelnen Schrittweiten nicht die tatsächlich gefahrenen Längen auf der Kreislinie, sondern stellen nur den Anteil der gefahrenen Länge gegenüber der Hälfte der Kreislänge, also von π , dar. Dies gilt natürlich auch analog für den Abschnitt 9.3.

Mathematische Grundlagen

Für die Durchführung der nichtlinearen Regressionsanalysen wird die Statistik-Software [Tea09, R] mit dem darin enthaltenen Paket *nlstools* [BD09] unter Anwendung der Methode *nls* (dies steht für „Nonlinear Least Squares“, also für die Methode der kleinsten Fehlerquadrate) verwendet. Über diese Methode können die Grundlagen des nichtlinearen Modells festgelegt werden, so erzeugt zum Beispiel der Aufruf

```
nls <- nls(V2 ~ alpha/(beta+V1),data,start=list(beta=1,alpha=1));
```

ein nichtlineares Modell zum Datensatz `data`, in dessen zweiter Spalte (V2) die relevanten empirischen Daten y_i in Beziehung zu den in der ersten Spalte (V1) festgelegten Ausprägungen x_i des Merkmals, stehen. Die zu untersuchende Beziehung zwischen dem Merkmal und den empirischen Daten mit den beiden Parametern α und β ist hier als

$$y_i \approx f(x) := \frac{\alpha}{\beta + x_i}$$

festgelegt, als Startwerte für die zu bestimmenden Parameter werden die anfänglichen Belegungen $\alpha = 1$ und $\beta = 1$ verwendet. Konkret berechnet die Methode der kleinsten Fehlerquadrate ausgehend von den anfänglichen Belegungen für die Parameter α und β derartige Ausprägungen, dass die Summe

$$RSS = \sum_{i=1}^n (y_i - f(x_i))^2 \text{ („Residual Sum of Squares“)}$$

der quadratischen Abstände zwischen den empirischen Daten und der durch das Modell definierten funktionalen Werte über alle gegebenen Ausprägungen minimal wird. Nach dem Aufruf von `overview(nls)` liefert [Tea09, R] die bei der Regressionsanalyse festgestellten Ergebnisse, diese werden in den später erfolgenden Analysen größtenteils, wie in Tabelle 8.3.1 zu sehen, tabellarisch dargestellt. Der Schätzwert ist dabei die bei

Schätzwert	Standardfehler	t-Wert	P (> t)
------------	----------------	--------	-----------

Tabelle 8.3.1.: Standarddarstellung der Ergebnisse zur Regressionsanalyse

der Minimierung der Summe RSS gefundene Ausprägung des zugehörigen Parameters, der Standardfehler beschreibt die Standardabweichung bei dieser Schätzung. Mit dem t -Wert wird schließlich die Relevanz des Parameters für das Modell untersucht, je höher dieser Wert, desto signifikanter ist der Parameter.

Für eine Auswertung des Modells in Unterabschnitt 8.3.1, allerdings nur über eine Teilmenge der Daten, ist die Ausgabe von [Tea09, R] in Quellcode 8.1 zu sehen.

Zusätzlich zu den tabellarisch dargestellten Ergebnissen liefert [Tea09, R] auch gleich die folgenden Daten:

- Den Wert von RSS .
- Ein asymptotisches Konfidenzintervall für die jeweiligen Parameter.
- Eine Korrelationsmatrix zwischen den einzelnen Parametern.
- Die Anzahl der Iterationen die der Algorithmus zur Berechnung der geschätzten Werte benötigt hat und die dabei erreichte Toleranz.

Quellcode 8.1: Beispielhafte Originalausgabe von [Tea09, R]

```

Formula: V2 ~ alpha/V1

Parameters:
      Estimate Std. Error t value Pr(>|t|)
5 alpha 1.000e+00  6.967e-05  14356   <2e-16 ***
-----
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.124e-05 on 62 degrees of freedom
10
Number of iterations to convergence: 1
Achieved convergence tolerance: 9.168e-11

-----
15 Residual sum of squares: 1.05e-07

-----

Asymptotic confidence interval:
      2.5%      97.5%
20 1 1.000005 1.000284

-----

Correlation matrix:
      alpha
25 alpha      1
-----

```

- Die Freiheitsgrade und die Standardabweichung der Residuen $y_i - f(x_i)$.

Das Bestimmtheitsmaß R^2 für das Modell berechnet sich anschließend über

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - f(x_i))^2}{\sum_{i=1}^n y_i^2} = 1 - \frac{RSS}{\sum_{i=1}^n y_i^2}. \quad (8.3.1)$$

Je näher dieses Maß an der eins liegt, um so genauer beschreibt das Modell den Zusammenhang zwischen den vorher festgelegten Ausprägungen des variablen Anteils und den tatsächlich auftretenden empirischen Daten.

8.3.1. Durchschnittliche Schrittweite

Als Grundlage für die betrachtete durchschnittliche Schrittweite dient die Entfernung, welche zwischen dem Start im Ausgangspunkt v_0 mit der Bearbeitung der ersten ausgewählten Anfrage bis zur Rückkehr in v_0 nach der Bearbeitung der letzten Anfrage zurückgelegt wurde und die Anzahl der besuchten Punkte.

Zusätzlich zu den empirischen durchschnittlichen Schrittweiten, in der Graphik als Punkte dargestellt, ist in der Abbildung 8.3.1 der Funktionsgraph für $\frac{1}{k}\pi$ (das ist die untere Kurve) enthalten. Wegen der Skalierung der Zahlenwerte entspricht dieser dem Funktionsverlauf von $\frac{1}{k}$. Die Daten der empirischen Ergebnisse sind in Tabelle B.1.1.

Die Kurve zu $\frac{1}{k}$ und die Punkte zu den empirisch berechneten durchschnittlichen Schrittweiten in der Abbildung 8.3.1 lassen sich nicht mehr unterscheiden. Dies wird auch bei der Durchführung einer nichtlinearen Regressionsanalyse durch [Tea09, R] mit dem Modell $\frac{\alpha}{k}$ bestätigt. Die Regressionsanalyse ergibt nämlich, ausgehend von dem Startwert $\alpha = 1$, das in Tabelle 8.3.2 zusammengefasste Ergebnis.

Schätzwert	Standardfehler	t-Wert	P ($> t $)
1,000	$5,159 \cdot 10^{-5}$	19386	$< 2 \cdot 10^{-16}$

Tabelle 8.3.2.: Ergebnis zur nichtlinearen Regression der durchschnittlichen Schrittweite für die gerichtete Rundreise auf der Kreislinie

Die Summe der quadrierten Abweichungen RSS zwischen den empirischen durchschnittlichen Schrittweiten und der gefundenen Regressionsfunktion bleibt dabei unter $2,06 \cdot 10^{-7}$. Daraus ergibt sich ein Bestimmtheitsmaß von $R^2 = 0,9999997$.

Bemerkung 8.3.1 (Unterscheidung zwischen der Bearbeitung mit verbleibenden Anfragen und immer neuen Anfragen).

Die oberste Linie aus Abbildung 8.3.1 zu $\frac{2}{k+1}\pi$, dies ist nach Satz 8.4.1 die Schrittweite des ersten Schritts mit k identischen gleichverteilten Anfragen auf der Kreislinie, zeigt den Verlauf der Schrittweite im ersten Schritt für einen größer werdenden Informationsverlauf.

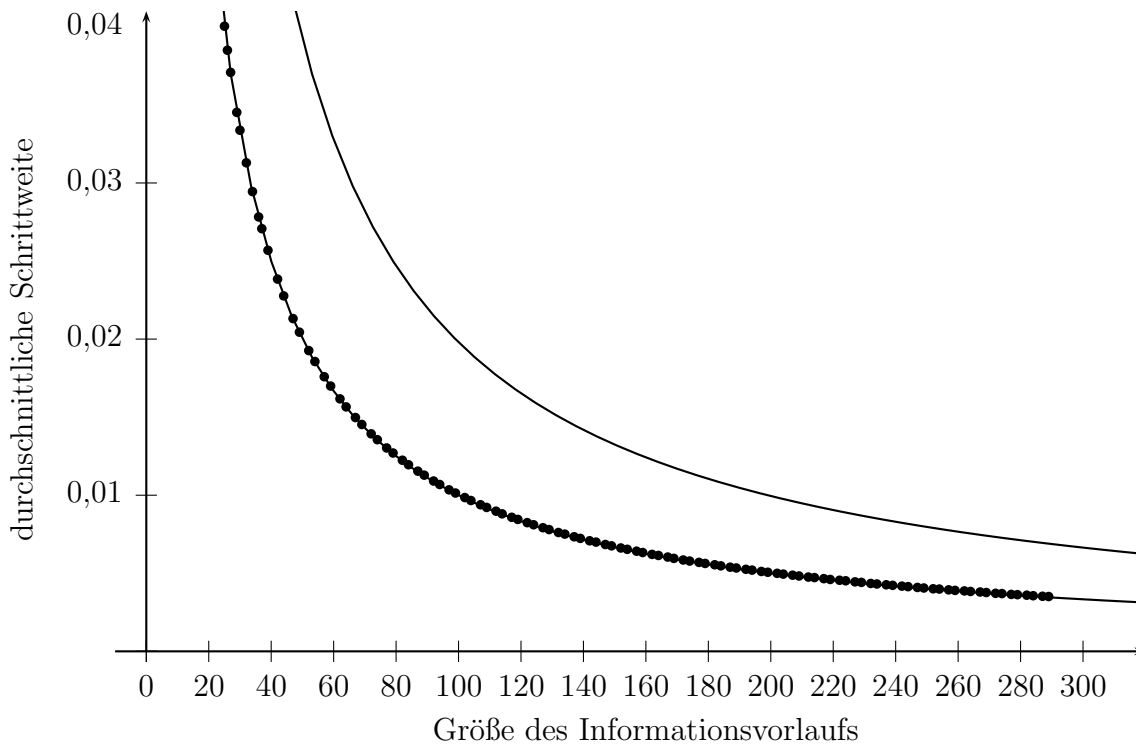


Abbildung 8.3.1.: Empirische Werte für die durchschnittliche Schrittweite der gerichteten Rundreise auf der Kreislinie bei unterschiedlichem Informationsvorlauf

Würden bei der vorgegebenen Problemstellung nicht immer die unbearbeiteten Anfragen behalten werden, sondern nach jeder erfolgten Auswahl einer Anfrage die restlichen in diesem Schritt nicht ausgewählten Anfragen verworfen und danach k neue Anfragen, deren Reisepunkte auf der gesamten Kreislinie gleichverteilt sind, generiert, so ergäbe sich dieser Kurvenverlauf für die erwartete Schrittweite jedes Schrittes.

Daraus, dass diese Linie immer deutlich über der anderen Linie und der Punktfolge liegt, lässt sich erkennen, dass das Verbleiben der Anfragen, und die damit verbundene Verkürzung der für die Reisepunkte der Anfragen noch möglichen Bereiche eine deutliche Verkleinerung (der Wert wird annähernd halbiert) der durchschnittlichen Schrittweite für die Online-Bearbeitung möglich macht.

Eine erweiterte Regressionsanalyse mit anderen Modellen ergab, wie in Tabelle 8.3.3 zu sehen, annähernd gleiche Ergebnisse. Dabei zeigt sich insbesondere bei dem Modell mit der Regressionsfunktion $\frac{\alpha}{(k+\beta)}$, dass der zweite Parameter β im Vergleich zum ersten Parameter α einen deutlich schlechteren t -Wert besitzt. Dies bedeutet, dass der Einfluss von β im Vergleich zu dem von α vernachlässigt werden kann.

Modell	Parameter	RSS	R^2	t -Werte
$\frac{\alpha}{k}$	$\alpha = 1,000$	$2,06 \cdot 10^{-7}$	0,9999997	19386
$\frac{1}{k^\alpha}$	$\alpha = 0,9998$	$1,46 \cdot 10^{-7}$	0,9999998	34849
$\frac{\alpha}{(k+\beta)}$	$\alpha = 1,0007021; \beta = 0,0015467$	$1,68 \cdot 10^{-7}$	0,9999997	9381,163; 5,347
$\frac{\beta}{k^\alpha}$	$\alpha = 0,9995; \beta = 0,9995$	$1,08 \cdot 10^{-7}$	0,9999998	203977; 13458

Tabelle 8.3.3.: Ergebnisse der verschiedenen Modelle zur durchschnittlichen Schrittweite für die gerichtete Rundreise auf der Kreislinie

8.3.2. Modifizierte durchschnittliche Schrittweite

Bei der Berechnung der gesamten durchschnittlichen Schrittweite in Unterabschnitt 8.3.1 wird die durchschnittliche Schrittweite aller Schritte einer Rundreise über n Stationen, dabei ist n die Größe der Sequenz der Anfragen, verwendet. Soll nun allerdings aus den empirisch gefundenen durchschnittlichen Schrittweiten auf die durchschnittliche Schrittweite einer unendlich großen Sequenz von Anfragen ohne den Start mit k Anfragen, deren Reisepunkte identisch gleichverteilt sind, geschlossen werden, so ergeben sich durch die Verwendung aller empirischen Schrittweiten zur zurückgelegten Entfernung gewisse Verfremdungen innerhalb des Ergebnisses:

- Bei den ersten $k - 1$ Schrittweiten innerhalb einer Rundreise mit Informationsvorlauf k sind auf jeden Fall immer mindestens zwei Anfragen gegeben, bei denen der noch mögliche Bereich für deren jeweiligen Reisepunkt identisch ist. Bei der Situation nach der Bearbeitung von $t \gg k$ Anfragen haben im Normalfall, außer in den ziemlich unwahrscheinlichen Fällen, dass
 - entweder bis dahin immer noch mindestens zwei der ersten k Anfragen noch nicht bearbeitet wurden,
 - oder eine Schrittweite von Null in Betracht gezogen wird,
alle dann noch nicht bearbeiteten Anfragen einen individuellen Bereich, innerhalb dessen Grenzen der jeweilige Reisepunkt liegen muss.
- Zudem stehen bei der Planung der letzten k Stationen der Rundreise nur noch weniger als k Anfragen für die Auswahl zur Verfügung, so dass sich auch dort eine Einschränkung bemerkbar macht.

Definition 8.3.2 (Bereinigte Schrittweite).

Die *bereinigte durchschnittliche Schrittweite* beschränkt sich aus den oben aufgezählten Gründen auf die durchschnittliche Entfernung, welche zwischen zwei Punkten des Abschnitts $g_k, \dots, g_{n-k+1}$ aus der durch den jeweiligen Algorithmus mit einem Informationsvorlauf von k Anfragen berechneten Rundreise, erreicht wird.

Innerhalb der, auf den bereinigten empirischen durchschnittlichen Schrittweiten zu dem Algorithmus 8.1.2 aufbauenden, Abbildung 8.3.2 lässt sich ebenfalls kein Unterschied zwischen den empirischen Punkten, diese können in Tabelle B.1.2 eingesehen werden, und der gezeichneten Linie für $\frac{1}{k}\pi$ erkennen. Die obere deutlich davon entfernte Funktion ist wieder die nach Satz 8.4.1 erwartete erste Schrittweite.

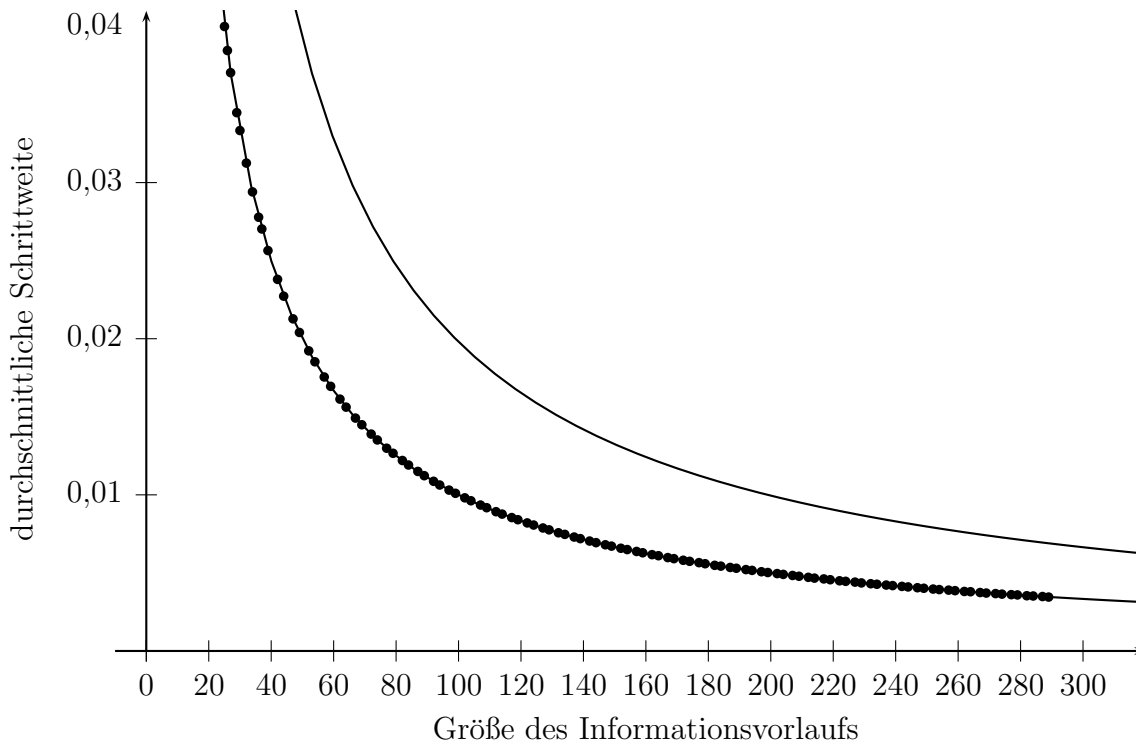


Abbildung 8.3.2.: Bereinigte empirische Werte für die durchschnittliche Schrittweite zur gerichteten Rundreise auf der Kreislinie

Die Regressionsanalyse ebenfalls wieder mit dem Modell $\frac{\alpha}{k}$ ergibt, wie in Tabelle 8.3.4 zu sehen, einen deutlich größeren t -Wert als in Tabelle 8.3.2. Auch der Residuenfehler in Form der Summe der quadratischen Abweichungen zwischen Modell und den empirischen durchschnittlichen Schrittweiten ist mit $RSS = 2,9 \cdot 10^{-10}$ um 10^3 kleiner als der Wert für die gesamte durchschnittliche Schrittweite.

Schätzwert	Standardfehler	t -Wert	$P(> t)$
1,000	$1,936 \cdot 10^{-6}$	516449	$< 2 \cdot 10^{-16}$

Tabelle 8.3.4.: Ergebnis zur nichtlinearen Regression der bereinigten durchschnittlichen Schrittweite für die gerichtete Rundreise auf der Kreislinie

Die durch die Bereinigung der empirischen Schrittweiten für die Berechnung erlangten Veränderungen in den Ergebnissen zur Regressionsanalyse, und die dadurch erreichte bessere Abbildung durch das Modell liefern noch einmal eine klare Rechtfertigung für dieses Vorgehen. Das dabei erreichte Bestimmtheitsmaß R^2 wird von der Statistik-Software [Tea09, R] als 1 angezeigt.

Anders als beim nicht bereinigten Fall liefert diesmal eine Regressionsanalyse mit einem exponierten Modell (siehe Tabelle 8.3.5) als Funktion der Regression exakt das gleiche Ergebnis wie das rationale Modell. Beim Modell $\alpha/(k + \beta)$ ergab sich mit dem anfänglichen minimalen Faktor von 2^{-10} für die Schrittweiten ein rascher Abbruch, da der Faktor zu klein war. Nach einer Reduktion des Faktors auf 10^{-12} wurde zwar immer noch kein reguläres Ende des Verfahrens erreicht, allerdings hatten sich die bis zum Abbruch gefundenen Lösungen immerhin auf die in Tabelle 8.3.5 eingetragenen Werte stabilisiert.

Modell	Parameter	RSS	R^2	t -Werte
$\frac{\alpha}{k}$	$\alpha = 1,000$	$2,28 \cdot 10^{-10}$	1	516449
$\frac{1}{k^\alpha}$	$\alpha = 1,000$	$9,9 \cdot 10^{-11}$	1	883540
$\frac{\alpha}{(k+\beta)}$	Auswertung bricht wegen eines zu kleinen Schrittweitenfaktors ab.			
$\frac{\beta}{k^\alpha}$	$\alpha = 1,000; \beta = 1,000$	$1,83 \cdot 10^{-10}$	1	495837; 327162

Tabelle 8.3.5.: Ergebnisse der verschiedenen Modelle für die bereinigten Daten zur gerichteten Rundreise auf der Kreislinie

Damit zeigen sich auch für die exponentiellen Modelle eine deutliche Verbesserung der Ergebnisse der Regressionsanalyse durch die Bereinigung der empirisch berechneten durchschnittlichen Schrittweiten.

8.3.3. Schrittweiten für einzelne Werte des Informationsvorlaufs

Die Abbildung 8.3.3 zeigt in der Reihenfolge von links oben nach rechts unten die erreichten ersten Schrittweiten für einen Informationsvorlauf k von zwei, vier, neunundzwanzig und neunundsechzig Anfragen. Wegen der großen Anzahl der Schritte, siehe die Größe der Sequenzen am Anfang des Kapitels, musste für die graphische Darstellung eine Auswahl der Schrittweiten getroffen werden, diese beschränken sich nun auf die ersten dreihundert Schritte. Zusätzlich ist in den jeweiligen Abbildungen noch eine erwartete Trendlinie mit $f(k) = \frac{1}{k}$ eingezeichnet, diese wurde in Unterabschnitt 8.3.2 als Regressionsmodell verwendet und wird später in Unterabschnitt 8.5.1 zumindest für $k = 2$ auch noch formell bewiesen.

In allen Abbildungen ist klar erkennbar, dass die größte Schrittweite immer im allerersten Schritt auftritt, und dass in den späteren Schritten der Verlauf der empirisch

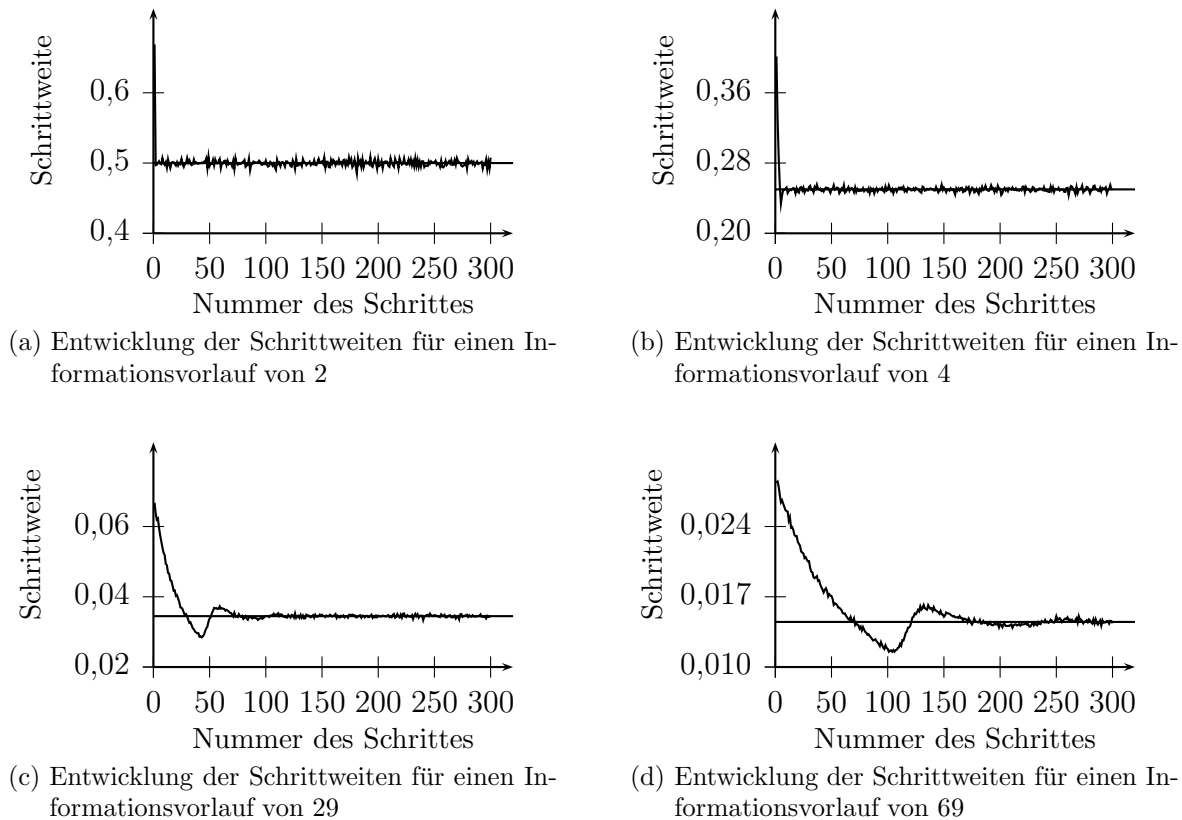


Abbildung 8.3.3.: Die Schrittweiten der ersten Schritte für einen Informationsvorlauf von 2, 4, 29 und 69 bei der gerichteten Rundreise auf der Kreislinie

erfassten Werte ziemlich nahe an der eingezeichneten Trendlinie verläuft. Bei den Abbildungen zu $k = 29$, sowie für $k = 69$ fällt außerdem auch noch deutlich auf, dass nach den k -ten Schritt die Schrittweite kurzfristig eine ziemlich geringe Größe erreicht, welche dann im späteren Verlauf nicht mehr erreicht wird.

Dieses Verhalten begründet sich darin, dass normalerweise spätestens ab dem Schritt k vermehrt die noch nicht bearbeiteten Anfragen aus dem ersten Informationsvorlauf abgearbeitet werden, da der zulässige Bereich für deren Reisepunkte dann nur noch relativ klein sein sollte. Weil dieser Effekt aber andererseits nur solange anhalten kann, bis diese Anfragen alle bearbeitet wurden, steigt nach einer kurzen Phase die Größe der Schrittweite auch wieder an, um sich danach zyklisch der Trend-Linie anzunähern.

Die zyklische Annäherung erklärt sich dabei durch eine Art Übersteuerung des Algorithmus. Das liegt daran, dass zum Beispiel Anfangs bedingt durch die Abarbeitung der Anfragen aus der ersten Bearbeitungsinformation sich die Schrittweite ziemlich verkürzt. Wegen dieser Verkleinerung der Schrittweite, werden dann aber die Bereiche der noch nicht bearbeiteten aber schon bekannt gegebenen Anfrage auch nicht im vollen Maß verkürzt. Aus dieser Tatsache resultieren, dann die daran anschließenden Vergrößerun-

gen der Schrittweite, dabei machen diese auch nicht vor der Trendlinie halt. Dadurch ergibt sich analog eine zusätzliche Verkleinerung der Bereiche der noch nicht bearbeitenden Anfragen, dies führt dann anschließend wiederum zum umgekehrten Effekt.

Anhand des in Abbildung 8.3.4 gezeigten Verlaufs für die dreihundert Schrittweiten zum Ende der Sequenz aus Anfragen, lässt sich gut erkennen, aus welchen Gründen in Unterabschnitt 8.3.2 bei der Berechnung der bereinigten durchschnittlichen Schrittweite für einen Informationsvorlauf mit k Anfragen auf die Werte der letzten k -Schrittweiten verzichtet wurde.

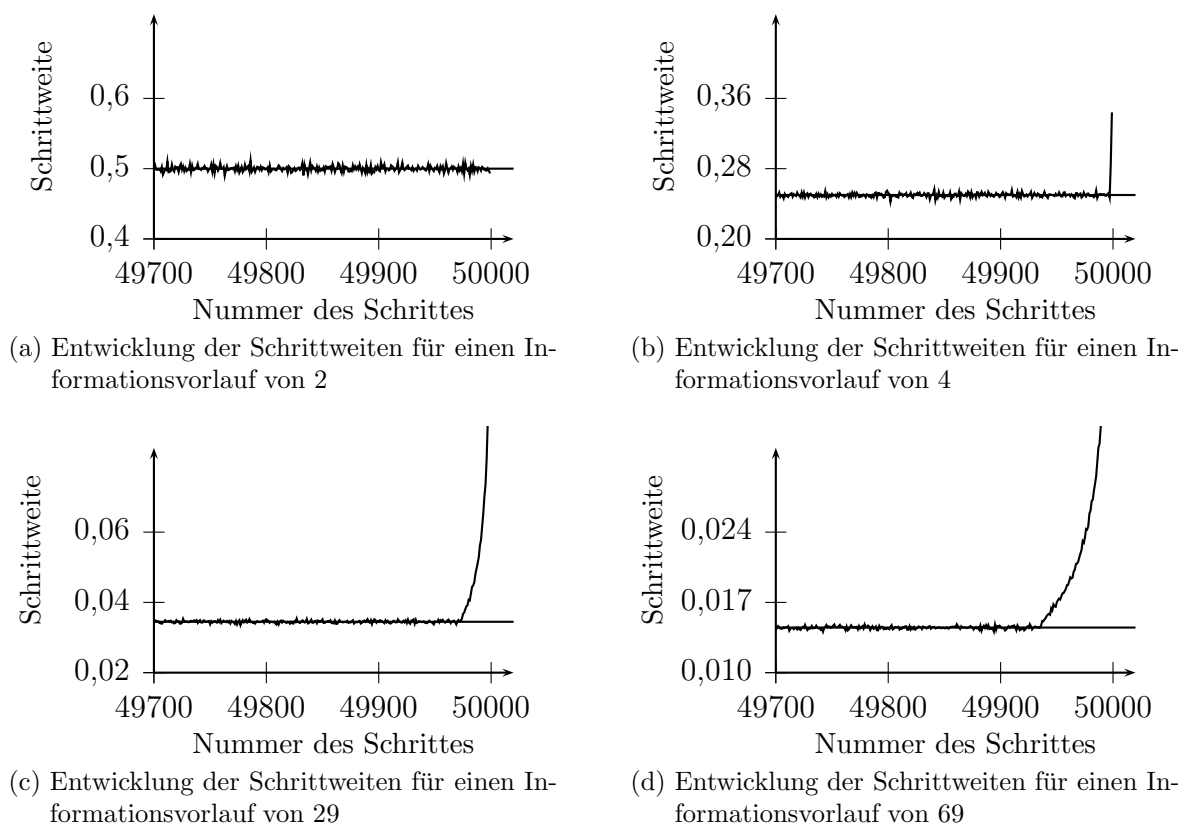


Abbildung 8.3.4.: Die Schrittweiten der letzten Schritte für einen Informationsvorlauf von 2, 4, 29 und 69 bei der gerichteten Rundreise auf der Kreislinie

Da es bei der Bearbeitung der letzten k Anfragen aus der Sequenz der Anfragen, keine weiteren unbekannten Anfragen mehr gibt, bleibt nur ein immer kleiner werdender Informationsvorlauf zurück. Dadurch schränken sich die Möglichkeiten zur Auswahl der als nächstes zu bearbeitenden Anfrage stark ein, und als Konsequenz daraus steigen die Schrittweiten der letzten Schritte.

Ansonsten bleiben die Schrittweiten der nicht in den beiden Abbildungen 8.3.3 und 8.3.4 enthaltenen Schritte relativ nahe an den eingezeichneten Niveau-Linien für $\frac{1}{k}$, dies entspricht dem in Unterabschnitt 8.3.2 und Unterabschnitt 8.3.1 durch die nichtlineare

Regressionsanalyse gefundenem Modell für den Einfluss des Informationsvorlaufs auf die durchschnittlichen Schrittweiten.

8.3.4. Schrittweiten der einzelnen Schritte unter Veränderung des Informationsvorlaufs

Die Abbildung 8.3.5 zeigt den Verlauf der Schrittweiten einzelner Schritte unter Variation des Informationsvorlaufs, eingezeichnet sind sowohl der Verlauf für den ersten Schritt (die obersten schwarzen Punkte), als auch der Verlauf für den zehnten Schritt (die Punkte in der zweiten Folge von oben), sowie der für den Schritt 70 (die grauen Punkte). Die darunter liegenden schwarzen Punkte zeigen die Entwicklung der Schrittweite des Schrittes 95. Als Vergleichsmaßstab ist zusätzlich, als Linie, der Verlauf der empirisch ermittelten bereinigten durchschnittlichen Schrittweiten eingezeichnet. Die Graphik in

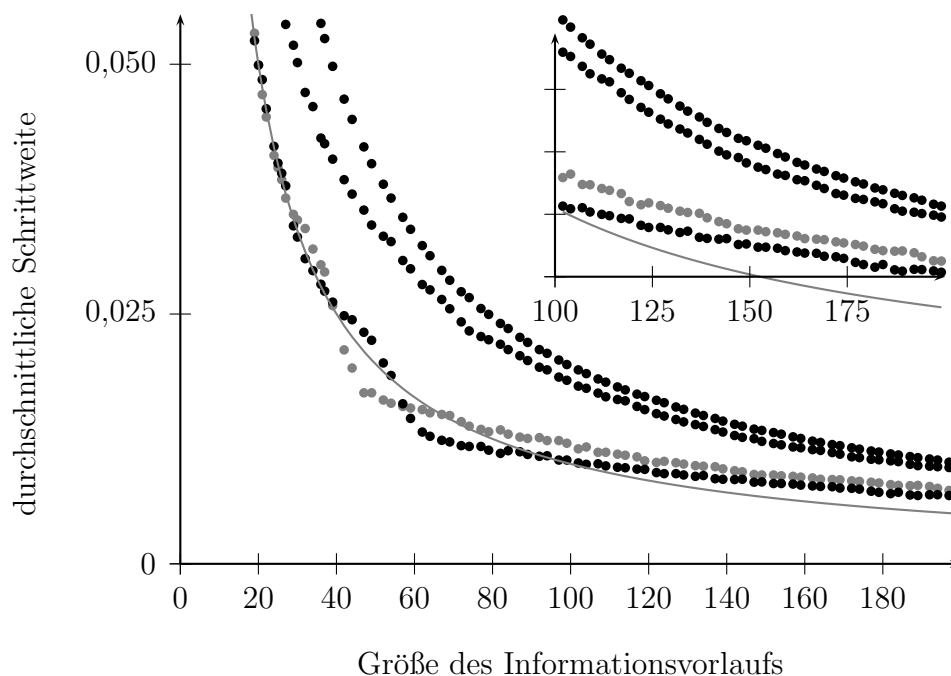


Abbildung 8.3.5.: Schrittweiten der Schritte 1, 10 und 70, sowie 95 unter Variation des Informationsvorlaufs für die gerichtete Rundreise auf der Kreislinie

der rechten oberen Ecke zeigt noch einmal den Verlauf für einen Informationsvorlauf von 100 bis 200 Anfragen aber diesmal mit einer besseren Skalierung auf der vertikalen Achse.

Aus der Abbildung 8.3.5 ist, wie eigentlich erwartet, klar ersichtlich, dass mit einer zunehmenden Größe des Informationsvorlaufs die Schrittweite des einzelnen Schrittes immer weiter abnimmt und auch die Schrittweiten für spätere Schritte kleiner sind als die der ersten Schritte.

Der erste Effekt ergibt sich einfach aus der Tatsache, dass bei einem vergrößerten Informationsvorlauf mehr Anfragen für die Auswahl des nächsten Zielpunkts zur Verfügung stehen und dadurch deren Reisepunkte tendenziell enger beisammen liegen. Der zweite Effekt mit den niedrigeren Schrittweiten für höhere Schritte ergibt sich daraus, dass die Reisepunkte der, nach der Auswahl des nächsten Ziels, verbleibenden Anfragen in einen kleineren Bereich liegen.

Da die Schrittweiten für die ersten Werte des Informationsvorlaufs im Vergleich zu den anderen relativ groß sind, ist im höheren Wertebereich des Informationsvorlauf der Unterschied zwischen den einzelnen Punkte-Kurven im Vergleich mit den ersten Teil nicht mehr wirklich zu erkennen. Deshalb wird dieser Abschnitt über der eigentlichen Graphik noch einmal vergrößert dargestellt.

In der graphischen Darstellung für die Schrittweiten im Schritt 70 zeigen sich eindeutig die Auswirkungen zwischen der Größe des Informationsvorlaufs und der Nummer des Schrittes:

- Im ersten Bereich des Informationsvorlaufs (so ungefähr bis 35 Anfragen) fällt die Schrittweite relativ regelmäßig mit zunehmenden Informationsvorlauf.
- Der danach folgende Bereich zeigt einen Verlauf, welcher wie ein Fragezeichen aufgebaut ist. Dabei entsteht erst eine stark fallende Rechtskrümmung, welche dann in eine leicht fallende Linkskrümmung wechselt und schließlich in einen stabileren Verlauf übergeleitet wird.
- Der Verlauf ab einem Informationsvorlauf um die 70 Anfragen hat dann wieder Ähnlichkeiten zu den für die beiden niedrigeren Schritte eingezeichneten Kurven.

Eine Erklärung für die einzelnen Bereiche eines betrachteten Schrittes t ergibt sich durch eine Betrachtung der Kurven aus Abbildung 8.3.3, wobei hierbei für den Schritt t die Graphiken mit den unterschiedlichen Werten des Informationsvorlaufs nach einander in der Reihenfolge der Größe des Informationsvorlaufs betrachtet werden. Dies entspricht einem Ablaufen der einzelnen Abschnitte der Schrittweite für einen Informationsvorlauf in entgegengesetzter Richtung unter der zusätzlichen Veränderung des Informationsvorlaufs.

- Im ersten Bereich steht bei der Planung des Schrittes t nur ein Informationsvorlauf von weniger als der Hälfte von t zur Verfügung. Das bedeutet, dass sich die Schrittweite bei dem ausgewählten Informationsvorlauf schon vor dem Schritt t auf einen stabileren Verlauf eingependelt hat. Dadurch ergibt sich der regelmäßig monoton fallende Verlauf in diesem Bereich, durch die schon stabileren Schrittweiten ergeben sich nämlich keine außergewöhnlichen Schwankungen.
- Der zweite Bereich umfasst die Größen des Informationsvorlaufs von ungefähr der Hälfte des Schrittes t bis zu t selbst. Innerhalb dieses Bereichs nähert sich die Größe des Informationsvorlaufs also immer mehr der Nummer des Schrittes an.

Nach dem Verlauf der Kurven aus Abbildung 8.3.3 führt dies dazu, dass dabei der tiefste Bereich der Schrittweite für den Informationsvorlauf überschritten wird. Auf dem Weg dahin ergeben sich folglich auch die größten Unterschiede in den Schrittweiten der einzelnen Werte des Informationsvorlaufs, was insbesondere den Wechsel der Krümmung in diesem Bereich erläutert.

- Beim Erreichen des letzten Bereichs, also bei einem größeren Informationsvorlauf als t , sind während der Planung des Schrittes t , definitiv noch Anfragen aus dem allerersten Informationsvorlauf zur Bearbeitung vorhanden. Dies führt dazu, dass die Schrittweite der Schritte vor t für diesen Informationsvorlauf entsprechend dem ersten Abschnitt aus Abbildung 8.3.3 monoton fällt, deshalb ist in diesen Abschnitt eine feste Struktur vorhanden. Zusätzlich wird diese Struktur auch noch dadurch unterstützt, dass mit zunehmenden Informationsvorlauf die Anzahl der Anfragen aus der Bearbeitungsinformation zur Planung des ersten Schrittes zunimmt. Insgesamt ergibt sich damit der gleichmäßigere Verlauf der Schrittweite des Schrittes t für den dritten und letzten Bereich.

Der Verlauf der Punkte für den Schritt 95 zeigt im ersten Abschnitt kaum einen Unterschied zu den Punkten für den Schritt 70. Der erste Wechsel des Verhaltens erfolgt dann allerdings bei einem Informationsvorlauf von circa 45 Anfragen, also wieder in der Nähe der Hälfte der Nummer des Schrittes. Insgesamt liegt der Verlauf der durch die Punkte definierten Kurve im ersten Bereich schon deutlich näher an der Kurve der empirisch bestimmten durchschnittlichen bereinigten Schrittweiten. Jedoch weichen diese wie auch schon die Punkte zum Schritt 70 in dem Bereich des Informationsvorlaufs von der Hälfte der Schrittnummer bis zur Schrittnummer am deutlichsten vom Verlauf dieser Kurve ab.

Mit größer werdenden Informationsvorlauf wird dann wieder ein ähnlicher Verlauf wie bei den empirischen Schrittweiten erreicht, dabei aber immer oberhalb des von diesen vorgegebenen Niveaus geblieben, wobei bei den späteren Schritten die Steigung gegenüber den empirischen Schrittweiten etwas geringer wird. Dies lässt sich damit erklären, dass bei den empirischen Schrittweiten dann immer ein größer werdender Informationsvorlauf zur Verfügung steht und deshalb dabei noch eine weitere Verkleinerung der Schrittweite erreicht werden kann.

8.4. Erste Schrittweiten

Vor der allgemeinen Analyse soll an Hand der Berechnung der ersten beiden Schrittweiten ein Eindruck davon vermittelt werden, wie sich das dazu notwendige grundsätzliche Vorgehen gestaltet und welche Eigenheiten beachtet werden müssen. Dabei werden folgende Konventionen angewandt:

- g_j ist die Anfrage, deren Reisepunkt im j -ten Schritt des verwendeten Algorithmus ausgewählt wurde. Das heißt, dass die nach dem Ende der Durchführung des Al-

gorithmus auf einer Sequenz von n Anfragen herauskommende Rundreise folgende Reisepunkte in der angegebenen Reihenfolge durchläuft:

$$g_0, g_1, g_2, \dots, g_n, g_{n+1}$$

Die Reisepunkte der zusätzlichen Anfragen g_0 und g_{n+1} entsprechen dabei dem Ausgangspunkt v_0 .

- τ_j ist die zurückgelegte Entfernung zwischen dem Reisepunkt der Anfrage g_j und dem Reisepunkt der davor ausgewählten Anfrage g_{j-1} , dabei erfolgt die Richtung der Bewegung entgegen dem Uhrzeigersinn. Das bedeutet, dass τ_j der Schrittweite im Schritt j entspricht.

8.4.1. Erste Schrittweite

Bei der Berechnung der relevanten Daten zu dem allerersten Schritt mit k gerade erst erschienenen Anfragen für die Problemstellung der gerichteten Rundreise auf der Kreislinie ergeben sich für diesen Schritt die Verteilung der Schrittweite und deren Erwartungswert wie folgt:

Satz 8.4.1 (Erwartungswert und Dichte beim ersten Schritt).

Der Erwartungswert $\mathbb{E}_k[\tau_1]$ und die Dichtefunktion $\varphi_k(\tau_1)$ für die zurückgelegte Distanz des ersten Schritts ergibt sich bei der Problemstellung der gerichteten Rundreise auf der Kreislinie aus Optimierungsproblem 8.1.1 abhängig von der Größe des Informationsvorlaufs k mit mindestens zwei Anfragen zu:

$$\mathbb{E}_k[\tau_1] = \frac{2}{k+1}\pi \qquad \varphi_k(\tau_1) = k \cdot \frac{(2\pi - \tau_1)^{k-1}}{(2\pi)^k}$$

Beweis zu Satz 8.4.1.

Ohne Beschränkung der Allgemeinheit wird davon ausgegangen, dass als erste bearbeitete Anfrage g_1 die Anfrage σ_1 ausgewählt wird.

Als Grundlage für die Integrale wird nicht die absolute Lage der Reisepunkte auf der Kreislinie verwendet, sondern die relative Lage gegenüber dem Reisepunkt der letzten bearbeiteten Anfrage. Das heißt zum Beispiel, dass sich bei einer Bearbeitungsinformation von zwei Anfragen und einer ersten Schrittweite von τ die folgende Situation ergibt:

- Der Reisepunkt der neu hinzugekommenen Anfrage kann auf der ganzen Kreislinie liegen.
- Der Reisepunkt der nach der Auswahl der ersten Anfrage verbliebenen Anfrage muss allerdings im Bereich $[\tau, 2\pi)$ liegen und ist somit maximal $2\pi - \tau$ vom Reisepunkt der ersten ausgewählten Anfrage, dem aktuellen Standort, entfernt.

Als Grundmenge für das anschließende Integral ergibt sich so der Bereich $[0, 2\pi - \tau)$ für die Entfernung des Reisepunktes der zweiten ausgewählten Anfrage zu dem Reisepunkt der ersten ausgewählten Anfrage.

Dadurch entsteht für die Berechnung des Erwartungswerts der ersten Schrittweite folgende Formel:

$$\begin{aligned}\mathbb{E}_k[\tau_1] &= k \cdot \int_0^{2\pi} \frac{\tau_1}{2\pi} \left(\int_{\tau_1}^{2\pi} \frac{1}{2\pi} d\sigma_i \right)^{k-1} d\tau_1 = k \cdot \int_0^{2\pi} \frac{\tau_1}{(2\pi)^k} (2\pi - \tau_1)^{k-1} d\tau_1 \\ &= \left[\frac{-\tau_1}{(2\pi)^k} (2\pi - \tau_1)^k - \frac{(2\pi - \tau_1)^{k+1}}{(k+1)(2\pi)^k} \right]_0^{2\pi} = \frac{(2\pi)^{k+1}}{(2\pi)^k(k+1)} = \frac{2}{k+1}\pi\end{aligned}$$

Die Dichtefunktion ergibt sich aus den Bestandteilen des Integrals über τ_1 ausgenommen der Multiplikation mit der Schrittweite selbst. $\square$

Bei einer Bearbeitung ohne einen Informationsvorlauf ergibt sich ein konstanter Erwartungswert von π für alle Schritte. Das heißt, ohne eine Verwendung von Informationsvorlauf hat der Algorithmus 8.1.2 nach der Anfrage σ_t im Erwartungswert schon eine Strecke der Länge $t\pi$ zurückgelegt.

Falls nach der Bearbeitung der ausgewählten Anfrage immer wieder k neue Anfragen erzeugt werden, ergibt sich mit Satz 8.4.1 nach dem Schritt t als bisherige Länge einer Rundreise ein Erwartungswert von $\frac{2t}{k+1}\pi$.

In der Abbildung 8.4.1 ist einerseits die erwartete Schrittweite des ersten Schritts als Kurve dargestellt, und andererseits durch die Punkte auch die empirisch berechneten durchschnittlichen Werte für den ersten Schritt. Dabei ist erkennbar, dass sich die berechneten und empirisch ermittelten Werte kaum unterscheiden. Dieser Eindruck wird ebenfalls durch eine nichtlineare Regressionsanalyse, die mit [Tea09, R] berechneten Ergebnisse dazu finden sich in Tabelle 8.4.1, unterstützt. Das verwendete Modell gestaltet

Parameter	Schätzwert	Standardfehler	t-Wert	P ($> t $)
β	0,989027	0,003109	318,1	$< 2 \cdot 10^{-16}$
α	2,001679	0,001468	1363,4	$< 2 \cdot 10^{-16}$

Tabelle 8.4.1.: Ergebnisse zur nichtlinearen Regression der ersten Schrittweite bei der gerichteten Rundreise auf der Kreislinie

sich dabei als $\frac{\alpha}{k+\beta}$, mit Startwerten $\alpha = 2$ und $\beta = 1$. Die quadratischen Abweichungen zwischen der gefundenen Ausprägung des Modells und den erzeugten empirischen Schrittweiten summieren sich schließlich auf $RSS = 1,87 \cdot 10^{-5}$. Dies ergibt dann ein Bestimmtheitsmaß von $R^2 = 0,999987$.

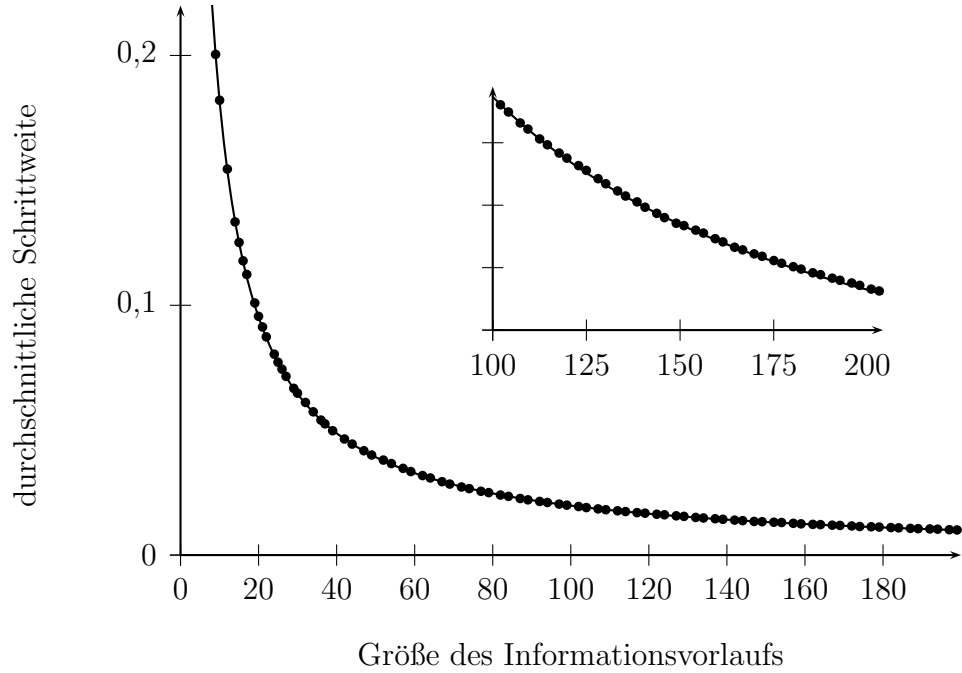


Abbildung 8.4.1.: Empirische und berechnete erwartete Schrittweite für den ersten Schritt bei der gerichteten Rundreise auf der Kreislinie

8.4.2. Zweite Schrittweite

Bei der Berechnung der Verteilung und des Erwartungswerts der Schrittweite zur zweiten ausgewählten Anfrage muss schon eine erste Fallunterscheidung eingebaut werden. Je nachdem, ob als zweite zu bearbeitende Anfrage eine der Anfragen aus $\sigma_2, \dots, \sigma_k$ oder die gerade eben bekannt gewordene Anfrage σ_{k+1} ausgewählt wird, ergeben sich verschiedene Situationen.

Satz 8.4.2 (Erwartungswert beim zweiten Schritt).

Der Erwartungswert für die zurückgelegte Distanz des zweiten Schritts ergibt sich bei Optimierungsproblem 8.1.1 der gerichteten Rundreise auf der Kreislinie mit einem Informationsvorlauf k von mehr als drei Anfragen zu:

$$\mathbb{E}_k[\tau_2] = \frac{2}{k+2}\pi$$

Definition 8.4.3 (Beitrag zu einem Integral).

