Inhaltsverzeichnis

1 Einleitung **1**

2 Algorithmen zur Bestimmung der Pareto-Optima **5**

2.1 Der Nemhauser/Ullmann-Algorithmus 5

2.2 Eine Verallgemeinerung auf das zweidimensionale Rucksackproblem 8

2.3 Ein Algorithmus für das mehrdimensionale Rucksackproblem . 14

3 Analyse in einem vereinfachten Modell **16**

3.1 Grundlegende Ergebnisse 16

3.2 Begriffsdefinitionen 17

3.2.1 Das vereinfachte Modell 17

3.2.2 Die erwartete Anzahl überdeckter Elemente 19

3.3 Berechnung des Erwartungswertes 21

3.3.1 Versuch einer exakten Bestimmung 21

3.3.2 Eine Abschätzung des Erwartungswertes 25

3.4 Die Konzentration der Anzahl pareto-optimaler Lösungen . . . 31

4 Ergebnisse aus der Analyse und den Testläufen **35**

4.1 Erwartete Laufzeit des verallgemeinerten Nemhauser/Ullmann-Algorithmus 35

4.2 Vergleich der erwarteten Anzahlen der pareto-optimalen Lösungen 35

4.3 Betrachtung von Resultaten aus Testläufen 37

4.4 Vergleich von Theorie und Praxis 38

5 Fazit **41**

Anhang **44**

A Sätze aus der Analysis und Stochastik **44**

Literatur **45**

1 Einleitung

In dieser Arbeit werden pareto-optimalen Lösungen betrachtet. Doch was sind pareto-optimale Lösungen? Allgemein gesprochen sind dies Lösungen von Problemen, die mehreren Optimierungsbedingungen genügen müssen. Bei solchen Optimierungsproblemen gibt es nicht immer eine optimale Lösung. So widersprechen sich in vielen Bereichen Zeit und Kosten oder Preis und Leistung. Dabei treten häufig „Lösungen“ auf, bei denen z.B. die Kosten nicht weiter gedrückt werden können, ohne die erforderliche Zeit zu erhöhen. Oder die Leistung kann nicht mehr erhöht werden, ohne dass der Preis ansteigen würde. Solche Lösungen bezeichnet man als pareto-optimal. Sie treten in der Wirtschaft, bei Ingenieuren und eben auch in der theoretischen Informatik auf.

Als bekanntes Beispiel lässt sich in der Informatik das Problem der ganzzahligen linearen Optimierung anführen. Während normale, lineare Optimierungsprobleme effizient gelöst werden können, ist dies für die durch eine Einschränkung des Lösungsraums auf ganze Zahlen entstehenden Probleme nicht mehr gegeben. Diese Problemklasse ist auch unter dem Namen Integer Programming, oder kurz IP, bekannt, welche NP-vollständig ist. Beim Integer Programming treten häufig pareto-optimale Lösungen auf.

Das mitunter einfachste und bekannteste Problem aus der Klasse IP ist das Rucksackproblem. Es besteht aus lediglich einer zu optimierenden Summe und einer einschränkenden Ungleichung. Beim Rucksackproblem sind Gegenstände mit einem möglichst hohem Gesamtnutzen in einen Rucksack zu packen, dessen Tragfähigkeit beschränkt ist.

Vom Rucksackproblem gibt es verschiedene Varianten. Bei einigen ist es

erlaubt, Gegenstände auch zum einem Anteil einzupacken. Diese Varianten sind dann nicht mehr NP-vollständig, sondern lassen sich einfach deterministisch lösen. Beim Multi-Rucksackproblem müssen die Gegenstände auf mehr als einen Rucksack verteilt werden. Das mehrdimensionale Rucksackproblem betrachtet Gegenstände, die entweder mehr als ein Gewicht oder mehr als einen Profit oder beides besitzen.

Das Rucksackproblem gehört zu denen am weitesten und intensivsten betrachteten Optimierungsproblemen in der Informatik. Zur exakten Lösung gibt es verschiedene Algorithmen. Es gibt einen einfachen dynamischen Algorithmus, dessen Laufzeit polynomiell in der Größe der Eingabewerte ist. Daher gehört das Rucksackproblem zur Klasse der pseudopolynomiell lösbaren Probleme. Unter den weiteren Algorithmen zu exakten Lösungen hat der so genannte Core-Algorithmus von Goldberg und Marchetti-Spaccamela [GMS84] die beste Laufzeit. Dieser Algorithmus basiert auf einem Konzept von Balas und Zemel [BZ80]. Dabei wird mit einer optimalen Lösung begonnen, bei der einige Elemente nur zum Teil eingepackt werden. Diese werden nach und nach durch „ganze“ Gegenstände ausgetauscht. Die dafür geeigneten Gegenstände, die bestimmte Bedingungen erfüllen müssen, werden in einer Menge zusammengefasst, die Core-Set genannt wird. Für „typische“ Instanzen des Rucksackproblems ist dieses Core-Set im average case relativ klein, wie Goldberg und Marchetti-Spaccamela in ihrer Analyse zeigen konnten.

Ein anderer Ansatz bestimmt in einer Art dynamischer Programmierung nach und nach alle pareto-optimalen Lösungen für eine Instanz des Rucksackproblems. Dieser einfache, elegante Algorithmus aus dem Jahr 1969 stammt von Nemhauser und Ullmann und wird seitdem in der Literatur

auch als Nemhauser/Ullmann-Algorithmus bezeichnet [NU69]. Die Laufzeit ist polynomiell in der Anzahl der Gegenstände n und der Anzahl der pareto-optimalen Lösungen q. Letztere ist für viele Instanzen des Rucksackproblems klein, d.h polynomiell in n. Im letzten Jahr konnten Beier und Vöcking die erste average-case Analyse vorlegen, die eine erwartete polynomielle Anzahl pareto-optimaler Lösungen für das Rucksackproblem unter verschiedenen stochastischen Modellen beweist [BV03]. Dieser Beweis und sein Ansatz dienten auch als Grundlage und Motivation für diese Arbeit. Dabei sollten die Ergebnisse von Beier und Vöcking auf das mehrdimensionale Rucksackproblem verallgemeinert werden. Ein direkte Übertragung ihrer Ergebnisse ist aber aus Gründen, die in Kapitel 3 näher erläutert werden, nicht möglich. Daher wurde ein anderer Ansatz gewählt, der auch eine Antwort auf die Frage nach dem Erwartungswert für die Anzahl der pareto-optimalen Lösungen zu liefern versucht. Dazu wurde ein vereinfachtes Modell betrachtet, das ein dem Rucksackproblem sehr ähnliches Problem beschreibt.

Zuerst aber wird der Nemhauser/Ullmann-Algorithmus erläutert. Im weiteren wird ein Algorithmus entwickelt, der effizient alle pareto-optimalen Lösungen für eine Eingabe des zweidimensionalen Rucksackproblems berechnet. Anschließend wird erläutert, warum ein ähnlich effizienter Algorithmus für höherdimensionale Rucksackprobleme wahrscheinlich nicht existiert und somit nur ein trivialer Ansatz bleibt.

Nachdem die beiden Algorithmen beschrieben und ihre Laufzeit bestimmt wurden, folgt in Kapitel 3 eine probabilistische Analyse, die die Anzahl der pareto-optimalen Lösungen q bestimmt. Dazu wird die oben angedeutete Vereinfachung des mehrdimensionalen Rucksackproblems vorgenommen. Diese ist notwendig, da eine komplette probabilistische Analyse des Rucksackpro-

blems zu komplex wäre.

Leider ist das aus dem vereinfachten Problem gewonnene Ergebnis nicht direkt auf das Rucksackproblem zu übertragen. Allerdings zeigen Testläufe auf zufälligen Instanzen des zweidimensionalen Rucksackproblems, dass die Anzahlen der pareto-optimalen Lösungen beider Problem wahrscheinlich polynomiell verknüpft sind. Diese Testläufe werden dann abschließend in Abschnitt 4.2 vorgestellt und den theoretischen Ergebnissen unter verschiedenen Annahmen gegenübergestellt.

2 Algorithmen zur Bestimmung der Pareto-Optima

In diesem Kapitel wird zuerst der Nemhauser/Ullmann-Algorithmus beschrieben, der effizient die pareto-optimalen Lösungen für eine Eingabe des Rucksackproblem berechnet. Die Laufzeit dieses Algorithmus liegt bei $O(nq)$, wobei n die Anzahl der Gegenstände beim Rucksackproblem und q die Anzahl der pareto-optimalen Lösungen ist. Im zweiten Abschnitt betrachten wir eine Verallgemeinerung des Algorithmus für das zweidimensionale Rucksackproblem durchgeführt. Dabei wird die Variante als Grundlage genommen, bei der jeder Gegenstand weiterhin einen Profit aber nun 2 Gewichte erhält. Die Laufzeit steigt bei diesem Algorithmus geringfügig auf $O(nq \log q)$ an.

Im dritten Abschnitt wird versucht darzulegen, warum ein ähnlich effizienter Algorithmus, dessen Laufzeit höchstens in einer Größenordnung von $O(q \log^{\mathrm{poly}(d)} q)$ bzgl. q liegt, wahrscheinlich nicht existiert. Es bleibt lediglich ein trivialer Ansatz, der eine Laufzeit von $O(q^2)$ liefert.

2.1 Der Nemhauser/Ullmann-Algorithmus

Im Rucksackproblem betrachten wir Bepackungen von Rucksäcken mit einer Auswahl von Gegenständen aus der Eingabe. Damit sind Rucksäcke Teilmengen der Menge aller Gegenstände. Um das Rucksackproblem für eine feste Eingabe zu lösen, genügt es, alle Rucksäcke zu bestimmen, die durch keinen anderen Rucksack dominiert werden. Ein Rucksack R dominiert einen anderen Rucksack R' genau dann, wenn die Summe der Profite der eingepackten Gegenstände von R größer der entsprechenden Summe von R' ist und R da-

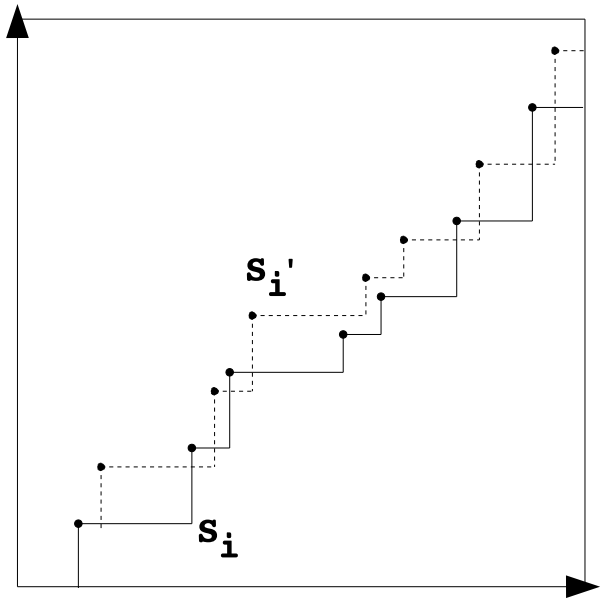

Abbildung 1: Darstellung von S_i und der verschobenen Kopie S_i' als Treppe

bei gleichzeitig leichter ist als R'. Das Gewicht eines Rucksacks ergibt sich hierbei aus der Summe der Gewichte aller enthaltenen Gegenstände.

