Verschnittoptimierung durch iteriertes Matching

Andreas Fritsch

Diplomarbeit im Studiengang Diplom Mathematik
an der Universität Osnabrück
im Fachbereich Mathematik/Informatik

vorgelegt bei
Professor Dr. Peter Brucker
am 04.01.1994

von
Andreas Fritsch
Glückaufstraße 176
49124 Georgsmarienhütte

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Optimierungsprobleme und insbesondere Verschnittprobleme haben in den vergangenen Jahrhunderten immer wieder das Interesse und den Ehrgeiz der Naturwissenschaftler erregt. Schon vor dem Jahr 1694 stellte Sir Isaac Newton (1643-1727) Vermutungen über das dichteste Packen von "Sphären im dreidimensionalen Raum" an [Saa 1975]. Ein verstärktes Interesse an Problemen der Optimierung entwickelte sich allerdings erst innerhalb der letzten Jahrzehnte. Durch die rasante Entwicklung in der Computertechnik wurde es möglich, komplexe und rechenintensive Probleme innerhalb kürzester Zeit zu lösen. Speziell zur Lösung der sehr komplexen Verschnittprobleme war eine solche Entwicklung dringend erforderlich. Es müssen effektive Verfahren zur Verschnittoptimierung entwickelt werden, da aufgrund der immer stärker steigenden Materialkosten, der zunehmenden Rohstoffverknappung und der wachsenden Umweltbelastung das Entstehen von Abfall bei der Herstellung von Produkten unbedingt vermieden werden muß.

Eine Verbindung zwischen Computertechnik und dem Themengebiet Optimierung läßt sich jedoch nicht nur dadurch herstellen, daß Optimierungsprobleme mit Computern gelöst werden. Die Organisation eines Rechners selbst verursacht Probleme, die der Schedulingtheorie — einem Teilgebiet der Optimierung — zuzuordnen sind. Zur effektiven Nutzung der Hardware muß beispielsweise die Reihenfolge festgelegt werden, in der die einzelnen Prozesse auf der CPU abgearbeitet werden, ohne große "Leerlaufzeiten" entstehen zu lassen. Es muß außerdem eine vernünftige Speicheraufteilung erfolgen, so daß die Daten, auf die ein Programm zugreift, möglichst schnell zur Verfügung stehen.

Man unterscheidet zwei Klassen von Optimierungsproblemen. Bei der ersten Klasse handelt es sich um Optimierungsprobleme mit stetigen Variablen, bei der zweiten Klasse um solche mit diskreten Variablen. Probleme, die der zweiten Klasse angehören, werden als kombinatorische Optimierungsprobleme bezeichnet. Dabei wird in einer endlichen oder möglicherweise abzählbar unendlichen Menge von zulässigen Lösungen eines Problems nach einer optimalen Lösung gesucht.

Allgemein läßt sich ein Optimierungsproblem wie folgt formulieren:

> Liegt ein Problem mit einer Menge von Lösungen, dem Lösungsraum L, vor und ist eine Funktion $f : L \to \mathbb{R}$ gegeben, die eine Lösung $l \in L$ bewertet, so stellt das Auffinden des besten Wertes $f(l)$ bezüglich eines vorgegebenen Zieles ein Optimierungsproblem dar.

Zu den bekanntesten Optimierungsproblemen gehören das Traveling Salesman Problem, das Knapsack Problem, lineare Programmierung und die Pack– und Verschnittprobleme in verschiedenen Dimensionen.

Sehr häufig treten Optimierungsprobleme in der Industrie auf. Die effektive Nutzung von Maschinenressourcen, das Erstellen von Arbeits- und Terminplänen und nicht zuletzt das Maximieren des Gewinns zählen offensichtlich alle zu den Optimierungsproblemen. Besonders häufig sind in der Industrie Verschnittprobleme zu lösen. Dabei geht es darum, die Bestellungen eines Kunden zu erfüllen, indem aus "großen" Rohlingen eines Materials die "kleinen" bestellten Stücke herausgeschnitten werden, so daß alle Bestellungen erfüllt und möglichst wenig Rohlinge verbraucht werden. Da ein Probleme dieser Art sehr komplex sind und eine effektive Lösung "per Hand" nicht mehr möglich ist, lohnt sich in den meisten Fällen die Entwicklung eines Computerprogramms. Dies verdeutlicht auch ein 1975 veröffentlichter Artikel [Was 1975]. Dort wird beschrieben, daß das Softwarepaket POWER zur Reduzierung von Verschnitt in einem stahlverarbeitenden Betrieb erfolgreich eingesetzt wurde. Das Programm POWER wird von der Firma Inbucon Productivity Services vertrieben. Nachdem die Firma Falcon Catering Company POWER Ende des Jahres 1974 installierte, stellte sie einen Rückgang des bei der Herstellung ihrer Produkte entstehenden Verschnitts um über 50 Prozent fest [Can 1979].

In dieser Arbeit wird ein iteratives Verfahren entwickelt, mit dessen Hilfe sich "kleine" Rechtecke aus "großen" Rechtecken ausschneiden lassen, wobei möglichst wenig Verschnitt verursacht werden soll. Es wird dabei auf Randbedingungen eingegangen, wie sie sich für dieses Problem in der Glasindustrie stellen. In Kapitel 2 soll zunächst ein Überblick über Verschnittprobleme im allgemeinen gegeben werden. Speziell für zweidimensionale Verschnittprobleme werden verschiedene Lösungsansätze vorgestellt. Darüber hinaus erfolgt eine Komplexitätsuntersuchung. Das dritte Kapitel beschäftigt sich mit Verschnittproblemen in der Glasindustrie. Es wird auf die Herstellung und die Probleme bei der Verarbeitung von Glas eingegangen. Eine genaue Formulierung des in dieser Arbeit behandelten Problems und die grundlegende Lösungsidee werden in Kapitel 4 angegeben. Im fünften Kapitel wird das im Rahmen dieser Diplomarbeit entwickelte Verfahren vorgestellt. Die Implementierung in der Programmiersprache C zeigt, daß sich mit diesem Verfahren gute Lösungen für das Verschnittproblem finden lassen. Testergebnisse und ein Vergleich zu den Ergebnissen eines kommerziellen Programms werden in Kapitel 6 aufgeführt. Das siebte Kapitel zeigt zusätzliche Nebenbedingungen auf, wie sie in der Praxis zu berücksichtigen sind, auf die aber in den Kapiteln 4 und 5 noch nicht eingegangen wurde. Es werden einige Ideen angegeben, wie diese zusätzlichen Anforderungen erfüllt werden können. Die Arbeit schließt mit einer Zusammenfassung in Kapitel 8.

2 Pack– und Verschnittprobleme

2.1 Allgemeiner Überblick

In der Praxis treten sehr häufig Pack- bzw. Verschnittprobleme auf. Bei näherer Untersuchung der jeweiligen Aufgabenstellungen wird deutlich, daß die Natur dieser beiden Probleme sich sehr ähnelt. Packprobleme treten überall dort auf, wo kleine Einheiten einer bestimmten Form in einem gegebenen Behälter plaziert werden sollen. Das Ziel dabei ist, von dem Behälter so wenig Platz wie nötig zu beanspruchen. Bei Verschnittproblemen läßt sich der Behälter als Rohling eines bestimmten Materials und die kleine Einheit als ein bestelltes Stück desselben Materials interpretieren. Ziel ist dann, die bestellten Stücke aus den Rohlingen auszuschneiden und dabei so wenig Abfall — d.h. nicht weiter verwertbare Rohlingsteile — wie möglich zu erzeugen. Eine enge Verwandtschaft dieser beiden Problemstellungen ist offensichtlich. Deshalb wird auch in der Literatur in vielen Fällen von einem Packproblem gesprochen, obwohl es sich eigentlich um ein Verschnittproblem handelt. In den meisten Fällen läßt sich die Aufgabenstellung so formulieren, daß anstelle eines Packproblems von einem Verschnittproblem (und umgekehrt) gesprochen werden kann. Es kann jedoch vorkommen, daß durch verschiedene Nebenbedingungen, die in konkreten Fällen berücksichtigt werden müssen, diese Umformulierung nur mit großem Aufwand möglich ist.

Pack- und Verschnittprobleme werden in verschiedene Kategorien aufgeteilt, je nachdem welche Dimensionalität das Problem besitzt. Soll beispielsweise ein LKW mit möglichst vielen Kisten, die von unterschiedlicher Größe sein dürfen, beladen werden, so spricht man von einem dreidimensionalen Packproblem. Die Aufteilung der Grundfläche eines Integrierten Schaltkreises (IC) in einzelne Zonen, in denen die Bauelemente des IC untergebracht werden sollen, entspricht einem zweidimensionalen Packproblem. Kurze Stäbe von langen Stangen abzuschneiden und dabei möglichst wenig Stangen zu verbrauchen, entspricht einem eindimensionalen Verschnittproblem.

Diese Arbeit befaßt sich mit einem zweidimensionalen Verschnittproblem. Das erstemal tauchten Verschnittprobleme in der Operations Research Literatur im Jahre 1957 auf. Eisemann untersuchte das eindimensionale Problem, von breiten Rollen eines Materials schmale Rollen abzuschneiden [Eis 1957]. Diese Aufgabenstellung tritt beispielsweise in der Papier-, Textil- und Röhrenindustrie auf. Zu ähnlichen Ergebnissen wie Eisemann kam auch schon Kantorovic im Jahre 1939, jedoch blieb sein in russischer Sprache veröffentlichter Artikel in der westlichen Welt zunächst weitgehend unbekannt [Kan 1939].

Zweidimensionale Verschnittprobleme treten häufig in der Holz-, Glas- und Textilindustrie auf. Die einzelnen Aufgabenstellungen unterscheiden sich zum einen dadurch, welche Formen aus dem Material herausgeschnitten werden sollen, zum anderen durch die Beschaffenheit des Grundmaterials. Während in der Textilindustrie oft unregelmäßige Formen aus langen Stoffbahnen ausgeschnitten werden sollen, beschränkt sich die Holz- und vor allem die Glasindustrie darauf, rechteckige Formen aus großen Lagerplatten auszuschneiden. Dies liegt sowohl an den eingeschränkten Möglichkeiten der Herstellung und Verarbeitung von Glas, als auch der späteren Verwendung der ausgeschnittenen Formen (vgl. Kapitel 3).

2.2 Ein– und zweidimensionale Verschnittprobleme

Bevor im nächsten Abschnitt die Komplexität von Verschnittproblemen untersucht wird, soll hier zunächst noch geklärt werden, was genau sich hinter dem Begriff Problem bzw. ein- oder zweidimensionales Verschnittproblem verbirgt. Garey und Johnson definieren den Begriff Problem folgendermaßen [Gar 1979]:

Definition 2.1 (Problem)

> Ein *Problem* ist eine spezielle Fragestellung, welche von diversen Parametern (auch Variable genannt) abhängt. □

Ein Problem wird durch zwei Punkte beschrieben:

1. durch eine konkrete Beschreibung seiner Parameter.
2. durch Festlegen der Eigenschaften, die eine Lösung des Problems besitzen soll.

Hinter dem Ausdruck Problem verbirgt sich also genau genommen eine ganze Problemklasse, da die Werte der Variablen nicht von Anfang an gegeben sind. Wird für jeden Parameter eines Problems ein konkreter Wert festgelegt, so bilden diese Werte eine *Instanz* des Problems.

Diese Begriffsbildung soll nun anhand eines Beispiels erläutert werden.

Problem 2.1 (eindimensionales Verschnittproblem)

> Gegeben sei ein unendlicher Vorat $S_1, S_2, S_3 \ldots$ an Stangen eines bestimmten Materials. Sei L die ganzzahlige Länge einer Stange. Von diesen Stangen sollen n bestellte Stäbe mit den Längen $l_1, l_2, \ldots, l_n$, wobei $l_i \leq L$ für alle $1 \leq i \leq n$ gelte, abgeschnitten werden. Ziel ist, die bestellten Stäbe zu erzeugen und dabei so wenig Stangen wie möglich zu verbrauchen. □

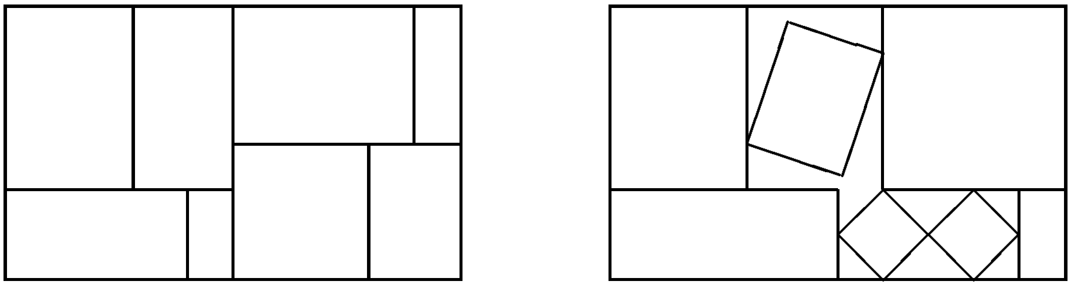
Eine Instanz für dieses Problem, oder besser gesagt diese Problemklasse, kann dann zum Beispiel folgendermaßen angegeben werden:

- $L = 20$ (Länge der Stangen)
- $n = 6$ (Anzahl der Stäbe)
- $l_1 = 9$, $l_2 = 15$, $l_3 = 5$, $l_4 = 5$, $l_5 = 3$, $l_6 = 2$ (Längen der Stäbe)

In der Literatur wird dieses Verschnittproblem oft als Packproblem formuliert. Dabei werden die Stangen als Behälter fester Breite und Höhe angesehen. Die Stäbe werden als Kästen interpretiert, die die gleiche Breite besitzen wie der Behälter und deren Höhe kleiner ist als die Höhe des Behälters. Die Kästen sollen so auf die Behälter verteilt werden, daß alle Kästen verstaut sind und möglichst wenig Behälter benötigt werden. Die Äquivalenz dieser beiden Aufgabenstellungen ist offensichtlich. Das Packproblem bezeichnet man auch als *eindimensionales Bin–Packing*.

Die zweidimensionalen Verschnittprobleme, wie sie im weiteren Verlauf der Arbeit untersucht werden, werden sich alle darauf beschränken, Rechtecke aus einem gegebenen Material herauszuschneiden. Andere Formen, wie zum Beispiel Kreise oder beliebige Polygone, bleiben unberücksichtigt.

Wie am Ende des Abschnitts 2.1 schon erwähnt, werden in einigen Aufgabenstellungen die Rechtecke aus vorgegebenen Platten und nicht aus unendlich langen Bahnen ausgeschnitten. Diese Platten werden auch als *Rohlinge* bezeichnet. Für einen Rohling und eine Menge auszuschneidender Rechtecke kann dann ein *Schnittmuster* angegeben werden. Dies geschieht, indem entweder die einzelnen Schnitte mit Schnittansatzpunkt, Schnittrichtung und Länge oder die genaue Plazierung und Ausrichtung der Rechtecke innerhalb des Rohlings angegeben wird. Anhand des Schnittmusters läßt sich dann ersehen, wie der Rohling zerteilt werden muß, um die kleinen Rechtecke zu erhalten.



a) orthogonal b) nicht orthogonal

Abbildung 1: Orthogonale und nicht orthogonale Schnittmuster

Beim Schneiden gilt es, zwischen verschiedenen Arten zu unterscheiden. Zum einen differenziert man zwischen *orthogonalen* und *nicht orthogonalen* Schnitten. Erstere verlaufen immer parallel zu einer Seite des Rohlings, letztere dürfen in beliebigen Richtungen durch den Rohling verlaufen. Diese unterschiedlichen Schnittarten sind in Abbildung 1 verdeutlicht. Eine spezielle Klasse der orthogonalen Schnitte stellen die *Guillotineschnitte* dar. Als Guillotineschnitte werden solche Schnitte bezeichnet, die an einem Rand des Rohlings starten und in gerader Linie bis zum gegenüberliegenden Rand verlaufen. Abbildung 2 verdeutlicht, wie ein Rohling durch Guillotineschnitte zerteilt werden kann. Die Nummern geben dabei eine Reihenfolge an, in der die einzelnen Schnitte durchgeführt werden können. Abbildung 3 zeigt ein Schnittmuster, das nicht durch Guillotineschnitte realisiert werden kann.

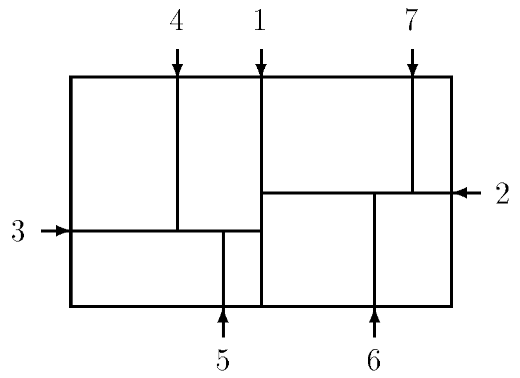


Abbildung 2: Durch Guillotineschnitte realisierbares Schnittmuster

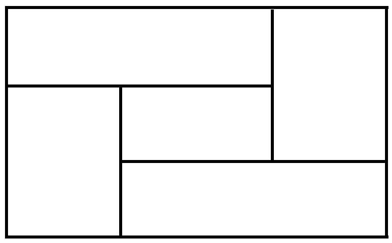


Abbildung 3: Nicht durch Guillotineschnitte realisierbares Schnittmuster

In ihrer Arbeit [Cof 1980] formulierten Coffman, Garey, Johnson und Tarjan folgendes zweidimensionale Packproblem:

Problem 2.2 (zweidimensionales Packproblem)

> Gegeben sei eine Menge von Rechtecken mit jeweils fester Breite und Höhe und ein Behälter mit fester Breite und unbegrenzter Höhe. Ziel ist, die Rechtecke orthogonal so in den Behälter zu packen, daß sich keine zwei Rechtecke überlappen und die Packhöhe des Behälters minimal ist. □

Es wird hier zwar von einem Packproblem gesprochen, jedoch ist in diesem Fall eine äquivalente Formulierung als Verschnittproblem ohne weiteres möglich. Dazu fasse man den Behälter als unendlich lange Bahn eines bestimmten Materials auf. Denkt man sich die Rechtecke als Schablonen, so läßt sich anhand der Plazierung der Schablonen auf der Bahn ein Schnittmuster angeben, nach dem die Rechtecke aus der Bahn ausgeschnitten werden können.

Eine etwas andere Aufgabenstellung für ein zweidimensionales Verschnittproblem formulierten Gilmore und Gomory in [Gil 1965]:

Problem 2.3 (zweidimensionales Verschnittproblem)

> Gegeben sei ein unendlicher Vorrat an Rohlingen der Breite B und Länge L. Es sollen N_i Rechtecke der Breite b_i und Länge l_i, $i = 1, \ldots, m$, aus den Rohlingen ausgeschnitten werden. Dabei dürfen nur Guillotineschnitte verwendet werden. Ziel ist, die Anzahl der benötigten Rohlinge zu minimieren. □

Auf ganz ähnliche Weise wird in Kapitel 4.1 das in dieser Arbeit behandelte Problem formuliert werden.

Im folgenden Abschnitt soll die Komplexität der bislang formulierten Probleme 2.1, 2.2 und 2.3 angegeben werden. Damit läßt sich dann später auf die Komplexität des im Rahmen dieser Arbeit behandelten Problems schließen.

2.3 Komplexität

Mit Hilfe der Komplexitätstheorie läßt sich überprüfen, ob ein zu berechnendes Problem "einfach" oder "schwer" zu lösen ist.

Um die Probleme 2.1, 2.2 und 2.3 in die Klasse "einfacher" oder "schwieriger" Probleme einzuordnen, ist es nötig, zunächst einige grundlegende Begriffe einzuführen und zu erläutern.

Mathematisch läßt sich ein Problem durch eine Funktion $f : D \to W$, mit $x \mapsto f(x)$, beschreiben. D ist dabei der Definitionsbereich der Instanzen, W der Wertebereich der möglichen Lösungen.

Mit Hilfe eines Algorithmus kann für jede Instanz $x \in D$ der Wert $f(x) \in W$ berechnet werden.

Verschlüsselt man eine Instanz nach einem bestimmten Schema, so läßt sich die Größe der Instanz anhand der Zeichenkette ermitteln, die durch die Codierung entsteht. Die Größe einer Instanz x werde mit $|x|$ bezeichnet. Es ist zu beachten, daß wirklich alle Werte einer Instanz codiert werden.

Ein Maß für die Effizienz eines Algorithmus, der aus einer Instanz x der Größe $s = |x|$ die Lösung $f(x)$ berechnet, läßt sich durch die obere Grenze $Z(s)$ der Anzahl von Schritten oder auch Zeiteinheiten angeben, die der Algorithmus zur Lösung des Problems benötigt.

Definition 2.2 (O–Notation)

> Seien f und g Funktionen $f, g : I\!N \rightarrow I\!R^+$. Es ist $f(n) = O(g(n))$, wenn eine Konstante $c \in I\!R$ und ein $n_0 \in I\!N$ so existieren, daß $f(n) \leq cg(n)$ für alle $n \geq n_0$. □

Ist p ein Polynom, s die Größe einer Instanz und gilt für einen Algorithmus, daß $Z(s) = O(p(s))$ ist, so ist die Laufzeit $Z(s)$ des Algorithmus polynomial begrenzt. Er wird als *polynomialer Algorithmus* bezeichnet. Läßt sich kein Polynom p finden, so daß gilt $Z(s) = O(p(s))$, so wird er als *exponentialer Algorithmus* bezeichnet.

Um die Komplexitätstheorie auf ein Problem der kombinatorischen Optimierung anwenden zu können, muß es in ein *Entscheidungsproblem* überführt werden. Auf ein Entscheidungsproblem kann nur mit Ja oder Nein geantwortet werden, das heißt der Wertebereich $W = \{Ja, Nein\}$. Ein Optimierungsproblem wird in ein Entscheidungsproblem überführt, indem die Aufgabenstellung "Finde eine optimale Lösung $f(x)$" in die Frage "Gibt es ein $f(x) \leq k$ für ein gegebenes k" umgewandelt wird.

Definition 2.3 (deterministisch)

> Ein Algorithmus heißt *deterministischer Algorithmus*, wenn jede seiner Operationen ein eindeutiges Ergebnis liefert. Kann nach einer Operation zwischen mehreren Ergebnissen gewählt werden, so wird er als *nichtdeterministischer Algorithmus* bezeichnet. □

Mit den oben eingeführten Begriffen läßt sich nun eine Aufteilung der Entscheidungsprobleme in zwei Klassen vornehmen.

Definition 2.4 (Klassen $\mathcal{P}$ und $\mathcal{NP}$)

> Mit $\mathcal{P}$ werde die Klasse aller Entscheidungsprobleme bezeichnet, die mit einem deterministischen Algorithmus in polynomialer Zeit gelöst werden können. $\mathcal{NP}$ bezeichne die Klasse aller Entscheidungsprobleme, die mit einem nichtdeterministischen Algorithmus in polynomialer Zeit gelöst werden können. □

Bei näherer Untersuchung der Klassen $\mathcal{P}$ und $\mathcal{NP}$ läßt sich erkennen, daß $\mathcal{P}$ eine Teilmenge von $\mathcal{NP}$ ist. Noch offen ist allerdings die Frage, ob es sich dabei um eine echte Teilmenge handelt oder ob möglicherweise $\mathcal{P} = \mathcal{NP}$ gilt. Nähere Untersuchungen hierzu finden sich in einer Arbeit von Cook [Coo 1971]. Auch viele andere Arbeiten beschäftigen sich mit dieser Frage. Als Forschungsergebnis läßt sich bis zum heutigen Zeitpunkt jedoch nur angeben, daß die allgemeine Vermutung dahin geht, daß $\mathcal{P} \neq \mathcal{NP}$ gilt.

Cook führt in seinem Artikel eine weitere Klasse ein: die $\mathcal{NP}$–vollständigen Probleme. Einfach ausgedrückt ist ein $\mathcal{NP}$–vollständiges Problem eines der am schwersten zu lösenden Probleme aus $\mathcal{NP}$. Ein prinzipieller Begriff in der Theorie der $\mathcal{NP}$–Vollständigkeit ist der Begriff der Reduktion.

Definition 2.5 (Reduktion)

> Gegeben seien zwei Entscheidungsprobleme P und Q. P heißt *reduzierbar* auf Q ($P \propto Q$), wenn eine in polynomialer Zeit berechenbare Funktion g existiert, die jede Instanz von P in eine Instanz für Q derart umwandelt, daß x genau dann eine Ja-Antwort für P liefert, wenn $g(x)$ eine Ja-Antwort für Q liefert. $\square$

Mit Hilfe der Reduktion läßt sich nun der Begriff der $\mathcal{NP}$–Vollständigkeit definieren.

Definition 2.6 ($\mathcal{NP}$–vollständig)

> Ein Entscheidungsproblem Q heißt *$\mathcal{NP}$–vollständig*, falls $Q \in \mathcal{NP}$ ist und für alle anderen Entscheidungsprobleme $P \in \mathcal{NP}$ gilt: $P \propto Q$. $\square$

Es gilt folgendes zu beachten: Falls sich ein einziges $\mathcal{NP}$–vollständiges Problem in polynomialer Zeit lösen ließe, so ließen sich alle Probleme in $\mathcal{NP}$ in polynomialer Zeit lösen, und es wäre $\mathcal{P} = \mathcal{NP}$.

Liegt also ein $\mathcal{NP}$–vollständiges Problem vor, so kann man davon ausgehen, daß innerhalb kurzer Zeit kein polynomialer Lösungsalgorithmus gefunden werden kann. Würde man dennoch einen solchen Algorithmus finden, so wäre dies ein bahnbrechender Erfolg in der modernen Mathematik.

Der Nachweis, ob ein $\mathcal{NP}$–vollständiges Problem P vorliegt, erscheint äußerst schwierig, da für jedes Entscheidungsproblem aus $\mathcal{NP}$ die Reduzierbarkeit auf P gezeigt werden muß. Cook bewies, daß das sogenannte Satifizierbarkeitsproblem (SAT) aus der Booleschen Logik $\mathcal{NP}$–vollständig ist. Er zeigt in seinem Artikel ebenfalls ein Verfahren auf, mit dem sich der Nachweis der $\mathcal{NP}$–Vollständigkeit führen läßt.

Soll für ein spezielles Problem P nachgewiesen werden, daß es $\mathcal{NP}$–vollständig ist, so muß gezeigt werden:

1. $P \in \mathcal{NP}$ und
2. es gibt ein $\mathcal{NP}$–vollständiges Problem Q mit $Q \propto P$.

Ein Optimierungsproblem wird in der Komplexitätstheorie als $\mathcal{NP}$*–schwierig* bezeichnet, wenn das zugehörige Entscheidungsproblem $\mathcal{NP}$–vollständig ist.

Eingehende Untersuchungen der Komplexität verschiedener Probleme und tiefgehendere Erläuterungen der hier eingeführten Begriffe und Methoden finden sich in einem Buch von Garey und Johnson [Gar 1979]. Hierin wird auch anhand einer Abbildung die Konstellation der einzelnen Problemklassen verdeutlicht (siehe Abbildung 4), unter der Voraussetzung, daß $\mathcal{P} \neq \mathcal{NP}$ gilt.

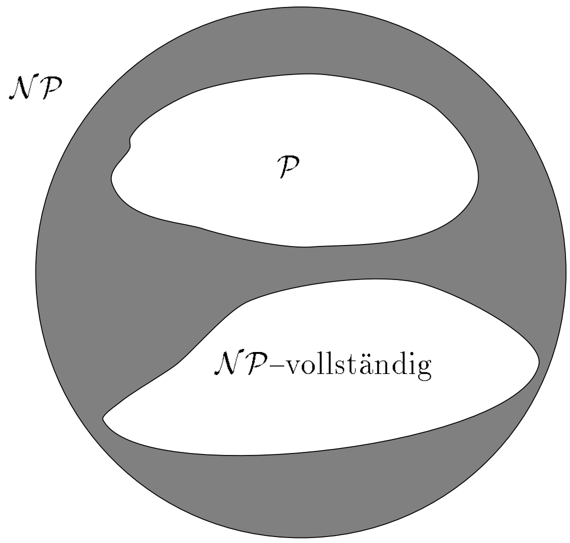


Abbildung 4: Die Welt von $\mathcal{NP}$, falls $\mathcal{P} \neq \mathcal{NP}$

Garey und Johnson gelangen in ihrem Buch unter anderem zu folgendem Ergebnis:

Satz 2.1 Das eindimensionale Bin–Packing Problem ist $\mathcal{NP}$–schwierig. □

Da sich, wie in Abschnitt 2.2 gezeigt, das Bin–Packing Problem äquivalent auch als Verschnittproblem formulieren läßt, gilt offensichtlich auch, daß das eindimensionale Verschnittproblem 2.1 $\mathcal{NP}$–schwierig ist.

Coffman, Garey, Johnson und Tarjan gelangen in ihrer Arbeit [Cof 1980] zu dem Ergebnis, daß sich das von ihnen formulierte Problem 2.2 als eine Kombination aus dem eindimensionalen Verschnittproblem und einem als $\mathcal{NP}$–schwierig erkannten Problem aus der Schedulingtheorie, dem sogenannten Makespan Minimierungsproblem, ansehen läßt. Daraus folgern sie:

Satz 2.2 Das zweidimensionale Packproblem 2.2 ist $\mathcal{NP}$–schwierig. □

Gilmore und Gomory erkannten bei der Komplexitätsuntersuchung des Problems 2.3, daß die Einschränkung auf Guillotineschnitte eine Vereinfachung des Problems bewirkt. Nach ihrer Problemstellung ist es jedoch auch erlaubt, die Rechtecke um 90 Grad zu drehen. Dadurch wird der Lösungsraum wieder vergrößert. Sie gelangen zu dem Schluß, daß das zweidimensionale Verschnittproblem 2.3 ein $\mathcal{NP}$–schwieriges Problem ist.

Da aufgrund der $\mathcal{NP}$–Schwierigkeit der Rechenaufwand der exakten Verfahren sehr hoch ist, ist es sinnvoll, die Probleme 2.1, 2.2 und 2.3 mit approximativen Verfahren zu lösen. Man erhält dann keine optimale Lösung, sondern eine Näherungslösung. Soll eine exakte Lösung angegeben werden, so muß man sich mit Algorithmen begnügen, die in ungünstigen Fällen äußerst lange rechnen, da es sich dabei um exponentielle Algorithmen handelt.

Im folgenden Abschnitt sollen kurz einige Lösungsansätze für Verschnitt– und Packprobleme vorgestellt werden, wie sie in der Literatur zu finden sind.

2.4 Bekannte Lösungsansätze

In der Operations Research Literatur lassen sich viele verschiedene Ansätze zur Lösung ein- und zweidimensionaler Verschnittprobleme finden. Dabei läßt sich eine Aufteilung der zur Lösung verwendeten Verfahren in zwei Kategorien vornehmen. Zum einen handelt es sich um exakte Verfahren, die mit Hilfe bestimmter Algorithmen eine optimale Lösung für das jeweilige Problem bestimmen, zum anderen existieren eine Reihe von Heuristiken, die keine optimalen, sondern Lösungen mit Qualitätsgarantie liefern.