Für alle weiteren Berechnungen ist $\Theta[x : y]$ der *Beitrag des Falles y zum Erwartungswert von x* . Er ergibt sich als bedingter Erwartungswert von x unter y gewichtet mit der Wahrscheinlichkeit von y , er sollte also folgendes erfüllen:

$$\Theta[x : y] = P(y) \mathbb{E}[x | y]$$

Beweis zu Satz 8.4.2.

Die Berechnung des Erwartungswerts gestaltet sich wie folgt

$$\begin{aligned} \mathbb{E}_k[\tau_2] &= \int_0^{2\pi} \varphi_k(\tau_1) \int_0^{2\pi-\tau_1} \frac{\tau_2}{2\pi} \left(\int_{\tau_2}^{2\pi-\tau_1} \frac{1}{2\pi-\tau_1} d\sigma_2 \right)^{k-1} d\tau_2 d\tau_1 \\ &+ (k-1) \int_0^{2\pi} \varphi_k(\tau_1) \int_0^{2\pi-\tau_1} \frac{\tau_2}{2\pi-\tau_1} \left(\int_{\tau_2}^{2\pi-\tau_1} \frac{1}{2\pi-\tau_1} d\sigma_2 \right)^{k-2} \int_{\tau_2}^{2\pi} \frac{1}{2\pi} d\sigma_{k+1} d\tau_2 d\tau_1. \end{aligned} \quad (8.4.1)$$

Der erste Teil des Erwartungswerts aus (8.4.1) entsteht wenn die neue Anfrage σ_{k+1} gewinnt.

$$\begin{aligned} \Theta_k[\tau_2 : g_2 = \sigma_{k+1}] &= \int_0^{2\pi} \frac{k}{(2\pi)^{k+1}} \int_0^{2\pi-\tau_1} \tau_2 (2\pi - \tau_1 - \tau_2)^{k-1} d\tau_2 d\tau_1 \\ &= \int_0^{2\pi} \frac{k}{(2\pi)^{k+1}} \left[\frac{-\tau_2 (2\pi - \tau_1 - \tau_2)^k}{k} - \frac{(2\pi - \tau_1 - \tau_2)^{k+1}}{k+1} \right]_0^{2\pi-\tau_1} d\tau_1 \\ &= \frac{1}{(2\pi)^{k+1}} \int_0^{2\pi} \frac{(2\pi - \tau_1)^{k+1}}{k+1} d\tau_1 = \frac{1}{(2\pi)^{k+1}} \left[\frac{-(2\pi - \tau_1)^{k+2}}{(k+1)(k+2)} \right]_0^{2\pi} \\ &= \frac{2}{(k+1)(k+2)} \pi \end{aligned} \quad (8.4.2)$$

Beim zweiten Teil des Erwartungswerts aus (8.4.1) gewinnt eine alte Anfrage zum Beispiel σ_2 und der Beitrag dieses Teils berechnet sich dann wie folgt:

$$\begin{aligned} \Theta_k[\tau_2 : g_2 = \sigma_2] &= \int_0^{2\pi} \frac{k}{(2\pi)^{k+1}} \int_0^{2\pi-\tau_1} \tau_2 (2\pi - \tau_2) (2\pi - \tau_1 - \tau_2)^{k-2} d\tau_2 d\tau_1 \\ &= \frac{k}{(2\pi)^{k+1}} \int_0^{2\pi} \left[\frac{-2\pi\tau_2 (2\pi - \tau_1 - \tau_2)^{k-1}}{k-1} - \frac{2\pi(2\pi - \tau_1 - \tau_2)^k}{k(k-1)} \right. \\ &\quad \left. + \frac{\tau_2^2 (2\pi - \tau_1 - \tau_2)^{k-1}}{k-1} + \frac{2\tau_2 (2\pi - \tau_1 - \tau_2)^k}{k(k-1)} + \frac{2(2\pi - \tau_1 - \tau_2)^{k+1}}{k(k-1)(k+1)} \right]_0^{2\pi-\tau_1} d\tau_1 \\ &= \frac{1}{(2\pi)^{k+1}} \int_0^{2\pi} \frac{2\pi(2\pi - \tau_1)^k}{(k-1)} - \frac{2(2\pi - \tau_1)^{k+1}}{(k-1)(k+1)} d\tau_1 \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{(2\pi)^{k+1}} \left(\frac{(2\pi)^{k+2}}{(k-1)(k+1)} - \frac{2(2\pi)^{k+2}}{(k-1)(k+1)(k+2)} \right) \\
 &= \frac{2\pi}{(k-1)(k+1)} \left(1 - \frac{2}{k+2} \right) = \frac{2k}{(k-1)(k+1)(k+2)} \pi \quad (8.4.3)
 \end{aligned}$$

Zusammengesetzt ergibt sich schließlich aus (8.4.3) und (8.4.2)

$$\begin{aligned}
 \mathbb{E}_k[\tau_2] &= (k-1) \cdot \Theta_k[\tau_2 : g_2 = \sigma_2] + \Theta_k[\tau_2 : g_2 = \sigma_{k+1}] \\
 &= \frac{2k}{(k+1)(k+2)} \pi + \frac{2}{(k+1)(k+2)} \pi = \frac{2}{k+2} \pi
 \end{aligned}$$

also der im Satz 8.4.2 angegebene Erwartungswert. $\square$

Das Ergebnis von Satz 8.4.2 ist in der Abbildung 8.4.2 zusammen mit den durchschnittliche empirischen Werten für die zweite Schrittweite, das sind wieder die einzelnen Punkte, eingezeichnet. Auch hier sind auf den ersten Blick keine wirklich gravieren-

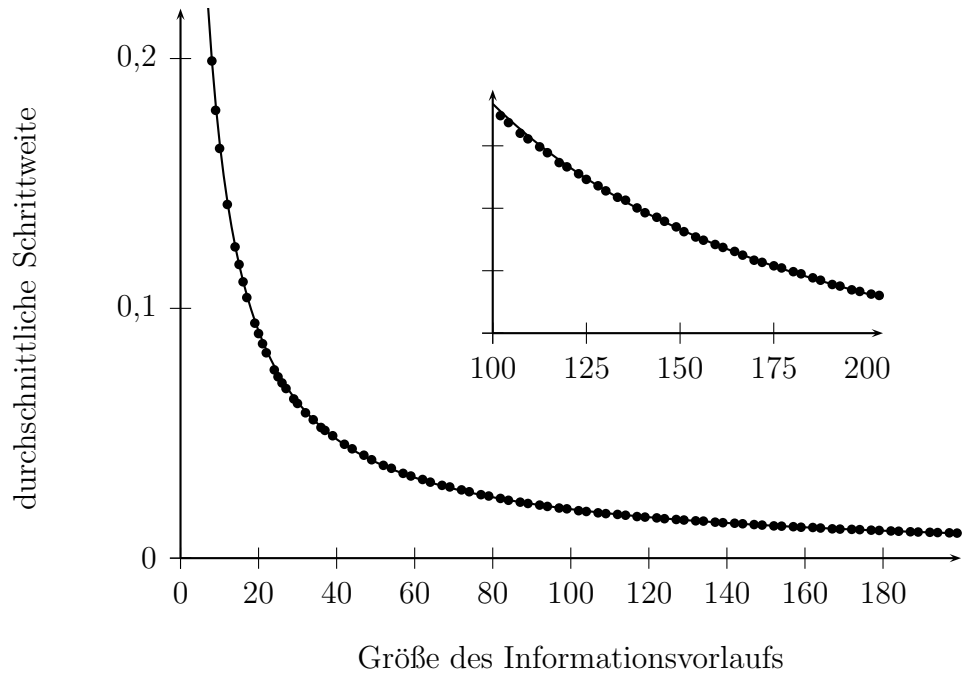


Abbildung 8.4.2.: Empirische und berechnete erwartete Schrittweite für den zweiten Schritt bei der gerichteten Rundreise auf der Kreislinie

den Unterschiede zwischen dem Verlauf der beiden Darstellungen erkennbar. Dies lässt sich durch eine nichtlineare Regressionsanalyse in [Tea09, R] mit den Ergebnissen aus Tabelle 8.4.2, aufbauend auf dem Modell $\frac{\alpha}{k+\beta}$ und den Startwerten $\alpha = 2$ sowie $\beta = 2$ verdeutlichen. Die Abweichungen zwischen den einzelnen Werten des Modells und den

Parameter	Schätzwert	Standardfehler	t-Wert	P ($> t $)
β	2,00274	0,00679	294,9	$< 2 \cdot 10^{-16}$
α	1,99116	0,00233	854,6	$< 2 \cdot 10^{-16}$

Tabelle 8.4.2.: Ergebnisse zur nichtlinearen Regression der zweiten Schrittweite bei der gerichteten Rundreise auf der Kreislinie

empirischen gefundenen ergeben als Summe der quadratischen Fehler einen Wert von $RSS = 3,18 \cdot 10^{-5}$, dabei wird durch das gefundene Modell ein Bestimmtheitsmaß von $R^2 = 0,9999678$ erreicht.

8.5. Allgemeine Schrittweite

Aufbauend auf den Erkenntnissen zu den Erwartungswerten der ersten beiden Schritte können nun einige allgemeinere Aussagen aufgestellt werden.

Seien dazu $u_1^t, \dots, u_{k-1}^t$ die noch nicht bearbeiteten alten Anfragen in Schritt t , sowie u_k^t die neu hinzu gekommene Anfrage, also:

$$\{u_1^t, \dots, u_{k-1}^t\} = \{\sigma_1, \dots, \sigma_{t+k-2}\} \setminus \{g_1, \dots, g_{t-1}\} \quad \text{und} \quad u_k^t = \sigma_{t+k-1}$$

Als zusätzliche Konvention seien die Anfragen $u_1^t, \dots, u_{k-1}^t$ nach ihrem Index in der gesamten Sequenz der Anfragen geordnet, so dass u_1^t diejenige Anfrage ist, welche von den unbearbeiteten Anfragen als erste bekannt wurde. Diese Sortierung der noch unbearbeiteten Anfragen ist deshalb so wichtig, weil diejenige mit dem kleinsten Index in der Sequenz der Anfragen unter diesen Anfragen den minimalen Bereich für mögliche Entfernungen ihres Reisepunktes zum aktuellen Standort besitzt und dadurch vorgibt, was die maximal mögliche Entfernung zum Reisepunkt der als nächstes ausgewählten Anfrage ist.

Zur Berechnung der Länge des für den Reisepunkt einer Anfrage σ_i noch möglichen Bereichs im Schritt t , unter Einbeziehung der zuvor gemachten Schritte, wird die Funktion gr wie folgt definiert:

$$gr(\sigma_i, t) = 2\pi - \sum_{j=\min\{i-k+1, 1\}}^{t-1} \tau_j$$

Ein Wert von $gr(\sigma_i, t) = 1$ bedeutet dann, dass der Reisepunkt der Anfrage σ_i im Schritt t maximal eine Distanz von eins in Laufrichtung von der aktuellen Position, also dem Reisepunkt der letzten bearbeiteten Anfrage g_{t-1} , entfernt liegen kann. Das impliziert gleichzeitig, dass der Algorithmus 8.1.2 seit der Bekanntgabe der Anfrage σ_i ausgehend vom damaligen Standort bei der Bearbeitung von anderen Anfragen schon eine Strecke

von $2\pi - 1$ zurückgelegt hat. Durch die oben vorgegebene Konvention mit der Sortierung der alten Anfragen nach deren Position in der Sequenz der Anfragen lässt sich

$$\min_{i=1}^k \{ \text{gr}(u_i^t, t) \} = \text{gr}(u_1^t, t)$$

direkt auflösen und damit ist die maximal mögliche Schrittweite im Schritt t eindeutig gegeben.

Mit diesen Vorgaben ergibt sich die Verteilung der Schrittweite im Schritt t bezogen auf die bis dahin schon benutzte Reihenfolge der Bearbeitung $g_1, \dots, g_{t-1}$ und der Auswahl der Anfrage g_t zu:

$$\frac{1}{\text{gr}(g_t)} \prod_{u_i^t \neq g_t} \int_{\tau_t}^{\text{gr}(u_i^t, t)} \frac{1}{\text{gr}(u_i^t, t)} du_i^t = \frac{1}{\text{gr}(g_t)} \prod_{u_i^t \neq g_t} \frac{\text{gr}(u_i^t, t) - \tau_t}{\text{gr}(u_i^t, t)} = \frac{\prod_{u_i^t \neq g_t} (\text{gr}(u_i^t, t) - \tau_t)}{2\pi \prod_{i=1}^{k-1} \text{gr}(u_i^t, t)} \quad (8.5.1)$$

Die Verteilung (8.5.1) lässt sich deshalb so allgemein darstellen, weil sich bedingt durch die Vereinfachung auf nur eine mögliche Laufrichtung, die Änderung am noch möglichen Bereich für den jeweiligen Reisepunkt der verbleibenden, unbearbeiteten Anfragen immer in der selben Form vorgenommen wird. Es reduziert sich nämlich immer nur die noch maximal mögliche Distanz vom Reisepunkt der letzten bearbeiteten Anfrage aus zum Reisepunkt der noch nicht bearbeiteten Anfrage um die Schrittweite, welche ausgehend vom Reisepunkt der davor bearbeiteten Anfrage bei der Bewegung zu dem neuen Standort zurückgelegt wurde.

Bei der Bearbeitung von Optimierungsproblem 7.1.1, also der Rundreise auf der Kreislinie mit Richtungswechsel, dagegen müssen mehrere verschiedene Aspekte, wie insbesondere die Laufrichtung und die Relation zwischen dem aktuellen Standort und dem verbleibenden Bereich des Reisepunktes der unbearbeiteten Anfrage auf der Kreislinie, bedacht werden. Eine erste ausführlichere Betrachtung darüber findet sich in Unterabschnitt 9.4.2.

Aus der Definition von gr ergibt sich für die alten Anfragen $u_1^t, \dots, u_{k-1}^t$, also diejenigen Anfragen die schon im Schritt $t - 1$ zur Verfügung standen, für Schritt t folgende Beziehung:

$$\text{gr}(u_j^t, t) = \text{gr}(u_j^t, t - 1) - \tau_{t-1} \quad \text{für } j < k \quad (8.5.2)$$

Bei der Kombination von (8.5.2) mit den Verteilungen aus (8.5.1) für die Schritte t und $t - 1$, wie in der Berechnung des Erwartungswerts für die Schrittweite τ_ℓ mit $\ell > t$, kann

wegen $\{u_1^t, \dots, u_{k-1}^t\} = \{u_1^{t-1}, \dots, u_{k-1}^{t-1}\} \setminus \{g_{t-1}\}$ wie folgt gekürzt werden:

$$\begin{aligned} \dots \int_0^{\text{gr}(u_1^{t-1}, t-1)} \frac{\prod_{u_i^{t-1} \neq g_{t-1}} (\text{gr}(u_i^{t-1}, t-1) - \tau_{t-1})}{2\pi \prod_{i=1}^{k-1} \text{gr}(u_i^{t-1}, t-1)} \int_0^{\text{gr}(u_1^t, t)} \frac{\prod_{u_i^t \neq g_t} (\text{gr}(u_i^t, t) - \tau_t)}{2\pi \prod_{i=1}^{k-1} \text{gr}(u_i^t, t)} \dots \\ = \dots \frac{1}{(2\pi)^2} \int_0^{\text{gr}(u_1^{t-1}, t-1)} \int_0^{\text{gr}(u_1^t, t)} \frac{\prod_{u_i^t \neq g_t} (\text{gr}(u_i^t, t) - \tau_t)}{\prod_{i=1}^{k-1} \text{gr}(u_i^{t-1}, t-1)} \dots \quad (8.5.3) \end{aligned}$$

Nach der sukzessiven Fortführung der Kürzung aus (8.5.3) über alle weiteren Integrale, entsteht so als Formel für den Beitrag einer möglichen Reihenfolge der Bearbeitung $g_1, \dots, g_t$ zum Erwartungswert über die Schrittweite im Schritt t das folgende Integral:

$$\Theta_k [\tau_t : g_1, \dots, g_t] = \frac{1}{(2\pi)^{k+t-1}} \int_0^{2\pi} \int_0^{2\pi-\tau_1} \dots \int_0^{\text{gr}(u_1^t, t)} \tau_t \prod_{u_i^t \neq g_t} (\text{gr}(u_i^t, t) - \tau_t) d\tau_t \dots d\tau_1 \quad (8.5.4)$$

Der gesamte Erwartungswert $\mathbb{E}_k [\tau_t]$ für die Schrittweite im Schritt t ergibt sich dann als Aufsummierung der Beiträge aller möglichen Bearbeitungsreihenfolgen bei einem Informationsvorlauf von k Anfragen.

$$\mathbb{E}_k [\tau_t] = \sum \Theta_k [\tau_t : g_1, \dots, g_t] \quad (8.5.5)$$

Mithilfe dieser Formel kann nun im Gegensatz zu den Berechnungen in Abschnitt 8.4 die erwartete Schrittweite ohne weitere Vorgaben an die Größe des Informationsvorlaufs durchgeführt werden.

8.5.1. Berechnung der Schrittweiten für einen Informationsvorlauf von zwei Anfragen

Um die in (8.5.5) festgelegte Formel zur Berechnung der erwarteten Schrittweiten etwas mehr zu veranschaulichen, wird nun in diesem Abschnitt erst einmal eine Analyse der Situation bei einem Informationsvorlauf von zwei Anfragen durchgeführt. Der Beweis zu Satz 8.5.1 zeigt zudem, dass in diesem Spezialfall die Formel aus (8.5.5) zur einer wesentlich einfacheren Formel für die Berechnung zusammengefasst werden kann.

Satz 8.5.1 (Erwartete Schrittweite bei einem Informationsvorlauf von zwei Anfragen). *Die erwartete Schrittweite bei einem Informationsvorlauf von zwei Anfragen, also $k = 2$, ist im ersten Schritt $\frac{2}{3}\pi$ und in allen weiteren Schritten danach konstant bei $\frac{1}{2}\pi$.*

Beweis zu Satz 8.5.1.

Das Ergebnis für die erste Schrittweite kann direkt aus Satz 8.4.1 entnommen werden. Allerdings ist der Nachweis, dass ab dem dritten Schritt alle anderen erwarteten Schrittweiten konstant bei $\frac{1}{2}\pi$ verharren nicht ganz so einfach.

Bei der Berechnung der erwarteten Schrittweite zum Schritt t nach (8.5.5), über alle möglichen Bearbeitungsreihenfolgen, können diese in die Struktur eines Binär-Baums gebracht werden. Die Abbildung 8.5.1 ist dazu wie folgt aufgebaut:

- Jede Kante in diesem Baum steht für eine vom Algorithmus 8.1.2 getroffene Entscheidung welche der zwei zur Verfügung stehenden Anfragen ausgewählt wurde.

Die Beschriftung mit „alt“ kennzeichnet, dass diejenige der beiden möglichen Anfragen ausgewählt wurde, deren Index der kleinere ist.

Eine Kante mit „neu“ repräsentiert analog die Auswahl der Anfrage mit dem größten Index, also der neu bekannt gewordenen Anfrage.

- Jeder Knoten ist mit der zugehörigen Nummer der im vorherigen Schritt unbearbeitet gebliebenen Anfrage beschriftet.

Dabei wurde hier die eigentliche Wurzel des Baumes mit der Situation vor Beginn der Bearbeitung der Problemstellung durch den Algorithmus 8.1.2 nicht eingebaut. Das heißt, der Knoten ganz links bezieht sich auf die verbliebene Anfrage nach der Auswahl der Anfrage g_1 im ersten Schritt. Ohne Beschränkung der Allgemeinheit wurde hierbei als g_1 die Anfrage σ_1 ausgewählt.

- Nach der Auswahl der ersten Anfrage stehen dann im zweiten Schritt für g_2 die beiden Anfragen σ_2 und σ_3 zur Auswahl, entsprechend ergeben sich von dem ersten Knoten mit der unbearbeiteten Anfrage σ_2 zwei Kanten mit den Beschriftungen „alt“ und „neu“.

Die sich daraus ergebenden Situationen befinden sich in dem eingerahmten Teil des Baums, welcher mit τ_2 gekennzeichnet ist. Dieser Rahmen kennzeichnet auch alle in dem gezeigten Abschnitt enthaltenen Knoten der ausgehend von der eigentlichen Wurzel zweiten Ebene von oben.

- Der Abschnitt auf der rechten Seite mit den gestrichelten Kegeln steht schließlich für die restlichen nicht mehr darstellbaren Äste des Baums zu den Schrittweiten zwischen τ_4 und τ_t .

Die auf der untersten Ebene des Baumes enthaltenen Knoten, also die Blätter, stehen jeweils für die nach dem letzten betrachteten Schritt zur weiteren Bearbeitung verbleibende Anfrage.

- Die eigentliche Nummerierung der Ebenen beginnt wegen der später definierten Rekursionsformel zur Auswertung hingegen von unten, das heißt, dass die Blätter

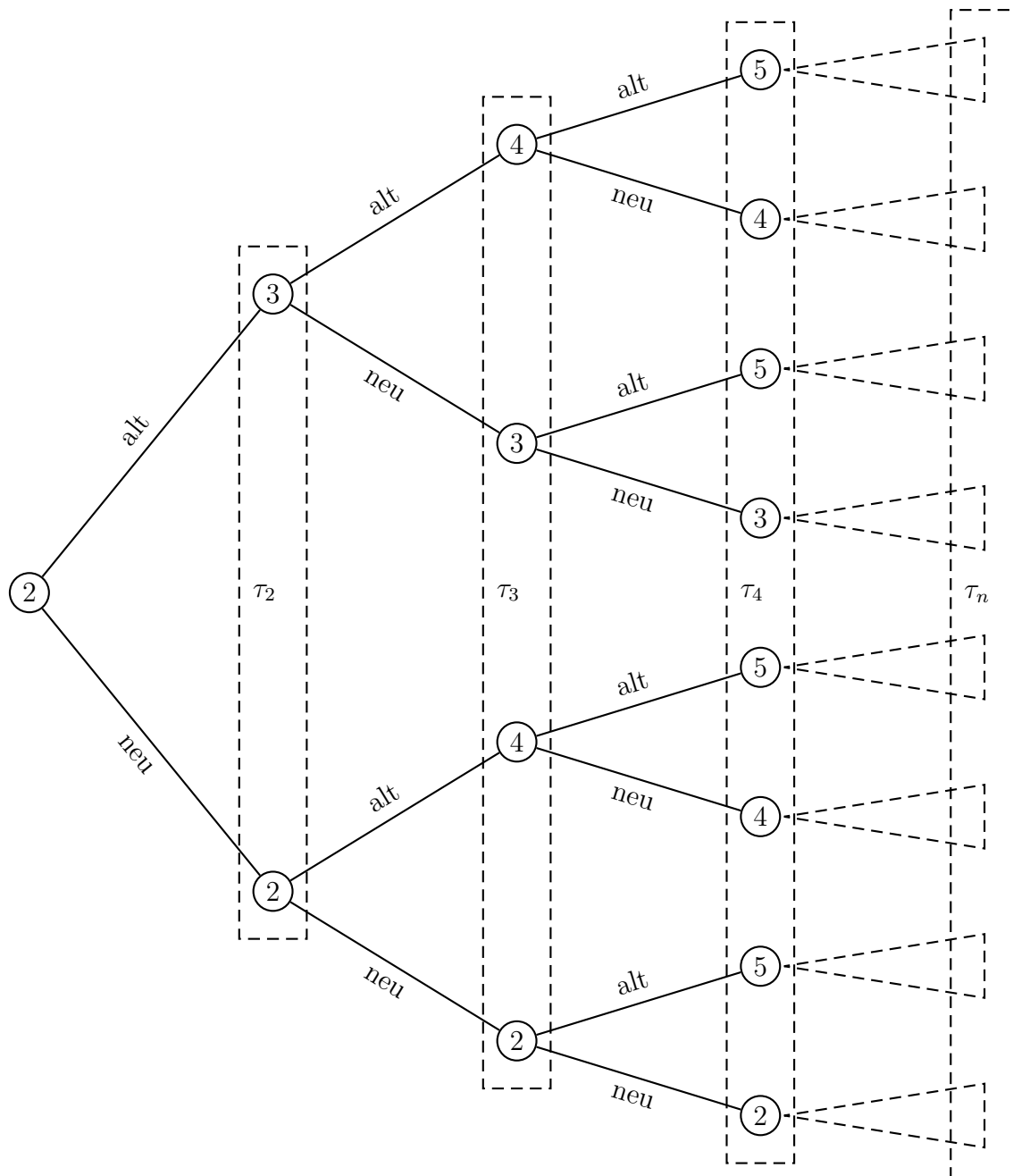


Abbildung 8.5.1.: Darstellung der Bearbeitungsreihenfolgen bis zum Schritt n für einen Informationsvorlauf von zwei Anfragen

in der Ebene Null gesammelt werden und der eigentliche Wurzelknoten die Ebene t darstellt.

Allgemein ist die Ebene j als die Menge aller Knoten definiert, welche jeweils eine Situation nach der Durchführung von $t - j$ Schritten darstellen, also $t - j$ Kanten von der eigentlichen Wurzel entfernt sind.

Zusammengefasst steht damit innerhalb der Knoten jeweils die Nummer der im vorherigen Schritt unbearbeiteten Anfrage, und die Beschriftung der Kanten gibt immer an, ob in dem Schritt auf dem Weg zum nächsten Knoten die schon länger bekannte oder die neue Anfrage ausgewählt wurde.

Für die Betrachtung der ersten t Schritte muss also ein Baum mit $t + 1$ Ebenen erzeugt werden. Soll diese Betrachtung dann auf den Schritt $t + 1$ erweitert werden, so genügt es an die Blätter des Baumes für den Schritt t jeweils die sich aus der Durchführung des nächsten Schrittes angewandt auf die durch das Blatt gegebene Situation ergebenden neuen Situationen mit den zugehörigen Kanten anzufügen.

Die Berechnung der Beiträge nach (8.5.4) für den Erwartungswert nach (8.5.5) kann mithilfe des Binärbaums aus Abbildung 8.5.1 auf eine Rekursionsformel vereinfacht werden. Um dieses Ziel zu erreichen, werden zu jeder der in Abbildung 8.5.1 enthaltenen Ebenen folgende Größen definiert:

- x_i steht für die im zugehörigen Schritt möglichen Schrittweiten.
- y_i steht immer für die Länge der Region der Kreislinie, in welcher bei effizienter Ausführung des aktuellen Schrittes die ausgewählte Anfrage nicht mehr liegen wird, das heißt, dass dann noch maximal eine Schrittweite von $2\pi - y_i$ möglich sein wird. Da hier ja ein Informationsvorlauf von zwei Anfragen vorgegeben ist, ist y_i durch die schon bekannte, jedoch noch nicht bearbeitete, Anfrage festgelegt und zwar als die Gesamtlänge der Wegstrecke des Algorithmus 8.1.2 seit der Bekanntgabe dieser Anfrage.

Da die nachfolgend definierte rekursive Formel auf der Ebene der Blätter beginnt, steht bei einer Auswertung über t Schritte x_1 für die Schrittweite im letzten Schritt und x_t für die im ersten Schritt. Für die Auswertung selbst wird eine Funktion $f_t(y_t)$ verwendet, welche folgende Eigenschaften erfüllen soll:

- Der Wert von $f_i(y_i)$ liefert die zu erwartende Schrittweite im Schritt i unter der Einschränkung, dass die ältere Anfrage nach dem vorherigen Schritt noch maximal $2\pi - y_i$ vom aktuellen Standort entfernt ist. Damit entspricht $f_1(y_1)$ der zu erwartenden Schrittweite bei Vorgabe der Länge y_1 des Bereichs, in welchem die ältere Anfrage nicht mehr liegt.
- Gleichzeitig ist $f_i(y_i)$ aber auch die Auswertung für einen Knoten der Ebene i , mit einer Beschränkung von y_i für die vom vorherigen Schritt nicht bearbeitete Anfrage, innerhalb der Berechnung der erwarteten Schrittweite für den Schritt

t . Dabei entspricht $f_i(y_i)$ nicht der rein zahlenmäßigen erwarteten Schrittweite, sondern liefert die funktionale Abhängigkeit dieser erwarteten Schrittweite von der Größe der Einschränkung durch y_i .

Zur Veranschaulichung ist die Beziehung zwischen den Auswertungen für zwei aufeinander folgende Ebenen in Abbildung 8.5.2 für einen Knoten exemplarisch dargestellt. Zusammengefasst berechnet sich $f_i(y_i)$ rekursiv aus den beiden zugehörigen Ergebnissen

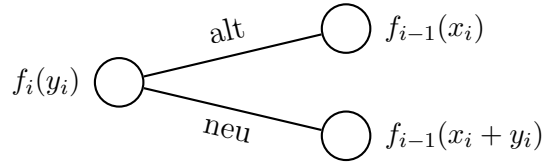


Abbildung 8.5.2.: Schematische Darstellung zur Struktur bei der rekursiven Berechnung der Schrittweite mit einem Informationsvorlauf von zwei Anfragen

der darunter liegenden Ebene des Baumes:

- $f_{i-1}(x_i)$ entspricht dabei der Auswertung der niedrigeren Ebene nach der Auswahl der älteren Anfrage in dem zur Ebene i gehörenden Schritt.
- $f_{i-1}(x_i + y_i)$ entspricht analog der Auswertung nach der Auswahl der neuen Anfrage.

Die Formel für $f_i(y_i)$ ergibt sich insgesamt als:

$$f_i(y_i) = \left(\frac{1}{2\pi} \right) \int_0^{2\pi-y_i} f_{i-1}(x_i) + f_{i-1}(x_i + y_i) dx_i \quad (8.5.6)$$

Der Startwert der rekursiven Auswertung der Formel (8.5.6) ist dabei durch

$$f_1(y_1) = \left(\frac{1}{2\pi} \right)^2 \int_0^{2\pi-y_1} x_1(2\pi - x_1) + x_1(2\pi - y_1 - x_1) dx_1 \quad (8.5.7)$$

gegeben. Dabei kann natürlich auch eine Formel für die unterste Ebene definiert werden

$$f_0(y_0) = x_1(2\pi - y_0) \quad (8.5.8a)$$

und über

$$f_1(y_1) = \left(\frac{1}{2\pi}\right)^2 \int_0^{2\pi-y_1} f_0(x_1) + f_0(x_1 + y_1) dx_1 \quad (8.5.8b)$$

wieder zu $f_1(y_1)$ zusammengesetzt werden.

Sei nun schon eine Auswahl der ersten j Anfragen $g_1, \dots, g_j$, mit $j \in \{0, \dots, t-1\}$, gegeben und E die Menge der Vektoren $v = (v_{j+1} \dots v_{t-1})$ mit der Eigenschaft, dass $g_1, \dots, g_j, v_{j+1}, \dots, v_{t-1}$ eine zulässige Reihenfolge der Bearbeitung ist. Außerdem sei auch noch v_t die ältere der im Schritt t noch nicht bearbeiteten Anfragen, sowie ν_i analog zu der Definition der Schrittweite τ_ℓ zwischen $g_{\ell-1}$ und g_ℓ die Schrittweite zum Reisepunkt v_i gemessen von dessen Vorgänger in der Reihenfolge der Bearbeitung.

Dann ergibt sich, bei der Berechnung der Schrittweite τ_t für den Schritt t , welche ja einen Baum der Tiefe $t+1$ mit sich zieht, für die Funktion $f_{t-j}(y_{t-j})$ mit $0 \leq j \leq t-1$ folgende Auswertung:

$$f_{t-j}(y_{t-j}) = \left(\frac{1}{2\pi}\right)^{t-j+1} \cdot \sum_{v \in E} \int_0^{2\pi-y_{t-j}} \dots \int_0^{\text{gr}(v_t, t)} \nu_t [(2\pi - \text{gr}(v_t, t) - \nu_t) + (2\pi - \nu_t)] d\nu_t \dots d\nu_{j+2} d\tau_{j+1}$$

Durch die Rekursionsformel aus (8.5.6) werden also die, durch die Unterscheidung zwischen der Auswahl der neuen oder der alten Anfrage als nächstes Ziel, entstandenen Teilbäume zu einem Baum der Tiefe i zusammengefügt. Dadurch ergibt sich der Erwartungswert für die Schrittweite τ_t dann schließlich als Auswertung der Funktion $f_t(y_t)$ mit $y_t = 0$, das entspricht dem Start mit einer Anfrage, deren Reisepunkt auf der ganzen Kreislinie liegen kann, zu:

$$\mathbb{E}_2[\tau_t] = f_t(0)$$

Das weitere Vorgehen gliedert sich nun wie folgt auf:

- (i) Berechnung der Ausdrücke $f_1(y_1)$, $f_2(y_2)$ und $f_3(y_3)$.
- (ii) Nachweis der allgemeinen Formel $f_t(y_t) = \frac{1}{4}(2\pi - y_t)$ für alle Knoten mit $t \geq 3$.
- (iii) Berechnung der Erwartungswerte $\mathbb{E}_2[\tau_t]$ mit $f_t(0)$ für die Schrittweiten τ_t .

Zu (i) Für die Berechnung der innersten Integrale aus der Formel (8.5.4) ergeben sich

nach (8.5.7) folgende Ergebnisse:

$$\begin{aligned} \left(\frac{1}{2\pi}\right)^2 \int_0^{2\pi-y_1} f_0(x_1) dx_1 &= \left(\frac{1}{2\pi}\right)^2 \int_0^{2\pi-y_1} x_1(2\pi - x_1) dx_1 \\ &= \left(\frac{1}{2\pi}\right)^2 \left(\pi(2\pi - y_1)^2 - \frac{1}{3}(2\pi - y_1)^3 \right) = \left(\frac{1}{2\pi}\right)^2 \frac{1}{3}(2\pi - y_1)^2 (\pi + y_1) \quad (8.5.9) \end{aligned}$$

wenn die alte Anfrage ausgewählt wurde und die neue Anfrage erst einmal zurückgestellt wird. Bei der Auswahl der neuen Anfrage und dem Verbleib der alten Anfrage für die weitere Bearbeitung ergibt sich:

$$\begin{aligned} \left(\frac{1}{2\pi}\right)^2 \int_0^{2\pi-y_1} f_0(x_1 + y_1) dx_1 &= \left(\frac{1}{2\pi}\right)^2 \int_0^{2\pi-y_1} x_1(2\pi - y_1 - x_1) dx_1 \\ &= \left(\frac{1}{2\pi}\right)^2 \left((2\pi - y_1) \frac{1}{2}(2\pi - y_1)^2 - \frac{1}{3}(2\pi - y_1)^3 \right) \\ &= \left(\frac{1}{2\pi}\right)^2 \frac{1}{6}(2\pi - y_1)^3 \quad (8.5.10) \end{aligned}$$

Beide Fälle ergeben zusammen unter Verwendung von (8.5.9) und (8.5.10) für den Anfang der Rekursionsformel:

$$\begin{aligned} f_1(y_1) &= \left(\frac{1}{2\pi}\right)^2 \left(\pi(2\pi - y_1)^2 - \frac{1}{3}(2\pi - y_1)^3 + \frac{1}{6}(2\pi - y_1)^3 \right) \\ &= \left(\frac{1}{2\pi}\right)^2 \left(\pi(2\pi - y_1)^2 - \frac{1}{6}(2\pi - y_1)^3 \right) = \frac{1}{6} \left(\frac{1}{2\pi}\right)^2 (2\pi - y_1)^2 (6\pi - 2\pi + y_1) \\ &= \frac{1}{6} \left(\frac{1}{2\pi}\right)^2 (2\pi - y_1)^2 (4\pi + y_1) \quad (8.5.11) \end{aligned}$$

Um nun die Formel für $f_2(y_2)$ berechnen zu können, wird in (8.5.11) y_1 ausgetauscht, dabei wird zur weiteren Berechnung nicht das endgültige Ergebnis verwendet, sondern die erste Zusammenfassung der beiden Fälle, denn damit ist die Integration einfacher.

Falls die ältere Anfrage gewählt wurde, wird die dabei betrachtete Schrittweite x_2 statt y_1 verwendet, und falls die neu bekannt gewordene Anfrage gewählt wurde, die Summe der betrachteten Schrittweiten seit der Bekanntgabe der älteren Anfrage, das heißt y_1 wird durch $y_2 + x_2$ ersetzt. Damit ergeben sich die Grundlagen zur

Berechnung der Formel für $f_2(y_2)$ bei Auswahl der alten Anfrage zu

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi-y_2} f_1(x_2) dx_2 &= \left(\frac{1}{2\pi}\right)^3 \int_0^{2\pi-y_2} \pi(2\pi-x_2)^2 - \frac{1}{6}(2\pi-x_2)^3 dx_2 \\ &= \left(\frac{1}{2\pi}\right)^3 \left[-\frac{\pi}{3}(y_2^3 - (2\pi)^3) + \frac{1}{24}(y_2^4 - (2\pi)^4) \right]. \end{aligned} \quad (8.5.12)$$

Bei der Auswahl der neuen Anfrage hingegen entsteht

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi-y_2} f_1(x_2 + y_2) dx_2 &= \left(\frac{1}{2\pi}\right)^3 \int_0^{2\pi-y_2} \pi(2\pi-y_2-x)^2 - \frac{1}{6}(2\pi-y_2-x)^3 dy \\ &= \left(\frac{1}{2\pi}\right)^3 \left[\frac{\pi}{3}(2\pi-y_2)^3 - \frac{1}{24}(2\pi-y_2)^4 \right]. \end{aligned} \quad (8.5.13)$$

Insgesamt ergibt sich für $f_2(y_2)$ aus (8.5.12) und (8.5.13):

$$\begin{aligned} f_2(y_2) &= \left(\frac{1}{2\pi}\right)^3 \left(\frac{\pi[-y_2^3 + (2\pi)^3 + (2\pi-y_2)^3]}{3} - \frac{[-y_2^4 + (2\pi)^4 + (2\pi-y_2)^4]}{24} \right) \\ &= \left(\frac{1}{2\pi}\right)^3 \left(\frac{\pi[(2\pi-y_2)^3 - y_2^3]}{3} - \frac{[(2\pi-y_2)^4 - y_2^4]}{24} + 2\pi^4 \right) \\ &= \left(\frac{1}{2\pi}\right)^3 \left(\frac{\pi[8\pi^3 - 12\pi^2 y_2 + 6\pi y_2^2 - 2y_2^3]}{3} \right. \\ &\quad \left. - \frac{[16\pi^4 - 32\pi^3 y_2 + 24\pi^2 y_2^2 - 8\pi y_2^3]}{24} + 2\pi^4 \right) \\ &= \left(\frac{1}{2\pi}\right)^3 \left(4\pi^4 - \frac{8}{3}\pi^3 y_2 + \pi^2 y_2^2 - \frac{1}{3}\pi y_2^3 \right) \\ &= \left(\frac{1}{2\pi}\right)^3 \pi(2\pi-y_2) \left(2\pi^2 - \frac{1}{3}\pi y_2 + \frac{1}{3}y_2^2 \right) \end{aligned} \quad (8.5.14)$$

Die dritte Ebene, also die Auswertung von $f_3(y_3)$, liefert schließlich jeweils angewandt auf den zweiten Term aus (8.5.14) für die Auswahl der alten Anfrage das

Ergebnis

$$\begin{aligned}
& \frac{1}{2\pi} \int_0^{2\pi-y_3} f_2(x_3) dx_3 \\
&= \left(\frac{1}{2\pi}\right)^4 \int_0^{2\pi-y_3} \frac{\pi}{3} [(2\pi-x_3)^3 - x_3^3] - \frac{1}{24} [(2\pi-x_3)^4 - x_3^4] + 2\pi^4 dx_3 \\
&= \left(\frac{1}{2\pi}\right)^4 \left(\frac{\pi}{12} [-y_3^4 + (2\pi)^4 - (2\pi-y_3)^4] \right. \\
&\quad \left. - \frac{1}{120} [-y_3^5 + (2\pi)^5 - (2\pi-y_3)^5] + 2\pi^4(2\pi-y_3) \right), \quad (8.5.15)
\end{aligned}$$

und für die Auswahl der neuen Anfrage

$$\begin{aligned}
& \frac{1}{2\pi} \int_0^{2\pi-y_3} f_2(x_3 + y_3) dx_3 \\
&= \left(\frac{1}{2\pi}\right)^4 \int_0^{2\pi-y_3} \frac{\pi}{3} [(2\pi-y_3-x_3)^3 - (x_3+y_3)^3] \\
&\quad - \frac{1}{24} [(2\pi-y_3-x_3)^4 - (x_3+y_3)^4] + 2\pi^4 dx_3 \\
&= \left(\frac{1}{2\pi}\right)^4 \left(\frac{\pi}{12} [(2\pi-y_3)^4 - (2\pi)^4 + y_3^4] \right. \\
&\quad \left. - \frac{1}{120} [(2\pi-y_3)^5 - (2\pi)^5 + y_3^5] + 2\pi^4(2\pi-y_3) \right). \quad (8.5.16)
\end{aligned}$$

Zusammen ergeben (8.5.15) und (8.5.16) dann die einfache Formel:

$$f_3(y_3) = \frac{1}{4}(2\pi - y_3) \quad (8.5.17)$$

Zu (ii) Nun wird $f_t(y_t) = \frac{1}{4}(2\pi - y_t)$ für $t \geq 3$ per Induktion bewiesen:

Da der Induktionsanfang für $t = 3$ schon in (8.5.17) nachgewiesen wurde, kann direkt mit dem Induktionsschritt von t nach $t + 1$ begonnen werden. Nach der Induktionsbehauptung gilt $f_t(y_t) = \frac{1}{4}(2\pi - y_t)$. Dieses Ergebnis wird angewandt, um daraus die Formel für $t + 1$ zu berechnen. Bei der Auswahl der alten Anfrage

in diesem Schritt ergibt sich

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi-y_{t+1}} f_t(x_{t+1}) dx_{t+1} &= \frac{1}{2\pi} \int_0^{2\pi-y_{t+1}} \frac{1}{4}(2\pi - x_{t+1}) dx_{t+1} \\ &= \frac{1}{16\pi}(-y_{t+1}^2 + (2\pi)^2), \quad (8.5.18) \end{aligned}$$

bei der Auswahl der neuen Anfrage ergibt sich analog

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi-y_{t+1}} f_t(x_{t+1} + y_{t+1}) dx_{t+1} &= \frac{1}{2\pi} \int_0^{2\pi-y_{t+1}} \frac{1}{4}(2\pi - y_{t+1} - x_{t+1}) dx_{t+1} \\ &= \frac{1}{16\pi}(2\pi - y_{t+1})^2. \quad (8.5.19) \end{aligned}$$

Die Zusammenfassung der Formeln (8.5.18) und (8.5.19) liefert das gewünschte Ergebnis:

$$\begin{aligned} f_{t+1}(y_{t+1}) &= \frac{1}{16\pi}(-y_{t+1}^2 + (2\pi)^2 + (2\pi - y_{t+1})^2) \\ &= \frac{1}{16\pi}(-y_{t+1}^2 + 4\pi^2 + 4\pi^2 - 4y_{t+1}\pi + y_{t+1}^2) \\ &= \frac{1}{16\pi}(4\pi(2\pi - y_{t+1})) = \frac{1}{4}(2\pi - y_{t+1}) \end{aligned}$$

Zu (iii) Der Erwartungswert der Schrittweite im ersten Schritt ergibt sich nach Satz 8.4.1 zu $\mathbb{E}_2[\tau_1] = \frac{2}{3}\pi$, der des zweiten Schritts ergibt sich aus (8.5.14) mit $y_2 = 0$ zu

$$\begin{aligned} \mathbb{E}_2[\tau_2] = f_2(0) &= \left(\frac{1}{2\pi}\right)^3 \pi(2\pi - 0) \left(2\pi^2 - \frac{1}{3}\pi \cdot 0 + \frac{1}{3} \cdot 0\right) \\ &= \left(\frac{1}{2\pi}\right)^3 \pi(2\pi)(2\pi^2) = \frac{4}{8}\pi = \frac{1}{2}\pi \end{aligned}$$

Für die Erwartungswerte der Schrittweite bei allen anderen Schritten $t \geq 3$ muss nur die per Induktion bewiesene Formel bei $y_t = 0$ ausgewertet werden, dies ergibt dann

$$\mathbb{E}_2[\tau_t] = f_t(0) = \frac{1}{4}(2\pi - 0) = \frac{1}{2}\pi$$

und somit ist das angegebene Ergebnis erreicht. $\square$

8.5.2. Formel zur allgemeinen Berechnung der Schrittweiten

Mit der Übertragung des Konzepts aus Unterabschnitt 8.5.1 auf den allgemeinen Fall, ergibt sich bei einem Informationsvorlauf von k Anfragen ein Baum wie in Abbildung 8.5.1, nur dass diesmal pro Knoten immer k Kanten in die darunter liegende Ebene gehen.

Für die weitere Betrachtung seien für einen Knoten aus der j -ten Ebene des Baums durch $u_1^j, \dots, u_{k-1}^j$ die $k-1$ alten Anfragen aus dem Informationsvorlauf, welche nach ihrem Index aufsteigend sortiert sind, gegeben. Dazu bezeichne a_i^j die Länge des noch möglichen Bereichs für den Reisepunkt der Anfrage mit dem i -ten kleinsten Index aus dem Informationsvorlauf in der Ebene j . Dann gilt wegen der Sortierung der Anfragen nach ihrer Reihenfolge in der Sequenz der Anfragen:

$$0 \leq a_1^j \leq a_2^j \leq \dots \leq a_{k-1}^j \leq 2\pi = a_k^j$$

Die Hilfsfunktion $f_j(a_1^j, \dots, a_{k-1}^j)$ definiert sich bei der Berechnung für den allgemeinen Fall analog zu (8.5.8) über den Anfang der Rekursion

$$f_0(a_1^0, \dots, a_{k-1}^0) = x_1 \prod_{i=1}^{k-1} a_i^0 \left(\frac{1}{2\pi} \right)^{k-1} \quad (8.5.20)$$

und aufbauend auf (8.5.6) mit dem Rekursionsschritt:

$$f_j(a_1^j, \dots, a_{k-1}^j) = \left(\frac{1}{2\pi} \right) \int_0^{a_1^j} \sum_{\ell=1}^k f_{j-1}(\phi(\ell, a_1^j, \dots, a_{k-1}^j)) dx_j \quad (8.5.21)$$

Hierbei erzeugt die Funktion $\phi(\ell, a_1^j, \dots, a_{k-1}^j)$ die einzelnen Längen der möglichen Bereiche zu den Reisepunkten der Anfragen im nächsten Schritt nach Auswahl der Anfrage u_ℓ^j im aktuellen Schritt. Damit entspricht $\phi(k, a_1^j, \dots, a_{k-1}^j)$ also der Situation nach der Auswahl der neuen Anfrage und $\phi(4, a_1^j, \dots, a_{k-1}^j)$ der Situation bei Auswahl der Anfrage mit dem viert kleinsten Index. Daher ergibt sich

$$\phi(j, a_1^j, \dots, a_{k-1}^j) = (a_1^{j-1}, \dots, a_{k-1}^{j-1})$$

mit $a_k^j = 2\pi$ zu:

$$a_i^{j-1} = \begin{cases} a_i^j - x_j & i < j \\ a_{i+1}^j - x_j & i \geq j \end{cases} \quad (8.5.22)$$

Bemerkung 8.5.2 (Rekursionsformel).

Der Vorfaktor von $\left(\frac{1}{2\pi}\right)^{k-1}$ in (8.5.20) entspricht dem nach der Durchführung der Kürzungen aus (8.5.3) übrig bleibendem Faktor.

Anders als in Unterabschnitt 8.5.1 wird die Rekursionsformel zur Berechnung hier für

den allgemeinen Fall nicht über die Länge des nicht mehr möglichen Bereichs, sondern über die Länge des noch möglichen Bereichs für die Reisepunkte der Anfragen aufgebaut.

Erster Schritt: Die Berechnung von f_1

Bei der Berechnung der rekursiven Funktionsdefinitionen ergibt sich aus (8.5.20) und (8.5.21) für $f_1(a_1^1, \dots, a_{k-1}^1)$ mit einer partiellen Integration folgendes Ergebnis:

$$\begin{aligned}
 f_1(a_1^1, \dots, a_{k-1}^1) &= \left(\frac{1}{2\pi}\right)^k \int_0^{a_1^1} \sum_{j=1}^k f_0(\phi(j, a_1^1, \dots, a_{k-1}^1)) dx_0 \\
 &= \left(\frac{1}{2\pi}\right)^k \int_0^{a_1^1} x_1 \sum_{j=1}^k \prod_{i=1, i \neq j}^k (a_i^1 - x_1) \\
 &= \left(\frac{1}{2\pi}\right)^k \left[x_1 \prod_{i=1}^k (a_i^1 - x_1) \right]_0^{a_1^1} + \left(\frac{1}{2\pi}\right)^k \int_0^{a_1^1} \prod_{i=1}^k (a_i^1 - x_1) \\
 &= \left(\frac{1}{2\pi}\right)^k \int_0^{a_1^1} \prod_{i=1}^k (a_i^1 - x_1) \quad (8.5.23)
 \end{aligned}$$

Das Integral aus (8.5.23) lässt sich bei einer Entwicklung nach $(a_1^1 - x_1)$ über eine partielle Integration auflösen.

$$\begin{aligned}
 \int \prod_{i=1}^k (a_i^1 - x_1) dx_1 &= -\frac{(a_1^1 - x_1)^2}{2} \prod_{i=2}^k (a_i^1 - x_1) - \int \frac{(a_1^1 - x_1)^2}{2} \sum_{j=2}^k \prod_{\substack{i=2, \\ i \neq j}}^k (a_i^1 - x_1) dx_1 \\
 &= -\frac{(a_1^1 - x_1)^2}{2} \prod_{i=2}^k (a_i^1 - x_1) + \frac{(a_1^1 - x_1)^3}{6} \sum_{j=2}^k \prod_{\substack{i=2, \\ i \neq j}}^k (a_i^1 - x_1) \\
 &\quad + \int \frac{(a_1^1 - x_1)^3}{6} \sum_{j=2}^k \sum_{\substack{\ell=2, \\ \ell \neq j}}^k \prod_{\substack{i=2, \\ i \neq j, \ell}}^k (a_i^1 - x_1) dx_1 \quad (8.5.24)
 \end{aligned}$$

Sei dazu S_{j-1} für (8.5.24) bis (8.5.27) so definiert, dass darin alle sortierten Teilmengen von $\{2, \dots, k\}$ mit $j-1$ Elementen, – für $j-1=0$ wird $S_0 = \{\emptyset\}$ gesetzt, – enthalten sind.