Der Nemhauser/Ullmann-Algorithmus berechnet effizient alle pareto-optimalen Lösungen für das Rucksackproblem; das sind genau die Rucksäcke, die nicht dominiert werden. Als Eingabe für das Rucksackproblem gibt es n Gegenstände mit jeweils einem Gewicht w_i und einem Profit u_i für $i \in \{1 \ldots n\}$. Der Algorithmus berechnet nun ähnlich einem dynamischen Algorithmus nacheinander die Mengen aller pareto-optimalen Lösungen, wenn nur die ersten i Gegenstände betrachtet werden. Die Menge aller dieser Lösungen für die ersten i Gegenstände werden aufsteigend nach ihrem Gewicht sortiert in der Liste S_i gespeichert. Die Liste S_0 besteht nur aus dem leeren Rucksack.

Nun muss nur noch beschrieben werden, wie aus der Liste S_i die Liste S_{i+1} berechnet wird. Dazu wird zuerst eine Kopie S_i' von S_i erzeugt und

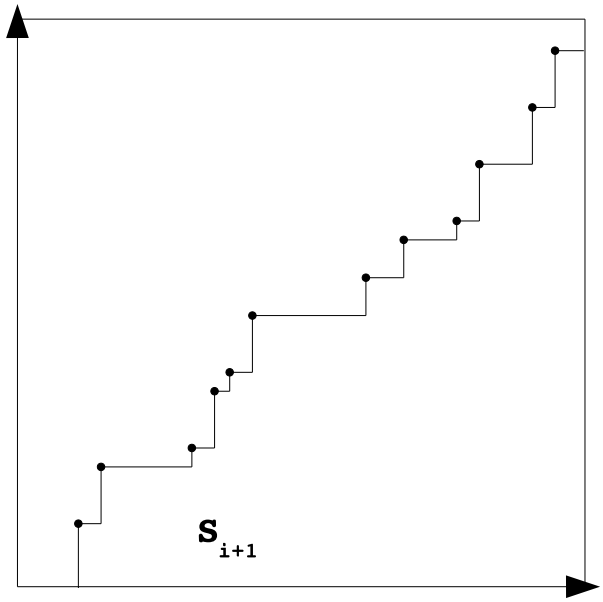

Abbildung 2: Die neu berechnete Treppe S_{i+1}

in jeden Rucksack der Kopie der Gegenstand $i + 1$ hinzu gepackt. Auch diese Kopie ist danach nach aufsteigenden Gewichten sortiert. Jetzt müssen beide Listen zu einer Liste zusammengefügt werden, wobei die dominierten Rucksäcke gelöscht werden. Zur Veranschaulichung lässt sich dieser Prozess gut in einer Graphik darstellen. Jeder Rucksack der Menge S_i wird dabei nach Gewicht und Profit in ein Koordinatensystem projiziert. Dort entsteht eine Art Treppe. Die Kopie S_i' ist nun eine um w_{i+1} in der X-Koordinate und um u_{i+1} in Y-Richtung verschobene zweite Treppe, die die erste teilweise überlappt (s. Abbildung 1). S_{i+1} ist nun die dabei entstandene obere Treppe, die in Abbildung 2 dargestellt ist. Diese lässt sich leicht in linearer Zeit bzgl. der Anzahl der Treppenstufen berechnen. Dazu reicht es, die beiden Treppen gleichzeitig von links nach rechts abzugehen und jeweils die höhere Stufe zu wählen.

Bezeichne q_i die Größe der Menge S_i, so benötigt die Berechnung von

$S_i + 1$ die Zeit $O(q_i)$. Unter der Annahme $q_{i+1} \geq q_i$ beträgt die gesamte Rechenzeit $O(nq)$, wobei q die Anzahl aller pareto-optimalen Lösungen der Eingabe, also q_n, ist.

Im worst-case ist die Anzahl aller pareto-optimalen Lösungen zwar 2^n, allerdings ist sie für typische Eingaben wesentlich kleiner. So haben Beier und Vöcking gezeigt, dass diese Anzahl für verschiedene zufallsverteilte Eingaben polynomiell in n ist [BV03]. In Kapitel 3 wird ein ähnliches Ergebnis für eine abgeschwächte Variante des mehrdimensionalen Rucksackproblems bewiesen. Zuerst wird aber nun der Algorithmus vorgestellt, mit dem sich die pareto-optimalen Lösungen für das zweidimensionale Rucksackproblem berechnen lassen.

2.2 Eine Verallgemeinerung auf das zweidimensionale Rucksackproblem

In diesem Abschnitt wird eine Verallgemeinerung des Nemhauser/Ullmann-Algorithmus für das mehrdimensionale Rucksackproblem mit 2 Gewichtswerten und einem Profit beschrieben. Die grundlegende Idee, der Reihe nach die Mengen S_i der dominierenden, oder auch pareto-optimalen, Rucksäcke zu bestimmen bleibt bestehen. Der Algorithmus erzeugt ebenfalls eine Duplikatliste der Einträge in S_i und addiert die Gewichtswerte und Profite des Gegenstandes $i + 1$ hinzu. Nur die Bestimmung der Menge S_{i+1} aus den beiden Listen ist aufwendiger. Zur Vereinfachung wird im weiteren Verlauf dieser Arbeit angenommen, dass keine zwei Rucksäcke denselben Profit oder dasselbe Gewicht haben. Diese Annahme vereinfacht auch die in Kapitel 3 durchgeführten Analysen.

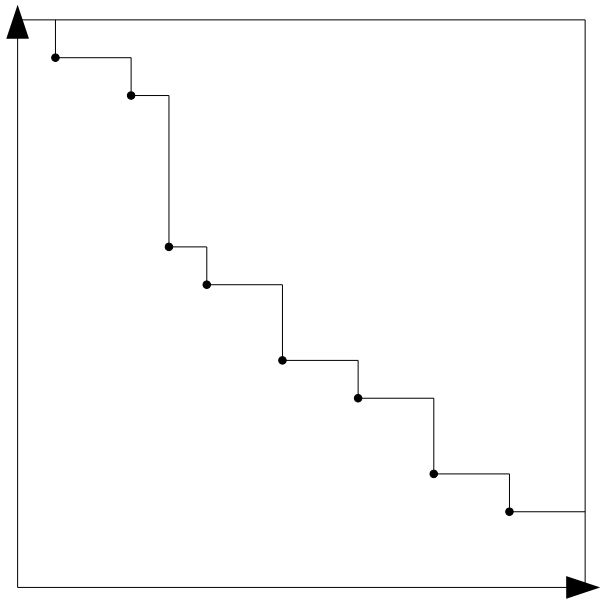

Abbildung 3: Die schon betrachteten Pareto-Optima bilden eine Art Front

Hierzu werden die Rucksäcke aus S_i und der daraus erzeugten, kopierten Liste S_i' nach absteigendem Profit in eine neue Liste sortiert, die damit alle möglichen pareto-optimalen Lösungen aus S_{i+1} enthält. Für die Rucksäcke dieser Liste wird nun nach und nach überprüft, ob sie pareto-optimal sind. Der zweidimensionale Fall, d.h. zwei Gewichte und ein Profit, lässt sich noch übersichtlich grafisch darstellen. Die Rucksäcke werden nach und nach anhand ihrer Gewichte in ein zweidimensionales Koordinatensystem projiziert. Dabei entsteht eine Art Front, abgesteckt durch die einzelnen Rucksäcke. Eine Veranschaulichung dieser Front ist in Abbildung 3 zu sehen.

Wird nun der nächste Rucksack, nennen wir ihn R, eingefügt, dessen Profit kleiner ist als der aller vorangegangenen Rucksäcke, treten zwei Fälle auf. Im Fall (a) befindet sich R rechts und oberhalb der Front (siehe Abbildung 4). Das bedeutet, dass R schwerer als bereits vorangegangene Rucksäcke ist. Da R aber einen kleineren Profit hat, muss er durch mindestens einen ande-

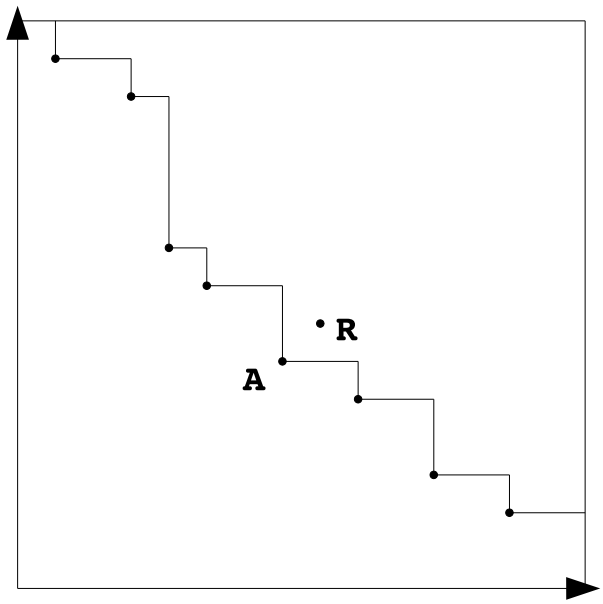

Abbildung 4: R liegt innerhalb der Front

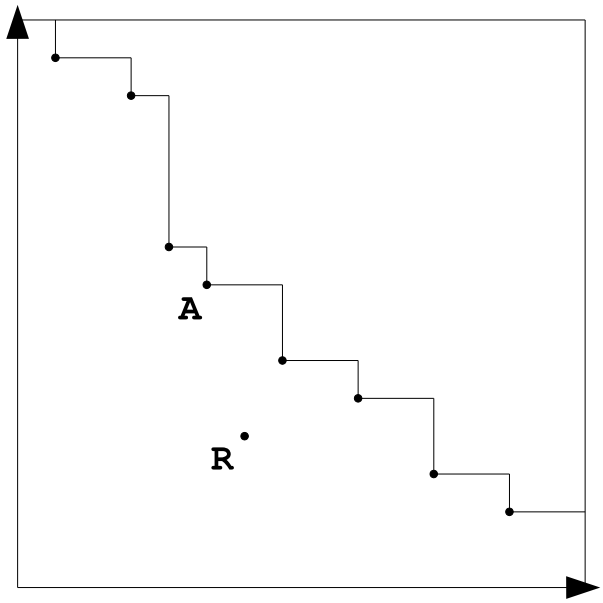

Abbildung 5: R liegt außerhalb der Front und ist somit ein neues Pareto-Optimum

ren Rucksack, in diesem Beispiel ist es Rucksack A, dominiert werden und ist somit nicht pareto-optimal. Der betrachtete Rucksack R wird danach verworfen.

Im Fall (b) befindet sich der neue Rucksack R links und unterhalb der Front (siehe Abbildung 5). R ist also leichter als alle vorangegangenen Rucksäcke. Da alle folgenden Rucksäcke eine kleineren Profit haben, ist er auch pareto-optimal. R muss nun noch in die Front und in die Menge S_{i+1} eingefügt werden. Beim Einfügen in die Front müssen dort alle Rucksäcke entfernt werden, die nun durch R überdeckt werden.