Ein weit verbreitetes heuristisches Verfahren zur Lösung eindimensionaler Verschnittprobleme ist das First–Fit–Decreasing (FFD). Anhand des eindimensionalen Verschnittproblems 2.1 soll die Arbeitsweise des FFD kurz erläutert werden. Die Liste der n bestellten Stäbe wird nach absteigenden Längen sortiert und neu indiziert. Es ist also $l_1 \geq l_2 \geq \ldots \geq l_n$. Die Stangen, von denen die Stäbe abgeschnitten werden, seien mit Indizes $1, 2, 3, \ldots$ versehen. Beginnend mit dem Stab der Länge l_1 werden nacheinander die bestellten Stäbe nach aufsteigenden Indizes von den Stangen abgeschnitten. Dabei muß folgende Regel beachtet werden: Der Stab, der abgeschnitten werden soll, wird immer von der Stange mit dem kleinsten Index abgeschnitten, deren Restlänge noch größer als die abzuschneidende Stablänge ist. Dieses Verfahren ist in Abbildung 5 verdeutlicht.

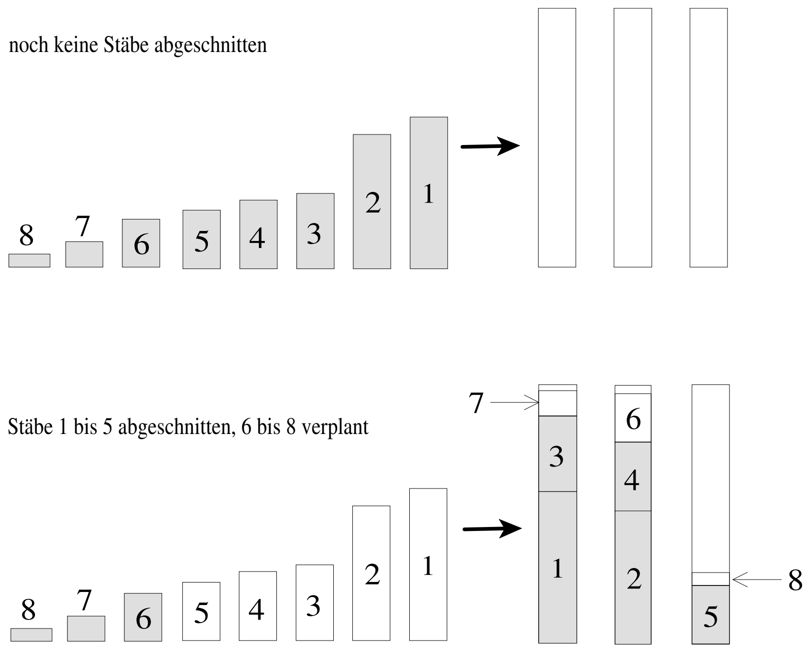


Abbildung 5: First–Fit–Decreasing (Bin–Packing)

Garey und Johnson analysieren in ihrem Buch [Gar 1979] das worst–case Verhalten dieses Algorithmus bezogen auf das Bin–Packing Problem. Mit FFD(I) werde das Ergebnis des FFD–Algorithmus für eine Instanz I bezeichnet, mit OPT(I) ein optimales Ergebnis für diese Instanz.

Satz 2.3 (Qualität des FFD)

Für alle Instanzen I des Bin–Packing Problems gilt

$$\mathrm{FFD}(I) = \frac{11}{9}\mathrm{OPT}(I) + 4 \qquad \square$$

Des weiteren lassen sich Instanzen angeben, so daß OPT(I) beliebig groß wird, und gilt

$$\mathrm{FFD}(I) = \frac{11}{9}\mathrm{OPT}(I)$$

FFD liefert also Ergebnisse, die höchstens um etwa 22 Prozent schlechter sind als die optimalen Ergebnisse.

Gilmore und Gomory [Gil 1961] formulierten ein ähnliches Problem wie das eindimensionale Verschnittproblem 2.1. Dieses Problem wird in der Literatur oft als *allgemeines eindimensionales Verschnittproblem* bezeichnet, da hier auch unterschiedliche Stangenlängen zugelassen sind.

Problem 2.4 (Allgemeines eindim. Verschnittproblem)

> Es sei ein Lager mit Stangen der Längen $L_1, L_2, \ldots, L_n$ eines bestimmten Materials gegeben, von denen bestellte Stäbe abgeschnitten werden sollen. Von jeder Stangenlänge $L_1, L_2, \ldots, L_n$ stehen unendlich viele zur Verfügung. Es soll eine Bestellung von N_i Stäben der Länge l_i, $i = 1, \ldots, m$, erfüllt werden, indem die bestellten Größen von den Stangen abgeschnitten werden. Dabei gelte für jede bestellte Länge l_i $(1 \leq i \leq m)$, daß mindestens eine Stange L_j $(1 \leq j \leq n)$ existiert, mit $l_i \leq L_j$. Jeder Stange wird ein Kostenfaktor zugewiesen. Die Kosten, die bei der Erfüllung der Bestellung entstehen, sind dann die totalen Kosten der zerschnittenen Stangen. Ziel ist, die Bestellung zu erfüllen und dabei so wenig Kosten wie möglich zu verursachen. □

Gilmore und Gomory formulierten dieses Problem folgendermaßen als ganzzahliges lineares Programm:

> Minimiere
> $$c_1x_1 + c_2x_2 + \ldots + c_nx_n$$
> unter den Nebenbedingungen
> $$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n - x_{n+i} = N_i, \qquad i = 1, \ldots m$$
> $$x_j \geq 0, \qquad j = 1, \ldots, n+m$$

Dabei stellt ein Vektor a_i ein sogenanntes Muster dar, durch das angegeben wird, auf welche Weise eine Stange in kleine Stücke zerteilt wird. Ein Muster mit Vektor $a_i = (3, 0, 2, 0)$ gibt an, daß von einer Stange dreimal die Länge l_1, zweimal die Länge l_3 und nicht die Längen l_2 und l_4 abgeschnitten werden. Die Kosten, die das Muster a_i verursacht, werden mit c_i bezeichnet. Lineare Programme werden üblicherweise mit Hilfe des Simplex–Verfahrens gelöst [Chv 1983]. In diesem Fall verursacht Formulierung als Lineares Programm jedoch zwei grundlegende Probleme. Das erste ist die große Anzahl n möglicher Muster, die berücksichtigt werden müssen, das zweite ist die Beschränkung auf ganzzahlige Werte. Das zweite Problem läßt sich lösen, indem man zunächst auch nicht ganzzahlige Werte zuläßt. Die entstehenden Lösungen sind dann in den meisten Fällen ebenfalls nicht ganzzahlig. Durch Aufrunden sucht man dann die nächstgrößere ganzzahlige Lösung auf und findet sich mit den etwas höheren Kosten ab. Oder man rundet die Ergebnisse ab und produziert die noch nicht erfüllten Bestellungen nach. Die Erfahrung zeigt, daß durch Auf– oder Abrunden der Ergebnisse zumeist eine optimale ganzzahlige Lösung erreicht wird. Ein größeres Problem stellt die mögliche große Anzahl von Spalten der Matrix A in der Nebenbedingung dar. Gilmore und Gomory lösten dieses Problem, indem sie die Anzahl der Spalten von A gleich der Anzahl von Zeilen setzten und in jedem Pivotschritt des Simplex–Verfahrens das Muster, welches zu der Variable gehört, die in die Basis eintritt, explizit durch ein Hilfsproblem berechnen. Dieses Hilfsproblem erweist sich als das bekannte eindimensionale Knapsack–Problem. Gilmore und Gomory lösten das Knapsack–Problem mit dynamischer Programmierung.

Das von Gilmore und Gomory vorgestellte Verfahren bildet die Grundlage vieler anderer vorgestellter Lösungsmethoden für ein- und zweidimensionale Verschnittprobleme. Obwohl das Simplex–Verfahren exponentiell und das als Entscheidungsproblem formulierte Knapsack–Problem $\mathcal{NP}$–vollständig ist [Gar 1979], lassen sich in den meisten Fällen mit dem oben erläuterten Verfahren in annehmbarer Zeit optimale Ergebnisse berechnen.

Gilmore und Gomory selbst stellten einige Verbesserungen ihres Lösungsansatzes vor. Unter anderem erweiterten sie die Aufgabenstellung auf das zweidimensionale Verschnittproblem 2.3 und gaben ein Verfahren an, mit dessen Hilfe sich eine optimale Lösung für dieses Problem finden läßt. Schwierigkeiten bereitete ihnen dabei zunächst die Umsetzung des eindimensionalen Knapsack–Problems auf zwei Dimensionen. Sie führten daher noch einige Nebenbedingungen bezüglich der zugelassenen Schnittarten ein, die an dieser Stelle jedoch nicht weiter erörtert werden sollen. Unter Berücksichtigung dieser Nebenbedingungen gelang es ihnen, das zweidimensionale Verschnittproblem auf ähnliche Weise zu lösen wie das eindimensionale.

Eine weitere Möglichkeit, eine optimale Lösung für ein Problem anzugeben, besteht darin, die Menge aller zulässigen Lösungen zu generieren und darin nach einer optimalen zu suchen. Mit zulässiger Lösung ist dabei eine Lösung des Problems gemeint, die die Nebenbedingungen nicht verletzt. Die Kardinalität der zu untersuchenden Menge kann dabei enorm groß sein. Daher ist es sinnvoll, die Anzahl der zu betrachtenden zulässigen Lösungen einzuschränken. Dies geschieht mit Hilfe von Branch & Bound–Verfahren. Für das eindimensionale Verschnittproblem stellte Eisemann ein solches Lösungsverfahren vor [Eis 1957]. Für zweidimensionale Verschnittprobleme entwickelten Whitlock & Christofides [Whi 1977] und Cani [Can 1979] derartige Algorithmen.

Anhand eines zweidimensionalen Verschnittproblems soll nun kurz das Prinzip dieser Verfahren beschrieben werden. Zu Beginn wird eine Reihenfolgeliste festgelegt, in der die Rechtecke aus den Rohlingen herausgeschnitten werden sollen. Dabei gilt es, nicht nur die Reihenfolge selbst, sondern zusätzlich die Ausrichtungen der Rechtecke festzulegen. Anhand der Reihenfolgeliste werden die möglichen Schnittmuster erzeugt, indem die Liste schrittweise abgearbeitet wird und die Rechtecke in den Rohlingen plaziert werden. Um Lösungen zu vermeiden, die unnötigen Verschnitt erzeugen, wird dabei nach dem sogenannten Bottom–Left–Verfahren vorgegangen. Das einzuplanende Rechteck wird immer so weit wie möglich links und so weit wie möglich unten in einem Rohling plaziert, ohne daß es sich mit einem schon eingeplanten Rechteck überlappt, die Rohlingsgrenzen überschritten oder die Nebenbedingungen verletzt werden (Abbildung 6).

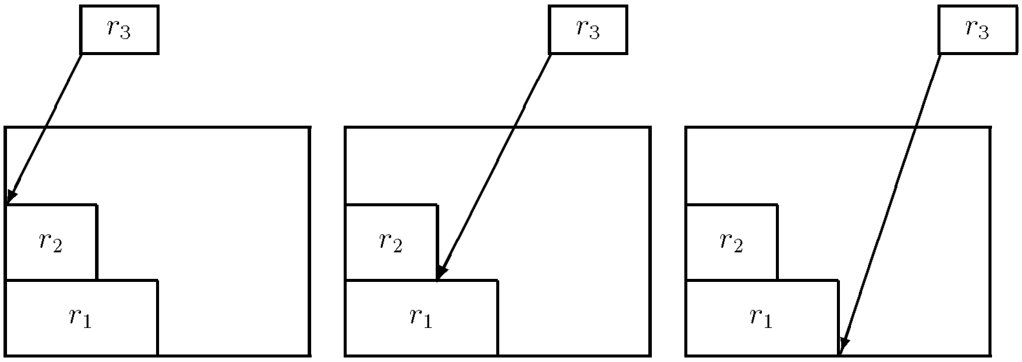


Abbildung 6: Mögliche Bottom–Left Plazierungen eines Rechtecks

Für eine Bottom–Left–Plazierung eines Rechtecks in einem Rohling kommen meistens mehrere Möglichkeiten in Frage. Mit Hilfe einer Baumstruktur werden alle möglichen Plazierungen gespeichert. Die Knoten des Baumes in der ersten Ebene geben alle möglichen Schnittmuster an, bei denen genau ein Rechteck verplant ist. Da die Ausrichtung und die Reihenfolge der Rechtecke festgelegt ist und das Bottom–Left–Verfahren verwendet wird, besteht der Baum in der ersten Ebene nur aus der Wurzel. Die Knoten der zweiten Ebene geben — unter Berücksichtigung der jeweiligen Plazierung des ersten Rechtecks — die Schnittmuster an, die beim Einplanen des zweiten Rechtecks entstehen können. Der Baum wird somit Ebene für Ebene aufgebaut, wobei sich die Anzahl der verplanten Rechtecke von Ebene zu Ebene um eins erhöht. Ein Blatt des Baumes gibt ein Schnittmuster an, bei dem alle Rechtecke verplant sind. Für drei zu verplanende Rechtecke ist das Verfahren in Abbildung 7 dargestellt. Mit Hilfe einer Funktion wird für jede Teillösung — also jeden Knoten des Baumes

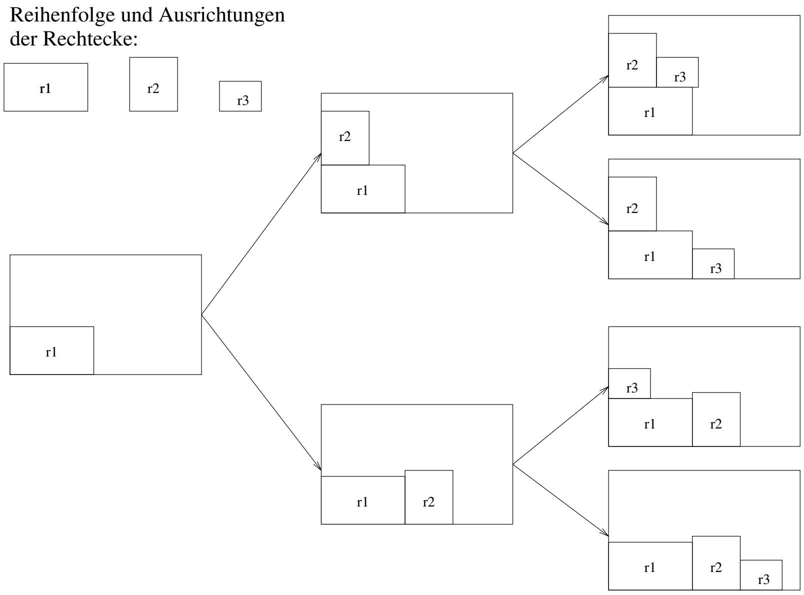


Abbildung 7: Baumaufbau eines Bottom–Left Verfahrens

— eine untere Schranke U für den Verschnitt berechnet, der in den Schnittmustern vorliegen wird, die sich aus dieser Teillösung entwickeln lassen. Um den Baum nicht zu groß werden zu lassen, werden beim Aufbau des Baumes jeweils nur die Teillösungen mit kleinstem U weiter entwickelt. Dadurch werden solche Teilbäume, die geringe Aussicht auf Lösungen mit wenig Verschnitt haben, nicht unnötig berechnet. Für jede mögliche Reihenfolge– und Ausrichtungsliste der Rechtecke werden alle Bäume simultan erzeugt. Dabei werden in allen Bäumen immer nur die Knoten mit kleinstem U bezüglich aller Bäume weiter entwickelt. Erreicht man ein Blatt eines Baumes, so hat man eine optimale Lösung des Problems gefunden, ohne vorher *alle* möglichen Lösungen erzeugt zu haben.

Ein Nachteil dieses Verfahrens besteht darin, daß in ungünstigen Fällen die zu betrachtende Lösungsmenge trotz des Branch & Bound–Verfahrens sehr groß sein kann, unter anderem dadurch bedingt, daß viele verschiedene Reihenfolgen und Ausrichtungen der Rechtecke berücksichtigt werden müssen. Das Verfahren eignet sich eher für eine kleine Anzahl auszuschneidender Rechtecke. Sollen zusätzlich zu anderen Nebenbedingungen nur Guillotineschnitte zugelassen sein, so muß bei der Bottom–Left–Plazierung darauf geachtet werden, daß diese spezielle Schnittart immer durchführbar ist.

Kakazu, Furukawa und Okino stellten ein iteratives Verfahren zur Lösung zweidimensionaler Verschnittprobleme vor, bei dem die Beschränkung auf Guillotineschnitte automatisch berücksichtigt wird [Kak 1976]. Sie erzeugen in jedem Iterationschritt aus den vorhandenen Rechtecken Paare, welche dann im nächsten Iterationsschritt jeweils als *ein* Rechteck angesehen werden. Diese neu entstandenen Rechtecke werden dann wieder gepaart. Das Verfahren wird solange fortgesetzt, bis Rechtecke entstanden sind, die die Größe eines Rohlings besitzen. Ein ähnliches Verfahren soll auch in dieser Arbeit entwickelt werden, da eine Beschränkung auf Guillotineschnitte bei Verschnittproblemen in der Glasindustrie erforderlich ist.

3 Glasherstellung und Glasverarbeitung

Soll das zweidimensionale Verschnittproblem in der Glasindustrie gelöst werden, so müssen einige Nebenbedingungen beachtet werden, die sich aus den speziellen Eigenschaften des Werkstoffes Glas und seiner Verarbeitung ergeben. Dazu ist es zunächst wichtig, etwas über die Glasherstellung zu erfahren.

Zur Herstellung von Glas werden die Rohmaterialen (im wesentlichen Quarzsand, Natriumcarbonat, Dolomit und Kalkstein) in einem Glasofen geschmolzen. Das flüssige Glas wird dann durch Walzen, Blasen oder Gießen in die gewünschte Form gebracht. Eine typische Form ist das sogenannte Flachglas, also Glas in der Form von Platten, Tafeln oder Scheiben (auch gebogen). Flachglas wird auf verschiedene Arten hergestellt, beispielsweise durch das Fourcault–Verfahren oder durch Walz–Verfahren. Besonders hochwertiges Flachglas erhält man durch das *Floatglas–Verfahren*. Das auf diese Weise erzeugte Glas wird zur Herstellung von Fenstern, Spiegeln und Autoscheiben verwendet, da es eine sehr gute Oberflächenqualität besitzt. Das Floatglas–Verfahren wurde in den fünfziger Jahren von der Firma Pilkington in England entwickelt. Dabei wird das Glas in einem Ofen geschmolzen. Das flüssige Material läuft direkt vom Ofen auf ein Zinnbad, auf dem es sich dann ausbreitet. Am Ende des Zinnbades wird das Glas von Walzen abgezogen. Durch diesen Zug entsteht ein langes Glasband. Während des Abkühlens auf dem Zinnbad wird das Glasband durch Heizkörper und Kühlungen auf der richtigen Temperatur gehalten, so daß sich das Glas noch verformen (ziehen) läßt, die Oberfläche dabei aber durch die Walzen nicht mehr beschädigt wird. Die Dicke des Glasbandes wird durch die Stärke des Zuges und durch gekühlte Halterungen (Borten) am Rand des Bandes beeinflußt. Möglich sind Dicken zwischen 2 und 20 mm. Die Breite von Floatglas liegt, bedingt durch das oben beschriebene Verfahren, um die drei Meter. Die schematische Darstellung einer Floatglasanlage (entnommen aus ABC Glas [Ill 1983]) findet sich in Abbildung 8.

Das durch das Floatglas–Verfahren entstandene Glasband wird anschließend auf unterschiedliche Arten verwendet. Wie in den vorherigen Abschnitten erwähnt, soll bei dem Verschnittproblem in der Glasindustrie eine Menge bestellter Rechtecke (Scheiben) hergestellt werden. Firmen, die das Floatglas selbst herstellen, haben die Möglichkeit, die bestellten Scheiben direkt von dem Glasband abzuschneiden. Eine andere Art der Weiterverarbeitung besteht darin, daß das Glasband zunächst in Tafeln (Rohlinge) fester Größe zerteilt wird. Diese Tafeln werden gelagert bzw. an Firmen verkauft, die selbst kein Glas herstellen. Die bestellten Rechtecke werden dann durch Zerschneiden dieser Glastafeln erzeugt.

Das Flachglas wird auf einem *Schneidtisch* zerschnitten. Mit Hilfe des Schneidrädchens, welches aus Hartmetall oder Diamant besteht, wird das Glas 0.5 bis

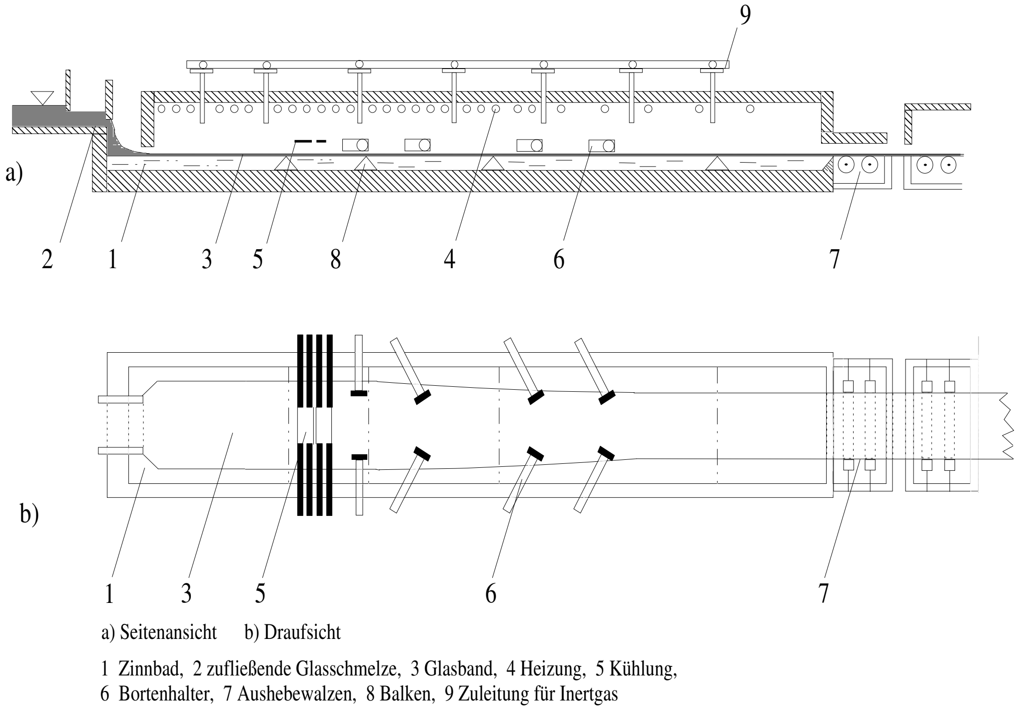


Abbildung 8: Schematische Darstellung einer Floatglasanlage

1 mm tief angeritzt. Anschließend wird das Glas einer Biegespannung ausgesetzt. Dadurch bricht es entlang des Risses. Gekrümmt verlaufende Risse können zwar ebenfalls angelegt werden, jedoch bereitet das Brechen auf einem Schneidtisch entlang eines solchen Risses Schwierigkeiten. Daher werden auf Schneidtischen nur Guillotineschnitte durchgeführt. Ein großer Vorteil gegenüber dem Zerschneiden von Holz oder Kunststoffen liegt beim Glaszuschnitt darin, daß keine Zerspanung stattfindet. Das bedeutet, daß durch den eigentlichen Schnittvorgang kein Material verloren geht. Auf den in der Industrie weit verbreiteten Rasterschneidtischen — das sind Tische, bei denen die Glastafeln sowohl in x– als auch in y–Richtung geritzt werden können — lassen sich Glastafeln von bis zu ca. 6000 mm x 3500 mm Größe bearbeiten. Die Scheiben, die durch das Schneiden entstehen, werden dann auf Rollenbahnen oder Luftpolstern vom Schneidtisch abtransportiert. Um die Glastafeln während der Bearbeitung besser handhaben zu können, ist es sinnvoll, zunächst einige Längs– oder Querschnitte durchzuführen, so daß der große Rohling in einige kleinere Teile zerlegt wird. Diese Schnitte werden auch als *Traversen* bezeichnet. Es kann vorkommen, daß eine Glastafel nicht komplett für den Zuschnitt der Scheiben genutzt wird. Das Stück der Tafel, welches in Längsrichtung gesehen noch in seiner vollen Breite erhalten ist, wird als *Restplatte* bezeichnet.

Eine solche Restplatte besitzt also dieselbe Breite und eine kleinere Länge als der ursprüngliche Rohling. Sie wird in der Praxis für den Zuschnitt anderer Scheiben weiterverwendet. In Abbildung 9 ist unter anderem die Restplatte eines Rohlings dargestellt.

Die Schneidtische werden mit Hilfe eines Computers gesteuert. Ein Optimierungsprogramm legt dazu Schnittmuster fest, aus denen hervorgeht, wie die bestellten Scheiben mit möglichst geringem Verschnitt aus den Rohlinge ausgeschnitten werden können. Die ermittelte Lösung wird auf Diskette oder Lochstreifen gespeichert und über diese Medien an das Steuerungsgerät des Schneidtisches weitergegeben. Bei neueren Schneidtischen kann die Übertragung der Daten zwischen Optimierungsprogramm und Steuereinheit auch direkt über Kabel stattfinden.

Dyson und Gregory untersuchten das Verschnittproblem in der Glasindustrie und wiesen auf einige weitere Nebenbedingungen hin, die beim Zuschnitt von Glas berücksichtigt werden müssen [Dys 1974]. Diese Nebenbedingungen spiegeln dabei hauptsächlich Probleme wider, die bei der Festlegung der Schnittfolgen entstehen: Scheiben, die zu einem Kundenauftrag gehören, sollen direkt nacheinander ausgeschnitten werden. Dyson und Gregory bemühten sich, dieses Problem zu lösen, kamen aber zu keinem befriedigenden Ergebnis. Im Rahmen dieser Arbeit soll auf diese Nebenbedingungen nur kurz in Kapitel 7 eingegangen werden. Hauptziel dieser Arbeit ist, ein Verfahren zu entwickeln, welches in kurzer Zeit Schnittmuster festlegt, die durch Guillotineschnitte realisierbar sind und für die der entstehende Verschnitt möglichst gering ist.

4 Verschnittprobleme in der Glasindustrie

4.1 Problemformulierung

In diesem Kapitel soll das in dieser Arbeit behandelte Verschnittproblem in der Glasindustrie vorgestellt werden. Bei der Problemformulierung ist darauf geachtet worden, eine praxisnahe Aufgabenstellung zu erhalten. Im weiteren Verlauf der Arbeit wird häufig von Rohlingen, Glastafeln oder Platten gesprochen werden. Damit sind die "großen" Rechtecke gemeint, aus denen die "kleinen" Rechtecke, im weiteren auch kurz als Rechtecke oder Scheiben bezeichnet, ausgeschnitten werden sollen. Dabei kann die eine Begriffsbildung jederzeit durch die andere ersetzt werden.

Zunächst ist es wichtig, etwas über die Auftragsabwicklung in einem glasverarbeitenden Betrieb zu erfahren. Die übliche Auftragsabwicklung gestaltet sich so, daß die Bestellungen der Kunden entgegengenommen, die bestellte Ware hergestellt und die fertigen Produkte an die Kunden ausgeliefert werden. Bei der Herstellung von Fensterscheiben — oder allgemein Scheiben — werden die einzelnen Aufträge der Kunden oft auch gesammelt, sofern es sich dabei nicht um Groß– oder Eilbestellungen handelt, bis eine bestimmte Gesamtmenge bestellter Scheiben vorliegt. Dies ist sinnvoll, da

- das Einrichten des Schneidtisches nicht für jeden Auftrag neu vollzogen werden muß
- für eine größere Anzahl Scheiben meist Schnittmuster mit geringerem Verschnitt existieren als für eine kleine Menge Scheiben, da die Anzahl der Schnittmöglickeiten größer ist
- nicht unnötig viele Restplatten entstehen, wie es bei einer Einzelabwicklung der Bestellungen der Fall wäre.

Andererseits ist darauf zu achten, daß das Volumen der gesammelten Aufträge nicht zu groß wird, da ansonsten

- die Verschnittoptimierung zu lange dauert
- eventuell zu viele bereits hergestellte Scheiben gelagert werden müssen, da andere Scheiben, die zum selben Auftrag gehören, möglicherweise noch nicht zurechtgeschnitten wurden
- die Kunden zu lange auf die Erfüllung ihrer Bestellung warten müssen.

Erfahrungen aus der Praxis haben gezeigt, daß das Sammeln von Aufträgen mit einer Gesamtzahl von 50 bis 150 Scheiben sinnvoll ist, abhängig von Scheibengröße und Auftragslage. Ein weiterer Grund, die Anzahl der herzustellenden Scheiben nicht zu groß werden zu lassen, ergibt sich daraus, daß eine Verschnittoptimierung mit Hilfe eines Computerprogramms bei einer zu großen Scheibenanzahl speichertechnische Probleme hervorruft. Daher soll auch in der nun folgenden Problemformulierung die maximale Anzahl der herzustellenden Scheiben auf 170 Stück beschränkt werden.

Problem 4.1 (Glaszuschnitt)

> Gegeben sei eine Liste $\{S_1, S_2, \ldots, S_N\}$ bestellter Scheiben. Sei $N \in I\!N$ und $N \leq 170$. Die Scheibe S_i besitze die Länge l_i und Breite b_i, mit $l_i, b_i \in I\!N$ für $i = 1, \ldots, N$. Weiterhin stehe ein unendlicher Vorrat an Rohlingen $R_1, R_2, R_3, \ldots$ zur Verfügung. Die Länge der Rohlinge werde mit $L \in I\!N$ und die Breite mit $B \in I\!N$ bezeichnet. Es gelte $\max(l_i, b_i) \leq \max(L, B)$ und $\min(l_i, b_i) \leq \min(L, B)$ für alle $i = 1, \ldots, N$. Die bestellten Scheiben sollen so aus den Rohlingen ausgeschnitten werden, daß die Bestellung komplett erfüllt wird und möglichst wenig Verschnitt entsteht. Es sind dabei nur Guillotineschnitte zulässig. $\Box$

Nach der obigen Formulierung ist es zulässig, daß zwei Scheiben S_i und S_j die gleiche Länge und Breite besitzen. Anders als in der Formulierung von Gilmore und Gomory (Problem 2.3) wird konkret jede bestellte Scheibe in der Liste aufgeführt. Man beachte, daß aufgrund der Isotropie des Werkstoffes Glas eine Drehung der Scheiben um 90 Grad zulässig ist. Es ist also erlaubt, eine Scheibe S_i sowohl in Längs– als auch in Querrichtung aus dem Rohling herauszuschneiden. Zur Erinnerung: die Einschränkung auf Guillotineschnitte bewirkt, daß nur orthogonale Schnitte zugelassen sind. Eine Einschränkung auf ganzzahlige Maße ist sinnvoll, da in der Praxis eine natürliche Grenze für die Genauigkeit der Maße existiert. Durch Skalierung läßt sich die Ganzzahligkeitsbedingung einhalten. Im weiteren werden die Maße in Millimetern angegeben.