Da jede aus den zwei Elementen j und ℓ bestehende Menge in (8.5.24) zweimal mit dem identischen Produkt aus den Termen $(a_i^1 - x_1)$ vorkommt, können die Summen $\sum_{j=2}^k \sum_{\ell=2, \ell \neq j}^k$ zu einer Summe über die sortierten zweier Mengen aus S_2 zusammenge-

fasst werden. Durch das zweimalige Auftreten wird jedoch noch ein Vorfaktor von zwei benötigt, welcher sich allerdings gleich mit dem Sechstel zu einem Drittel kürzen lässt.

$$\begin{aligned}
 & \sum_{j=1}^2 \frac{(j-1)!}{(j+1)!} (-1)^j (a_1^1 - x_1)^{j+1} \sum_{S \in S_{j-1}} \prod_{i \notin (S \cup \{1\})} (a_i^1 - x_1) \\
 & \quad + \int \frac{(a_1^1 - x_1)^3}{3} \sum_{S \in S_2} \prod_{i \notin (S \cup \{1\})} (a_i^1 - x_1) dx_1 \\
 & = \sum_{j=1}^2 \frac{(j-1)!}{(j+1)!} (-1)^j (a_1^1 - x_1)^{j+1} \sum_{S \in S_{j-1}} \prod_{i \notin (S \cup \{1\})} (a_i^1 - x_1) \\
 & \quad - \frac{(a_1^1 - x_1)^4}{12} \sum_{S \in S_2} \prod_{i \notin (S \cup \{1\})} (a_i^1 - x_1) - \int \frac{(a_1^1 - x_1)^4}{12} \sum_{\substack{S \in S_2 \\ j \notin S}} \prod_{i \notin (S \cup \{1\})} (a_i^1 - x_1) dx_1
 \end{aligned} \tag{8.5.25}$$

Aus den sortierten Mengen S von S_2 mit je zwei Elementen und dem Index j entstehen in (8.5.25) Mengen mit drei Elementen. Da es für jede dieser neuen Mengen genau drei Möglichkeiten zur Entstehung gibt, ergibt sich bei einer Zusammenfassung zu den Mengen aus S_3 ein Multiplikator von drei, dieser kann allerdings wieder aus der Formel heraus gekürzt werden.

$$\begin{aligned}
 & \sum_{j=1}^3 \frac{(j-1)!}{(j+1)!} (-1)^j (a_1^1 - x_1)^{j+1} \sum_{S \in S_{j-1}} \prod_{i \notin S \cup \{1\}} (a_i^1 - x_1) \\
 & \quad - \int \frac{(a_1^1 - x_1)^4}{4} \sum_{S \in S_3} \prod_{i \notin S \cup \{1\}} (a_i^1 - x_1) dx_1
 \end{aligned} \tag{8.5.26}$$

Durch eine weiterführende Anwendung der partiellen Integration auf (8.5.26) lässt sich das ganze Integral in das verwendete Muster bringen. Dieses kann anschließend mit den Integrationsgrenzen 0 und a_1^1 , sowie dem Vorfaktor $\left(\frac{1}{2\pi}\right)^k$ ausgewertet werden zu:

$$\begin{aligned}
 f_1(a_1^1, \dots, a_{k-1}^1) & = \left(\frac{1}{2\pi}\right)^k \left[\sum_{j=1}^k \frac{(j-1)!}{(j+1)!} (-1)^j (a_1^1 - x_1)^{j+1} \sum_{S \in S_{j-1}} \prod_{i \notin S \cup \{1\}} (a_i^1 - x_1) \right]_0^{a_1^1} \\
 & = \left(\frac{1}{2\pi}\right)^k \sum_{j=1}^k \frac{(j-1)!}{(j+1)!} (-1)^{j+1} (a_1^1)^{j+1} \sum_{S \in S_{j-1}} \prod_{i \notin S \cup \{1\}} a_i^1
 \end{aligned} \tag{8.5.27}$$

Bemerkung 8.5.3 (Andere Herleitung der Formel (8.5.27)).

Das identische Ergebnis kann auch erreicht werden, wenn in (8.5.23) das Produkt über alle Terme $(a_i^1 - x)$ ausmultipliziert wird, und über die dadurch entstehende Summe integriert wird, sowie danach die Integrationsgrenzen 0 und a_1^1 eingesetzt werden. Nach einer Zusammenfassung der Summanden mit gleichen a_i^1 Potenzen ist dann ebenfalls das in (8.5.27) angewandte Muster erkennbar.

Bei der Auswertung des Ergebnisses von (8.5.27) die Berechnung der ersten Schrittweite beim Start des Algorithmus 8.1.2, gilt die Belegung $a_1^1 = \dots = a_k^1 = 2\pi$ für die Längen der möglichen Bereiche zu den Reisepunkten der verbliebenen Anfragen aus dem Informationsvorlauf. Damit ergibt sich aus (8.5.27) durch direktes Einsetzen:

$$\mathbb{E}_k[\tau_1] = f_1(2\pi, \dots, 2\pi) = \left(\frac{1}{2\pi}\right)^k \sum_{j=1}^k \frac{(j-1)!}{(j+1)!} (-1)^{j+1} (2\pi)^{j+1} \sum_{S \in S_{j-1}} \prod_{i \notin S \cup \{1\}} 2\pi \quad (8.5.28)$$

Da es genau $\binom{k-1}{j-1}$ mögliche Mengen S aus S_{j-1} gibt, sowie außerdem aus $i \notin S \cup \{1\}$ direkt $i \in \{2, \dots, k\} \setminus S$ folgt und damit das Produkt über genau $k-j$ Elemente geht, lässt sich (8.5.28) weiter vereinfachen zu:

$$\begin{aligned} \mathbb{E}_k[\tau_1] &= \left(\frac{1}{2\pi}\right)^k \sum_{j=1}^k \frac{(j-1)!}{(j+1)!} (-1)^{j+1} (2\pi)^{j+1} \binom{k-1}{j-1} (2\pi)^{k-j} \\ &= 2\pi \sum_{j=1}^k (-1)^{j+1} \frac{(k-1)!}{(j+1)!(k-j)!} = \frac{2\pi}{k(k+1)} \sum_{j=1}^k (-1)^{j+1} \binom{k+1}{j+1} \end{aligned} \quad (8.5.29)$$

Mit einer Substitution von $j+1$ durch i verschiebt sich die Summe in (8.5.29) wie folgt:

$$\mathbb{E}_k[\tau_1] = \frac{2\pi}{k(k+1)} \sum_{i=2}^{k+1} (-1)^i \binom{k+1}{i} \quad (8.5.30)$$

Nach der Ergänzung der bei $i = 2$ startenden Summe in (8.5.30) um die fehlenden Glieder mit dem Wert 1 für $i = 0$ und dem Wert $-(k+1)$ für $i = 1$ lässt sich die ganze Formel noch einmal deutlich vereinfachen.

$$\begin{aligned} \mathbb{E}_k[\tau_1] &= \frac{2\pi}{k(k+1)} \left(-1 + (k+1) + \sum_{i=0}^{k+1} (-1)^i \binom{k+1}{i} \right) \\ &= \frac{2\pi}{k+1} + \frac{2\pi}{k(k+1)} (1-1)^{k+1} = \frac{2\pi}{k+1} \end{aligned} \quad (8.5.31)$$

Damit wurde die Aussage von Satz 8.4.1 über die erwartete Schrittweite des ersten Schrittes erneut bestätigt.

Erkenntnisse

Aus der Berechnung von f_1 lassen sich folgende Schlüsse ziehen:

- Generell sollte die Berechnung von f_t auch im Allgemeinen möglich sein, dabei kann die Integration über das Produkt aus den Längen $(a_i - x)$ der Bereiche zu den Reisepunkten der Anfragen zwar grundsätzlich wie oben gesehen durchgeführt werden, allerdings enthält der nächste Schritt dann für jedes dieser Produkte eine Summe über jede Teilmenge mit wiederum jeweils derartigen Produkten als Summanden.
- Zusätzlich führt der Fall mit der Auswahl der ältesten Anfrage als nächstes Ziel dazu, dass nach Anwendung der Rekursionsformel (8.5.21) nicht nur Ausdrücke mit a_1 als maximale Schrittweite – diese ist die obere Grenze bei der Integration, damit stellt (8.5.27) schließlich ein Polynom über diese Schrittweite dar – vorkommen, sondern auch ein Ausdruck bei welchem a_2 die Rolle von a_1 übernimmt. Das heißt, dass immerhin die Ausdrücke zu den anderen Möglichkeiten bei der Auswahl der nächsten Anfrage ähnlich aufgebaut sind, der eine spezielle Ausdruck für die Auswahl der neuen Anfrage jedoch zwar eine analoge Struktur besitzt, dabei unterscheidet sich dessen konkreter Aufbau doch deutlich von den anderen. Daher lässt sich dieser auch nicht einfach mit den anderen in einen Ausdruck zusammenfassen.

9. Verwendung von Informationsvorlauf bei der Rundreise auf der Kreislinie mit Richtungswechsel

Anders als in Kapitel 8 wird in diesem Kapitel mit der Problemstellung für die Rundreise auf der Kreislinie mit Richtungswechsel nach Optimierungsproblem 7.1.1 gearbeitet. Das heißt somit aber auch, dass bei der Bearbeitung einer Sequenz von Anfragen die Richtung der Bewegung gewechselt werden darf.

Diese Aufgabe der Einschränkung auf eine Richtung, in welcher die Bewegung erfolgen musste, führt allerdings bei der Problemstellung zu einer zusätzlichen Komplexität, da nun immer zwischen den beiden möglichen Laufrichtungen unterschieden werden muss. Dadurch ergeben sich außerdem Veränderungen bei der Betrachtung des noch möglichen Bereichs der Reisepunkte der nach der Auswahl der nächsten zu bearbeitenden Anfrage verbleibenden unbearbeiteten Anfragen.

9.1. Algorithmus

Der verwendete Algorithmus ist relativ einfach und baut auf den Algorithmus „Nearest-Neighbour“, welcher zum Beispiel bei [Bor01, Abschnitt 26.2] gefunden werden kann, auf.

Algorithmus 9.1.1 (Greedy für die Rundreise auf der Kreislinie).

- (i) Es werden die ersten k Anfragen $\sigma_1, \dots, \sigma_k$ ausgelesen, $g_0 = v_0$ ist der Startpunkt und damit aktueller Standort, weiterhin wird $i = 1$ gesetzt.
- (ii) Diejenige der bekannten und nicht bearbeiteten Anfragen wird ausgewählt, deren Abstand zu g_{i-1} minimal ist. Der Abstand zwischen zwei Punkten v_j und v_t berechnet sich dabei wie in Definition 7.3.1 als

$$\text{dist}(v_j, v_t) = \min \{|v_j - v_t|, 2\pi - |v_j - v_t|\}.$$

- (iii) Die ausgewählte Anfrage $g_i = \sigma_t$ mit Reisepunkt v_t wird nun bearbeitet und die neue Position des Algorithmus ist danach v_t . Die Anfrage σ_t wird außerdem aus der Liste der Anfragen, welche noch nicht bearbeitet wurden, entfernt.
- (iv) Die Anfrage σ_{i+k} wird ausgelesen und zur Bearbeitung freigegeben, und der Zähler i wird auf den Wert $i + 1$ gesetzt. Danach geht es wieder mit der Auswahl der nächsten Anfrage in Schritt (ii) weiter.

Untersucht wird in diesem Kapitel die durchschnittliche Schrittweite des Algorithmus („Average-Case“-Analyse) bei unterschiedlich großen Werten für den Informationsvorlauf. Da die Betrachtung möglichst umfassend sein soll, wird zur Messung der durchschnittlichen Schrittweite nicht nur die empirische Analyse verwendet, sondern auch eine rekursive Formel für die exakte Berechnung des Erwartungswert für die Schrittweiten bei der Bearbeitung einer unendlichen Sequenz von Anfragen hergeleitet.

Als Informationsvorlauf wird wieder eine Bearbeitungsinformation von k Anfragen nach Definition 6.2.1 verwendet. Das Ordnungsmerkmal der Anfragen ist der Index innerhalb der Sequenz aller Anfragen, also die Reihenfolge der Anfragen in der Sequenz.

9.2. Simulation einer Rundreise auf der Kreislinie mit Richtungswechsel

Als Veranschaulichung für die Bearbeitung von Optimierungsproblem 7.1.1 mithilfe von Algorithmus 9.1.1 soll Beispiel 9.2.1 mit einer Sequenz von zwanzig Anfragen sowie einer Bearbeitungsinformation von fünf Anfragen dienen.

Beispiel 9.2.1 (Beispiel zur Rundreise auf der Kreislinie mit Richtungswechsel).

Die Reisepunkte der für die Bearbeitung von erst einmal unbekannten Anfragen werden auf der äußeren Kreislinie gezeichnet und die Reisepunkte der schon bekannten Anfragen auf dem inneren Kreis. Der große Punkt stellt die aktuelle Position des Algorithmus dar.

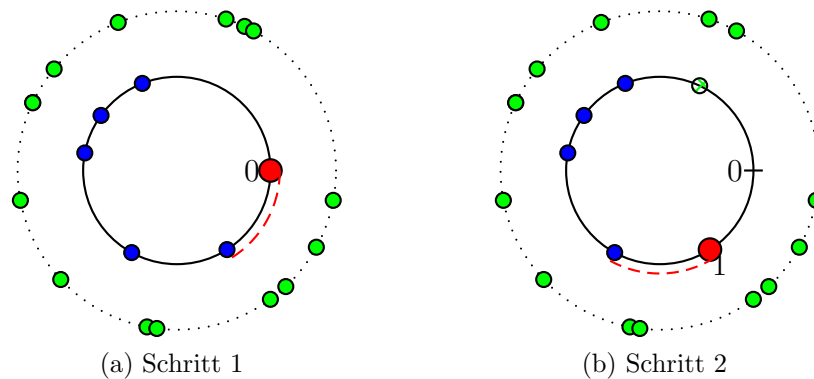


Abbildung 9.2.1.: Erster und zweiter Schritt für eine Rundreise auf der Kreislinie mit Richtungswechsel

In den beiden ersten Schritten ergibt sich eine Bewegung, dargestellt durch die gestrichelte Linie knapp oberhalb des innersten Kreises, entlang des Uhrzeigersinns, da auf der inneren Kreislinie in dieser Richtung der nächste der noch nicht bearbeiteten Reisepunkte liegt. Nach dem Erreichen der ersten ausgewählten Anfrage wird die Information über den nächsten Reisepunkt (der schraffierte Punkt auf dem inneren Kreis) aus der Anfrage-Sequenz erhalten. Das bedeutet, dass einer der Reisepunkte von der äußeren Kreislinie, nämlich derjenige, dessen Anfrage in der Sequenz direkt nach dem ersten Informationsvorlauf kommt, auf die innere Kreislinie wechselt.

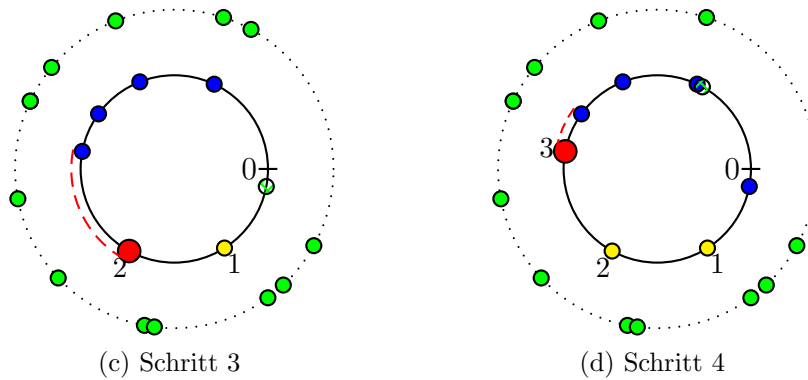


Abbildung 9.2.1 (Fortsetzung): Dritter und vierter Schritt für eine Rundreise auf der Kreislinie mit Richtungswechsel

Auch im dritten und vierten Schritt bleibt die bisherige Laufrichtung erhalten, allerdings ist die Schrittweite im vierten Schritt gegenüber den vorherigen Schritten deutlich kleiner. Die Reisepunkte der schon bearbeiteten Anfragen werden nun immer als helle Punkte auf der inneren Kreislinie dargestellt. Weiterhin steht neben jedem Reisepunkt einer schon bearbeiteten Anfrage jeweils die Nummer des Schrittes, in welchem diese bearbeitet wurde.

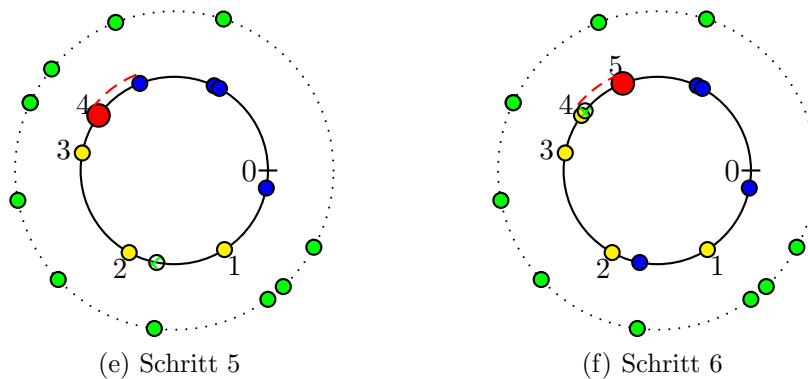


Abbildung 9.2.1 (Fortsetzung): Fünfter und sechster Schritt für eine Rundreise auf der Kreislinie mit Richtungswechsel

Im fünften Schritt bleibt weiterhin alles beim alten, allerdings positioniert sich diesmal der Reisepunkt der danach neu hinzukommenden Anfrage so nahe an der neuen Position, dass der Abstand zu diesem Reisepunkt minimal ist. Da im sechsten Schritt eine Bewegung vom aktuellen Standort zu diesem neuen Reisepunkt entgegen dem Uhrzeigersinn die kleinere Schrittweite hat, wird nun das erste Mal die Laufrichtung gewechselt.

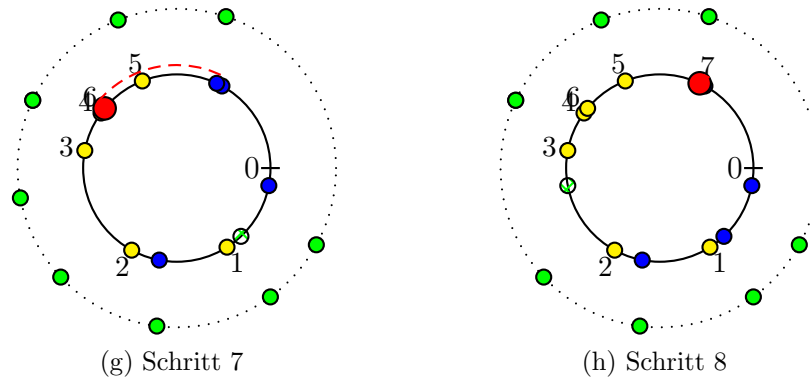


Abbildung 9.2.1 (Fortsetzung): Siebter und achter Schritt für eine Rundreise auf der Kreislinie mit Richtungswechsel

Nach dem Wechsel der Richtung im sechsten Schritt, siehe Abbildung 8.2.1f, ist die Bewegung zum nächstgelegenen Reisepunkt erneut mit einem Wechsel der Richtung verbunden, so dass die Laufrichtung wieder im Uhrzeigersinn ist. Im achten Schritt schließlich ist keine richtige Bewegung erkennbar, da der nächste Zielpunkt zu nahe an dem aktuellen Standort liegt.

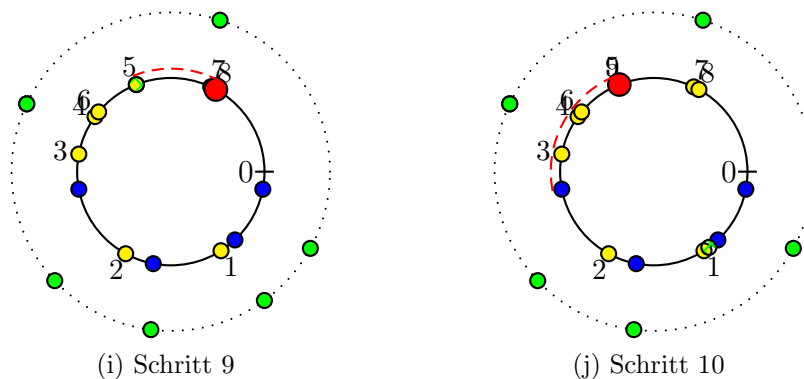
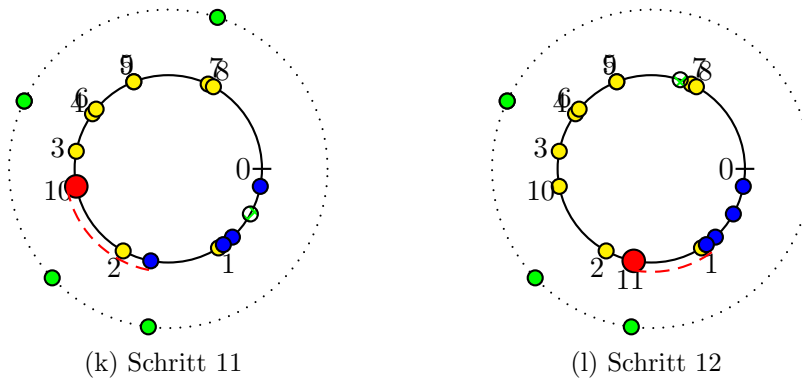


Abbildung 9.2.1 (Fortsetzung): Neunter und zehnter Schritt für eine Rundreise auf der Kreislinie mit Richtungswechsel

Im Übergang vom achten zum neunten Schritt wird abermals die Laufrichtung gewechselt, es geht jetzt also entgegen dem Uhrzeigersinn weiter, diese Bewegungsrichtung bleibt nun bis zum Ende des Beispiels erhalten.



(k) Schritt 11

(l) Schritt 12

Abbildung 9.2.1 (Fortsetzung): Elfter und zwölfter Schritt für eine Rundreise auf der Kreislinie mit Richtungswechsel

9.3. Empirische Analyse

Innerhalb dieses Abschnitts soll eine empirische Auswertung der erwarteten Schrittweite für den Algorithmus 9.1.1 betrachtet werden. Die Grundvoraussetzungen sind genauso wie am Anfang von Abschnitt 8.3 aufgeführt.

9.3.1. Durchschnittliche Schrittweite

In Abbildung 9.3.1 ist der Verlauf der gesamten durchschnittlichen Schrittweite (das sind die Punkte) innerhalb des Algorithmus 9.1.1 dargestellt. Die genauen Werte der empirischen Ergebnisse sind in Tabelle B.2.1 aufgelistet. Zusätzlich wurde als Vergleich außerdem noch der Funktionsverlauf von $\frac{1}{k+1}$, das ist die Linie, aufgenommen. Dieser Funktionsverlauf entspricht der erwarteten Schrittweite (siehe Satz 9.4.1) des Algorithmus 9.1.1, wenn der verbleibende Informationsvorlauf von $k-1$ Anfragen nach der Bearbeitung der ausgewählten Anfrage nicht mit einer neuen Anfrage aus der vorher generierten Sequenz von Anfragen aufgefüllt wird, sondern diese verbleibenden Anfragen entfernt und k neue auf der Kreislinie gleichverteilte Anfragen generiert werden.

Die Positionierung der Punkte im Vergleich zu der Linie zeigt, dies ist vor allem im ersten Teil der Abbildung 9.3.1 deutlich zu sehen, dass sich bei der Bearbeitung der Problemstellung aus Optimierungsproblem 7.1.1, nämlich der Rundreise auf der Kreislinie mit Richtungswechsel, mit dem Algorithmus 9.1.1 im Vergleich zu der Analyse in Kapitel 8 für die gerichtete Rundreise doch einiges verändert. Bei der allgemeineren Variante sind die im Durchschnitt über alle Schritte in der vom Algorithmus 9.1.1 gefundenen Rundreise errechneten Schrittweiten, größer als die erwartete Schrittweite des ersten Schrittes. Diese Erkenntnis wird später unter anderem in Unterabschnitt 9.4.2

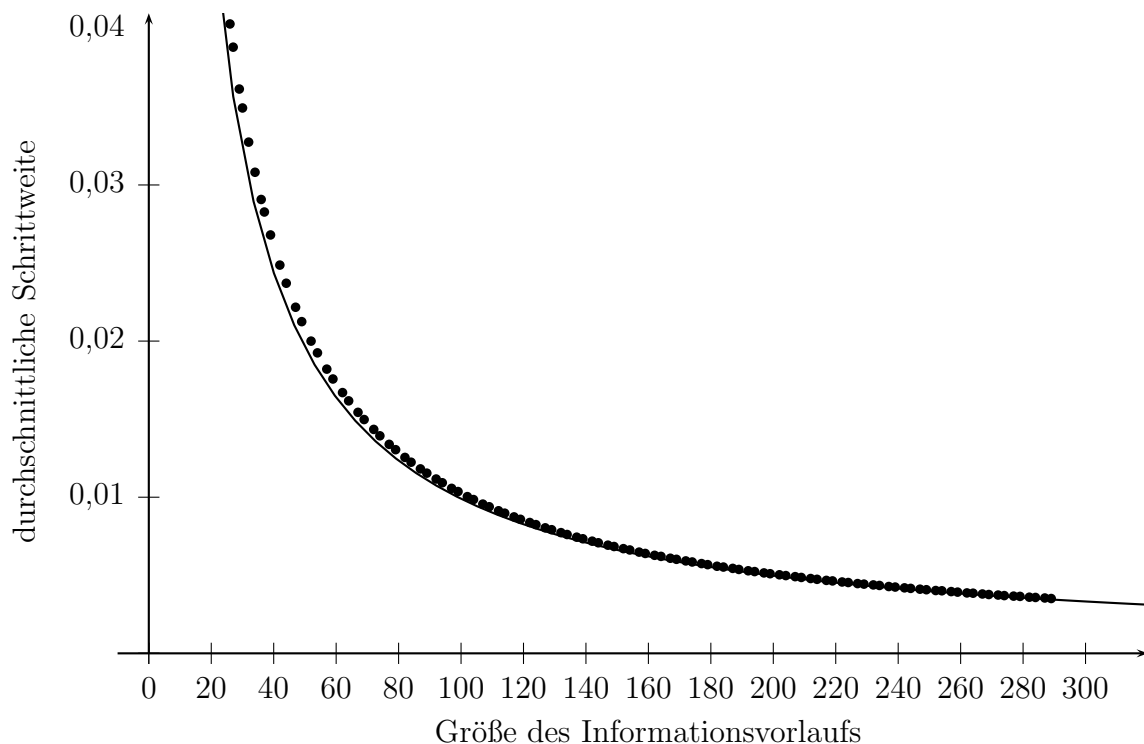


Abbildung 9.3.1.: Empirische Werte für die durchschnittliche Schrittweite bei unterschiedlichem Informationsvorlauf zur Rundreise auf der Kreislinie mit Richtungswechsel

durch den dort berechneten Erwartungswert für die im zweiten Schritt zu erwartende Entfernung bestärkt.

Ein Grund für dieses veränderte Verhalten liegt darin, dass ein Wechsel der Richtung während der Bearbeitung, meist zum Reisepunkt der gerade neu hinzugekommenen Anfrage, dafür sorgt, dass in einem Bereich um den dann neu erreichten Standort kein Reisepunkt einer der verbliebenen Anfragen liegen kann, sondern einzig allein nur noch der Reisepunkt der neu hinzukommenden Anfrage. Damit ist dann, falls im nächsten Schritt eine der verbliebenen Anfragen ausgewählt wird, schon einmal eine echt positive minimale Schrittweite vorgegeben. Dieser Sachverhalt wird später in Abbildung 9.5.1 deutlich dargestellt.

Bei der Rundreise auf der Kreislinie mit Wechsel der Bewegungsrichtung kann also eine Bearbeitung der Sequenz aus Anfragen mit jeweils k neuen Anfragen potenziell eine geringere durchschnittliche Schrittweite haben. Das heißt, dass der Verbleib der nicht ausgewählten Anfragen nicht immer automatisch zu einer Verringerung der späteren Schrittweiten führen muss. Allerdings würden, wenn die $k - 1$ nicht ausgewählten Anfragen verworfen werden, natürlich nicht alle Anfragen bearbeitet werden, sondern immer nur eine Anfrage von k Anfragen. Dies würde dann dazu führen, dass insgesamt nur $\lceil \frac{n}{k} \rceil$

Anfragen aus einer Sequenz von n Anfragen bearbeitet werden können, die restlichen $n - \lceil \frac{n}{k} \rceil$ würden unbearbeitet bleiben. Damit bietet diese Form der Bearbeitung, unter der eigentlich gewünschten Zielsetzung einer vollständiger Abarbeitung aller Anfragen, keine wirkliche Alternative zum Algorithmus 9.1.1.

Wird eine nichtlineare Regressionsanalyse unter Verwendung von $\frac{\alpha}{k+\beta}$ als Modell, und den Startwerten $\alpha = 1$ sowie $\beta = 1$, auf die empirischen Daten aus Tabelle B.2.1 zu den gesamten durchschnittlichen Schrittweiten angewandt, so ergeben sich mit [Tea09, R] die in Tabelle 9.3.1 zusammengetragenen Ergebnisse. Die mit dem Modell errechneten Werte weichen von den empirischen durchschnittlichen Schrittweiten bei der quadratischen Gewichtung der Unterschiede in der Summe um $RSS = 5,4 \cdot 10^{-5}$ ab. Daraus ergibt sich ein Bestimmtheitsmaß von $R^2 = 0,9998608$.

Parameter	Schätzwert	Standardfehler	t -Wert	P ($> t $)
α	1,113099	0,002682	415,0	$< 2 \cdot 10^{-16}$
β	1,332890	0,011497	115,9	$< 2 \cdot 10^{-16}$

Tabelle 9.3.1.: Ergebnisse zur nichtlinearen Regression der durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel

Anders als bei der Regressionsanalyse in Abschnitt 8.3 für die gerichtete Rundreise weichen hier nun die mit den verschiedenen Modellen erstellten Funktionen zur Regression, welche sich zusammengefasst in Tabelle 9.3.2 finden lassen, doch deutlich von einander ab. Insgesamt lässt sich daher keine durch alle betrachteten Modelle gestützte Tendenz zu einer bestimmten Funktion feststellen. Außerdem treten bei den beiden Modellen mit zwei Parametern entgegen den Analysen aus Abschnitt 8.3 keine so extremen Unterschiede in den t -Werten der jeweiligen Parameter auf.

Modell	Parameter	RSS	R^2	t -Werte
$\frac{1}{k^\alpha}$	$\alpha = 1,10062$	0,0265	0,931675	72,27
$\frac{\alpha}{(k+\beta)}$	$\alpha = 1,113099; \beta = 1,332891$	$5,4 \cdot 10^{-5}$	0,9998608	415,0; 115,9
$\frac{\beta}{k^\alpha}$	$\alpha = 0,839745; \beta = 0,636802$	0,00202	0,9947918	124,4; 81,7

Tabelle 9.3.2.: Ergebnisse der verschiedenen Modelle zur Rundreise auf der Kreislinie mit Richtungswechsel

9.3.2. Modifizierte durchschnittliche Schrittweite

Aus den gleichen Gründen wie in Abschnitt 8.3, bei der empirischen Betrachtung von Optimierungsproblem 8.1.1 unter Verwendung von Algorithmus 8.1.2, wird auch hier in

Abbildung 9.3.2 eine nach Definition 8.3.2 um die ersten und letzten k Schritte bereinigte Berechnung der durchschnittlichen Schrittweite herangezogen. Die dazu gehörenden

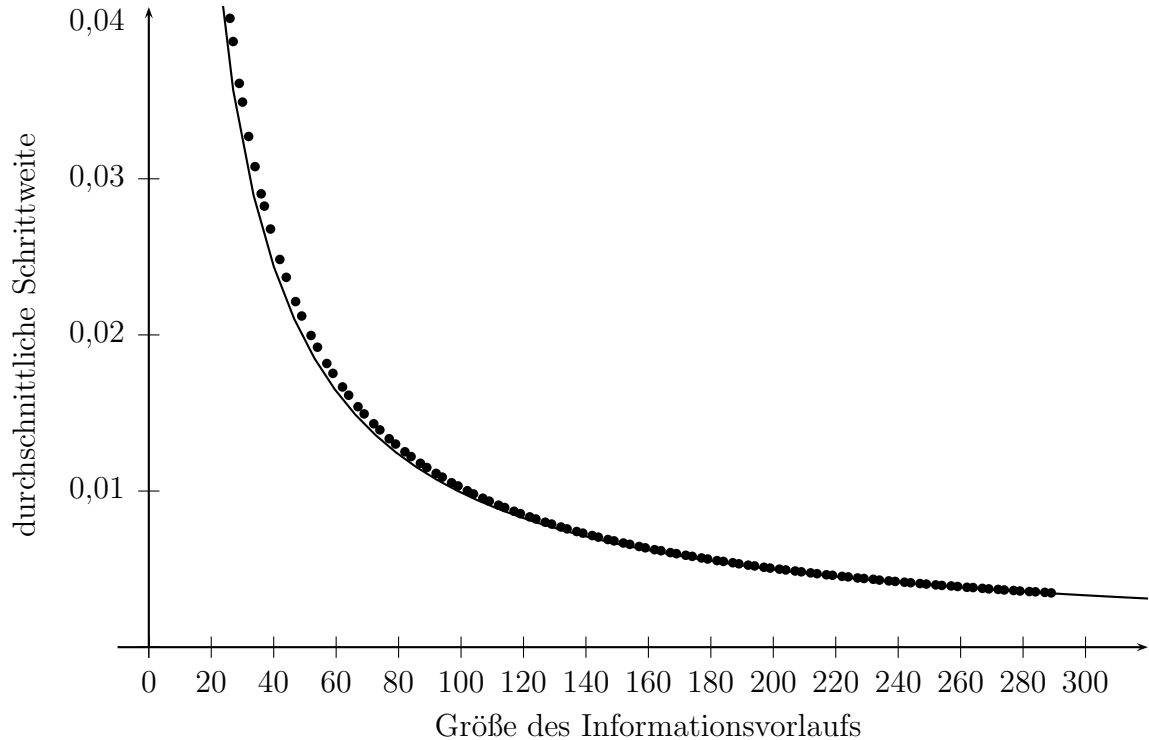


Abbildung 9.3.2.: Bereinigte empirische Werte für die durchschnittliche Schrittweite zur Rundreise auf der Kreislinie mit Richtungswechsel

Daten finden sich in Tabelle B.2.2. An der Struktur der Abbildung 9.3.2 verändert sich im Vergleich zur Abbildung 9.3.1 fast nichts.

Eine Analyse in [Tea09, R] mit einem nichtlinearen Regressionsmodell der Form $\frac{\alpha}{k+\beta}$ und den Startwerten $\alpha = 1$, sowie $\beta = 1$ ergibt die Ergebnisse in Tabelle 9.3.3 und einen quadratischen Fehler von $RSS = 5,78 \cdot 10^{-5}$ mit einem Bestimmtheitsmaß von $R^2 = 0,999851$. Hierbei ist das Bestimmtheitsmaß jedoch sogar geringfügig kleiner als das Bestimmtheitsmaß zu der gesamten durchschnittlichen Schrittweite.

Parameter	Schätzwert	Standardfehler	t -Wert	P ($> t $)
α	1,12292	0,002774	400,9	$< 2 \cdot 10^{-16}$
β	1,330029	0,011889	111,9	$< 2 \cdot 10^{-16}$

Tabelle 9.3.3.: Ergebnis zur nichtlinearen Regression der bereinigten durchschnittlichen Schrittweite für die Rundreise auf der Kreislinie mit Richtungswechsel

Das bedeutet, dass anders als in Unterabschnitt 8.3.2 bei der gerichteten Rundrei-

se auf der Kreislinie hier keine echte Verbesserung der Güte der Regression durch die Nichtberücksichtigung der Schrittweiten der ersten k Schritte und der letzten k Schritte entsteht. Weiterhin liefert die Regression auf den bereinigten empirischen durchschnittlichen Schrittweiten etwas schlechtere Standardfehler und auch geringere t -Werte.

Der Vergleich der verschiedenen Modelle zur Regression, in der Tabelle 9.3.4 sind die relevanten Ergebnisse zusammengefasst, liefert auch für die bereinigten empirischen durchschnittlichen Schrittweiten kein einheitliches Bild.

Modell	Parameter	RSS	R^2	t -Werte
$\frac{1}{k^\alpha}$	$\alpha = 1,10083$	0,0265	0,9316505	72,3
$\frac{\alpha}{(k+\beta)}$	$\alpha = 1,112293; \beta = 1,30030$	$5,78 \cdot 10^{-5}$	0,999851	400,9; 111,9
$\frac{\beta}{k^\alpha}$	$\alpha = 0,840149; \beta = 0,637071$	0,00204	0,9947384	123,86; 81,37

Tabelle 9.3.4.: Ergebnisse der verschiedenen Modelle für die Rundreise auf der Kreislinie mit Richtungswechsel

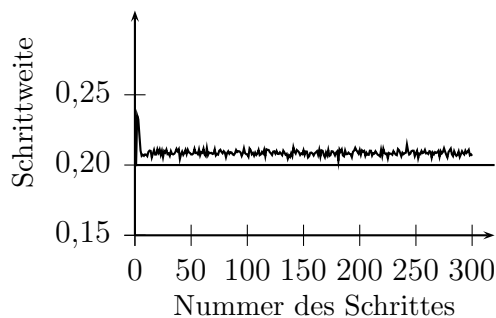
9.3.3. Schrittweiten für einzelne Werte des Informationsvorlaufs

In Abbildung 9.3.3 finden sich für einen Informationsvorlauf von $k = 4, 29, 69, 129$ jeweils die Schrittweiten der ersten dreihundert Schritte. Zusätzlich ist in diesen vier Abbildungen eine waagrechte Linie mit dem erwarteten Wert der ersten Schrittweite $\frac{1}{k+1}$ eingezeichnet. Dabei ist jeweils ersichtlich, dass der Grenzwert über dem erwarteten Wert für die Schrittweite im ersten Schritt liegt.

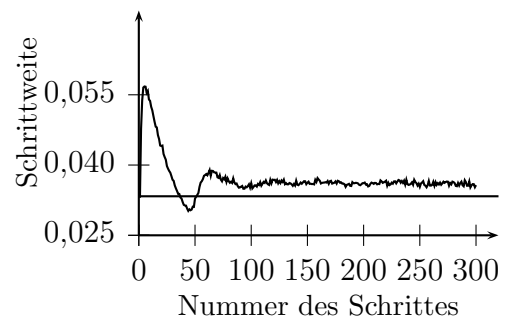
Im Unterschied zu der gerichteten Rundreise auf der Kreislinie mit Abbildung 8.3.3 ist in Abbildung 9.3.3 deutlich erkennbar, dass die Schrittweite zwischen den einzelnen Stationen am Anfang des Algorithmus 9.1.1 zuerst einmal ziemlich ansteigt. Danach sinkt diese mit einer geringeren Steigung wieder ab um sich dann mit einem zyklischen Verlauf dem Grenzwert anzupassen. Dieses Verhalten lässt sich insbesondere in den Abbildungen zu den größeren Werten des Informationsvorlaufs erkennen.

In diesem Verhalten liegt im besonderen auch die Begründung dafür, dass sich bei der Regressionsanalyse über die um die ersten und letzten k Schrittweiten bereinigte Sequenz der Schrittweiten keine Verbesserung des Bestimmtheitsmaßes ergeben hat. Bei den Abbildungen 9.3.3c und 9.3.3d ist nämlich erkennbar, dass der erste Abschnitt nicht nur wie bei der gerichteten Rundreise auf der Kreislinie eine Hinführung auf den späteren Verlauf der Kurve ist. Durch den darin enthaltenen aufsteigenden und dann wieder absteigenden Abschnitt der Kurve entspricht dieser Teil eindeutig dem weiteren Verlauf der Kurve.

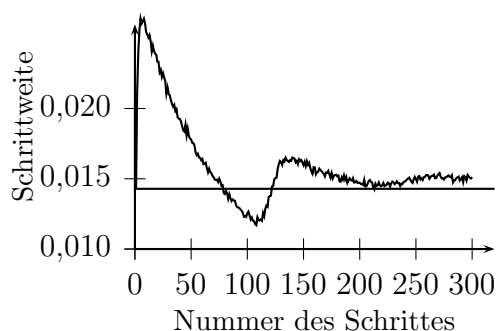
Das bedeutet damit aber auch, dass durch das Entfernen der Schrittweiten zu den ersten k ausgewählten Anfragen nicht nur eine Abnormität, wie etwa der rasante Anstieg der Schrittweiten am Ende der Sequenz der Anfragen, beseitigt wird, sondern ein



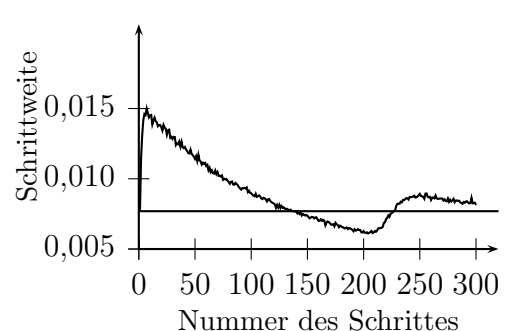
(a) Entwicklung der Schrittweite für einen Informationsvorlauf von 4



(b) Entwicklung der Schrittweite für einen Informationsvorlauf von 29



(c) Entwicklung der Schrittweite für einen Informationsvorlauf von 69



(d) Entwicklung der Schrittweite für einen Informationsvorlauf von 129

Abbildung 9.3.3.: Die Schrittweiten der ersten dreihundert Schritte für einen Informationsvorlauf von 4, 29, 69 und 129 bei der Rundreise auf der Kreislinie mit Richtungswechsel

zum eigentlichen Verlauf gehörender Abschnitt der Kurve. Als Folgerung daraus bleibt die Feststellung, dass sich die Situation bei der Auswahl der ersten k zu bearbeitenden Anfragen, offensichtlich rein vom Verlauf her gesehen, nicht so sehr von der Situation bei der Auswahl der Anfragen im mittleren Abschnitt der Sequenz der Anfragen unterscheidet. Bei den absoluten Werten ist allerdings immer noch ein deutlicher Unterschied erkennbar.

Anders als bei den Analysen zu den ersten dreihundert Schrittweiten für die gerichtete Rundreise auf der Kreislinie in Unterabschnitt 8.3.3 kann hier nach dem Schritt k nicht davon ausgegangen werden, dass vermehrt die verbliebenen Anfragen aus dem allerersten Informationsvorlauf abgearbeitet werden. Dieser Unterschied ergibt sich dadurch, dass mit der Möglichkeit des Wechsels der Bewegungsrichtung nicht notwendiger Weise jeder Bereich der Kreislinie nach einer gewissen Zeit erreicht worden sein muss. Mit jedem Wechsel der Bewegungsrichtung wird nämlich immer erst einmal ein schon im vorherigen Schritt abgefahrter Bereich der Kreislinie erreicht.

Jedoch ergibt sich nach einiger Zeit, zumindest im Mittel über alle untersuchten In-

stanzen gesehen, doch eine Tendenz hin zu dem Bereich der verbliebenen Anfragen aus dem ersten Informationsvorlauf, dies resultiert dann in natürlicher Weise in einer Verringerung der erzielten Schrittweiten. Nach dem Erreichen dieses Bereichs ergibt sich eine Phase von Bewegungen mit, im Vergleich zu den Anfangs auftretenden Schrittweiten, ziemlich kleinen Schrittweiten. Während dieser Phase häuft sich schließlich wieder die Anzahl der neu bekannt werdenden Anfragen deren Reisepunkt sich nicht in diesem Bereich oder dessen Nähe befindet.

Wegen des sehr niedrigen Niveaus der Schrittweiten in dieser Phase ergibt sich beim Verlassen dieses Bereich eine Überreaktion, welche dazu führt, dass die Schrittweite über das Niveau steigt, das bei den späteren Schritten erreicht wird. Dies führt dann wiederum zu verringerten Schrittweiten und insgesamt ergibt sich daher eine zyklische Annäherung an die schließlich erreichte durchschnittliche Schrittweite.

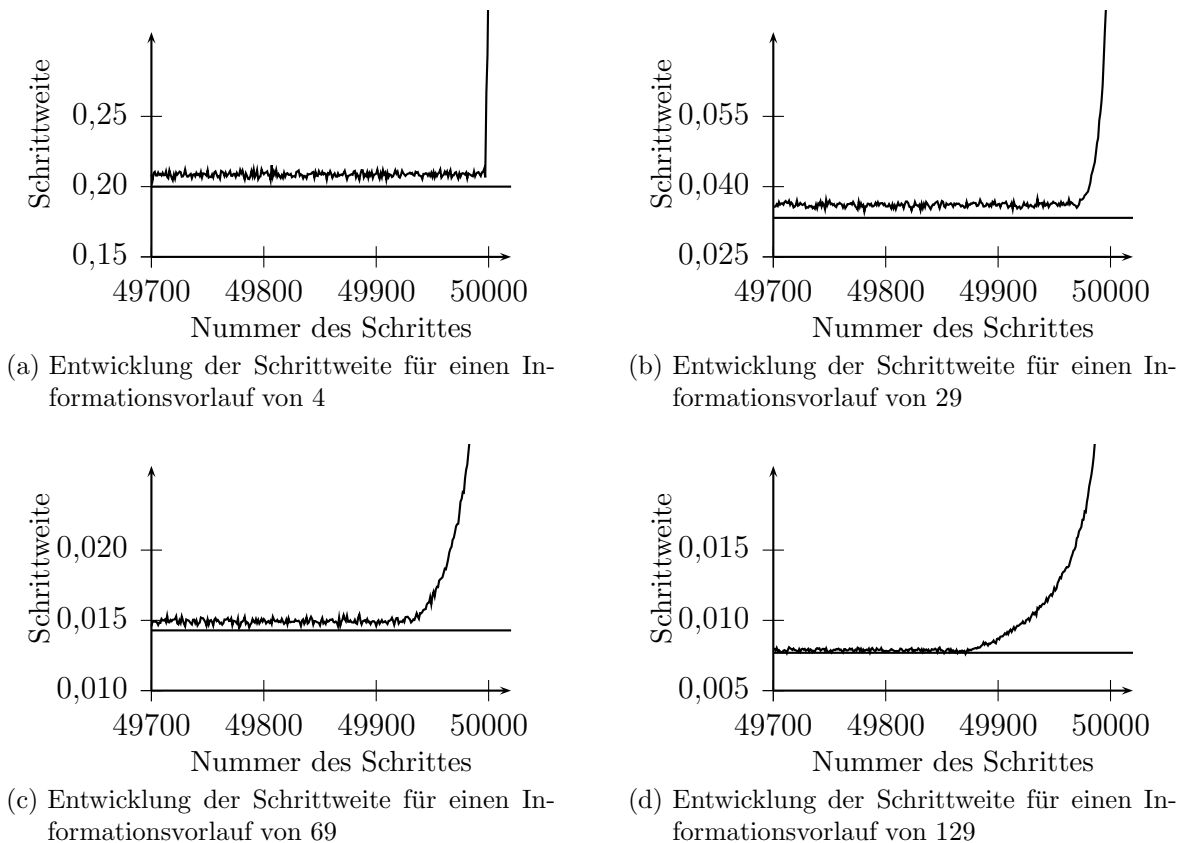


Abbildung 9.3.4.: Die Schrittweiten der letzten dreihundert Schritte für einen Informationsvorlauf von 4, 29, 69 und 129 bei der Rundreise auf der Kreislinie mit Richtungswechsel

Der Verlauf der Kurven in Abbildung 9.3.4 mit den Schrittweiten zu den letzten dreihundert Schritten liefert hingegen keine Überraschung, sondern verhält sich grundsätzlich weitgehend analog zu den entsprechenden Kurven in Abbildung 8.3.4. Das heißt,

dass sich zumindest am strukturellen Verlauf der Schrittweiten am Ende der Sequenz der Anfragen durch die zusätzliche Laufrichtung nicht viel verändert hat.

Auch hier ist wieder das schon mehrmals angesprochene Verhalten der Längen der Schrittweiten gegenüber der erwarteten ersten Schrittweite erkennbar. So ist in fast jeder der vier Abbildungen, bis auf diejenige zum Informationsvorlauf von 129 Anfragen, deutlich sichtbar, dass auch der langfristige Grenzwert der Schrittweite über den erwarteten Wert der ersten Schrittweite liegt.

9.3.4. Schrittweiten der einzelnen Schritte unter Veränderung des Informationsvorlaufs

Die oberste der drei Kurven in der Abbildung 9.3.5 stellt den Verlauf der Schrittweite im zehnten Schritt dar, die darunter liegende schwarze Kurve beinhaltet die Werte der ersten Schrittweite. Dabei fällt gleich auf, dass zwischen diesen beiden Schrittweiten aus der empirischen Berechnung ein deutlicher Abstand besteht, dies bestätigt die Einschätzung von Unterabschnitt 9.3.1, da die Schrittweiten für den früheren/kleineren Schritt die aus den unteren Punkten entstehende Kurve ergeben. Die graue Kurve entspricht schließlich dem empirischen Verlauf des Schrittes 70.

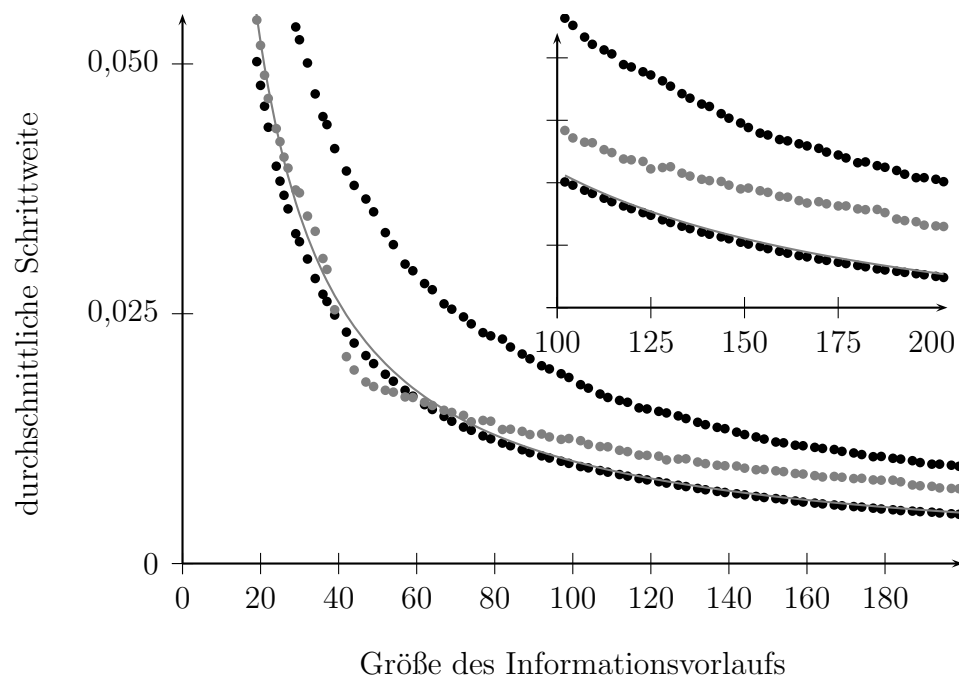


Abbildung 9.3.5.: Schrittweiten der Schritte 1, 10 sowie 70 unter Variation des Informationsvorlaufs für die Rundreise auf der Kreislinie mit Richtungswechsel

Der graphische Verlauf für einen größeren Informationsvorlauf als hundert Anfragen wurde in der rechten oberen Ecke der Abbildung 9.3.5 zur Verdeutlichung noch einmal

mit verändertem Maßstab eingezeichnet.