Um die Position eines neuen Rucksacks bezüglich der Front zu bestimmen und ein neues Pareto-Optimum in die Front einzufügen, wird eine Datenstruktur benötigt, die ihre Elemente nach einer Größe, für diesen Algorithmus ist es das erste Gewicht, sortiert und dabei folgende Laufzeit-Eigenschaften aufweist:

1. Suche in $O(\log n)$.
2. Einfügen eines Elementes in $O(\log n)$.
3. Löschen eines Elementes in $O(\log n)$.
4. Bestimmung eines direkten Nachfolgers zu einem Element in $O(1)$.

Hierfür eignen sich z.B. Red-Black-Trees oder Bayer-Bäume, bei denen die Daten nur in den Blättern stehen und zusätzlich über eine Liste verkettet werden können, um auch die Eigenschaft 4 zu erfüllen.

Dieser Baum, mit dem die Front repräsentiert wird, wird nach dem ersten Gewicht sortiert. Dann lässt sich anhand einer Suche die Ecke A der Front

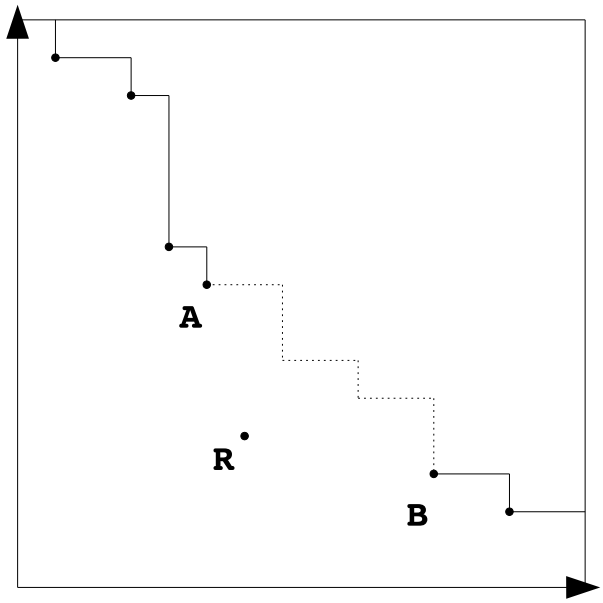

Abbildung 6: Alle Ecken zwischen A und B können gelöscht werden

bestimmen, die direkt links von R liegt. Ist das zweite Gewicht dieser Ecke kleiner als das von R, so liegt R rechts und oberhalb der Front und kann verworfen werden (siehe Abbildung 4).

Andernfalls ist R pareto-optimal. Es werden nun solange alle im Baum auf A folgenden Ecken der Front entfernt, bis eine Ecke B erreicht wird, deren zweites Gewicht kleiner als das von R ist. Die zu entfernenden Ecken, samt der sich daraus ergebenden Kanten, der Front sind in Abbildung 6 gestrichelt dargestellt. Nach dem Löschen der alten Ecken wird R in den Baum eingefügt. Dadurch werden implizit zwei neue Kanten in die Front eingebaut, wie sich in Abbildung 7 erkennen lässt. Damit wäre die neue Front fertig berechnet.

Die Laufzeitanalyse des Algorithmus gestaltet sich recht einfach. Betrachten wir zuerst den Aufwand für die Berechnung von S_{i+1} aus S_i. Dazu sind die Bestimmung von A sowie ein eventuelles Löschen von Ecken und das

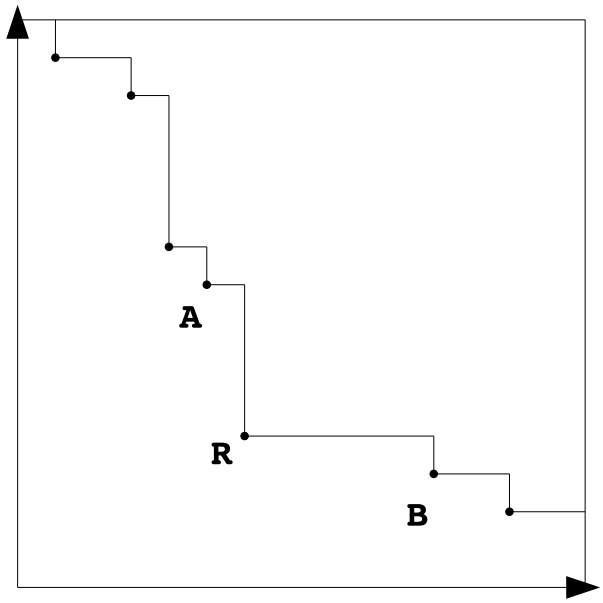

Abbildung 7: Nun bildet R mit den Ecken A und B die neue Front

Hinzufügen von R zu berücksichtigen. Eine Suche im Baum kann in logarithmischer Zeit durchgeführt werden. Das Entfernen von Ecken aus der Front ist für jede Ecke ebenfalls in logarithmischer Zeit möglich. Hierbei ist zu beachten, dass jedes mögliche Pareto-Optimum höchstens einmal als Ecke aus der Front entfernt werden kann. Damit ergibt sich für das Entfernen der Ecken insgesamt eine Laufzeit von $O(q_i \log q_i)$. Das Einfügen in den Baum wiederum ist in logarithmischer Zeit möglich. Die gesamte Laufzeit, die sich durch das Betrachten des $i+1$-ten Gegenstandes ergibt, liegt somit in der Größenordnung $O(q_i \log q_i)$.

Wird dieser Vorgang nun für alle $i \in \{1 \ldots n\}$ wiederholt, liegt die gesamte Rechenzeit für den Algorithmus für das zweidimensionalen Rucksackproblem bei $O\left(\sum_{i=1}^{n} q_i \log q_i\right)$. Unter der Annahme $q_{i+1} \geq q_i$ ergibt sich eine Laufzeit von $O(nq \log q)$ mit $q = q_n$.

2.3 Ein Algorithmus für das mehrdimensionale Rucksackproblem

Nach dem Ergebnis des letzten Abschnitts besteht die Hoffnung, auch für das mehrdimensionale Rucksackproblem einen ähnlichen Algorithmus entwerfen zu können, der effizient alle pareto-optimalen Lösungen bestimmt. Leider gibt es aber keine nahe liegende Verallgemeinerung. Das Problem liegt in der Suche nach dem möglichen, bereits betrachteten Pareto-Optimum, das einen neuen Rucksack dominiert. Während diese Suche beim zweidimensionalen Rucksackproblem noch in einer Zeit von $O(\log q)$ möglich ist, lässt sich diese Laufzeit bereits bei drei Dimensionen nicht mehr annähernd erreichen. Bei der zweidimensionalen Front war es möglich, effizient zu bestimmen, auf welcher Seite ein Rucksack R liegt. Bei einer höherdimensionalen Front ist dies nicht mehr realisierbar.

Wird die Suche aus einem anderen Blickwinkel betrachtet, wird vielleicht klarer, warum es wahrscheinlich keine effiziente Suche im höherdimensionalen Fall gibt. Zu jedem der d Gewichtswerte eines gerade betrachteten Rucksacks R_i existiert eine Menge G_i^k von Rucksäcken, die leichter sind und dabei einen höheren Profit haben. Ist die Schnittmenge G_i all dieser Mengen nicht leer, so existiert mindestens ein Element A, dessen Gewichtswerte alle kleiner sind als die von R_i und das einen höheren Profit hat. Damit wäre dann R_i nicht pareto-optimal. Ist die Schnittmenge G_i dagegen leer, so ist R_i pareto-optimal. Insgesamt gibt es bis zu q^d unterschiedliche mögliche Schnittmengen. Damit scheidet eine Berechnung durch ein Preprocessing aus. Eine Bestimmung von G_i zum Zeitpunkt der Betrachtung von R_i ist wohl auch nicht in logarithmischer Zeit bezüglich q möglich, auch wenn über die Mengen G_i^k einige spezielle Eigenschaften bekannt sind.

Damit bleibt als Algorithmus für den mehrdimensionalen Fall vorerst der naive Ansatz, für jeden neu zu betrachtenden Rucksack R_i alle bis zu $O(q)$ möglichen Rucksäcke darauf zu testen, ob sie R_i dominieren. Dadurch fällt für die Betrachtung jedes Rucksacks eine Laufzeit von $O(dq)$ an, da alle d Gewichtswerte getestet werden müssen. Die gesamte Laufzeit dieses trivialen Ansatzes liegt somit bei $O(nq^2)$, da für jeden neuen Gegenstand bis zu q Rucksäcke auf ihre Eigenschaft als Pareto-Optimum getestet werden müssen. Diese Laufzeit führt, wie später noch genauer erläutert wird, leider zu keiner Aussage über eine erwartete Laufzeit auf Instanzen des Rucksackproblems mit zufällig gewählten Eingaben.

3 Analyse in einem vereinfachten Modell

3.1 Grundlegende Ergebnisse

Für das eindimensionale Rucksackproblem haben Beier und Vöcking die erwartete Anzahl der Pareto-Optima für verschiedene Zufallsverteilungen untersucht[BV03]. Hierbei ergibt sich für die erwartete Anzahl der Pareto-Optima bei einer uniformen Verteilung der Gewichte - die Profite wählt ein Adversary unabhängig hiervon - eine obere Schranke von $O(n^3)$.

In einem ersten Ansatz wurde versucht, die Ergebnisse für die erwartete Höhe einer Stufe aus der Arbeit von Beier und Vöcking [BV03] auf mehrere Dimensionen zu verallgemeinern. Dieser Ansatz sah zunächst viel versprechend aus. Die Ergebnisse für die Stufenhöhe gelten fast unverändert auch für das mehrdimensionale Rucksackproblem. Nur die anschließend im eindimensionalen Falle so einfache Schlussfolgerung von der erwarteten Stufenhöhe auf die Anzahl der Stufen erwies sich im mehrdimensionalen Fall als nahezu unmöglich. Gilt z.B. für alle Gegenstände einer Eingabe für das zweidimensionale Rucksackproblem, dass $w_{i1} + w_{i2} = 1$, so liegen genauso wie diese Gegenstände auch alle pareto-optimalen Lösungen diagonal zueinander, wobei die Diagonalen senkrecht zur ersten Winkelhalbierenden liegen. Auf einer solchen Diagonalen könnten sich beliebig viele Pareto-Optima befinden, ohne dass die Stufenhöhe dadurch kleiner sein müsste. Auf einer Diagonalen sind nämlich nur Stufen zu Rucksäcke außerhalb dieser Diagonalen möglich. Somit ließ sich eine Analyse über die Stufenhöhe für das mehrdimensionale Rucksackproblem nicht realisieren.

Eine erschöpfende, probabilistische Analyse des mehrdimensionalen Ruck-

sackproblems stellte sich aufgrund der Abhängigkeiten der Rucksäcke untereinander, die durch die einzupackenden Gegenstände erzeugt werden, als äußerst schwierig heraus. Daher betrachte ich in dieser Arbeit nicht das normale Rucksackproblem. Vielmehr besteht mein Modell aus „Rucksäcken“, die allerdings einzelne, unabhängige Elemente darstellen. Deren Anzahl m muss, um einen Vergleich mit dem Rucksackproblem zu ermöglichen, exponentiell in der Anzahl der Gegenstände sein, die in die Rucksäcke im normalen Rucksackproblem eingepackt werden können. Im Vergleich zum Rucksackproblem mit n Gegenständen und damit 2^n möglichen Rucksäcken werden $m = 2^n$ Elemente betrachtet. Der einzige Unterschied zum Rucksackproblem liegt dann noch in der fehlenden Abhängigkeit dieser Elemente, welche die Analyse vereinfacht.