In den Problemen, wie sie in den vorherigen Abschnitten formuliert wurden, wird als Ziel gefordert, die Bestellung zu erfüllen und dabei "so wenig Rohlinge wie möglich" zu verbrauchen. Diese Formulierung ist weniger restriktiv als die Forderung, möglichst wenig Verschnitt zu verursachen. In der Parxis ist es üblich, die beim Zuschnitt entstehenden Restplatten weiter zu verwerten. Es reicht also nicht zu fordern, daß möglichst wenig Rohlinge verbraucht werden. Vielmehr muß gefordert werden, daß die Schnittmuster eine möglichst dichte Plazierung der Scheiben in den Rohlingen vorsehen. Dadurch wird bewirkt,

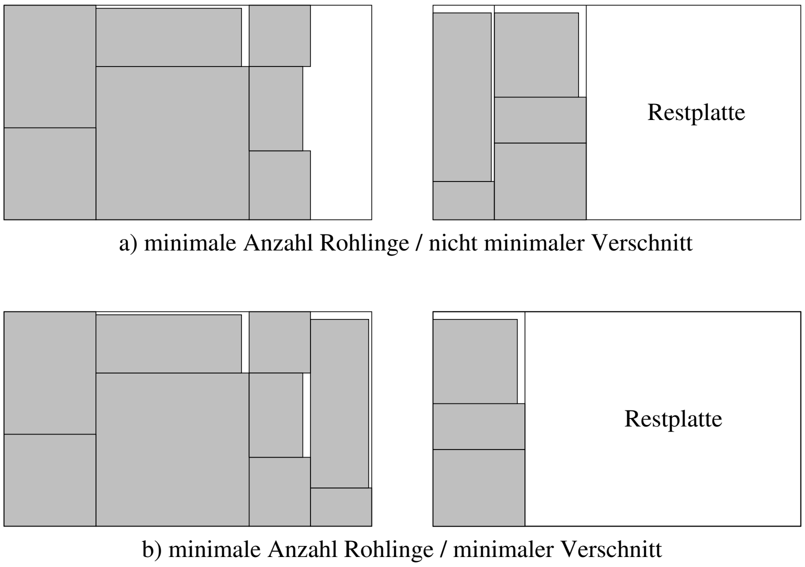


Abbildung 9: Minimale Anzahl Rohlinge vs. minimaler Verschnitt

daß von dem letzten angebrochenen Rohling ein großes zusammenhängendes Reststück übrig bleibt.

Diese Überlegung soll anhand eines Beispiels erläutert werden. Abbildung 9 stellt den Zuschnitt von 12 Scheiben (grau) dar. Die Mindestanzahl der benötigten Rohlinge ist zwei. Insgesamt gesehen ist die Verschnittfläche (weiß) in den Schnittmustern (a) und (b) gleich groß. Möchte man jedoch die entstandenen Restplatten für den nächsten Zuschnitt weiter benutzen, so ist die eine Restplatte in (b) besser geeignet als die zwei Restplatten in (a). Man sagt dann, daß das Schnittmuster (b) weniger Verschnitt verursacht als das Schnittmuster (a), da die längste Restplatte nicht zur Verschnittfläche hinzugerechnet wird. Um ein genaues Maß für die Größe des Verschnitts zu erhalten, soll hier der Begriff der Rohlingszahl eingeführt werden.

Definition 4.1 (Rohlingszahl RZ)

> Sei $k \in I\!N$ die Anzahl der für einen Zuschnitt der Scheiben $S_1, \ldots, S_N$ benötigten Rohlinge R_i, $i = 1, \ldots, k$. Sei L die Länge der Rohlinge. O.B.d.A. sei R_k der Rohling, von dem die längste Restplatte abfällt. Die Länge dieser Restplatte werde mit r bezeichnet. Es ist $r \leq L$. $RZ := (k-1) + \frac{r}{L}$ heißt *Rohlingszahl*. $RZ \in I\!R$ gibt die *reelle Anzahl* der für den Zuschnitt benötigten Rohlinge an. $\square$

Eine Minimierung der Rohlingszahl ist gleichbedeutend mit einer Minimierung des Verschnitts.

Das Problem 4.1 ähnelt stark dem in Kapitel 2.2 vorgestellten zweidimensionalen Verschnittproblem 2.3. Unterschiede treten nur in der Formulierung der Rechteckdaten und in der Strenge des Zieles auf. Man stellt fest, daß das Problem 4.1 offensichtlich nicht einfacher ist als das von Gilmore und Gomory vorgestellte Problem. Es läßt sich daher folgern

Folgerung 4.1 Das Glaszuschnittproblem 4.1 ist $\mathcal{NP}$–schwierig. □

Es ist daher sinnvoll, ein approximatives Verfahren zu entwickeln, mit dem sich gute Näherungslösungen für das zweidimensionale Verschnittproblem in der Glasindustrie finden lassen.

4.2 Lösungsidee und grundlegende Definitionen

Eine Lösung für das Problem 4.1 läßt sich durch ein *Schnittmuster* angeben. Ein Schnittmuster gibt an, in welcher Reihenfolge und Ausrichtung die einzelnen Schnitte durchzuführen sind, um die Rohlinge in die bestellten Scheiben zu zerteilen. Um solch ein Schnittmuster zu erhalten, soll einmal angenommen werden, daß die bestellten Scheiben schon hergestellt worden sind. Setzt man diese Scheiben dann unter Berücksichtigung aller Nebenbedingungen wieder zu Rohlingen zusammen, so läßt sich anhand der zusammengesetzten Rohlinge das Schnittmuster erkennen. Es ist daher zulässig, vom Plazieren einer Scheibe in einem Rohling oder von der Paarung zweier Scheiben zu sprechen. Die Scheiben existieren noch nicht wirklich, sondern man betrachtet sie lediglich als eine Art Schablone.

Die im folgenden Abschnitt eingeführten Begriffe sollen jeweils anhand der nun folgenden Beispielinstanz verdeutlicht werden. Die Rohlingsausdehnungen werden zunächst noch offengelassen.

Beispiel 4.1

> Gegeben sei eine Bestellung von $N = 4$ Scheiben $\{S_1, S_2, S_3, S_4\}$ mit den Maßen (in mm): $l_1 = 2000$, $b_1 = 1000$, $l_2 = 500$, $b_2 = 700$, $l_3 = 1000$, $b_3 = 1300$, $l_4 = 1500$, $b_4 = 600$. □

Bei dem in dieser Arbeit entwickelten Lösungsalgorithmus handelt es sich um ein iteratives Verfahren. Dem Verfahren liegt folgende Idee zugrunde: In jedem Iterationsschritt werden aus den vorliegenden Rechtecken Paare gebildet. Ein

Rechteckpaar stellt dann wieder ein neues (größeres) Rechteck dar. Die neuen Rechtecke werden wiederum miteinander gepaart. Das Verfahren wird solange fortgesetzt, bis die vorliegenden Rechtecke so groß sind, daß eine Paarung nicht mehr möglich ist, ohne daß die aus der Paarung resultierenden Rechtecke die Rohlingsgrenzen überschreiten. Das Verfahren endet ebenfalls, wenn nur noch ein Rechteck übrig ist. Die so entstandenen Rechtecke stellen jedes für sich ein Schnittmuster dar, nach dem ein Rohling zerteilt werden muß, um die bestellten Scheiben zu erhalten. Ganz automatisch wird dabei der Einschränkung auf Guillotineschnitte Rechnung getragen. Die Schnitte, die durchgeführt werden müssen, um die gepaarten Rechtecke wieder voneinander zu trennen, verlaufen immer in gerader Linie von der einen Seite des Rechtecks zur gegenüberliegenden. Dies wird auch in Abbildung 10 deutlich, in der *eine* mögliche Paarung der vier Scheiben aus Beispiel 4.1 dargestellt wird. Die entstandene Verschnittfläche ist schraffiert dargestellt.

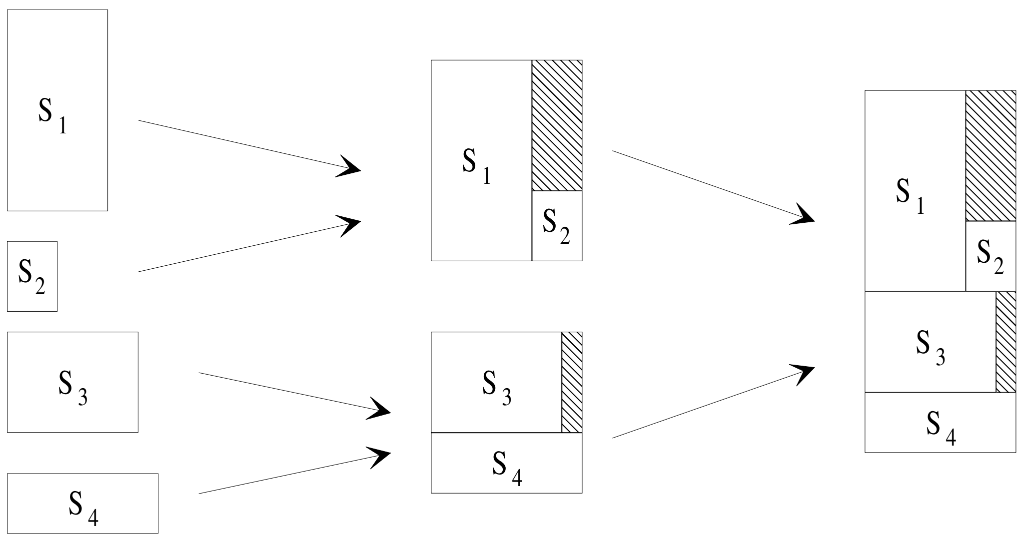


Abbildung 10: Eine mögliche Paarung von vier Rechtecken

Als geeignete Datenstruktur, mit der die einzelnen Paarungen zum Ausdruck gebracht werden können, erweist sich ein *binärer Baum*. Die inneren Knoten des Baumes stellen die durch Paarung erzeugten Rechtecke dar. Ein Blatt des Baumes entspricht einer bestellten Scheibe. Zwei Knoten besitzen einen gemeinsamen Vater, wenn die Rechtecke, die von diesen Knoten repräsentiert werden, ein Paar sind. Abbildung 11 zeigt den Baum, der die Paarungen aus Abbildung 10 verdeutlicht.

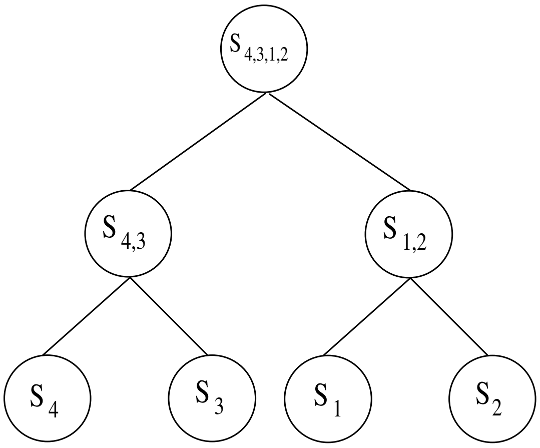


Abbildung 11: Paarung aus Abb. 10 als binärer Baum

Der Baum wird iterativ von unten nach oben, also von den Blättern zur Wurzel, aufgebaut. In jedem Iterationsschritt werden die Rechtecke gepaart und der Baum, der diese Paarungen repräsentiert, wächst um eine Stufe. Um den Zuschnitt einer Glastafel durchzuführen, durchläuft man den entstandenen Baum anschließend von oben nach unten, also von der Wurzel zu den Blättern, und stellt dabei fest, wie die Scheiben gepaart wurden, wie die Glastafel also zerteilt werden soll. Durch die Baumstruktur alleine werden jedoch noch nicht genügend Informationen zur Zerlegung des Rohlings gegeben. So kann man beispielsweise dem Baum in einem Top–Down–Durchlauf nicht entnehmen, wie groß das jeweils zu zerschneidende Rechteck ist und wie der Schnitt durchgeführt werden muß. Daher wird beim Aufbau des Baumes in jedem Knoten eine Schnittanweisung gespeichert.

Definition 4.2 (Schnittanweisung)

> Eine *Schnittanweisung* ist ein 4–Tupel der Form (x, y, o, p). Dabei gibt x die horizontale Ausdehnung und y die vertikale Ausdehnung des zu zerschneidenden Rechtecks an. Die *Schnittorientierung* $o \in \{v, h\}$ gibt an, ob der Schnitt in horizontaler (h) oder vertikaler (v) Richtung verlaufen soll. Die *Schnittposition* p gibt an, bei welcher x–Ausdehnung (y–Ausdehnung) des Rechtecks der vertikal (horizontal) verlaufende Schnitt durchzuführen ist. □

Der Baum mit Schnittanweisungen, der das Schnittmuster für die Paarungen aus Abbildung 10 angibt, ist in Abbildung 12 dargestellt. In den Blättern ist dabei für die Scheiben S_2 und S_3 noch ein sogenannter Trimmzuschnitt nötig, um die Scheiben auf ihre endgültige Form zu bringen. Es soll nun der Begriff des Schnittbaumes definiert werden.

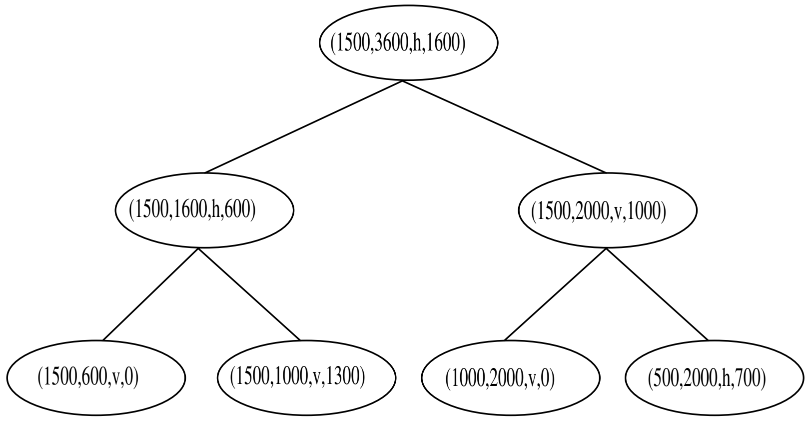


Abbildung 12: Paarung aus Abb. 10 als Schnittbaum

Definition 4.3 (Schnittbaum)

> Gegeben sei ein Rohling R, aus dem k Scheiben S_i, $i = 1, \ldots, k$ ausgeschnitten werden sollen. Ein *Schnittbaum* T des Rohlings R ist ein Baum mit k Blättern und einer Wurzel w. Jedes Blatt repräsentiert eine Scheibe S_i. Alle inneren Knoten des Baumes T besitzen genau zwei Kinder und enthalten mindestens eine Schnittanweisung. Für jeden Knoten v aus der Menge der Baumknoten V_T ist $T(v)$ der *Teilbaum* von T mit Wurzel v, der alle Nachfolger von v enthält. $B(v)$ bezeichnet die Blätter oder Scheiben von $T(v)$. Besteht ein Baum lediglich aus einer Wurzel, so wird er als *einelementiger Baum* bezeichnet. □

Durch einen Schnittbaum wird also das Schnittmuster eines Rohlings angegeben. Da bei einer großen Anzahl bestellter Scheiben in der Regel mehr als ein Rohling benötigt wird, entsteht durch das iterative Paaren für jeden Rohling ein eigener Schnittbaum. Liegen in einem Iterationsschritt k Rechtecke vor, so ist T_j der Schnittbaum, der das j–te Rechteck repräsentiert ($1 \leq j \leq k$). Man spricht der Einfachheit halber von dem Rechteck T_j. Die Schnittbäume, die nach dem letzten Iterationsschritt des Verfahrens vorliegen, geben jeweils ein Schnittmuster für einen Rohling an. Sie werden dann mit denselben Indizes versehen wie die Rohlinge. Somit ist T_j der zum Rohling R_j gehörige Schnittbaum.

4.2.1 Formfunktionen

Ein Nachteil vieler Verfahren besteht darin, daß zwar eine Drehung der Scheiben um 90 Grad zugelassen wird, die Ausrichtung der Scheiben jedoch zu

Beginn des Verfahrens unwiderruflich festgelegt wird. Dies ist auch in dem oben angegebenen Beispiel der Paarung von vier Scheiben (Abbildung 10) der Fall. Sinnvoller ist es jedoch, gerade bei dem hier gewählten Ansatz der Paarbildung, die Ausrichtung der Scheiben erst so spät wie möglich festzulegen. Dadurch erhält man zwar eine größere Anzahl möglicher Schnittanweisungen, hat dafür aber viel variablere Möglichkeiten der Paarbildung. Es soll nun eine Methode vorgestellt werden, mit der man Schnittbäume erstellt, die alle sinnvollen Ausrichtungen der Scheiben berücksichtigen. Ein weiterer Vorteil dieser Methode wird sein, daß sich anhand des Schnittbaumes feststellen läßt, wie lang der Rohling bei gegebener Rohlingsbreite sein muß, um den Zuschnitt erfüllen zu können. Ist die Rohlingslänge gegeben, so läßt sich umgekehrt auch die nötige Breite ermitteln.

Bei der Paarung zweier Scheiben gibt es zwei Möglichkeiten, wie die Schnittlinie, die das Paar wieder auseinander bringt, verlaufen kann. Es kann sich um eine *horizontale* oder eine *vertikale Schnittlinie* handeln. Zusätzlich sind für jede Schnittlinienart vier verschiedene Ausrichtungen der Scheiben möglich. Um dies zu verdeutlichen, sind in Abbildung 13 alle möglichen horizontalen und vertikalen Paarungen der Scheiben S_3 und S_4 aus Beispiel 4.1 aufgeführt. Welche der Scheiben dabei rechts und links bzw. oben und unten liegen, spielt für die Größe des resultierenden Rechtecks keine Rolle. An dieser Stelle wird auch deutlich, daß je nach Paarung und Ausrichtung der Scheiben ein unterschiedlicher Verschnitt vorliegen wird. Betrachtet man die gepaarten Rechtecke als ein großes Rechteck, so entsprechen die Flächen der großen Rechtecke, die keiner Scheibenfläche angehören, der Verschnittfläche, die durch diese Paarungen entstanden ist. In der Abbildung 13 sind diese Verschnittflächen schraffiert gezeichnet. Um alle Ausrichtungsmöglichkeiten zu ermitteln und anhand dieser dann die Schnittanweisungen anzugeben, benötigt man die sogenannten Formfunktionen (vgl. [Mue 1991]).

Definition 4.4 (Formfunktion)

> Eine *Formfunktion* $s : \mathbb{R}_0^+ \to \mathbb{R}^+$ ist eine monoton fallende stückweise stetige Funktion. Falls alle stetigen Segmente von s konstant und alle Funktionswerte sowie Unstetigkeitsstellen von s ganzzahlig sind, ist s eine *Treppenfunktion.* Zu gegebener Rohlingslänge liefert s die für einen Zuschnitt benötigte minimale Breite des Rohlings. Zu gegebener Rohlingsbreite liefert s die für einen Zuschnitt benötigte minimale Länge des Rohlings. □

Die Rohlingslänge und Rohlingsbreite sowie die Längen und Breiten der Scheiben sind als ganzzahlig vorausgesetzt. Die formgebenden Gebilde sind Rechtecke, woraus folgt, daß die stetigen Segmente der Formfunktion konstant

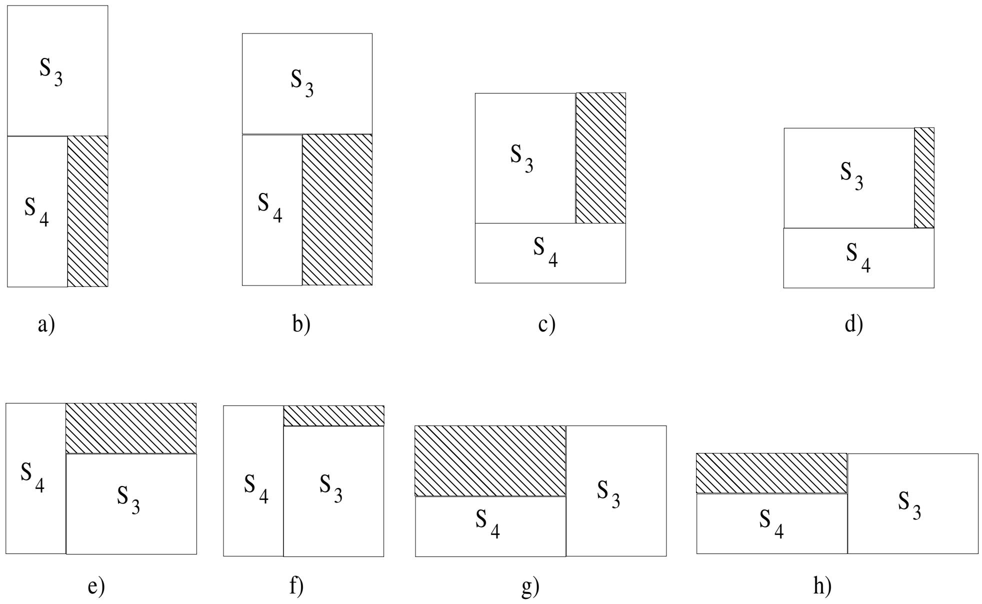


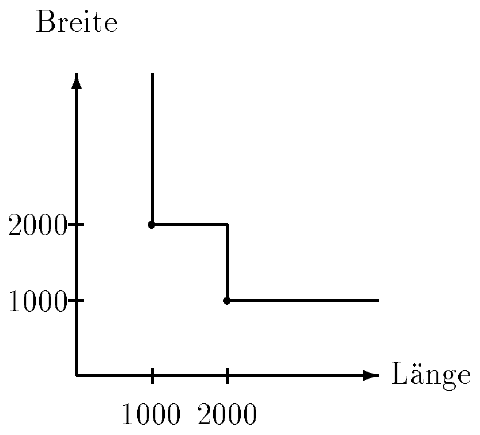
Abbildung 13: Horizontale und vertikale Paarungen

sind. Daher handelt es sich bei den hier zu betrachtenden Formfunktionen um Treppenfunktionen. Die Unstetigkeitsstellen der Treppenfunktion werden Sprungstellen genannt. Für die Ausrichtung einer einzelnen Scheibe existieren zwei Möglichkeiten. Die zu einer Scheibe gehörige Formfunktion enthält zwei Sprungstellen. Abbildung 14 stellt die Formfunktion der Scheibe S_1 des obigen Beispiels 4.1 dar. Dabei ist der Deutlichkeit halber die Linie auch an den Unstetigkeitsstellen eingezeichnet.

Soll nun der Schnittbaum iterativ aufgebaut werden, so muß zunächst für jede Scheibe die zugehörige Formfunktion ermittelt werden. Diese wird dann in dem einelementigen Schnittbaum vermerkt, der die jeweilige Scheibe repräsentiert. Hat man alle in einem Iterationschritt durchzuführenden Paarungen festgelegt, werden die Formfunktionen der aus den Paarungen resultierenden Rechtecke berechnet. Die Schnittbäume zweier zu paarender Rechtecke werden durch einen gemeinsamen Vaterknoten verbunden. Die berechneten Formfunktionen werden in den jeweiligen Vaterknoten gespeichert.

Die Formfunktion des Vaters wird aus den Formfunktionen der beiden zu paarenden Söhne ermittelt. Dies geschieht mit Hilfe der *Kompositionsprozedur*. Otten [Ott 1983] und Stockmeyer [Sto 1983] entwickelten diese spezielle Prozedur, die hier in erweiterter Form vorgestellt werden soll.

Eine Formfunktion läßt sich als eine sortierte Folge von Zahlenpaaren angeben. Ist s eine Formfunktion, so gilt $s = ((x_1, y_1), \ldots, (x_m, y_m))$ mit $x_i < x_{i+1}$ und



Anhand der Formfunktion läßt sich erkennen, daß die Scheibe aus einem Rohling mit einer Länge $L < 1000$ nicht ausgeschnitten werden kann. Ist $1000 \leq L < 2000$, so muß der Rohling eine Mindestbreite von 2000 besitzen. Für Rohlingslängen $L \geq 2000$ reicht eine Breite von 1000 aus, um die Scheibe ausschneiden zu können. Analog lassen sich diese Aussagen auch für gegebene Breite B und festzustellende Mindestlänge L des Rohlings machen.

Abbildung 14: Formfunktion eines Rechtecks (rotieren erlaubt)

$y_i > y_{i+1}$ für $i \in \{1, \ldots, m-1\}$. Diese geordnete Folge beschreibt die Treppenfunktion, wobei jedes Zahlenpaar einer Sprungstelle entspricht. Jedes Zahlenpaar spiegelt also die Dimensionen eines (zu zerlegenden) Rechtecks wider. Die Größe $|s|$ einer Formfunktion s ist gleich der Anzahl ihrer Sprungstellen.

Da man, wie oben schon erwähnt, zur genauen Festlegung der Schnitte die Schnittanweisungen benötigt, sollen nun die Zahlenpaare in einer Formfunktion auf 4–Tupel erweitert werden. Jedes 4–Tupel stellt eine Schnittanweisung dar. Die ersten beiden Komponenten der Schnittanweisung entsprechen den Sprungstellen der Treppenfunktion. Die anderen Komponenten des 4–Tupel geben die Schnittorientierung und die Schnittposition an. Eine Sprungstelle einer Formfunktion entspricht somit genau einer Schnittanweisung. Für die Formfunktionen der herzustellenden Scheiben wird als Schnittorientierung immer "v" und als Schnittposition immer Null angegeben.

Es soll nun erläutert werden, wie die Kompositionsprozedur die Formfunktion eines Vaterknotens aus den Formfunktionen seiner Söhne berechnet. Mit $sp^i = (x^i, y^i, o^i, p^i)$, $i = 1, 2$, seien zwei Sprungstellen zweier Formfunktionen s_1 und s_2 bezeichnet. Die Kompositionsprozedur liefert beim Kombinieren von s_1 und s_2 für die beiden Sprungstellen sp^i, $i = 1, 2$, je nach Schnittorientierung folgende Schnittanweisungen

- horizontaler Schnitt:

  $\text{komp}(\text{hor}, sp^1, sp^2) := (\max(x^1, x^2), y^1 + y^2, h, \min(y^1, y^2))$

- vertikaler Schnitt:

  $\text{komp}(\text{ver}, sp^1, sp^2) := (x^1 + x^2, \max(y^1, y^2), v, \min(x^1, x^2))$

Man bestimmt nun die aus horizontalen Schnitten resultierende Formfunktion s^{hor}, indem die Sprungstellen von s_1 mit den Sprungstellen von s_2 komponiert werden. Dazu addiert man s_1 und s_2 im Durchschnitt ihrer Definitionsbereiche. Um s^{ver} zu erzeugen, invertiert man s_1 und s_2 und addiert die entstandenen Funktionen im Durchschnitt ihrer Definitionsbereiche. Durch Invertieren der daraus resultierenden Formfunktion erhält man s^{ver}. Dieses Verfahren wird in Abbildung 15 für die Paarung der Scheiben S_3 und S_4 aus Beispiel 4.1 dargestellt. Dabei entspricht s_1 der Formfunktion von Scheibe S_4 und s_2 der Formfunktion von S_3. Jede Sprungstelle gibt eine mögliche Gestalt (Schnittanweisung) des Paares an (vgl. auch Abbildung 13).

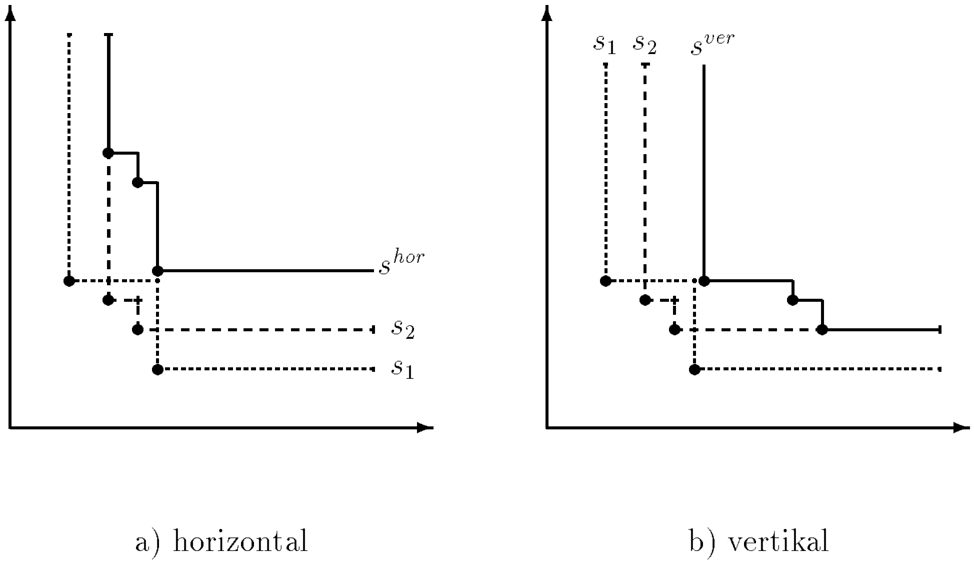


Abbildung 15: Formfunktion berechnen mit gegebener Schnittlinienorientierung

Auffällig an diesem Beispiel ist, daß es laut Abbildung 13 für jede Schnittorientierung vier Ausrichtungsmöglichkeiten der Scheiben gibt, die Formfunktionen s^{hor} und s^{ver} jedoch gemäß Abbildung 15 jeweils nur drei Sprungstellen enthalten. Dies liegt daran, daß die Ausrichtungsart (c) bzw. (e) in Abbildung 13 von der Ausrichtungsart (d) bzw. (f) dominiert wird. Dominiert bedeutet, daß bei gleicher x–Ausdehnung eine geringere y–Ausdehnung möglich ist — (d) dominiert (c) — oder bei gleicher y–Ausdehnung eine geringere x–Ausdehnung möglich ist — (f) dominiert (e). Es sei darauf hingewiesen, daß die dominierten Ausrichtungen einen unnötig großen Verschnitt verursachen. Sie werden durch die Kompositionsprozedur automatisch nicht als Sprungstellen in die Formfunktionen aufgenommen.