Bei den Schrittweiten für den Schritt 70 ergibt sich ein deutlich unterschiedlicher Verlauf für einen Informationsvorlauf von $k < 70$, also einer kleineren Größe des Informationsvorlaufs als die Nummer des Schrittes, und für den zweiten Bereich mit $k > 70$. Im ersten Teil fällt die Schrittweite ziemlich schnell und mit zunehmenden Informationsvorlauf wird der Verlauf dieses Abstiegs unregelmäßiger. Im zweiten Teil, in diesem sind im Informationsvorlauf für den Schritt 70 noch mindestens $k - 70$ verbliebene Anfragen aus dem allerersten Informationsvorlauf vorhanden, nimmt das Gefälle deutlich ab.

Als Vergleich ist in der Abbildung 9.3.5 ebenfalls noch die sich aus den empirischen durchschnittlichen Schrittweiten ergebene Kurve als Linie eingezeichnet. Im ersten Teil der Graphik ist diese noch zu erkennen, sie wird dann allerdings im zweiten Teil von den Punkten der Schrittweiten des ersten Schrittes überdeckt.

9.4. Erste Schrittweiten

In diesem Abschnitt wird eine allgemeinere Analyse der ersten beiden Schrittweiten der Rundreise auf der Kreislinie mit Richtungswechsel, nach Optimierungsproblem 7.1.1, durchgeführt, unter der Voraussetzung, dass die Anfragen jeweils auf $[0, 2\pi]$ gleichverteilt sind.

9.4.1. Erste Schrittweite

Zur einfacheren Strukturierung der Formeln gelten wieder die ergänzten Konventionen aus Kapitel 8:

- g_j ist die Anfrage die im j -ten Schritt des verwendeten Algorithmus ausgewählt wurde. Der Start der Rundreise beginnt mit der Anfrage g_0 , deren Reisepunkt mit dem Ausgangspunkt v_0 übereinstimmt.
- τ_j ist die zurückgelegte Entfernung zwischen dem Reisepunkt der Anfrage g_j und dem Reisepunkt der davor ausgewählten Anfrage g_{j-1} . Die Entfernung wird unabhängig von der Laufrichtung betrachtet, ist also immer positiv.
- $\ell_{i,j}$ ist die Distanz zwischen dem Reisepunkt v_i der Anfrage σ_i und dem Reisepunkt der Anfrage g_j .

Als Erwartungswert für die Entfernung des Reisepunktes der ersten ausgewählten Anfrage g_1 zum Startpunkt v_0 ergibt sich folgendes Integral.

$$\mathbb{E}_k [\tau_1] = \frac{k}{\pi^k} \int_0^\pi \tau_1 \int_{\tau_1}^\pi \dots \int_{\tau_1}^\pi d\tau_{1,0} d\ell_{2,0} \dots d\ell_{k,0} \quad (9.4.1)$$

Dabei wird ohne Beschränkung der Allgemeinheit σ_1 als diejenige Anfrage mit dem kleinsten Abstand zu dem Reisepunkt der Anfrage g_0 , also zum Ausgangspunkt v_0 , angenommen. Die Dichtefunktion für den Abstand der einzelnen Anfragen zu v_0 ist wegen der Gleichverteilung der Punkte einfach $\frac{1}{\pi}$ und der Vorfaktor von k ergibt sich daraus, dass insgesamt k Anfragen für die Auswahl zur Verfügung stehen. Wird nun der Ausdruck aus (9.4.1) berechnet, so ergeben sich folgende Erkenntnisse für die Dichte f von τ_1 und den Erwartungswert:

Satz 9.4.1 (Dichte und Erwartungswert des ersten Greedy-Schrittes).

Der Erwartungswert der Schrittweite zwischen dem Startpunkt und dem Reisepunkt der ersten ausgewählten Anfrage ergibt sich mit einem Informationsvorlauf k von mindestens zwei Anfragen zu:

$$\mathbb{E}_k[\tau_1] = k \int_0^\pi \tau_1 \tilde{f}_k(\tau_1) d\tau_1 = \frac{\pi}{k+1}$$

Die Dichtefunktion für diese Schrittweite berechnet sich als:

$$\tilde{f}_k(\tau_1) = k \frac{(\pi - \tau_1)^{k-1}}{\pi^k}$$

Beweis zu Satz 9.4.1.

Die Berechnungen ergeben sich dabei wie folgt:

- Die Dichte für $k = 1$ ist natürlich trivialerweise $\frac{1}{\pi}$. Bei $k = 2$ ergibt sich die Dichte als Ableitung der Wahrscheinlichkeit $P_2(\tau_1 \leq t)$ wobei diese wie folgt berechnet wird:

$$P_2(\tau_1 \leq t) = 2 \int_0^t \frac{1}{\pi} \int_{\tau_1}^\pi \frac{1}{\pi} d\ell_{2,0} d\tau_1 = 2 \int_0^t \frac{1}{\pi^2} (\pi - \tau_1)$$

Mithilfe einer Induktion über k ergibt sich schließlich:

$$\begin{aligned} P_k(\tau_1 \leq t) &= k \int_0^t \frac{1}{\pi} \int_{\tau_1}^\pi \frac{(\pi - \tau_1)^{k-2}}{\pi^{k-1}} d\ell_{2,0} d\tau_1 \\ &= k \int_0^t \frac{(\pi - \tau_1)^{k-1}}{\pi^k} d\tau_1 = \int_0^t \tilde{f}_k(\tau_1) d\tau_1 \end{aligned}$$

- Der Erwartungswert $\mathbb{E}_k[\tau_1]$ lässt sich dann mit der Dichtefunktion und einer par-

tiellen Integration ausrechnen:

$$\begin{aligned}
 \mathbb{E}_k[\tau_1] &= k \int_0^\pi \tau_1 \tilde{f}_k(\tau_1) d\tau_1 = k \int_0^\pi \tau_1 \cdot \frac{(\pi - \tau_1)^{k-1}}{\pi^k} \\
 &= \frac{k}{\pi^k} \left\{ \left[-\frac{\tau_1}{k} (\pi - \tau_1)^k \right]_0^\pi + \int_0^\pi \frac{1}{k} (\pi - \tau_1)^k \right\} \\
 &= \frac{k}{\pi^k} \left[\frac{1}{k(k+1)} (\pi - \tau_1)^{k+1} \right]_0^\pi = \frac{1}{\pi^k} \cdot \frac{1}{(k+1)} \pi^{k+1} = \frac{\pi}{k+1}
 \end{aligned}$$

Damit ist die Berechnung der ersten Schrittweite abgeschlossen. $\square$

In der Abbildung 9.4.1 ist ebenfalls wieder erkennbar, dass die empirischen Ergebnisse und die berechnete funktionale Darstellung für die erste Schrittweite relativ nahe beieinander liegen.

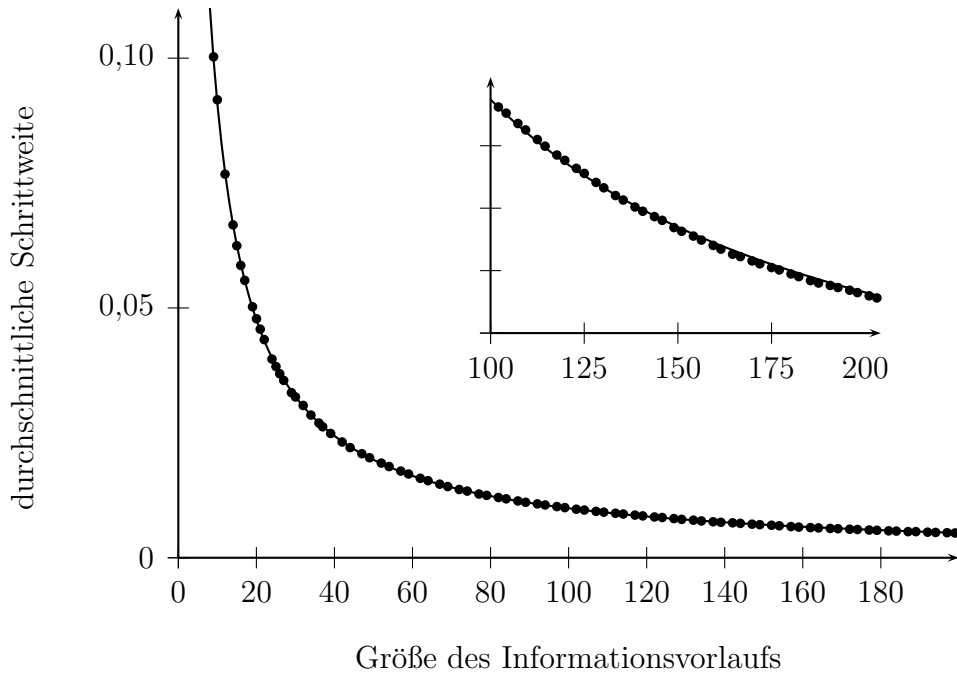


Abbildung 9.4.1.: Empirische und berechnete erwartete Schrittweite für den ersten Schritt der Rundreise auf der Kreislinie mit Richtungswechsel

Aus der Regressionsanalyse mit [Tea09, R] für das Modell $\frac{\alpha}{k+\beta}$ und den anfänglichen Werten $\alpha = 1$ sowie $\beta = 1$ ergeben sich die Ergebnisse in Tabelle 9.4.1, sowie eine quadratische Abweichung zum Regressionsmodell von $RSS = 5,05 \cdot 10^{-6}$ und ein Bestimmtheitsmaß von $R^2 = 0,999986$.

Parameter	Schätzwert	Standardfehler	t-Wert	P ($> t $)
α	0,9952498	0,0007591	1311,0	$< 2 \cdot 10^{-16}$
β	0,9687422	0,0032099	301,8	$< 2 \cdot 10^{-16}$

Tabelle 9.4.1.: Ergebnisse zur nichtlinearen Regression der ersten Schrittweite für die Rundreise auf der Kreislinie mit Richtungswechsel

9.4.2. Zweite Schrittweite

Die Berechnung der durchschnittlichen Entfernung des Reisepunktes der Anfrage g_2 zum Reisepunkt der Anfrage g_1 gestaltet sich schon etwas schwieriger. Erstens wird die neue Anfrage σ_{k+1} bekannt, und zweitens sind die alten Anfragen σ_2 bis σ_k nun leider nicht mehr wie am Anfang auf der ganzen Kreislinie gleichverteilt, sondern nur noch auf dem Abschnitt $[\tau_1, 2\pi - \tau_1]$, dabei wird, wie schon bei der Berechnung der Schrittweite für den ersten Schritt, ohne Beschränkung der Allgemeinheit angenommen, dass der Reisepunkt der ersten ausgewählten Anfrage g_1 im Bereich $[0, \pi]$ liegt.

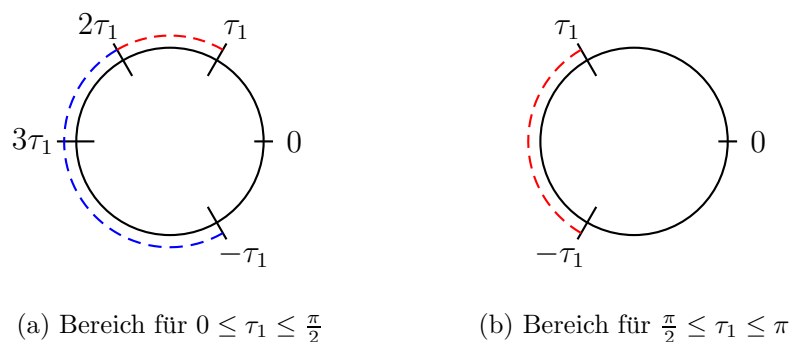


Abbildung 9.4.2.: Zulässige Bereiche für die alten Anfragen bei der Rundreise auf der Kreislinie mit Richtungswechsel

Die Dichtefunktion spaltet sich nun, je nachdem welche Wegstrecke im ersten Schritt zurückgelegt wurde, in zwei verschiedene Typen von Funktionen auf:

- (i) Bei $\tau_1 \in [0, \frac{\pi}{2}]$ liegen die Reisepunkte der alten Anfragen innerhalb des gestrichelten Bereichs aus der Abbildung 9.4.2a. Wegen des Kriteriums zur Auswahl für g_1 kann ausgehend vom neuen Standort eine Entfernung zwischen 0 und $2\tau_1$ nur in der gleicher Laufrichtung wie schon auf dem Weg zum Reisepunkt der Anfrage g_1 gefahren werden, größere Entfernungen bis zu π können jedoch in beiden Laufrichtungen erreicht werden.

Dadurch ergibt sich mit $\tau_1 \in [0, \frac{\pi}{2}]$:

- für $\ell_{i,1} \in [0, 2\tau_1]$ gilt: $f_2(\ell_{i,1}) = \frac{1}{2(\pi - \tau_1)}$ (Bereich zwischen τ_1 und $3\tau_1$)

- für $\ell_{i,1} \in [2\tau_1, \pi]$ gilt: $f_2(\ell_{i,1}) = \frac{1}{\pi - \tau_1}$ (Bereich zwischen $3\tau_1$ und $-\tau_1 = 2\pi - \tau_1$)
- für $\ell_{i,1} \notin [0, \pi]$ gilt: $f_2(\ell_{i,1}) = 0$

Eine Graphische Darstellung der einzelnen Bereiche für unterschiedliche Schrittweiten $\tau_1 \in \{0, \frac{1}{8}\pi, \frac{1}{4}\pi, \frac{3}{8}\pi\}$ findet sich in Abbildung 9.4.3. Die dünn gepunktete Linie approximiert den Verlauf der Sprungstellen in den einzelnen Dichtefunktionen.

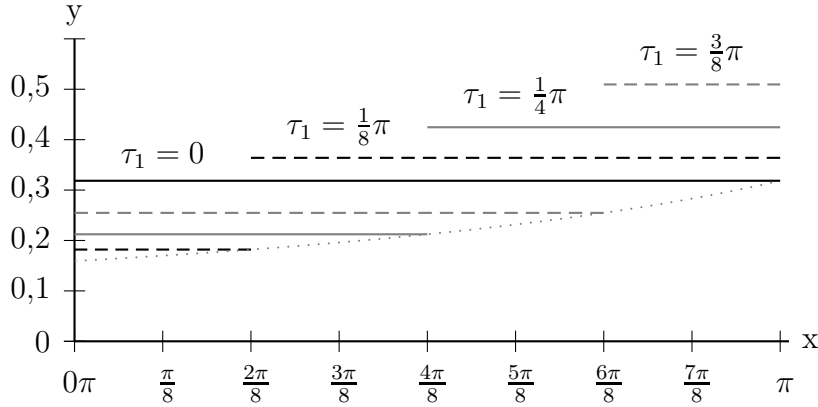


Abbildung 9.4.3.: Dichtefunktionen für die möglichen Distanzen zum Reisepunkt einer verbleibenden Anfrage bei der Rundreise auf der Kreislinie mit Richtungswechsel nach dem ersten Schritt und einer Bewegung um weniger als einen viertel Kreis

- (ii) Gilt allerdings schon $\tau_1 \geq \frac{\pi}{2}$, dann kann entgegen der Laufrichtung zum Reisepunkt der Anfrage g_1 keiner der noch möglichen Punkte aus $[\tau_1, 2(\pi - \tau_1)]$ (der in Abbildung 9.4.2b gestrichelte Bereich des Kreisbogens) noch effizient erreicht werden und die Dichtefunktion der Entfernung verliert den Bereich, welcher in beiden Richtungen verwirklicht werden kann. Es gilt dann mit $\tau_1 \geq \frac{\pi}{2}$:

$$\ell_{i,1} \in [0, 2(\pi - \tau_1)] \text{ und } f(\ell_{i,1}) = \frac{1}{2(\pi - \tau_1)}$$

Eine Veranschaulichung dieser Dichtefunktionen findet sich in der Abbildung 9.4.4.

Hierbei bleibt zu beachten, dass beide Arten der Dichtefunktionen für $\tau_1 = \frac{\pi}{2}$ die gleiche Funktion, nämlich $f_2(\ell_{i,1}) = \frac{1}{\pi}$ für $\ell_{i,1} \in [0, \pi]$ liefern.

Nach der Zusammenfassung der beiden Bereiche für τ_1 , so erhält die Dichtefunktion für die Entfernung der alten Anfragen zum Reisepunkt der ersten ausgewählten Anfrage

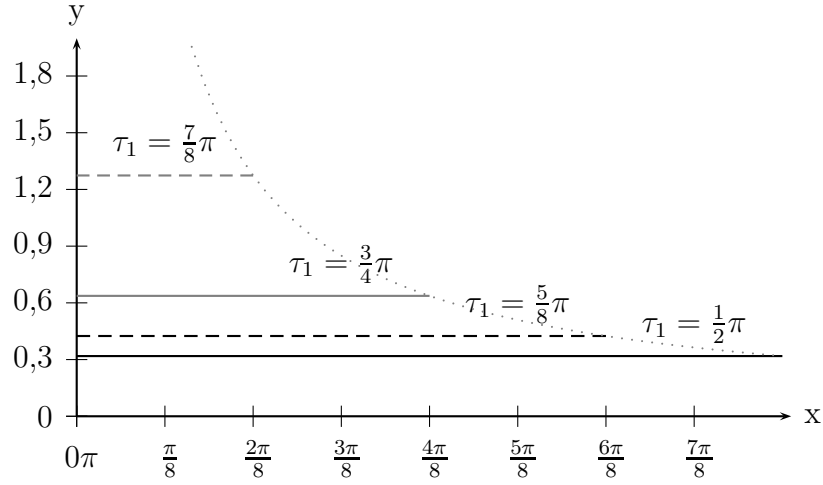


Abbildung 9.4.4.: Dichtefunktionen für die möglichen Distanzen zu einer verbleibenden Anfrage nach dem ersten Schritt und einer Bewegung um mehr als einen viertel Kreis

g_1 folgende Gestalt:

$$f_2(\ell_{i,1}) = \begin{cases} \frac{1}{2(\pi-\tau_1)} & \text{falls } 0 \leq \ell_{i,1} < \min\{2(\pi-\tau_1), 2\tau_1\} \\ \frac{1}{(\pi-\tau_1)} & \text{falls } \min\{2(\pi-\tau_1), 2\tau_1\} \leq \ell_{i,1} < \min\{\pi, 2(\pi-\tau_1)\} \\ 0 & \text{sonst} \end{cases} \quad (9.4.2)$$

Der maximal mögliche Abstand zu dem Reisepunkt von g_1 ist π , dieser kann allerdings nur erreicht werden, wenn τ_1 kleiner gleich $\frac{\pi}{2}$ ist und dann auch nur durch die Bewegung zu dem eindeutigen Punkt auf der genau gegenüberliegenden Position der Kreislinie, also dem zweiten Schnittpunkt der Kreislinie mit der Verbindungsgeraden des Reisepunktes zum Mittelpunkt des Kreises. Da dieser Abstand somit nur unter ganz speziellen Umständen – alle noch nicht bearbeiteten Anfragen müssten ihre Reisepunkte genau auf dieser einen Position haben – überhaupt möglich ist, und die Dichtefunktion dadurch relativ unhandlich werden würde, wird dieser in deren Fällen nicht explizit eingebaut. Dies ändert außerdem bei den Betrachtungen der Dichtefunktion nichts, da ein einzelner Punkt bei der kontinuierlichen Betrachtung nur eine Nullmenge darstellt und damit unerheblich ist.

Zur Auswahl für die zweite Station seiner Rundreise stehen dem Algorithmus 9.1.1 nun wieder k Anfragen zur Verfügung. Allerdings sind bei diesen k möglichen ($k-1$) der Reisepunkte nach der Dichtefunktion aus (9.4.2) verteilt und der Abstand zum Reisepunkt der neu bekannt gewordenen Anfrage σ_{k+1} ist auf $[0, \pi]$ gleichverteilt.

Je nachdem, ob die als zweite Station ausgewählte Anfrage nun aus der Menge der schon vorher bekannten Anfragen $\{\sigma_2, \dots, \sigma_k\}$ kommt, oder sogar gleich die gerade erst

bekannt gewordene Anfrage σ_{k+1} ist, ergeben sich daraus verschiedene Situationen.

Jeder Reisepunkt der Anfragen $\{\sigma_2, \dots, \sigma_k\}$ liegt, wie oben festgestellt, im Bereich $[\tau_1, 2\pi - \tau_1]$ und dieser hat als Länge des Intervalls eben genau den Wert $2(\pi - \tau_1)$ und deshalb kann im zweiten Schritt höchstens eine Entfernung von diesem Wert zum Reisepunkt der Anfrage g_1 zurückgelegt werden. Falls dabei $\tau_1 < \frac{\pi}{2}$ gilt, so ist zwar $2(\pi - \tau_1)$ größer als π aber bei einer Schrittweite von mehr als π zwischen zwei direkt aufeinander folgenden Stationen wird nicht effizient gearbeitet, da eine Fahrt in die andere Richtung eine kürzere Schrittweite ergibt. Daher ergibt sich $\omega = \min\{\pi, 2(\pi - \tau_1)\}$ als der maximal mögliche Abstand zum Reisepunkt von g_1 .

Definition 9.4.2 (Bedingter Beitrag zu einem Integral).

Als Erweiterung zu Definition 8.4.3 wird nun ein bedingter Beitrag eingeführt. Der *bedingte Beitrag des Falls z bedingt an den Voraussetzungen y zum Erwartungswert von x* ist als $\Theta[x | y : z]$ definiert. Dabei bezeichnet er den Beitrag zum Erwartungswert von x für den Fall z unter der Bedingung, dass die in y angegebenen Voraussetzungen eingetreten sind.

Der in (9.4.6) verwendete Ausdruck

$$\Theta[\tau_2 | \tau_1 : g_2 = \sigma_{k+1}, \tau_2 \leq 2\tau_1]$$

steht zum Beispiel für den Beitrag zum bedingten Erwartungswert von τ_2 bedingt daran, dass im ersten Schritt eine Schrittweite von τ_1 aufgetreten ist, für den Fall, dass im zweiten die Anfrage σ_{k+1} ausgewählt wird, sowie für die zweite Schrittweite die Oberschranke $\tau_2 \leq 2\tau_1$ eingehalten wird. Es gilt also:

$$\Theta[x | y : z] = P(z | y) \mathbb{E}[x | z, y]$$

Der eigentliche Erwartungswert $\mathbb{E}[x]$ ergibt sich dann mit der Dichte f für die Voraussetzungen y zu:

$$\mathbb{E}[x] = \int_y f(y) \sum_z \Theta[x | y : z]$$

Bei der Berechnung des Erwartungswerts für die Schrittweite im zweiten Schritt ergeben sich nun folgende zwei Fälle:

- (i) Die neue Anfrage σ_{k+1} gewinnt. Dann ergibt sich der Beitrag des Erwartungswert des Abstands τ_2 zum Reisepunkt der Anfrage g_1 für einen festen Wert von τ_1 als:

$$\Theta_k[\tau_2 | \tau_1 : g_2 = \sigma_{k+1}] = \int_0^\omega \frac{\tau_2}{\pi} \int_{\tau_2}^\omega f_2(\ell_{2,1}) \dots \int_{\tau_2}^\omega f_2(\ell_{k,1}) d\ell_{k,1} \dots d\ell_{2,1} d\tau_2 \quad (9.4.3)$$

- (ii) Eine der alten Anfragen σ_t hat den kürzesten Abstand zum Reisepunkt der Anfrage g_1 , zur besseren Gruppierung der Integrale sei σ_2 die Anfrage, welche den kürzesten

Abstand hat. Der Erwartungswert des Abstands τ_2 zum Reisepunkt von g_1 bei Auswahl der alten Anfrage und für einem festen Wert von τ_1 berechnet sich dann als:

$$\Theta_k [\tau_2 \mid \tau_1 : g_2 = \sigma_2] = \int_0^\omega \tau_2 f_2(\tau_2) \int_{\tau_2}^\omega f_2(\ell_{3,1}) \dots \int_{\tau_2}^\pi \frac{1}{\pi} d\ell_{(k+1),1} \dots d\ell_{3,1} d\tau_2 \quad (9.4.4)$$

Beide Beiträge zum Erwartungswert aus (9.4.3) und (9.4.4) hängen also entscheidend davon ab, wie groß die Schrittweite τ_1 des ersten Teilstücks der Rundreise war.

Der gesamte Erwartungswert über alle Werte von τ_1 kann durch die Gewichtung von (9.4.3) und (9.4.4) mit der Dichtefunktion aus Satz 9.4.1 an der jeweiligen Stelle τ_1 sowie der anschließenden Integration über alle möglichen Werte für τ_1 ebenfalls berechnet werden:

$$\mathbb{E}_k [\tau_2] = \int_0^\pi \tilde{f}_k(\tau_1) \sum_{t=2}^{k+1} \Theta_k [\tau_2 \mid \tau_1 : g_2 = \sigma_t] d\tau_1$$

Bevor es an die konkrete Berechnung der beiden Beiträge aus (9.4.3) und (9.4.4) geht, sollte zuerst geklärt werden, welchen Wert das gemeinsame Element der beiden Formeln, nämlich das Integral von τ_2 bis $\min \{\pi, 2(\pi - \tau_1)\}$ über die von τ_1 abhängige Dichte einer der alten Anfragen ergibt.

Korollar 9.4.3.

Das Integral über die von τ_1 abhängige Dichte zur Lage des Reisepunktes einer der alten Anfragen ergibt:

$$\int_{\tau_2}^{\min\{\pi, 2(\pi - \tau_1)\}} f_2(\ell_{i,1}) d\ell_{i,2} = \frac{1}{2(\pi - \tau_1)} (2\pi - \tau_2 - \max \{2\tau_1, \tau_2\})$$

Beweis zu Korollar 9.4.3.

Beim Beweis muss zwischen zwei Fällen für die Belegung von τ_1 unterschieden werden:

- (i) Falls dabei $\tau_1 \geq \frac{\pi}{2}$ ist, also

$$\min \{\pi, 2(\pi - \tau_1)\} = 2(\pi - \tau_1)$$

gilt, so reduziert sich die Dichte der Distanz zum Reisepunkt der schon vorher bekannten Anfrage auf $f_2 = \frac{1}{2(\pi - \tau_1)}$ und das Integral ergibt sich zu:

$$\int_{\tau_2}^{2(\pi - \tau_1)} \frac{1}{2(\pi - \tau_1)} d\ell = \frac{2\pi - 2\tau_1 - \tau_2}{2(\pi - \tau_1)}$$

(ii) Gilt jedoch $\tau_1 \leq \frac{\pi}{2}$, das bedeutet

$$\min \{ \pi, 2(\pi - \tau_1) \} = \pi,$$

dann ist die Dichtefunktion in die unterschiedlichen Bereiche $[0, 2\tau_1]$ und $[2\tau_1, \pi]$ unterteilt und das Integral ergibt:

$$\int_{\tau_2}^{\max\{\tau_2, 2\tau_1\}} \frac{1}{2(\pi - \tau_1)} d\ell + \int_{\max\{\tau_2, 2\tau_1\}}^{\pi} \frac{1}{(\pi - \tau_1)} d\ell = \frac{2\pi - \tau_2 - \max\{\tau_2, 2\tau_1\}}{2(\pi - \tau_1)}$$

Der zweite Fall muss beim ersten Integral das Maximum als obere Grenze bei der Integration ansetzen, weil sonst für große Werte von $\tau_2 > \frac{\pi}{2}$ das Integral negativ werden könnte. Da im ersten Fall $2\tau_1 \geq \pi$ und somit $\max\{\tau_2, 2\tau_1\} = 2\tau_1$ gilt, können beide Formeln zusammengefasst werden. $\square$

Satz 9.4.4 (Erwartungswert des zweiten Greedy-Schrittes).

Für die erwartete Schrittweite zwischen dem Reisepunkt der ersten bearbeitenden Anfrage g_1 und dem Reisepunkt der danach ausgewählten Anfrage g_2 unter Verwendung eines Informationsvorlaufs von $k \geq 3$ gilt:

$$\mathbb{E}_k[\tau_2] = \Theta_k[\tau_2 : g_2 = \sigma_{k+1}] + (k-1)\Theta_k[\tau_2 : g_2 = \sigma_2] = \frac{\frac{3}{2}k + 1}{(k+1)(k+2)}\pi$$

Beweis zu Satz 9.4.4.

Bei der Berechnung des erwarteten Wertes im zweiten Schritt des Algorithmus 9.1.1 muss eine Fallunterscheidung getroffen werden, ob die Anfrage, welche ausgewählt wird, aus der Menge der schon vorher bekannten Anfragen ist, oder die eben erst bekannt gewordene Anfrage σ_{k+1} ausgewählt wurde.

Bei der Berechnung des Falls mit der Auswahl einer der schon vorhandenen Anfragen ist es unerheblich welche der Anfragen aus $\sigma_2, \dots, \sigma_k$ ausgewählt wird. Daher kann die Summation der einzelnen Unterfälle durch das Multiplizieren des Faktors $(k-1)$ zu dem Beitrag für die Auswahl von σ_2 vereinfacht werden.

Erster Oberfall: Die neue Anfrage wird ausgewählt.

Wenn die neue Anfrage ausgewählt wird, so kann deren Reisepunkt nicht weiter als das Minimum aus $\{\pi, 2(\pi - \tau_1)\}$ von der ersten ausgewählten Anfrage entfernt sein. Ansonsten wäre die Entfernung zu jedem Reisepunkt der schon bekannten Anfragen kürzer als die zu dem Reisepunkt der neuen Anfrage.

Somit ergibt sich der Erwartungswert in Abhängigkeit zu der ersten Schrittweite als:

$$\begin{aligned}
 \Theta[\tau_2 \mid \tau_1 : g_2 = \sigma_{k+1}] &= \int_0^{\min\{\pi, 2(\pi-\tau_1)\}} \frac{\tau_2}{\pi} \left(\frac{2\pi - \tau_2 - \max\{2\tau_1, \tau_2\}}{2(\pi - \tau_1)} \right)^{k-1} d\tau_2 \\
 &= \int_0^{\min\{2\tau_1, 2(\pi-\tau_1)\}} \frac{\tau_2}{\pi} \left(\frac{2\pi - \tau_2 - 2\tau_1}{2(\pi - \tau_1)} \right)^{k-1} d\tau_2 + \int_{\min\{2\tau_1, 2(\pi-\tau_1)\}}^{\min\{\pi, 2(\pi-\tau_1)\}} \frac{\tau_2}{\pi} \left(\frac{\pi - \tau_2}{\pi - \tau_1} \right)^{k-1} d\tau_2
 \end{aligned} \tag{9.4.5}$$

Der erste Teil der Integralsumme aus (9.4.5) beachtet $\tau_2 \leq 2\tau_1$ und der zweite den umgekehrten Fall. Da außerdem für $\tau_1 \in [0, 2\pi]$ immer entweder τ_1 oder $2(\pi - \tau_1)$ kleiner gleich π sind, kann aus $\min\{2\tau_1, \pi, 2(\pi - \tau_1)\}$ der Ausdruck π entfernt werden.

- (i) *Im Fall* $\tau_2 \leq 2\tau_1$, also im ersten Teil der obigen Integralsumme (9.4.5) ergibt sich dann für den Beitrag zum Erwartungswert von τ_2 in Abhängigkeit zu der Belegung von τ_1 die nachfolgende Berechnung. Um die Formeln etwas handlicher zu gestalten wird $\mu := \min\{2\tau_1, 2(\pi - \tau_1)\}$ verwendet:

$$\begin{aligned}
 &\Theta[\tau_2 \mid \tau_1 : g_2 = \sigma_{k+1}, \tau_2 \leq 2\tau_1] \\
 &= \left[\frac{1}{\pi} \left(\frac{1}{2(\pi - \tau_1)} \right)^{k-1} \left(-\frac{\tau_2}{k} (2\pi - \tau_2 - 2\tau_1)^k - \frac{(2\pi - \tau_2 - 2\tau_1)^{k+1}}{k(k+1)} \right) \right]_0^\mu = \\
 &\frac{1}{\pi} \left(\frac{1}{2(\pi - \tau_1)} \right)^{k-1} \left[-\frac{\mu}{k} (2\pi - \mu - 2\tau_1)^k - \frac{(2\pi - \mu - 2\tau_1)^{k+1}}{k(k+1)} + \frac{(2\pi - 2\tau_1)^{k+1}}{k(k+1)} \right]
 \end{aligned} \tag{9.4.6}$$

Für $\tau_1 \leq \frac{\pi}{2}$ gilt somit $\mu = 2\tau_1$ und es ergibt sich mit der Dichtefunktion aus Satz 9.4.1 für (9.4.6) folgender Beitrag für den Erwartungswert von τ_2 :

$$\begin{aligned}
 &\Theta[\tau_2 : g_2 = \sigma_{k+1}, \tau_2 \leq 2\tau_1, \tau_1 \leq \frac{\pi}{2}] \\
 &= \int_0^{\frac{\pi}{2}} \frac{4}{\pi^{k+1}} \left[-\tau_1 (\pi - 2\tau_1)^k - \frac{(\pi - 2\tau_1)^{k+1}}{k+1} + \frac{(\pi - \tau_1)^{k+1}}{k+1} \right] d\tau_1 \\
 &= \frac{4\pi^{-(k+1)}}{(k+1)} \left[\frac{\tau_1 (\pi - 2\tau_1)^{k+1}}{2} + \frac{(\pi - 2\tau_1)^{k+2}}{4(k+2)} + \frac{(\pi - 2\tau_1)^{k+2}}{2(k+2)} - \frac{(\pi - \tau_1)^{k+2}}{(k+2)} \right]_0^{\frac{\pi}{2}} \\
 &= \frac{4}{(k+1)(k+2)} \left[-\frac{\pi}{2^{k+2}} - \frac{\pi}{4} - \frac{\pi}{2} + \pi \right] = \frac{2^k - 1}{2^k(k+1)(k+2)} \pi
 \end{aligned} \tag{9.4.7}$$

Ist hingegen $\tau_1 \geq \frac{\pi}{2}$ so gilt $\mu = 2(\pi - \tau_1)$ und ein großer Teil der Formel (9.4.6) für $\Theta[\tau_2 \mid \tau_1 : g_2 = \sigma_{k+1}, \tau_2 \leq 2\tau_1]$ wird zu 0. Mit Satz 9.4.1 ergibt sich

$$\begin{aligned} \Theta\left[\tau_2 : g_2 = \sigma_{k+1}, \tau_2 \leq 2\tau_1, \tau_1 \geq \frac{\pi}{2}\right] &= \int_{\frac{\pi}{2}}^{\pi} \frac{4}{\pi^{k+1}} \left[-0 - 0 + \frac{(\pi - \tau_1)}{k+1}\right] d\tau_1 \\ &= \frac{4}{\pi^{k+1}} \left[\frac{-(\pi - \tau_1)^{k+2}}{(k+1)(k+2)}\right]_{\frac{\pi}{2}}^{\pi} = \frac{1}{2^k(k+1)(k+2)}\pi. \end{aligned} \quad (9.4.8)$$

Zusammengefasst ergibt sich dadurch aus den Beiträgen (9.4.7) und (9.4.8) für die einzelnen Bereiche – unter der Bedingung, dass die neue Anfrage als zweites ausgewählt wird und die Entfernung zum ausgewählten zweiten Reisepunkt maximal die doppelte Entfernung wie beim ersten ausgewählten Reisepunkt ist – folgendes Ergebnis für die Bewegung vom Reisepunkt der ersten ausgewählten Anfrage zum Reisepunkt der zweiten ausgewählten Anfrage:

$$\Theta[\tau_2 : g_2 = \sigma_{k+1}, \tau_2 \leq 2\tau_1] = \frac{\pi}{(k+1)(k+2)} \quad (9.4.9)$$

(ii) Im Fall $\tau_2 \geq 2\tau_1$, also im zweiten Teil der Integralsumme aus (9.4.5) folgt:

$$\begin{aligned} \Theta[\tau_2 \mid \tau_1 : g_2 = \sigma_{k+1}, \tau_2 \geq 2\tau_1] &= \Theta\left[\tau_2 \mid \tau_1 : g_2 = \sigma_{k+1}, \tau_2 \geq 2\tau_1, \tau_1 \leq \frac{\pi}{2}\right] \\ &= \left[\frac{1}{\pi} \left(\frac{1}{\pi - \tau_1}\right)^{k-1} \left(-\frac{\tau_2(\pi - \tau_2)^k}{k} - \frac{(\pi - \tau_2)^{k+1}}{k(k+1)}\right)\right]_{2\tau_1}^{\pi} \\ &= \frac{1}{k\pi} \left(\frac{1}{\pi - \tau_1}\right)^{k-1} \left(2\tau_1(\pi - 2\tau_1)^k + \frac{(\pi - 2\tau_1)^{k+1}}{k+1}\right) \end{aligned} \quad (9.4.10)$$

Daher ergibt sich hier aus (9.4.10) der Beitrag zum Erwartungswert von τ_2 zu:

$$\begin{aligned} \Theta[\tau_2 : g_2 = \sigma_{k+1}, \tau_2 \geq 2\tau_1] &= \int_0^{\frac{\pi}{2}} \frac{1}{\pi^{k+1}} \left[2\tau_1(\pi - 2\tau_1)^k + \frac{(\pi - 2\tau_1)^{k+1}}{k+1}\right] d\tau_1 \\ &= \frac{1}{\pi^{k+1}} \left[-\frac{\tau_1(\pi - 2\tau_1)^{k+1}}{k+1} - \frac{(\pi - 2\tau_1)^{k+2}}{2(k+1)(k+2)} - \frac{(\pi - 2\tau_1)^{k+2}}{2(k+1)(k+2)}\right]_0^{\frac{\pi}{2}} \\ &= \frac{1}{\pi^{k+1}} \left(\frac{\pi^{k+2}}{2(k+1)(k+2)} + \frac{\pi^{k+2}}{2(k+1)(k+2)}\right) = \frac{\pi}{(k+1)(k+2)} \end{aligned} \quad (9.4.11)$$

Der Unterfall mit $\tau_1 \geq \frac{\pi}{2}$ fällt hier weg, da aus $\tau_1 \geq \frac{\pi}{2}$ sofort $\tau_2 \geq \pi$ folgen würde,

dies führt jedoch zu einem ineffizienten Laufweg zum Reisepunkt der zweiten ausgewählten Anfrage g_2 .

Zusammengefasst liefern die beiden Beiträge (9.4.9) und (9.4.11) als Beitrag für den Erwartungswert der Distanz zum Reisepunkt der zweiten ausgewählten Anfrage für den Fall, dass die zweite ausgewählte Anfrage gerade die neue Anfrage σ_{k+1} ist, folgenden Wert:

$$\Theta [\tau_2 : g_2 = \sigma_{k+1}] = \frac{2}{(k+1)(k+2)} \pi \quad (9.4.12)$$

Zweiter Oberfall: Eine der schon vorher bekannten Anfragen wird ausgewählt

Wählt der Algorithmus 9.1.1 als zweite zu bearbeitende Anfrage eine der Anfrage aus $\{\sigma_2, \dots, \sigma_k\}$, das heißt eine der schon am Anfang bekannten Anfragen, so ergibt sich folgendes Ergebnis für einen bestimmten Wert von τ_1 :

$$\begin{aligned} & \Theta [\tau_2 \mid \tau_1 : g_2 = \sigma_2] \\ &= \int_0^{\min\{\pi, 2(\pi-\tau_1)\}} \tau_2 f_2(\tau_2) \int_{\tau_2}^{\pi} \frac{1}{\pi} \left(\frac{2\pi - \tau_2 - \max\{2\tau_1, \tau_2\}}{2(\pi - \tau_1)} \right)^{k-2} d\ell_{k+1,1} d\tau_2 \\ &= \int_0^{\min\{2\tau_1, 2(\pi-\tau_1)\}} \frac{\tau_2 \pi - \tau_2^2}{\pi} \frac{(2\pi - \tau_2 - 2\tau_1)^{k-2}}{(2(\pi - \tau_1))^{k-1}} d\tau_2 + \int_{\min\{2\tau_1, 2(\pi-\tau_1)\}}^{\min\{\pi, 2(\pi-\tau_1)\}} \frac{\tau_2}{\pi} \left(\frac{\pi - \tau_2}{\pi - \tau_1} \right)^{k-1} d\tau_2 \end{aligned} \quad (9.4.13)$$

Genauso wie bei der obigen Betrachtung über den Beitrag zum Erwartungswert von τ_2 unter der Bedingung, dass die Anfrage σ_{k+1} als zweite zu bearbeitende Anfrage ausgewählt wird, muss auch hier wieder zwischen zwei Fällen unterschieden werden:

- (i) *Im Fall* $\tau_2 \leq 2\tau_1$, also dem ersten Abschnitt des Terms aus (9.4.13) ergibt sich mit $\mu = \min\{2\tau_1, 2(\pi - \tau_1)\}$ folgendes Ergebnis:

$$\begin{aligned} & \Theta [\tau_2 \mid \tau_1 : g_2 = \sigma_2, \tau_2 \leq 2\tau_1] = \\ & \left(\frac{1}{2(\pi - \tau_1)} \right)^{k-1} \frac{1}{\pi(k-1)} \left[-\pi \tau_2 (2\pi - \tau_2 - 2\tau_1)^{k-1} - \frac{\pi}{k} (2\pi - \tau_2 - 2\tau_1)^k \right. \\ & \left. + \tau_2^2 (2\pi - \tau_2 - 2\tau_1)^{k-1} + \frac{2\tau_2}{k} (2\pi - \tau_2 - 2\tau_1)^k + \frac{2(2\pi - \tau_2 - 2\tau_1)^{k+1}}{k(k+1)} \right]_0^{\mu} \end{aligned}$$

$$\begin{aligned}
 &= \left(\frac{1}{2(\pi - \tau_1)} \right)^{k-1} \frac{1}{\pi(k-1)} \left[-\pi\mu(2\pi - \mu - 2\tau_1)^{k-1} - \frac{\pi}{k}(2\pi - \mu - 2\tau_1)^k \right. \\
 &\quad + \mu^2(2\pi - \mu - 2\tau_1)^{k-1} + \frac{2\mu}{k}(2\pi - \mu - 2\tau_1)^k + \frac{2(2\pi - \mu - 2\tau_1)^{k+1}}{k(k+1)} \\
 &\quad \left. + \frac{\pi}{k}(2\pi - 2\tau_1)^k - \frac{2(2\pi - 2\tau_1)^{k+1}}{k(k+1)} \right] \quad (9.4.14)
 \end{aligned}$$

Für $\tau_1 \leq \frac{\pi}{2}$ ist $\mu = 2\tau_1$ und aus (9.4.14) ergibt sich mit der Dichte aus Satz 9.4.1 folgendes:

$$\begin{aligned}
 &\Theta \left[\tau_2 : g_2 = \sigma_2, \tau_2 \leq 2\tau_1, \tau_1 \leq \frac{\pi}{2} \right] \\
 &= \frac{1}{\pi^{k+1}(k-1)} \int_0^{\frac{\pi}{2}} (\pi - 2\tau_1)^{k-1} k (4\tau_1^2 - 2\tau_1\pi) + (\pi - 2\tau_1)^k (8\tau_1 - 2\pi) \\
 &\quad + \frac{8(\pi - 2\tau_1)^{k+1}}{(k+1)} + 2\pi(\pi - \tau_1)^k - \frac{8(\pi - \tau_1)^{k+1}}{k+1} d\tau_1 \\
 &= \frac{1}{\pi^{k+1}(k-1)} \left[4 \left(-\frac{\tau_1^2}{2}(\pi - 2\tau_1)^k - \frac{2\tau_1(\pi - 2\tau_1)^{k+1}}{4(k+1)} - \frac{(\pi - 2\tau_1)^{k+2}}{4(k+1)(k+2)} \right) \right. \\
 &\quad + 2\pi \left(\frac{\tau_1}{2}(\pi - 2\tau_1)^k + \frac{(\pi - 2\tau_1)^{k+1}}{4(k+1)} \right) + 8 \left(-\frac{\tau_1(\pi - 2\tau_1)^{k+1}}{2(k+1)} - \frac{(\pi - 2\tau_1)^{k+2}}{4(k+1)(k+2)} \right) \\
 &\quad \left. + \frac{\pi(\pi - 2\tau_1)^{k+1}}{k+1} - \frac{8(\pi - 2\tau_1)^{k+2}}{2(k+1)(k+2)} - \frac{2\pi(\pi - \tau_1)^{k+1}}{k+1} + \frac{8(\pi - \tau_1)^{k+2}}{(k+1)(k+2)} \right]_0^{\frac{\pi}{2}} = \\
 &\frac{\pi}{(k-1)(k+1)} \left[-\frac{1}{2^k} + \frac{2}{2^k(k+2)} + \frac{1}{k+2} - \frac{1}{2} + \frac{2}{k+2} - 1 + \frac{4}{k+2} + 2 - \frac{8}{k+2} \right] \\
 &= \frac{\pi(2 - (k+2))}{2^k(k-1)(k+1)(k+2)} + \frac{\pi(k+2-2)}{2(k-1)(k+1)(k+2)} \\
 &= \pi \left(\frac{-k}{2^k(k-1)(k+1)(k+2)} + \frac{2^{k-1}k}{2^k(k-1)(k+1)(k+2)} \right) \\
 &= \frac{k(2^{k-1} - 1)}{2^k(k-1)(k+1)(k+2)} \pi \quad (9.4.15)
 \end{aligned}$$

Gilt aber andererseits $\tau_1 \geq \frac{\pi}{2}$, dann ist auch $\mu = 2(\pi - \tau_1)$ und ein Großteil des Terms aus (9.4.14) vereinfacht sich zu 0. Mit Satz 9.4.1 berechnet sich der Beitrag

wie folgt:

$$\begin{aligned}
 \Theta \left[\tau_2 : g_2 = \sigma_2, \tau_2 \leq 2\tau_1, \tau_1 \geq \frac{\pi}{2} \right] &= \int_{\frac{\pi}{2}}^{\pi} \frac{1}{\pi^{k+1}} \frac{1}{2^{k-1}(k-1)} \left[\pi(2\pi - 2\tau_1)^k - \frac{2(2\pi - 2\tau_1)^{k+1}}{k+1} \right] d\tau_1 \\
 &= \frac{1}{\pi^{k+1}} \frac{1}{(k-1)(k+1)} \left[-2\pi(\pi - \tau_1)^{k+1} + 8 \frac{(\pi - \tau_1)^{k+2}}{k+2} \right]_{\frac{\pi}{2}}^{\pi} \\
 &= \frac{\pi}{(k-1)(k+1)} \left(\frac{1}{2^k} - \frac{2}{2^k(k+2)} \right) = \frac{\pi(k+2-2)}{(k-1)(k+1)2^k(k+2)} \\
 &= \frac{k}{2^k(k-1)(k+1)(k+2)} \pi \quad (9.4.16)
 \end{aligned}$$

Für $\tau_2 \leq 2\tau_1$ folgt also aus (9.4.15) und (9.4.16) der folgende Wert:

$$\Theta [\tau_2 : g_2 = \sigma_2, \tau_2 \leq 2\tau_1] = \frac{k}{2(k-1)(k+1)(k+2)} \pi \quad (9.4.17)$$

- (ii) *Im Fall $\tau_2 \geq 2\tau_1$, das ist der zweite Abschnitt des Terms aus (9.4.13), gestaltet sich das Integral wieder etwas einfacher:*

$$\begin{aligned}
 \Theta [\tau_2 | \tau_1 : g_2 = \sigma_2, \tau_2 \leq 2\tau_1] &= \left(\frac{1}{\pi - \tau_1} \right)^{k-1} \frac{1}{k\pi} \left[-\tau_2(\pi - \tau_2)^k - \frac{(\pi - \tau_2)^{k+1}}{k+1} \right]_{\min\{2\tau_1, 2(\pi - \tau_1)\}}^{\min\{\pi, 2(\pi - \tau_1)\}} \quad (9.4.18)
 \end{aligned}$$

Mit der gleichen Begründung wie schon im Fall der Auswahl der neuen Anfrage, kann hier auch wieder auf den Bereich $\tau_1 \geq \frac{\pi}{2}$ verzichtet werden. Somit gilt dann in Weiterführung von (9.4.18) mit der Dichte aus Satz 9.4.1:

$$\begin{aligned}
 \Theta [\tau_2 : g_2 = \sigma_2, \tau_2 \leq 2\tau_1] &= \int_0^{\frac{\pi}{2}} \frac{1}{\pi^{k+1}} \left[2\tau_1(\pi - 2\tau_1)^k + \frac{(\pi - 2\tau_1)^{k+1}}{k+1} \right] d\tau_1 \\
 &= \frac{1}{\pi^{k+1}} \left[-\frac{\tau_1(\pi - 2\tau_1)^{k+1}}{k+1} - \frac{(\pi - 2\tau_1)^{k+2}}{2(k+1)(k+2)} - \frac{(\pi - 2\tau_1)^{k+2}}{2(k+1)(k+2)} \right]_0^{\frac{\pi}{2}} \\
 &= \left(\frac{\pi}{2(k+1)(k+2)} + \frac{\pi}{2(k+1)(k+2)} \right) = \frac{1}{(k+1)(k+2)} \pi \quad (9.4.19)
 \end{aligned}$$

Wird bei der Auswahl der zweiten gewählten Anfrage eine der Anfragen aus $\sigma_2, \dots, \sigma_k$ ausgewählt, so ergibt sich mit (9.4.19) und (9.4.17) folgender Wert:

$$\begin{aligned} \Theta[\tau_2 : g_2 = \sigma_2] &= \frac{1}{(k+1)(k+2)}\pi + \frac{k}{2(k-1)(k+1)(k+2)}\pi \\ &= \frac{2(k-1) + k}{2(k-1)(k+1)(k+2)}\pi = \frac{3k-2}{2(k-1)(k+1)(k+2)}\pi \quad (9.4.20) \end{aligned}$$

Die beiden Beiträge (9.4.20) und (9.4.12) zusammen ergeben folgende Erkenntnis für die erwartete Schrittweite bei der Bewegung zwischen dem ersten und dem zweiten Reisepunkt des Algorithmus 9.1.1:

$$\mathbb{E}[\tau_2] = \frac{2}{(k+1)(k+2)}\pi + \frac{(k-1)(3k-2)}{2(k-1)(k+1)(k+2)}\pi = \frac{\frac{3}{2}k+1}{(k+1)(k+2)}\pi \quad \square$$

Bei der Betrachtung der Differenz der Erwartungswerte der ersten und der zweiten Schrittweite, entsteht folgendes Ergebnis:

$$\mathbb{E}[\tau_1] - \mathbb{E}[\tau_2] = \frac{1}{k+1}\pi - \frac{\frac{3}{2}k+1}{(k+1)(k+2)}\pi = \frac{k+2 - \frac{3}{2}k-1}{(k+1)(k+2)} = \frac{1 - \frac{1}{2}k}{(k+1)(k+2)}$$

Die erwartete Schrittweite im zweiten Schritt ist also für $k \geq 2$ größer als die erwartete Schrittweite im ersten Schritt. Dies ergibt sich aus der Tatsache, dass eine Bewegung in die eine Richtung auch immer Auswirkungen auf die andere Richtung hat. Wenn nämlich zur Bearbeitung der ausgewählten Anfrage eine Distanz von ℓ im Uhrzeigersinn zurückgelegt wurde, so hat anschließend bei einer Bewegung entgegen dem Uhrzeigersinn von der neuen Position jede der nach diesem Schritt verbleibenden alten Anfragen eine Mindestentfernung von 2ℓ .