3.2 Begriffsdefinitionen

3.2.1 Das vereinfachte Modell

Im vereinfachten Modell besteht die Eingabe aus m Elementen. Jedes Element hat d Gewichtswerte w_{ik} mit $i \in \{1 \dots m\}$ und $k \in \{1 \dots d\}$, die uniform zufällig aus dem Intervall $[0; 1]$ gewählt werden. Die Profite u_i dürfen wiederum von einem Adversary beliebig, aber unabhängig von den Gewichten, bestimmt werden.

In den beiden folgenden Analysen wird nun der Erwartungswert für die Anzahl der pareto-optimalen Elemente berechnet. Dieser sollte in der Größenordnung polylog(m) liegen. Diese Größenordnung würde der polynomiellen Abhängigkeit der Anzahl der pareto-optimalen Lösungen von der Anzahl der Gegenstände im Rucksackproblem entsprechen.

Zur Bestimmung des Erwartungswertes werden in den folgenden Abschnitten zwei verschiedene Ansätze verfolgt. Der Ansatz aus dem ersten Abschnitt ist bereits in verwandten Arbeiten gewählt worden. So ist zum Beispiel die Frage nach der Anzahl der Punkte einer konvexen Hülle im $\mathbb{R}^d$ für zufällig gewählte Punkte bereits hinreichend betrachtet worden. Das vereinfachte Modell für das Rucksackproblem lässt sich auf diese Fragestellung zur konvexen Hülle zurückführen. Das erste Ergebnis aus dem Jahre 1963/64 für die erwartete Anzahl extremer Punkte einer konvexen Hülle von zufälligen Punkten kommt von Rényi und Sulanke [RS63, RS64]. In einer neueren Arbeit zu dem gleichen Thema betrachten Damerow und Sohler die Anzahl dieser extremen Punkte in einer smoothed Analyse [DS04].

Eine smoothed Analyse ist eine neuere Form der Betrachtung von Laufzeiten, bei der versucht wird, einen Hinweis auf die Laufzeit zwischen average-case und worst-case zu liefern. Dabei wird typischerweise eine worst-case Eingabe, oder eine Gruppe von worst-case Eingaben, genommen und danach randomisiert leicht verändert. Das kann z.B. ein Verrauschen der Werte oder eine Änderung in der Reihenfolge, z.B. bei Sortierproblemen, sein. Typischerweise wird dabei versucht zu zeigen, das eine solch „gesmoothte“ worst-case Eingabe Laufzeiten in der Nähe des average-cases erreicht. Damit wäre dann die Bedeutung des worst-cases stark gemindert, da die Laufzeit selbst bei worst-case ähnlichen Eingaben klein ist.

Ein solches Ergebnis haben Damerow und Sohler für die Anzahl der extremen Punkte einer konvexen Hülle aus zufällig gewählten Punkten gezeigt. Eine direkte Übertragung auf das Rucksackproblem ist zwar nicht ohne weiteres möglich, aber für das hier betrachtete vereinfachte Modell lassen sich die Ergebnisse anwenden.

In dieser Arbeit steht mehr die Konzentration der erwarteten Anzahl von pareto-optimalen Lösungen im Vordergrund. Daher wird in der zweiten Analyse ein etwas anderer Ansatz über eine Chernoff-Schranke gewählt. Damit lässt sich zeigen, dass die Anzahl mit hoher Wahrscheinlichkeit in der Größenordnung des Erwartungswertes liegt.

3.2.2 Die erwartete Anzahl überdeckter Elemente

Für die Analyse des Problems erweist sich der Begriff des Überdeckens von anderen Elementen als praktisch. Ein Element R_i überdeckt ein anderes Element R_j, falls keines der d Gewichte von R_j größer als das entsprechende Gewicht von R_i ist. Ein Element überdeckt sich nach dieser Definition insbesondere immer selbst. In Abbildung 8 wird diese Eigenschaft für den Fall $d = 2$ anschaulich dargestellt.

Formal wird für das Überdecken eine Zufallsvariable X_{ij} definiert. Hierbei ist zu beachten, dass ein Element sich nach dieser Definition immer selbst überdeckt.

$$X_{ij} = \begin{cases} 1, \text{ falls } \forall k \in \{1 \dots d\}\ w_{ik} \geq w_{jk} \\ 0, \text{ sonst} \end{cases}$$

Die erwartete Anzahl der von einem Element R_i überdeckten Elemente ist abhängig von dem Hypervolumen, das es überdeckt. Dieses beträgt im $[0;1]^d$ genau $p_i = \prod_{k=1}^{d} w_{ik}$. Also ist die Wahrscheinlichkeit, dass $X_{ij} = 1$ ist, genau p_i. Mit $X_i = \sum_{j=1}^{m} X_{ij}$ folgt für die erwartete Anzahl der von R_i überdeckten Elemente über die Linearität des Erwartungswertes $\mathbf{E}[X_i] = \mathbf{E}[\sum_{j=1}^{m} X_{ij}] = \sum_{j=1}^{m} \mathbf{E}[X_{ij}] = p_i(m-1) + 1$, da $X_{ii} = 1$.

Die Wahrscheinlichkeit, dass das Element R_i nun ein Pareto-Optimum

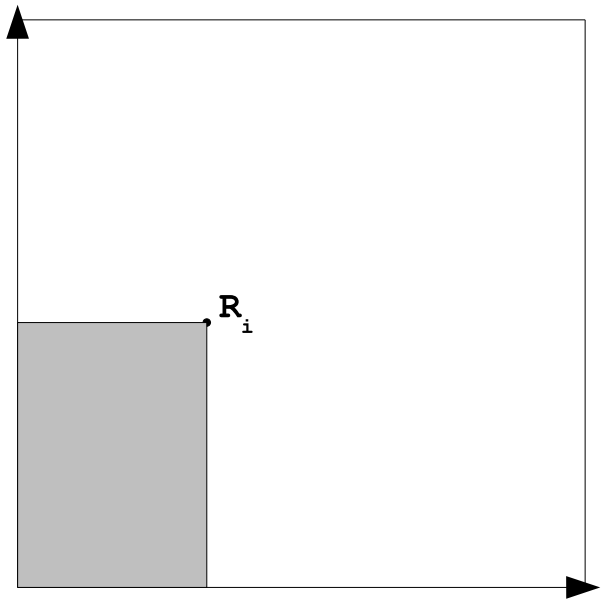

Abbildung 8: Das Element R_i überdeckt alle Elemente, die in dem grauen Bereich liegen

darstellt, ist gleich der Wahrscheinlichkeit, dass R_i den größten Profit aller von ihm überdeckten Elemente hat. Da der Profit aber unabhängig von den Gewichten ist, ist diese Wahrscheinlichkeit genau $\frac{1}{X_i}$. Auch hierzu wird eine Zufallsvariable definiert.

$$O_i = \begin{cases} 1, \text{ falls } R_i \text{ pareto-optimal ist} \\ 0, \text{ sonst} \end{cases}$$

3.3 Berechnung des Erwartungswertes

3.3.1 Versuch einer exakten Bestimmung

Um den Erwartungswert von O_i genau zu berechnen, muss man nach der Definition des Erwartungswertes die Summe über das Produkt der Ereignisse und ihrer Wahrscheinlichkeiten bestimmen. Die Ereignisse sind hierbei das Überdecken von genau s Elementen. Zusätzlich wird noch eine weitere Zufallsvariable für das Ereignis definiert, dass ein Element den höchsten Profit von s Elementen hat. Die beiden Zufallsvariablen für die beschriebenen Ereignisse sind im folgenden auch formal beschrieben.

$$U_{is} = \begin{cases} 1, \text{ falls } R_i \text{ genau } s \text{ Elemente überdeckt, d.h. } \sum_{l=1}^{m} X_{il} = s \\ 0, \text{ sonst} \end{cases}$$

$$H_{is} = \begin{cases} 1, \text{ falls } R_i \text{ den höchsten Profit von } s \text{ Elementen hat} \\ 0, \text{ sonst} \end{cases}$$

Mit diesen Zufallsvariablen und der Gegenwahrscheinlichkeit q_i zu p_i lässt sich der Erwartungswert von O_i wie folgt darstellen.

$$\begin{aligned} \mathbf{E_{exakt}}[O_i] &= 0 \cdot \mathbf{Pr}[O_i = 0] + 1 \cdot \mathbf{Pr}[O_i = 1] \\ &= \sum_{s=1}^{m} \mathbf{Pr}[U_{is} = 1 \wedge H_{is} = 1] \\ &= \sum_{s=1}^{m} \mathbf{Pr}[U_{is} = 1] \cdot \mathbf{Pr}[H_{is} = 1 | U_{is} = 1] \end{aligned}$$

Die Wahrscheinlichkeit, dass genau s Elemente von R_i überdeckt werden, lässt sich nun leicht angeben. Dabei ist zu beachten. dass sich R_i nach De-

finition immer selbst überdeckt und somit noch weitere $s-1$ Elemente der $m-1$ möglichen überdeckt werden müssen. Die Wahrscheinlichkeit dafür ist durch folgenden Ausdruck gegeben.

$$\mathbf{Pr}[U_{is}=1] = \binom{m-1}{s-1} p_i^{s-1}(1-p_i)^{m-s}$$

Die Wahrscheinlichkeit, dass R_i den höchsten Profit von s Elementen hat, wenn es genau s Elemente überdeckt, ist genau $\frac{1}{s}$, da die Profite unabhängig von den Gewichten gewählt wurden. Insgesamt gilt damit für den Erwartungswert von O_i

$$\begin{aligned}\mathbf{E_{exakt}}[O_i] &= \sum_{s=1}^{m} \mathbf{Pr}[U_{is}=1] \cdot \mathbf{Pr}[H_{is}=1|U_{is}=1] \\ &= \sum_{s=1}^{m} \left(\binom{m-1}{s-1} p_i^{s-1}(1-p_i)^{m-s} \frac{1}{s} \right)\end{aligned}$$

Mittels einer Umformung des Binominalkoeffizienten lässt sich diese Summe nun stark vereinfachen. Dazu wird folgende Beziehung genutzt.

$$\binom{m-1}{s-1}\frac{1}{s} = \frac{(m-1)!}{(s-1)!(m-s)!}\frac{1}{s} = \frac{1}{m}\binom{m}{s}$$

Mit dieser Beziehung lässt sich die Summe auflösen. Dabei wird der fehlende Summand für $s=0$ ergänzt, um eine Summe zu erhalten, die sich nach dem

binomischen Lehrsatz vereinfachen lässt.

$$\begin{aligned}
\mathbf{E}_{\mathbf{exakt}}[O_i] &= \sum_{s=1}^{m} \left(\binom{m-1}{s-1} p_i^{s-1} (1-p_i)^{m-s} \frac{1}{s} \right) \\
&= \frac{1}{m} \sum_{s=1}^{m} \left(\binom{m}{s} p_i^{s-1} q_i^{m-s} \right) \\
&= \frac{1}{p_i m} \left(\sum_{s=0}^{m} \left(\binom{m}{s} p_i^{s} q_i^{m-s} \right) - \binom{m}{0} p_i^0 q_i^m \right) \\
&= \frac{1}{p_i m} \left((p_i + q_i)^m - q_i^m \right) \\
&= \frac{1 - q_i^m}{p_i m}
\end{aligned}$$

Um nun den Erwartungswert zu ermitteln, reicht es nach Satz A.2 aus dem Anhang aus, das Integral dieses Ausdrucks multipliziert mit den Dichtefunktionen der Zufallsvariablen w_{ik} für $k \in \{1 \ldots d\}$ zu bilden. Da die Gewichtsvariablen uniform zufällig aus dem Intervall $[0; 1]$ sind, ergibt sich nach Folgerung A.1 folgendes Integral.

$$\mathbf{E}_{\text{exakt}}[O_i] = \int_0^1 \frac{1 - (1 - \prod_{k=1}^{d} w_i k)^m}{m \prod_{k=1}^{d} w_{ik}} \prod_{k=1}^{d} dw_{ik}$$

Hierbei ist zu beachten, dass die Variable d, die die Anzahl der Dimension angibt, nicht mit dem d aus dem Infitisimal dw_{ik} verwechselt wird.