Hat man die horizontale und die vertikale Formfunktion für ein Rechteckpaar gebildet, so liegen zwei Formfunktionen vor, von denen jede eine ein-

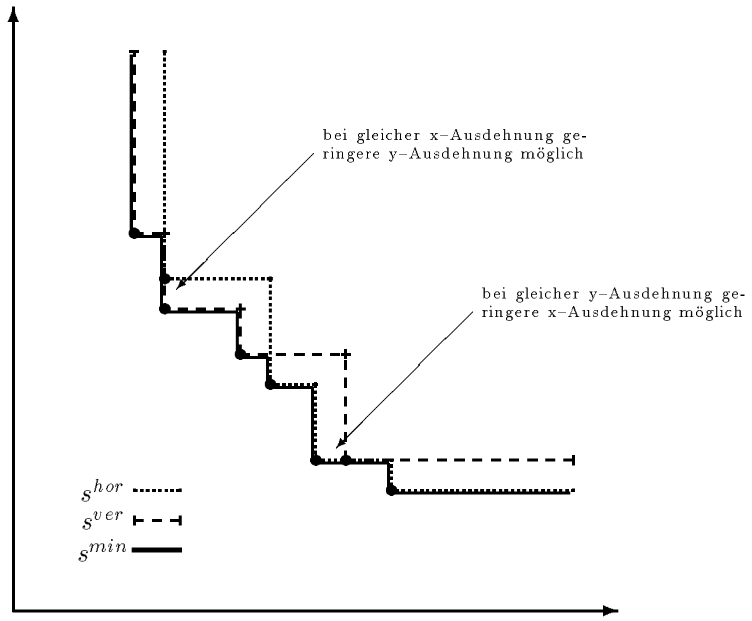


Abbildung 16: Minimumbildung über zwei Formfunktionen

deutige Schnittorientierung repräsentiert. Die Schnittorientierung sollte aber ebenso wie die Ausrichtung der Rechtecke so lange wie möglich offengelassen werden. Daher berechnet man als Formfunktion s für einen inneren Baumknoten die Minimumfunktion bezüglich s^{hor} und s^{ver} dieses Knotens. Bei der Minimumbildung werden die beiden Folgen s^{hor} und s^{ver} gemischt, und Sprungstellen, die sich dominieren, werden entfernt. Das bedeutet folgendes: Sei $\mathrm{sp}^1 = (x^1, y^1, h, p^1)$ eine Sprungstelle der Formfunktion s^{hor} und $\mathrm{sp}^2 = (x^2, y^2, v, p^2)$ eine Sprungstelle der Formfunktion s^{ver}. Ist $x^1 = x^2$ so wird in die Minimumfunktion nur die Sprungstelle sp^i $(i = 1, 2)$ mit dem kleineren y–Wert aufgenommen. Haben zwei Sprungstellen dieselben y–Koordinaten, so wird nur die Sprungstelle mit dem kleineren x–Wert in die Minimumfunktion aufgenommen. Ist $x^1 = x^2$ und $y^1 = y^2$, so wird nur eine der beiden Sprungstellen aufgenommen. Es soll an dieser Stelle festgelegt werden, daß dies die Sprungstelle mit horizontaler Schnittorientierung sein soll. In Abbildung 16 wird die Minimumbildung bezüglich zweier Formfunktionen s^{hor} und s^{ver} dargestellt.

Beim Aufbau der Schnittbäume wird für jeden entstehenden Knoten seine Formfunktion nach dem hier vorgestellten Verfahren berechnet. Sind die Schnittbäume komplett aufgebaut, d.h. kein Iterationsschritt ist mehr durchzuführen, so läßt sich anhand der Schnittanweisungsliste in jedem Baumknoten ablesen, wie das durch den Baumknoten repräsentierte Rechteck zerschnitten werden kann. Soll ein Rohling R_j der Breite B und Länge L zerschnitten werden, so beginnt man die Baumtraversierung in der Wurzel des Schnittbaumes T_j. In der Formfunktion der Wurzel sucht man nach der ersten Sprungstelle (x, y, o, p), für die $y \leq B$ ist. Schon an dieser Stelle weiß man dann, daß von dem Rohling R_j nur die Länge $x \leq L$ genutzt wird. Anhand der Schnittorientierung o und der Schnittposition p der gefundenen Sprungstelle zerteilt man den Rohling in zwei Teile. Diese Teile werden durch die Teilbäume repräsentiert, deren Wurzeln genau die Söhne der Wurzel von T_j sind. Das Zerteilungsverfahren wird nun solange fortgesetzt, bis alle Scheiben ausgeschnitten sind. Ein eventuell notwendiger Trimmzuschnitt für eine Scheibe wird aus den Scheibendaten selbst ermittelt, da in den Formfunktionen als Schnittorientierung "v" und als Schnittposition Null angegeben sind.

Es soll nun noch auf einen Nachteil hingewiesen werden, der sich bei der Verwendung von Formfunktionen ergibt. Man betrachte die Formfunktion s für die oben durchgeführte Paarung von S_3 und S_4 aus Beispiel 4.1:

$$\begin{aligned} s = \{ & (1000, 2800, h, 1300), \quad (1300, 2500, h, 1000), \quad (1500, 1600, h, 600), \\ & (1600, 1500, v, 600), \quad (2500, 1300, v, 1000), \quad (2800, 1000, v, 1300) \ \} \end{aligned}$$

Schon an dieser Beispielliste läßt sich erkennen, daß die Anzahl der Schnittanweisungen in einer einzigen Formfunktion sehr groß werden kann. Genauer gesagt kann sich die Anzahl der Schnittanweisungen einer Liste, die aus zwei Formfunktionen s_1 und s_2 gebildet wird, auf $2 \cdot (|s_1| + |s_2| - 1)$ erhöhen. Hier liegen auch die weiter oben erwähnten speichertechnischen Probleme begründet, aufgrund derer (neben anderen Argumenten) die Beschränkung der Scheibenanzahl auf $N \leq 170$ erfolgt. Bewirkt durch die Art der einzelnen Paarungen kann die Tiefe des Baumes $depth(T)$ für N Scheiben bis zu $N - 1$ betragen (wenn in jedem Iterationsschritt nur eine Paarung stattfindet). Müller untersucht in seiner Arbeit [Mue 1991] das mit der hier erläuterten Formfunktionsbildung verwandte Sizing–Problem. Er weist darauf hin, daß dieses Problem im schlechtesten Fall nur noch in Zeit $O(2^{depth(T)} \cdot S)$ zu lösen ist, wobei s_i die Formfunktion der Scheibe S_i sei und $S = \sum_{i=1}^{N} |s_i|$ ist. Der Algorithmus zur Berechnung aller Formfunktionen des Baumes besitzt im schlechtesten Fall exponentielle Laufzeit. Derart "entartete" Schnittbäume treten in der Praxis jedoch so gut wie nie auf. Durch das oben angegebene Verfahren lassen sich daher in akzeptabler Zeit alle Schnittanweisungslisten berechnen. Um trotzdem unnötigen Speicherbedarf zu vermeiden, werden sofort nach dem Erzeugen einer Formfunktion die Sprungstellen eliminiert, deren x–Wert und y–Wert größer ist als die maximale Rohlingsausdehnung; denn die durch diese Sprungstellen repräsentierten Rechtecke werden garantiert nicht in den Rohling passen und sind damit für den weiteren Verlauf des Verfahrens bedeutungslos. Wird im weiteren Text der Begriff Formfunktionen verwendet, so sind die Sprungstellenlisten gemeint, die nach der Eliminierung der unnützen Sprungstellen vorliegen.

4.2.2 Maximum–Weight–Matching

Bislang wurde davon gesprochen, daß die Formfunktionen für Rechtecke bestimmt werden, die aus der Paarung zweier anderer Rechtecke hervorgehen. Nicht geklärt wurde jedoch, wie bestimmt wird, *welche* Rechtecke jeweils zu paaren sind. Als nächstes soll nun ein Verfahren angegeben werden, mit dessen Hilfe sich in jedem Iterationsschritt ermitteln läßt, welche der möglichen Paarungen aller vorliegenden Rechtecke für eine Verschnittoptimierung am sinnvollsten sind.

Als geeignete Datenstruktur, mit der sich alle möglichen Paarungen der vorliegenden Rechtecke ausdrücken lassen, erweist sich ein ungerichteter gewichteter vollständiger Graph. Jeder Knoten des Graphen repräsentiert ein zur Paarung zur Verfügung stehendes Rechteck. Die in einem Iterationsschritt zur Paarung zur Verfügung stehenden Rechtecke entsprechen den bis zu diesem Zeitpunkt generierten Schnittbäumen. Die Kanten des Graphen stellen alle möglichen Paarungen dar. Um die *Güte* einer Paarung anzugeben, wird jede Kante des Graphen mittels einer Funktion bewertet. Diese Funktion wird auch *Benefitfunktion* genannt. Sie bestimmt die Güte einer Paarung aus den Formfunktionen der zu paarenden Rechtecke. Welche konkrete Gestalt die Benefitfunktion besitzt, soll in Kapitel 5 angegeben werden. Hier soll zunächst angenommen werden, daß die Kanten bereits bewertet worden sind, die Güte der Paarungen also bekannt ist. Um alle optimalen Paarungen eines Iterationsschrittes zu bestimmen, verwendet man das Maximum–Weight–Matching aus der kombinatorischen Optimierung.

Problem 4.2 (Maximum–Weight–Matching)

> Gegeben sei ein Graph $G = (V, E)$. Sei $V = \{v_1, v_2, \ldots, v_n\}$ die Menge der Knoten mit $n \in I\!N$. Sei E die Menge der ungerichteten Kanten. Mit $e(i, j)$ werde eine Kante zwischen Knoten v_i und v_j bezeichnet, wobei $i \neq j$ und $i, j \in \{1, \ldots, n\}$ gelte. Sei $c \in I\!N^{n \times n}$ die Wertmatrix des Graphen, wobei $c_{i,j}$ den Wert der Kante $e(i, j)$ angibt. Mit $x \in I\!N^{n \times n}$ werde die Zuordnungsmatrix der Graphknoten bezeichnet. Dabei ist $x_{i,j} = 1$, wenn Knoten v_i und Knoten v_j gepaart werden sollen, und Null sonst.

Das *Maximum–Weight–Matching*–Problem läßt sich dann folgendermaßen formulieren:

Maximiere

$$\sum_{i,j=1}^{n} c_{i,j} \cdot x_{i,j}$$

unter den Nebenbedingungen

$$\begin{array}{rcl} \sum_{j=1}^{n} x_{i,j} \leq 1 & \text{für} & i = 1, \ldots, n \\ x_{i,i} = 0 & \text{für} & i = 1, \ldots, n \\ x_{i,j} = x_{j,i} & \text{für} & i, j = 1, \ldots, n \\ x_{i,j} \in \{0, 1\} & \text{für} & 1 \leq i \leq j \leq n \end{array}$$

□

Es werden in dem Graphen die Kanten so bestimmt, daß eine Kostenmaximierung stattfindet und die maximum mögliche Anzahl disjunkter Kanten gewählt wird. Es ist wichtig, zwischen der *maximum* möglichen Anzahl und der *maximal* möglichen Anzahl gewählter Kanten zu unterscheiden. Maximum bedeutet, daß keine weitere Kante gewählt werden kann, ohne daß zwei Kanten mit einem Knoten inzidieren. Maximal bedeutet, daß keine weitere Kante gewählt werden kann, ohne daß zwei Kanten mit einem Knoten inzidieren und die größtmögliche Anzahl Kanten gewählt wurde. Dieser Sachverhalt wird in Abbildung 17 verdeutlicht. Mit Hilfe des Maximum–Weight–Matching werden

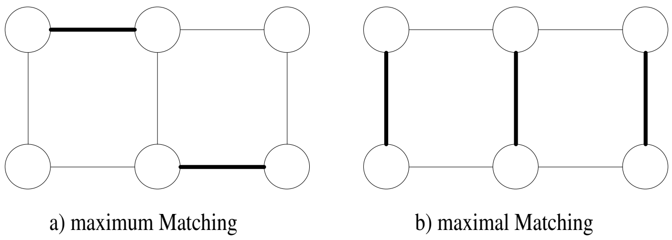


Abbildung 17: Maximum und Maximal Matching in einem Graphen

also die Kanten des Graphen bestimmt, für die die Summe ihrer Kantenwerte am größten ist. Dabei dürfen keine zwei der ausgewählten Kanten einen gemeinsamen Endpunkt besitzen. Es soll die maximum mögliche Anzahl Kanten gewählt werden, ohne die Endpunktbedingung zu verletzen. Kanten, die von der Endpunktbedingung her ausgewählt werden dürfen, deren Wert jedoch Null ist, werden nicht gewählt, das heißt in der Zuordnungsmatrix wird für solche Kanten eine Null eingetragen. Dies bewirkt, daß keine Paare gebildet werden, deren Güte Null ist. Existieren in einem Graphen mehrere Möglichkeiten, den maximalen Gesamtwert zu erreichen, so werden zunächst die Kanten mit den größeren Werten ausgewählt. Vereinfacht ausgedrückt wird mehr Wert auf das Finden des maximalen Gesamtwertes als auf das Finden der maximum möglichen Matchingkanten gelegt. Dieser Umstand wird in Abbildung 18 anhand eines Graphen mit drei unterschiedlichen Bewertungen verdeutlicht. Die fett gezeichneten Linien geben die vom Maximum–Weight–Matching gewählten Kanten an.

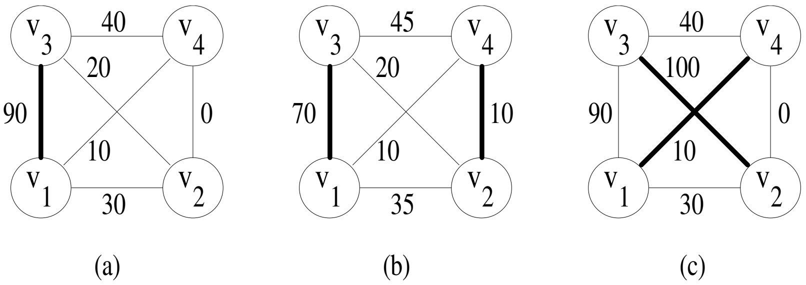


Abbildung 18: Matching in einem Graphen bei unterschiedlichen Bewertungen

Im Graphen (a) hätte die Kante $e(2,4)$ nach der Endpunktbedingung zusätzlich gewählt werden dürfen. Sie wurde jedoch nicht gewählt, da diese Paarung keine Wertverbesserung erzielt. Im Graphen (b) wurden $e(1,3)$ und $e(2,4)$ gewählt, obwohl $e(1,2)$ und $e(3,4)$ denselben Gesamtwert erzielt hätten. Wie oben erläutert, werden als erstes die Zuordnungen mit höheren Werten ausgewählt. Die Wahl der Kanten im Graphen (c) ist offensichtlich. Die Anzahl der maximum möglichen Kanten entspricht in diesem Fall der Anzahl maximal möglicher Kanten. Anhand der vom Matching erstellten Zuordnungsmatrix lassen sich die in einem Iterationsschritt durchzuführenden Paarungen ablesen.

Der in dieser Arbeit verwendete Matchingalgorithmus wurde von Ed Rothberg in der Programmiersprache C implementiert. Der Algorithmus basiert auf H. Gabows "N-cubed weighted matching" Algorithmus, den dieser im Jahre 1973 vorstellte [Gab 1973]. Das Maximum–Weight–Matching in einem Graphen läßt sich mit polynomiellem Aufwand bestimmen. Dies läßt sich nachweisen, indem das hier vorgestellte Maximum–Weight–Matching–Problem mit polynomiellem Aufwand auf eine von Papadimitriou und Steiglitz vorgeschlagene Problemformulierung zurückgeführt wird [Pap 1982]. Die beiden Autoren entwickelten einen Algorithmus, der das von ihnen formulierte Problem in Zeit $O(n^4)$ löst, wobei n die Anzahl der Graphknoten ist.

Zusammenfassend läßt sich der in dieser Arbeit entwickelte Algorithmus zur Verschnittoptimierung bislang wie folgt beschreiben: Zu Beginn des Verfahrens sind N einelementige Schnittbäume gegeben. Die Wurzel eines jeden Schnittbaumes ist mit der Formfunktion des von ihm repräsentierten Rechtecks initialisiert. Es wird ein vollständiger ungerichteter gewichteter Graph erzeugt, dessen Knoten genau den Wurzeln der vorliegenden Schnittbäume entsprechen. Die Kanten des Graphen werden mit Hilfe einer Benefitfunktion bewertet, welche die Güte der von der Kante repräsentierten Paarung angibt. Mit Hilfe des Maximum–Weight–Matching werden die für die Verschnittminimierung optimalen Paarungen bestimmt. Die ausgewählten Paarungen werden durchgeführt, d.h. die Wurzeln der Schnittbäume, die die beiden zu paarenden Rechtecke darstellen, erhalten einen gemeinsamen Vater. Dadurch werden zwei Schnittbäume zu einem zusammengesetzt. Der Wurzelknoten eines entstandenen Schnittbaumes wird mit der Formfuktion versehen, die sich aus der Komposition der beiden Formfunktionen in den Wurzeln der gepaarten Schnittbäume berechnen läßt. Dieses Verfahren — Graph bilden und bewerten, Matching durchführen, Schnittbäume aufdatieren — wird solange iterativ fortgesetzt, bis keine Paarung zweier Rechtecke mehr existiert, deren resultierendes Rechteck die Rohlingsgrenzen nicht überschreitet. Jeder der dann vorliegenden Schnittbäume gibt die Zuschnittmöglichkeiten für den von diesem Schnittbaum repräsentierten Rohling an. Um von einem Rohling, der zugeschnitten werden soll, eine möglichst lange Restplatte übrig zu behalten, bestimmt man in der Formfunktion des Wurzelknotens die erste Sprungstelle, deren y–Wert kleiner oder gleich der Rohlingsbreite B ist. Anhand der durch diese Sprungstelle repräsentierten Schnittanweisung wird der Rohling durch einen Guillotineschnitt in zwei Teile zerlegt. Wie diese beiden Teile weiter zerschnitten werden müssen, läßt sich aus den Formfunktionen der beiden Söhne des Wurzelknotens entnehmen. Die Zerlegung der entstehenden Rechtecke wird solange fortgesetzt, bis die bestellten Scheiben vorliegen.

Nicht geklärt wurde bislang, wie sich eine sinnvolle Bewertung der Graphkanten finden läßt. Dazu muß eine geeignete Benefitfunktion festgelegt werden. Aufgabe dieser Funktion ist, die Paarung zweier Rechtecke unter dem Gesichtspunkt der Verschnittminimierung zu bewerten. Ein Problem stellt dabei die große Flexibilität der in jedem Iterationsschritt des Verfahrens vorliegenden Rechtecke dar. Es werden in jedem Iterationsschritt zwar die Paarungen durchgeführt, die Ausrichtungen der Rechtecke sind jedoch noch beliebig. Es muß eine Benefitfunktion gefunden werden, die diese unterschiedlichen Ausrichtungsmöglichkeiten berücksichtigt. Da mit jedem Iterationsschritt die Anzahl der gebildeten Paare und damit auch die Anzahl der zu berücksichtigenden Ausrichtungen der Rechtecke wächst, läßt sich keine geschlossene Funktion finden, die in jedem Iterationsschritt des Verfahrens eine sinnvolle Bewertung der Kanten liefert. Daher wird der Aufbau der Schnittbäume in mehrere Phasen aufgeteilt.

5 Ein vierphasiges Lösungsverfahren

Zur genaueren Begriffsbildung soll zunächst eine weitere Bezeichnung eingeführt werden.

Bezeichnung 5.1 (Metarechteck)

> Ein *Metarechteck* ist ein durch Paarung zweier Rechtecke entstandenes Rechteck. Dabei darf es sich bei den gepaarten Rechtecken ebenfalls um Metarechtecke handeln. □

Die Rechtecke, die durch einen inneren Schnittbaumknoten repräsentiert werden, sind Metarechtecke. Ein Metarechteck enthält alle Scheiben, aus deren Paarung dieses Metarechteck entstanden ist. Die möglichen Ausrichtungen aller dieser Scheiben werden durch die zum Metarechteck gehörige Formfunktion ermittelt. Man muß streng zwischen dem Metarechteck an sich und einer möglichen *Gestalt* des Metarechtecks unterscheiden. Jede Sprungstelle seiner Formfunktion repräsentiert eine andere Gestalt des Metarechtecks. Im weiteren soll nur in Fällen, in denen eine Unterscheidung zwischen Rechteck und Metarechteck unbedingt erforderlich ist, die konkreten Bezeichnungen benutzt werden. Andernfalls soll der Einfachheit halber weiterhin von Rechtecken gesprochen werden.

Wie sich in Abbildung 13 erkennen läßt, ergibt sich beim Zusammensetzen zweier Rechtecke automatisch ein Verschnitt. Besitzen die gepaarten Rechtecke gleiche Ausdehnungen, so kann dieser Verschnitt auch Null sein. Beim iterativen Erzeugen der Schnittbäume ist darauf zu achten, daß der Verschnitt, der beim Zusammensetzen zweier Rechtecke entsteht, möglichst gering bleibt, da die Verschnittflächen nicht weiter genutzt werden können. Nur so kann erreicht werden, daß sich der gesamte Verschnitt in einem Rohling in Grenzen hält. Eine Benefitfunktion, die die Güte einer Paarung bewertet, muß daher so konstruiert sein, daß sie Paare, die einen geringeren Verschnitt hervorrufen, höher bewertet, als Paare, die einen großen Verschnitt verursachen. Dabei kann das aus einer Paarung entstandene Metarechteck durchaus eine Gestalt mit besonders wenig und eine mit besonders viel Verschnitt besitzen. Dies läßt sich anhand der Abbildung 13 verdeutlichen. Für ein und dieselbe Paarung liegt für die Gestalt in Abbildung 13(d) augenscheinlich ein geringerer Verschnitt vor als für die Gestalt in Abbildung 13(g). Aufgabe der Benefitfunktion ist, bei der Bewertung einer Paarung diesen Sachverhalt zu berücksichtigen und die für den Verschnitt günstigen Paarungen hoch zu bewerten, so daß das Maximum–Weight–Matching später die sinnvollen Paare auswählt.

Der Verschnitt, der in einer Gestalt eines Metarechtecks vorliegt, läßt sich anhand des Teilbaumes, der dieses Metarechteck repräsentiert, ermitteln. Sei w

die Wurzel dieses Teilbaumes. Durch die Sprungstellen der Formfunktion in w sind die Ausdehnungen für die verschiedenen Gestalten des Metarechtecks gegeben. Anhand der Ausdehnungen (x– und y–Wert der betrachteten Sprungstelle) läßt sich die von einer Gestalt des Metarechtecks *geforderte* Fläche ermitteln. Die für die Herstellung der Scheiben *genutzte* Fläche in dieser Gestalt läßt sich aus den Blättern $B(w)$ des Teilbaumes ermitteln. Anhand von $B(w)$ berechnet man die Summe der Flächen aller Scheiben, aus denen sich das Metarechteck zusammensetzt. Sei $sp = (x, y, o, p)$ die die Gestalt bestimmende Sprungstelle der Formfunktion in w. Dann läßt sich der prozentuale Verschnitt $PV_{T(w)}(sp)$ für die Gestalt sp des durch den Schnittbaum $T(w)$ repräsentierten Metarechtecks folgendermaßen berechnen:

$$PV_{T(w)}(sp) = 100 \cdot \left(1 - \frac{\sum \text{Scheibenflächen} \quad \text{in} \quad B(w)}{x \cdot y}\right)$$

Bislang arbeitet das Verfahren so, daß die Rechtecke solange gepaart werden, bis die Fläche der entstandenen Metarechtecke so groß geworden ist, daß keine weitere Paarung mehr möglich ist, ohne die Grenzen des Rohlings zu überschreiten. Da die Anzahl der gepaarten Rechtecke und damit auch die Anzahl der möglichen Gestalten mit jedem Iterationschritt wächst, ließ sich — bis zum jetzigen Zeitpunkt — keine einzelne Benefitfunktion finden, die in jedem Iterationsschritt des Verfahrens eine sinnvolle Bewertung der Graphkanten liefert. Wie in Kapitel 3 erläutert, ist es in der Praxis wünschenswert, daß beim Zerteilen der Glastafeln durch die Einhaltung der Traversen handliche Abschnitte entstehen. Dies ist bei der Erzeugung der großen Metarechtecke nicht sichergestellt. Je größer das Metarechteck, desto unhandlicher können die Teile sein, die aus dem Schnittmuster des Metarechtecks entstehen. Das Verfahren soll daher dahingehend modifiziert werden, daß in jedem Iterationsschritt eine sinnvolle Bewertung der Paare möglich ist und beim Zerteilen der Glastafeln nach den ersten Schnitten handliche Abschnitte entstehen. Die grundlegende Idee des iterierten Matching soll beibehalten werden.

Dazu wird das Verfahren in vier Phasen unterteilt. Die ersten beiden Phasen dienen zur Erzeugung der sogenannten *Rohlingsabschnitte*. Rohlingsabschnitte sind Rechtecke, die eine Gestalt besitzen, in der die y–Ausdehnung des Rechtecks nahezu der Breite des Rohlings entspricht und die x–Ausdehnung kleiner als die Rohlingslänge ist. Die y–Ausdehnung eines Rohlingsabschnitts wird, konform zu den Bezeichnungen der Rohlingsausdehnungen, als Breite und die x–Ausdehnung als Länge des Abschnitts bezeichnet. Man beachte, daß die Länge eines Abschnitts — entgegen der intuitiven Bedeutung — kleiner sein kann als die Breite. In den ersten beiden Phasen werden durch iteriertes Matching die Scheiben zu Rohlingsabschnitten gepaart. Genauer gesagt werden die Scheiben zu Metarechtecken gepaart, die eine Gestalt besitzen, die einen Rohlingsabschnitt darstellt. In der dritten Phase des Verfahrens findet eine

Nachoptimierung der Abschnitte statt. Hat man alle Scheiben zu Rohlingsabschnitten zusammengesetzt und ist die Nachoptimierung abgeschlossen, so wird festgelegt, welcher Abschnitt aus welchem Rohling ausgeschnitten werden soll. Die Länge der Abschnitte kann unterschiedlich sein. Legt man die Breite aller Abschnitte als B fest, so entspricht das Problem, die Abschnitte auf die Rohlinge zu verteilen, einem eindimensionalen Packproblem. Wie in Kapitel 2 erläutert, läßt sich dieses Problem mit Hilfe des First–Fit–Decreasing Algorithmus heuristisch lösen. In den folgenden Abschnitten sollen die vier Phasen des Verfahrens genauer erläutert werden.

5.1 Phase 1: Bilden universeller Rechtecke

In den ersten beiden Phasen sollen durch den iterativen Aufbau der Schnittbäume Rechtecke erzeugt werden, die einen Rohlingsabschnitt darstellen. Dazu soll zunächst geklärt werden, wie die Gestalt eines Rechtecks ermittelt wird, die einen sinnvollen Rohlingsabschnitt darstellt.

Wie schon in dem grundlegenden Verfahren werden auch beim Vierphasenmodell beim Aufbau der Schnittbäume nur solche Paarungen durchgeführt, aus denen ein Metarechteck resultiert, das mindestens eine Gestalt besitzt, in der es in einen Rohling paßt. Das Ziel ist, die Rohlingsfläche so gut wie möglich zu nutzen, um den Verschnitt in einem Rohling so gering wie möglich zu halten. Man wählt für ein Rechteck diejenige Gestalt aus, in der es die Breite des Rohlings so gut wie möglich nutzt und die Länge des Rohlings so wenig wie nötig beansprucht. Um diese Gestalt eines Rechtecks zu bestimmen, ermittelt man in der zugehörigen Formfunktion die erste Sprungstelle, deren y–Wert kleiner oder gleich der Rohlingsbreite ist. Aufgrund der Formfunktionseigenschaften wird dadurch automatisch die kleinstmögliche Längenausdehnung der Gestalt ermittelt. Man hat die Ausrichtung des Rechtecks ermittelt, die soviel wie möglich von der Breite und so wenig wie nötig von der Länge des Rohlings nutzt. Dies ist gleichbedeutend mit einem dichten Plazieren der Scheiben innerhalb des Rohlingsabschnitts. Um die *prozentuale Ausbeute* eines Abschnitts zu bestimmen, berechnet man wie oben angegeben den prozentualen Verschnitt des zugehörigen Metarechtecks für die gegebene Gestalt und subtrahiert diesen Wert von 100. Die prozentuale Ausbeute wird analog zum prozentualen Verschnitt mit $PA_T(sp)$ bezeichnet. Man sagt dann, der Abschnitt besitzt eine Ausbeute von $PA_T(sp)$ Prozent.

Die Größe der Scheiben ist in der Praxis im Gegensatz zur Rohlingsgröße meist klein. Die Rohlingsgröße liegt bei ca. $L = 6000$ mm und $B = 3200$ mm. Die größte Ausdehnung einer Scheibe liegt in der Regel bei maximal 3000 mm. Es ist daher selten (aber nicht ausgeschlossen), daß eine einzelne Scheibe einen Rohlingsabschnitt darstellt. Das Ziel der ersten Phase soll nun sein, die Scheiben so zu paaren, daß Rechtecke entstehen, die anschließend in einem einzigen Iterationschritt zu Rohlingsabschnitten mit einer bestimmten Ausbeute zusammengesetzt werden können. Für diese Art von Rechtecken soll eine spezielle Bezeichnung eingeführt werden.

Bezeichnung 5.2 (universelle Rechtecke)

> Ein Rechteck A heißt *universell bezüglich* p, wenn mindestens ein anderes Rechteck B vorliegt, so daß das aus der Paarung von A und B resultierende Rechteck eine Gestalt besitzt, in der es einem Rohlingsabschnitt mit einer Ausbeute von mindestens p Prozent entspricht. Das Rechteck B ist ebenfalls universell bezüglich p. □

Man beachte, daß sich in der Bezeichnung der Begriff Rechteck auch durch den Begriff Metarechteck ersetzen ließe. Durch die gewählte Formulierung wird verdeutlicht, daß auch eine einzelne Scheibe einen Rohlingsabschnitt darstellen kann.

Im weiteren Verlauf dieses Abschnitts wird erläutert, wie sich ein Rechteck auf Universalität überprüfen läßt und wie sich universelle Rechtecke durch iteriertes Matching erzeugen lassen.

5.1.1 Test auf Universalität

Das Testen auf die universelle Eigenschaft und das Paaren der vorliegenden Rechtecke zu universellen Rechtecken soll anhand eines Iterationsschrittes erläutert werden. Dazu muß zunächst die Situation zu Beginn des Iterationsschrittes beschrieben werden. Sei k die Anzahl der vorliegenden Rechtecke. Zu Beginn des Verfahrens ist $k = N$. Mit T_j $(j = 1, \ldots, k)$ werde der zu einem Rechteck gehörige Schnittbaum bezeichnet. Der Einfachheit halber spricht man auch von dem Rechteck T_j anstatt von dem "durch den Schnittbaum T_j repräsentierten" Rechteck. Bevor die vorliegenden Rechtecke mit Hilfe des Matchingalgorithmus gepaart werden, wird jedes Rechteck auf Universalität bezüglich einer prozentualen Ausbeute p überprüft. Welchen Wert p sinnvollerweise haben sollte, wird im nächsten Abschnitt erläutert. O.B.d.A. sei T_1 das auf die universelle Eigenschaft zu überprüfende Rechteck. Es werden probehalber die Formfunktionen s_{j-1} $(j = 2, \ldots, k)$ durch Komponieren der in den Wurzeln von T_1 und T_j vorliegenden Formfunktionen gebildet. Läßt sich in einer Formfunktion s_{j-1} $(j = 2, \ldots, k)$ eine Sprungstelle finden, die einen Rohlingsabschnitt mit einer Ausbeute von mindestens p Prozent widerspiegelt, so sind die Rechtecke T_1 und T_j universell bezüglich p. Läßt sich keine solche Sprungstelle finden, so ist das von T_1 repräsentierte Rechteck nicht universell bezüglich p. Die als universell erkannten Rechtecke werden als solche markiert. Der Test auf Universalität bezüglich p wird für jedes der k vorliegenden Rechtecke durchgeführt, sofern es nicht bereits als universell markiert wurde.