Wie in den anderen Abbildungen zuvor ist auch in Abbildung 9.4.5 für die zweite Schrittweite kein besonders großer Unterschied zwischen der berechneten Kurve und den empirischen Punkten erkennbar.

Aus der Regressionsanalyse durch [Tea09, R] mit dem Modell $\frac{\alpha k+1}{(k^2+\beta_1 k+\beta_2)}$, sowie den Startwerten $\beta_1 = 3$ und $\beta_2 = 2$ sowie $\alpha = 1,5$ ergibt sich Tabelle 9.4.2. Die Summe der quadratischen Abweichungen zwischen den empirischen Schrittweiten und den Werten der gefundenen Ausprägung des Modells berechnet sich zu $RSS = 7,21 \cdot 10^{-6}$ und damit liegt das Bestimmtheitsmaß bei $R^2 = 0,9999854$.

9.5. Überlegungen zur allgemeinen Schrittweite

Sei eine Rundreise gegeben für die bis einschließlich zum Schritt $t-1$ schon für alle Schritte Anfragen ausgewählt worden sind. Dann steht nach der Bearbeitung der Anfrage g_{t-1} im Schritt $t-1$ die Auswahl der Anfrage g_t im Schritt t an.

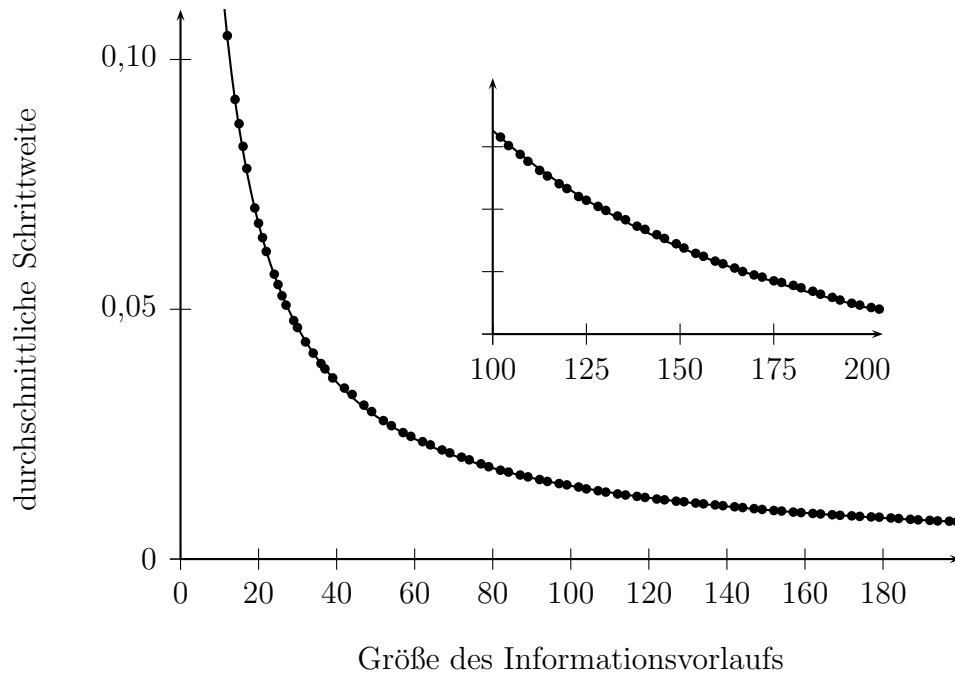


Abbildung 9.4.5.: Empirische und berechnete erwartete Schrittweite für den zweiten Schritt bei der Rundreise auf der Kreislinie mit Richtungswechsel

Dazu sei σ_{j_i} eine der möglichen Anfragen aus dem Informationsvorlauf für die Auswahl im Schritt $t + 1$, die sich daraus ergebende Situation gestaltet sich dann wie folgt:

- $\{\sigma_{j_1}, \dots, \sigma_{j_k}\} = (\{\sigma_1, \dots, \sigma_{t+k-2}, \sigma_{t+k-1}\}) \setminus \{g_1, \dots, g_{t-1}\}$. Dabei seien $\{\nu_{j_1}, \dots, \nu_{j_k}\}$ als die zugehörigen möglichen Wegstrecken vom Reisepunkt zu g_{t-1} zu den Reisepunkten der Anfragen $\sigma_{j_1}, \dots, \sigma_{j_k}$ definiert.
- $\{g_1, \dots, g_{t-1}\}$ ist eine zulässige Sequenz zur Bearbeitung der bisherigen Anfragen $\{\sigma_1, \dots, \sigma_{t+k-2}\}$ bei einer Bearbeitungsinformation von k Anfragen.

Um bei der Auswahl von g_t zur Verfügung stehen zu können, muss eine Anfrage σ_j einerseits innerhalb der Sequenz der Anfragen σ_1 bis σ_{t+k-1} enthalten sein, und darf an-

Parameter	Schätzwert	Standardfehler	t -Wert	$P(> t)$
α	1,502297	0,001839	816,73	$< 2 \cdot 10^{-16}$
β_1	2,978553	0,016202	183,84	$< 2 \cdot 10^{-16}$
β_2	2,032471	0,027492	73,93	$< 2 \cdot 10^{-16}$

Tabelle 9.4.2.: Ergebnisse zur nichtlinearen Regression der zweiten Schrittweite für die Rundreise auf der Kreislinie mit Richtungswechsel

dererseits noch nicht als eine der Anfragen $g_1, \dots, g_{t-1}$ ausgewählt worden sein. Damit der Reisepunkt der Anfrage σ_j bis zur Wahl der Anfrage g_t noch nicht als Station eingeplant wurde, muss dieser zu den Reisepunkten der Anfragen $g_{\{j-k\}_++1}, \dots, g_{t-1}$ eine größere Distanz besitzen als deren direkte Nachfolger.

Für diesen Zweck wird die Funktion $Z_k(j, t)$ definiert, welche für eine Anfrage σ_j den Bereich berechnet, in welchem deren Reisepunkt noch liegen kann, wenn diese Anfrage zum Zeitpunkt der Auswahl der Station $t + 1$ während der Durchführung des Algorithmus 9.1.1 noch nicht ausgewählt wurde. Aus den vorherigen Überlegungen ergibt sich

$$Z_k(j, t) = \{p \mid \text{dist}(p - g_{i-1}) \geq \text{dist}(g_i - g_{i-1}) \quad \forall i = \{j - k\}_+ + 1, \dots, t\}. \quad (9.5.1)$$

Natürlich gilt für σ_{t+k-1} , also für die neue Anfrage, bei der Auswahl in Schritt t für den möglichen Bereich $Z_k(t + k - 1, t - 1) = [0, 2\pi]$ dieser reduziert sich anschließend nach Schritt t zu

$$Z_k(t + k - 1, t) = [0, 2\pi] \setminus \{p \mid \text{dist}(p, g_{t-1}) < \text{dist}(g_t, g_{t-1})\}.$$

Dabei gilt weiterhin $Z_k(j_\ell, t) \subseteq Z_k(j_{\ell+1}, t)$ für die Bereiche der Reisepunkte zweier nach Schritt t noch nicht bearbeiteter Anfragen σ_{j_ℓ} und $\sigma_{j_{\ell+1}}$, wobei σ_{j_ℓ} in der Sequenz der Anfragen vor $\sigma_{j_{\ell+1}}$ liegt. Die Dichtefunktion der Reisepunkte v_j der einzelnen unbearbeiteten Anfragen ist bestimmt durch:

$$f_k(\sigma_j) = \begin{cases} |Z_k(j, t - 1)|^{-1} & \text{für } v_j \in Z_k(j, t - 1) \\ 0 & \text{sonst} \end{cases}$$

Dabei müssen die in Abbildung 9.5.1 aufgezeigten Besonderheiten beachtet werden.

Die Abbildung 9.5.1a zeigt die Bewegung zur ersten ausgewählten Anfrage und kennzeichnet gleichzeitig mit dem gestrichelten Bereich noch mögliche Positionen für die Reisepunkte der im Informationsvorlauf verbleibenden Anfragen.

In Abbildung 9.5.1b wird die neu hinzugekommene Anfrage in umgekehrter Richtung direkt angefahren und der mögliche Bereich (wieder gestrichelt) für die Reisepunkte der immer noch verbliebenen Anfragen hat sich noch weiter verkleinert.

Abbildung 9.5.1c schließlich zeigt die danach entstehende Problematik bei der Betrachtung der Bewegung zum Reisepunkt einer der unbearbeiteten Anfragen auf. Aus dem vorherigen Teilbild ist bekannt, dass die Reisepunkte der verbliebenen Anfragen auf den dort gestrichelten und hier durchgezogenen Bereich der Kreislinie liegen müssen.

Das allerdings führt nun dazu, dass es einen Bereich der Schrittweite gibt, bei welchem mit einer Bewegung ausgehend von der aktuellen Position, dargestellt durch den Punkt auf der Kreislinie, keine der noch möglichen Positionen der Reisepunkte der verbliebenen Anfragen erreicht wird. An diesen ersten Bereich der Schrittweite schließt sich ein weiterer Bereich an, bei welchem zwar eine Bewegung gegen den Uhrzeigersinn

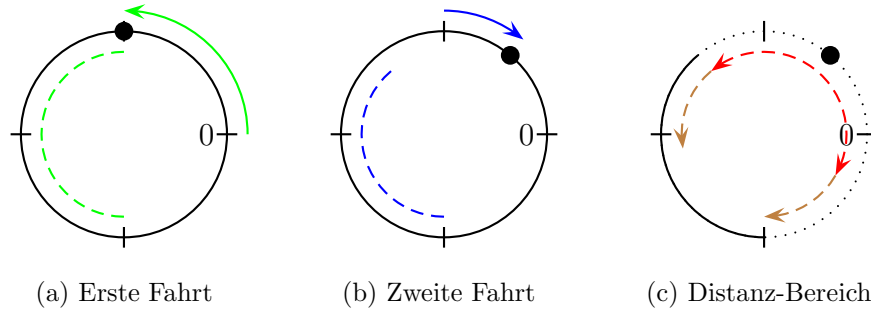


Abbildung 9.5.1.: Distanz-Bereiche zu verbleibenden Anfragen bei der Rundreise auf der Kreislinie mit Richtungswechsel

in den durchgezogenen Bereich der Kreislinie führt, allerdings bei einer Bewegung im Uhrzeigersinn dieser Bereich noch nicht erreicht wird. Abschließend besteht auch noch die Möglichkeit, dass es Schrittweiten gibt mit denen in beiden Richtungen der noch mögliche Bereich der Reisepunkte für die verbliebenen unbearbeiteten Anfragen erreicht werden kann.

Allgemein ergibt sich bei der Berechnung des Erwartungswertes für die Schrittweite τ_t während der Durchführung des Algorithmus 9.1.1 beim Übergang zwischen g_{t-1} und g_t mit Informationsvorlauf k bei $g_t = \sigma_{j_1}$ das folgende Integral über die im Schritt t verfügbaren Anfragen $\sigma_{j_1}, \dots, \sigma_{j_k}$:

$$\int_{Z_k(j_1, t-1)} \tau_t f_k(v_{j_1}) \int_{Z_k(j_2, t)} f_k(v_{j_2}) \cdots \int_{Z_k(j_k, t)} f_k(v_{j_k}) dv_{j_k} \dots dv_{j_1} \quad (9.5.2)$$

Dabei ergibt sich dann der gesamte Erwartungswert zum Zeitpunkt t mit der Einbeziehung der Integrale über die schon vorher ausgewählten Anfragen als:

$$\mathbb{E}_k[\tau_t] = \sum_{W \in \Omega} \Theta[\tau_t : (g_1, \dots, g_t) = W]$$

mit Ω als Menge aller zulässigen Rundreisen mit Länge t für $(\sigma_1, \dots, \sigma_{k+t-1})$ und Informationsvorlauf k .

Hierbei muss allerdings beachtet werden, dass sich zwischen zwei Anfragen die Laufrichtung im Vergleich zum vorherigen Schritt ändern kann. Daher sind theoretisch bis zu $2^t k^t$ Bearbeitungsreihenfolgen mit verschiedenen Richtungswechseln möglich.

Nach einer Änderung der Sichtweise, von dem zu der Position des Reisepunktes der zuletzt bearbeiteten Anfrage relativ unabhängigen Bereichs $Z_k(i, t)$ der Kreislinie, hin zu der schon in Kapitel 8 verwendeten Messung des Abstands zur Position der zuletzt bearbeiteten Anfrage, so lässt sich der Bereich $Z_k(i, t)$ durch ein Intervall $[u_i, o_i]$ darstellen. Dabei ist u_i die minimale Schrittweite zu der Anfrage σ_i in Richtung der vorher erfolgten Bewegung und o_i analog dazu die maximale Schrittweite in dieser Laufrichtung.

Somit ergeben sich für die k Anfragen $\sigma_{j_1}, \dots, \sigma_{j_k}$ im Informationsvorlauf, wobei diese wiederum nach ihrer Reihenfolge in der Sequenz der Anfragen sortiert seien, folgende Eigenschaften für die Bereiche ihrer Reisepunkte:

$$2\pi \geq u_{j_1} \geq \dots \geq u_{j_k} = 0 \quad \text{und} \quad 0 \leq o_{j_1} \leq \dots \leq o_{j_k} = 2\pi$$

Diese Werte lassen sich über $2\pi - o_i$ als minimale Distanz und $2\pi - u_i$ als maximale Distanz auch für die Berechnung zur Bewegung in die andere Richtung nutzen. Die Dichtefunktion ergibt sich dabei in beiden Fällen jeweils als $\frac{1}{o_i - u_i}$ auf dem noch zulässigen Bereich dieser Anfrage.

Der relevante Bereich zur Integration bei der einzelnen Anfrage ergibt sich dann nach der Bewegung um die Schrittweite τ mit den neuen Begrenzungen u'_i und o'_i zu $[u'_i, o'_i]$. Da es bei der Rundreise auf der Kreislinie mit Richtungswechsel passieren kann, dass die Position des Reisepunktes der letzten bearbeiteten Anfrage nicht an den Bereich des Reisepunktes der noch nicht bearbeiteten Anfrage angrenzt (dann gilt $u_i > 0$) und somit von dieser Position mit der ausgewählten Laufrichtung erst eine Distanz von u_i zurückgelegt werden muss um an diesen Bereich anzuschließen, kann auch wiederum $u'_i > 0$ gelten.

Als maximal mögliche Schrittweite zum Reisepunkt der nächsten ausgewählten Anfrage ergibt sich bei dieser Betrachtungsweise mit einer Bewegung in der vorherigen Laufrichtung eine Schrittweite von $\min\{o_1, \pi\}$. Das Integral aus (9.5.2), mit Auswahl von σ_i als nächste zu bearbeitende Anfrage, lässt sich daher umschreiben zu:

$$\begin{aligned} & \int_{\min\{u_i, \pi\}}^{\min\{o_{j_1}, \pi\}} \frac{\tau}{o_i - u_i} \prod_{j_\ell \neq i} \left(\int_{u'_{j_\ell}}^{o_{j_\ell}} \frac{1}{o_{j_\ell} - u_{j_\ell}} dt \right) d\tau \\ &= \prod_{\ell=1}^k \frac{1}{o_{j_\ell} - u_{j_\ell}} \int_{\min\{u_i, \pi\}}^{\min\{o_{j_1}, \pi\}} \tau \prod_{j_\ell \neq i} (o'_{j_\ell} - u'_{j_\ell}) d\tau \quad (9.5.3) \end{aligned}$$

Bei einer Bewegung in die andere Richtung müssten die einzelnen Werte des Bereichs wie oben erläutert umgerechnet werden. Da im daran anschließendem Schritt die Dichten der verbleibenden Anfragen unabhängig von der Laufrichtung jeweils über

$$\frac{1}{o'_{j_\ell} - u'_{j_\ell}}$$

definiert sein werden, kann also bei der Rundreise auf der Kreislinie mit Richtungswechsel dieser Anteil wie auch schon in (8.5.3) gekürzt werden.

Beim Übergang nach der Bearbeitung der aktuellen ausgewählten Anfrage g_t und der Bekanntgabe der neuen Anfrage σ_{t+k} , verändern sich die Grenzen des noch möglichen

Bereichs für den Reisepunkt einer in diesem Schritt noch nicht bearbeiteten Anfrage σ_i bei einer Schrittweite von τ_t zu:

- (i) $u'_i = \max\{0, u_i - \tau_t\}$ und $o'_i = \min\{o_i - \tau_t, 2\pi - 2\tau_t\}$ falls die Bewegung in der gleichen Richtung wie davor verläuft.
- (ii) $u'_i = \max\{0, 2\pi - o_i - \tau_t\}$ und $o'_i = \min\{2\pi - u_i - \tau_t, 2\pi - 2\tau_t\}$ falls die Bewegung in gegensätzlicher Richtung zur vorherigen verläuft.

Das Maximum bei der Berechnung der neuen Werte u'_i zur unteren Grenze des Intervalls wird benötigt, weil für $u_i > 0$ einerseits bei einer kleineren Schrittweite τ_t als u_i der Bereich des Reisepunktes der Anfrage gar nicht erreicht wird und somit ein Abstand von $u_i - \tau_t$ zum aktuellen Bereich verbleibt. Andererseits wird bei einer größeren Schrittweite das Intervall sicher erreicht und somit ist die Differenz $u_i - \tau_t$ negativ, eine daraus resultierende unsinnige negative Entfernung wird durch die Einbeziehung der Null in das Maximum verhindert.

Für die obere Grenze des Intervalls o_i muss bei der neuen Berechnung bedacht werden, dass durch die Möglichkeit des Wechsels der Richtung, eine Bewegung in die eine Richtung nur dann durchgeführt wird, wenn auch in der anderen Richtung kein näheres Ziel vorhanden ist. Die Bewegung um τ_t in der Laufrichtung führt zu einer Verkürzung der maximalen Entfernung auf $o_i - \tau_t$, gleichzeitig ergibt sich aber auch, dass der Reisepunkt der Anfrage ebenfalls entgegen der Laufrichtung einen minimalen Abstand von τ_t zum Ausgangspunkt haben muss. Daher ist dann vom neuen Ausgangspunkt keine größere Entfernung als $2\pi - 2\tau_t$ mehr möglich.

Die Rekursionsformel aus (8.5.21) mit (8.5.20) lässt sich auch wieder beinahe analog erzeugen, allerdings muss diesmal beim Rekursionsschritt eine Unterscheidung in beide Laufrichtungen getroffen werden. Diese ergibt sich dann zu:

$$\begin{aligned}
 & f_t((u_1, o_1), \dots, (u_{k-1}, o_{k-1})) \\
 &= \frac{1}{2\pi} \sum_{j=1}^k \int_{\min\{u_j, \pi\}}^{\min\{\pi, o_1\}} f_{t-1}(\phi(j, (u_1, o_1), \dots, (u_{k-1}, o_{k-1}))) d\tau_t \\
 &+ \frac{1}{2\pi} \sum_{j=1}^k \int_{\min\{2\pi - o_j, \pi\}}^{\min\{\pi, 2\pi - u_1\}} f_{t-1}(\phi(j, (2\pi - o_1, 2\pi - u_1), \dots, (2\pi - o_{k-1}, 2\pi - u_{k-1}))) d\tau_t
 \end{aligned} \tag{9.5.4}$$

Dabei wird die Funktion zur Umwandlung

$$\phi(j, (u_1, o_1), \dots, (u_{k-1}, o_{k-1})) = (u'_1, o'_1), \dots, (u'_{k-1}, o'_{k-1})$$

ähnlich zu (8.5.22) mit $o_k = 2\pi$ und $u_k = 0$ wie folgt definiert:

$$u'_i = \begin{cases} \max \{0, u_i - \tau_t\} & \text{für } i < j \\ \max \{0, u_{i+1} - \tau_t\} & \text{für } i \geq j \end{cases} \quad (9.5.5a)$$

$$o'_i = \begin{cases} \min \{o_i - \tau_t, 2\pi - 2\tau_t\} & \text{für } i < j \\ \min \{o_{i+1} - \tau_t, 2\pi - 2\tau_t\} & \text{für } i \geq j \end{cases} \quad (9.5.5b)$$

Die Rekursionsfolge wird schließlich über

$$\begin{aligned} f_0((u_1, o_1), \dots, (u_{k-1}, o_{k-1})) \\ = \left(\frac{1}{2\pi}\right)^{k-1} \tau_1 \prod_{i=1}^{k-1} \left(\int_{u_i}^{o_i} 1 \, dy \right) = \left(\frac{1}{2\pi}\right)^{k-1} \tau_1 \prod_{i=1}^{k-1} (o_i - u_i) \end{aligned} \quad (9.5.6)$$

gestartet.

Beispielrechnung

Dass diese eigentlich übersichtliche Rekursionsformel einiges an Tücken in sich hat, sei hier am Beispiel eines Informationsvorlaufs von zwei Anfragen und der allgemeinen Berechnung von $f_1((u, o))$ und einer Hinführung zu $f_2((u, o))$ aufgezeigt. Zur Vereinfachung der Bezeichnungen sei die alte noch unbearbeitete Anfrage auf den Bereich $[u, o]$ für ihren Reisepunkt beschränkt. Die neu bekannt gewordene Anfrage ist mit dem Bereich $[0, 2\pi]$ für den Reisepunkt definiert.

Die Berechnung von $f_1((u, o))$ gestaltet sich dann nach (9.5.4) zu:

$$\begin{aligned} f_1((u, o)) &= \frac{1}{2\pi} \int_{\min\{u, \pi\}}^{\min\{\pi, o\}} f_0(\phi(1, (0, 2\pi))) \, d\tau_1 + \frac{1}{2\pi} \int_0^{\min\{\pi, o\}} f_0(\phi(2, (u, o))) \, d\tau_1 \\ &+ \frac{1}{2\pi} \int_{\min\{2\pi-o, \pi\}}^{\min\{\pi, 2\pi-u\}} f_0(\phi(1, (0, 2\pi))) \, d\tau_1 + \frac{1}{2\pi} \int_0^{\min\{\pi, 2\pi-u\}} f_0(\phi(2, (2\pi-o, 2\pi-u))) \, d\tau_1 \end{aligned} \quad (9.5.7)$$

Bei der Auswertung der Auswahl der neuen Anfrage und ohne einen Richtungswechsel

ergibt sich:

$$\begin{aligned}
 & \frac{1}{2\pi} \int_0^{\min\{\pi, o\}} f_0(\phi(2, (u, o))) dx_1 \\
 &= \frac{1}{2\pi} \int_0^{\min\{\pi, o\}} f_0((\max\{u - \tau_1, 0\}, \min\{o - \tau_1, 2\pi - 2\tau_1\})) d\tau_1 \\
 &= \left(\frac{1}{2\pi}\right)^2 \int_0^{\min\{\pi, o\}} \tau_1 (\min\{o - \tau_1, 2\pi - 2\tau_1\} - \max\{u - \tau_1, 0\}) d\tau_1 \\
 &= \left(\frac{1}{2\pi}\right)^2 \left(\int_0^{\min\{2\pi-o, o\}} \tau_1 (o - \tau_1) d\tau_1 + \int_{\min\{2\pi-o, o\}}^{\min\{\pi, o\}} \tau_1 (2\pi - 2\tau_1) d\tau_1 + \int_0^{\min\{\pi, u\}} \tau_1 (u - \tau_1) d\tau_1 \right) \\
 &= \left(\frac{1}{2\pi}\right)^2 \left(\frac{1}{2} o \min\{o, 2\pi - o\}^2 - \frac{1}{3} \min\{o, 2\pi - o\}^3 \right. \\
 &\quad + \pi \min\{\pi, o\}^2 - \pi \min\{2\pi - o, o\}^2 - \frac{2}{3} \min\{\pi, o\}^3 + \frac{2}{3} \min\{2\pi - o, o\}^3 \\
 &\quad \left. + \frac{1}{2} u \min\{u, \pi\}^2 - \frac{1}{3} \min\{u, \pi\}^3 \right) \quad (9.5.8)
 \end{aligned}$$

Entgegen der vorherigen Laufrichtung, also bei einem Richtungswechsel, liefert die Auswahl der neuen Anfrage analog:

$$\begin{aligned}
 & \frac{1}{2\pi} \int_0^{\min\{\pi, 2\pi-u\}} f_0(\phi(2, (2\pi - o, 2\pi - u))) d\tau_1 \\
 &= \left(\frac{1}{2\pi}\right)^2 \left(\frac{1}{2} (2\pi - u) \min\{2\pi - u, u\}^2 - \frac{1}{3} \min\{2\pi - u, u\}^3 + \pi \min\{\pi, 2\pi - u\}^2 \right. \\
 &\quad - \pi \min\{u, 2\pi - u\}^2 - \frac{2}{3} \min\{\pi, 2\pi - u\}^3 + \frac{2}{3} \min\{u, 2\pi - u\}^3 \\
 &\quad \left. + \frac{1}{2} (2\pi - o) \min\{2\pi - o, \pi\}^2 - \frac{1}{3} \min\{2\pi - o, \pi\}^3 \right) \quad (9.5.9)
 \end{aligned}$$

Wenn nicht die neue Anfrage ausgewählt wird, sondern die schon vorher bekannte alte, bisher noch nicht bearbeitete, Anfrage, dann ergibt sich bei einer Bewegung in der

vorherigen Laufrichtung

$$\begin{aligned} \frac{1}{2\pi} \int_{\min\{u,\pi\}}^{\min\{\pi,o\}} f_0(\phi(1, (0, 2\pi))) d\tau_1 &= \left(\frac{1}{2\pi}\right)^2 \int_{\min\{u,\pi\}}^{\min\{\pi,o\}} \tau_1(2\pi - 2\tau_1) d\tau_1 \\ &= \left(\frac{1}{2\pi}\right)^2 \left(\pi \min\{\pi, o\}^2 - \pi \min\{\pi, u\}^2 - \frac{2}{3} \min\{\pi, o\}^3 + \frac{2}{3} \min\{\pi, u\}^3 \right) \end{aligned} \quad (9.5.10)$$

und entgegen der vorherigen Laufrichtung

$$\begin{aligned} \frac{1}{2\pi} \int_{\min\{2\pi-o,\pi\}}^{\min\{\pi, 2\pi-u\}} f_0(\phi(1, (0, 2\pi))) d\tau_1 &= \left(\frac{1}{2\pi}\right)^2 \left(\pi \min\{\pi, 2\pi - u\}^2 - \pi \min\{\pi, 2\pi - o\}^2 \right. \\ &\quad \left. - \frac{2}{3} \min\{\pi, 2\pi - u\}^3 + \frac{2}{3} \min\{\pi, 2\pi - o\}^3 \right). \end{aligned} \quad (9.5.11)$$

Aus den vier einzelnen Teilen (9.5.8), (9.5.9), (9.5.10) und (9.5.11) ergibt sich dann zusammengefasst folgende Darstellung:

$$\begin{aligned} f_1((u, o)) &= \left(\frac{1}{2\pi}\right)^2 \left(-\frac{1}{2}(2\pi - u) \min\{\pi, u\}^2 + \frac{1}{3} \min\{\pi, u\}^3 \right. \\ &\quad + 2\pi \min\{\pi, 2\pi - u\}^2 - \frac{4}{3} \min\{\pi, 2\pi - u\}^3 - \frac{1}{2}u \min\{u, 2\pi - u\}^2 \\ &\quad + \frac{1}{3} \min\{u, 2\pi - u\}^3 - \frac{1}{2}o \min\{\pi, 2\pi - o\}^2 + \frac{1}{3} \min\{\pi, 2\pi - o\}^3 + 2\pi \min\{o, \pi\}^2 \\ &\quad \left. - \frac{4}{3} \min\{o, \pi\}^3 - \frac{1}{2}(2\pi - o) \min\{o, 2\pi - o\}^2 + \frac{1}{3} \min\{o, 2\pi - o\}^3 \right) \end{aligned} \quad (9.5.12)$$

Mit der Auswertung von $f_1((0, 2\pi))$ über (9.5.12) ergibt sich als erwartete erste Schrittweite bei zwei Anfragen:

$$f_1((0, 2\pi)) = \frac{-0 + 0 + 2\pi^3 - \frac{4}{3}\pi^3 - 0 + 0 - 0 + 0 + 2\pi^3 - \frac{4}{3}\pi^3 - 0 + 0}{4\pi^2} = \frac{1}{3}\pi$$

Für die allgemeine Berechnung von $f_2((u, o))$ werden nun die beiden möglichen Bereiche des Reisepunktes der nach dem Schritt verbleibenden Anfrage in (9.5.12) eingesetzt. Dabei werden der eigentlich zur Einordnung der Schrittweite τ benötigte Index, sowie der Vorfaktor $\left(\frac{1}{2\pi}\right)^3$ zur platzsparenderen Darstellung der Formel weggelassen.

- Bei einer Bewegung in der vorherigen Laufrichtung ergibt sich bei der Auswahl der

älteren Anfrage zur Bearbeitung mit

$$[0, 2\pi - 2\tau]$$

als den Bereich für den Reisepunkt der verbleibenden, gerade bekanntgegeben Anfrage aus (9.5.12) die Darstellung:

$$\begin{aligned} & \int_{\min\{u, \pi\}}^{\min\{\pi, o\}} \left(2\pi^3 - \frac{4}{3}\pi^3 - \frac{2\pi - 2\tau}{2} \min\{\pi, 2\tau\}^2 + \frac{1}{3} \min\{\pi, 2\tau\}^3 + 2\pi \min\{2\pi - 2\tau, \pi\}^2 \right. \\ & \quad \left. - \frac{4}{3} \min\{2\pi - 2\tau, \pi\}^3 - \tau \min\{2\pi - 2\tau, 2\tau\}^2 + \frac{1}{3} \min\{2\pi - 2\tau, 2\tau\}^3 \right) d\tau \end{aligned} \quad (9.5.13)$$

- Falls nicht die ältere, sondern die neu bekannt gegebene Anfrage ausgewählt wird, so kommt mit dem verbleibenden Bereich

$$[\max\{0, u - \tau\}, \min\{o - \tau, 2\pi - 2\tau\}]$$

für den Reisepunkt der älteren Anfrage in (9.5.12) eine zusätzliche Schwierigkeit, nämlich die Integration über ein Minimum angewandt auf ein Maximum, dazu.

$$\begin{aligned} & \int_0^{\min\{\pi, o\}} \left(-\frac{2\pi - \max\{0, u - \tau\}}{2} \min\{\pi, \max\{0, u - \tau\}\}^2 + \frac{\min\{\pi, \max\{0, u - \tau\}\}^3}{3} \right. \\ & \quad + 2\pi \min\{\pi, 2\pi - (u - \tau)\}^2 - \frac{4}{3} \min\{\pi, 2\pi - (u - \tau)\}^3 \\ & \quad - \frac{1}{2} \max\{0, u - \tau\} \min\{\max\{0, u - \tau\}, 2\pi - (u - \tau)\}^2 \\ & \quad + \frac{1}{3} \min\{\max\{0, u - \tau\}, 2\pi - (u - \tau)\}^3 \\ & \quad - \frac{1}{2} \min\{o - \tau, 2\pi - 2\tau\} \min\{2\pi - \min\{o - \tau, 2\pi - 2\tau\}, \pi\}^2 \\ & \quad + \frac{1}{3} \min\{2\pi - \min\{o - \tau, 2\pi - 2\tau\}, \pi\}^3 \\ & \quad + 2\pi \min\{o - \tau, 2\pi - 2\tau, \pi\}^2 - \frac{4}{3} \min\{o - \tau, 2\pi - 2\tau, \pi\}^3 - \\ & \quad \left. \frac{2\pi - \min\{o - \tau, 2\pi - 2\tau\}}{2} \min\{2\pi - \min\{o - \tau, 2\pi - 2\tau\}, \min\{o - \tau, 2\pi - 2\tau\}\}^2 \right. \\ & \quad \left. + \frac{1}{3} \min\{2\pi - \min\{o - \tau, 2\pi - 2\tau\}, \min\{o - \tau, 2\pi - 2\tau\}\}^3 \right) d\tau \end{aligned} \quad (9.5.14)$$

Dabei war es möglich unter Anwendung von

$$\min \{x, 2\pi - \max \{0, u - \tau\}\} = \min \{x, 2\pi - (u - \tau)\}$$

für $x \leq 2\pi$ einige der Terme aus (9.5.14) etwas zu vereinfachen.

Beiden Formeln (9.5.13) und (9.5.14) ist nun ein relativ langer Grundterm mit verschiedenen Minima gemeinsam, wobei auf diese Formeln nach ihrer Berechnung dann auch noch die Umwandlung der Grenzen für die andere Laufrichtung angewandt werden müsste. Zusammen ergäbe sich somit eine Summe aus vier unterschiedlich aufgebauten Termen, welche auch noch verschiedene Extrema in sich vereinigen würden.

Die vereinzelte Schachtelung der Extremwertbestimmungen führt nun dazu, dass bei einer weiteren Berechnung ausgehend von (9.5.14) das Auflösen der einzelnen Extremwertfunktionen nicht mehr ganz so einfach über eine Aufteilung des Bereichs der Integration durchgeführt werden kann. Zwar lassen sich durchaus gewisse Ähnlichkeiten erkennen, und es ergeben sich die folgenden zwei Aufteilungen erstellen, was sich aber insgesamt nicht als wirklich hilfreich für eine weitere Berechnung herausstellt:

(i) Einerseits eine Unterscheidung zwischen der Schrittweite τ und der Länge des minimalen Abstands u nach

- $0 \leq \tau \leq \min \{\pi, u\}$ mit den Ergebnissen:

$$\begin{aligned} \max \{0, u - \tau\} &= u - \tau \\ \min \{\max \{0, u - \tau\}, 2\pi - (u - \tau)\} &= \min \{u - \tau, 2\pi - (u - \tau)\} \\ \min \{\pi, 2\pi - (u - \tau)\} & \\ \min \{\pi, \max \{0, u - \tau\}\} &= \min \{\pi, u - \tau\} \end{aligned}$$

Damit muss hier noch einmal unterschieden werden, ob τ kleiner oder größer gleich $u - \pi$ ist.

- $\min \{\pi, u\} \leq \tau \leq \min \{\pi, o\}$, – dabei muss $u \leq \pi$ gelten, ansonsten ist der dadurch definierte Bereich nur $\{\pi\}$, – mit den Ergebnissen:

$$\begin{aligned} \max \{0, u - \tau\} &= 0 \\ \min \{\max \{0, u - \tau\}, 2\pi - (u - \tau)\} &= \min \{0, 2\pi - (u - \tau)\} = 0 \\ \min \{\pi, 2\pi - (u - \tau)\} &= \pi \\ \min \{\pi, \max \{0, u - \tau\}\} &= \min \{\pi, 0\} = 0 \end{aligned}$$

(ii) Andererseits aber auch eine Unterscheidung zwischen $o - \tau$ und $2\pi - 2\tau$, also zwischen τ und $2\pi - o$ zu

- $0 \leq \tau \leq \min \{2\pi - o, o\}$ mit:

$$\begin{aligned}
 \min \{o - \tau, 2\pi - 2\tau\} &= o - \tau \\
 \min \{2\pi - \min \{o - \tau, 2\pi - 2\tau\}, \pi\} &= \min \{2\pi - (o - \tau), \pi\} \\
 &= \min \{2\pi - o + \tau, \pi\} \\
 \min \{o - \tau, 2\pi - 2\tau, \pi\} &= \min \{o - \tau, \pi\} \\
 \min \{2\pi - (o - \tau), (o - \tau)\} &= \min \{2\pi - o + \tau, o - \tau\}
 \end{aligned}$$

Die verbleibenden Minima lassen sich in τ größer gleich oder kleiner $o - \pi$ auflösen.

- $\min \{2\pi - o, o\} \leq \tau \leq \min \{\pi, o\}$, dieser Fall ist nur für $o \geq \pi$ relevant, mit den Auflösungen:

$$\begin{aligned}
 \min \{o - \tau, 2\pi - 2\tau\} &= 2\pi - 2\tau \\
 \min \{2\pi - \min \{o - \tau, 2\pi - 2\tau\}, \pi\} &= \min \{2\pi - (2\pi - 2\tau), \pi\} \\
 &= \min \{2\tau, \pi\} \\
 \min \{o - \tau, 2\pi - 2\tau, \pi\} &= \min \{2\pi - 2\tau, \pi\} \\
 \min \{2\pi - (2\pi - 2\tau), (2\pi - 2\tau)\} &= 2 \min \{\tau, \pi - \tau\}
 \end{aligned}$$

Hier ist noch eine Differenzierung in τ größer gleich oder kleiner als $\frac{\pi}{2}$ notwendig.

Damit ergeben sich nun in den Grenzen zu den Bereichen der Integrale jeweils verschachtelte Extrema, was sich dadurch natürlich in die daraus resultierende funktionale Darstellung nach der Integration fortsetzen wird. Insgesamt würden schon alleine bei der Auswertung von $f_2((u, o))$ sieben Fallunterscheidungen und damit sieben verschiedene Bereiche zur Berechnung des Integrals entstehen.

10. Diskretisierung der Problemstellungen

In diesem Kapitel soll die diskrete Version der Problemstellungen zur Rundreise auf der Kreislinie eingeführt werden. Dabei werden zuerst die neuen Gegebenheiten definiert und die Erwartungswerte für den ersten Schritt der beiden Problemstellungen berechnet.

Danach wird als allgemeines Konzept zur Berechnung der erwarteten durchschnittlichen Schrittweite die Markovkette vorgestellt und die zur Anwendung dieses Konzept notwendige Vorgehensweise anhand eines Beispiels und einer Implementierung in Maple (verwendet wird die Version 11 unter Kubuntu mit 64 Bit mit einem Dual-Core-Prozessor und 4 Gigabyte RAM) dargestellt.

Aus dieser Implementierung werden schließlich für feste Werte des Informationsvorlaufs die durchschnittlichen Schrittweiten unter Variation der Anzahl der Positionen berechnet und für diese versucht eine geeignete funktionale Beschreibung mithilfe der rationalen Interpolation zu finden.

10.1. Aufbau der diskretisierten Problemstellung

Bei der diskreten Verteilung der Reisepunkte auf dem Einheitskreis soll dieser, bei n infrage kommenden möglichen Positionen, für die Reisepunkte nach einem gleich-verteilten Raster wie folgt aufgeteilt sein:

$$p_i = i \cdot \frac{2\pi}{n} \quad i = 0, \dots, n-1 \quad (10.1.1)$$

Dies ergibt während der Bearbeitung einer Sequenz aus Anfragen, bei einem Informationsvorlauf von k Anfragen, insgesamt eine Anzahl von n^{k+1} möglichen Situationen für die Verteilung der zu den Anfragen aus dem Informationsvorlauf gehörenden Reisepunkte und dem aktuellen Standort. Dabei setzt sich diese Anzahl, unter der Annahme, dass jeweils nach den einzelnen Anfragen unterschieden wird, grundsätzlich wie folgt zusammen:

- Es gibt n verschiedene Positionen für den aktuellen Standort,
- die Reisepunkte der k Anfragen aus dem Informationsvorlauf liegen ebenfalls jeweils auf einer der n Positionen.

Da die Anzahl von n^{k+1} möglichen Situation ziemlich hoch ist, und sich insbesondere auch viele Situationen etwa nur dadurch unterscheiden, dass Reisepunkte von zwei oder mehreren Anfragen vertauscht werden, muss über eine Vereinfachung der Darstellung zu den Situationen und einer damit einhergehenden Reduzierung von deren Anzahl nachgedacht werden.

Mit den folgenden Modifikationen lässt sich das vorgegebene Ziel der Vereinfachung erreichen:

- Werden die Reisepunkte losgelöst von der Unterscheidung der jeweils zugehörigen einzelnen Anfragen betrachtet, so entfallen schon mal all diejenigen Situationen, welche sich nur durch eine Permutation der Platzierungen der einzelnen Reisepunkte unterscheiden. Das heißt, es ist dann unerheblich, ob der Reisepunkt auf Position i zu der Anfrage σ_t oder zur Anfrage σ_j gehört.
- Wird die Entfernung der betrachteten Reisepunkte zum aktuellen Standort nicht mehr an die konkrete Position des aktuellen Standorts gekoppelt, sondern nur noch relativ für sich betrachtet, so entfallen auch die zu treffenden Unterscheidungen auf Grund der Position des aktuellen Standorts. Dadurch werden Situationen, welche sich nur durch eine Verschiebung aller Reisepunkte, einschließlich des aktuellen Standorts, um eine bestimmte Anzahl von Positionen unterscheiden, zusammengefasst.

Beide Maßnahmen zusammen führen schließlich zu einer Reduzierung der Anzahl der möglichen Situationen auf folgenden Wert:

$$\binom{n+k-1}{k} \quad (10.1.2)$$

Diese doch deutlich geringere Anzahl ergibt sich daher, dass nun nur noch k mögliche Entfernungen zum aktuellen Standort interessieren und es unerheblich ist, welche Positionen auf der Kreislinie beziehungsweise Anfragen sich dahinter verbergen.

Die erste Beschreibung der Situationen über die Positionen der Reisepunkte und des Standortes entsprach einem Ziehen von $k+1$ Kugeln aus einer Urne mit n Kugeln wobei jede gezogene Kugel wieder zurückgelegt wird und die Reihenfolge wesentlich ist. Die neue Beschreibung über die Entfernungen hingegen entspricht einem k -maligen Ziehen mit Zurücklegen aus der gleichen Urne mit n Kugeln, mit dem Zusatz, dass nicht mehr auf die Reihenfolge der Ziehung geachtet werden muss.

Für das Optimierungsproblem 8.1.1, also die gerichtete Rundreise auf der Kreislinie, ergibt sich folgende Liste für die möglichen Entfernungen:

$$0, \frac{2\pi}{n}, \dots, (n-1) \cdot \frac{2\pi}{n} \quad (10.1.3a)$$

Analog ergeben sich die Entfernungen für die Rundreise auf der Kreislinie mit Richtungswechsel nach Optimierungsproblem 7.1.1 zu:

$$0, \frac{2\pi}{n}, \dots, \left\lfloor \frac{n}{2} \right\rfloor \cdot \frac{2\pi}{n}, -\left\lfloor \frac{n-1}{2} \right\rfloor \cdot \frac{2\pi}{n}, \dots, -\frac{2\pi}{n} \quad (10.1.3b)$$

Hierbei entspricht ein positiver Wert einer Bewegung entgegen dem Uhrzeigersinn (beziehungsweise während der Bearbeitung einer Bewegung in der gleichen Richtung wie bei der letzten Bewegung) und ein negativer Wert einer entgegengesetzten Bewegung.

Die zwei gerundeten Werte ergeben sich daher, dass bei einer geraden Anzahl n von Positionen eine Position mit Entfernung π zum aktuellen Standort existiert. Ohne Beschränkung der Allgemeinheit wird im Fall einer Fahrt zu dieser Position eine Bewegung in der bisherigen Laufrichtung angenommen.

Bei geradem n gibt es also bei der Rundreise auf der Kreislinie mit Richtungswechsel $\frac{n}{2}$ mögliche Punkte mit positiver Entfernung und $\frac{n}{2} - 1$ mögliche Punkte mit negativer Entfernung zum aktuellen Standort. Wenn n dagegen ungerade ist, dann gilt $\left\lfloor \frac{n}{2} \right\rfloor = \frac{n-1}{2}$, das heißt, dass in beiden Richtungen jeweils gleich viele Positionen vorhanden sind.

Bei der Erzeugung aller möglichen Situationen für einen festen Informationsvorlauf von k Anfragen und n möglichen Positionen auf der Kreislinie können die einzelnen Situationen dann durch Vektoren aus der Menge Ω codiert werden.

$$\Omega = \{x \in \{0, 1, \dots, k\}^n : \mathbf{1}^T x = k\} \quad \text{mit} \quad |\Omega| = \binom{n+k-1}{k}$$

Bei der Übersetzung eines Vektors x aus der Menge Ω in eine mögliche Situation während der Bearbeitung eines Problems der Rundreise auf der Kreislinie bedeutet dann ein Eintrag t in der Komponente x_i , dass auf der Position in einer Entfernung von

$$(i-1) \cdot \frac{2\pi}{n}$$

vom aktuellen Standort bei einer Bewegung in der bisherigen Laufrichtung die Reisepunkte von t Anfragen zu finden sind.

10.2. Erwartungswerte erste Schrittweite

Exemplarisch sollen nun in der Diskretisierung der Problemstellungen zur Rundreise auf der Kreislinie analog zu den Beweisen für Satz 8.4.1 und Satz 9.4.1 die Erwartungswerte der ersten Schrittweite berechnet werden.

10.2.1. Gerichtete Rundreise auf der Kreislinie

Als erstes wird die gerichtete Rundreise auf der Kreislinie betrachtet. Für diese ergibt sich der Erwartungswert $\mathbb{E}[\tau_1]$ für die Schrittweite im ersten Schritt als die Auswertung der folgenden Formel:

$$\mathbb{E}[\tau_1] = k \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} i \cdot \frac{2\pi}{n} \cdot \frac{1}{n} \left(\sum_{j=i}^{n-1} \frac{1}{n} \right)^{(k-1)} \quad (10.2.1)$$

Dabei ergibt sich der Vorfaktor von k in (10.2.1) daraus, dass die Reisepunkte von k Anfragen für die Auswahl des zu bearbeitenden Reisepunktes zur Verfügung stehen. Die Wahrscheinlichkeit, dass der Reisepunkt einer bestimmten Anfrage auf einer Position i liegt ist $\frac{1}{n}$ und $i \cdot \frac{2\pi}{n}$ ist die Schrittweite zur Position i . Die Wahrscheinlichkeit dafür, dass keiner der Reisepunkte der anderen $k-1$ Anfragen näher am aktuellen Standort liegt als der Reisepunkt der ausgewählten Anfrage ist $(\sum_{j=i}^{n-1} \frac{1}{n})^{k-1}$. Aus (10.2.1) ergibt sich in der Fortführung dann:

$$\mathbb{E}[\tau_1] = k \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} \frac{2\pi}{n^2} \cdot i \cdot \left(\frac{n-i}{n} \right)^{k-1} \quad (10.2.2)$$

Wird nun $n-i$ durch j substituiert, so ändert sich die Form der Summanden aus (10.2.2) zu:

$$\begin{aligned} \mathbb{E}[\tau_1] &= 2\pi k \lim_{n \rightarrow \infty} \sum_{j=1}^n \frac{(n-j)j^{k-1}}{n^{k+1}} = 2\pi k \lim_{n \rightarrow \infty} \left(\sum_{j=1}^n \frac{j^{k-1}}{n^k} - \sum_{j=1}^n \frac{j^k}{n^{k+1}} \right) \\ &= 2\pi k \left(\frac{1}{k} - \frac{1}{k+1} \right) = \frac{2}{k+1} \pi \end{aligned} \quad (10.2.3)$$

Dieser Wert ergab sich auch in Satz 8.4.1 als erwartete erste Schrittweite wenn die Reisepunkte der Anfragen gleichmäßig über die ganze Kreislinie verteilt sind.

Bemerkung 10.2.1 (Faulhaber Formel).

Bei der vorletzten Umformung von (10.2.3) wird die Faulhabersche Formel, mit β_j als der j -ten Bernoulli-Zahl, sowie $\beta_0 = 1$,

$$\sum_{i=1}^n i^p = \frac{1}{p+1} \sum_{j=0}^p \binom{p+1}{j} \beta_j n^{p+1-j}, \quad (10.2.4)$$

welche zum Beispiel in [Weia] gefunden werden kann, für die Summierung über eine Zahlenreihe mit gleichen Potenzen angewendet.

Multipliziert mit $n^{-(p+1)}$ ergibt sich aus (10.2.4) im Grenzwert für n gegen unendlich der erste Summand der Summe auf der rechten Seite der Gleichung. Da außerdem noch

$\beta_0 = 1$ vorausgesetzt ist, gilt:

$$\lim_{n \rightarrow \infty} \frac{1}{n^{p+1}} \sum_{i=1}^n i^p = \frac{1}{p+1} \beta_0 = \frac{1}{p+1} \quad (10.2.5)$$

Da die Berechnung mit diskreten Wahrscheinlichkeiten arbeitet, sind eigentlich die Fälle in denen mindestens eine weitere Anfrage den gleichen Reisepunkte wie die ausgewählte Anfrage besitzt mehrmals eingerechnet, allerdings geht die Wahrscheinlichkeit für ein solches zufälliges Treffen, wie im kontinuierlichen Fall, mit zunehmender Anzahl der möglichen Punkte und damit vor allem im Grenzwert für n mit unendlich vielen Positionen gegen 0.

10.2.2. Normale Rundreise auf der Kreislinie

Bei der Berechnung des Erwartungswertes $\mathbb{E}[\tau_1]$ der ersten Schrittweite für die Rundreise auf der Kreislinie mit Richtungswechsel ergibt sich eine Unterscheidung in zwei Fälle.