Für dieses Integral lässt sich keine kompakte symbolische Lösung angeben. Daher wurde es numerisch mit dem algebraischen System *maple* ausgewertet. Die Ergebnisse dieser Auswertung werden für den zweidimensionalen Fall dann im Abschnitt 4.2 mit der Approximation des Erwartungswertes verglichen, die im folgenden Abschnitt beschrieben wird.

Des weiteren wird eine nahe liegende Näherung für dieses Integral betrachtet. Für die typischerweise großen Werte von m drängt sich die Näherung $1 - q^m = 1$ quasi auf, da q fast immer kleiner Eins ist. Allerdings wird das Integral durch die Näherung uneigentlich und ist nicht mehr definiert. Daher wird ein Trick angewendet, der die Größenordnung der Lösung nicht ändert und auch im folgenden Abschnitt auftaucht. Dazu werden einfach nur die Elemente betrachtet, deren Gewichte alle größer als $\frac{1}{m}$ sind. Alle anderen Elemente können als pareto-optimal betrachtet werden, da ihre erwartete Anzahl kleiner als $md\frac{1}{m} = d$ und damit konstant ist. Das Integral bekommt dann als untere Grenze $\frac{1}{m}$. Das ist zulässig, da die als pareto-optimal betrachteten Elemente weiterhin überdeckt werden und sich damit die Wahrscheinlichkeit, dass ein Element den größten Profit aller von ihm überdeckten Elemente hat, nicht verändert.

Nach der Abschätzung des Zählers mit Eins, lässt sich das Integral einfach in d einfache Integrale ausmultiplizieren, da die anderen Variablen für ein gegebenes Infitisimal nur als Konstante fungieren.

$$\begin{aligned}
\mathbf{E}_{\text{exakt}}[O_i] &\leq \int\limits_{\frac{1}{m}}^{1} \frac{1}{m \prod_{k=1}^{d} w_{ik}} \prod_{k=1}^{d} dw_{ik} + \frac{d}{m} \\
&= \frac{1}{m} \prod_{k=1}^{d} \int\limits_{\frac{1}{m}}^{1} \frac{1}{w_{ik}} \, dw_{ik} + \frac{d}{m} \\
&= \frac{1}{m} \left(\ln 1 - \ln \frac{1}{m} \right)^d + \frac{d}{m} \\
&= \frac{1}{m} \ln^d m + \frac{d}{m}
\end{aligned}$$

Nun folgt auf einfache Weise über die Linearität des Erwartungswertes ei-

ne obere Schranke für die erwarte Anzahl pareto-optimaler Lösungen. Diese Schranke stimmt auch mit dem Ergebnis aus dem nächsten Abschnitt überein.

$$\begin{aligned}\mathbf{E}[q] &= \mathbf{E}_{\mathbf{exakt}}\left[\sum_{i=1}^{m} O_i\right] = \sum_{i=1}^{m} \mathbf{E}_{\mathbf{exakt}}[O_i] = m\,\mathbf{E}_{\mathbf{exakt}}[O_i] \\ &\leq \ln^d m + d\end{aligned}$$

3.3.2 Eine Abschätzung des Erwartungswertes

In diesem Abschnitt wird nun mit Hilfe einer Chernoff-Schranke der Erwartungswert für die Anzahl der pareto-optimalen Elemente approximiert. Durch die Chernoff-Schranke kann zusätzlich gezeigt werden, dass die Anzahl stark konzentriert um den Erwartungswert herum liegt. Dies geschieht in Abschnitt 3.4.

Als Ausgangspunkt dient wiederum die Berechnung des Erwartungswertes von O_i. Hierzu wurde nun eine Chernoff-Schranke auf X_i angewendet.

$$\begin{aligned}\varepsilon &= \sqrt{\frac{2\ln m}{p_i(m-1)+1}} \\ \mathbf{Pr}\left[X_i <= (1-\varepsilon)\,\mathbf{E}[X_i]\right] &\leq \exp\left(-\mathbf{E}[X_i]\frac{\varepsilon^2}{2}\right) = \frac{1}{m}\end{aligned}$$

In allen Fällen, in denen das Element R_i weniger als $(1-\varepsilon)\,\mathbf{E}[X_i]$ Elemente überdeckt, kann angenommen werden, dass R_i pareto-optimal ist. Dadurch werden im schlimmsten Fall maximal $m\frac{1}{m} = 1$ zusätzliches Element als Pareto-Optimum betrachtet, welches keines ist. In den übrigen Fällen ist das Element R_i mit einer Wahrscheinlichkeit von höchstens $((1-\varepsilon)\,\mathbf{E}[X_i])^{-1} = \left(\mathbf{E}[X_i] - \sqrt{2\,\mathbf{E}[X_i]\ln m}\right)^{-1}$ pareto-optimal.

Um nun alle möglichen Position von R_i zu betrachten, muss wiederum über alle möglichen Gewichte integriert werden. Dabei entsteht das folgende Integral.

$$\mathbf{E_{approx}}[O_i] = \int_0^1 \frac{1}{p_i(m-1)+1-\sqrt{2(p_i(m-1)+1)\ln m}}\, dp_i + \frac{1}{m}$$

Damit dieses Integral zu lösen ist, werden analog zu 3.3.1 nur die Elemente betrachtet, deren Gewichte nicht unterhalb einer bestimmten Grenze liegen. Die Grenze liegt diesmal aus rechnerischen Gründen bei $v = \frac{4 \ln m}{m-1}$. Über alle Elemente ist die erwartete Anzahl der Elemente, die mindestens ein Gewicht kleiner als v besitzen, kleiner als $md \cdot v \approx 4d \ln m$. Aufgrund dieser Annahme gilt , dass jedes w_{ik} mindestens so groß wie $v = \frac{4 \ln m}{m-1}$ ist.

Da der Term im Integranden der Kehrwert aus der abgeschätzten Anzahl der überdeckten Elemente ist, kann er nicht größer als Eins werden. Allerdings ist es aus rechnerischen Gründen wünschenswert, dass die konstante Eins aus dem Nenner verschwindet. Daher wird das Integral für kleine p_i mit Eins abgeschätzt. Für größere p_i kann auch der nach unten abgeschätzte Nenner nicht mehr zu klein werden. Die Variable v wurde so gewählt, dass sie sich auch zur Aufsplittung des Integrals eignet.

$$\begin{aligned}
\mathbf{E_{approx}}[O_i] &\leq \int_v^1 \frac{dp_i}{p_i(m-1)+1-\sqrt{2(p_i(m-1)+1)\ln m}} + \\
&\quad \int_0^v 1\, dp_i + \frac{1}{m} + dv \\
&\leq \int_v^1 \frac{dp_i}{p_i(m-1)-\sqrt{3p_i(m-1)\ln m}} + \frac{1}{m} + (d+1)v
\end{aligned}$$

Die Abschätzung des Nenners gilt für alle $m \geq 2$.

Nun wird p_i durch $\prod_{k=1}^{d} w_{ik}$ ersetzt und anschließend für jedes Gewicht w_{ik} eine Substitution $t_k^2 = w_{ik}$ durchgeführt. Dabei gilt folgende Substitutionsregel.

$$\int f(w_{ik})\, dw_{ik} = \int 2t_i f(t_i^2)\, dt_i \text{ , da } \frac{dw_{ik}}{dt_i} = 2t_i$$

Weiterhin sei $t = \prod_{k=1}^{d} t_k$ und $c = \sqrt{\frac{3 \ln m}{m-1}}$. Nach der Substitution gilt natürlich für jedes t_i sowie das Produkt t, das sie nicht kleiner als $\sqrt{v}$ sind.

$$\begin{aligned}
I &= \int_v^1 \frac{\prod_{k=1}^{d} dw_i k}{\left(\prod_{k=1}^{d} w_{ik}\right)(m-1) - \sqrt{3\left(\prod_{k=1}^{d} w_{ik}\right)(m-1)\ln m}} \\
&= \int_{\sqrt{v}}^1 \frac{2^d t}{t^2(m-1) - t\sqrt{3(m-1)\ln m}} \prod_{k=1}^{d} dt_k \\
&= \int_{\sqrt{v}}^1 \frac{2^d}{t(m-1) - (m-1)\sqrt{\frac{3\ln m}{m-1}}} \prod_{k=1}^{d} dt_k \\
&= \frac{2^d}{m-1} \int_{\sqrt{v}}^1 \frac{\prod_{k=1}^{d} dt_k}{\left(\prod_{k=1}^{d} t_k\right) - c}
\end{aligned}$$

Dieses Mehrfachintegral lässt sich nun einmal analytisch integrieren, wie folgende symbolische Rechnung auf zwei Variablen veranschaulicht.

$$\int\int \frac{1}{xy - c}\, dx\, dy = \int \frac{\ln(xy - c)}{y}\, dy \text{ , mit } xy - c > 0$$

Das dabei entstehende Integral ist nicht auf algebraischem Weg zu lösen ist. Daher wird das betrachtete Integral zuerst einmal nur zur ersten Variable t_1 integriert. Danach wird der so entstandene Ausdruck derart abgeschätzt, dass das darauf folgende Integral analytisch zu lösen ist.

$$\begin{aligned} I' &= \int_{\sqrt{v}}^{1} \frac{\prod_{k=1}^{d} dt_k}{\left(\prod_{k=1}^{d} t_k\right) - c} \\ &= \int_{\sqrt{v}}^{1} \frac{\ln\left(1\left(\prod_{k=2}^{d} t_k\right) - c\right) - \ln\left(\sqrt{v}\left(\prod_{k=2}^{d} t_k\right) - c\right)}{\prod_{k=2}^{d} t_k} \prod_{k=2}^{d} dt_k \end{aligned}$$

Die Funktion im Zähler wird nun durch ihr Maximum im Intervall $[\sqrt{v}; 1]$ abgeschätzt. Damit fällt die Abhängigkeit des Zählers von irgendeinem t_i weg.