5.1.2 Universelle Rechtecke durch iteriertes Matching

Nachdem der Test abgeschlossen ist, müssen die noch nicht markierten Rechtecke zu universellen Rechtecken gepaart werden. Die Paarungen werden durch das Maximum–Weight–Matching bestimmt. Dazu wird der Matchinggraph von allen in diesem Iterationsschritt vorliegenden Rechtecken erzeugt. Jeder Knoten des Graphen entspricht einem Rechteck, welches durch seinen Schnittbaum repräsentiert wird. Alle Kanten des Graphen, die mit mindestens einem als universell markierten Rechteck inzidieren, werden mit Null bewertet. Die Bewertung der anderen Graphkanten erfolgt mit Hilfe der Benefitfunktion für die erste Phase. Diese werde mit $bene_1$ bezeichnet. Sei k die Anzahl der in dem Iterationsschritt vorliegenden Rechtecke. O. B. d. A. seien T_j $(j = 1, \ldots, k')$ mit $k' \leq k$ die nicht universellen Rechtecke. Um den Benefit einer Paarung der Rechtecke T_i und T_j $(1 \leq i < j \leq k')$ zu bestimmen, werden T_i und T_j probehalber zu T_{ij} gepaart. Sei s die aus dieser Paarung resultierende Formfunktion. Läßt sich in s keine Sprungstelle finden, die eine Rechteckgestalt widerspiegelt, welche die Rohlingsgrenzen nicht überschreitet, so ist der Wert der Paarung gleich Null. Andernfalls wird die mittlere Ausbeute aller durch die Sprungstellen in s repräsentierten Rechteckgestalten ermittelt. Dazu addiert man die prozentualen Ausbeuten an den einzelnen Sprungstellen sp^m $(1 \leq m \leq |s|)$ der Formfunktion s und dividiert durch die Anzahl $|s|$ der Sprungstellen. Als mittlere Ausbeute für die aus den Sprungstellen der Formfunktion s in der Wurzel des Baumes T_{ij} resultierenden Gestalten berechnet man somit

$$MA_{T_{ij}} := \frac{\sum_{m=1}^{|s|} PA_{T_{ij}}(sp^m)}{|s|}$$

Da für die Bewertung der Kanten im Matchinggraph nur ganzzahlige Werte zugelassen sind, werden nicht ganzzahlige Werte auf den nächstkleineren ganzzahligen Wert abgerundet. Ist die mittlere Ausbeute größer oder gleich 95 Prozent, so liefert $bene_1$ diesen Wert als Benefit der Paarung. Liegt die mittlere Ausbeute unter 95 Prozent, so wird die Paarung mit Null bewertet. Die Benefitfunktion $bene_1$ läßt sich somit wie folgt beschreiben:

$$bene_1(T_i, T_j, L, B) = \begin{cases} \lfloor MA_{T_{ij}} \rfloor & \text{falls } T_i \text{ und } T_j \text{ nicht universell} \\ & \text{und } \exists sp^m = (x^m, y^m, o^m, p^m) \in s \text{ mit} \\ & \quad \min(x^m, y^m) \leq \min(L, B) \text{ und} \\ & \quad \max(x^m, y^m) \leq \max(L, B) \\ & \text{und } MA_{T_{ij}} \geq 95 \\ 0 & \text{sonst} \end{cases}$$

Der mittlere Ausbeutenwert von 95 Prozent wurde anhand einiger Testläufe für verschiedene Instanzen festgelegt. An dieser Stelle läßt sich erkennen, warum die mit $bene_1$ gebildeten Rechtecke als universell bezeichnet werden. Bei diesen grundlegenden Paarungen wird darauf geachtet, daß die aus einer Paarung entstehenden Rechtecke im Mittel in jeder möglichen Gestalt einen geringen Verschnitt verursachen. Egal welche Gestalt für ein solches Rechteck dann im weiteren Verlauf des Verfahrens gewählt wird, der Beitrag zum Gesamtverschnitt ist im Mittel gering. Das Rechteck ist *universell* einsetzbar.

Die Matchinggraphen jedes Iterationschrittes während der ersten Phase besitzen die Kantenwerte Null oder einen ganzzahligen Wert zwischen 95 und 100. Durch das Maximum–Weight–Matching werden in jedem Iterationsschritt die optimalen Paarungen bestimmt, und die Schnittbäume werden, wie in Kapitel 4 beschrieben, aufdatiert. Die erste Phase wird solange ausgeführt, bis entweder alle vorliegenden Rechtecke universell sind oder keine Paarungen mehr durchgeführt werden, da alle Graphkanten mit Null bewertet worden sind. Im letzteren Fall werden für den weiteren Verlauf des Verfahrens auch die nicht universellen Rechtecke als universell angesehen. Bei gegebener Mindestausbeute p läßt sich die erste Phase algorithmisch wie folgt beschreiben:

1. REPEAT
2. Markiere universelle Rechtecke bezüglich p;
3. Erzeuge Matchinggraph G für alle vorliegenden Rechtecke unter Benutzung der Benefitfunktion $bene_1$;
4. Führe Maximum–Weight–Matching auf G durch;
5. Datiere Schnittbäume gemäß der in 4. bestimmten Matchingkanten auf;
6. UNTIL (alle Kanten in G besitzen den Wert Null);

5.2 Phase 2: Bilden der Rohlingsabschnitte

In der ersten Phase wird aus den gegebenen Scheiben eine Menge universeller Rechtecke bezüglich einer Mindestausbeute erzeugt. Wie groß diese Mindestausbeute sein sollte, wurde bislang noch nicht geklärt. Es ist sinnvoll, die Mindestausbeute eines Rohlingsabschnitts möglichst hoch anzusetzen, da sich jeglicher Verschnitt in einem Rohlingsabschnitt später in dem Gesamtverschnitt des Rohlings wiederfindet. Würde die Mindestausbeute jedoch zu hoch angesetzt, so würde der Test auf Universalität nur wenige Rechtecke als universell markieren. Die nichtuniversellen Rechtecke würden durch iteriertes Matching immer weiter zusammengesetzt, so daß die entstehenden Metarechtecke wieder sehr viele Gestalten besäßen. Dies sollte aber gerade verhindert werden.

Anhand einiger Testläufe für verschiedene Instanzen ist die zu erfüllende Mindestausbeute auf 96 Prozent festgelegt worden. Durch die erste Phase werden somit universelle Rechtecke bezüglich einer Ausbeute von 96 Prozent erzeugt.

5.2.1 Rohlingsabschnitte durch iteriertes Matching

In der zweiten Phase sollen die aus der ersten Phase hervorgegangenen Rechtecke zu Abschnitten gepaart werden. Dazu muß eine weitere Benefitfunktion festgelegt werden, die die möglichen Paarungen sinnvoll bewertet, so daß der Matchingalgorithmus anschließend die optimalen Zuordnungen festlegen kann. Um den Gesamtverschnitt gering zu halten, müssen Abschnitte mit einer großen Ausbeute erzeugt werden. Die Benefitfunktion wird daher eine Paarung mit der ganzzahligen prozentualen Ausbeute des aus der Paarung entstehenden Abschnitts bewerten. Paarungen, aus denen Metarechtecke hervorgehen, die in keiner Gestalt in den Rohling passen, werden mit Null bewertet. Anders als in der ersten Phase werden alle Benefitwerte zwischen Null und Einhundert zugelassen. Die zweite Phase besteht aus genau einem Iterationsschritt, in dem jeweils zwei der vorliegenden Rechtecke zu einem Abschnitt gepaart werden. Im Vorausblick auf die vierte Phase, in der die Abschnitte durch First–Fit–Decreasing auf die einzelnen Rohlinge verteilt werden, gilt es, einen weiteren Gesichtspunkt für den Benefitwert einer Paarung zu berücksichtigen. Wie in Abschnitt 4.1 erläutert, entspricht eine Verringerung der Rohlingszahl einer Verringerung des Verschnitts. Es muß daher darauf geachtet werden, daß die x–Ausdehnungen der entstehenden Abschnitte von einer solchen Form sind, daß durch das FFD die Rohlinge so dicht wie möglich mit Abschnitten bepackt werden können. Dadurch wird eine kleine Rohlingszahl erreicht. Hierbei läßt sich auf zweierlei Arten argumentieren. Zum einen kann man verlangen, daß die Abschnitte schmal (nicht lang) sein sollen, um möglichst viele Variationsmöglichkeiten beim Verteilen auf die Rohlinge zu erhalten. Zum anderen kann es auch sinnvoll sein, die x–Ausdehnung der Abschnitte groß werden zu lassen, um mit einem Abschnitt — der bedingt durch Phase 1 eine hohe Ausbeute besitzt — schon möglichst viel Rohlingsfläche zu belegen. Die Benefitfunktion sollte bei der Bewertung einer Zuordnung in Phase 2 somit nicht nur auf die Ausbeute, sondern zusätzlich auf die x–Ausdehnung des entstehenden Abschnitts eingehen. Um *beide* Argumente bezüglich der Längenausdehnung eines Abschnitts *und* die Ausbeute zu berücksichtigen, wurde die folgende Benefitfunktion $bene_2$ für die zweite Phase gewählt. Zu bewerten sei die Paarung der durch T_i und T_j repräsentierten Rechtecke. Die Rohlingslänge sei L, die Rohlingsbreite B. Sei $t = 10^u$ mit $u \in \mathbb{Z}$ so, daß $1 \leq \frac{L}{t} < 10$ gilt. Mit Hilfe von t läßt sich L auf eine Zahl zwischen 1 und 10 normieren. Sei s die probehalber aus der Paarung von T_i und T_j erzeugte Formfunktion in der Wurzel von T_{ij}. Sei $sp^k = (x^k, y^k, o^k, p^k)$, $k \leq |s|$, die erste Sprungstelle in s für die gilt $y^k \leq B$.

Sei $p = PA_{T_{ij}}(sp^k)$ die Ausbeute des Abschnitts, der sich aus der Sprungstelle s^k ergibt.

Die Benefitfunktion $bene_2$ der zweiten Phase lautet:

$$bene_2(T_i, T_j, L, B) = \begin{cases} 0 & \text{für} \quad p < 50 \\ \left\lfloor p - 50 + p^3 \cdot \left(\frac{2x^k - L}{1000t}\right)^2 \right\rfloor & \text{für} \quad 50 \leq p \leq 100 \end{cases}$$

Abbildung 19 zeigt eine Skizze des Graphen von $bene_2$ bei einer gegebenen Rohlingslänge von 6000 mm für Ausbeuten von $50 \leq p \leq 100$ Prozent (in Fünferschritten). Dabei sind die Benefitwerte zur besseren Veranschaulichung noch nicht auf ganzzahlige Werte abgerundet worden. Ansonsten läge eine Schar von Treppenfunktionen vor.

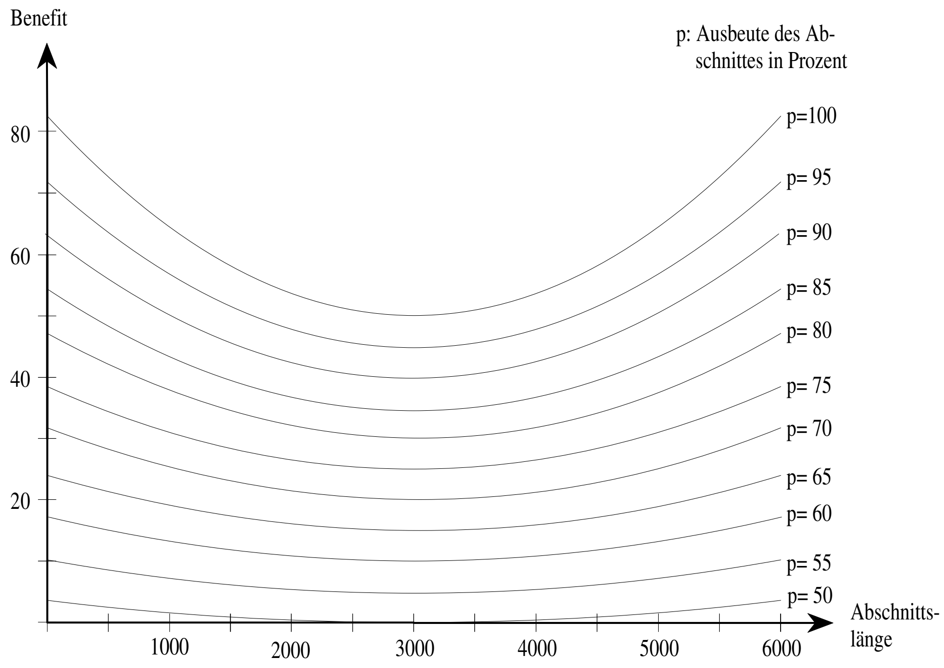


Abbildung 19: Skizze der Benefitfunktion $bene_2$

Liegt die Ausbeute eines entstehenden Abschnitts unter 50 Prozent, so würde die Benefitfunktion negative Werte liefern. Für die Kantenwerte des Matchinggraphen sind nur ganzzahlige Werte größer oder gleich Null zulässig. Daher wird ein negativer Benefit durch Null ersetzt. Der Matchingalgorithmus bestimmt anhand des Graphen die optimalen Paarungen. Diese werden durchgeführt, indem die entsprechenden Schnittbäume durch einen gemeinsamen Vaterknoten verbunden werden. Am Ende der zweiten Phase liegt eine Menge von Schnittbäumen vor, von denen jeder einen Rohlingsabschnitt repräsentiert.

Es ist möglich, daß am Ende der ersten Phase eine ungerade Anzahl Rechtecke vorliegt. Das bedeutet, daß mindestens ein Rechteck in der zweiten Phase keinen Partner findet. Dieses "allein gebliebene" Rechteck wird ebenfalls als Rohlingsabschnitt angesehen.

Ein Nachteil des iterierten Matching liegt in der Gier des Algorithmus. Es werden in jedem Iterationsschritt alle Paarungen durchgeführt, die vom Matchingalgorithmus als sinnvoll erachtet werden. Es kann aber durchaus Situationen geben, in denen es besser wäre, eine Paarung nicht durchzuführen. Beispielsweise könnte schon im nächsten Iterationsschritt ein Metarechteck vorliegen, mit dem sich eine bessere Paarung ergeben würde. Auch die umgekehrte Richtung ist denkbar. In einem Iterationsschritt kann ein Metarechteck vorliegen, welches optimal mit einer Scheibe gepaart werden könnte. Die Scheibe wurde aber möglicherweise schon in einem vorherigen Iterationsschritt einem anderen Rechteck zugeordnet. Es ist daher sinnvoll, eine Nachoptimierung der in Phase 2 erzeugten Rohlingsabschnitte durchzuführen.

Diese Nachoptimierung wird in zwei Schritte unterteilt. Der erste Schritt, bei dem einige der in Phase 2 entstandenen Abschnitte komplett neu generiert werden, soll als nächstes vorgestellt werden. Der andere Schritt der Nachoptimierung, bei der die bereits erzeugten Schnittbäume modifiziert werden, stellt die dritte Phase des Verfahrens dar. Sie soll im Abschnitt 5.3 dieser Arbeit vorgestellt werden.

5.2.2 Nachoptimieren durch iteriertes Matching

Am Ende der zweiten Phase liegen Schnittbäume vor, die Rohlingsabschnitte repräsentieren. Einige dieser Rohlingsabschnitte besitzen — bewirkt durch die Benefitfunktion $bene_2$ — eine Ausbeute von mindestens 96 Prozent. Diese Abschnitte werden markiert. Es können aber durchaus auch Abschnitte vorliegen, deren Ausbeute kleiner ist als 96 Prozent. Ziel des Verfahrens soll jedoch sein, möglichst wenig Verschnitt bei der Herstellung der Scheiben zu verursachen. Daher werden die Abschnitte mit einer Ausbeute kleiner als 96 Prozent komplett aufgelöst. Die Schnittbäume dieser Abschnitte werden gelöscht. Das heißt, die in diesen Abschnitten enthaltenen Scheiben stehen alle wieder zur Paarung zur Verfügung. Man führt nun die erste und die zweite Phase angewandt auf die wieder freigegebenen Scheiben durch. Dabei können nun weitere Abschnitte mit einer Mindestausbeute von 96 Prozent entstehen. Denn Rechtecke, die bereits in einem markierten Abschnitt liegen, nehmen nun keinen "negativen Einfluß" mehr auf die Paarungen anderer Rechtecke, wie es eventuell beim ersten Erzeugen der Abschnitte geschehen ist.

Dieses Auflösen und Zusammensetzten bezüglich 96 prozentiger Abschnitte wird solange wiederholt, bis auch beim erneuten Zusammensetzen keine Abschnitte mehr mit einer Mindestausbeute von 96 Prozent generiert werden konnten. Die unmarkierten Abschnitte werden wieder aufgelöst, und der Ausbeutengrenzwert für die ersten beiden Phasen wird um zwei Prozent verringert. Das heißt, durch Anwenden der ersten und zweiten Phase sollen nun Rohlingsabschnitte mit mindestens 94 prozentiger Ausbeute erzeugt werden. Dieses Vorgehen wiederholt man solange, bis auch beim erneuten Zusammensetzen der Scheiben keine Abschnitte mehr mit einer Mindestausbeute von 92 Prozent generiert werden konnten. Es werden dann alle vorliegenden Rechtecke als Rohlingsabschnitte angesehen. Dieses Verfahren erscheint sehr aufwendig. Die Erfahrung zeigt jedoch, daß nach dem ersten Durchlaufen der Phasen 1 und 2 ein Großteil der Abschnitte schon die gewünschte Ausbeute besitzt. Die Anzahl der Scheiben, die wieder neu gepaart werden, verringert sich in der Regel nach jeder Abschnittserzeugung. Die Geschwindigkeit, in der die Abschnitte generiert werden, nimmt daher im Laufe des Verfahrens zu. Denn liegen weniger Scheiben vor, so sind auch weniger Matchingmöglichkeiten zu bewerten, festzulegen und durchzuführen. Das Verfahren terminiert, da entweder alle erzeugten Abschnitte die gewünschte Mindestausbeute besitzen oder der Grenzwert von 92 Prozent erreicht ist. Zusammenfassend läßt sich das Verfahren bislang wie folgt algorithmisch beschreiben:

1. p:=96;
2. Führe Phase 1 und Phase 2 bezüglich der Mindestausbeute von p Prozent auf die bestellten Scheiben durch;
3. WHILE ($p > 90$) DO
4.   Löse alle Abschnitte mit einer Ausbeute kleiner p Prozent auf;
5.   Führe Phase 1 und Phase 2 bezüglich einer Mindestausbeute von p Prozent auf die wieder freigewordenen Scheiben durch;
6.   IF (neue Abschnitte besitzen alle Ausbeute kleiner p Prozent)
7.   THEN $p := p - 2$; END;
8. END;

Die Phasen 1 und 2 einschließlich der oben erläuterten Nachoptimierung bilden den Kern der Verschnittoptimierung durch iteriertes Matching. Die Phasen 3 und 4 dienen zur Nachoptimierung und zur Verteilung der Abschnitte auf die Rohlinge. Sie werden kein Matching mehr verwenden. Wird im weiteren Text die Phase 2 erwähnt, so soll das Bilden der Rohlingsabschnitte aus den universellen Rechtecken einschließlich der oben erläuterten Nachoptimierung durch iteriertes Matching gemeint sein. Die Phasen 1 und 2 laufen somit nicht nacheinander, sondern ineinander geschachtelt ab. In der zweiten Phase wird die erste Phase mehrmals ausgeführt.

5.3 Phase 3: Nachoptimieren der Abschnitte

Durch die Phasen 1 und 2 sind Rohlingsabschnitte entstanden, die eine Ausbeute von mindestens 92 Prozent besitzen sollten. Ein kleiner Teil der vorliegenden Abschnitte kann jedoch, bedingt durch ungünstige Einwirkungen der Scheibengrößen auf die Benefitfunktionen, geringere Ausbeuten besitzen. Um den Gesamtverschnitt zu beschreiben, läßt sich wie in Abschnitt 4.1 erläutert die Rohlingszahl berechnen. Es soll an dieser Stelle noch davon ausgegeangen werden, daß die Abschnitte auf die Rohlinge optimal verteilt werden können, d. h. bis auf einen Rohling wird die Länge aller Rohlinge komplett ausgenutzt. Man spricht dann auch von der *Mindestrohlingszahl*. Die Mindestrohlingszahl berechnet sich aus der Summe aller Abschnittlängen dividiert durch die Rohlingslänge. Ein Abschnitt, der eine geringe Ausbeute besitzt, trägt zur Erhöhung der Rohlingszahl bei, da er im Verhältnis zu seiner Breite eine große Länge benötigt. Ziel der Nachoptimierung in der dritten Phase ist, die Gesamtlänge aller Abschnitte, also die Summe der x–Ausdehnungen aller Abschnitte, zu verringern. Dies soll durch systematisches Vertauschen von Teilbäumen zwischen den einzelnen Schnittbäumen erzielt werden. Dabei wird ausgenutzt, daß die Ausrichtungen der einzelnen Scheiben nicht festgelegt sind. Ein Scheibenpaar, welches in einem Abschnitt durch eine vertikale Schnittlinie getrennt ist, kann nach dem Austausch mit einem Rechteck aus einem anderen Abschnitt, in diesem anderen Abschnitt durchaus horizontal angeordnet sein. Die möglichen Gestalten der einzelnen Rechtecke sind durch die Formfunktionen in den einzelnen Schnittbaumknoten gegeben. Abbildung 20 zeigt an einem einfachen Beispiel, wie durch einen einzigen Teilbaumtausch die Gesamtlänge zweier Abschnitte und somit die Mindestrohlingszahl verringert werden kann.

Es muß festgelegt werden, welche Abschnitte nachoptimiert werden müssen und welche Teilbäume konkret ausgetauscht werden sollen. Es ist nicht sinnvoll, alle Teilbäume aller Abschnitte miteinander zu vertauschen und nach jedem Tausch zu überprüfen, ob die Mindestrohlingszahl kleiner geworden ist. Zum einen wäre der Aufwand zu groß und zum anderen ist es nicht sinnvoll, Abschnitte wieder zu zerstören, die schon eine sehr hohe Ausbeute besitzen. Es soll an dieser Stelle festgelegt werden, daß alle Abschnitte mit einer Ausbeute kleiner als 96 Prozent nachoptimiert werden sollen. Ist die Anzahl dieser Abschnitt kleiner als zehn, so sollen die zehn Abschnitte mit geringster Ausbeute nachoptimiert werden. Es wird eine nach aufsteigenden Ausbeuten sortierte Liste der Schnittbäume angelegt, die die nachzuoptimierenden Abschnitte repräsentieren. Es werden systematisch alle Teilbäume dieser Schnittbäume probeweise miteinander vertauscht. Man beachte, daß durch das Vertauschen der Teilbäume die Anzahl der Abschnitte bzw. Schnittbäume nicht verändert wird. In den so probehalber entstandenen Schnittbäumen werden die Formfunktionen neu berechnet. Es genügt dabei nur die Formfunktionen in den Vorgängern der vertauschten Baumknoten neu zu berechnen. Man beginnt

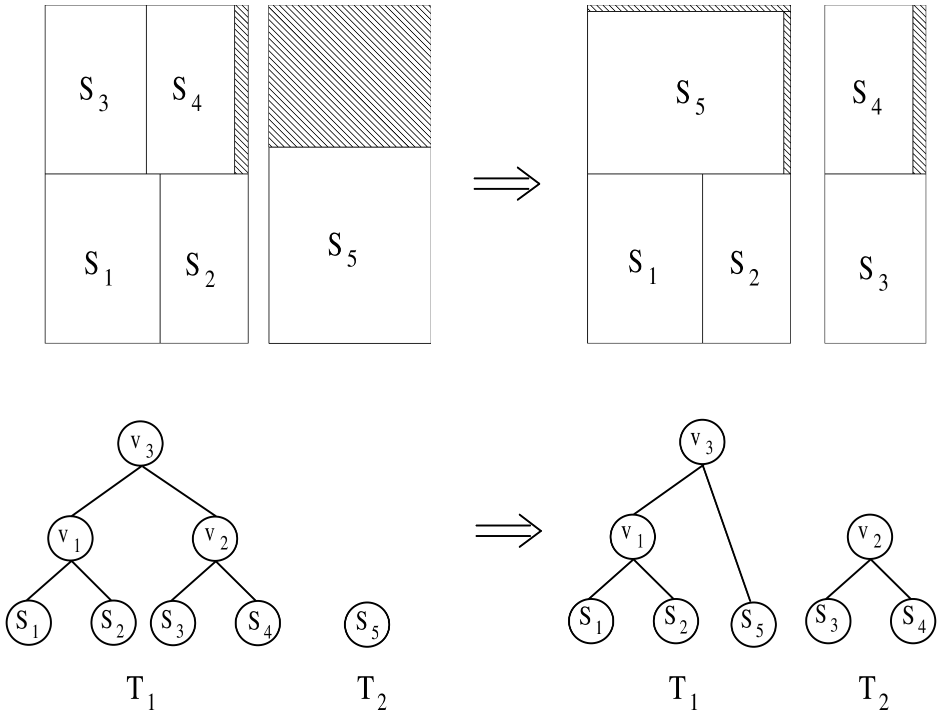


Abbildung 20: Teilbaumtausch zur Verbesserung der Gesamtlänge

mit dem Tausch der Teilbäume der zwei ersten Bäume in der Liste. Unsinnige Vertauschungen — zum Beispiel das Vertauschen zweier Blätter, die Scheiben der gleichen Größe repräsentieren — werden nicht durchgeführt. Nach jedem probeweisen Tausch wird überprüft, ob sich die Mindestrohlingszahl verringern würde. Wenn dem so ist, wird der Tausch tatsächlich durchgeführt und die Liste der Schnittbäume wird auf den neuesten Stand gebracht. Wenn nicht, so wird die nächste Tauschmöglichkeit überprüft. Es gibt zwei Abbruchkriterien für diese Art der Nachoptimierung. Das Verfahren bricht ab, wenn kein Tausch mehr gefunden werden konnte, der die Rohlingszahl sinken läßt. Da nicht sichergestellt ist, daß dieser Zeitpunkt in annehmbarer Zeit erreicht wird, ist es sinnvoll ein zeitliches Limit zu setzen, bei dessen Erreichen die Nachoptimierung abbrechen soll. Anhand der getesteten Instanzen ließ sich eine Tendenz dahingehend feststellen, daß der "natürliche" Abbruch der Nachoptimierung in den meisten Fällen nach annehmbarer Zeit eintritt. Die benötigten Zeiten und die erzielten Verbesserungen durch Phase 3 werden in Kapitel 6 genauer untersucht.

5.4 Phase 4: Abschnitte auf Rohlinge verteilen

In der letzten Phase des Verfahrens muß festgelegt werden, welcher Abschnitt aus welchem Rohling ausgeschnitten werden soll. Dabei soll die Länge eines jeden Rohlings möglichst komplett genutzt werden. Von dem letzten angebrochenen Rohling soll eine möglichst lange Restplatte abfallen. Als Prüfwert für die Nachoptimierung in Phase 3 wurde die Mindestrohlingszahl verwendet. Diese Mindestzahl wird sich in den meisten Fällen nicht einhalten lassen, da sie das Problem der Abschnittzuordnung idealisiert. Es ist beim Bestimmen der Mindestrohlingszahl erlaubt, einen Abschnitt in Querrichtung zu zerteilen, um genau die Länge eines Rohlings zu füllen. Dieses Zerteilen ist in der Praxis jedoch unsinnig, da hierdurch die bestellten Scheiben zerstört würden. Als Prüfwert für die Güte der Abschnittverteilung auf die Rohlinge wird daher die in Abschnitt 4.1 definierte *reelle Rohlingszahl* verwendet. Je kleiner die reelle Rohlingszahl ist, desto besser ist die Aufteilung der Abschnitte und desto geringer ist der Verschnitt.

Es muß ein Verfahren gefunden werden, welches beim Einplanen der Abschnitte deren Längen berücksichtigt. Da die Breite der vorliegenden Abschnitte nahezu der Rohlingsbreite B entspricht, soll die Breite eines jeden Abschnitts konkret als B festgelegt werden. Ein Abschnitt wird wie beim Bottom–Left–Verfahren so weit wie möglich links und so weit wie möglich unten in einem Rohling plaziert. Durch das Festlegen der Breite auf den Wert B ist es nicht mehr möglich, etwas über einem Abschnitt zu plazieren; die Abschnitte liegen ausschließlich nebeneinander. Bei der Anordnung der Abschnitte handelt es sich somit um ein eindimensionales Packproblem. Ein solches Problem ist mit Hilfe des First–Fit–Decreasing heuristisch zu lösen. Sei k die Anzahl der vorliegenden Abschnitte. Mit abschnitt[i]. x werde die Länge des i–ten Abschnitts angegeben. In abschnitt[i]. marke werde die Nummer des Rohlings vermerkt, aus dem der i–te Abschnitt ausgeschnitten werden soll. Es gelte abschnitt[i]. x $\geq$ abschnitt[$i+1$]. x für $i = 1, \ldots, k-1$. Eine algorithmische Beschreibung des FFD läßt sich wie folgt angeben:

1. FOR $i = 1$ TO k DO abschnitt[i]. marke:= 0; END;
2. nummer:= 0;
3. WHILE (es gibt noch Abschnitte mit Marke Null) DO
4.   nummer:=nummer+1;
5.   rest:= L;
6.   FOR $i = 1$ TO k DO
7.   IF (abschnitt[i]. marke= 0) AND (abschnitt[i]. x$\leq$rest) THEN
8.   rest:=rest-abschnitt[i]. x;

```
            abschnitt[i].marke:=nummer;
        END;
    END;
END;
```

Beim Testen des in dieser Arbeit entwickelten Verfahrens konnte festgestellt werden, daß die Ergebnisse von FFD für die Abschnittsverteilung nicht befriedigend sind. Daher wird das FFD ausschließlich dafür benutzt, eine Startlösung für die Verteilung der Abschnitte zu erhalten. Ausgehend von dieser Startlösung werden dann einzelne Abschnitte oder Blöcke von Abschnitten zwischen den Rohlingen ausgetauscht auf der Suche nach einer verbesserten Lösung. Ob sich die Lösung verbessert hat, läßt sich anhand der reellen Rohlingszahl ermitteln. Je kleiner die reelle Rohlinszahl, desto dichter sind die Rohlinge gepackt und desto geringer ist der Gesamtverschnitt. Um den Austausch der Abschnitte zwischen den von FFD erzeugten Rohlingen zu organisieren, werden die Rohlinge aufsteigend nach Länge der von ihnen abfallenden Restplatte sortiert. Für jeden Rohling wird eine Liste angelegt, in der alle möglichen Lückengrößen vermerkt sind, die entstehen können, wenn einer oder mehrere der Abschnitte aus dem Rohling entfernt werden. Dabei ist die Länge der jeweiligen Restplatte in jedem Fall mit zu berücksichtigen, da die Restplatte selbst eine Lücke in dem Rohling darstellt. Subtrahiert man von den Elementen einer Lückenliste die Länge der Restplatte, so erhält man die Längenlisten. Diese geben an, welche möglichen Längen von Abschnitten oder Blöcken von Abschnitten aus einem Rohling herausgenommen werden können. Abbildung 21 zeigt zwei Rohlinge mit ihren Längen– und Lückenlisten. Um das Vorgehen des Vertauschungsverfahrens zu erklären, seien ein Rohling R_i und ein Rohling R_j gegeben, wobei $j > i$ gelte. Es wird ein Wert gap_i in der Lückenliste des Rohlings R_i und ein Wert len_j in der Längenliste des Rohlings R_j gesucht, so daß

1. $gap_i \geq len_j$ ist
2. der zu gap_i gehörige Wert len_i in der Längenliste von Rohling R_i kleiner oder gleich dem zu len_j gehörigen Wert gap_j aus der Lückenliste des Rohlings R_j ist
3. durch einen Austausch der Abschnitte, aus denen len_i und len_j resultieren, die Länge der Restplatte in Rohling R_i schrumpft.