Wird nämlich eine gerade Anzahl n an Positionen auf der Kreislinie verwendet, so gibt es immer genau eine Position auf der Kreislinie, welche exakt gegenüber dem aktuellen Standort liegt, dass führt nun dazu, dass eine Schrittweite von π zum nächsten Punkt nur mit einer Position realisiert werden kann und nicht wie bei den anderen $\lfloor \frac{n-1}{2} \rfloor$ Entfernungen je einmal pro Laufrichtung. Deshalb sind bei diesen die Wahrscheinlichkeiten für die Realisierung der Schrittweite auch $\frac{2}{\pi}$ statt nur $\frac{1}{\pi}$. Der Erwartungswert ergibt sich dann mit:

$$\begin{aligned} \mathbb{E}[\tau_1] &= k \lim_{n \rightarrow \infty} \left[0 + \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} \frac{2}{n} i \frac{2\pi}{n} \left(\sum_{j=i}^{\lfloor \frac{n-1}{2} \rfloor} \frac{2}{n} + \{n \text{ gerade}\} \frac{1}{n} \right)^{k-1} + \{n \text{ gerade}\} \frac{1}{n} \pi \frac{1}{n^k} \right] = \\ &= k\pi \left(\frac{2}{n} \right)^{k+1} \lim_{n \rightarrow \infty} \left[\sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} i \left(\left\lfloor \frac{n-1}{2} \right\rfloor + 1 - i + \frac{\{n \text{ gerade}\}}{2} \right)^{k-1} \right] + k \lim_{n \rightarrow \infty} \frac{\{n \text{ gerade}\} \pi}{n^{k+1}} \end{aligned} \quad (10.2.6)$$

Der zweite Teil von (10.2.6), also $k \frac{\{n \text{ gerade}\} \pi}{n^{k+1}}$, verschwindet beim Grenzübergang von n gegen unendlich und damit reduziert sich der Erwartungswert auf:

$$\mathbb{E}[\tau_1] = k\pi \left(\frac{2}{n} \right)^{k+1} \lim_{n \rightarrow \infty} \left[\sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} i \left(\left\lfloor \frac{n+1}{2} \right\rfloor - i + \frac{\{n \text{ gerade}\}}{2} \right)^{k-1} \right] \quad (10.2.7)$$

Nach einer Substitution mit $j = \lfloor \frac{n+1}{2} \rfloor - i$ ergibt sich aus (10.2.7) folgende Formel:

$$\mathbb{E}[\tau_1] = k\pi \left(\frac{2}{n}\right)^{k+1} \lim_{n \rightarrow \infty} \left[\sum_{j=1}^{\lfloor \frac{n-1}{2} \rfloor} \left(\left\lfloor \frac{n+1}{2} \right\rfloor - j \right) \left(j + \frac{\{n \text{ gerade}\}}{2} \right)^{k-1} \right] \quad (10.2.8)$$

Für ungerades n , also eine ungerade Anzahl an Positionen lässt sich (10.2.8) fortführen zu:

$$\mathbb{E}[\tau_1] = \lim_{n \rightarrow \infty} \left(\frac{2}{n}\right)^{k+1} \pi \sum_{j=1}^{\frac{n-1}{2}} \left(\frac{n+1}{2} - j \right) j^{k-1}$$

Dies lässt sich ebenfalls mit der Faulhaberschen Formel und dem Übergang zum Grenzwert nach (10.2.5) aus Bemerkung 10.2.1 mit

$$\lim_{n \rightarrow \infty} \left(\frac{n+1}{2} \right) \left(\frac{n-1}{2} \right)^k \left(\frac{2}{n} \right)^{k+1} = 1 \quad (10.2.9)$$

vereinfachen zu:

$$\mathbb{E}[\tau_1] = k\pi \left(\frac{1}{k} - \frac{1}{k+1} \right) = \frac{1}{k+1} \pi$$

Ist dagegen die Anzahl der Positionen eine gerade Zahl, so entsteht aus (10.2.8) die folgende Formel:

$$\mathbb{E}[\tau_1] = k\pi \left(\frac{2}{n}\right)^{k+1} \lim_{n \rightarrow \infty} \left[\sum_{j=1}^{\frac{n}{2}-1} \left(\frac{n}{2} - j \right) \left(j + \frac{1}{2} \right)^{k-1} \right] \quad (10.2.10)$$

Auf (10.2.10) kann nun allerdings die Faulhabersche Formel Bemerkung 10.2.1 nicht direkt angewandt werden, sondern es muss erst einmal nach oben und unten abgeschätzt werden. Die Abschätzung von (10.2.10) nach unten erfolgt durch das Weglassen von $\frac{1}{2}$ in $(j + \frac{1}{2})^{k-1}$, dies ergibt schließlich nach der Anwendung der Faulhaber Formel und des Grenzwertes wie in (10.2.5) sowie mit (10.2.9):

$$\mathbb{E}[\tau_1] = k\pi \left(\frac{2}{n}\right)^{k+1} \lim_{n \rightarrow \infty} \sum_{j=1}^{\frac{n}{2}-1} \left(\frac{n}{2} j^{k-1} - j^k \right) = k\pi \left(\frac{1}{k} - \frac{1}{k+1} \right) = \frac{1}{k+1} \pi$$

Nach oben ergibt sich die Abschätzung zu (10.2.10) über das Aufrunden von $\frac{1}{2}$ zu 1 und einer Verschiebung der Grenzen der Summe, sowie wiederum durch die Anwendung von

(10.2.5) und (10.2.9):

$$\begin{aligned}\mathbb{E}[\tau_1] &= k\pi \left(\frac{2}{n}\right)^{k+1} \lim_{n \rightarrow \infty} \sum_{j=1}^{\frac{n}{2}-1} \left(\frac{n}{2} - j\right) (j+1)^{k-1} \\ &= k\pi \left(\frac{2}{n}\right)^{k+1} \lim_{n \rightarrow \infty} \left(\sum_{j=1}^{\frac{n}{2}} \left[\left(\frac{n}{2} + 1\right) j^{k-1} - j^k \right] - \frac{n}{2} \right) = k\pi \left(\frac{1}{k} - \frac{1}{k+1} \right) = \frac{1}{k+1} \pi\end{aligned}$$

Da somit sowohl die Abschätzung nach oben als auch die Abschätzung nach unten den gleichen Grenzwert ergeben, geht also auch der Erwartungswert für die erste Schrittweite bei gerader Anzahl an Positionen im Grenzwert gegen $\frac{1}{k+1}\pi$, dieses Ergebnis ist identisch mit dem Grenzwert des Erwartungswertes für ungerade n und bestätigt die Aussage von Satz 9.4.1 über den Erwartungswert der ersten Schrittweite mit stetig verteilten Reisepunkten der Anfragen.

10.3. Allgemeine Schrittweite

Nach den Überlegungen in Abschnitt 10.1 ist die Anzahl der möglichen Situationen für einen gegebenen Informationsvorlauf k und einer Unterteilung der Kreislinie in n Positionen mit (10.1.2) gleich

$$\binom{n+k-1}{k} = \frac{(n+k-1)!}{k!(n-1)!}$$

und damit nicht wirklich für explizite von Hand durchführbare Berechnungen geeignet. Eine Möglichkeit, diese Schwierigkeit doch noch zu umgehen bietet das Konzept der Markovketten, – wie sie in [CJSW93] oder auch in [NR00] für Betrachtungen über das Bin Packing verwendet werden, – mit der Berechnung eines Fixpunkt-Vektors unter der Verwendung von Computer-Programmen zum Beispiel Maple. Eine ausführlichere Beschreibung zur Modellierung mit Markovketten findet sich etwa in [Pot09].

Da sich die Situationen in den beiden Problemstellungen nur dadurch unterscheiden, dass bei der Rundreise auf der Kreislinie mit Richtungswechsel statt einer Entfernung d , wenn diese größer als π ist, die Entfernung $2\pi - d$ und eine Änderung der Bewegungsrichtung verwendet wird, sind die theoretisch möglichen Situationen für beide Problemstellungen identisch. Einzig die hinter den einzelnen Komponenten stehenden Werte zur Distanz sind – wie in (10.1.3) gesehen – teilweise unterschiedlich.

Bemerkung 10.3.1 (Voraussetzungen für die Anwendung der Markovketten).

Voraussetzungen zur Anwendung dieser Methode für die Rundreise auf der Kreislinie sind:

- (i) Es wird eine unendliche Sequenz σ von Anfragen σ_i mit einer endlichen Grundmenge für die Reisepunkte betrachtet.
- (ii) Die Größe des Informationsvorlaufs k ist festgelegt und wird nicht mehr verändert.
- (iii) Die Menge aller möglichen Situationen für die Anordnung der Reisepunkte der Anfragen im Informationsvorlauf ist endlich. Diese Voraussetzung ist allerdings schon durch die beiden Voraussetzungen in Punkt (i) und Punkt (ii) erfüllt.
- (iv) Die Auswahl der nächsten zu bearbeitenden Anfrage hängt nur von deren Reisepunkt und den Reisepunkten der anderen Anfragen aus dem Informationsvorlauf ab. Damit ist gewährleistet, dass die Auswahl unabhängig von früheren Entscheidungen oder Situationen ist.

Unter den Voraussetzungen von Bemerkung 10.3.1 können die Wahrscheinlichkeiten für die Übergänge zwischen den einzelnen Situationen bei Verwendung des zu betrachtenden Algorithmus in einer quadratischen Matrix A mit Dimension b zusammengefasst werden, diese Matrix A wird im weiteren Verlauf dieser Arbeit als *Übergangsmatrix* bezeichnet. Dabei ist die Dimension b durch die Anzahl der möglichen Situationen vorgegeben.

Der Eintrag $A_{i,j}$ gibt dabei die Wahrscheinlichkeit an, mit welcher der Algorithmus einen Übergang von der Situation j in die Situation i nach der Bearbeitung des ausgewählten nächsten Reisepunktes und der Ankunft einer weiteren neuen Anfrage, mit einem auf allen Positionen gleichverteilten Reisepunkt, durchführt. Also gilt:

$$A_{i,j} = P(j \rightarrow i) \text{ sowie } \mathbf{1}^T A = \mathbf{1}^T$$

Dabei ist $P(j \rightarrow i)$ die Wahrscheinlichkeit dafür, dass bei vorgegebener Situation j nach der Bearbeitung der ausgewählten Anfrage durch den Algorithmus und der Bekanntgabe der neuen Anfrage die Situation i entsteht. In der Spalte $A_{\cdot,j}$ stehen somit die einzelnen Wahrscheinlichkeiten mit denen aus der Situation j eine der anderen Situationen wird. Die Diagonale $A_{j,j}$ muss dabei nicht zwingend gleich Null sein, da es auch möglich ist, dass eine Situation wieder in sich selbst übergeht.

Sei $\mathcal{P}$ nun ein Wahrscheinlichkeitsraum über die einzelnen Situationen und $p \in \mathcal{P}$ ein Vektor mit $p \geq 0$, sowie $p^T \mathbf{1} = 1$, dann beschreibt der Eintrag p_i die momentane Wahrscheinlichkeit für die Situation i .

Durch die Festlegung von A ergeben sich durch $p_{neu} = A \cdot p_{alt}$ die erwarteten Wahrscheinlichkeiten $p_{neu} \in \mathcal{P}$ für die einzelnen Situationen im nächsten Schritt aufbauend auf einem vorherigen Wahrscheinlichkeitsvektor $p_{alt} \in \mathcal{P}$ zu den möglichen Situationen. Hierbei repräsentieren die Einheitsvektoren die einzelnen Situationen als feste Zustände.

Wenn sich zu der Markovkette mit der Matrix A nun ein eindeutiger stabiler Punkt nach Definition 10.3.2 finden lässt, so kann einiges über das Average-Case-Verhalten des Algorithmus herausgefunden werden.

Um aus der Matrix mit den Wahrscheinlichkeiten für die Übergänge die erwartete durchschnittliche Schrittweite zu erhalten, muss für jede einzelne der Situationen aus Ω die durch die vorgegebene Belegung festgelegte Schrittweite im nächsten Schritt über die zur Problemstellung gehörige Distanzfunktion und den verwendeten Algorithmus bestimmt werden. Dies ergibt einen Vektor mit den Schrittweiten $\bar{\tau}$ über alle möglichen Situationen aus Ω , zusammen mit einem Wahrscheinlichkeitsvektor $\bar{p}$ über die Situationen, kann dann durch $\bar{\tau}^T \bar{p}$ die erwartete Schrittweite ausgerechnet werden.

Für die erwartete Schrittweite zur Grenzverteilung der Wahrscheinlichkeiten muss die Matrix A darauf untersucht werden, ob es einen Vektor v mit

$$v_i \geq 0 \quad \forall i \quad \text{und} \quad \mathbf{1}^T v = 1 \quad (10.3.1)$$

gibt, welcher sich nach einer Multiplikation mit A nicht verändert, also für den $Av = v$ gilt.

Definition 10.3.2 (Fixpunkt-Vektor/stabiler Punkt).

Ein Vektor v mit $Dv = v$ wird als *Fixpunkt-Vektor* zu D bezeichnet. Allgemein kann hierbei auch vom *stabilen Punkt* der Markovkette zu D gesprochen werden.

Ein Fixpunkt-Vektor v zu der Übergangsmatrix A , welcher zusätzlich noch die Anforderungen aus (10.3.1) erfüllt, beinhaltet die Wahrscheinlichkeiten für die einzelnen Situationen im Grenzwert der unendlichen Betrachtung. Das heißt, dass bei der Bearbeitung des zugehörigen Problems der Algorithmus nach einer genügend großen Zahl von Schritten mit Wahrscheinlichkeit 1 diesen Wahrscheinlichkeitsvektor der Situationen erreichen wird. Damit der Fixpunkt-Vektor diese Eigenschaft erfüllt, muss die Lösung von (10.3.1) eindeutig sein, dafür muss nach Bemerkung 10.3.1 ein Zusammenhalt zwischen den einzelnen Situationen bestehen.

Hierbei ist mit dem oben erwähnten Zusammenhalt der Situationen gemeint, dass im gerichteten Graphen der Übergänge zwischen den verschiedenen Situationen zwischen allen Paaren von Knoten ein gerichteter Weg existiert. Dieser Graph besteht dabei aus zu den einzelnen Situationen gehörenden Knoten und aus Bögen, wobei es zwischen zwei Knoten i und j genau dann einen gerichteten Bogen $i \rightarrow j$ gibt, wenn von der Situation i zur Situation j ein direkter Übergang möglich ist. Ein einzelner Knoten hat also folglich einen Ausgangsgrad von n , da es nach der Bearbeitung der ausgewählten Anfrage genau n verschiedene mögliche Positionen für den Reisepunkt der neuen Anfrage gibt.

Eine Markovkette, deren Graph diese Eigenschaft erfüllt, hat dann für jedes Paar aus Situationen eine positive Wahrscheinlichkeit für einen Übergang zwischen diesen beiden Situationen. Dieser Übergang muss nicht notwendigerweise direkt erfolgen, sondern kann auch einen Umweg über andere Situationen nehmen. Nach [Pot09, Kapitel 6] ist eine Markovkette mit dieser Eigenschaft regulär und damit auch ergodisch. Aus der Regularität der Markovkette ergibt sich außerdem die Existenz eines eindeutigen stabilen Punktes mit den gewünschten Eigenschaften aus (10.3.1).

Bei den beiden Rundreisen auf der Kreislinie und den zu ihnen vorgestellten Algorithmen ist, da von jeder Situation jede andere Situation mit endlich vielen Schritten erreicht werden kann, dieser Zusammenhalt gegeben.

Satz 10.3.3 (Erreichbarkeit der Situationen).

Sowohl für die Bearbeitung der Problemstellung der Rundreise auf der Kreislinie mit Richtungswechsel durch Algorithmus 9.1.1 als auch für die Bearbeitung der gerichteten Rundreise durch Algorithmus 8.1.2 gilt, dass von jeder beliebigen Situation aus alle anderen Situationen in endlicher Zeit erreichbar sind. Damit sind die zugehörigen Markovketten regulär und die Existenz des stabilen Punktes gesichert.

Beweis zu Satz 10.3.3.

Der Beweis gliedert sich in zwei Teile, zuerst wird die Aussage für die gerichtete Rundreise auf der Kreislinie und danach für die Rundreise auf der Kreislinie mit Richtungswechsel bewiesen, dabei wird jeweils der zur Problemstellung definierte Algorithmus verwendet. Als grundsätzliches Vorgehen wird für den jeweiligen Algorithmus beschrieben wie unter dessen Anwendung ausgehend von einer Situation x_0 mit $x_0 \in \Omega$ eine beliebige andere Situation $\bar{x} \in \Omega$ erreicht werden kann.

Gerichtete Rundreise: Sei dazu i die Position des Reisepunktes der Anfragen aus dem Informationsvorlauf zu x_0 , welcher den größten Abstand zu dem Standort – dieser kann ohne Beschränkung der Allgemeinheit als die Position 0 angenommen werden – von x_0 hat.

Als erstes werden nun solange neue Anfragen, mit einem Reisepunkt auf der Position i generiert, bis auf dieser Position genau k Reisepunkte versammelt sind, derweil werden durch die Anwendung von Algorithmus 8.1.2 die im Informationsvorlauf von x_0 vorhandenen Anfragen, deren Reisepunkt nicht auf der Position i liegt, bearbeitet.

Anschließend sind im Informationsvorlauf genau k Anfragen mit Reisepunkt auf der Position i , das heißt als nächstes wird eine dieser Anfragen bearbeitet und der aktuelle Standort wechselt auf die Position i . Wenn nun während der Abarbeitung dieser k Anfragen mit den Reisepunkten auf der Position i die nächsten maximal k neuen Anfragen so erzeugt werden, dass deren Reisepunkte zu dem aktuellen Standort, also der Position i , die in $\bar{x}$ festgelegten Entfernungen erzeugen, dann entsteht schlussendlich die Situation $\bar{x}$.

Der Aufwand dieser Vorgehensweise beträgt maximal $2k$ Schritte.

Rundreise mit Richtungswechsel: Da bei der Rundreise mit Richtungswechsel eine Bewegung sowohl im als auch gegen den Uhrzeigersinn möglich ist, muss die Vorgehensweise aus der gerichteten Rundreise angepasst werden.

Sei dazu wiederum die Position 0 der aktuelle Standort und i die Position des Reisepunktes der Anfragen aus dem Informationsvorlauf, welcher in der durch

die in der nächsten Auswahl von Algorithmus 8.1.2 vorgegebenen Richtung der Bewegung, die größte Entfernung zum aktuellen Standort besitzt.

Ohne Beschränkung der Allgemeinheit wird eine Bewegung entgegen den Uhrzeigersinn angenommen, auch in dem Fall, dass eine Anfrage mit einem Reisepunkt mit Entfernung Null vorhanden ist. Das bedeutet hier natürlich gleichzeitig, dass bei einer Bewegung zur Position i im Uhrzeigersinn kein anderer Reisepunkt auf einer der Positionen dazwischen liegen darf.

Nun wird solange der Algorithmus 8.1.2 mit Informationsvorlauf k angewandt, bis der aktuelle Standort auf die Position i aufgerückt ist. Dabei werden die Reisepunkte der neuen Anfragen auf zwei verschiedene Arten gesetzt. Wenn die Richtung der Bewegung zum Reisepunkt der Anfrage, welche im nächsten Schritt aus den $k - 1$ alten Anfragen des Informationsvorlaufs ausgewählt werden würde, wieder entgegen den Uhrzeigersinn ist, so wird der Reisepunkt der neuen Anfrage auf die Position i gesetzt, ansonsten auf die nächste Position entgegen den Uhrzeigersinn ausgehend vom aktuellen Standort.

Durch die Ermittlung der Richtung einer möglichen Bewegung innerhalb der $k - 1$ verbliebenen Anfragen und der eventuellen Anpassung des Reisepunktes der neuen Anfrage, wird verhindert, dass sich während der Bearbeitung die Richtung der Bewegung verändert. Damit bleibt der Bereich im Uhrzeigersinn zwischen dem aktuellen Standort und der Position i frei von den Reisepunkten der neuen Anfragen.

Wenn schließlich die Position i erreicht ist, dann sind im Informationsvorlauf genau $k - 1$ Anfragen mit einem Reisepunkt an der Position i und nun kann durch sukzessives Hinzufügen von Anfragen mit geeigneten Reisepunkten die Situation $\bar{x}$ erzeugt werden.

Damit werden in diesem Fall maximal $n + k$ Schritte benötigt. $\square$

Bemerkung 10.3.4 (Anmerkung zum Beweis von Satz 10.3.3).

Die im Beweis verwendeten Vorgehensweisen sind im Allgemeinen nicht die kürzesten Methoden zur Erzeugung der gewünschten Situation $\bar{x}$ aus der Ausgangssituation x_0 , das ist aber für den Beweis auch nicht notwendig.

10.3.1. Ein kurzes Beispiel zur Berechnung

Um das oben angesprochene Verfahren mit den Markovketten nicht nur abstrakt oder wie in Unterabschnitt 10.3.3 anhand von empirischen Ergebnissen zu betrachten soll in diesem Unterabschnitt ein einführendes Beispiel erläutert werden.

Beispiel 10.3.5 (Veranschaulichung der Markovketten und des stabilen Punktes).

Es sei ein Informationsvorlauf von zwei Anfragen und fünf mögliche Positionen auf der Kreislinie vorausgesetzt. Dann gibt es nach (10.1.2) während der Bearbeitung der gerichteten Rundreise auf der Kreislinie nach Optimierungsproblem 8.1.1 genau $\binom{6}{2} = 15$ mögliche Situationen, diese sind alle in Tabelle 10.3.1 aufgelistet.

Dabei sieht die Struktur von Tabelle 10.3.1 wie folgt aus, in der ersten Spalte steht als Ordnungsmerkmal die Nummer der jeweiligen Situation, in den nächsten fünf Spalten ist die Situation über die Darstellung mit den Entfernungen, also die Darstellung von $x_i \in \Omega$, enthalten. Die letzten beiden Spalten enthalten bei der Bearbeitung der vorliegenden Situation durch Algorithmus 8.1.2, einerseits die Menge der möglichen nachfolgenden Situationen, dabei ist, um es etwas übersichtlicher zu gestalten, immer nur die zur jeweiligen Situation gehörende Nummer angegeben, und andererseits die Schrittweite, welche sich aus dieser Situation für den Algorithmus 8.1.2 ergibt.

Nr.	0	$\frac{2}{5}\pi$	$\frac{4}{5}\pi$	$\frac{6}{5}\pi$	$\frac{8}{5}\pi$	mögliche Nachfolger nach Nr.	Schrittweite
1	2	0	0	0	0	{1, 2, 3, 4, 5}	0
2	1	1	0	0	0	{2, 6, 7, 8, 9}	0
3	1	0	1	0	0	{3, 7, 10, 11, 12}	0
4	1	0	0	1	0	{4, 8, 11, 13, 14}	0
5	1	0	0	0	1	{5, 9, 12, 14, 15}	0
6	0	2	0	0	0	{1, 2, 3, 4, 5}	$\frac{2}{5}\pi$
7	0	1	1	0	0	{2, 6, 7, 8, 9}	$\frac{2}{5}\pi$
8	0	1	0	1	0	{3, 7, 10, 11, 12}	$\frac{2}{5}\pi$
9	0	1	0	0	1	{4, 8, 11, 13, 14}	$\frac{2}{5}\pi$
10	0	0	2	0	0	{1, 2, 3, 4, 5}	$\frac{4}{5}\pi$
11	0	0	1	1	0	{2, 6, 7, 8, 9}	$\frac{4}{5}\pi$
12	0	0	1	0	1	{3, 7, 10, 11, 12}	$\frac{4}{5}\pi$
13	0	0	0	2	0	{1, 2, 3, 4, 5}	$\frac{6}{5}\pi$
14	0	0	0	1	1	{2, 6, 7, 8, 9}	$\frac{6}{5}\pi$
15	0	0	0	0	2	{1, 2, 3, 4, 5}	$\frac{8}{5}\pi$

Tabelle 10.3.1.: Mögliche Situationen der Entfernungen für die Reisepunkte der Anfragen aus dem Informationsvorlauf bei der gerichteten Rundreise

Als Übergangsmatrix ergibt sich mit diesen Daten die in (10.3.2) dargestellte Matrix.

$$A := \begin{pmatrix} 1 & & & & 1 & & & & 1 & & & & 1 & & & & 1 & & & & 1 \\ 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 \\ 1 & & 1 & & 1 & & 1 & & 1 & & 1 & 1 & & 1 & & & 1 & & & & 1 \\ 1 & & & 1 & 1 & & & & 1 & 1 & & & & 1 & & & 1 & & & & 1 \\ 1 & & & & 1 & 1 & & & 1 & & & & 1 & & & 1 & & & & 1 \\ & 1 & & & & 1 & & & & 1 & & & 1 & & & & 1 \\ & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & & & 1 \\ & 1 & & 1 & & 1 & & 1 & & 1 & & & 1 & & & 1 \\ & 1 & & & 1 & 1 & & & 1 & & & & 1 & & & 1 \\ & & 1 & & & 1 & & & & 1 & & & 1 & & & & 1 \\ & & 1 & 1 & & & 1 & 1 & & & 1 & & & 1 \\ & & 1 & & 1 & & 1 & & & 1 & & & 1 \\ & & & 1 & & & & 1 & & & 1 \\ & & & & 1 & 1 & & & 1 \\ & & & & & 1 & & & 1 \\ & & & & & & 1 \end{pmatrix} \quad (10.3.2)$$

Zu dieser Matrix gibt es natürlich, wie in den Anmerkungen zum Zusammenhalt nach der Definition 10.3.2 angesprochen, einen Graphen zur Darstellung der Übergänge. Da in

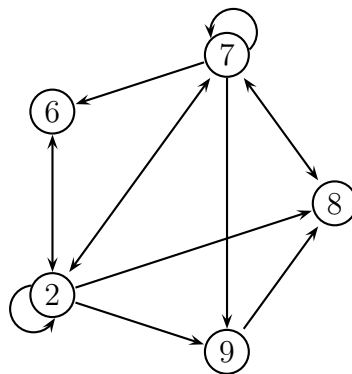


Abbildung 10.3.1.: Ausschnitt des Graphen der Übergänge zu Beispiel 10.3.5

dem Graphen zu der hier gegebenen Problemstellung 15 Knoten mit jeweils einem Grad von fünf Bögen enthalten sind, wird in Abbildung 10.3.1 nur ein Ausschnitt des Graphen zu den Situationen mit den Nummern 2, 6, 7, 8 und 9 gezeigt. Dabei steht ein Bogen mit einer Pfeilspitze an jeder Seite für zwei gegensätzliche Bögen zwischen den beiden Knoten, dass heißt, dass zwischen beiden Situationen ein direkter Wechsel möglich ist. Der komplette Graph ist auf Grund der Anzahl von $5 \cdot 15 = 75$ Bögen zu überladen um diesen übersichtlich darzustellen zu können. Dabei kann in Abbildung 10.3.1 gut erkannt

werden, dass sowohl die Situation 2 als auch die Situation 7 beide einen Übergang auf sich selbst zulassen, ebenso kann theoretisch zwischen den beiden Knoten 2 und 7 ein Kreislauf entstehen.

Die Matrix aus (10.3.2) besitzt schließlich folgenden stabilen Punkt:

$$v = \left(\frac{1}{25} \quad \frac{11}{100} \quad \frac{9}{100} \quad \frac{7}{100} \quad \frac{1}{20} \quad \frac{7}{100} \quad \frac{3}{25} \quad \frac{1}{10} \quad \frac{2}{25} \quad \frac{1}{20} \quad \frac{2}{25} \quad \frac{3}{50} \quad \frac{3}{100} \quad \frac{1}{25} \quad \frac{1}{100} \right) \quad (10.3.3)$$

Mit den Angaben über die vom Algorithmus 8.1.2 benötigten Distanzen zu den unterschiedlichen Situationen in Tabelle 10.3.1 und dem stabilen Punkt aus (10.3.3) errechnet sich insgesamt eine erwartete durchschnittliche Schrittweite von $\frac{2}{5}\pi$.

Eine Möglichkeit zur Berechnung von größeren Problemstellungen mit Maple wird in den beiden folgenden Unterabschnitten vorgestellt.

10.3.2. Implementierung in Maple

Um alle Vektoren aus Ω ohne einen rekursiven Durchlauf bestimmen zu können, kann auf einen kleinen Trick zurückgegriffen werden:

Zuerst wird eine Schleife mit allen Zahlen von 0 bis $n^k - 1$ durchlaufen, zu diesen Zahlen wird dann eine Basis-Darstellung zur Basis n berechnet, das ergibt einen k dimensionalen Vektor z mit Einträgen zwischen 0 und $n - 1$. Solch ein Vektor z wird dann als $x \in \Omega$ wie folgt interpretiert:

In der i -ten Komponente von z steht die Entfernung der i -ten noch nicht bearbeiteten Anfrage, das heißt wenn zum Beispiel $z_5 = 8$ gilt, so wird der neunte Eintrag von x um eins erhöht. Leider werden bei dieser Methode noch Duplikate erzeugt, so dass die berechneten Vektoren in einer Menge gespeichert werden müssen um diese Duplikate wirkungsvoll zu entfernen.

Durch die Anordnung der Entfernungen der Größe nach kann bei der Durchführung der Auswahl der nächsten zu bearbeitenden Anfrage, wie zum Beispiel in Schritt (ii) von Algorithmus 9.1.1, und deren Bearbeitung der neue Distanzvektor für die verbleibenden $k - 1$ Anfragen relativ einfach berechnet werden:

Nach der Auswahl der Anfrage $\sigma_{\pi(\ell)}$ mit Entfernung $(i - 1) \cdot \frac{2\pi}{n}$ als an Stelle ℓ zu bearbeitende Anfrage, ergibt sich der nachfolgende Entfernungsvektor x' ohne die neue Anfrage $\sigma_{\ell+k}$ aus dem aktuellen Entfernungsvektor x nach folgender Rechenvorschrift:

$$x'_1 = x_i - 1, \quad x'_j = x_p \text{ mit } p = [(i + j - 2) \bmod n] + 1 \quad \forall i \in \{2, \dots, n\} \quad (10.3.4)$$

Um daraus den neuen aktuellen Entfernungsvektor zu erhalten, muss nur noch die neue Anfrage $\sigma_{\ell+k}$ eingearbeitet werden. Dazu wird die Entfernung des Reisepunktes von $\sigma_{\ell+k}$ zum aktuellen Standort bestimmt und an der entsprechenden Stelle im Vektor x' der Wert des Eintrags um eins erhöht.

Bemerkung 10.3.6.

Bei der Betrachtung mit den Entfernungsvektoren ist darauf zu achten, dass bei der

Auswahl der nächsten Anfrage diejenige noch nicht bearbeitete Anfrage verwendet wird, deren Reisepunkt die kürzeste Entfernung zum Reisepunkt der letzten bearbeiteten Anfrage besitzt.

Die kürzeste Entfernung wird dabei allerdings für Optimierungsproblem 7.1.1, also der Rundreise auf der Kreislinie mit Richtungswechsel durch die in Definition 7.3.1 definierte Distanzfunktion unabhängig von der benutzten Laufrichtung gemessen. Eine Bewegung zu einem Reisepunkt mit Entfernung $-\frac{\pi}{4}$ wird dabei genauso gewertet wie die Bewegung zu einem Reisepunkt mit Entfernung $\frac{\pi}{4}$, das heißt, dass der Vektor x abwechselnd von beiden Richtungen aus, also in der Reihenfolge $x_1, x_2, x_n, x_3, x_{n-1}, \dots$, untersucht werden muss.

Die für die Berechnung der durchschnittlichen Schrittweite mithilfe der Markovkette notwendigen Prozeduren wurden in zwei Maple-Bibliotheken *CreateKreis* und *AverageKreis* gesammelt. Eine genauere Beschreibung dieser Bibliotheken findet sich in Abschnitt A.3. Für die in Unterabschnitt 10.3.3 vorgestellten Ergebnisse sind aus diesen Bibliotheken nur die beiden Prozeduren *avgeinfach* und *avgnormal* wichtig, diese berechnen jeweils für zwei Werte k , die Größe des Informationsvorlaufs, und n , die Anzahl der Positionen auf der Kreislinie, die im Durchschnitt zu erwartende Schrittweite mithilfe der oben beschriebenen Methode.

10.3.3. Ergebnisse mit Maple

Da die Anzahl der möglichen Situationen aus Ω nach (10.1.2) leider nicht polynomial in den beiden Eingabegrößen für den Informationsvorlauf und die Anzahl der Positionen ist, kommt unglücklicher Weise auch Maple (zumindest in der Version 9.5 unter XP (32 Bit) mit einem Dual-Core-Prozessor) bei der Berechnung des Fixpunkt-Vektors irgendwann an seine Grenzen des Machbaren.

Gerichtete Rundreise auf der Kreislinie

Bei der gerichteten Rundreise auf der Kreislinie berechnet Maple mit einem Informationsvorlauf von $k = 2$, die in Tabelle 10.3.2 dargestellten Werte. Insgesamt ergibt sich dabei eine funktionale Beziehung von

$$\bar{\tau} = \frac{n-1}{2n}\pi$$

und somit wird damit die Aussage von Satz 8.5.1, mit $\bar{\tau} = \frac{1}{2}\pi$ für $k = 2$ im Grenzwert für n gegen unendlich bestätigt.

Die Berechnungen für einen Informationsvorlauf von drei Anfragen ergaben das in Tabelle 10.3.3 gesammelte Ergebnis. Diese Werte lassen sich für $k = 3$ alle durch die Formel

$$\bar{\tau} = \frac{n-1}{3n}\pi$$

beschreiben und diese liefert als Grenzwert für n gegen unendlich den Wert $\frac{1}{3}\pi$. In der

n	2	3	4	5	6	7	8	9	10	11	12	13	14
$\frac{\bar{\tau}}{\pi}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{3}{8}$	$\frac{2}{5}$	$\frac{5}{12}$	$\frac{3}{7}$	$\frac{7}{16}$	$\frac{4}{9}$	$\frac{9}{20}$	$\frac{5}{11}$	$\frac{11}{24}$	$\frac{6}{13}$	$\frac{13}{28}$
n	15	16	17	18	19	20	21	22	23	24	25	26	27
$\frac{\bar{\tau}}{\pi}$	$\frac{7}{15}$	$\frac{15}{32}$	$\frac{8}{17}$	$\frac{17}{36}$	$\frac{9}{19}$	$\frac{19}{40}$	$\frac{10}{21}$	$\frac{21}{44}$	$\frac{11}{23}$	$\frac{23}{48}$	$\frac{12}{25}$	$\frac{25}{52}$	$\frac{13}{27}$
n	28	29	30	31	32	33	34	35	36	37	38	39	40
$\frac{\bar{\tau}}{\pi}$	$\frac{27}{56}$	$\frac{14}{29}$	$\frac{29}{60}$	$\frac{15}{31}$	$\frac{31}{64}$	$\frac{16}{33}$	$\frac{33}{68}$	$\frac{17}{35}$	$\frac{35}{72}$	$\frac{18}{37}$	$\frac{37}{76}$	$\frac{19}{39}$	$\frac{39}{80}$
n	41	42	43	44	45	46	47	48	49	50	51	52	53
$\frac{\bar{\tau}}{\pi}$	$\frac{20}{41}$	$\frac{41}{84}$	$\frac{21}{43}$	$\frac{43}{88}$	$\frac{22}{45}$	$\frac{45}{92}$	$\frac{23}{47}$	$\frac{47}{96}$	$\frac{24}{49}$	$\frac{49}{100}$	$\frac{25}{51}$	$\frac{51}{104}$	$\frac{26}{53}$

Tabelle 10.3.2.: Werte für einen Informationsvorlauf von zwei Anfragen bei der diskreten gerichteten Rundreise auf der Kreislinie

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$\frac{\bar{\tau}}{\pi}$	$\frac{1}{6}$	$\frac{2}{9}$	$\frac{1}{4}$	$\frac{4}{15}$	$\frac{5}{18}$	$\frac{2}{7}$	$\frac{7}{24}$	$\frac{8}{27}$	$\frac{3}{10}$	$\frac{10}{33}$	$\frac{11}{36}$	$\frac{4}{13}$	$\frac{13}{42}$	$\frac{14}{45}$	$\frac{5}{16}$	$\frac{16}{51}$
n	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
$\frac{\bar{\tau}}{\pi}$	$\frac{17}{54}$	$\frac{6}{19}$	$\frac{19}{60}$	$\frac{20}{63}$	$\frac{7}{22}$	$\frac{22}{69}$	$\frac{23}{72}$	$\frac{8}{25}$	$\frac{25}{78}$	$\frac{26}{81}$	$\frac{9}{28}$	$\frac{28}{87}$	$\frac{29}{90}$	$\frac{10}{31}$	$\frac{31}{96}$	$\frac{32}{99}$

Tabelle 10.3.3.: Werte für einen Informationsvorlauf von drei Anfragen bei der diskreten gerichteten Rundreise auf der Kreislinie

letzten angegebenen Instanz mit 27 möglichen Positionen für die Reisepunkte müssen allerdings schon 3654 Situationen berücksichtigt werden.

Bei einem Informationsvorlauf von vier Anfragen ist die Grenze des berechenbaren schnell erreicht, immerhin gibt es dabei für 33 zur Verfügung stehende Positionen auch schon 6545 unterschiedliche Situationen. Zu den in Tabelle 10.3.4 erfassten Daten wird nach der Interpolation mit einer rationalen Funktion, als Ergebnis die Funktion

$$\bar{\tau} = \frac{n-1}{4n}\pi$$

vorgeschlagen. Der Grenzwert für unendlich viele Positionen auf der Kreislinie ergibt

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\frac{\bar{\tau}}{\pi}$	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{3}{16}$	$\frac{1}{5}$	$\frac{5}{24}$	$\frac{3}{14}$	$\frac{7}{32}$	$\frac{2}{9}$	$\frac{9}{40}$	$\frac{5}{22}$	$\frac{11}{48}$	$\frac{3}{13}$	$\frac{13}{56}$	$\frac{7}{30}$

Tabelle 10.3.4.: Werte für einen Informationsvorlauf von vier Anfragen bei der diskreten gerichteten Rundreise auf der Kreislinie

somit im Fall eines Informationsvorlaufs von $k = 4$ Anfragen den Wert $\frac{1}{4}\pi$.

Dadurch, dass für die ersten Werte des Informationsvorlaufs bei der Beschreibung der

errechneten Daten über den Informationsvorlauf immer $\frac{1}{k}\pi$ als Grenzwert für n gegen unendlich, also für den Fall von kontinuierlich verteilten Reisepunkten auf der Kreislinie, herauskommt, ergibt sich eine Bestätigung der in Abschnitt 8.3 festgestellten Funktion aus der Regressionsanalyse.

Rundreise auf der Kreislinie mit Richtungswechsel

Bei den Berechnungen zur Rundreise auf der Kreislinie mit Richtungswechsel ergeben sich dagegen die in Tabelle 10.3.5 zusammengefassten Werte für einen Informationsvorlauf von zwei Anfragen. Zusammen lassen sich diese Schrittweiten durch

n	2	3	4	5	6	7	8	9	10	11	12	13	14
$\frac{\bar{\tau}}{\pi}$	$\frac{1}{4}$	$\frac{8}{27}$	$\frac{5}{16}$	$\frac{8}{25}$	$\frac{35}{108}$	$\frac{16}{49}$	$\frac{21}{64}$	$\frac{80}{243}$	$\frac{33}{100}$	$\frac{40}{121}$	$\frac{143}{432}$	$\frac{59}{169}$	$\frac{65}{196}$
n	15	16	17	18	19	20	21	22	23	24	25	26	27
$\frac{\bar{\tau}}{\pi}$	$\frac{224}{675}$	$\frac{85}{256}$	$\frac{96}{289}$	$\frac{323}{972}$	$\frac{120}{361}$	$\frac{133}{400}$	$\frac{440}{1323}$	$\frac{161}{484}$	$\frac{176}{529}$	$\frac{575}{1728}$	$\frac{208}{625}$	$\frac{225}{676}$	$\frac{728}{2187}$
n	28	29	30	31	32	33	34	35	36	37	38	39	40
$\frac{\bar{\tau}}{\pi}$	$\frac{261}{784}$	$\frac{280}{841}$	$\frac{899}{2700}$	$\frac{320}{961}$	$\frac{341}{1024}$	$\frac{1088}{3267}$	$\frac{385}{1156}$	$\frac{408}{1225}$	$\frac{1295}{3888}$	$\frac{456}{1369}$	$\frac{481}{1444}$	$\frac{1520}{4563}$	$\frac{533}{1600}$

Tabelle 10.3.5.: Werte für einen Informationsvorlauf von zwei Anfragen bei der diskreten Rundreise auf der Kreislinie mit Richtungswechsel

$$\bar{\tau} = \frac{n^2 - 1}{3n^2} \pi$$

in Abhängigkeit von der Anzahl der Positionen beschreiben. Der Grenzwert für n gegen unendlich liefert hierbei mit $k = 2$ einen Wert von $\frac{1}{3}\pi = \frac{1}{k+1}\pi$.

Anders sieht es dagegen bei den errechneten Ergebnissen zur Schrittweite bei einem Informationsvorlauf von drei Anfragen aus. Hier wird die Anzahl der Stellen für den Bruch $\frac{\bar{\tau}}{\pi}$ ziemlich schnell größer und es lässt sich, anders als bei den vorherigen Tabellen der Werte, nicht mehr mithilfe einer rationalen Interpolation eine brauchbare Darstellung als Funktion finden. Die gefundene Funktion verwendet nämlich sowohl im Nenner als auch im Zähler Polynome mit Grad neun um die berechneten 19 Schrittweiten zu interpolieren.

Die in Tabelle 10.3.6 präsentierten Werte sind nur unvollständig wiedergegeben, das

n	2	3	4	5	6	7	8	9	10	11	12	13	14
$\frac{\bar{\tau}}{\pi}$	$\frac{1}{6}$	$\frac{52}{243}$	$\frac{1801}{7728}$	$\frac{354923}{1466000}$	...								

Tabelle 10.3.6.: Werte für einen Informationsvorlauf von drei Anfragen bei der diskreten Rundreise auf der Kreislinie mit Richtungswechsel

liegt daran, dass schon der Eintrag zu $n = 6$ einen Nenner mit zehn Stellen besitzt. Die

Anzahl der Stellen steigt dann soweit, dass am Ende für die Schrittweite zu $n = 20$ allein schon fünf Zeilen benötigt werden.

10.4. Fazit

Nach der Diskretisierung der Rundreise auf der Kreislinie kann aus den mit Maple gewonnenen Ergebnissen folgendes Fazit gezogen werden:

- Bei der gerichteten Variante bestätigt sich der Eindruck aus den empirischen Beobachtungen in Unterabschnitt 8.3.2, dass es eine allgemeine Formel für die erwartete Schrittweite abhängig vom Informationsvorlauf gibt.
- Die Berechnungen zur Rundreise auf der Kreislinie mit Richtungswechsel hingegen zeigen wie auch schon die Ergebnisse der Regressionsanalyse der empirischen Daten in Abschnitt 9.3, dass sich die für die gerichtete Rundreise auf der Kreislinie gewonnenen Erkenntnisse nicht so einfach übertragen lassen.

Allgemein muss allerdings auch festgestellt werden, dass durch die Reduktion zu endlich vielen Positionen auf der Kreislinie sich nicht die eigentlich erhoffte Vereinfachung eingestellt hat. Dies liegt insbesondere in der Formel für die Anzahl der Situationen für eine bestimmte Konstellation der Rundreise auf der Kreislinie begründet, welche, durch ihre nicht gegebene Polynomialität in den Eingabedaten, eine einfachere Berechnung der erwarteten Schrittweite verhindert.

11. Ausblick auf die Rundreise auf der Oberfläche einer Kugel

Nach der Betrachtung der Rundreise auf der Kreislinie, also auf einer geschlossenen eindimensionalen Kurve im zweidimensionalen Raum, soll in diesem Kapitel ein kurzer Ausblick auf die Situation im dreidimensionalen Raum, mit der zweidimensionalen Oberfläche der Kugel als Grundlage für die Reisepunkte der Anfragen, aufgezeigt werden.

11.1. Problemstellung

Zuerst einmal werden die grundlegenden Begriffe für die einzelnen Kugelschnitte festgelegt.

Definition 11.1.1 (Grundlegende Begriffe).

Längengrad: Der *Längengrad* $\theta \in (-\pi, \pi]$ ist der Winkel, welcher sich nach der Projektion des Punktes auf die (x_1, x_2) -Ebene zwischen der x_1 -Achse und der Verbindung der Projektion mit dem Nullpunkt ergibt. Die positive x_2 -Achse hat dabei einen Längengrad von $\frac{\pi}{2}$, der negative Teil der x_2 -Achse entsprechend einen Längengrad von $-\frac{\pi}{2}$.

Breitengrad: Durch den *Breitengrad* $\Phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ hingegen wird der Winkel zwischen der (x_1, x_2) -Ebene und der Verbindung des Punktes zum Nullpunkt angegeben. Die obere Halbkugel entspricht dabei dem Winkelbereich $[0, \frac{\pi}{2}]$, die untere Halbkugel entspricht analog dem Bereich $[-\frac{\pi}{2}, 0]$.

Groß- und Kleinkreis: Ein *Großkreis* entsteht durch den Schnitt einer durch den Mittelpunkt der Kugel gehenden Ebene mit der Oberfläche der Kugel, liegt der Mittelpunkt der Kugel nicht in der Ebene, so entsteht ein *Kleinkreis* beim Schnitt der Ebene mit der Kugel.

Äquator: Als *Äquator* wird derjenige Großkreis bezeichnet, welcher aus dem Schnitt der Oberfläche der Kugel mit der (x_1, x_2) -Ebene entsteht.

Kugelschicht: Eine *Kugelschicht* entsteht dadurch, dass mit zwei parallelen Ebenen durch eine Kugel geschnitten wird, und zwar ist die Kugelschicht dann genau der Teil der Kugel, welcher sich zwischen den beiden Ebenen befindet.

Kugelzone: Als Oberfläche der Kugelschicht wird in dieser Arbeit nur derjenige Teil der Kugelschicht verwendet, welcher auch auf der Oberfläche der gesamten Kugel liegt, dieser Teil der Oberfläche wird auch *Kugelzone* genannt. Das Innere der Schnittkreise gehört folglich nach der Definition nicht zur Kugelzone.

Kugelkappe: Wenn bei einer Kugelschicht eine der Schnittebenen tangential auf der Oberfläche der Kugel steht, so wird die sich daraus ergebende Kugelzone auch *Kugelkappe* genannt. Eine Kugelkappe ist also ausgehend von einem Punkt auf der Oberfläche der Kugel der ganze Bereich auf der Kugeloberfläche bis zu einem Kleinkreis in einer bestimmten vorher festgelegten Entfernung. Dabei ist der Normalenvektor dieses Kleinkreises parallel zu der Verbindung zwischen dem Punkt auf der Oberfläche und dem Mittelpunkt der Kugel.

Als nächstes wird die Problemstellung der Rundreise zwischen verschiedenen Punkten auf der Oberfläche der Kugel definiert.

Optimierungsproblem 11.1.2 (Rundreise auf der Oberfläche der Kugel).

Bei der Rundreise auf der Oberfläche der Kugel, ist anders als bei den beiden Problemstellungen der Rundreise auf der Kreislinie nun nicht mehr nur die Kreislinie die mögliche Grundmenge, sondern die Anfragen verteilen sich auf der gesamten Oberfläche der Kugel. Da diese in den dreidimensionalen Raum eingebettet ist, reicht für die Darstellung der Reisepunkte nicht mehr nur ein einzelner Winkel aus, sondern wir benötigen dazu sowohl den Längengrad $\theta \in (-\pi, \pi]$ als auch den Breitengrad $\Phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

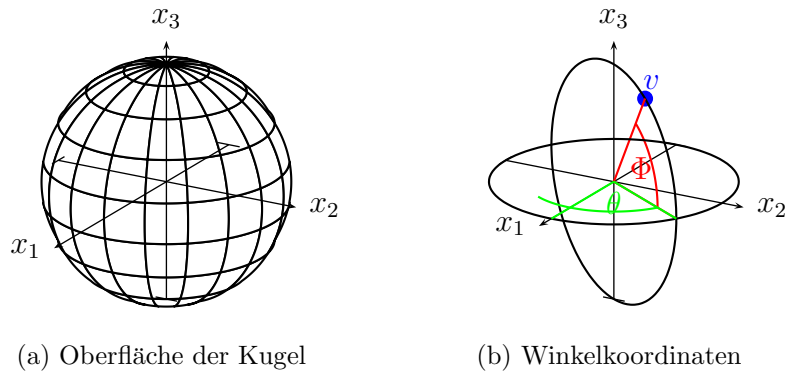


Abbildung 11.1.1.: Darstellung der Oberfläche der Kugel

In Abbildung 11.1.1 ist im ersten Teil die Oberfläche der Kugel mit einzelnen Segmenten dargestellt. Der zweite Teil zeigt einen Punkt v und dessen Längengrad θ sowie dem Breitengrad Φ . Da das Optimierungsproblem 11.1.2 nur auf der Oberfläche der Kugel arbeitet, reichen die Angabe des Längen- und Breitengrads (im weiteren Verlauf als Winkelkoordinaten bezeichnet) für die eindeutige Identifikation des Punktes aus. Der

für die Polarkoordinaten eigentlich noch notwendige Radius r entspricht dem Radius der Kugel und kann ohne Beschränkung der Allgemeinheit auf eins gesetzt werden.

Die Umrechnung der Winkelkoordinaten (θ, Φ) mit den hier vorgegebenen Bereichen in das normale dreidimensionale Koordinatensystem erfolgt über folgende Formel:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \cos(\theta) \cos(\Phi) \\ \sin(\theta) \cos(\Phi) \\ \sin(\Phi) \end{pmatrix} \quad (11.1.1)$$

Als Distanz τ zwischen zwei Punkten v_1, v_2 auf der Kugel kann, analog zur Rundreise auf der Kreislinie, wiederum der Winkel zwischen den beiden Verbindungskanten zum Mittelpunkt der Kugel M verwendet werden.

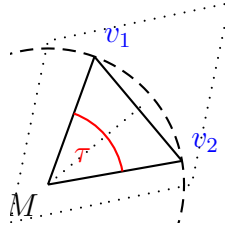


Abbildung 11.1.2.: Berechnung der Distanz zweier Punkte

Allgemein kann die Berechnung dieses Winkels, unabhängig von der Dimension des Raumes, durch eine Reduktion auf die durch den Mittelpunkt M der Sphäre und die zwei Punkte v_1, v_2 auf der Oberfläche bestimmte Ebene vereinheitlicht werden.