Da die Logarithmusfunktion streng monoton steigend ist, nimmt der Zähler den größten Wert dann an, wenn der erste Summand maximal und der zweite minimal wird. Dazu muss das erste Produkt der Variablen t_i maximal gewählt werden. Dieses Maximum liegt bei Eins, da keine der Integrationsvariablen größer als die obere Grenze des Integrals werden kann. Der kleinste Wert, den das zweite Produkt der Variablen t_i und $\sqrt{v}$ annehmen kann, ist aber $\sqrt{v}$ selbst. Dies ist nämlich nach der Aufsplittung des ursprünglichen Integrals der kleinste Wert, den das Produkt aller t_i erreichen kann.

Nach dieser Abschätzung des Zählers lässt er sich als Konstante aus dem Integral ziehen und der übrig gebliebene Term kann leicht integriert werden. Hierzu wird analog zu 3.3.1 das Mehrfachintegral in ein Produkt einzelner

Integrale zerlegt, die den natürlich Logarithmus als Stammfunktion haben.

$$\begin{aligned}
I' &\leq \int_{\sqrt{v}}^{1} \frac{\ln(1-c) - \ln(\sqrt{v}-c)}{\prod_{k=2}^{d} t_k} \prod_{k=2}^{d} dt_k \\
&= \ln\left(\frac{1-c}{\sqrt{v}-c}\right) \int_{\sqrt{v}}^{1} \frac{\prod_{k=2}^{d} dt_k}{\prod_{k=2}^{d} t_k} \\
&= \ln\left(\frac{1-c}{\sqrt{v}-c}\right) (\ln 1 - \ln(\sqrt{v}))^{d-1} \\
&= \ln\left(\frac{m-1-\sqrt{3(m-1)\ln m}}{\sqrt{4(m-1)\ln m}-\sqrt{3(m-1)\ln m}}\right) \left(\frac{1}{2}\ln\left(\frac{m-1}{4\ln m}\right)\right)^{d-1}
\end{aligned}$$

Die beiden Loagrithmen werden nun abgeschätzt, da die meisten Terme die Größenordnung des Ergebnisses nicht beeinflussen. Für den ersten Logarithmus ergibt sich aufgrund seiner monotonen Steigung folgende Abschätzung. Diese gilt für $m \geq 15$, da ansonsten der Nenner kleiner als Eins ist.

$$\begin{aligned}
& \ln\left(\frac{m-1-\sqrt{3(m-1)\ln m}}{\sqrt{4(m-1)\ln m}-\sqrt{3(m-1)\ln m}}\right) \\
= \; & \ln\left(m-1-\sqrt{3(m-1)\ln m}\right) - \\
& \ln\left(\sqrt{4(m-1)\ln m}-\sqrt{3(m-1)\ln m}\right) \\
\leq \; & \ln(m-1)
\end{aligned}$$

Die Abschätzung des zweiten Logarithmus ist einfacher, da dort der Nenner bereits für $m \geq 2$ größer als Eins wird.

$$\ln\left(\frac{m-1}{4\ln m}\right) = \ln(m-1) - \ln(4\ln m) \leq \ln(m-1)$$

Mit diesen Abschätzungen ergibt sich für das ursprüngliche Integral die folgende Näherung.

$$\begin{aligned} I &= \frac{2^d}{m-1} I' \\ &\leq \frac{2^d}{m-1} \frac{1}{2^{d-1}} \ln^d(m-1) \\ &= \frac{2}{m-1} \ln^d(m-1) \end{aligned}$$

Eingesetzt in den gesamten Ausdruck folgt diese obere Schranke für den Erwartungswert von O_i.

$$\begin{aligned} \mathbf{E}_{\mathbf{approx}}[O_i] &\leq \int_v^1 \frac{\prod_{k=1}^{d} dw_i k}{p_i(m-1) - \sqrt{3p_i(m-1)\ln m}} + \frac{1}{m} + (d+1)v \\ &\leq \frac{2}{m-1} \ln^d m - 1 + \frac{1}{m} + (d+1)v \\ &= \frac{2}{m-1} \left(2\ln m + 2d \ln m + \ln^d(m-1)\right) + \frac{1}{m} \end{aligned}$$

Über die Linearität des Erwartungswertes lässt sich nun die erwartete Anzahl pareto-optimaler Lösungen bestimmen.

$$\begin{aligned} \mathbf{E}[q] &= \mathbf{E}_{\mathbf{approx}} \left[\sum_{i=1}^{m} O_i \right] = \sum_{i=1}^{m} \mathbf{E}_{\mathbf{approx}}[O_i] = m\, \mathbf{E}_{\mathbf{approx}}[O_i] \\ &\leq \frac{2m}{m-1} \left(2\ln m + 2d \ln m + \ln^d(m-1)\right) + 1 \\ &= O(\log^d m) \end{aligned}$$

3.4 Die Konzentration der Anzahl pareto-optimaler Lösungen

Durch den Einsatz der Chernoff-Schranke im letzten Abschnitt lässt sich für die erwartete Anzahl pareto-optimaler Lösungen zeigen, dass diese Anzahl stark konzentriert ist. Fügt man dem ε einen konstanten Faktor a unter der Wurzel hinzu, lässt sich die Wahrscheinlichkeit, dass weniger als $(1-\varepsilon)\,\mathbf{E}[X_i]$ Elemente überdeckt werden, beliebig nahe an 1 drücken. Allerdings muss ε noch leicht angepasst werden, damit es für große a nicht zu groß wird.

$$\varepsilon = \sqrt{\frac{2a\ln^{d/2} m}{p_i(m-1)+1}}$$

$$\mathbf{Pr}[X_i \leq (1-\varepsilon)\,\mathbf{E}[X_i]] \;\leq\; \exp\left(-\mathbf{E}[X_i]\frac{\varepsilon^2}{2}\right) = \left(\frac{1}{m^a}\right)^{\ln^{d/2-1} m} \leq \frac{1}{m^a}$$

Wie wir sehen werden, ändert sich die Größenordnung der Anzahl pareto-optimaler Lösungen q dadurch nur unwesentlich. Die Abschätzung für den Erwartungswert von O_i liefert analog zur Rechnung in 3.3.2 folgenden Term.

$$\begin{aligned}
\mathbf{E}[O_i] \;&\leq\; \int_v^1 \frac{\prod_{k=1}^{d} dw_{ik}}{p_i(m-1)+1-\sqrt{2a(p_i(m-1)+1)\ln^{\frac{d}{2}} m}} + \\
&\quad \int_0^v 1\,dp_i + \frac{1}{m^a} + dv \\
&\leq\; \int_v^1 \frac{\prod_{k=1}^{d} dw_{ik}}{p_i(m-1)-\sqrt{3ap_i(m-1)\ln^{\frac{d}{2}} m}} + \frac{1}{m^a} + (d+1)v
\end{aligned}$$

Die Abschätzung des Nenners bleibt von der Konstanten a fast unberührt. Allerdings müssen die Hilfsvariablen v und c angepasst werden, damit die beiden Nenner nicht negativ werden können. Die Variable v nimmt nun den Wert $\frac{4a\ln^{0,75d} m}{m-1}$ an und c ist gleich $\sqrt{\frac{3a\ln^{\frac{d}{2}} m}{m-1}}$. Die Substitution und das Unterteilen des Integrals bleiben unverändert. Die restliche Integration verläuft analog. Im Endergebnis taucht die Konstante a dann wieder auf. Das Integral, dass nach den Substitutionen entsteht, wird dann wieder nach oben abgeschätzt. Die Abschätzung gilt wiederum für alle $m > m_0$ für ein geeignetes m_0, das diesmal von a und d abhängt und nur durch das Lösen eines Polynoms dritten Grades algebraisch hergeleitet werden kann.

$$\begin{aligned}
I' &= \ln\left(\frac{1-c}{\sqrt{v}-c}\right)(\ln 1 - \ln(\sqrt{v}))^{d-1} \\
&= \ln\left(\frac{m-1-\sqrt{3a(m-1)\ln^{\frac{d}{2}} m}}{\sqrt{4a(m-1)\ln^{\frac{3}{4}d} m} - \sqrt{3a(m-1)\ln^{\frac{d}{2}} m}}\right)\left(\frac{1}{2}\ln\left(\frac{m-1}{4\ln^{\frac{3}{4}d} m}\right)\right)^{d-1} \\
&\leq \frac{1}{2^{d-1}}\ln^d(m-1)
\end{aligned}$$

Hierbei ist zu beachten, dass bei der Substitution die Konstante 2^d vor das Integral gezogen wurde und damit den Term $2^{-(d-1)}$ wieder unabhängig von d macht. Für den Erwartungswert von O_i ergibt sich damit folgender Ausdruck, der wiederum bis auf die zusätzliche Variable a identisch zu dem Term aus 3.3.2 ist.

$$\mathbf{E}[O_i] \leq \frac{2}{m-1}\left(2ad\ln m + 2a\ln m + \ln^d(m-1)\right) + \frac{1}{m^a}$$

Daraus resultiert folgendes Endergebnis für den Erwartungswert von q.

$$\begin{aligned}
\mathbf{E}[q] &= \mathbf{E}\left[\sum_{i=1}^{m} O_i\right] + 1 + mdv \\
&= \sum_{i=1}^{m} \mathbf{E}[O_i] + 1 + mdv = m\,\mathbf{E}[O_i] + 1 + mdv \\
&\leq \frac{2m}{m-1}\left(2a\ln^{0,75d} m + 2ad\ln^{0,75d} m + \ln^d(m-1)\right) + \frac{1}{m^{a-1}} \\
&= O(ad\log^{0,75d} m + \log^d m)
\end{aligned}$$

Da a als konstant angenommen werden kann, unterscheidet sich die Größenordnung im asymptotischen Bereich nicht von dem Ergebnis aus 3.3.2.

Die Größenordnung der erwarteten Anzahl an pareto-optimalen Lösungen hat sich also nicht verändert. Es wurde somit gezeigt, dass der Erwartungswert mit hoher Wahrscheinlichkeit erreicht wird. Es muss lediglich noch sichergestellt werden, dass die Anzahl der Elemente, die nur Gewichte kleiner v haben, nicht zu groß wird.

Die Wahrscheinlichkeit, dass ein Gewicht kleiner $v = \frac{4a\ln^{0,75d} m}{m-1}$ ist, beträgt genau v, da die Gewichte uniform zufällig aus dem Intervall $[0;1]$ gewählt sind. Die Wahrscheinlichkeit, dass ein Element mindestens ein Gewicht kleiner v hat, beträgt bei d Gewichten höchstens dv. Abschließend ist die Wahrscheinlichkeit, dass mehr als $\ln^d m$ Elemente Gewichte kleiner v haben und damit die Größenordnung der Anzahl der pareto-optimalen Lösungen beeinflussen, höchstens $(dv)^{\ln^d m}$. Betrachtet man bei diesem Term nur die Abhängigkeit von m, da a und d als konstant angenommen werden können, ergibt sich eine Größenordnung von $O\left(\left(\frac{\ln^{0,75d} m}{m}\right)^{\ln^d m}\right)$. Dieser Ausdruck fällt wesentlich schneller als die Wahrscheinlichkeit $1/m^{a\ln^{d/2-1} m}$ aus der Chernoff-Schranke, da der mit m wachsende Exponent größer ist.