Lassen sich eine solche Lücke und eine solche Länge finden, so werden die Abschnitte, aus denen gap_i beziehungsweise len_j resultieren, ausgetauscht. Nach dem oben angegebenen Prinzip werden alle möglichen Vertauschungen zwischen Rohling R_i und Rohling R_j durchgeführt. Man beginnt die Vertauschung

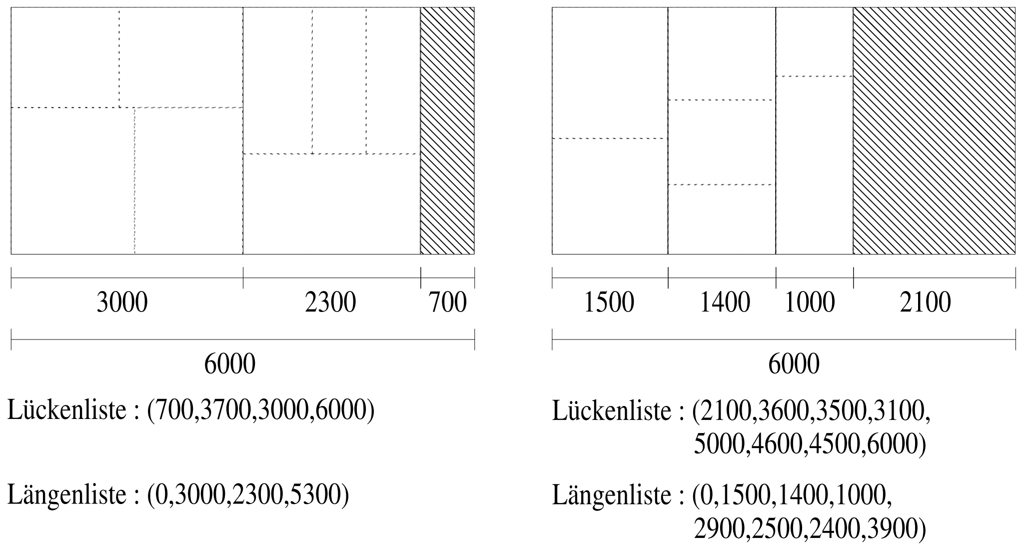


Abbildung 21: Zwei Rohlinge mit Lücken– und Längenlisten

mit Rohling R_i mit $i = 1$ und Rohling R_j mit $j = 2$. Läßt sich keine weitere Verringerung der Restplattenlänge in Rohling R_i mehr finden, so wird die Rohlingsliste neu sortiert, die Lücken und Längenlisten werden neu berechnet, und man versucht nun, Verbesserungen zwischen Rohling R_{j+1} und den Rohlingen R_i mit $i = 1, \ldots, j$ zu erzielen. Dieses Verfahren setzt man fort, bis auch bei Vertauschungen zwischen R_k und R_{k-1} keine Verbesserung mehr erzielt werden konnte, wobei k die Anzahl der nach Anwendung von FFD vorliegenden Rohlinge sei. Es werden also nacheinander die Restplattenlängen in den Rohlingen verringert, bzw. in dem Rohling mit längster Restplatte wird die Restplattenlänge vergrößert. Durch Austauschen der Abschnitte verringert sich die reelle Rohlingszahl. Für die beiden Rohlinge aus Abbildung 21 zeigt die Abbildung 22 einen möglichen Abschnittsaustausch. Die Restplattenlänge des zweiten Rohlings hat sich dabei um 600 vergrößert, die genutzte Länge des ersten Rohlinges um 600 erhöht. Lägen mehr als zwei Rohlinge vor, so würde nun durch Austausch der Abschnitte versucht werden, die Länge der Restplatte des dritten Rohlings zu erhöhen.

Dieses Vorgehen erscheint sehr aufwendig. Die Anzahl der Abschnitte und der Rohlinge ist in der Praxis jedoch meist so gering, daß die Laufzeiten für dieses modifizierte First–Fit–Decreasing (MFFD) nur einen Bruchteil der Gesamtlaufzeit des Verfahrens ausmacht. Genauere Werte sind in Kapitel 6 angegeben. Dort findet auch ein Vergleich zwischen Lösungen, die mit FFD, und solchen, die mit MFFD berechnet wurden, statt.

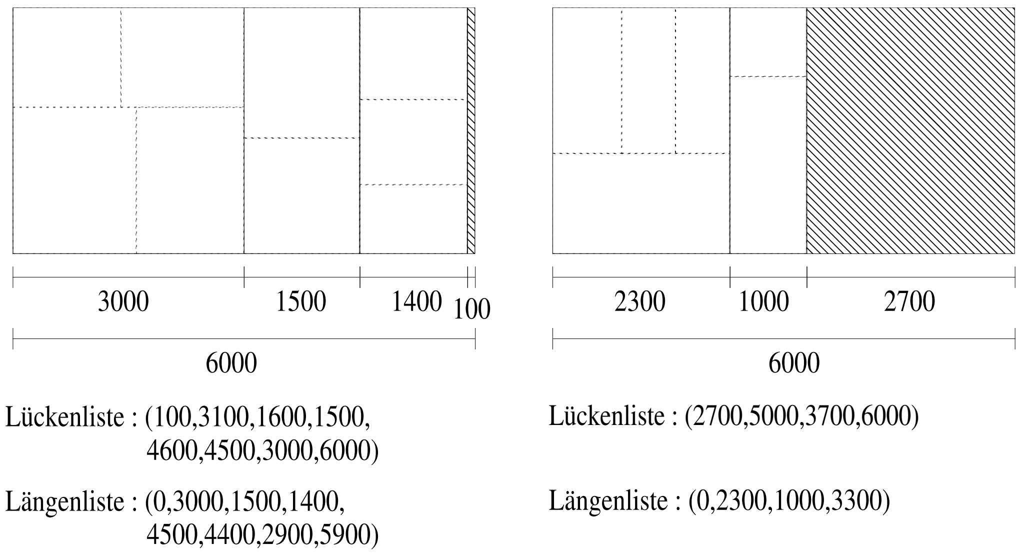


Abbildung 22: Abschnitte vertauschen zwischen zwei Rohlingen

Es liegt nun also eine Menge von Schnittbäumen vor, von denen jeder einen Rohlingsabschnitt repräsentiert. Besser gesagt läßt sich in der Formfunktion des Wurzelknotens eines jeden Schnittbaumes eine Sprungstelle finden, für die die Gestalt des Rechtecks einen Rohlingsabschnitt darstellt. Für jeden dieser Abschnitte wurde in einer Marke vermerkt, von welchem Rohling er abgeschnitten werden soll. Es werden nun nacheinander alle Schnittbäume, die einen Abschnitt mit gleicher Rohlingsmarke repräsentieren, zusammengesetzt. Die Reihenfolge, in der die einzelnen Bäume zusammengesetzt werden, spielt dabei keine Rolle. Zwei Bäume erhalten einen gemeinsamen Vaterknoten, in dem die Formfunktion vermerkt wird, die sich aus der Komposition der Formfunktionen in den Wurzelknoten der zusammenzusetzenden Bäume ergibt. Der neue Schnittbaum stellt wiederum einen (längeren) Rohlingsabschnitt dar und wird mit derselben Rohlingsmarke versehen, wie sie die beiden gepaarten Abschnitte besaßen. Die vorliegenden Abschnitte werden solange zusammengesetzt, bis für jede Rohlingsmarke genau ein Abschnitt vorliegt. Man beachte, daß durch das Kombinieren der Schnittbäume durchaus Schnittmuster entstehen können, die eine geringere Rohlingslänge benötigen als die Summe der Längen der zusammengesetzten Abschnitte beträgt. Dies liegt daran, daß durch das Kombinieren ein Metarechteck entstehen kann, das eine Gestalt mit dichterer Plazierung besitzt als die nicht kombinierten Rechtecke.

5.5 Schnittbaumtraversierung

Am Ende der vierphasigen Verschnittoptimierung durch iteriertes Matching liegt eine Menge von Schnittbäumen vor, von denen jeder ein (oder mehrere) Schnittmuster für einen Rohling repräsentiert. Die Schnittmuster lassen sich, wie in Kapitel 4 beschrieben, durch einen Top–Down–Durchlauf des Baumes realisieren.

Bei der Implementation des Programms ist darauf geachtet worden, eine allgemeine Form zur Darstellung der Schnittmuster zu erhalten. Es ist in der Praxis nicht sinnvoll, der Steuereinheit eines Schneidtisches den kompletten Schnittbaum zu übermitteln (vgl. Kapitel 3). Die Schnittbäume enthalten eine große Anzahl möglicher Gestalten der Metarechtecke. Bei einer konkreten Glastafelgröße ist jedoch nur eine Gestalt pro Schnittbaum für den Zuschnitt festzulegen. Bei der Traversierung des Baumes werden daher nur die Daten dieser einen Gestalt in einer Datei gespeichert. Diese Datei gibt dann das zu einem Rohling gehörige Schnittmuster an.

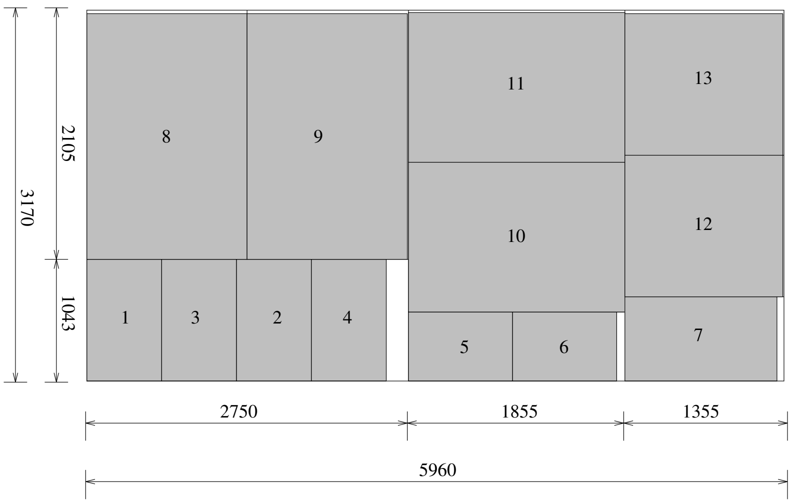


Abbildung 23: Schnittmuster eines Rohlings

Abbildung 24 zeigt die Datei, aus der das Schnittmuster für den in Abbildung 23 dargestellten Rohling generiert wurde.

```
Anzahl Scheiben      : 13
Rohlingslaenge       : 5960
Rohlingsbreite       : 3170
genutzte Laenge      : 5960
Flaeche d. Scheiben  : 18510399
prozent. Verschnitt  : 2.026131
Scheibendaten (Nummer Laenge Breite   Nummer Laenge Breite):
1   643 1043    2   643 1043
3   643 1043    4   643 1043
5   593  898    6   593  898
7  1305  725    8  2105 1375
9  2105 1375    10 1855 1280
11 1855 1280    12 1355 1210
13 1355 1210
Paarungsstring:
II=II1-3-I2-4-I8|9|=I5|6|=10-11-=7-=12-13-
Schnittanweisungen:
c 4605 0 4605 3170
c 2750 0 2750 3170
c 0 1043 2750 1043
c 1286 0 1286 1043
c 643 0 643 1043
r 643 1043 0 0 1
r 643 1043 643 0 3
c 1929 0 1929 1043
r 643 1043 1286 0 2
r 643 1043 1929 0 4
c 1375 1043 1375 3170
r 1375 2105 0 1043 8
r 1375 2105 1375 1043 9
c 2750 593 4605 593
c 3648 0 3648 593
r 898 593 2750 0 5
r 898 593 3648 0 6
c 2750 1873 4605 1873
r 1855 1280 2750 593 10
r 1855 1280 2750 1873 11
c 4605 725 5960 725
r 1305 725 4605 0 7
c 4605 1935 5960 1935
r 1355 1210 4605 725 12
r 1355 1210 4605 1935 13
```

Abbildung 24: Daten eines Schnittmusters

Am Anfang der Datei wird die Anzahl der Scheiben angegeben, die aus dem Rohling erzeugt werden sollen. Sie entspricht der Anzahl der Blätter des Schnittbaumes, der diesen Rohling repräsentiert. Die Rohlingsdaten werden ebenfalls aufgeführt. Die genutzte Rohlingslänge wird ermittelt, indem in der Formfunktion der Wurzel des Schnittbaumes die erste Sprungstelle mit y–Wert kleiner oder gleich der Rohlingsbreite aufgesucht wird. Die Gesamtfläche der Scheiben wird ebenfalls aus den Blättern des Schnittbaumes ermittelt. Mit Hilfe der Rohlingsfläche läßt sich dann der prozentuale Verschnitt des Rohlings berechnen.

Um die Ausdehnungen der einzelnen Scheiben zur Verfügung zu haben, werden die Scheibendaten der Instanz mit in die Datei aufgenommen. Als nächstes wird der sogenannte Paarungsstring erzeugt. Er entsteht aus einer Preorder–Traversierung des Schnittbaumes. Aus ihm lassen sich die Ausrichtungen der Scheiben und die Art der Paarung (horizontal oder vertikal) ermitteln. Wird eine Scheibe mit ihrer Längsseite parallel zur Längsseite des Rohlings eingeplant, so wird in dem Paarungsstring die Scheibennummer gefolgt von einem waagerechten Strich "–" eingetragen. Andernfalls folgt der Scheibennummer ein senkrechter Strich "|". Vor einem Rechteckpaar ist jeweils die Art der Paarung vermerkt. Dabei bedeutet "I", daß ein vertikaler und "=", daß ein horizontaler Schnitt die Rechtecke trennt.

Mit Hilfe des Paarungsstrings und der Daten in den Knoten des Schnittbaumes lassen sich die Schnittanweisungen und Plazierungen der Scheiben in dem Rohling ermitteln. Dazu wird der zu zerschneidende Rohling in einem Koordinatensystem plaziert, wobei die linke untere Rohlingsecke im Ursprung liegt. Die zu einem Schnitt gehörigen Daten werden mit einem "c" gekennzeichnet, die zu einer Scheibe gehörigen Daten mit einem "r". Dieselbe Sprungstelle, aus der sich die genutzte Rohlingslänge ermitteln ließ, liefert auch den ersten durchzuführenden Schnitt. In dem hier vorliegenden Fall handelt es sich um einen vertikalen Schnitt an der x–Koordinate 4605. Da es sich um Guillotineschnitte handelt, verläuft der Schnitt über die gesamte Rohlingsbreite von y–Koordinate 0 bis 3170. Die Schnittdaten werden in der Form

$$c \quad x_1 \quad y_1 \quad x_2 \quad y_2$$

angegeben. Der Schnitt verläuft vom Punkt (x_1, y_1) zum Punkt (x_2, y_2). Der Rohling ist somit in zwei Teile zerfallen. Wie diese beiden Teile weiter zerlegt werden müssen, läßt sich aus den Formfunktionen in den beiden Söhnen der Wurzel des Schnittbaumes ermitteln. Es werden alle Schnitte durchgeführt, die sich bei der Traversierung des Baumes ergeben. Durch die Schnitte erfolgt eine Aufteilung des Rohlings in einzelne Sektoren, in denen die bestellten Scheiben liegen. Da in manchen Fällen der Sektor größer ist als die herzustellende Scheibe, müssen für einige Scheiben Trimmzuschnitte durchgeführt werden. Daher sind zusätzlich die Scheibendaten in der Datei vermerkt. Die Angabe über die Plazierung einer Scheibe erfolgt in der Form

$$r \quad x \quad y \quad nx \quad ny \quad nr.$$

Die Nummer der Scheibe, um deren Daten es sich handelt, wird in nr angegeben. Die Ausdehnung der Scheibe in x–Richtung entspricht dem Wert x und die Ausdehnung der Scheibe in y–Richtung dem Wert y. Je nach Ausrichtung der Scheibe entspricht x der Länge oder Breite und y entsprechend der Breite oder Länge der Scheibe mit Nummer nr. Die Werte nx und ny geben die Lage der linken unteren Ecke der Scheibe im Rohling an.

Anhand einer solchen Datei lassen sich dann problemlos die Rohlinge in die bestellten Scheiben zerlegen. In der Datei sind alle notwendigen Daten gespeichert. Zur Steuerung des Schneidtisches kann auf sie zurückgegriffen werden. Doch auch zur Ausgabe der Schnittmuster auf Drucker oder Bildschirm können diese Dateien verwendet werden. Die im nächsten Kapitel dargestellten Schnittmuster für die getesteten Instanzen beispielsweise werden alle anhand solcher Dateien erzeugt. Dazu werden die Daten über Schnitte und Rechtecke mit Hilfe eines kurzen C–Programms in das Dateiformat des Graphikprogramms XFIG konvertiert.

6 Testergebnisse im Vergleich

Das in den Kapiteln 4 und 5 vorgestellte Lösungsverfahren für zweidimensionale Verschnittprobleme in der Glasindustrie wurde im Rahmen dieser Diplomarbeit in der Programmiersprache C implementiert. Anhand einiger Testinstanzen soll nun die Effektivität der Verschnittoptimierung durch iteriertes Matching ermittelt werden. Um praxisnahe Testinstanzen zu erhalten, stellte die Firma Glas Honsel KG sieben Datensätze zur Verfügung. Die Datensätze geben jeweils eine Menge herzustellender Scheiben und die zu verwendenden Rohlinge mit ihren Ausdehnungen an. Bei den Datensätzen handelt es sich somit um Instanzen für das Glaszuschnittproblem 4.1, wie es in Abschnitt 4.1 beschrieben wurde. Die zur Verfügung gestellten Instanzen entsprechen tatsächlichen Bestellungen, die an die Firma Honsel gerichtet worden sind.

Die Firma Honsel benutzt zur Verschnittoptimierung das kommerzielle Computerprogramm XOPT, welches von der Firma Albert und Wirsam entwickelt wurde. Anhand der Schnittmuster, die dieses Programm für die sieben Testinstanzen erstellte, soll die Güte der Lösungen überprüft werden, die mit Hilfe des iterierten Matching erzeugt werden. Dazu werden jeweils Vergleiche bezüglich der benötigten Rohlingszahl, des mittleren Verschnitts und der Laufzeit der beiden Programme angestellt. Zusätzlich sollen die unterschiedlichen Ergebnisse des in dieser Arbeit entwickelten Verfahrens angegeben werden, die sich durch Anwenden beziehungsweise Nichtanwenden der Nachoptimierungen in Phase 3 und Phase 4 ergeben. Die Ergebnisse des Matchingprogramms wurden auf einem Rechner mit 486er Prozessor und einer Taktfrequenz von 33 MHz berechnet. Das kommerzielle Programm lief auf einem Rechner mit 386er Prozessor und einer Taktfrequenz von 16 MHz. Diese Tatsache muß im weiteren bei einem Vergleich der Laufzeiten der beiden Programme berücksichtigt werden.

Wie auch bei dem Matchingverfahren, liegen als Lösungen des Programms XOPT Schnittmuster vor, die für jeden Rohling zunächst mehrere Rohlingsabschnitte erzeugen. Dies liegt daran, daß in der Praxis sogenannte x–Traversen eingehalten werden müssen. Dabei handelt es sich um Schnitte, die orthogonal zur Längsseite des Rohlings verlaufen und die einen bestimmten Abstand zueinander haben müssen. Die Traversen entsprechen somit dem in Kapitel 3 eingeführten Wunsch nach Rohlingsabschnitten zur besseren Handhabung der zerteilten Glastafeln.

Um die Lösungen der beiden Programme zu vergleichen, sollen jeweils die Gesamtlänge aller Abschnitte, die Rohlingszahl und der mittlere prozentuale Verschnitt der Rohlinge betrachtet werden. Auf welche Weise die Gesamtlänge der Abschnitte und die Rohlingszahl ermittelt wird, wurde im vorangegangenen Text erläutert. Der mittlere prozentuale Verschnitt (MPV) in den berechneten Rohlingen einer Instanz läßt sich aus der reellen Rohlingszahl RZ, den

Rohlingsausdehnungen L und B und der Summe der Flächen aller N Scheiben mit Ausdehnungen l_i und b_i ($i \in \{1, \dots, n\}$) berechnen. Es gilt

$$\text{MPV} := 100 \cdot \left(1 - \frac{\sum_{i=1}^{N}(l_i \cdot b_i)}{L \cdot B \cdot RZ}\right)$$

Die entsprechenden Werte der Lösungen des Programms XOPT laßen sich aus den von Honsel bereitgestellten Schnittmustern entnehmen.

Es ist in der Praxis üblich, die linke und die untere Seite eines Rohlings mit einem sogenannten Anschnitt zu versehen. Dazu wird von diesen beiden Rohlingsseiten ein schmaler Rand abgeschnitten. Es werden in den Testinstanzen zwei Rohlingsarten verwendet, die sich in der Dicke des Glases unterscheiden. Je nach Dicke der Glastafeln werden unterschiedliche Anschnitte vollzogen. Die durch den Anschnitt abfallenden Glasstreifen werden nicht zum Verschnitt hinzugerechnet, da dieser Verlust unvermeidlich ist. Als Rohlingsdaten werden daher für die Instanzen die Länge und Breite des bereits angeschnittenen Rohlings angegeben, da diese die tatsächlich zur Verfügung stehende Rohlingsfläche beschreiben.

Die sieben Instanzen sind mit Honsel_1 bis Honsel_7 bezeichnet. Aus Notationsgründen werden in den Tabellen, die die Daten der Instanzen angeben, Scheiben gleicher Länge und Breite zu einer *Scheibenart* zusammengefaßt. Für jede Scheibenart wird die herzustellende Anzahl angegeben. Um die Daten in die in der Problemformulierung 4.1 gewünschte Form zu bringen, müssen alle Scheiben systematich durchnumeriert werden. Die sich daraus ergebene Scheibenanzahl N findet sich ebenfalls in den Tabellen. Die Summe aller Scheibenflächen wird zur Berechnung des MPV benötigt und ist daher ebenfalls aufgeführt. Die Maße werden, sofern nicht ausdrücklich erwähnt, in Millimetern angegeben.

In dem folgenden Abschnitt werden die Daten der sieben Instanzen angegeben und die Ergebnisse der Verschnittoptimierung durch iteriertes Matching erläutert. Dabei werden für jede Instanz die Unterschiede zwischen Lösungen mit und ohne Nachoptimierungen herausgestellt. Ein Vergleich zwischen den Ergebnissen des Programms XOPT und denen, die mit dem Matchingprogramm berechnet werden, erfolgt im letzten Abschnitt dieses Kapitels.

6.1 Ergebnisse des Matchingverfahrens

6.1.1 Instanz Honsel_1

Die erste Instanz umfaßt eine Bestellung von 91 Scheiben. Es liegen 39 verschiedene Scheibenarten vor. Die zur Verfügung stehenden Rohlinge besitzen die Längenausdehnung 5960 mm und die Breite 3170 mm. Die Summe aller Scheibenflächen macht eine Gesamtscheibenfläche von circa 110.36 m^2 aus. Diese Daten und die Ausdehnungen der einzelnen Scheibenarten sind in Tabelle 1 aufgeführt.

Honsel_1							
Rohlingsdaten	:	$L = 5960$ mm	$B = 3170$ mm				
Scheiben	:	$N = 91$ Stück	39 Arten				
Gesamtscheibenfläche	:	110356737 mm^2					
Art	Anzahl	Länge	Breite	Art	Anzahl	Länge	Breite
1	2	285	850	21	4	847	2017
2	2	342	1770	22	1	880	1735
3	1	416	989	23	2	951	1162
4	2	490	1835	24	6	975	805
5	2	561	2007	25	2	989	1308
6	2	565	700	26	2	1167	957
7	4	573	1918	27	1	1176	1355
8	1	615	1010	28	1	1226	1355
9	1	655	1930	29	2	1228	1320
10	1	675	1292	30	1	1245	1410
11	8	714	1126	31	2	1245	1332
12	6	720	877	32	4	1323	978
13	2	725	635	33	1	1325	1155
14	1	734	1928	34	2	1340	980
15	2	767	2175	35	2	1565	668
16	2	780	1775	36	3	1740	1185
17	2	792	2042	37	2	1785	1085
18	2	842	2135	38	2	2132	845
19	2	845	2135	39	2	2132	842
20	4	845	2015				

Tabelle 1: Daten der Instanz Honsel_1

Die Ergebnisse, die das Matchingprogramm liefert, werden in einer weiteren Tabelle angegeben. In dieser werden die nach jeder Phase des Verfahrens vorliegenden Ergebnisse aufgelistet. Nach Abschluß der Phasen 1 und 2, also nach dem Erzeugen der Rohlingsabschnitte, läßt sich die Qualität der bis dahin vorliegenden Lösung anhand der Gesamtlänge aller Rohlingsabschnitte bestimmen. Je kleiner die Gesamtlänge, desto geringer wird der Verschnitt sein. Die Verbesserung, die durch die Nachoptimierung in Phase 3 eintritt, kann ebenfalls an der vorliegenden Gesamtlänge festgestellt werden. Dazu berechne man die Differenz zwischen der Gesamtlänge vor und nach der dritten Phase.

Ein genaues Maß für den Verschnitt ist allein durch die Gesamtlänge jedoch noch nicht gegeben. Daher werden zusätzlich die Mindestrohlingszahl und die reelle Rohlingszahl angegeben. Die reelle Rohlingszahl kann erst bestimmt werden, wenn nach der vierten Phase die Abschnitte den Rohlingen zugeordnet worden sind. Die Qualität dieser Abschnittsverteilung läßt sich aus der Differenz zwischen reeller Rohlingszahl und Mindestrohlingszahl ablesen. Je kleiner die Differenz ist, desto näher liegt die vom Matchingprogramm vorgenommene Abschnittsverteilung an der Idealverteilung. Um ein Maß für die Güte des in Phase 4 benutzten modifizierten First–Fit–Decreasing zu erhalten, werden zusätzlich die Ergebnisse angegeben, die aus einer Anwendung des "normalen" FFD resultieren würden.

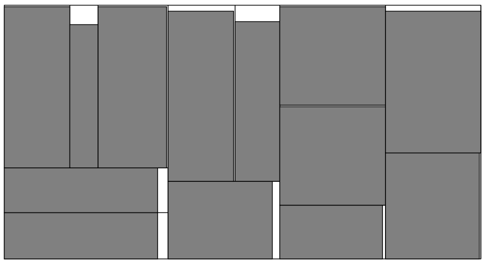
Da die Rohlingszahl ein sehr abstrakter Begriff ist, wird außerdem der mittlere prozentuale Verschnitt (MPV) in den Rohlingen der Instanz bestimmt. Die Berechnung des MPV wurde zu Beginn dieses Kapitels erläutert.

Verschnittoptimierung für Honsel_1					
Schritt	Zeit in Sekunden	GL in mm	reelle RZ	Mindest– RZ	MPV
Matching	39	37072	—	6.220	—
Nachopti	31	36343	—	6.098	—
FFD	< 1	36343	6.333	6.098	7.76
MFFD	1	36343	6.103	6.098	4.29

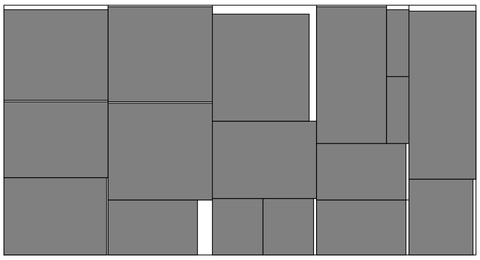
Tabelle 2: Ergebnisse für Instanz Honsel_1

Tabelle 2 stellt die Ergebnisse des Matchingverfahrens für die Instanz Honsel_1 dar. Für jeden Schritt ist die benötigte Zeit in Sekunden angegeben. Das Erzeugen der Abschnitte durch iteriertes Matching dauerte in diesem Fall somit 39 Sekunden. Die Nachoptimierung in Phase 3, die die Gesamtlänge aller Abschnitte um 729 mm auf 36343 mm sinken ließ, nahm 31 Sekunden in Anspruch. Die Rohlingszahlen sind auf drei Nachkommastellen gerundet angegeben. Die Mindestrohlingszahl für eine idealisierte Verteilung beträgt 6.098. Das bedeutet: Könnte man die Rohlinge so mit den vorliegenden Abschnitten füllen, daß bis auf einen Rohling in jedem die gesamte Rohlingslänge genutzt würde, so würde man 6.098 Rohlinge benötigen. Obwohl dieser idealisierte Fall recht selten eintreten wird (ohne die Abschnitte in y–Richtung zu zerschneiden), stellt die Mindestrohlingszahl einen guten Vergleichswert zur reellen Rohlingszahl dar.

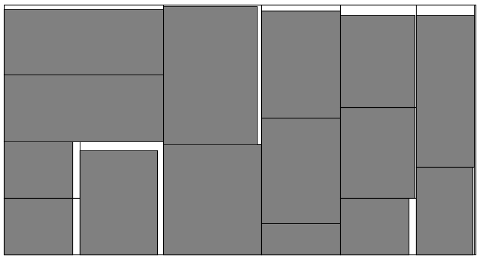
Würde das Matchingprogramm die Abschnitte mit FFD auf die Rohlinge verteilen, so würden 6.333 Rohlinge benötigt werden. Der Unterschied in der benötigten Rohlingslänge, der zwischen reeller Aufteilung nach FFD und idealisierter Aufteilung vorläge, betrüge circa 1.4 Meter. Das bedeutet, daß von dem am wenigsten gefüllten Rohling 1.4 m der Länge mehr genutzt würde, als im Idealfall. Das Verfahren plant die Rohlingsabschnitte jedoch nicht nach FFD, sondern nach MFFD ein. In diesem Fall liegt eine reelle Rohlingszahl von 6.103 vor. Es wird daher im reellen Fall nur eine Länge von circa 30 mm mehr benötigt als im Idealfall. Der MPV verringert sich durch die Benutzung von MFFD auf 4.29 Prozent gegenüber 7.76 Prozent, wenn FFD benutzt würde.