In Abbildung 11.1.2 sind die dazu notwendigen Einzelheiten eingezeichnet, die gepunktete Raute steht dabei für die durch die drei Punkte definierte Ebene zur Berechnung des Winkels, der gestrichelte Bogen stellt die gemeinsame Verbindung der beiden Punkte v_1 und v_2 , deren Länge gesucht wird, dar. Damit ergibt sich der Abstand $\tau \in [0, \pi)$ bei einem Radius von eins zu:

$$\tau = 2 \arcsin \left(\frac{1}{2} \|v_1 - v_2\| \right) \quad (11.1.2)$$

11.2. Erzeugung von gleichverteilten Zufallspunkten auf der Oberfläche der Kugel

Die einfachste und schnellste Variante zur Erzeugung von gleichmäßig verteilten Zufallspunkten auf der Oberfläche der Kugel wäre sicherlich die Generierung zweier Zufallszahlen z_1, z_2 aus $[-1, 1]$ und danach eine Umrechnung in Winkelkoordinaten über:

$$\theta = \pi z_1 \text{ und } \Phi = \frac{1}{2}\pi z_2 \quad (11.2.1)$$

Bemerkung 11.2.1.

Der Versuch der Erzeugung von gleich verteilten Punkten auf der Oberfläche einer Kugel über (11.2.1) funktioniert zwar für den Längengrad θ , allerdings nicht für den Breitengrad Φ . Das liegt daran, dass beim Breitengrad, anders als beim Längengrad, die durch die Punkte mit dem gleichen Breitengrad definierten Kreise alle einen unterschiedlichen Umfang besitzen. Der Äquator hat als Großkreis zum Beispiel eine Länge von 2π , jeder andere Breitengrad $\Phi \neq 0$ hat jedoch nur eine Länge von $2\pi \cos(\Phi)$ und ist somit ein Kleinkreis. Im Gegensatz zu den unterschiedlichen Kreislängen wäre nach der einfachen Verteilung die Wahrscheinlichkeit einen Punkt auf dem Äquator zu erhalten, identisch mit der, dass der Punkt auf einem anderen Breitenkreis liegt.

In der Abbildung 11.2.1b ist eine Menge von Punkten auf der Oberfläche der Kugel abgebildet, welche nach der in (11.2.1) gegebenen Methode zufällig erzeugt wurden. Dabei ist klar erkennbar, dass sich diese Punkte am Nordpol deutlich häufen. In der daneben liegenden Abbildung 11.2.1a wurden die Punkte mit der in (11.2.2) definierten Methode erzeugt.

Diese Häufung der Punkte in dem Bereich der Kugel an den beiden Polstellen der x_3 -Achse kann dadurch behoben werden, dass zur Berechnung der Winkelkoordinaten beim Breitengrad nicht $\Phi = \frac{1}{2}\pi z_2$ verwendet wird, sondern eine nach [Weib] korrigierte Version:

$$\theta = \pi z_1 \text{ und } \Phi = \arccos(z_2) - \frac{\pi}{2} \quad (11.2.2)$$

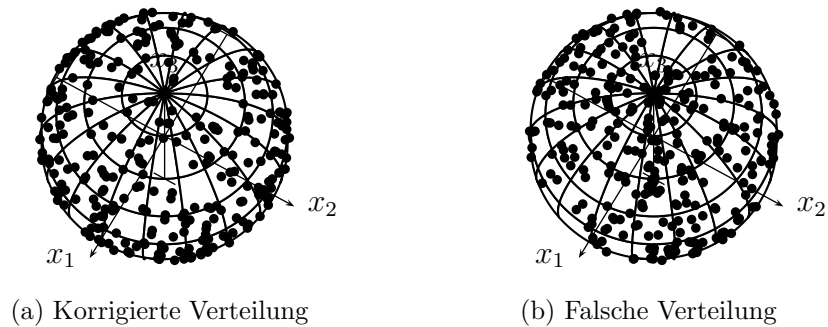


Abbildung 11.2.1.: Veranschaulichung der Verteilungen für zufällige Punkte auf der Oberfläche einer Kugel

Seien nun zwei unterschiedliche Kugelschichten A_1, A_2 mit den dazugehörigen Kugelzonen der Kugel, welche jeweils durch einen festgelegten Bereich innerhalb des Breitengrads mit Größe $\Delta > 0$ definiert sind, gegeben. Dann kann ausgehend von deren minimalen Breitengraden Φ_1 und Φ_2 , zuerst einmal der Flächeninhalt der zugehörigen Kugelzonen

bestimmt und die zugehörigen Wahrscheinlichkeiten berechnet werden, die sich daraus ergebenden Quotienten sollten dann bei einer Gleichverteilung übereinstimmen.

Der Flächeninhalt $\lambda_{n-1}(\omega_n)$ zur Oberfläche ω_n einer n -dimensionalen Kugel Ω_n mit Winkelkoordinaten

$$\Phi_1 \in [0, 2\pi], \Phi_2 \in [0, \pi], \dots, \Phi_{n-1} \in [0, \pi]$$

berechnet sich nach [Bor87, Abschnitt 6.2] allgemein zu:

$$\lambda_{n-1}(\omega_n) = \int_0^{2\pi} \int_0^\pi \cdots \int_0^\pi \sin(\Phi_2) (\sin(\Phi_3))^2 \cdots (\sin(\Phi_{n-1}))^{n-2} d\Phi_{n-2} \cdots d\Phi_3 d\Phi_2 d\Phi_1 \quad (11.2.3)$$

Nach einer Anpassung von (11.2.3) auf die geänderten Bereiche

$$\varphi_1 \in [-\pi, \pi], \varphi_2 \in \left[-\frac{1}{2}\pi, \frac{1}{2}\pi\right], \dots, \varphi_{n-1} \in \left[-\frac{1}{2}\pi, \frac{1}{2}\pi\right]$$

der Winkelkoordinaten ergibt sich mit $\sin(\Phi_i) = \sin(\varphi_i + \frac{1}{2}\pi) = \cos(\varphi_i)$:

$$\lambda_{n-1}(\omega_n) = \int_{-\pi}^\pi \int_{-\frac{1}{2}\pi}^{\frac{1}{2}\pi} \cdots \int_{-\frac{1}{2}\pi}^{\frac{1}{2}\pi} \cos(\varphi_2) (\cos(\varphi_3))^2 \cdots (\cos(\varphi_{n-1}))^{n-2} d\varphi_{n-2} \cdots d\varphi_3 d\varphi_2 d\varphi_1$$

Mit den in Optimierungsproblem 11.1.2 vorgegebenen Bezeichnern errechnet sich dann der Flächeninhalt zur Oberfläche der Kugel im $\mathbb{R}^3$ durch die Formel:

$$\lambda_2(\omega_3) = \int_{-\pi}^\pi \int_{-\frac{1}{2}\pi}^{\frac{1}{2}\pi} \cos(\phi) d\phi d\theta \quad (11.2.4)$$

Der Flächeninhalt F der Kugelzone mit einem Breitengrad zwischen $\tilde{\Phi}$ und $\tilde{\Phi} + \Delta$ berechnet sich dann abgeleitet aus (11.2.4) als:

$$\begin{aligned} F &= \int_{-\pi}^\pi \int_{\tilde{\Phi}}^{\tilde{\Phi}+\Delta} \cos(\phi) d\phi d\theta = \int_{\tilde{\Phi}}^{\tilde{\Phi}+\Delta} \cos(\phi) \int_{-\pi}^\pi d\theta d\phi \\ &= \int_{\tilde{\Phi}}^{\tilde{\Phi}+\Delta} 2\pi \cos(\phi) d\phi = 2\pi \left(\sin(\tilde{\Phi} + \Delta) - \sin(\tilde{\Phi}) \right) \quad (11.2.5) \end{aligned}$$

Die Wahrscheinlichkeit, dass ein, mit Verwendung der Formel zur Umrechnung aus (11.2.2), erzeugter Punkt auf der Kugelzone zur Kugelschicht A_i liegt, errechnet sich wie folgt:

$$\begin{aligned} P(\phi \in A_i) &= P\left(\arccos(z_2) - \frac{\pi}{2} \in [\tilde{\Phi}, \tilde{\Phi} + \Delta]\right) \\ &= P\left(z_2 \in \left[\cos\left(\frac{\pi}{2} + \tilde{\Phi} + \Delta\right), \cos\left(\frac{\pi}{2} + \tilde{\Phi}\right)\right]\right) = \frac{1}{2} \left(\cos\left(\frac{\pi}{2} + \tilde{\Phi}\right) - \cos\left(\frac{\pi}{2} + \tilde{\Phi} + \Delta\right)\right) \\ &= \frac{1}{2} \left(\cos\left(\frac{\pi}{2} - \tilde{\Phi} - \Delta\right) - \cos\left(\frac{\pi}{2} - \tilde{\Phi}\right)\right) = \frac{1}{2} \left(\sin(\tilde{\Phi} + \Delta) - \sin(\tilde{\Phi})\right) \quad (11.2.6) \end{aligned}$$

Mit (11.2.6) ergibt sich somit der Quotient der beiden Wahrscheinlichkeiten für die beiden Kugelzonen zu A_1 und A_2 als:

$$\frac{P(\phi \in A_1)}{P(\phi \in A_2)} = \frac{\frac{1}{2}(\sin(\Phi_1 + \Delta) - \sin(\Phi_1))}{\frac{1}{2}(\sin(\Phi_2 + \Delta) - \sin(\Phi_2))} = \frac{2\pi(\sin(\Phi_1 + \Delta) - \sin(\Phi_1))}{2\pi(\sin(\Phi_2 + \Delta) - \sin(\Phi_2))} = \frac{F_1}{F_2}$$

Anders als bei der Verwendung von (11.2.1) sind hierbei die Wahrscheinlichkeiten für unterschiedliche Kugelzonen mit gleicher Bandbreite des Breitengrads nur dann identisch, wenn für die minimalen Breitengrade eine der beiden folgenden Bedingungen erfüllt ist:

$\Phi_1 = \Phi_2$: In diesem Fall gilt dann aber auch gleich $A_1 = A_2$.

$\Phi_2 = -\Phi_1 - \Delta$: Dies entspricht allerdings einer Symmetrie zwischen A_1 und A_2 mit dem Äquator als Spiegelebene.

Als Ergebnis bleibt dabei festzuhalten, dass beide Bedingungen jeweils auch gleich zu identischen Flächeninhalten der Kugelzonen führen, womit nachgewiesen ist, dass der für (11.2.1) geltende Fehler durch die in (11.2.2) gegebene Formeln zur Erzeugung behoben wurde.

11.3. Durchführung einer Rundreise

Auf der Oberfläche der Kugel kann nun nicht mehr wie bei Optimierungsproblem 7.1.1 der Rundreise auf der Kreislinie mit Richtungswechsel nur in zwei Richtungen gefahren werden oder wie in Optimierungsproblem 8.1.1 der gerichteten Variante sogar auf eine Richtung beschränkt werden, sondern es muss das ganze Spektrum der Winkel von 0 bis 2π berücksichtigt werden. Das heißt insbesondere aber auch, dass statt der Summe über die beiden Möglichkeiten der Bewegung, in und gegen den Uhrzeigersinn, bei der Kreislinie im Fall der Oberfläche der Kugel ein Integral über alle möglichen Winkel im Bogenmaß zwischen 0 und 2π notwendig wird.

Es gilt allerdings weiterhin, dass bei der Anwendung eines Algorithmus, welcher analog zu Algorithmus 8.1.2 das Prinzip des nächsten Nachbarn verwendet, eine effiziente Bewegung in Richtung ϕ mit Schrittweite τ zu einem bestimmten Reisepunkt nur dann möglich ist, wenn kein weiterer Reisepunkt eine kürzere Entfernung zur aktuellen Position besitzt. Dazu wird erst einmal ausgehend vom aktuellen Standort auf der Verbindung zum Mittelpunkt der Kugel eine Entfernung von $1 - \cos(\tau)$ abgemessen und in dieser Position die zur Verbindung senkrecht stehende Ebene F betrachtet. Damit der ausgewählte Reisepunkt die kürzeste Entfernung vom aktuellen Standort besitzt, darf in dem Inneren der durch die Ebene F und den aktuellen Standort festgelegten Kugelkappe kein anderer unbearbeiteter Reisepunkt enthalten sein. Dadurch müssen weitere vor dieser Bewegung ebenfalls schon vorhandene Reisepunkte definit in dem restlichen Bereich der Kugel liegen.

Der noch mögliche Platz für den Reisepunkt einer verbliebenen Anfrage schränkt sich, wie auch schon bei der Rundreise auf der Kreislinie, mit den Bewegungen seit der Bekanntgabe der Anfrage weiter ein. Allerdings ergibt eine Vereinigung von Kugelkappen nicht zwingend wieder ein „einfaches“ geometrisches Gebilde, und zudem kommt auch noch erschwerend eine Fülle von möglichen Bewegungsrichtungen hinzu. Eine weitere Komplikation ergibt sich auch daraus, dass der verbleibende Bereich für den Reisepunkt einer Anfrage nicht notwendigerweise zusammenhängend sein muss. Sei zum Beispiel bei einem Informationsvorlauf von zwei Anfragen die folgende Sequenz von Reisepunkten zu Anfragen

$$(\theta \quad \Phi), \left(\frac{1}{8}\pi \quad 0\right), \left(\frac{1}{4}\pi \quad 0\right), \dots, \left(\frac{1}{2}\pi \quad 0\right), \left(-\frac{7}{8}\pi \quad 0\right), \dots, \left(-\frac{1}{8}\pi \quad 0\right), (0 \quad 0)$$

gegeben. Wenn bei der Bearbeitung in den ersten 16 Schritten nacheinander die Reisepunkte $\left(\frac{i}{8}\pi \quad 0\right)$ für i von 1 bis 16 ausgewählt werden, so kann daraus geschlossen werden, dass der Reisepunkt $(\theta \quad \Phi)$ zu jedem dieser Reisepunkte mindestens einen Abstand von $\frac{1}{8}\pi$ hat. Da allerdings die Folge

$$(0 \quad 0), \left(\frac{1}{8}\pi \quad 0\right), \left(\frac{1}{4}\pi \quad 0\right), \dots, \left(\frac{1}{2}\pi \quad 0\right), \left(-\frac{7}{8}\pi \quad 0\right), \dots, \left(-\frac{1}{8}\pi \quad 0\right), (0 \quad 0)$$

eine Rundreise auf dem Äquator darstellt, wurde der mögliche Bereich für den verbliebenen Reisepunkt in zwei nicht verbundene Teile aufgespalten und ist daher nicht mehr zusammenhängend.

Aus diesen Gründen wird die explizite Aufstellung des für den Reisepunkt einer schon länger nicht bearbeiteten Anfrage noch möglichen Bereichs ziemlich unübersichtlich und damit die Berechnung der allgemeinen Schrittweite komplex. Theoretisch wäre eine erweiterte Anpassung der Rekursionsformel aus (8.5.21) über (9.5.4) hinaus zwar möglich, es müsste dann aber statt der Summe über den Bereich $[0, 2\pi]$ integriert werden. Jedoch ist die Transformation der Bereiche für die verbleibenden Reisepunkte ohne Rückgriff auf die Beschreibung mit den minimalen Abständen zu den ausgewählten Reisepunkten analog zu (9.5.1) nicht wirklich beschreibbar.

Empirische Analyse

In einer am Lehrstuhl Diskrete Mathematik, Optimierung und Operations Research von Professor Borgwardt betreuten Diplomarbeit [Haa09] analysiert Frau Haas mithilfe eines Programms rein empirisch die Rundreise auf der Oberfläche der Kugel für Sphären im drei und vier dimensionalen Raum. Grundlage dazu bilden die Daten eines Algorithmus nach dem Konzept „Nearest-Neighbor“ [Bor01, Kapitel 26.2] sowie auch eines Algorithmus unter Verwendung des Konzepts einer besten Einfügung.

- Der erste Algorithmus (ONN, für Online Nearest-Neighbor) sucht sich dabei immer diejenige Anfrage aus dem Informationsvorlauf als nächste zu bearbeitende Anfrage aus, deren Reisepunkt am nächsten zu dem aktuellen Standort des Algorithmus liegt.
- Dagegen arbeitet der zweite Algorithmus (OBE, für Online Beste Einfügung) mit einer komplett geplanten Rundreise über alle bereits bekannten Anfragen. Hierbei wird die Rundreise in zwei strukturell unterschiedlich zu behandelnde Teile aufgeteilt.

Auf der einen Seite besteht die geplante Rundreise aus einer Folge von schon bearbeiteten Anfragen, deren Reisepunkte nach der Bekanntgabe der zugehörigen Anfragen schon besucht worden sind. Diese Anfragen, unter denen sich auch die letzte bearbeitete Anfrage mit dem aktuellen Standort als Reisepunkt befindet, bilden den ersten Teil der Rundreise. Da alle Anfragen aus diesem Teil schon bearbeitet sind, darf deren Einordnung in die Rundreise nicht mehr verändert werden.

Der zweite Teil der Rundreise besteht aus einem Weg vom aktuellen Standort über alle Reisepunkte der Anfragen aus dem Informationsvorlauf zurück zum Ausgangspunkt der Rundreise. Anders als die Anfragen des ersten Teils sind die zum zweiten Teil gehörenden Anfragen noch nicht bearbeitet, das heißt, dass durch die Einfügung einer neuen Anfrage mit einem weiteren Reisepunkt, die Reihenfolge der Abarbeitung der im zweiten Teil enthaltenen Anfragen durchaus noch verändert werden darf.

Bei der Implementierung von Frau Haas wird die neu bekannt gewordenen Anfrage nach dem Prinzip „Nearest Addition“, wie in [Bor01, Abschnitt 26.2] oder auch [Jun08, Kapitel 15], eingefügt. Dazu wird auf dem Weg im zweiten Teil der Rundreise diejenige Verbindung zwischen zwei Reisepunkten der Anfragen gesucht, bei deren Ersetzung durch zwei Verbindungen zum neuen Reisepunkt die Länge der gesamten Rundreise am wenigsten zunimmt.

Als Modell für die nichtlineare Regressionsanalyse wird in der Arbeit αk^β verwendet, dabei sind α und β die zu bestimmenden Parameter und k steht für die Größe des Informationsvorlaufs. Als Grundlage für die Analyse dienen Berechnungen zu den beiden Algorithmen ONN und OBE mit folgenden Stellgrößen:

- Der Informationsvorlauf k variiert mit den folgenden Werten:

$$B := \{10, 20, 30, 40, 50, 100, 150, 200, 300, 400\}$$

- Die Anzahl der Anfragen n je Sequenz liegt in der Menge:

$$M := \{1.000, 5.000, 10.000, 20.000, 30.000, 40.000, 50.000\}$$

- Es werden immer 200 Dateien mit jeweils einer Sequenz von Anfragen bearbeitet.

Die Regressionsanalyse über die empirischen Daten, diese ergeben sich durch die Sammlung aller Berechnungsergebnisse über alle Werte $n \in M$, liefert die in Tabelle 11.3.1 zusammengestellten Schätzungen für die Parameter. Bei der Analyse wurden die Daten bereinigt, das heißt es wurden analog zu dem Vorgehen in Unterabschnitt 8.3.2 und Unterabschnitt 9.3.2 die ersten und letzten k Schrittweiten der jeweiligen Rundreise zu einer Sequenz von Anfragen nicht berücksichtigt.

	α	β
ONN	1,7202	-0,4692
OBE	2,2591	-0,4644

(a) Sphäre im drei dimensional
Raum

	α	β
ONN	1,5659	-0,3316
OBE	1,8863	-0,3249

(b) Sphäre im vier dimensional
Raum

Tabelle 11.3.1.: Ergebnisse der Regressionsanalyse für die betrachteten Sphären

Allerdings berechnen die funktionalen Modelle die numerische Schrittweite und nicht wie bei den Ergebnissen und Berechnungen in Kapitel 8 und Kapitel 9 den Vorfaktor zur Zahl π . Die Daten aus Tabelle 11.3.1 zur Regressionsanalyse bedeuten beispielsweise, dass beim Algorithmus ONN im drei dimensional Raum bei einem Informationsvorlauf von 10 Anfragen eine durchschnittliche Schrittweite von

$$1,7202 \cdot 10^{-0,4692} = 0,5839 \dots \approx 0,1858\pi$$

erwartet werden kann.

Aus den funktionalen Darstellungen der Regressionsanalyse kann nun festgestellt werden, dass der Algorithmus ONN sowohl im drei als auch im vier dimensional eine kleinere durchschnittliche Schrittweite erzeugt als der Algorithmus OBE, dies zeigt sich in den bei ONN kleineren Werten für α und β .

Beiden Algorithmen gemeinsam ist allerdings die Tatsache, dass die Parameter α der Funktionen zur Regression im vier dimensional Fall kleiner sind als die für die Sphäre im drei dimensional Raum, die Parameter β dagegen sind bei der funktionalen Darstellung für die Sphäre im vier dimensional Raum größer als beim drei dimensional Fall.

11. Ausblick auf die Rundreise auf der Oberfläche einer Kugel

Insgesamt ergibt sich, dass mit der Erhöhung der Dimension des Raumes um eins, von drei auf vier dimensional, die durchschnittliche Schrittweite zwischen zwei Punkten auf der Rundreise bei beiden Algorithmen zunimmt.

Teil IV.

Anhang

A. Implementierung der Programme

In diesem Kapitel sollen die einzelnen Implementierungen zum Teil III zusammengestellt und erläutert werden. Die Implementierungen beinhalten eine Realisierung der graphischen Darstellung der Beispiele in Abschnitt 8.2 und Abschnitt 9.2, sowie die Funktionalitäten zu den Berechnungen und Auswertungen für die empirischen Analysen in Abschnitt 8.3 und Abschnitt 9.3, als auch die Methodik für die diskreten Varianten der Rundreise auf der Kreislinie aus Kapitel 10.

A.1. Programm zur Simulation einer Rundreise auf der Kreislinie

Das im folgenden dargestellte Programm dient zur Visualisierung der Arbeitsweise der beiden in dieser Arbeit vorgestellten Algorithmen zur Bearbeitung der beiden Probleme, siehe Optimierungsproblem 7.1.1 und Optimierungsproblem 8.1.1 der Rundreise auf der Kreislinie. Die Steuerung des Programm erfolgt über ein JApplet, welches in der Klasse *Simulation* realisiert wird. Diese Abschnitt ist so aufgebaut, dass jeweils ein Paket mit seinen Klassen und Methoden dargestellt wird.

A.1.1. Das Steuerprogramm

Innerhalb des Applets zur Klasse *Simulation* gibt es grundsätzlich folgende Möglichkeiten zur Interaktion:

- Erstellung einer neuen zufälligen Sequenz von Anfragen mit einer Gleichverteilung der Reisepunkte auf der Kreislinie durch „New“.
- Speichern der aktuellen Sequenz aus Anfragen mit „Save“ in einer auszuwählenden Datei, dabei wird immer die Nummer der Anfrage und deren Reisepunkt als vielfaches von π gespeichert (da dabei die amerikanische Notation verwendet wird, also 0.5 für $\frac{1}{2}\pi$).

Ebenso kann eine solche Datei mit „Load“ auch wieder eingelesen werden. Dabei kann der Speicherort der Datei jeweils beliebig gewählt werden, da durch die Verwendung der Klasse *FileChooser* bei der Auswahl der Dateipfade die Möglichkeit zur Navigation innerhalb des Dateisystems gegeben ist.

- Durchführung des nächsten Schritts über die Schaltfläche „*Forward*“, das heißt eine Bewegung zur nächsten ausgewählten Anfrage, wobei die Problemstellung in dem Applet visualisiert wird. Bei Bedarf können über die Schaltfläche „*Back*“ auch wieder frühere Situationen der Bearbeitung abgerufen werden.
- Die Konfiguration der Problemstellung kann über „*Config*“ verändert werden, darunter fällt zum Beispiel sowohl die Anzahl der Anfragen innerhalb der Sequenz als auch die Größe des Informationsvorlaufs.

Weiterhin besteht hier die Möglichkeit zwischen den beiden Optimierungsproblemen umzuschalten, hierbei steht „one way“ für das Optimierungsproblem 8.1.1, also die gerichtete Rundreise auf der Kreislinie, und „all ways“ für die Rundreise auf der Kreislinie mit Richtungswechsel nach Optimierungsproblem 7.1.1.

Zur besseren Darstellung und größeren Übersichtlichkeit kann zusätzlich auch noch die Anzahl der äquidistant verteilten Punkte auf der Kreislinie, welche als mögliche Positionen für die Reisepunkte der Anfragen zur Verfügung stehen, verändert werden.

- Schließlich kann mit „*Export to LaTeX*“ die Bearbeitung der momentanen Sequenz aus Anfragen in der ausgewählten Problemstellung mit den eingestellten Parametern als $\text{\TeX}$ -Datei zur Darstellung mit $\text{\LaTeX}$ ausgegeben werden. Damit kann dann schließlich auch eine PDF-Datei mit den einzelnen Schritten der Rundreise auf der Kreislinie erzeugt werden. Allerdings ist dafür bei der Kompilierung wegen der Verwendung des $\text{\LaTeX}$ -Pakets *pstricks* nicht direkt `pdflatex` zu verwenden, sondern es muss ein Umweg über das Ghostscript-Format gemacht werden.

In der $\text{\TeX}$ -Datei sind außerdem die Daten zu den Reisepunkten der einzelnen Anfragen jedoch nicht als vielfache von π angegeben sondern in Grad zwischen 0 und 360, das liegt daran, dass $\text{\LaTeX}$ zum Zeichnen in Polarkoordinaten die Winkelangaben im Gradmaß benötigt.

Die in Abschnitt 8.2 und Abschnitt 9.2 enthaltenen Darstellungen zur Arbeitsweise des jeweiligen Algorithmus wurden mithilfe des Programms zur Simulation und der darin vorhandenen Export-Methode erzeugt.

Bei der Bedienung des Programms sind folgende Hinweise zu beachten:

- (i) Die Anzahl der Anfragen in der Sequenz beschränkt natürlicher Weise die Anzahl der insgesamt möglichen Schritte des Algorithmus.
- (ii) Es kann nicht weiter zurückgegangen werden als bis zur anfänglichen Situation mit keiner bearbeiteten Anfrage.
- (iii) Eine Bearbeitung, beziehungsweise Exportierung kann erst erfolgen, wenn entweder eine Sequenz von Anfragen neu erzeugt oder aus einer Datei geladen wurde.

- (iv) Eine Änderung der Konfiguration wirkt sich nicht mehr auf das aktuelle Beispiel aus, sondern nur auf danach neu erzeugte Sequenzen von Anfragen, dass heißt um eine Sequenz aus Anfragen zum Beispiel mit verschiedenen Werten des Informationsvorlaufs zu untersuchen, sollte diese in eine Datei abgespeichert werden und nach jeder Änderung der Konfiguration wieder neu eingelesen werden.

A.1.2. Das Paket tools

Im Paket Tools sind alle weiteren Klassen für das Programm zur Simulation neben der Klasse zur Steuerung vereinigt. Die einzelnen Aufgaben sind dabei wie folgt auf die Klassen aufgeteilt:

Anfrage: Hier sind die grundlegenden Informationen zu den einzelnen Anfragen hinterlegt.

Generator: Das ist die Klasse, welche sich um die Erzeugung der Sequenz von Anfragen mit gleichverteilten Reisepunkten kümmert.

Planung: In dieser Klasse findet sich alles, was mit der Planung der Rundreise zu tun hat, also zum Beispiel die Implementierung der beiden Algorithmen.

Drawer: Eine Hilfsklasse zur Zeichnung der Situation innerhalb eines Schrittes der Rundreise auf der Kreislinie.

Configurator: Hierbei handelt es sich um die Klasse, in welcher das Fenster zur Änderung der Parameter implementiert wurde.

Export: Diese Klasse kümmert sich um den Export der Daten, wie das Speichern und Laden von Sequenzen aus Anfragen und insbesondere auch um den Export der Darstellung der Rundreise auf der Kreislinie in eine TeX-Datei.

A.2. Programme zur Berechnung und Sammlung der empirischen Ergebnisse

Die Zusammensetzung der Programme zur Berechnung der empirischen Daten bei der Bearbeitung der Optimierungsprobleme 7.1.1 und 8.1.1 zur Rundreise auf der Kreislinie durch die Algorithmen 9.1.1 und 8.1.2 gestaltet sich schon vom Aufbau etwas komplexer als das Simulationsprogramm. Grundsätzlich sind dabei drei Programme mit den folgenden Aufgaben implementiert worden:

- Erzeugung der Anfragen

- Berechnung der Rundreisen
- Sammlung der Daten

A.2.1. Das Programm zur Erzeugung der Sequenzen der Anfragen

Das Programm **ErzeugePunkte** zur Erzeugung der verschiedenen Sequenzen von Anfragen, wobei die Reisepunkte der einzelnen Anfragen gleichverteilt auf der Kreislinie generiert werden, besteht hauptsächlich aus den beiden Klassen *ErzeugePunkte* und *Tools*.

In der Klasse *ErzeugePunkte* befinden sich die Methoden, welche zur Erzeugung einer Sequenz von Anfragen notwendig sind.

Die andere Klasse *Tools* beinhaltet eine Methode zur Findung der Position eines Musters aus „*“ in einem String und eine Umwandlung, welche zu den Zahlen $\{0, 1, \dots, 9\}$ die jeweilige Zeichendarstellung als Char liefert.

A.2.2. Das Programm zur Lösung der Instanzen der Problemstellung mit verschiedenen Werten für den Informationsvorlauf

Zur vereinfachten Steuerung des Programms **Rundreise** kann eine Parameter-Datei namens *RundReise_Param.log*, siehe Quellcode A.1, verwendet werden. In dieser Datei lassen sich folgende Parameter einstellen:

- (i) Die Anzahl der zu bearbeitenden Sequenzen von Anfragen.
- (ii) Die Anzahl der Anfragen in jeder Sequenz.
- (iii) Der Speicherort, an welchem die Daten der Berechnung gespeichert werden sollen. Da temporär jeweils für jede Sequenz von Anfragen eine eigene Datei mit der gewählten Rundreise angelegt wird, sollte innerhalb dieser Partition einiges an Speicherplatz vorhanden sein. Der Abschnitt „****“ steht dabei für die Position der Nummer der Sequenz.
- (iv) Eine Liste mit den verschiedenen zu untersuchenden Werten für den Informationsvorlauf.
- (v) Schlussendlich noch als 0/1 Variable die Festlegung ob die gerichtete Rundreise auf der Kreislinie oder die Rundreise auf der Kreislinie mit Richtungswechsel verwendet werden soll.

Durch die Verwendung dieser Parameter-Datei muss nicht jedes mal für eine Änderung an den Einstellungen des Programms der Quellcode neu kompiliert werden, sondern es

genügt völlig das ausführbare Programm einmal zu erzeugen und danach kann mithilfe verschiedener Versionen der Parameter-Datei das Programm wie gewünscht gesteuert werden.

Quellcode A.1: RundReise_Param.log

```
5 ### Parameter Datei zur Berechnung von Online-Touren für eine
   ↳ Sequenz von Punkten
10000 # Anzahl der Dateien mit Punktesequenzen <= 10000
../.. /BSP/test****.txt # Vorlage für die Namen der
   ↳ einzulesenden Dateien, muss **** enthalten.
50000 # Anzahl der Punkte pro Datei
/home/mtinkl/Promotion/Programme/Kreisrundreise/LSG/ # Speicherort
   ↳ für die Dateien mit den Auswertungen ./tmp/Prog
{3, 5, 6, 7, 8, 10, 12, 15, 17, 21, 24, 25, 27, 30, 34, 37, 42,
   ↳ 47, 52, 57, 62, 67, 72, 77, 82, 87, 92, 97, 102, 107, 112, 117,
   ↳ 122, 127, 132, 137, 142, 147, 152, 157, 162, 167, 172, 177,
   ↳ 182, 187, 192, 197, 202, 207, 212, 217, 222, 227, 232, 237,
   ↳ 242, 247, 252, 257, 262, 267, 272, 277, 282, 287} # Liste der
   ↳ Werte für die Vorschau #
0 # Soll die Reiserichtung festgelegt sein (0=nein)
```

Das Programm ist in mehrere Klassen unterteilt:

Punkt: Diese Klasse dient der Repräsentation des Reisepunktes einer einzelnen Anfrage mit deren relevanten Daten, dazu gehört in diesem Fall auch der Zustand der Bearbeitung.

Tools: Hier sind wieder die beiden Methoden aus der Klasse *Tools* des Programms **ErzeugePunkte** zu finden, ergänzt um eine Methode zur Erzeugung neuer Datei-Ordner (funktioniert nur unter Linux) und einer Methode zur Umwandlung einer Zeichenkette über verschiedene Zahlenwerte in einen Array mit den jeweiligen zugehörigen ganzzahligen Werten.

Planung: Dort finden sich hauptsächlich Funktionalitäten, welche mit der Planung der Rundreise auf der Kreislinie nach einem der beiden vorgestellten Algorithmen zu tun haben, wie die Erstellung einer Rundreise und die Ausgabe der dazugehörigen Daten in eine Datei.

Auswertung: Hierin finden sich Methoden zur Auswertung der vorher berechneten Ergebnisse. Eine ausführlichere Beschreibung der dabei entstehenden Dateien und deren Inhalte findet sich weiter unten.

Rundreise: Dies ist die Datei zur Steuerung des gesamten Ablauf des Programms. Sie kümmert sich um das Einlesen, und erstellt falls gewünscht eine Vorlage für die Parameter-Datei *RundReise_Param.log*, siehe Quellcode A.1.

Nach einem Durchlauf des Programms **Rundreise** sollten folgende Dateien und Verzeichnisse erzeugt worden sein:

- Für jeden Wert x aus der vorgegebenen Liste mit verschiedenen Werten des Informationsvorlaufs ein Verzeichnis Kx , – hier werden auch die Zwischenergebnisse in Form der einzelnen Rundreisen abgelegt, diese werden allerdings nach der Auswertung wieder gelöscht, – in diesen Verzeichnissen zu den einzelnen Werten des Informationsvorlauf sollten jeweils folgende Dateien vorhanden sein:
 - In *AVG.txt* sind gesammelt die durchschnittlichen Schrittweiten der Sequenzen von Anfragen, also für jede Sequenz deren Durchschnitt über alle Schrittweiten.
 - *RAVG.txt* ist analog zu *AVG.txt* aufgebaut nur das dort die bereinigten durchschnittlichen Schrittweiten aufgelistet sind. Das heißt, dass hier nicht mehr der Durchschnitt aller Schrittweiten verwendet wird, sondern wie in Unterabschnitt 8.3.2 eine Bereinigung vorgenommen wurde.
 - Die durchschnittlichen Schrittweiten der einzelnen Schritte genommen über alle Sequenzen der Anfragen finden sich schließlich in *DAVG.txt*.
- Im Hauptverzeichnis zur Berechnung entstehen zwei neue Dateien. Die erste dieser beiden Dateien *KAVG.txt* enthält für jeden Informationsvorlauf die durchschnittliche Schrittweite über alle Sequenzen von Anfragen und alle Schritte hinweg. In der zweiten Datei *KRAVG.txt* stehen die analogen Informationen wie in der ersten, nur dass hier wieder die betrachteten Schritte wie in Unterabschnitt 8.3.2 erläutert bereinigt wurden.

A.2.3. Das Programm zum Einsammeln der bei der Auswertung gewonnenen Daten

Da die Methoden zur Auswertung aus dem Programm **Rundreise** jeweils immer nur die aktuell neu berechneten Daten erfassen, wurde ein drittes Programm **Collect** zur Erfassung der Daten aus mehreren Durchläufen notwendig. Dieses besteht aus folgenden Klassen:

Collector: Der Klasse zur Steuerung der Auswertung.

Parameter: Eine Klasse zur Ausgabe und zum Einlesen der Parameter zur Einstellung der Funktionalitäten, diese werden wie in Quellcode A.2 dargestellt, in der Datei *Collect_Param.txt* abgelegt.

CollectAvg: Dieser Teil der Klasse *Collector* dient zur Erfassung der durchschnittlichen Daten für mehrere Werte des Informationsvorlaufs aus verschiedenen Verzeichnissen.

CollectSteps: Mit dem zweiten Teil der Methoden der Klasse *Collector* ist es möglich zu einzelnen Schritten *t* die durchschnittlichen Werte der Schrittweite in diesem Schritt für die verschiedenen Werte des Informationsvorlaufs in einer Datei zu sammeln. Diese Dateien werden in einem eigenen Unterordner *Steps* des Ordners für die Lösungen (zum Beispiel *LSG* oder *OLSG*) angelegt.

Gesteuert werden kann auch dieses Programm über eine Parameter-Datei, und zwar ist das diesmal die Datei *Collect_Param.txt*.

Quellcode A.2: *Collect_Param.txt*

```
5  ### Parameter Datei zur Sammlung von Daten aus Berechnungen zur
    ↳ Kreisrundreise
2222 # Type der Sammlung 2222 für die Durchschnittswerte und
    ↳ 1111 für die einzelnen Schritte mit größeren
    ↳ Informationsvorlauf 1122 für kleineren Informationsvorlauf
    ../../LSG/ # Ordner für den Arbeitsbereich.
{2, 4, 3, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17, 19, 20, 21, 22,
 ↳ 24, 25, 26, 27, 29, 30, 32, 34, 36, 37, 39, 42, 44, 47, 49, 52,
 ↳ 54, 57, 59, 62, 64, 67, 69, 72, 74, 77, 79, 82, 84, 87, 89, 92,
 ↳ 94, 97, 99, 102, 104, 107, 109, 112, 114, 117, 119, 122, 124,
 ↳ 127, 129, 132, 134, 137, 139, 142, 144, 147, 149, 152, 154,
 ↳ 157, 159, 162, 164, 167, 169, 172, 174, 177, 179, 182, 184,
 ↳ 187, 189, 192, 194, 197, 199, 202, 204, 207, 209, 212, 214,
 ↳ 217, 219, 222, 224, 227, 229, 232, 234, 237, 239, 242, 244,
 ↳ 247, 249, 252, 254, 257, 259, 262, 264, 267, 269, 272, 274,
 ↳ 277, 279, 282, 284, 287, 289} # Liste der Werte für die
    ↳ Vorschau
10 # Erster zu untersuchender Schritt
11 # Letzter zu untersuchender Schritt
```

Dabei können folgende Parameter verändert werden:

- (i) Die Art der Sammlung, eine genauere Erläuterung befindet sich in dem nachfolgenden Absatz.
- (ii) Der Speicherplatz der Dateien von vorherigen Rechendurchläufen.

- (iii) Eine Liste mit den verschiedenen Werten für den Informationsvorlauf.
- (iv) Die Anzahl der Schritte, welche untersucht werden sollen. Diese beiden Werte sind nur relevant, wenn der die durchschnittliche Schrittweite in einzelnen Schritten zusammengefasst werden soll.

Der Art der Sammlung legt fest, welche Daten gesammelt werden, infrage kommen folgende Möglichkeiten:

- Die durchschnittlichen Wegstrecke über aller Werte des Informationsvorlaufs, jeweils bereinigt und komplett über die ganze Rundreise. (Einstellung 2222)
- Für einen bestimmten Bereich der Schrittweiten, die Werte des jeweiligen Schrittes in Abhängigkeit vom Informationsvorlauf, wobei nur Werte von Berechnungen ausgewählt werden, bei denen der zur Berechnung verwendete Informationsvorlauf
 - kleiner gleich der Nummer des betrachteten Schrittes ist. (Einstellung 1122)
Dann wurden schon mehr Reisepunkte angefahren als Anfragen im Informationsvorlauf enthalten sind.
 - größer gleich der Nummer des betrachteten Schrittes ist. (Einstellung 1111)
Somit ist mit großer Wahrscheinlichkeit noch mindestens eine der Anfragen aus dem Informationsvorlauf bei Beginn der Rundreise noch nicht bearbeitet worden.

A.3. Bibliotheken zur diskreten Variante der Rundreise auf der Kreislinie

Für die in Abschnitt 10.3 betrachteten Ergebnisse der Berechnungen mit den Markovketten werden zwei, in Maple implementierte, Bibliotheken verwendet. Die erste der beiden Bibliotheken dient der Erzeugung aller möglichen Situationen für eine vorgegebene Größe des Informationsvorlaufs und einer ebenfalls vorgegebenen Anzahl an gleichmäßig auf der Kreislinie verteilten Positionen, der diskreten Variante der Rundreise auf der Kreislinie. Durch die zweite Bibliothek kann aus der Menge von Situationen mithilfe weiterer zur Verfügung gestellter Schnittstellen die erwartete Schrittweite für beide Arten der Rundreise auf der Kreislinie berechnet werden.

A.3.1. Die Bibliothek zur Erzeugung der wesentlichen Bestandteile

Die erste der beiden Bibliotheken, nämlich *CreateKreis* bündelt folgende aufrufbare Methoden:

- Über *anzahl* kann für eine Größe k des Informationsvorlaufs und eine Anzahl n an Positionen die Anzahl der möglichen Situationen nach (10.1.2) berechnet werden.

- Durch *computeNewRep* wird für eine gegebene Situation der nächste Reisepunkt ausgewählt und anschließend die Bewegung dorthin realisiert. Dazu wird mithilfe von (10.3.4) die repräsentative Darstellung für die neue Situation mit den verbliebenen $k - 1$ Reisepunkten erzeugt.
- Die Methode *createRepresents* dient der vorher angesprochenen Erzeugung aller möglichen Situationen für n Positionen und einen Informationsvorlauf von k Anfragen, es wird also die in Abschnitt 10.1 angesprochene Menge Ω erzeugt.
- Von *createStates* wird ein Array mit den möglichen Positionen und ein zweiter Array mit den Distanzen zum Nullpunkt erzeugt.
- Mit *prodTrafo* wird die Übergangsmatrix für die Markovkette nach Abschnitt 10.3 generiert.

A.3.2. Die Bibliothek mit den Methoden zur Berechnung der erwarteten Schrittweiten

Nachdem nun mit *CreateKreis* in Unterabschnitt A.3.1 eine Bibliothek zur Erzeugung der notwendigen Vorarbeiten zur Verfügung steht, wird zur Auswertung von diesen eine zweite Bibliothek verwendet. Für die Berechnung der erwarteten Schrittweiten steht die Bibliothek *AverageKreis* zur Verfügung, mit dieser kann über den Aufruf von

$$avg(k::integer, n::integer, one::boolean)$$

für einen Informationsvorlauf von k Anfragen und n auf der Kreislinie gleichmäßig verteilten Positionen die zugehörige zu erwartende Schrittweite bestimmt werden. Der zusätzliche logische Parameter dient dabei der Unterscheidung, ob die gerichtete Rundreise nach Optimierungsproblem 8.1.1, dann muss `true` als Wert gesetzt werden, oder die Rundreise auf der Kreislinie mit Richtungswechsel nach Optimierungsproblem 7.1.1, dann wird `false` übergeben, betrachtet werden soll. Zur Vereinfachung existieren außerdem noch die beiden Methoden

$$avgeinfach(k::integer, n::integer) \text{ und } avgnormal(k::integer, n::integer)$$

zum direkten Aufruf der Berechnungen für die jeweilige Problemstellung.

B. Empirische Daten

Nachdem nun in Anhang A die für die empirische Berechnungen notwendigen Programme vorgestellt wurden, sollen hier in diesem Kapitel, die wichtigsten der berechneten empirischen Daten aufgeführt werden.

Dabei ist die Struktur der folgenden Tabellen, mit $\bar{\tau}_i$ für die durchschnittliche Schrittweite bei der Bearbeitung der Sequenz i , wie folgt aufgebaut:

- In der ersten Spalte (IV) steht die Größe des Informationsvorlaufs gefolgt von der durchschnittlichen Schrittweite ($\emptyset \bar{\tau}_i$) über alle Sequenzen.
- Die minimal ($\min \bar{\tau}_i$) und maximal ($\max \bar{\tau}_i$) bei der Bearbeitung der einzelnen Sequenzen i erzielte durchschnittliche Schrittweite steht in der fünften und letzten Spalte.
- In der dritten (min Intervall) und vierten Spalte (max Intervall) finden sich schließlich noch der untere und obere Wert des jeweiligen Bereichs der durchschnittlichen Schrittweite für den Informationsvorlauf, in welchem 90 Prozent der empirisch berechneten durchschnittlichen Schrittweiten $\bar{\tau}_i$ der einzelnen Sequenzen liegen. Die restlichen 10 Prozent verteilen sich gleichmäßig darunter und darüber.

B.1. Daten für die gerichtete Rundreise auf der Kreislinie

Zuerst einmal sind in Tabelle B.1.1 die empirischen durchschnittlichen Schrittweiten des Algorithmus 8.1.2 für die gerichtete Rundreise, welche als Grundlage für die Auswertung in Unterabschnitt 8.3.1 dienen, gesammelt.

IV	$\emptyset \bar{\tau}_i$	min Intervall	max Intervall	$\min \bar{\tau}_i$	$\max \bar{\tau}_i$
2	0.500017	0.49787	0.50219	0.49535	0.50447
3	0.333349	0.331953	0.334753	0.329953	0.336513
4	0.250021	0.248955	0.251035	0.247635	0.252595
5	0.200032	0.199196	0.200876	0.198116	0.201916
6	0.166701	0.165997	0.167397	0.165117	0.168557
7	0.142889	0.142277	0.143517	0.141357	0.144277
8	0.125029	0.124518	0.125557	0.123838	0.126157

Tabelle B.1.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
9	0.111142	0.110678	0.111598	0.110078	0.112238
10	0.100036	0.099638	0.100478	0.099118	0.100958
12	0.0833705	0.0829983	0.0837183	0.0825583	0.0841983
14	0.0714641	0.0711586	0.0717586	0.0707586	0.0721186
15	0.0667021	0.0663987	0.0669987	0.0659987	0.0674387
16	0.0625362	0.0622788	0.0627987	0.0618788	0.0631187
17	0.0588597	0.0585988	0.0591188	0.0582388	0.0594788
19	0.05267	0.052439	0.0528789	0.052119	0.0531589
20	0.0500381	0.049839	0.050239	0.049599	0.050559
21	0.0476576	0.0474391	0.047879	0.0471991	0.048119
22	0.0454928	0.0452791	0.0456791	0.0450391	0.0459591
24	0.0417035	0.0415192	0.0418792	0.0412392	0.0420792
25	0.0400392	0.0398792	0.0401992	0.0396792	0.0403992
26	0.0384994	0.0383192	0.0386792	0.0381592	0.0388392
27	0.0370764	0.0369193	0.0372393	0.0367193	0.0374393
29	0.0345201	0.0343593	0.0346793	0.0341593	0.0349193
30	0.0333747	0.0332393	0.0335193	0.0330793	0.0337193
32	0.0312898	0.0311594	0.0314394	0.0309994	0.0315994
34	0.029452	0.0293194	0.0295594	0.0291594	0.0297194
36	0.0278183	0.0277194	0.0279194	0.0275194	0.0280794
37	0.0270679	0.0269595	0.0271995	0.0268395	0.0273595
39	0.0256811	0.0255595	0.0257995	0.0254395	0.0259195
42	0.0238503	0.0237595	0.0239595	0.0236395	0.0240795
44	0.0227689	0.0226795	0.0228795	0.0225595	0.0229995
47	0.0213174	0.0212396	0.0213996	0.0211196	0.0215196
49	0.0204496	0.0203596	0.0205196	0.0201996	0.0206396
52	0.0192715	0.0191996	0.0193596	0.0190796	0.0194796
54	0.0185605	0.0184796	0.0186396	0.0183996	0.0187996
57	0.0175857	0.0175196	0.0176796	0.0173997	0.0177596
59	0.0169914	0.0169197	0.0170797	0.0168397	0.0171997
62	0.0161707	0.0161197	0.0162397	0.0159997	0.0163197
64	0.0156664	0.0155997	0.0157197	0.0155197	0.0158397
67	0.014967	0.0149197	0.0150397	0.0148397	0.0151197
69	0.0145346	0.0144797	0.0145997	0.0143997	0.0146797
72	0.0139315	0.0138797	0.0139997	0.0137997	0.0140797
74	0.0135554	0.0134797	0.0135997	0.0134397	0.0136797
77	0.013029	0.0129597	0.0130797	0.0128797	0.0131597
79	0.0127011	0.0126397	0.0127597	0.0125597	0.0128397

Tabelle B.1.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie (wird fortgesetzt)

B.1. Daten für die gerichtete Rundreise auf der Kreislinie

IV	$\varnothing \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
82	0.0122377	0.0121998	0.0122798	0.0121198	0.0123598
84	0.0119472	0.0118798	0.0119998	0.0118398	0.0120798
87	0.0115367	0.0114798	0.0115998	0.0114398	0.0116398
89	0.0112785	0.0112398	0.0113198	0.0111598	0.0113998
92	0.0109123	0.0108798	0.0109598	0.0107998	0.0109998
94	0.0106809	0.0106398	0.0107198	0.0105998	0.0107998
97	0.0103522	0.0103198	0.0103998	0.0102798	0.0104398
99	0.0101441	0.0101198	0.0101998	0.0100398	0.0102398
102	0.00984732	0.0097998	0.0098798	0.0097598	0.0099598
104	0.00965813	0.00959981	0.00967981	0.00955981	0.0097598
107	0.00938927	0.00935981	0.00943981	0.00931981	0.00947981
109	0.00921734	0.00915982	0.00923982	0.00911982	0.00931981
112	0.008972	0.00891982	0.00899982	0.00887982	0.00907982
114	0.00881527	0.00875982	0.00883982	0.00871983	0.00887982
117	0.00859043	0.00855983	0.00863983	0.00847983	0.00867983
119	0.00844683	0.00839983	0.00847983	0.00835983	0.00851983
122	0.00823978	0.00819984	0.00827983	0.00815984	0.00831983
124	0.00810788	0.00807984	0.00815984	0.00803984	0.00815984
127	0.0079177	0.00787984	0.00795984	0.00783984	0.00799984
129	0.00779599	0.00775984	0.00783984	0.00771985	0.00787984
132	0.00761929	0.00759985	0.00763985	0.00755985	0.00767985
134	0.00750641	0.00747985	0.00751985	0.00743985	0.00755985
137	0.00734303	0.00731985	0.00735985	0.00727985	0.00739985
139	0.00723797	0.00719986	0.00727985	0.00715986	0.00731985
142	0.00708618	0.00703986	0.00711986	0.00699986	0.00715986
144	0.00698841	0.00695986	0.00699986	0.00691986	0.00703986
147	0.00684631	0.00679986	0.00687986	0.00675986	0.00691986
149	0.00675517	0.00671987	0.00679986	0.00667987	0.00679986
152	0.00662291	0.00659987	0.00663987	0.00655987	0.00667987
154	0.00653693	0.00651987	0.00655987	0.00647987	0.00659987
157	0.00641305	0.00639987	0.00643987	0.00635987	0.00647987
159	0.00633249	0.00631987	0.00635987	0.00627987	0.00639987
162	0.00621656	0.00619988	0.00623988	0.00615988	0.00627987
164	0.0061412	0.00611988	0.00615988	0.00607988	0.00619988
167	0.00603241	0.00599988	0.00603988	0.00595988	0.00607988
169	0.00596106	0.00591988	0.00599988	0.00591988	0.00599988
172	0.0058576	0.00583988	0.00587988	0.00579988	0.00591988
174	0.00579178	0.00575988	0.00579988	0.00571989	0.00583988

Tabelle B.1.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
177	0.00569342	0.00567989	0.00571989	0.00563989	0.00575988
179	0.00563159	0.00559989	0.00563989	0.00555989	0.00567989
182	0.00553832	0.00551989	0.00555989	0.00547989	0.00559989
184	0.00547882	0.00543989	0.00551989	0.00543989	0.00551989
187	0.00539292	0.00535989	0.00539989	0.00531989	0.00543989
189	0.00533422	0.00531989	0.00535989	0.00527989	0.00539989
192	0.00525109	0.0052399	0.00527989	0.0051999	0.00531989
194	0.00519901	0.0051599	0.0052399	0.0051599	0.0052399
197	0.00512019	0.0050799	0.0051599	0.0050799	0.0051599
199	0.00507067	0.0050399	0.0050799	0.0050399	0.0051199
202	0.0049957	0.0049599	0.0049999	0.0049599	0.0050399
204	0.00494753	0.0049199	0.0049599	0.0048799	0.0049999
207	0.00487606	0.0048399	0.0048799	0.0048399	0.0049199
209	0.00483044	0.0047999	0.0048399	0.0047599	0.0048799
212	0.00476077	0.0047599	0.0047999	0.00471991	0.0047999
214	0.00471779	0.00467991	0.00471991	0.00467991	0.0047599
217	0.00465051	0.00463991	0.00467991	0.00459991	0.00467991
219	0.00460815	0.00459991	0.00463991	0.00455991	0.00463991
222	0.004551	0.00451991	0.00455991	0.00451991	0.00459991
224	0.00451082	0.00447991	0.00451991	0.00447991	0.00455991
227	0.00444724	0.00443991	0.00447991	0.00439991	0.00447991
229	0.00440853	0.00439991	0.00443991	0.00435991	0.00443991
232	0.0043565	0.00431991	0.00435991	0.00431991	0.00439991
234	0.00431853	0.00427991	0.00431991	0.00427991	0.00435991
237	0.00426526	0.00423992	0.00427991	0.00419992	0.00431991
239	0.0042315	0.00419992	0.00423992	0.00419992	0.00427991
242	0.00417509	0.00415992	0.00419992	0.00411992	0.00419992
244	0.00414425	0.00411992	0.00415992	0.00411992	0.00419992
247	0.00408994	0.00407992	0.00411992	0.00403992	0.00411992
249	0.00406061	0.00403992	0.00407992	0.00399992	0.00407992
252	0.00400963	0.00399992	0.00403992	0.00395992	0.00403992
254	0.00398181	0.00395992	0.00399992	0.00395992	0.00403992
257	0.00393336	0.00391992	0.00395992	0.00387992	0.00395992
259	0.00390838	0.00387992	0.00391992	0.00387992	0.00395992
262	0.00386222	0.00383992	0.00387992	0.00383992	0.00387992
264	0.00383598	0.00379992	0.00383992	0.00379992	0.00387992
267	0.00379377	0.00375992	0.00379992	0.00375992	0.00383992
269	0.00376069	0.00375992	0.00375992	0.00371993	0.00379992

Tabelle B.1.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
272	0.00372029	0.00371993	0.00371993	0.00367993	0.00375992
274	0.00369078	0.00367993	0.00371993	0.00367993	0.00371993
277	0.00365165	0.00363993	0.00367993	0.00363993	0.00367993
279	0.00363315	0.00359993	0.00363993	0.00359993	0.00367993
282	0.00359479	0.00355993	0.00359993	0.00355993	0.00363993
284	0.00356228	0.00355993	0.00359993	0.00351993	0.00359993
287	0.00352439	0.00351993	0.00355993	0.00347993	0.00355993
289	0.00350816	0.00347993	0.00351993	0.00347993	0.00351993

Tabelle B.1.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie

Für die bereinigten durchschnittlichen Schrittweiten zur Analyse von Algorithmus 8.1.2 in Unterabschnitt 8.3.2 ergaben sich die in Tabelle B.1.2 aufgelisteten empirischen Ergebnisse.