Insgesamt wurde also in diesem Abschnitt ein Konzentrationsergebnis für den Erwartungswert aus 3.3.2 bewiesen. Die Anzahl der pareto-optimalen Lösungen liegt mit einer Wahrscheinlichkeit von mehr als $1 - 1/m^a$ in der Größenordnung $O(ad \log^{0,75d} m + \log^d m)$.

Leider ist es auch mit diesem Konzentrationsergebnis nicht möglich, einen Erwartungswert für die Laufzeit des trivialen Algorithmus für das mehrdimensionale Rucksackproblem aus 2.3 herzuleiten. Dieser Algorithmus hat eine Laufzeit von $O(nq^2)$. Selbst wenn der Erwartungswert von q noch so stark konzentriert ist, lassen sich unwahrscheinliche Ausreißer nicht ausschließen. So ist nach der Markov-Ungleichung die Wahrscheinlichkeit, dass eine Eingabe mehr als t pareto-optimale Lösungen hat, durch $\mathbf{E}[q]/t$ nach oben beschränkt. Trotzdem treiben solche Ausreißer bereits für $t = m^\delta$ für ein beliebiges $\delta > 0$ die Laufzeit des trivialen Algorithmus in die Höhe. So ergibt für dieses Beispiel ein Beitrag zur Laufzeit von $\mathbf{Pr}[q \geq m^\delta] \cdot m^{2\delta}$. Mit der Markov-Ungleichung und dem Ergebnis aus 3.3.2 folgt hieraus ein Beitrag zur Laufzeit in einer Größenordnung von $O(nm^\delta \log^d m)$. Dies entspricht mit $m = 2^n$ einer exponentiellen Laufzeit.

4 Ergebnisse aus der Analyse und den Testläufen

4.1 Erwartete Laufzeit des verallgemeinerten Nemhauser/Ullmann-Algorithmus

Die erwartete Laufzeit des verallgemeinerten Nemhauser/Ullmann-Algorithmus lässt sich nicht direkt angeben. Aufgrund der Abhängigkeit der Rucksäcke im mehrdimensionalen Rucksackproblem ist dort eine andere Anzahl an Pareto-Optima zu erwarten als bei der abgeschwächten Version des Problems mit $m = 2^n$ unabhängigen Elementen. Anhand von Testläufen wird sich zeigen, dass die fehlende Unabhängigkeit der Rucksäcke im mehrdimensionalen Rucksackproblem die Anzahl der Pareto-Optima vergrößert. Wahrscheinlich steht die Anzahl der Pareto-Optima aber in einem polynomiellen Zusammenhang mit der Anzahl beim abgeschwächten Problem. Darauf weisen die weiter unten beschriebenen Ergebnisse der Testläufe für $d = 2$ hin. Auch Beier kommt in [Bei04] auf dieses Ergebnis für den zweidimensionalen Fall, das er ebenfalls aus experimentellen Werten abgeleitet hat.

4.2 Vergleich der erwarteten Anzahlen der pareto-optimalen Lösungen

In diesem Abschnitt sind die aus Abschnitt 3.3.1 exakt bestimmten Erwartungswerte für unabhängige Rucksäcke mit dem algebraischen System *maple* berechnet und der Approximation aus Abschnitt 3.3.2 gegenübergestellt worden. Das Ergebnis in Abbildung 9 zeigt die Approximation und die exakten

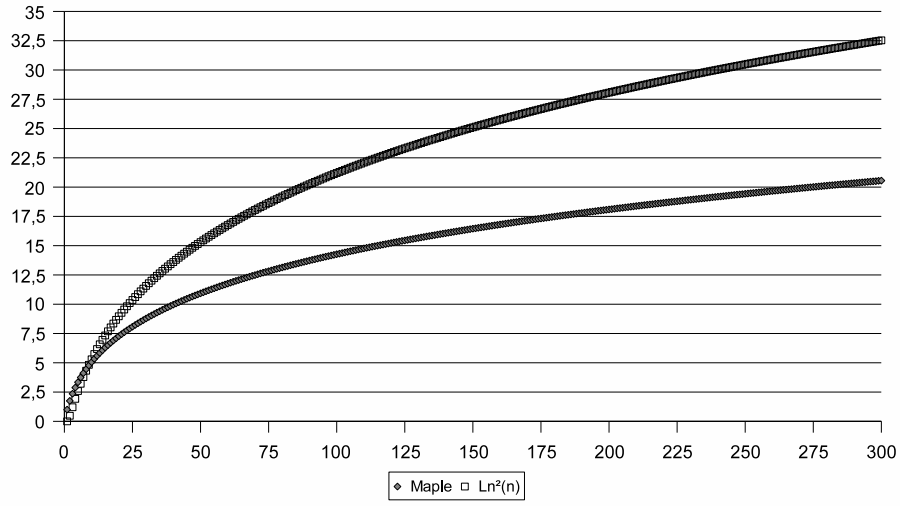

Abbildung 9: Die erwartete Anzahl der Pareto-Optima in Abhängigkeit von n für das mehrdimensionale Rucksackproblem aus der Approximation und der Auswertung des exakten Erwartungswertes mit *maple*

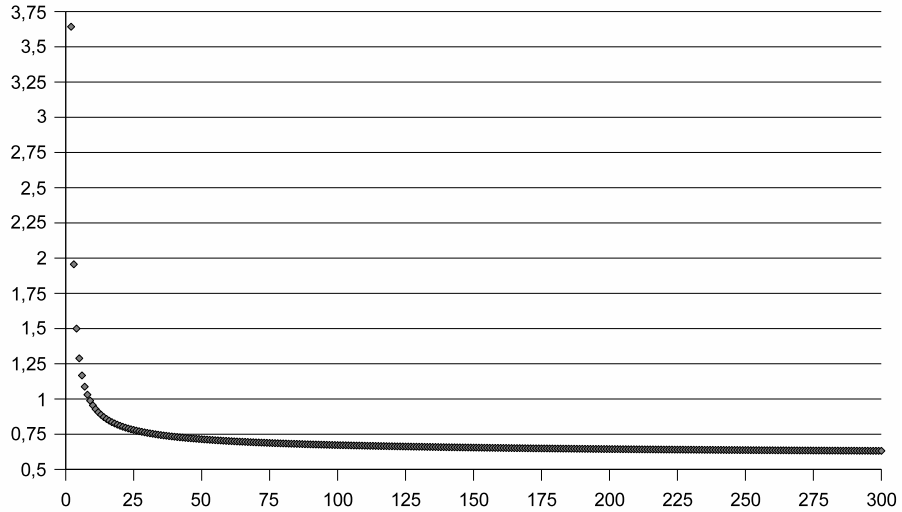

Abbildung 10: Der Quotient aus der exakten Berechnung und der Approximation des Erwartungswertes in Abhängigkeit von n für das mehrdimensionale Rucksackproblem

Berechnungen. Hieran lässt sich aber noch wenig über die Güte der Approximation aussagen. Daher ist in Abbildung 10 zusätzlich der Quotient aus den exakten und den approximierten Erwartungswerten dargestellt. Es zeigt sich, dass dieser Quotient scheinbar einer konstanten Grenze entgegen strebt. Somit ist die Approximation nur um einen konstanten Faktor vom exakten Erwartungswert entfernt.

4.3 Betrachtung von Resultaten aus Testläufen

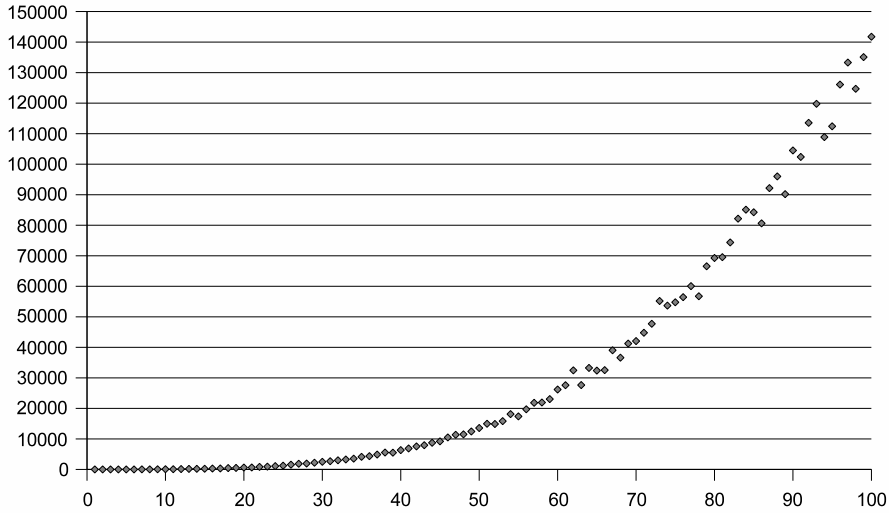

Abbildung 11: Die Anzahl der Pareto-Optima q aus den Experimenten in Abhängigkeit von n für das zweidimensionale Rucksackproblem

Für das zweidimensionale Rucksackproblem wurden für $n = 1 \ldots 50$ jeweils 100 und für $n = 51 \ldots 100$ jeweils 50 Testläufe durchgeführt. Dazu wurde der verallgemeinerte Nemhauser/Ullmann-Algorithmus aus Abschnitt 2.2 implementiert. Als Datenstruktur wurde hierbei ein Red-Black-Tree verwendet. Der Algorithmus gibt die Anzahl der gefundenen Pareto-Optima aus. Das Resultat ist in Abbildung 11 zu sehen. Eine Ausgleichskurve legt ein Poly-

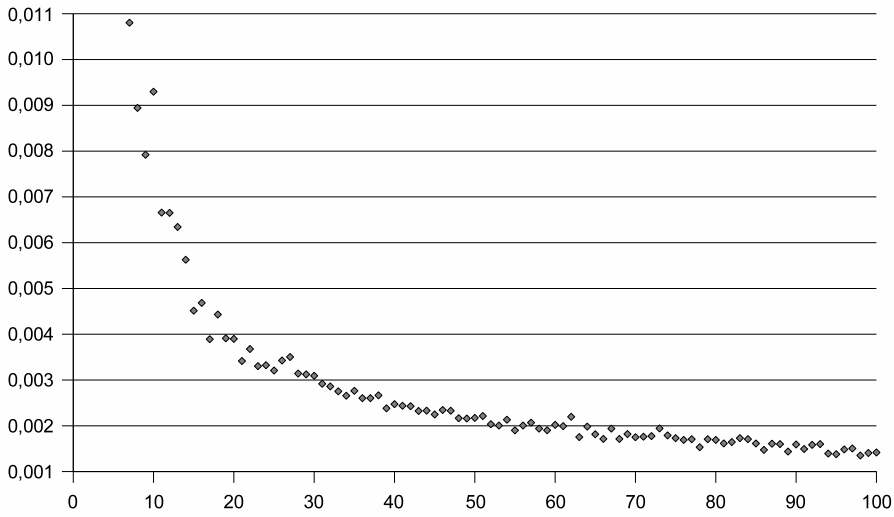

Abbildung 12: Der Quotient aus der praktisch ermittelten Anzahl q der Pareto-Optima und n^4 in Abhängigkeit von n für das zweidimensionale Rucksackproblem

nom vierten Grades nahe. Daher wurde in Abbildung 12 der Quotient aus den Ergebnissen des Testlaufs und n^4 gebildet. Hieran lässt sich gut erkennen, dass die Anzahl der pareto-optimalen Lösungen beim zweidimensionalen Rucksackproblem höchstens mit der vierten Potenz der Anzahl der Elemente wächst, da der Quotient streng monoton fällt. Umfangreichere Tests, die dieses Ergebnis untermauern, sind in [Bei04] zu finden.