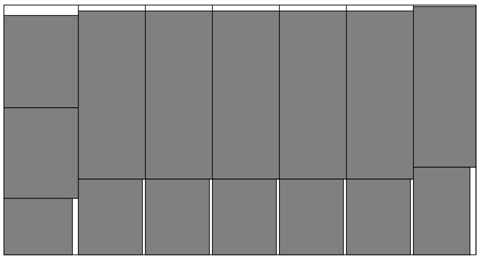
Schnittmuster Rohling 1; proz. Verschnitt: 3.22



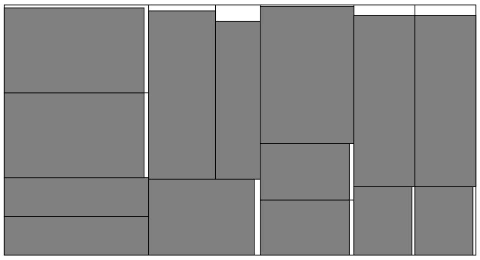
Schnittmuster Rohling 2; proz. Verschnitt: 3.46



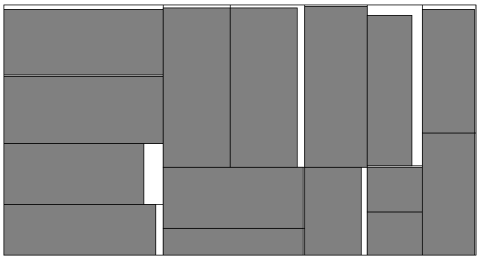
Schnittmuster Rohling 3; proz. Verschnitt: 5.25



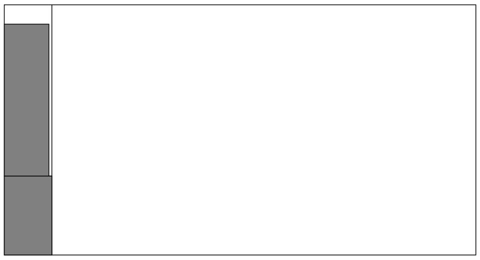
Schnittmuster Rohling 4; proz. Verschnitt: 3.74



Schnittmuster Rohling 5; proz. Verschnitt: 3.87



Schnittmuster Rohling 6; proz. Verschnitt: 5.45



Schnittmuster Rohling 7; proz. Verschnitt: 11.77 (90.90)

Abbildung 25: Schnittmuster der Instanz Honsel_1

Die einzelnen Schnittmuster für Instanz Honsel_1, die durch das Matchingprogramm berechnet worden sind, lassen sich der Abbildung 25 entnehmen. Da diese Art der Darstellung sehr aufwendig ist und sich den Schnittmustern keine genauen Daten entnehmen lassen, sollen für alle Instanzen die einzelnen Rohlingsdaten zusätzlich in einer Tabelle angegeben werden. Zur Beschreibung der Rohlinge reicht es aus, den jeweiligen prozentualen Verschnitt und die genutzte Rohlingslänge aufzuführen. Ist der Verschnitt hoch und die Länge komplett genutzt, so ist das vom Matching generierte Schnittmuster schlecht, da die einzelnen Paarungen einen hohen Verschnitt verursachen. Ist die Rohlingslänge nicht komplett genutzt, so wird ein Teil des Verschnitts in dem Rohling durch den nicht genutzten Rohlingsstreifen verursacht. Das bedeutet, daß bei einem geringen prozentualen Verschnitt und einer nicht ganz genutzten Rohlingslänge die Scheiben, die aus dem Rohling geschnitten werden, vom Matchingverfahren dicht plaziert worden sind.

Verschnitte und genutzte Längen für Honsel_1					
Rohling Nr.	Verschnitt in Prozent	genutzte Länge	Rohling Nr.	Verschnitt in Prozent	genutzte Länge
1	3.22	5959	5	3.87	5960
2	3.46	5960	6	5.45	5950
3	5.25	5940	7	11.77 (90.90)	615
4	3.74	5959			

Tabelle 3: Erzeugte Rohlinge für Instanz Honsel_1

Die Daten der für Instanz Honsel_1 erzeugten Rohlinge sind in Tabelle 3 aufgeführt. Der Verschnitt des Rohlings von dem die Restplatte abfällt — also der Rohling mit der geringsten Längennutzung — wird für den tatsächlich genutzten Rohlingsbereich angegeben, denn die Restplatte kann für andere Zuschnitte weiterverwendet werden und stellt somit keinen wirklichen Verschnitt dar. Der komplette Verschnitt dieses Rohlings wird jedoch zusätzlich in Klammern angegeben. Für die anderen Rohlinge wird zur Berechnung des Verschnitts jeweils die gesamte Rohlingsfläche zugrunde gelegt. Pro Instanz fällt somit höchstens eine Restplatte ab.

6.1.2 Instanz Honsel_2

Tabelle 4 beschreibt die Werte der Parameter für die zweite Instanz. Es liegen insgesamt 116 bestellte Scheiben vor. Die zur Verfügung stehenden Glastafeln haben eine Längenausdehnung von 5960 mm und eine Breitenausdehnung von 3170 mm. Die Summe aller Scheibenflächen beträgt ca. 131.31 m^2.

Honsel_2							
Rohlingsdaten	:	$L = 5960$ mm	$B = 3170$ mm				
Scheiben	:	$N = 116$ Stück	45 Arten				
Gesamtscheibenfläche	:	131309371 mm^2					
Art	Anzahl	Länge	Breite	Art	Anzahl	Länge	Breite
1	2	325	1153	24	2	972	777
2	2	325	797	25	2	982	1197
3	1	390	857	26	4	1000	1865
4	4	547	1950	27	2	1031	777
5	4	570	570	28	2	1032	777
6	4	570	1950	29	4	1034	777
7	4	575	1950	30	1	1055	785
8	6	593	898	31	6	1068	777
9	2	629	1763	32	10	1074	777
10	4	643	1043	33	2	1077	1432
11	2	753	1085	34	4	1079	777
12	2	755	1895	35	2	1084	1966
13	1	765	800	36	2	1305	725
14	2	790	1880	37	2	1307	1393
15	1	816	832	38	2	1355	1210
16	2	824	590	39	2	1395	1280
17	2	835	1295	40	2	1414	786
18	2	842	2307	41	2	1420	1325
19	1	865	2170	42	2	1855	1280
20	2	870	1280	43	2	2105	1375
21	1	932	857	44	2	2210	1280
22	2	952	777	45	2	2215	1280
23	2	953	1116				

Tabelle 4: Daten der Instanz Honsel_2

In Tabelle 5 sind die Ergebnisse des Matchingprogramms mit den Zwischenergebnissen aufgeführt.

Durch iteriertes Matching wurden in den Phasen 1 und 2 innerhalb von 54 Sekunden Rohlingsabschnitte mit einer Gesamtlänge von 43826 mm erzeugt. Nach der dritten Phase hat sich diese Gesamtlänge um 480 mm verringert.

Würden die Abschnitte nach dem "normalen" FFD–Algorithmus eingeplant, so würden sie nicht besonders günstig den Rohlingen zugeordnet werden. Trotz einer Mindestrohlingszahl von 7.272 würde eine reelle Rohlingszahl von 7.760 benötigt. Der Längenunterschied zwischen reeller und Idealverteilung betrüge

Verschnittoptimierung für Honsel_2					
Schritt	Zeit in Sekunden	GL in mm	reelle RZ	Mindest– RZ	MPV
Matching	54	43826	—	7.353	—
Nachopti	9	43346	—	7.273	—
FFD	< 1	43343	7.760	7.272	10.49
MFFD	1	43334	7.304	7.271	4.85

Tabelle 5: Ergebnisse für Instanz Honsel_2

circa 2.90 Meter. Die Abschnitte werden jedoch mit Hilfe des MFFD den Rohlingen zugeordnet. Bei einer Mindestrohlingszahl von 7.271 und einer reellen Rohlingszahl von 7.304 beträgt der Unterschied in der genutzten Länge nur circa 19 Zentimeter. Der Unterschied in den von FFD und MFFD benötigten Laufzeiten ist unerheblich. Aus der Tabelle läßt sich entnehmen, daß die Gesamtlänge der Abschnitte in den Schritten FFD und MFFD noch einmal schrumpft. Dies liegt daran, daß nachdem die Abschnitte den Rohlingen zugeordnet worden sind, die einzelnen Schnittbäume der Abschnitte zusammengefügt werden, so daß am Ende des Verfahrens für jeden Rohling genau ein Schnittbaum vorliegt. Durch dieses Zusammensetzen der Bäume kann sich die Gesamtlänge weiter verringern, da — aufgrund der Verwendung von Formfunktionen — neue mögliche Gestalten für die Abschnitte aus den entstehenden Formfunktionen ermittelt werden können. Diese Möglichkeit wurde bereits in Abschnitt 5.4 erläutert.

In Tabelle 6 sind die einzelnen prozentualen Verschnitte und die genutzten Längen der erzeugten Rohlinge angegeben.

Verschnitte und genutzte Längen für Honsel_2					
Rohling Nr.	Verschnitt in Prozent	genutzte Länge	Rohling Nr.	Verschnitt in Prozent	genutzte Länge
1	7.43	5958	5	6.19	5858
2	3.19	5959	6	2.50	5919
3	6.77	5940	7	3.92	5948
4	4.34	5938	8	3.58 (70.65)	1814

Tabelle 6: Erzeugte Rohlinge für Instanz Honsel_2

Anhand der Daten des ersten Rohlings läßt sich die bereits erwähnte Auswertungsmöglichkeit der Paarungsgüte erläutern. Für Rohling 1 liegt bei fast komplett genutzter Länge ein Verschnitt von 7.43 Prozent vor. Dies läßt vermuten, daß die vom Matching bestimmten Paarungen in diesem einen Rohling nicht besonders günstig sind.

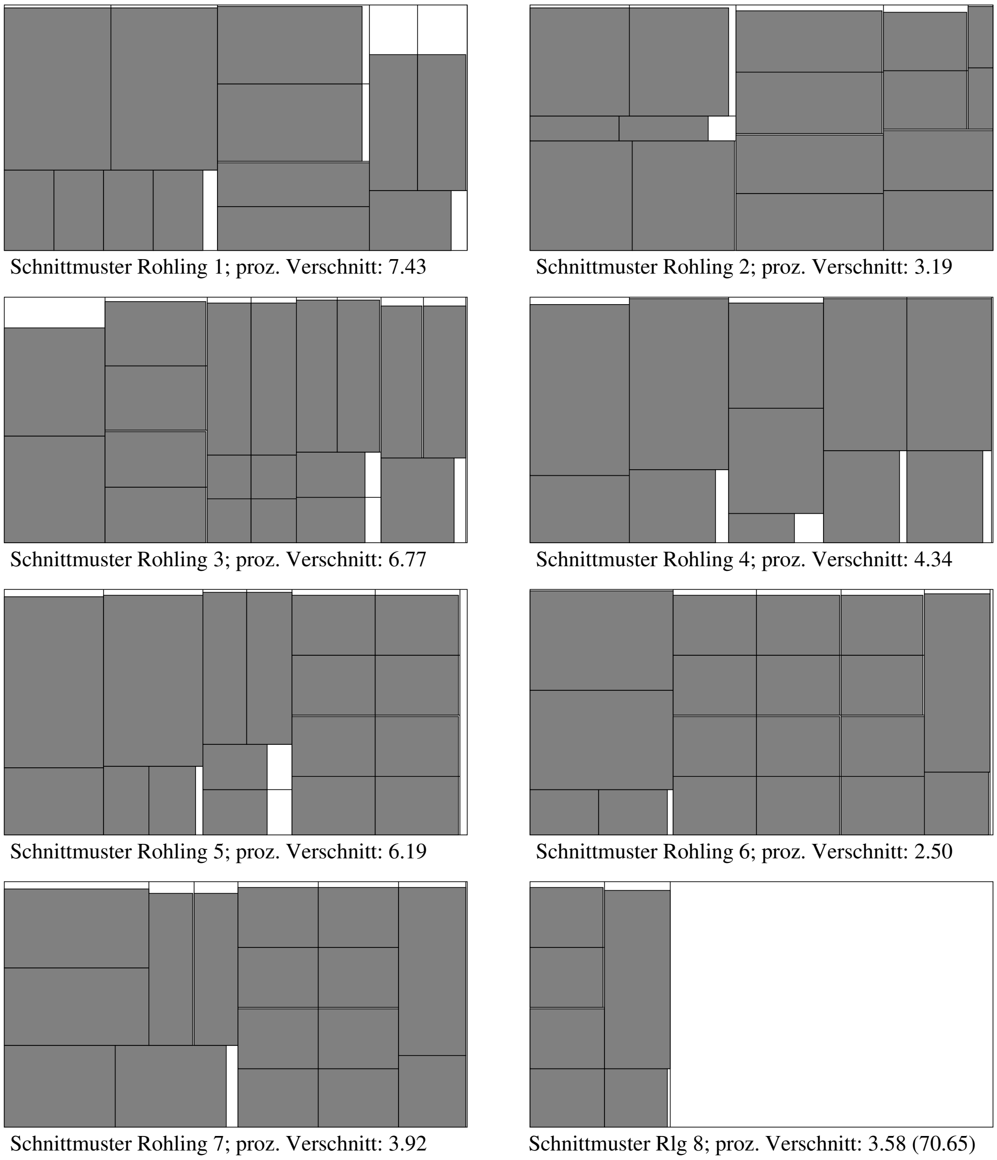


Abbildung 26: Schnittmuster der Instanz Honsel_2

Betrachtet man das Schnittmuster dieses Rohlings in Abbildung 26, so bestätigt sich diese Vermutung. Insgesamt jedoch beträgt der mittlere prozentuale Verschnitt pro Rohling in dieser Instanz lediglich 4.85 Prozent.

6.1.3 Instanz Honsel_3

In Tabelle 7 sind die Daten der dritten Instanz angegeben. Man beachte, daß die Rohlingsausdehnungen sich von denen der anderen Instanzen unterscheiden. Die Ursache hierfür liegt in einem anderen Anschnitt der Rohlinge. Die bestellten Scheiben sollten aus 5 mm dickem Floatglas hergestellt werden. Für die Scheiben der anderen Instanzen wird Glas der Dicke 4 mm verwendet.

Honsel_3			
Rohlingsdaten	:	$L = 5940$ mm	$B = 3150$ mm
Scheiben	:	$N = 52$ Stück	17 Arten
Gesamtscheibenfläche	:	58237035 mm^2	

Art	Anzahl	Länge	Breite	Art	Anzahl	Länge	Breite
1	6	423	423	10	1	905	682
2	1	632	760	11	1	992	1023
3	2	670	1023	12	1	992	682
4	2	670	1848	13	2	1000	836
5	2	813	1088	14	1	1023	905
6	4	813	908	15	4	1078	1023
7	1	813	1908	16	2	1078	1848
8	6	823	1908	17	2	1078	914
9	14	823	1898				

Tabelle 7: Daten der Instanz Honsel_3

Diese Instanz beinhaltet von allen Instanzen die kleinste Menge herzustellender Scheiben. Wie im Abschnitt 4.1 erwähnt, werden in der Praxis die Bestellungen mehrerer Kunden für eine Optimierung zusammengefaßt. Dabei sollten mindestens 50 Scheiben herzustellen sein, da sonst die Anzahl der möglichen Schnittmuster zu gering wird. Ist die Anzahl möglicher Schnittmuster gering, so ergeben sich dann in den meisten Fällen Rohlinge mit höherem MPV, da weniger Kombinationsmöglichkeiten für die Scheiben vorliegen. Diese Tatsache spiegelt sich auch in den Ergebnissen in Tabelle 8 wider.

Der mittlere prozentuale Verschnitt ist im Gegensatz zu den Ergebnissen der anderen Instanzen relativ hoch. Im Vergleich zu dem von der Firma Honsel verwendeten Programm XOPT wird sich zeigen, daß auch dieses Programm keine bessere Verschnittoptimierung für diese relativ kleine Anzahl Scheiben ermitteln konnte.

Die einzelnen prozentualen Verschnitte und genutzten Längen der erzeugten Rohlinge sind in der Tabelle 9 aufgeführt. Man beachte dabei, daß die nicht

Verschnittoptimierung für Honsel_3					
Schritt	Zeit in Sekunden	GL in mm	reelle RZ	Mindest– RZ	MPV
Matching	17	21283	—	3.583	—
Nachopti	243	19742	—	3.324	—
FFD	< 1	19742	3.418	3.324	8.94
MFFD	< 1	19742	3.414	3.324	8.83

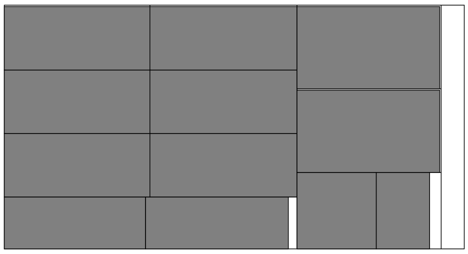
Tabelle 8: Ergebnisse für Instanz Honsel_3

komplett genutzten Rohlingslängen einen Beitrag zum prozentualen Verschnitt liefern. Im Rohling 1 beispielsweise liegt ein prozentualer Verschnitt von 6.34 Prozent vor. Es wird eine Rohlingslänge von 5644 mm genutzt. Das bedeutet, daß 296 mm der Länge auf ihrer gesamten Breite ungenutzt bleiben. Diese Rohlingsfläche ($296 \cdot 3150$ mm^2) stellt schon knapp fünf Prozent der gesamten Rohlingsfläche dar und trägt somit stark zum Verschnitt bei. Ähnliches gilt auch für die anderen Rohlinge. Der Verschnitt beruht somit nicht allein auf schlecht gewählten Paarungen durch das iterierte Matching, sondern auch auf einer ungünstigen Abschnittsverteilung.

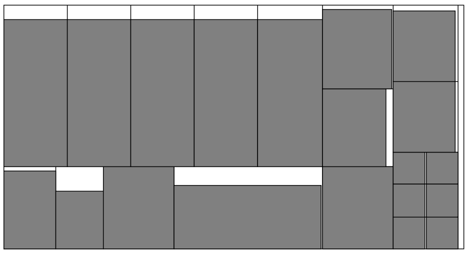
Verschnitte und genutzte Längen für Honsel_3					
Rohling Nr.	Verschnitt in Prozent	genutzte Länge	Rohling Nr.	Verschnitt in Prozent	genutzte Länge
1	6.34	5644	3	10.33	5764
2	9.97	5875	4	8.48 (62.11)	2459

Tabelle 9: Erzeugte Rohlinge für Instanz Honsel_3

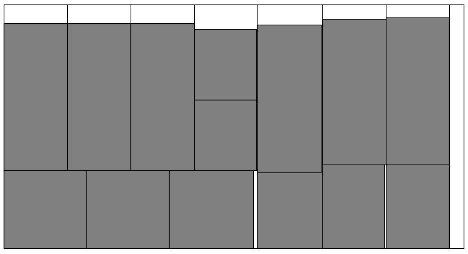
Dies bestätigt sich bei einer Betrachtung der Schnittmuster in Abbildung 27.



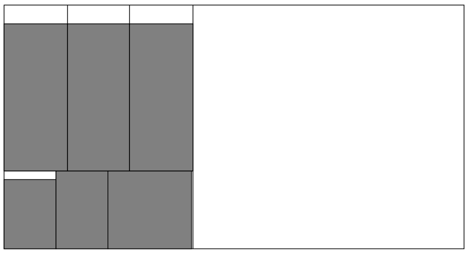
Schnittmuster Rohling 1; proz. Verschnitt: 6.34



Schnittmuster Rohling 2; proz. Verschnitt: 9.97



Schnittmuster Rohling 3; proz. Verschnitt: 10.33



Schnittmuster Rlg. 4; proz. Verschnitt: 8.48 (62.11)

Abbildung 27: Schnittmuster der Instanz Honsel_3

6.1.4 Instanz Honsel_4

Honsel_4							
Rohlingsdaten	:	$L = 5960$ mm	$B = 3170$ mm				
Scheiben	:	$N = 131$ Stück	24 Arten				
Gesamtscheibenfläche	:	100206712 mm^2					
Art	Anzahl	Länge	Breite	Art	Anzahl	Länge	Breite
1	12	260	260	13	6	775	1920
2	4	373	1345	14	12	775	838
3	1	455	2150	15	6	790	1148
4	4	490	490	16	16	800	1088
5	4	552	1988	17	2	920	540
6	4	554	1400	18	12	930	590
7	8	564	670	19	2	1010	1740
8	2	575	575	20	4	1120	748
9	4	667	1400	21	8	1148	710
10	4	685	1988	22	4	1148	654
11	2	700	2095	23	2	1228	1125
12	4	748	1345	24	4	1820	730

Tabelle 10: Daten der Instanz Honsel_4

Honsel_4 stellt eine Instanz mit 131 herzustellenden Scheiben dar. Für diese Instanz liegt wieder die Rohlingsgröße von $L = 5960$ mm und $B = 3170$ mm vor. Die weiteren Daten finden sich in Tabelle 10.

Die Zwischenergebnisse des Programmablaufes für Instanz Honsel_4 sind in Tabelle 11 dargestellt.

Verschnittoptimierung für Honsel_4					
Schritt	Zeit in Sekunden	GL in mm	reelle RZ	Mindest– RZ	MPV
Matching	62	33024	—	5.541	—
Nachopti	28	32784	—	5.501	—
FFD	< 1	32784	5.607	5.501	5.40
MFFD	1	32784	5.505	5.501	3.65

Tabelle 11: Ergebnisse für Instanz Honsel_4

Um die Güte der Paarungen zu ermitteln, die das iterierte Matching erzeugt, betrachte man den idealisierten Fall der Scheibenaufteilung auf die Rohlinge. Dazu gehe man — ähnlich wie bei der idealisierten Abschnittsverteilung — davon aus, daß die Flächen der ersten Rohlinge komplett genutzt werden und nur die Fläche des letzten angebrochenen Rohlings nicht ganz in der Länge, aber doch komplett in der Breite genutzt wird. In diesem Fall ließe sich die idealisierte Gesamtlänge berechnen, indem man die Gesamtscheibenfläche durch die Rohlingsbreite dividiert. Daß diese idealisierte Gesamtlänge so gut wie nie eingehalten werden kann, ohne die bestellten Scheiben zu zerstückeln, ist offensichtlich. Vergleicht man jedoch einmal die nach den Phasen 1 und 2 (Matchingphase) vorliegende Gesamtlänge (33024 mm) mit der Idealgesamtlänge (31611 mm), so stellt man fest, daß die Differenz nur circa 1.41 Meter beträgt. Dies deutet auf eine gute Paarung der Scheiben hin.

Der Tabelle 11 läßt sich entnehmen, daß die durch das MFFD erreichte Rohlingszahl nur unwesentlich von der Mindestrohlingszahl abweicht. Die tatsächliche Längenabweichung beträgt 23 Millimeter. Dies deutet darauf hin, daß die Längen der Rohlinge nahezu komplett genutzt sein müssen. Daß dem wirklich so ist, läßt sich anhand der Werte in Tabelle 12 feststellen. Der entstandene Verschnitt rührt somit tatsächlich aus den einzelnen Paarungen der Scheiben. Wie aus den obigen Überlegungen hervorgeht, ist dieser Verschnitt jedoch gering. Dies wird auch durch den kleinen mittleren prozentualen Verschnitt und anhand der Abbildung 28 deutlich. Bei einer Rohlingsgröße von $L = 5960$ mm und $B = 3170$ mm beträgt die Gesamtfläche eines Rohlings knapp 18.9 Quadratmeter. Ein Verschnitt von 3.65 Prozent in dem Rohling entspricht dann einer Fläche von nur knapp 0.69 Quadratmetern.

Verschnitte und genutzte Längen für Honsel_4					
Rohling Nr.	Verschnitt in Prozent	genutzte Länge	Rohling Nr.	Verschnitt in Prozent	genutzte Länge
1	2.28	5948	4	6.21	5960
2	3.01	5953	5	4.23	5956
3	2.91	5959	6	2.87 (50.98)	3008

Tabelle 12: Erzeugte Rohlinge für Instanz Honsel_4

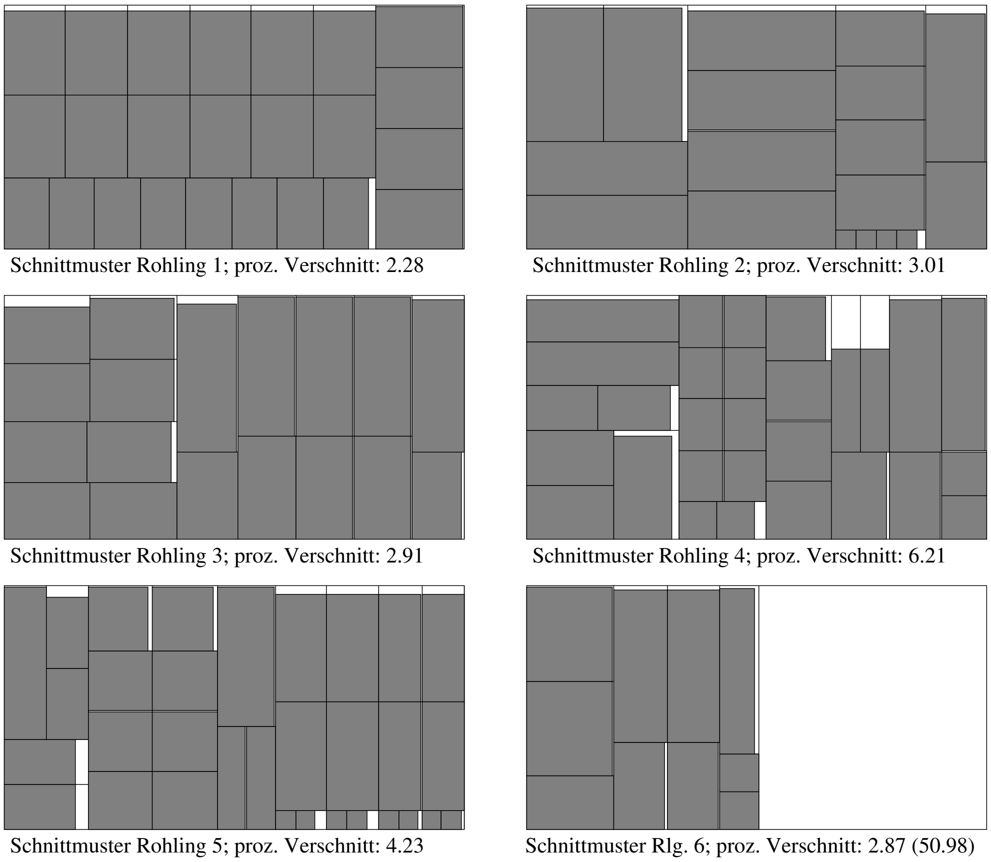


Abbildung 28: Schnittmuster der Instanz Honsel_4

6.1.5 Instanz Honsel_5

In Tabelle 13 sind die Daten der fünften Instanz angegeben.

Honsel_5							
Rohlingsdaten	:	$L = 5960$ mm	$B = 3170$ mm				
Scheiben	:	$N = 88$ Stück	34 Arten				
Gesamtscheibenfläche	:	76925672 mm^2					
Art	Anzahl	Länge	Breite	Art	Anzahl	Länge	Breite
1	2	414	1674	18	2	1136	966
2	4	414	864	19	2	1146	946
3	2	608	793	20	4	1150	817
4	1	660	440	21	4	1156	956
5	4	668	438	22	2	1200	1350
6	1	695	750	23	2	1244	1194
7	4	730	987	24	4	1280	982
8	4	750	997	25	4	1280	492
9	2	816	566	26	2	1280	840
10	2	836	956	27	2	1474	1104
11	4	844	414	28	2	1474	384
12	4	864	384	29	2	1533	893
13	4	865	840	30	2	1654	1104
14	2	976	1136	31	2	1654	414
15	2	997	750	32	2	1827	750
16	2	1094	1674	33	2	1840	740
17	2	1130	587	34	2	2120	560

Tabelle 13: Daten der Instanz Honsel_5

Wie auch während des Programmablaufs für die zweite Instanz tritt hier ebenfalls eine Verringerung der Gesamtlänge aller Abschnitte ein, wenn die abschnittrepräsentierenden Schnittbäume zu je einem Schnittbaum pro Rohling zusammengefaßt werden. Hierin läßt sich wieder der Vorteil in der Benutzung von Formfunktionen erkennen. Die Gestalten der Rechtecke bleiben bis zum Ende des Verfahrens variabel. Des weiteren lassen sich in der Ergebnistabelle 14 die typischen Merkmale des Matchingverfahrens erkennen, wie sie auch schon für die Instanzen Honsel_1 bis Honsel_4 vorlagen. Eine Verbesserung durch die Nachoptimierung ist vorhanden. Sie macht jedoch nicht den Hauptteil der Verschnittoptimierung aus. Sie verbessert lediglich die nach dem Matching vorliegende Gesamtlänge der Abschnitte um einige Zentimeter. Eine wichtige Aufgabe übernimmt das MFFD, da eine schlechte Aufteilung der

Abschnitte den geringen Verschnitt, der durch die Paarungen des iterierten Matching entstanden ist, wieder zunichte machen würde.

Verschnittoptimierung für Honsel_5					
Schritt	Zeit in Sekunden	GL in mm	reelle RZ	Mindest– RZ	MPV
Matching	30	25400	—	4.262	—
Nachopti	42	25185	—	4.226	—
FFD	< 1	25185	4.397	4.226	7.39
MFFD	< 1	25169	4.248	4.223	4.16

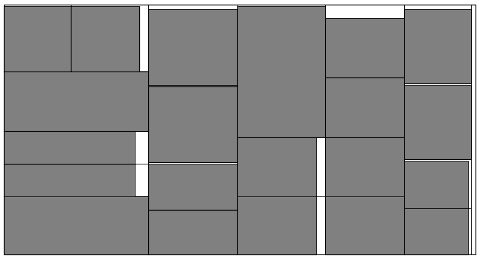
Tabelle 14: Ergebnisse für Instanz Honsel_5

Die einzelnen Rohlingsdaten sind in Tabelle 15 aufgeführt. Es werden 5 Rohlinge benötigt. Zu bemerken ist, daß die Rohlingslängen nicht komplett genutzt werden und dennoch ein niedriger mittlerer prozentualer Verschnitt vorliegt. Dies deutet auf eine dichte Plazierung der Scheiben bzw. auf Paarbildungen mit wenig Verschnitt hin.

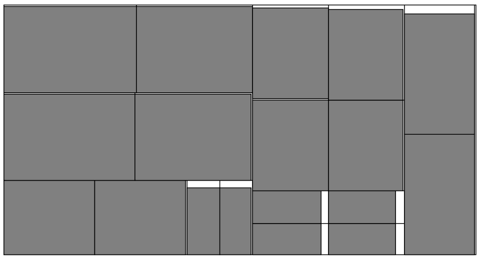
Verschnitte und genutzte Längen für Honsel_5					
Rohling Nr.	Verschnitt in Prozent	genutzte Länge	Rohling Nr.	Verschnitt in Prozent	genutzte Länge
1	4.82	5900	4	4.72	5956
2	2.78	5943	5	7.43 (77.01)	1480
3	3.51	5890			

Tabelle 15: Erzeugte Rohlinge für Instanz Honsel_5

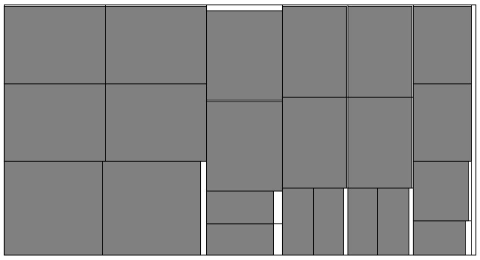
Abbildung 29 zeigt die einzelnen Schnittmuster der Rohlinge für diese Instanz.