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
2	0.500004	0.497846	0.502165	0.495338	0.504476
3	0.333327	0.331919	0.334723	0.32991	0.336515
4	0.249994	0.24894	0.251009	0.247634	0.252554
5	0.200003	0.199165	0.200862	0.198079	0.201888
6	0.166669	0.165971	0.167371	0.165075	0.16853
7	0.142857	0.142261	0.14347	0.141328	0.144255
8	0.124996	0.124472	0.125531	0.123797	0.126143
9	0.111108	0.110649	0.111578	0.11003	0.112178
10	0.100001	0.0995831	0.100432	0.0990856	0.100942
12	0.083334	0.0829756	0.0836876	0.0825199	0.0841837
14	0.0714267	0.0711257	0.0717329	0.0707271	0.0720801
15	0.0666642	0.0663796	0.0669505	0.0659736	0.06741
16	0.0624978	0.0622352	0.0627644	0.0618384	0.0630592
17	0.0588212	0.0585688	0.059071	0.0582288	0.0594474
19	0.0526313	0.052407	0.0528514	0.0521054	0.0531152
20	0.0499988	0.0497901	0.0502096	0.0495389	0.0505095
21	0.0476183	0.0474164	0.047823	0.0471726	0.0480858
22	0.0454532	0.0452581	0.0456462	0.0450135	0.0458978

Tabelle B.1.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
24	0.0416636	0.0414863	0.0418387	0.0412098	0.0420491
25	0.0399994	0.0398303	0.0401706	0.0396255	0.0403722
26	0.0384592	0.038296	0.038622	0.0380951	0.038811
27	0.037036	0.0368753	0.0371928	0.0366836	0.0373882
29	0.0344794	0.0343353	0.0346265	0.034139	0.0348573
30	0.0333337	0.0331935	0.0334768	0.0330461	0.0336659
32	0.0312487	0.0311159	0.0313802	0.0309513	0.031554
34	0.0294105	0.0292846	0.0295356	0.0291053	0.0296903
36	0.0277767	0.0276605	0.0278935	0.0274882	0.0280313
37	0.0270263	0.0269113	0.0271391	0.0267779	0.0272968
39	0.0256395	0.0255311	0.0257494	0.0254034	0.0258754
42	0.0238085	0.0237075	0.023909	0.0235778	0.0240278
44	0.0227271	0.0226299	0.0228234	0.0225136	0.0229495
47	0.0212751	0.0211828	0.0213662	0.0210696	0.0214785
49	0.0204073	0.0203205	0.0204927	0.020161	0.0206023
52	0.0192289	0.0191466	0.0193106	0.0190419	0.0194267
54	0.0185178	0.0184391	0.0185965	0.0183488	0.0187402
57	0.0175429	0.0174688	0.0176182	0.0173662	0.0177148
59	0.0169483	0.0168768	0.0170202	0.0168088	0.0171567
62	0.0161276	0.0160584	0.0161965	0.0159749	0.0162782
64	0.0156233	0.0155567	0.0156895	0.0154751	0.0157741
67	0.0149238	0.0148598	0.0149871	0.0147788	0.0150696
69	0.0144915	0.0144314	0.0145521	0.0143453	0.0146312
72	0.0138882	0.013829	0.0139476	0.0137445	0.0140214
74	0.0135121	0.0134543	0.0135702	0.0133972	0.0136547
77	0.0129855	0.0129293	0.013042	0.0128375	0.0131208
79	0.0126575	0.0126028	0.0127125	0.0125288	0.0127829
82	0.0121938	0.0121423	0.012246	0.0120799	0.0123117
84	0.0119036	0.0118535	0.0119546	0.0117854	0.0120307
87	0.0114931	0.0114433	0.0115426	0.0113784	0.0115953
89	0.0112347	0.0111874	0.0112828	0.0111111	0.0113465
92	0.0108684	0.0108217	0.0109152	0.0107425	0.0109711
94	0.0106372	0.010592	0.0106824	0.0105367	0.0107408
97	0.0103083	0.0102645	0.0103523	0.0102201	0.010413
99	0.0101002	0.0100577	0.0101429	0.0100019	0.0102021
102	0.00980321	0.00976109	0.00984504	0.00970819	0.00991182
104	0.00961423	0.00957251	0.00965513	0.00950529	0.00971536
107	0.00934494	0.00930447	0.00938545	0.0092609	0.00944179

Tabelle B.1.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
109	0.00917329	0.00913496	0.00921144	0.00908192	0.00926393
112	0.00892782	0.00889031	0.00896568	0.00883751	0.00901513
114	0.00877107	0.00873373	0.00880915	0.00867369	0.00885041
117	0.00854603	0.00850928	0.00858174	0.00845689	0.00864064
119	0.00840246	0.00836722	0.00843792	0.00831652	0.00848863
122	0.0081954	0.00816019	0.00823031	0.00811813	0.00827244
124	0.00806343	0.00802945	0.00809818	0.0079929	0.00813658
127	0.00787317	0.00783884	0.00790681	0.00779335	0.00795504
129	0.00775124	0.00771841	0.00778466	0.00768067	0.00782497
132	0.00757466	0.00754214	0.00760687	0.00750014	0.0076463
134	0.00746161	0.00742973	0.00749415	0.00739115	0.00752967
137	0.00729835	0.00726709	0.00732957	0.00722129	0.00736598
139	0.00719308	0.0071626	0.00722409	0.00712555	0.00729796
142	0.00704158	0.00701172	0.00707147	0.00696456	0.00710575
144	0.00694341	0.00691344	0.00697335	0.00688319	0.00700598
147	0.00680183	0.00677302	0.00683049	0.00672669	0.00686883
149	0.00671015	0.00668194	0.00673859	0.00664076	0.00677216
152	0.00657803	0.00655015	0.00660557	0.0065019	0.00664643
154	0.0064925	0.00646549	0.00652007	0.00642678	0.00656087
157	0.00636861	0.00634192	0.00639554	0.00630432	0.00643068
159	0.00628811	0.00626175	0.00631523	0.00622734	0.00636502
162	0.00617187	0.00614574	0.00619799	0.00611661	0.00623296
164	0.00609632	0.00607083	0.00612188	0.00603332	0.0061519
167	0.00598684	0.00596138	0.00601225	0.00592063	0.00604405
169	0.00591632	0.00589098	0.00594203	0.00586018	0.00597303
172	0.00581283	0.0057879	0.00583766	0.00576141	0.00587019
174	0.00574616	0.00572173	0.00577063	0.0056889	0.00580943
177	0.00564898	0.00562466	0.00567325	0.00558667	0.00570613
179	0.00558554	0.00556194	0.0056096	0.00553239	0.00564267
182	0.0054934	0.00547012	0.00551714	0.00544144	0.00555395
184	0.00543363	0.00541043	0.0054567	0.00538272	0.00548652
187	0.00534665	0.00532353	0.00536924	0.0052859	0.00539699
189	0.00529001	0.00526738	0.00531242	0.00523465	0.00535295
192	0.00520735	0.0051854	0.00522935	0.00516018	0.00525568
194	0.00515345	0.00513171	0.00517521	0.00510451	0.00520285
197	0.00507511	0.00505335	0.00509686	0.00502875	0.00512139
199	0.00502404	0.0050032	0.00504531	0.00497853	0.0050785
202	0.0049493	0.00492799	0.00497022	0.00490082	0.00499694

Tabelle B.1.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
204	0.00490085	0.00487934	0.00492138	0.00484631	0.00495497
207	0.0048296	0.00480893	0.00485015	0.00477722	0.0048786
209	0.0047834	0.0047626	0.00480404	0.00473308	0.00483165
212	0.00471587	0.00469579	0.00473601	0.00467151	0.00476249
214	0.00467167	0.00465183	0.00469146	0.00462317	0.0047196
217	0.00460724	0.00458769	0.00462665	0.00455174	0.00465287
219	0.00456513	0.00454571	0.00458415	0.00451433	0.00461104
222	0.0045033	0.00448368	0.00452231	0.00446118	0.00454202
224	0.00446325	0.00444446	0.00448193	0.00441961	0.00450798
227	0.0044043	0.00438532	0.00442336	0.00436139	0.00445073
229	0.00436578	0.00434737	0.00438455	0.00432314	0.0044091
232	0.0043093	0.00429039	0.00432782	0.00426537	0.0043515
234	0.00427233	0.00425396	0.00429042	0.00423086	0.0043158
237	0.00421826	0.00419984	0.00423611	0.00417173	0.00425517
239	0.00418316	0.00416496	0.00420098	0.00414184	0.00422374
242	0.00413125	0.00411363	0.00414881	0.00408865	0.00417078
244	0.00409748	0.00408005	0.00411482	0.00405707	0.00413542
247	0.00404755	0.00403018	0.00406476	0.00400741	0.00408614
249	0.00401491	0.0039979	0.00403222	0.00397478	0.00405352
252	0.00396724	0.00395037	0.00398443	0.00393117	0.00400975
254	0.00393567	0.00391911	0.0039526	0.00389783	0.00398383
257	0.00389006	0.0038735	0.00390668	0.00385428	0.00392398
259	0.00385996	0.00384337	0.00387669	0.00382238	0.00389934
262	0.00381577	0.00379936	0.00383203	0.00377837	0.00385123
264	0.00378712	0.00377096	0.00380351	0.00374882	0.00382516
267	0.00374428	0.00372853	0.00376055	0.00370963	0.00378101
269	0.00371635	0.0037006	0.00373187	0.00368141	0.00375008
272	0.00367544	0.00365977	0.00369122	0.00363923	0.00371266
274	0.00364856	0.00363315	0.0036644	0.00361361	0.00368758
277	0.00360903	0.00359337	0.00362428	0.0035743	0.003648
279	0.00358325	0.00356797	0.00359839	0.00354726	0.00361943
282	0.00354489	0.00352983	0.00356004	0.00350726	0.00358091
284	0.00352016	0.00350499	0.00353524	0.00348631	0.00355273
287	0.00348325	0.00346843	0.0034982	0.00344878	0.00351577
289	0.00345907	0.00344418	0.0034738	0.00342695	0.00349477

Tabelle B.1.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der gerichteten Rundreise auf der Kreislinie

B.2. Daten für die Rundreise auf der Kreislinie mit Richtungswechsel

Bei der Rundreise auf der Kreislinie mit Richtungswechsel ergaben sich nach der Bearbeitung mit Algorithmus 9.1.1 für die in Unterabschnitt 9.3.1 vorgenommene Analyse die in Tabelle B.2.1 gesammelten empirischen Werte für die durchschnittliche Schrittweite,

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
2	0.33333	0.331963	0.334686	0.330073	0.336483
3	0.255829	0.254808	0.256865	0.25338	0.258095
4	0.208796	0.207941	0.209666	0.206795	0.211001
5	0.176654	0.175892	0.177417	0.174825	0.178282
6	0.15314	0.152458	0.153824	0.151635	0.154846
7	0.135095	0.13447	0.135723	0.133685	0.136505
8	0.120806	0.120218	0.121395	0.119418	0.122071
9	0.109189	0.108647	0.109745	0.107755	0.110652
10	0.0995636	0.0990337	0.100089	0.0982583	0.100861
12	0.0845304	0.0840554	0.0850079	0.08345	0.0856464
14	0.0733225	0.0728983	0.0737544	0.0723012	0.0743476
15	0.0687379	0.06832	0.0691596	0.0677954	0.0696916
16	0.0646743	0.0642695	0.0650685	0.0637833	0.0654916
17	0.0610451	0.0606595	0.0614362	0.0602425	0.0619319
19	0.0548495	0.0544872	0.0552178	0.0539643	0.0558105
20	0.0521893	0.0518392	0.0525437	0.0513822	0.0529864
21	0.0497624	0.0494293	0.0500902	0.0489127	0.0505253
22	0.0475466	0.0472205	0.0478775	0.0468239	0.048403
24	0.0436376	0.0433295	0.043954	0.042856	0.0443681
25	0.0419117	0.0416129	0.0422146	0.0412743	0.0426375
26	0.0403074	0.040016	0.0406027	0.0396523	0.0410781
27	0.0388208	0.038533	0.039108	0.0381627	0.0394448
29	0.0361449	0.0358742	0.0364172	0.0355184	0.0368775
30	0.0349376	0.0346766	0.0352086	0.0343046	0.0355549
32	0.0327437	0.0324891	0.0330038	0.0322122	0.0333547
34	0.0308058	0.0305652	0.0310489	0.0302861	0.0313846
36	0.0290726	0.0288426	0.0293106	0.0285458	0.0296161
37	0.0282746	0.0280507	0.0285049	0.0277254	0.0288837
39	0.0268038	0.0265876	0.0270243	0.0263205	0.0273232
42	0.0248611	0.0246606	0.0250675	0.0244346	0.0253936
44	0.0237111	0.0235149	0.0239096	0.0232556	0.0242331
47	0.0221702	0.0219891	0.0223578	0.0218031	0.0227594

Tabelle B.2.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
49	0.0212451	0.0210712	0.0214226	0.0208234	0.021636
52	0.0199959	0.0198301	0.0201672	0.0196149	0.0203745
54	0.0192405	0.0190773	0.0194093	0.0189052	0.0196711
57	0.0182065	0.0180511	0.0183648	0.0178753	0.018537
59	0.0175731	0.0174275	0.0177234	0.0172543	0.0179266
62	0.0167053	0.016564	0.0168536	0.0164091	0.0170519
64	0.0161724	0.016035	0.0163138	0.0158773	0.0165184
67	0.0154297	0.0152987	0.0155634	0.015148	0.0157453
69	0.0149738	0.0148471	0.0151032	0.0146696	0.0152824
72	0.0143363	0.0142142	0.0144648	0.0140792	0.0146396
74	0.0139407	0.013826	0.014064	0.013683	0.01425
77	0.013385	0.0132702	0.0135028	0.0131225	0.0136377
79	0.0130382	0.0129265	0.0131553	0.0127912	0.0133313
82	0.012551	0.0124443	0.0126626	0.0123292	0.012857
84	0.0122455	0.0121435	0.012355	0.0120274	0.0125388
87	0.0118144	0.0117153	0.0119187	0.0116049	0.0120722
89	0.0115447	0.0114478	0.0116474	0.0113325	0.0118277
92	0.0111594	0.0110652	0.0112596	0.0109581	0.0114
94	0.0109176	0.0108266	0.0110156	0.0107273	0.0111563
97	0.010574	0.0104838	0.0106694	0.0103774	0.010844
99	0.0103564	0.0102689	0.0104472	0.0101779	0.010597
102	0.0100459	0.00996152	0.0101338	0.00987215	0.0102737
104	0.00984878	0.00976718	0.00993608	0.00969109	0.0100341
107	0.00956768	0.00948817	0.00965115	0.0094047	0.0097563
109	0.00938977	0.00931145	0.00947138	0.00922186	0.00961091
112	0.0091331	0.00905868	0.00921455	0.00897961	0.00934769
114	0.00897057	0.00889605	0.00904983	0.00880482	0.00917534
117	0.00873763	0.00866601	0.00881564	0.00858973	0.0089373
119	0.00858792	0.00851792	0.00866264	0.00844471	0.00877798
122	0.00837363	0.0083045	0.00844781	0.00822644	0.00857656
124	0.00823579	0.00816877	0.00830727	0.00810403	0.00842576
127	0.0080391	0.00797348	0.0081108	0.0078916	0.00822651
129	0.00791279	0.00784913	0.00798217	0.00777437	0.00806684
132	0.00773056	0.00766889	0.00779823	0.00760945	0.00791043
134	0.00761296	0.00755275	0.00767976	0.00746547	0.0077871
137	0.00744448	0.00738592	0.00751081	0.00732229	0.00759691
139	0.00733515	0.00727736	0.0073984	0.00720922	0.00752651
142	0.00717869	0.00712145	0.00724109	0.00706795	0.00736132

Tabelle B.2.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel (wird fortgesetzt)

IV	$\varnothing \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
144	0.00707797	0.00702148	0.00714084	0.0069594	0.0072416
147	0.0069314	0.00687758	0.00699124	0.00682087	0.00710768
149	0.00683758	0.0067844	0.0068965	0.00673394	0.00699697
152	0.0067009	0.00664787	0.00676034	0.00656876	0.00687843
154	0.00661322	0.0065607	0.00667061	0.00651362	0.00678653
157	0.00648528	0.00643359	0.00654185	0.00637872	0.00662985
159	0.006403	0.00635284	0.0064578	0.00630575	0.00654589
162	0.00628273	0.00623382	0.00633572	0.00617849	0.00641678
164	0.00620556	0.00615718	0.00626031	0.00610658	0.00635091
167	0.00609297	0.00604604	0.00614508	0.0059972	0.00623151
169	0.00602031	0.00597391	0.00607093	0.00592145	0.0061354
172	0.00591458	0.0058689	0.00596448	0.00581492	0.00605503
174	0.00584549	0.00580036	0.00589456	0.00574557	0.00597619
177	0.00574594	0.00570273	0.00579353	0.00566042	0.00586947
179	0.00568136	0.00563697	0.00573006	0.00558413	0.00581822
182	0.00558686	0.00554506	0.00563485	0.00549638	0.00569996
184	0.00552583	0.00548256	0.0055723	0.00544642	0.00564573
187	0.00543666	0.00539465	0.00548325	0.00535419	0.0055568
189	0.00537892	0.00533982	0.00542427	0.00529691	0.00550532
192	0.00529441	0.00525575	0.00533829	0.00520999	0.00544551
194	0.00523875	0.00519875	0.00528317	0.00515422	0.00536465
197	0.00515876	0.00512046	0.0052017	0.0050852	0.00528309
199	0.0051064	0.00506655	0.00514952	0.00502751	0.0052188
202	0.00503047	0.0049919	0.00507274	0.00495755	0.00515105
204	0.00498075	0.00494191	0.00502175	0.00490849	0.00508246
207	0.00490786	0.00487047	0.00494818	0.00483185	0.0050032
209	0.00486105	0.00482328	0.00490216	0.00478436	0.00496648
212	0.00479218	0.00475847	0.00483206	0.00472486	0.00489929
214	0.00474694	0.00471333	0.00478573	0.00467263	0.0048539
217	0.00468115	0.00464759	0.00472089	0.0046128	0.00478326
219	0.00463842	0.00460584	0.00467649	0.00456874	0.00475534
222	0.00457546	0.00453997	0.00461301	0.00451048	0.00466685
224	0.00453506	0.00450041	0.00457217	0.00447547	0.00462358
227	0.00447425	0.00444334	0.00451103	0.00440859	0.00456079
229	0.0044347	0.00440299	0.00447117	0.0043727	0.00452932
232	0.0043777	0.00434758	0.00441323	0.00431439	0.00446622
234	0.00434054	0.00431251	0.00437561	0.00427781	0.00442394
237	0.00428546	0.00425207	0.00432076	0.00422962	0.00438204

Tabelle B.2.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel (wird fortgesetzt)

IV	$\varnothing \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
239	0.0042497	0.00421876	0.00428439	0.0041913	0.00435445
242	0.00419667	0.00416285	0.00423068	0.00413804	0.0042866
244	0.00416221	0.00412957	0.00419545	0.00410898	0.00427109
247	0.00411095	0.00407939	0.00414433	0.00405335	0.00419798
249	0.00407826	0.00404529	0.00411145	0.00402657	0.00417164
252	0.00402903	0.00399781	0.00406239	0.00396925	0.00411971
254	0.00399846	0.00396713	0.00403018	0.00394519	0.00408077
257	0.00395127	0.00391957	0.00398311	0.00390172	0.00404194
259	0.00392198	0.00389413	0.00395368	0.00386619	0.00400516
262	0.00387692	0.00384638	0.00390799	0.00382414	0.00395242
264	0.00384679	0.00382489	0.00387826	0.00378871	0.00395014
267	0.00380378	0.00378132	0.00383389	0.00374782	0.0038884
269	0.00377438	0.00374977	0.00380632	0.00372968	0.00385448
272	0.00373264	0.003709	0.00376363	0.00368315	0.00380988
274	0.00370657	0.00367551	0.00373589	0.00365752	0.00378425
277	0.00366678	0.00363671	0.00369567	0.00361461	0.00375164
279	0.00364049	0.00362052	0.00366941	0.00358858	0.00372449
282	0.00360163	0.00358281	0.0036307	0.00355078	0.00369904
284	0.00357481	0.00354822	0.00360416	0.00352942	0.00365038
287	0.0035383	0.00351001	0.00356817	0.00349773	0.00361699
289	0.00351523	0.00349453	0.00354299	0.00346132	0.00358952

Tabelle B.2.1.: Empirische Daten zur durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel

Die bereinigten Werte für die empirische Auswertung in Unterabschnitt 9.3.2 zur Bearbeitung der Rundreise auf der Kreislinie mit Richtungswechsel durch Algorithmus 9.1.1 stehen in Tabelle B.2.2.

IV	$\varnothing \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
2	0.333327	0.331961	0.334685	0.330062	0.336478
3	0.255821	0.254799	0.256858	0.253366	0.258077
4	0.208784	0.207929	0.209656	0.206794	0.210997
5	0.17664	0.17588	0.177403	0.174825	0.178262
6	0.153125	0.152445	0.15381	0.151608	0.154829
7	0.135078	0.134455	0.135704	0.13365	0.136499

Tabelle B.2.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
8	0.120788	0.120202	0.121376	0.119394	0.12205
9	0.10917	0.108628	0.10973	0.107734	0.110631
10	0.0995436	0.0990137	0.100069	0.0982372	0.100841
12	0.0845089	0.0840331	0.0849855	0.083442	0.0856401
14	0.0732997	0.0728754	0.0737309	0.0722846	0.0743106
15	0.0687146	0.0682997	0.069135	0.0677736	0.0696648
16	0.0646506	0.064248	0.0650442	0.0637521	0.0654722
17	0.0610209	0.0606364	0.0614107	0.0602191	0.0619066
19	0.0548244	0.0544598	0.0551936	0.0539306	0.0557932
20	0.0521639	0.0518144	0.0525171	0.0513592	0.0529713
21	0.0497369	0.0494034	0.0500667	0.0488734	0.050502
22	0.0475205	0.0471948	0.0478534	0.0467981	0.0483829
24	0.0436111	0.0433024	0.0439273	0.0428154	0.0443378
25	0.0418849	0.0415871	0.0421884	0.0412498	0.0426126
26	0.0402802	0.039988	0.0405752	0.0396267	0.0410538
27	0.0387934	0.0385062	0.0390795	0.0381268	0.0394225
29	0.036117	0.0358478	0.0363895	0.0354832	0.0368523
30	0.0349095	0.0346495	0.0351797	0.0342835	0.0355224
32	0.0327152	0.0324602	0.0329755	0.0321885	0.033318
34	0.0307769	0.0305362	0.0310202	0.0302464	0.0313658
36	0.0290435	0.0288138	0.0292828	0.0285164	0.0295961
37	0.0282455	0.028022	0.0284756	0.0277031	0.0288593
39	0.0267744	0.0265579	0.0269951	0.0262948	0.0272924
42	0.0248312	0.024631	0.0250378	0.0244072	0.0253714
44	0.0236811	0.0234827	0.023881	0.0232293	0.0241933
47	0.0221397	0.0219603	0.0223275	0.0217624	0.0227366
49	0.0212144	0.0210414	0.0213912	0.0207852	0.0215965
52	0.019965	0.0197986	0.0201372	0.019587	0.0203411
54	0.0192094	0.0190473	0.0193777	0.0188746	0.0196454
57	0.0181751	0.01802	0.0183323	0.0178513	0.0185086
59	0.0175417	0.0173959	0.017691	0.0172223	0.0179006
62	0.0166736	0.0165315	0.0168218	0.0163741	0.0170317
64	0.0161406	0.0160046	0.0162826	0.0158557	0.0164819
67	0.0153977	0.015267	0.0155325	0.0151146	0.0157156
69	0.0149418	0.0148157	0.0150712	0.014616	0.0152623
72	0.0143041	0.0141828	0.014432	0.014043	0.0146148
74	0.0139084	0.0137943	0.0140319	0.0136505	0.014216
77	0.0133525	0.013238	0.0134702	0.013089	0.0136175

Tabelle B.2.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel (wird fortgesetzt)

IV	$\emptyset \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
79	0.0130057	0.0128954	0.0131217	0.012758	0.0133012
82	0.0125184	0.0124126	0.0126291	0.012295	0.0128186
84	0.0122127	0.0121104	0.0123222	0.0120033	0.0125059
87	0.0117816	0.0116831	0.0118857	0.0115758	0.0120381
89	0.0115118	0.0114153	0.0116149	0.0113059	0.0117879
92	0.0111263	0.0110327	0.0112253	0.0109182	0.0113645
94	0.0108844	0.0107936	0.0109833	0.010691	0.0111244
97	0.0105407	0.0104511	0.0106355	0.0103498	0.0108012
99	0.0103233	0.0102363	0.0104135	0.0101416	0.010559
102	0.0100125	0.00992882	0.0101001	0.00983177	0.0102363
104	0.00981538	0.00973432	0.00990144	0.00965356	0.0100004
107	0.0095341	0.00945578	0.00961744	0.00936075	0.00972259
109	0.00935611	0.00927804	0.0094386	0.00919342	0.0095858
112	0.0090995	0.00902588	0.00918185	0.00894882	0.00931526
114	0.0089369	0.00886274	0.00901609	0.0087786	0.00914067
117	0.00870393	0.00863275	0.00878105	0.00856346	0.0089121
119	0.0085541	0.00848443	0.00862885	0.00841665	0.00873491
122	0.00833984	0.00827171	0.00841269	0.00819398	0.00853049
124	0.00820188	0.00813542	0.00827288	0.00807752	0.00840055
127	0.00800514	0.00793981	0.00807633	0.00786047	0.00819352
129	0.00787872	0.00781584	0.00794722	0.00775396	0.00803493
132	0.00769657	0.00763545	0.00776312	0.00757114	0.00788006
134	0.00757875	0.00751865	0.00764543	0.00743941	0.00775981
137	0.00741034	0.00735213	0.00747634	0.00729124	0.00756995
139	0.0073009	0.0072438	0.00736357	0.0071696	0.00748726
142	0.00714451	0.00708787	0.00720627	0.00702382	0.00732383
144	0.00704377	0.00698894	0.00710505	0.00692828	0.0072046
147	0.00689718	0.00684417	0.00695744	0.0067956	0.00707968
149	0.00680319	0.00675016	0.00686174	0.00670604	0.00695974
152	0.0066665	0.00661453	0.00672547	0.00653269	0.00684874
154	0.00657889	0.00652703	0.00663583	0.00648062	0.00675381
157	0.00645095	0.00640034	0.0065068	0.00634631	0.00659021
159	0.00636856	0.00631918	0.00642326	0.00627079	0.00651792
162	0.00624825	0.00620006	0.00630182	0.00614394	0.00639321
164	0.0061712	0.00612357	0.00622533	0.00607207	0.00631522
167	0.00605848	0.00601241	0.00611003	0.00597175	0.00619745
169	0.0059858	0.00594	0.00603615	0.00589678	0.00610675
172	0.00587997	0.00583478	0.00592913	0.00578611	0.00601284

Tabelle B.2.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel (wird fortgesetzt)

IV	$\varnothing \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
174	0.00581091	0.00576641	0.00585924	0.0057163	0.00595259
177	0.00571111	0.00566811	0.00575817	0.00563103	0.00583371
179	0.00564675	0.00560421	0.00569537	0.00555978	0.00577916
182	0.00555217	0.00551011	0.00559887	0.00546783	0.00566364
184	0.00549108	0.00544996	0.00553664	0.00540891	0.00560493
187	0.00540199	0.00536232	0.00544715	0.0053188	0.00551552
189	0.00534412	0.00530501	0.00538853	0.00526244	0.00546912
192	0.00525954	0.00522008	0.00530266	0.00518473	0.00540535
194	0.00520392	0.00516529	0.00524741	0.00511968	0.00534158
197	0.00512389	0.00508639	0.00516582	0.00504586	0.00525155
199	0.0050716	0.00503498	0.00511403	0.00499806	0.00517266
202	0.00499548	0.00495979	0.00503707	0.00491984	0.00510327
204	0.00494592	0.00491041	0.00498591	0.0048768	0.00504161
207	0.00487278	0.00483734	0.00491245	0.00480273	0.00497697
209	0.00482604	0.00479169	0.00486662	0.00475053	0.00492755
212	0.00475712	0.00472274	0.00479624	0.00468997	0.00486316
214	0.00471166	0.00467862	0.00474992	0.00462987	0.00481338
217	0.00464635	0.00461276	0.00468506	0.0045825	0.00474397
219	0.00460359	0.00457091	0.00464063	0.00454152	0.00472415
222	0.00454021	0.00450839	0.00457705	0.00447866	0.00463528
224	0.00449976	0.00446847	0.00453611	0.00444305	0.00459035
227	0.00443958	0.00440944	0.00447524	0.00437607	0.00452704
229	0.00439997	0.00436898	0.0044351	0.00434399	0.00449552
232	0.00434235	0.00431198	0.00437681	0.00427303	0.00443323
234	0.00430521	0.00427529	0.00433941	0.00424912	0.00439076
237	0.00425004	0.00422105	0.00428417	0.00419355	0.0043378
239	0.00421391	0.00418491	0.00424737	0.0041579	0.00431209
242	0.00416161	0.00413339	0.00419456	0.00409945	0.00424651
244	0.00412664	0.00409858	0.00415969	0.00406976	0.00423549
247	0.00407619	0.0040488	0.00410786	0.00401933	0.00416879
249	0.00404277	0.00401531	0.00407494	0.00398791	0.00413328
252	0.00399418	0.00396712	0.00402584	0.00393937	0.00407907
254	0.00396267	0.00393631	0.00399439	0.00390963	0.00404349
257	0.00391602	0.00389029	0.00394644	0.00386446	0.00399369
259	0.00388587	0.00385945	0.00391762	0.00383349	0.00396789
262	0.00384091	0.00381514	0.00387157	0.00379085	0.00391942
264	0.00381123	0.00378556	0.00384215	0.0037575	0.00392413
267	0.00376787	0.00374306	0.00379748	0.0037189	0.00385758

Tabelle B.2.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel (wird fortgesetzt)

IV	$\varnothing \overline{\tau}_i$	min Intervall	max Intervall	min $\overline{\tau}_i$	max $\overline{\tau}_i$
269	0.00373995	0.00371554	0.00376927	0.00369295	0.0038266
272	0.00369819	0.00367404	0.0037273	0.00364861	0.0037728
274	0.00367101	0.00364668	0.00369987	0.00362234	0.00375045
277	0.00363088	0.00360733	0.00365966	0.00357725	0.00370916
279	0.00360464	0.00358159	0.00363262	0.00355889	0.00368541
282	0.00356615	0.00354298	0.00359417	0.00352273	0.00365629
284	0.00354035	0.00351766	0.00356757	0.00349423	0.00361071
287	0.00350317	0.003481	0.00353094	0.00346316	0.0035719
289	0.00347875	0.00345662	0.00350624	0.00343488	0.00354997

Tabelle B.2.2.: Empirische Daten zur bereinigten durchschnittlichen Schrittweite bei der Rundreise auf der Kreislinie mit Richtungswechsel

Index

A

Algorithmen

Bahncard	31
deterministisch	25
einfacher Ignore	85
einfacher Replan	84
First Fit	42
Greedy	
gerichtete Rundreise auf der Kreis-	
linie	96
Rundreise auf der Kreislinie .	137
Least-Recently-Used	44
Offline fair	21
Offline-Bearbeitung	7
Online	6, 8
Punktsuche	39
randomisiert	25
Ski Fahrer	35
Smart-Start	54
Wait Smartly	47
Zufall	25

Algorithmus

Online	
optimaler	20

Anfrage	5f., 80
Bearbeitung	7
erfüllte	80
maximale Verzögerung	73
Offline-Bearbeitung	7
Reisepunkt	80
Sequenz	6, 80
Wartezeit	72

Zeitmenge	80
-----------------	----

Average-Case-Analyse	10
----------------------------	----

B

Bearbeitungsinformation	<i>siehe</i>
Informationsvorlauf	
bereinigte Schrittweite	106

D

Distanzfunktion	83, 97
-----------------------	--------

E

Erwartungswert

Beitrag	116
bedingt	155

I

Informationsvorlauf

Bearbeitung	71
Startmethoden	72
Feststart	72
Fliegender Start	72
Vollstart	72
Vorausschau	69

K

kombinierter Informationsvorlauf ...	73
kompetitiv	20
strikt	20
Kompetitive Analyse	19
amortisierte Kosten	23

Böswilliger Gegner	24
blinder Gegner	26
offline adaptiver Gegner	26
online adaptiver Gegner	26
Potenzialfunktionen	23
Kompetitiver Faktor	19
bei Zufall	26
strikt	20
Kugel	
Äquator	193
Breitengrad	193
Großkreis	193
Kleinkreis	193
Kugelkappe	194
Kugelschicht	193
Kugelzone	194
Längengrad	193

M

Markovkette	181ff.
Übergangsmatrix	182
Fixpunkt-Vektor	183

O

Offline-Bearbeitung	7
Offline-Optimierung	7
Offline-Problem	7
Online-Optimierung	6
Grundalgorithmus	8
Optimierungsprobleme	11–17
<i>k</i> -server	14
Kaufentscheidung	11
Listen-Aktualisierung	15
Netzwerk Routing	16
Packen	12
Punktsuche	12
Rundreise	13
dial-a-ride	14
traveling repairman	13
traveling salesman	13

Rundreise auf der Kreislinie .	<i>siehe</i>
Rundreise auf der Kreislinie	
Rundreise auf der Kreislinie mit Rich-	
tungswechsel .	<i>siehe</i> Rundreise
auf der Kreislinie	
Tasksysteme metrisch	17
Transport	15
Verwaltung von Speicherseiten .	12

R

Rundreise auf der Kreislinie	79
Ausgangspunkt	79
Distanzfunktion	83, 97
Erweiterungen	93f.
gerichtet	96
maximale Dauer	85
MILP-Formulierung	91
Erläuterung	92
Reisepunkte	79
Beispiel	81
Route	80
Streckenlänge	84
Sequenz	80
Veranschaulichung	98–101, 138–141

S

stochastische Optimierung	9
Strategiekomponenten	21
Ignore	22
Replan	21
Warten	22

V

Vorausschauinformation	<i>siehe</i>
Informationsvorlauf	

Y

Yao's Prinzip	26
Online	27

Literaturverzeichnis

- [AAL05] Luca Allulli, Giorgio Ausiello, and Luigi Laura. On the power of lookahead in on-line vehicle routing problems. In Lusheng Wang, editor, *Computing and Combinatorics, 11th Annual International Conference, COCOON 2005, Kunming, China, August 16-29, 2005, Proceedings*, volume 3595 of *Lecture Notes in Computer Science*, pages 728–736. Springer, 2005. (Zitiert auf der Seite 67.)
- [AFL⁺95] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. Competitive algorithms for the on-line traveling salesman. In *In Proceedings of the 4th Workshop on Algorithms and Data Structures (WADS 95)*, pages 206–217, 1995. (Zitiert auf den Seiten 88 und 89.)
- [AFL⁺01] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. Algorithms for the on-line travelling salesman. *Algorithmica* 29, pages 560–581, July 13 2001. (Zitiert auf der Seite 53.)
- [AKR00] Norbert Ascheuer, Sven O. Krumke, and Jörg Rambau. Online dial-a-ride problems: Minimizing the completion time. In Horst Reichel and Sophie Tison, editors, *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, STACS 2000 (Lille, France, February 17-19, 2000)*, volume 1770 of *LNCS*, pages 639–650. Springer-Verlag, Berlin-Heidelberg, 2000. (Zitiert auf den Seiten 14 und 93.)
- [Alb93] Susanne Albers. The influence of lookahead in competitive paging algorithms (extended abstract). In Thomas Lengauer, editor, *European Symposium on Algorithms (ESA)*, volume 726 of *Lecture Notes in Comput. Sci.*, pages 1–12, Berlin-Heidelberg-New York-London-Paris-Tokyo-Hong Kong-Barcelona-Budapest, 1993. Springer-Verlag. (Zitiert auf der Seite 67.)
- [Alb96] Susanne Albers. Competitive online algorithms. *BRICS Lecture Series*, LS-96-2, September 1996. (Zitiert auf den Seiten 6, 14, 16, 23, 26 und 27.)
- [Alb98] Susanne Albers. A competitive analysis of the list update problem with lookahead. *Theoretical Computer Science*, 197(1–2):95–109, 1998. (Zitiert auf der Seite 67.)

- [Alb99] Susanne Albers. Online algorithms: A study of graph-theoretic concepts, December 14 1999. (Zitiert auf der Seite 13.)
- [Alb03] Susanne Albers. Online algorithms: a survey. *Mathematical Programming*, 97(1–2):3–26, 2003. (Zitiert auf den Seiten 6 und 17.)
- [Bam10] Eva Bamman. Online-Paging bei Informationsvorlauf. Bachelorarbeit, Universität Augsburg, 2010. (Weitere betreute Abschlussarbeit.)
- [BB01] Shai Ben-David and Allan Borodin. A new measure for the study of online algorithms, July 23 2001. (Zitiert auf der Seite 6.)
- [BCR93] Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J. E. Rawlins. Searching in the plane. *Inf. Comput*, 106(2):234–252, October 1993. (Zitiert auf den Seiten 12, 39 und 40.)
- [BD09] Florent Baty and Marie-Laure Delignette-Muller. *nlstools: tools for nonlinear regression diagnostics*, 2009. R package version 0.0-9. (Zitiert auf der Seite 101.)
- [BE98] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, 1 edition, 1998. (Zitiert auf den Seiten 5 und 19.)
- [BKdPS00] Michiel Blom, Sven O. Krumke, Willem E. de Paepe, and Leen Stougie. The online-TSP against fair adversaries. In *CIAC: Italian Conference on Algorithms and Complexity*, 2000. (Zitiert auf der Seite 21.)
- [Bor87] Karl Heinz Borgwardt. *The Simplex Method. A Probabilistic Analysis*, volume 1 of *Algorithms and Combinatorics*. Springer, Berlin, 1987. (Zitiert auf der Seite 197.)
- [Bor92] Allan Borodin. Can competitive analysis be made competitive? In John E. Botsford, Arthur G. Ryman, Jacob Slonim, and David Taylor, editors, *Proceedings of the 1992 Conference of the Centre for Advanced Studies on Collaborative Research, November 9-12, 1992, Toronto, Ontario, Canada, 2 Volumes*, pages 359–367. IBM, 1992. (Zitiert auf den Seiten 5 und 19.)
- [Bor01] Karl Heinz Borgwardt. *Optimierung, Operations Research, Spieltheorie: mathematische Grundlagen*. Birkhäuser, Basel, 1 edition, 2001. (Zitiert auf den Seiten 26, 27, 97, 137 und 200.)
- [Bre96] Dany Breslauer. On competitive on-line paging with lookahead. In *Symposium on Theoretical Aspects of Computer Science*, pages 593–603, 1996. (Zitiert auf der Seite 67.)

- [CJSW93] Edward G. Coffman, jr, David S. Johnson, Peter W. Shor, and Richard R. Weber. Markov chains, computer proofs, and average-case analysis of best fit bin packing. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 412–421, New York, NY, USA, 1993. ACM Press. (Zitiert auf der Seite 181.)
- [Dam02] Peter Damaschke. Two short notes on the on-line travelling salesman: handling times and lookahead. *Theor. Comput. Sci.*, 289(1):845–852, 2002. (Zitiert auf der Seite 67.)
- [FR06] Philipp Frieze and Jörg Rambau. Online-optimization of multi-elevator transport systems with reoptimization algorithms based on set-partitioning models. *Discrete Applied Mathematics*, (13):1908–1931, 2006. (Zitiert auf der Seite 5.)
- [FW98] Amos Fiat and Gerhard Woeginger. *Online algorithms: the state of the art*, volume 1442 of *Lecture Notes in Computer Science*. Springer-Verlag Inc., pub-SV:adr, 1 edition, 1998. (Zitiert auf der Seite 17.)
- [GKR01] Martin Grötschel, Sven O. Krumke, and Jörg Rambau. Online optimization of complex transportation systems. In *Online Optimization of Large Scale Systems*, chapter VIII, pages 705–729. Springer, 2001. (Zitiert auf der Seite 15.)
- [Gro95] Edward F. Grove. Online bin packing with lookahead. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'95 (San Francisco, California, January 22-24, 1995)*, pages 430–436, Philadelphia, PA, 1995. ACM SIGACT, SIAM, Society for Industrial and Applied Mathematics. (Zitiert auf den Seiten 19 und 67.)
- [Haa09] Carmen Haas. Empirische Online-Optimierung des Travelling-Salesman-Problems auf Sphären des drei und vier dimensionalen Raumes. Diplomarbeit, Universität Augsburg, 2009. (Zitiert auf der Seite 200.)
- [Hag05] Torben Hagerup. Online-optimierung. Vorlesungsskript, 2005. (Zitiert auf den Seiten 6, 11, 13 und 16.)
- [Hei05] Stefan Heinz. Policies for online target date assignment problems: Competitive analysis versus expected performance. Master's thesis, Technische Universität Berlin, December 2005. (Zitiert auf der Seite 6.)
- [HIKL99] Christoph Hipke, Christian Icking, Rolf Klein, and Elmar Langetepe. How to find a point on a line within a fixed distance. *Discrete Appl. Math.*, 93(1):67–73, 1999. (Zitiert auf der Seite 40.)

- [HKR00] Dietrich Hauptmeier, Sven O. Krumke, and Jörg Rambau. The online dial-a-ride problem under reasonable load. In *CIAC: Italian Conference on Algorithms and Complexity*, 2000. (Zitiert auf der Seite 14.)
- [HKRW01] Dietrich Hauptmeier, Sven O. Krumke, Jörg Rambau, and Hans-Christoph Wirth. Euler is standing in line — dial-a-ride problems with precedence-constraints. *Discrete Applied Mathematics*, 113(1):87–107, 2001. (Zitiert auf der Seite 93.)
- [Jun08] Dieter Jungnickel. *Graphs, networks and algorithms*, volume 5 of *Algorithms and Computation in Mathematics*. Springer, Berlin, third edition, 2008. (Zitiert auf der Seite 200.)
- [KdPPS03] Sven O. Krumke, Willem E. de Paepe, Diana Poensgen, and Leen Stougie. News from the online traveling repairman. *TCS: Theoretical Computer Science*, 295, 2003. (Zitiert auf der Seite 14.)
- [KdPRS01] Sven O. Krumke, Willem E. de Paepe, Jörg Rambau, and Leen Stougie. Online bin coloring. In *ESA: Annual European Symposium on Algorithms*, 2001. (Zitiert auf der Seite 12.)
- [KR02] Sven O. Krumke and Jörg Rambau. Online-optimierung. Vorlesungsskript, 2002. (Zitiert auf den Seiten 5, 11, 12, 13, 14, 16, 17, 23, 24, 26, 27, 31, 35, 37, 38, 39, 42, 45, 47, 54 und 55.)
- [KRT03] Sven O. Krumke, Jörg Rambau, and Luis M. Torres. Online-dispatching of automobile service units. *Appeared in: Operations Research Proceedings 2002. Selected Papers of the International Conference on Operations Research (SOR 2002), Klagenfurt, September 2-5, 2002*, pages 168–173, 2003. (Zitiert auf der Seite 14.)
- [Kru02] Sven O. Krumke. *Online Optimization: Competitive Analysis and Beyond*. Technische habilitationsschrift, Technische Universität Berlin, May 07 2002. (Zitiert auf den Seiten 6, 12, 14, 21, 47, 53 und 94.)
- [Mai10] Christina Mai. Probabilistische Austestung von Online-Algorithmen bei der Optimierung von Scheduling-Problem. Bachelorarbeit, Universität Augsburg, 2010. (Weitere betreute Abschlussarbeit.)
- [NR00] Nir Naaman and Raphael Rom. Average case analysis of bounded space bin packing algorithms. *EE Publication No. 1274*, February 12 2000. (Zitiert auf der Seite 181.)
- [Pot09] Jürgen Potthoff. Mehrstufige Prozesse: Modellierung mit Markovketten, 2009. (Zitiert auf den Seiten 181 und 183.)

- [ST85] Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, February 1985. (Zitiert auf der Seite 19.)
- [Tea09] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0. (Zitiert auf den Seiten 101, 102, 103, 104, 108, 115, 118, 143, 144, 151 und 163.)
- [Weia] Eric W. Weisstein. Faulhaber’s formula. MathWorld—A Wolfram Web Resource. (Zitiert auf der Seite 178.)
- [Weib] Eric W. Weisstein. Sphere point picking. MathWorld—A Wolfram Web Resource. (Zitiert auf der Seite 196.)
- [Wur07] Jutta Wurm. Eine empirische Analyse des Nutzens von Informationsvorlauf bei der Optimierung von Online-Rundreisen. Diplomarbeit, Universität Augsburg, 2007. (Weitere betreute Abschlussarbeit.)

Lebenslauf

seit 01.04.2005 wissenschaftlicher Mitarbeiter an der Universität Augsburg im Institut für Mathematik am Lehrstuhl Diskrete Mathematik, Optimierung und Operations Research

23.03.2005 Diplom an der Universität Augsburg mit den Schwerpunkten Finanz- und Bankwirtschaft sowie Informatik

23.03.2005 Abgabe der Diplomarbeit über gemischt-ganzzahlige quadratische Optimierung am Lehrstuhl Diskrete Mathematik, Optimierung und Operations Research bei Professor Schmidt

31.07.2002 Vordiplom

WS 2000/01 Beginn des Studiums der Wirtschaftsmathematik an der Universität Augsburg

1990 bis 1999 Dominikus Zimmermann Gymnasium in Landsberg am Lech, Abschluss am 25.06.1999 mit den Leistungskursen Mathematik sowie Wirtschaft

1985 bis 1990 Grund- und Hauptschule Kaufering

07. April 1979 Geburt in Starnberg