4.4 Vergleich von Theorie und Praxis

Um die theoretischen Ergebnisse der Analyse mit den praktischen Werten aus den Experimenten zu vergleichen, ist eine Annahme über den Zusammenhang der Anzahl der Pareto-Optima beim Rucksackproblem q' und der Anzahl bei der abgeschwächten Variante q notwendig. Die Verteilung von q' in Abbildung

11 scheint, wie im letzten Abschnitt dargestellt, ein Polynom vierten Grades zu sein. Die Analyse im vereinfachten Modell ergibt eine Abhängigkeit von $O(\log^2 m)$. Dies entspricht für $m = 2^n$ einer quadratischen Abhängigkeit von n.

Daraus lassen sich nun verschiedene Annahmen über den Zusammenhang von q und q' herleiten. Werden die Ergebnisse aus dem vereinfachten Modell als der Wahrheit recht nahe liegend bewertet, könnte die Anzahl der pareto-optimalen Lösungen im vereinfachten Modell nur um einen polynomiellen Faktor von der Anzahl im Rucksackproblem entfernt liegen.

Annahme 4.1 *Aus einer Anzahl von q pareto-optimalen Lösungen im vereinfachten Rucksackmodell ergeben sich $q' = qn^2$ Pareto-Optima für das mehrdimensionale Rucksackproblem.*

Damit ließe sich für d Gewichte folgende Schlussfolgerung ziehen.

Folgerung 4.1 *Die Anzahl der pareto-optimalen Lösungen im mehrdimensionalen Rucksackproblem mit d Gewichten und einem Profit beträgt $O(n^{d+2})$.*

Eine wesentlich schwächere Annahme über den Zusammenhang von q und q' geht davon aus, das die Anzahl pareto-optimaler Lösungen beim Rucksackproblem ein Polynom in der Anzahl der Lösungen des vereinfachten Modells ist.

Annahme 4.2 *Aus einer Anzahl von q pareto-optimalen Lösungen im vereinfachten Rucksackmodell ergeben sich $q' = q^2$ Pareto-Optima für das mehrdimensionale Rucksackproblem.*

Aus dieser Annahme folgt dann für den d-dimensionalen Fall ein anderes Resultat.

Folgerung 4.2 *Die Anzahl der pareto-optimalen Lösungen im mehrdimensionalen Rucksackproblem mit d Gewichten und einem Profit beträgt $O(n^{2d})$.*

Für den Fall $d = 2$ sind die beiden Annahmen natürlich gleichwertig. Um die stärkere Annahme 4.1 zu untermauern, wären noch weitere Experimente für höherdimensionale Fälle ($d > 2$) notwendig. Allerdings legen beide Annahmen nahe, dass die Anzahl der Pareto-Optima im mehrdimensionalen Rucksackproblem in der Größenordnung $n^{\Theta(d)}$ liegt.

5 Fazit

Auf uniformen verteilten Eingaben lässt sich vermuten, dass die erwartete Anzahl pareto-optimaler Lösungen q für das mehrdimensionale Rucksackproblem in der Größenordnung $n^{\Theta(d)}$ liegt. Für den zweidimensionalen Fall, bei dem die Laufzeit des verallgemeinerten Nemhauser/Ullmann-Algorithmus bei $O(nq \log q)$ liegt, ergibt sich somit eine Laufzeit in der Größenordnung $n^{\Theta(d)}$. Dabei wurde $\log q$ grob mit n abgeschätzt. In höherdimensionalen Fällen lässt sich leider keine Aussage über eine Laufzeit eines nahe liegenden Algorithmus machen. Für den trivialen Algorithmus aus 2.3 mit einer Laufzeit von $O(nq^2)$ ist es mit Hilfe des Ergebnisses für $\mathbf{E}[q]$ selbst mit dem Konzentrationsergebnis nicht möglich, eine Aussage über eine effiziente, erwartete Laufzeit abzuleiten.

Die Ergebnisse der Analysen sind da schon konkreter. Sie gelten zumindest für das vereinfachte Modell für beliebig dimensionale Probleme. Die Experimente für $d = 2$ geben sogar Hinweise auf eine Größenordnung von $O(n^4)$ für die erwartete Anzahl der Pareto-Optima (siehe Abbildung 11) beim zweidimensionalen Rucksackproblem. Damit besteht Grund zur Annahme, die Anzahl der pareto-optimalen Lösungen beim mehrdimensionalen Rucksackproblem steht in einem polynomiellen Zusammenhang mit der Anzahl beim analysierten, vereinfachten Modell.

Für andere Zufallsverteilungen lassen sich wahrscheinlich auf ähnlichem Wege obere Schranken beweisen, wie dies für die uniforme Verteilung im vereinfachten Modell in Kapitel 3 geschehen ist. Eine Betrachtung verschiedener Zufallsverteilungen für die Eingabe beim eindimensionalen Rucksackproblems ist in [BV03] bereits gelungen.

Schwieriger ist wohl eine direkte Übertragung der Analysen auf das Rucksackproblem, da dort Abhängigkeiten der Rucksäcke über die eingepackten Elemente auftreten. Diese verkomplizieren eine probabilistische Analyse sehr. Eine exakte Berechnung des Erwartungswertes ist schon für das abgeschwächte Problem kaum möglich (siehe Abschnitt 3.3.1) und auch eine Approximation für die erwartete Anzahl der Pareto-Optima im mehrdimensionalen Rucksackproblem wird schwierig.

Ein anderer Ansatz wäre, über das in dieser Arbeit gezeigte Ergebnis einen Erwartungswert für das Rucksackproblem abzuleiten, indem ein Zusammenhang zwischen der Anzahl der Pareto-Optima beim Rucksackproblem und der Anzahl bei der abgeschwächten Variante hergestellt und gezeigt wird. Die dabei entstehenden oberen Schranken für das mehrdimensionale Rucksackproblem wären allerdings wahrscheinlich nicht sehr genau. Auch denkbar wäre, über die in [BV03] bewiesene erwartete Stufenhöhe, die sich leicht auch für das mehrdimensionale Rucksackproblem beweisen lässt, eine Schlussfolgerung auf die Anzahl der pareto-optimalen Lösungen zu finden. Dies ist im Verlaufe dieser Arbeit leider nicht gelungen.

Interessant wäre auch eine Untersuchung einer Mehrzieloptimierung. Dort gäbe es nicht nur mehrere Gewichte sondern auch mehrere Profite pro Element. Vielleicht ließe sich zeigen, dass über eine Umwandlung der Profite in zusätzliche Gewichte, z.B. $w_{d+i} = 1 - u_i$, eine Rückführung auf das mehrdimensionale Rucksackproblem mit einem Profit möglich ist.

Zuletzt ist vielleicht auch eine allgemeine Betrachtung von Anzahlen von pareto-optimalen Lösungen für beliebige Probleme aus der Klasse des Integer Programmings möglich. Dabei müssten u.U. die betrachteten Probleme und/oder stochastischen Modelle stark angepasst werden. Ergebnisse dieser

Art wären aber erst dann von mehr als nur theoretischem Interesse, wenn es auch Algorithmen gäbe, die für beliebige Integer Programme effizient alle pareto-optimalen Lösungen bestimmen. Dies ist bereits für das dreidimensionale Rucksackproblem bisher nicht gelungen. Es existieren sogar Probleme, bei denen pareto-optimale Lösungen auftauchen und für die es keinen Algorithmus gibt, der diese auch nur annähernd effizient berechnet. So ist z.B. für das Problem des minimalen Spannbaums mit zwei Gewichten pro Kante bisher kein Ansatz bekannt, alle pareto-optimalen Lösungen effizient bzgl. ihrer Anzahl zu bestimmen.

Anhang

A Sätze aus der Analysis und Stochastik

(Chernoff-Schranke) Satz A.1 *Seien $X_1, \ldots, X_n$ unabhängige Variablen mit $\mathbf{Pr}[X_i = 1] = p_i$ and $\mathbf{Pr}[X_i = 0] = 1 - p_i$. Sei $X = \sum_{i=1}^{n} X_i$ und sei $\mu \geq \mathbf{E}[X]$. Dann gilt für jedes ε mit $1 \geq \varepsilon \geq 0$,*

$$\mathbf{Pr}[X \leq (1-\varepsilon) \cdot \mu] \leq \exp\left(-\frac{\varepsilon^2 \mu}{2}\right).$$

Satz A.2 *Sei X eine beliebige Zufallsvariable und $f(X)$ die Dichtefunktion von X. Sei weiterhin $Y = g(X)$ eine Zufallsvariable, die durch eine Funktion g auf X definiert ist. Dann gilt für den Erwartungswert von Y,*

$$\mathbf{E}[Y] = \int_{-\infty}^{\infty} g(x) \cdot f(x)\, dx.$$

Folgerung A.1 *Sei X eine uniform auf dem Intervall $[0;1]$ verteilte Zufallsvariable. Sei weiterhin $Y = g(X)$ eine Zufallsvariable, die durch eine Funktion g auf X definiert ist. Dann gilt für den Erwartungswert von Y,*

$$\mathbf{E}[Y] = \int_{0}^{1} g(x)\, dx.$$

Literatur

[Bei04] R. Beier, *Probabilistic Analysis of Discrete Optimization Problems*, PhD thesis, Universität Saarbrücken, May 2004.

[BV03] R. Beier and B. Vöcking, *Random knapsack in expected polynomial time*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC), ACM, ed., pp. 232–241, ACM Press, New York, NY, USA, 2003.

[BZ80] E. Balas and E. Zemel, *An algorithm for large zero-one knapsack problems*, in *Operations Research*, vol. 28, 1980, pp. 1130–1154.

[DS04] V. Damerow and C. Sohler, *Extreme points under random noise*, in Proceedings of the 12th Annual Symposium on Algorithms (ESA), S. Albers and T. Radzik, eds., pp. 264–274, Springer, Berlin, Germany, 2004.

[GMS84] A. Goldberg and A. Marchetti-Spaccamela, *On finding exact solution to a zero-one knapsack problem*, in Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC), ACM, ed., pp. 359–368, ACM Press, New York, NY, USA, 1984.

[NU69] G. Nemhauser and Z. Ullmann, *Discrete Dynamic Programming and Capital Allocation*, in *Management Science*, 1969, pp. 494–505.

[RS63] A. Rényi and R. Sulanke, *Über die konvexe hülle von n zufällig gewählten punkten i.*, Zeitschrift für Wahrscheinlichketistheorie 2, pp. 75–84, 1963.

[RS64] A. Rényi and R. Sulanke, *Über die konvexe hülle von n zufällig gewählten punkten ii.*, Zeitschrift für Wahrscheinlichketistheorie 3, pp. 136–147, 1964.