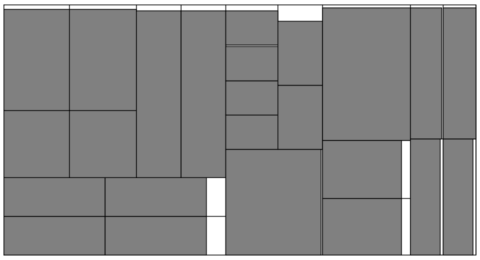
Schnittmuster Rohling 1; proz. Verschnitt: 4.82



Schnittmuster Rohling 2; proz. Verschnitt: 2.78



Schnittmuster Rohling 3; proz. Verschnitt: 3.51



Schnittmuster Rohling 4; proz. Verschnitt: 4.72

Schnittmuster Rohling 5; proz. Verschnitt: 7.43 (77.01)

Abbildung 29: Schnittmuster der Instanz Honsel_5

6.1.6 Instanz Honsel_6

In Tabelle 16 werden alle Werte der sechsten Instanz angegeben. Es sollen in dieser Instanz 161 Scheiben hergestellt werden. Es liegen dabei nur 18 verschiedene Scheibenarten vor. Dies kann in der Praxis vorkommen, wenn ein Auftrag vorliegt, bei dem eine große Anzahl gleichartiger Fenster hergestellt werden soll.

Honsel_6							
Rohlingsdaten	:	$L = 5960$ mm		$B = 3170$ mm			
Scheiben	:	$N = 161$ Stück		18 Arten			
Gesamtscheibenfläche	:	124828209 mm^2					
Art	Anzahl	Länge	Breite	Art	Anzahl	Länge	Breite
1	6	355	440	10	3	865	1145
2	6	480	1100	11	38	980	385
3	2	482	482	12	4	980	910
4	8	642	635	13	2	1000	780
5	2	685	850	14	6	1020	1100
6	6	745	1040	15	4	1130	1272
7	6	745	965	16	12	1145	865
8	4	823	1688	17	42	1183	865
9	6	835	1025	18	4	1777	900

Tabelle 16: Daten der Instanz Honsel_6

In der Tabelle 17 sind die Zwischenergebnisse aufgeführt, die bei der Anwendung der Verschnittoptimierung durch iteriertes Matching für die Instanz Honsel_6 berechnet werden.

Verschnittoptimierung für Honsel_6					
Schritt	Zeit in Sekunden	GL in mm	reelle RZ	Mindest–RZ	MPV
Matching	163	45363	—	7.611	—
Nachopti	1945	41043	—	6.886	—
FFD	< 1	41043	7.145	6.886	7.53
MFFD	1	41043	6.939	6.886	4.79

Tabelle 17: Ergebnisse für Instanz Honsel_6

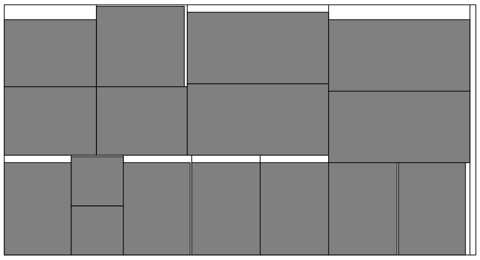
Die Zeit, die von der Matchingphase in Anspruch genommen wird, ist größer als bei den bisher betrachteten Instanzen. Man bedenke dabei aber, daß die Anzahl der herzustellenden Scheiben in dieser Instanz um 45 Stück, also fast vierzig Prozent, größer ist als beispielsweise in der Instanz Honsel_2. Auffällig ist die besonders große Zeitspanne, die für das Nachoptimieren benötigt wird. Der Grund für diese lange Zeitspanne liegt darin, daß einige der Rohlingsabschnitte eine große Zahl an Scheiben — teilweise mehr als 8 Scheiben — beinhalten. Dadurch liegt in der dritten Phase eine entsprechend große Anzahl zu vertauschender Teilbäume vor. Analysiert man dieses Problem genauer, so stellt man fest, daß sich die Gesamtlänge der Rohlingsabschnitte schon nach den ersten 40 Sekunden der Nachoptimierungsphase auf 41717 mm verbessert hat. In den restlichen 32 Minuten verringert sich die Gesamtlänge lediglich um 674 mm. An dieser Stelle wäre daher ein Zeitlimit für die Dauer der dritten Phase angebracht. Auf welche Art ein Abbruchkriterium für die dritte Phase festgelegt werden kann, wird unter anderem im nächsten Kapitel behandelt.

Die einzelnen prozentualen Verschnitte der Schnittmuster und die genutzten Längen der Rohlinge für die Instanz Honsel_6 sind in der folgenden Tabelle 18 aufgeführt.

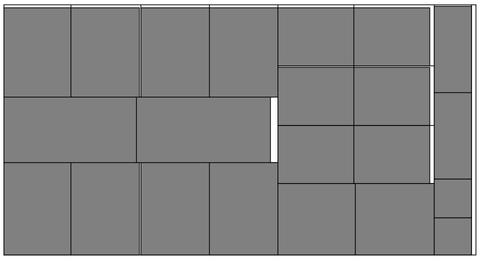
Verschnitte und genutzte Längen für Honsel_6					
Rohling Nr.	Verschnitt in Prozent	genutzte Länge	Rohling Nr.	Verschnitt in Prozent	genutzte Länge
1	7.31	5882	5	4.59	5905
2	2.34	5902	6	2.59	5960
3	3.07	5862	7	10.06 (15.54)	5597
4	3.85	5935			

Tabelle 18: Erzeugte Rohlinge für Instanz Honsel_6

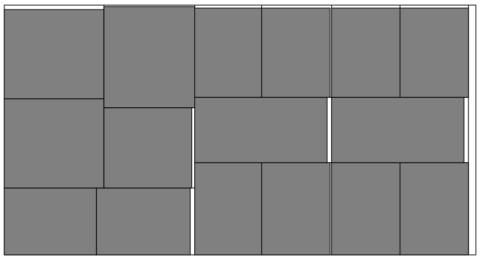
Die Schnittmuster lassen sich der Abbildung 30 entnehmen.



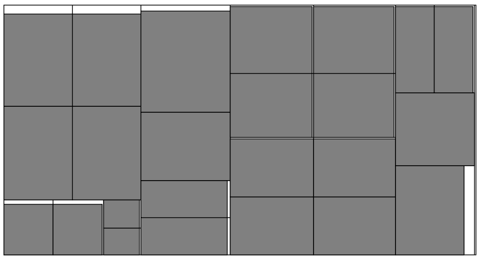
Schnittmuster Rohling 1; proz. Verschnitt: 7.31



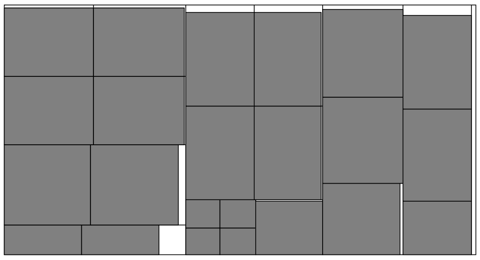
Schnittmuster Rohling 2; proz. Verschnitt: 2.34



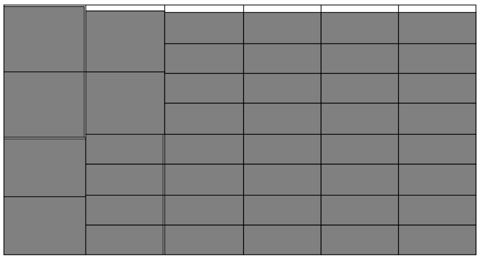
Schnittmuster Rohling 3; proz. Verschnitt: 3.07



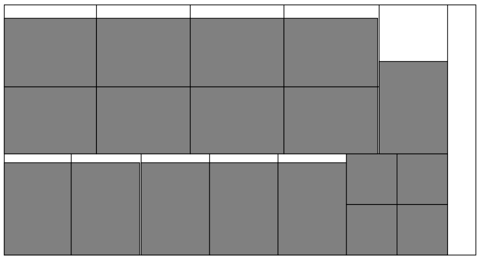
Schnittmuster Rohling 4; proz. Verschnitt: 3.85



Schnittmuster Rohling 5; proz. Verschnitt: 4.59



Schnittmuster Rohling 6; proz. Verschnitt: 2.59



Schnittmuster Rohling 7; proz. Verschnitt: 10.06 (15.54)

Abbildung 30: Schnittmuster der Instanz Honsel_6

6.1.7 Instanz Honsel_7

Die siebte Instanz, deren Werte in Tabelle 19 dargestellt sind, stellt die von der herzustellenden Scheibenfläche her gesehen größte Instanz dar. Die Anzahl der Scheiben beträgt 157 Stück, die Gesamtscheibenfläche circa 160 m^2. Im Gegensatz zu der Instanz Honsel_6 ist die Anzahl der verschiedenen Scheibenarten jedoch größer. Es sollen hier 51 verschiedene Scheibenarten hergestellt werden. Anhand der Ergebnistabelle 20 lassen sich wieder die typischen Eigenschaften der einzelnen Phasen erkennen. Das FFD würde in diesem Fall eine relativ gute Abschnittsverteilung liefern. Der MPV hätte bei Verwendung des FFD lediglich 5.45 Prozent betragen. Daß jedoch eine noch bessere Verteilung der Abschnitte existiert, läßt sich aufgrund der Differenz zwischen Mindestrohlingszahl und reeller Rohlingszahl bei Anwendung des FFD vermuten. Durch Anwenden des MFFD bestätigt sich diese Vermutung. Die Abschnitte werden erfolgreich auf die Rohlinge verteilt.

An dieser Stelle läßt sich erkennen, daß eine große Scheibenanzahl bzw. Gesamtscheibenfläche die Nachoptimierung in Phase 3 nicht notwendigerweise zum Scheitern bringt. Dieser Schluß hätte sich aus den Ergebnissen der sechsten Instanz ziehen lassen können. Es wird hier deutlich, daß selbst bei großer Scheibenzahl sehr gute Ergebnisse erzielt werden. Die Zeit, die das MFFD in Anspruch nimmt, ist in diesem Fall etwas höher als bisher. Dies liegt an der großen Anzahl der zu verplanenden Abschnitte. Eine genauere Analyse ergibt, daß nach der dritten Phase 48 Abschnitte vorliegen. Dadurch erhöht sich zwangsläufig die Zahl der möglichen Abschnittsvertauschungen, die zwischen den Rohlingen stattfinden können. Der benötigte Zeitraum für das MFFD stellt jedoch nur einen sehr geringen Teil der insgesamt benötigten Zeit dar.

Die Daten der durch das Matchingverfahren erzeugten Rohlinge sind in Tabelle 21 aufgelistet. Anhand der genutzten Längen erkennt man, daß sieben der neun Rohlinge komplett bzw. bis auf drei Millimeter in ihrer gesamten Länge genutzt werden. Der Verschnitt, der in den Rohlingen vorliegt, ist somit der tatsächlich durch die Paarungen des iterierten Matching entstandene Verschnitt. In Abbildung 31 zeigt sich einmal mehr, daß der Matchingansatz gute Lösungen für eine Verschnittoptimierung liefert.

Honsel_7							
Rohlingsdaten	:	$L = 5960$ mm		$B = 3170$ mm			
Scheiben	:	$N = 157$ Stück		51 Arten			
Gesamtscheibenfläche	:	159191327 mm^2					
Art	Anzahl	Länge	Breite	Art	Anzahl	Länge	Breite
1	2	277	1719	27	2	1017	1125
2	1	357	575	28	2	1017	582
3	4	374	670	29	2	1017	1487
4	2	376	646	30	2	1055	1170
5	2	466	987	31	8	1078	374
6	6	521	739	32	2	1136	1356
7	1	580	630	33	2	1136	896
8	2	693	1695	34	4	1136	866
9	2	721	2215	35	2	1136	716
10	2	724	1724	36	6	1196	876
11	2	726	1926	37	2	1219	1080
12	2	736	1926	38	10	1238	603
13	6	739	562	39	2	1277	985
14	2	746	1926	40	2	1346	1136
15	6	777	986	41	2	1695	658
16	8	787	737	42	2	1746	1136
17	4	870	957	43	2	1776	1136
18	2	877	777	44	2	1806	1136
19	2	885	1600	45	2	1891	818
20	8	886	1136	46	1	1982	1376
21	4	896	1196	47	2	1986	866
22	2	905	374	48	2	2370	885
23	4	940	1957	49	6	2370	795
24	2	987	974	50	2	2370	575
25	2	992	374	51	4	2370	523
26	2	1017	1232				

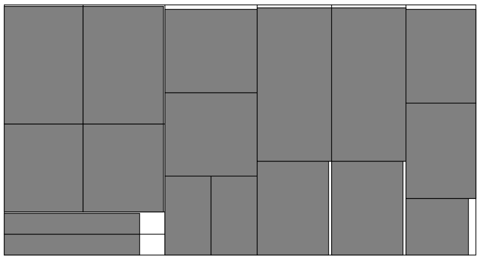
Tabelle 19: Daten der Instanz Honsel_7

Verschnittoptimierung für Honsel_7					
Schritt	Zeit in Sekunden	GL in mm	reelle RZ	Mindest– RZ	MPV
Matching	61	52051	—	8.733	—
Nachopti	40	51902	—	8.708	—
FFD	1	51902	8.911	8.708	5.45
MFFD	4	51902	8.714	8.708	3.31

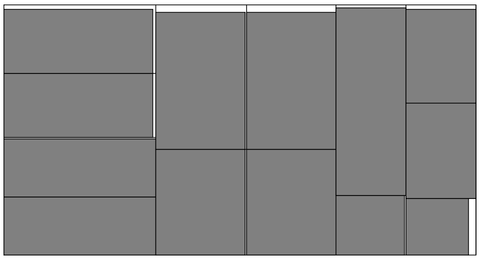
Tabelle 20: Ergebnisse für Instanz Honsel_7

Verschnitte und genutzte Längen für Honsel_7					
Rohling Nr.	Verschnitt in Prozent	genutzte Länge	Rohling Nr.	Verschnitt in Prozent	genutzte Länge
1	2.53	5960	6	6.17	5960
2	2.36	5959	7	2.46	5959
3	4.08	5959	8	2.43	5957
4	1.37	5960	9	2.56(30.41)	4257
5	5.61	5931			

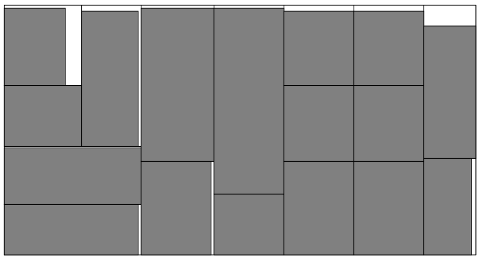
Tabelle 21: Erzeugte Rohlinge für Instanz Honsel_7



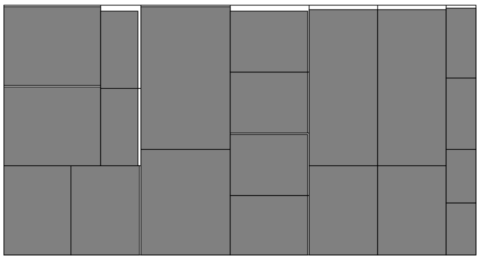
Schnittmuster Rohling 1; proz. Verschnitt: 2.53



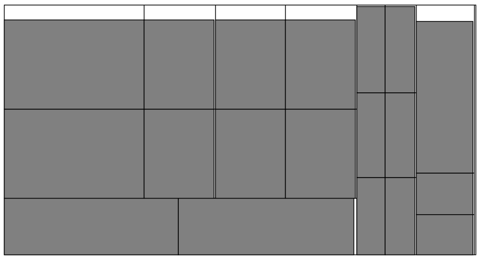
Schnittmuster Rohling 2; proz. Verschnitt: 2.36



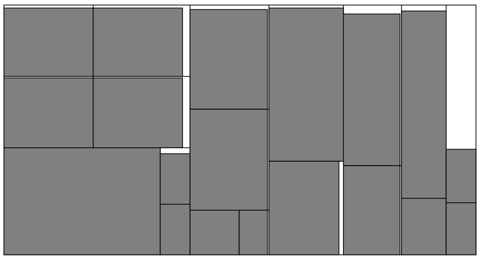
Schnittmuster Rohling 3; proz. Verschnitt: 4.08



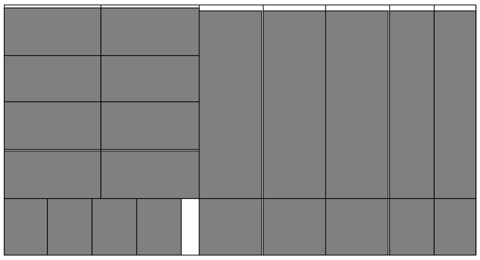
Schnittmuster Rohling 4; proz. Verschnitt: 1.37



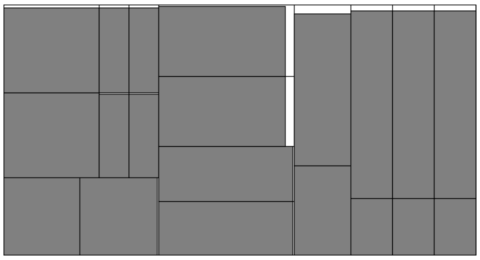
Schnittmuster Rohling 5; proz. Verschnitt: 5.61



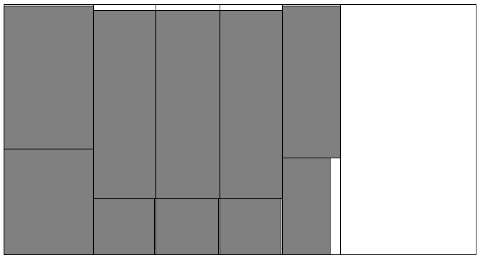
Schnittmuster Rohling 6; proz. Verschnitt: 6.17



Schnittmuster Rohling 7; proz. Verschnitt: 2.46



Schnittmuster Rohling 8; proz. Verschnitt: 2.43



Schnittmuster Rohling 9; proz. Verschnitt: 2.56 (30.41)

Abbildung 31: Schnittmuster der Instanz Honsel_7

6.1.8 Zusammenfassung der Matchingergebnisse

Bislang lassen sich die Ergebnisse der Testinstanzen folgendermaßen zusammenfassen: Die vierte Phase (MFFD) liefert für das Verteilen und Kombinieren der Rohlingsabschnitte Ergebnisse, die in dem überwiegenden Teil der Fälle nur unwesentlich von der idealisierten Lösung abweichen. Im Mittel beträgt die Abweichung der reellen Rohlingszahl von der Mindestrohlingszahl 0.031. Dieser Wert ist sehr abstrakt. Um ein verständlicheres Maß zu erhalten, soll die mittlere Längenabweichung der reellen von der idealisierten Lösung berechnet werden. Dazu muß jedoch beachtet werden, daß für die Instanz Honsel_3 eine andere Rohlingslänge als für die restlichen Instanzen vorliegt. Berechnet man $6 \cdot 0.031 \cdot 5960$ mm $+0.031 \cdot 5940$ mm und dividiert das Ergebnis durch 7, so erhält man eine mittlere Längenabweichung der reellen von der idealisierten Abschnittsverteilung von circa 18.5 Zentimetern.

Durch die dritte Phase wird die Gesamtlänge der Abschnitte in jedem der vorliegenden Fälle vermindert. Dabei stellt für die Instanz Honsel_6 die Nachoptimierung ein zeitliches Problem dar. Dieses ließe sich jedoch durch Einführung eines Zeitlimits für Phase 3 lösen. Man beachte, daß in Phase 3 für Honsel_6 jedoch auch von allen Nachoptimierungen die größte Längenverringerung stattfindet.

Die von der Matchingphase (Phase 1 und 2) gelieferten Rohlingsabschnitte besitzen eine hohe Ausbeute. Betrachtet man einmal alle Instanzen ausschließlich Honsel_6, so stellt man fest, daß die mittlere Verringerung der Gesamtlänge durch Phase 3 circa 56 Zentimeter beträgt. Die guten Endergebnisse — im Mittel über alle Instanzen beträgt der Verschnitt pro Rohling nur 4.84 Prozent — entstehen nicht allein durch die Nachoptimierung, sondern vielmehr durch die Güte der durch das iterierte Matching gebildeten Abschnitte. Daß die mit dem Matchingprogramm berechneten Ergebnisse mit denen eines in der Praxis erfolgreich eingesetzten Programms konkurrieren können, soll im folgenden Abschnitt durch einen Vergleich gezeigt werden.

6.2 Iteriertes Matching im Vergleich zu XOPT

Der Vergleich der Laufzeiten der beiden Programme gestaltet sich schwierig, da die Ergebnisse, wie schon erwähnt, auf zwei unterschiedlichen Rechnern erzeugt wurden. Nach Auskunft eines Mitarbeiters der Firma Honsel lagen die Laufzeiten von XOPT für jede Instanz unter drei Minuten. Aus den Tabellen des vorherigen Abschnitts läßt sich ermitteln, daß die Laufzeiten des Matchingprogramms bis auf zwei Fälle unter zwei Minuten liegen. Unter Berücksichtigung der Rechnerleistungen läßt sich somit auf eine geringfügig bessere Laufzeit des Programms XOPT schließen.

Zum Vergleich der berechneten Lösungen werden die Ergebnisse des Programms XOPT und die im vorherigen Kapitel dargestellten Ergebnisse in zwei Tabellen aufgeführt. Dabei sollen als Vergleichskriterien die Gesamtlänge der Rohlingsabschnitte, die Mindest– und reelle Rohlingszahl, der mittlere prozentuale Verschnitt und die Länge der abfallenden Restplatte dienen. Tabelle 22 zeigt die Ergebnisse des Matchingprogramms als Zusammenfassung der im vorherigen Abschnitt aufgeführten Werte.

Ergebnisse des Matchingverfahrens					
Instanz	Gesamtlänge in mm	Rohlingszahl mindest	Rohlingszahl reell	MPV	Restlänge in mm
Honsel_1	36343	6.098	6.103	4.29	5345
Honsel_2	43334	7.271	7.304	4.85	4146
Honsel_3	19742	3.324	3.414	8.83	3481
Honsel_4	32784	5.501	5.505	3.65	2952
Honsel_5	25169	4.223	4.248	4.16	4480
Honsel_6	41043	6.886	6.939	4.79	363
Honsel_7	51902	8.708	8.714	3.31	1703

Tabelle 22: Gesamtdarstellung der Ergebnisse des Matchingverfahrens

Die Ergebnisse des Programms XOPT laßen sich aus den von der Firma Honsel zusammen mit den Instanzdaten gelieferten Schnittmustern ermitteln. Sie lassen sich der Tabelle 23 entnehmen.

Ergebnisse des Programms XOPT					
Instanz	Gesamtlänge in mm	Rohlingszahl mindest	Rohlingszahl reell	MPV	Restlänge in mm
Honsel_1	36615	6.143	6.189	5.62	4834
Honsel_2	43353	7.274	7.291	4.67	4227
Honsel_3	20171	3.396	3.416	8.88	3471
Honsel_4	33022	5.541	5.547	4.39	2697
Honsel_5	25255	4.237	4.257	4.36	4426
Honsel_6	41106	6.897	6.923	4.57	457
Honsel_7	51832	8.697	8.719	3.36	1676

Tabelle 23: Gesamtdarstellung der Ergebnisse des Programms XOPT

Um den Vergleich übersichtlicher zu gestalten, werden in einer weiteren Tabelle die Differenzen der jeweiligen Werte angegeben. Die Ergebnisse von XOPT werden dazu von den Matchingergebnissen subtrahiert. Das bedeutet, daß ein negativer Wert in Tabelle 24 eine bessere Lösung des Matchingprogramms gegenüber der des Programms XOPT darstellt. Man beachte, daß für die Restplattenlänge genau der umgekehrte Fall gilt: Je größer der (positive) Wert, desto besser ist das Matchingergebnis gegenüber XOPT.

Differenzen: Matching – XOPT					
Instanz	Gesamtlänge in mm	Rohlingszahl		MPV	Restlänge in mm
		mindest	reell		
Honsel_1	−272	−0.045	−0.086	−1.33	511
Honsel_2	−19	−0.003	0.013	0.18	−81
Honsel_3	−429	−0.072	−0.002	−0.05	10
Honsel_4	−238	−0.040	−0.042	−0.74	255
Honsel_5	−86	−0.014	−0.009	−0.20	54
Honsel_6	−63	−0.011	0.016	0.22	−94
Honsel_7	70	0.011	−0.005	−0.05	27

Tabelle 24: Vergleich der Matchingergebnisse mit den XOPT–Ergebnissen

Zwei der von dem Matchingverfahren berechneten Lösungen sind geringfügig schlechter als die Lösungen von XOPT. Die anderen Lösungen sind besser als die von XOPT, davon zwei sogar wesentlich (Honsel_1 und Honsel_4). Auffällig ist, daß die Gesamtlänge der Rohlingsabschnitte in den meisten Fällen in den Matchingergebnissen geringer ist als in den XOPT–Ergebnissen. Auch für die Instanzen Honsel_2 und Honsel_6, bei denen die Endergebnisse des Matchingprogramms schlechter als die XOPT–Ergebnisse sind, ist diese Gesamtlänge geringer. Dies läßt darauf schließen, daß entweder MFFD nicht effizient genug arbeitet oder aber, daß die erzeugten Rohlingsabschnitte in diesen Fällen ungünstige Maße besitzen, so daß sie nicht sinnvoll auf die Rohlinge verteilt werden können. In den Matchingergebnissen für die siebte Instanz liegt genau der umgekehrte Fall vor: Die Gesamtlänge ist schlechter, die reelle Rohlingszahl besser als bei XOPT. Für Instanz Honsel_3 liegen Schnittmuster vor, deren Verschnitt höher ist als in den Schnittmustern der anderen Instanzen. Aus der Vergleichstabelle läßt sich entnehmen, daß auch das Programm XOPT für diese Instanz keine bessere Lösung ermitteln konnte.

Die gesparte Rohlingsfläche, die sich bei der Benutzung des Matchingverfahrens ergibt, beträgt allein für diese sieben Instanzen circa 2.16 Quadratmeter.

7 Weiterführende Überlegungen

In diesem Kapitel werden einige weitergehende Aufgabenstellungen aufgeführt, die an das Matchingprogramm zusätzlich zur Verschnittminimierung gestellt werden könnten. Es werden einige Ideen angegeben, wie diese zusätzlichen Aufgaben beim Ablauf des Programms berücksichtigt werden könnten.

Wie in Kapitel 3 erläutert, liegen in der Praxis nicht immer Glastafeln vor, aus denen die Scheiben ausgeschnitten werden sollen. Bei dem zur Verfügung stehenden Material kann es sich auch um ein (theoretisch) endlos langes Glasband handeln. Soll das Matchingprogramm ein Schnittmuster für diese Materialgestalt berechnen, so entfällt lediglich die vierte Phase des Programms. Eine Aufteilung der Abschnitte ist nicht mehr notwendig, denn die Rohlingsabschnitte können direkt nacheinander von dem Glasband abgeschnitten werden. In diesem Fall würde dann die Güte der Verschnittoptimierung einzig von der Güte der iterativ gebildeten Paare abhängen.

Ebensowenig Probleme verursacht die Forderung, daß auch Rohlinge unterschiedlicher Länge als Material zur Verfügung stehen dürfen. Es muß in diesem Fall lediglich wieder die vierte Phase des Programms durch einen effektiven Algorithmus ersetzt werden, der diese Nebenbedingung beim Verteilen der Abschnitte berücksichtigt. Zu beachten ist dabei, daß die Breite aller Rohlinge gleich sein sollte und keine der vorliegenden Abschnittslängen größer als das Maximum der zur Verfügung stehenden Rohlingslängen ist. Das Problem entspricht dann dem in Kapitel 2 vorgestellten allgemeinen eindimensionalen Verschnittproblem 2.4. In demselben Kapitel wurde bereits ein effektiver Algorithmus beschrieben, den Gilmore und Gomory zur Lösung dieses Problems entwickelt haben. In kleinem Umfang wurde diese zusätzliche Forderung bereits implementiert. Das Programm bietet seinem Benutzer die Möglichkeit, die Restplatte, die bei der Erzeugung der Schnittmuster einer Instanz entstanden ist, bei der Optimierung der nächsten Instanz zu verwerten. Wichtig dabei ist, daß die Rohlinge der zweiten Instanz dieselbe Breite besitzen, wie die der ersten.

Die obigen zusätzlichen Aufgabenstellungen nahmen keinen oder nur geringen Einfluß auf das grundlegende iterierte Matching. Anders die nun folgenden möglichen Nebenbedingungen. Es wurde bislang davon ausgegangen, daß das verwendete Glas isotropisch ist. In der Praxis wird mitunter auch sogenanntes Strukturglas verarbeitet, dessen Oberfläche bestimmte Musterungen aufweist. Bei dieser Glasart ist darauf zu achten, daß die einzelnen Scheiben dieselbe Struktur erhalten. Sie dürfen daher beim Einplanen in die Rohlinge nicht rotiert werden. Soll diese Nebenbedingung berücksichtigt werden, so muß die Berechnung der Formfunktionen anders gestaltet werden. Es existieren für ein Metarechteck, das aus zwei Scheiben besteht, nicht mehr vier Gestalten pro Schnittorientierung, sondern lediglich eine. Es werden in den Formfunktionen nur noch die Sprungstellen aufgenommen, die sich durch horizontales bzw. vertikales Paaren der Scheiben mit festgelegter Ausrichtung ergeben. Die Berechnung dieser Formfunktionen gestaltet sich somit einfacher als in Kapitel 4 erläutert.

Aus Floatglas, dessen Herstellung in Kapitel 3 beschrieben wurde, werden in der Regel Fensterscheiben hergestellt. Dabei besteht ein Fenster zumeist aus zwei aufeinanderliegenden gleich großen Glasscheiben. Aus lagertechnischen Gründen kann es sinnvoll sein, die zueinander gehörigen Scheiben direkt nebeneinander in einem Rohling zu plazieren. Um dies zu erreichen, kann vor Beginn der Phase 1 eine zusätzliche Gruppierungsphase durchgeführt werden, in der die Schnittbäume der zueinander gehörigen Scheiben bereits kombiniert werden. Um eine mögliche Ausrichtungsvarianz der Scheiben zu erhalten, werden dabei wie üblich die Formfunktionen der Scheibenpaare berechnet. Die Gruppierungsphase entspricht somit genau einem Iterationsschritt, in dem die Kanten des Matchinggraphen, die zwei zueinander gehörige Scheiben verbinden, mit eins und die anderen Kanten mit null bewertet werden. Zu beachten ist, daß diese gebildeten Paare bei der Nachoptimierung oder beim Auflösen der Rohlingsabschnitte nicht wieder getrennt werden.

Ein ähnliches Problem stellt die Einhaltung der einzelnen Auftragszugehörigkeiten dar. Wie schon erwähnt, ist es in der Praxis üblich, Bestellungen zu sammeln und zu einer Instanz zusammenzufassen. Aus denselben Gründen wie oben ist es sinnvoll, die Scheiben eines Auftrages direkt nacheinander herzustellen. Dieses Problem läßt sich mit Hilfe des Matchingprogramms lösen, indem die Scheiben mit einer Auftragsnummer versehen und in jedem Iterationsschritt die Paarungen zweier Scheiben mit unterschiedlicher Auftragsnummer mit Null bewertet werden. Nach Beendigung der zweiten Phase liegen dann Abschnitte vor, in denen jeweils nur Scheiben eines Auftrages enthalten sind. Beim Verteilen der Abschnitte müssen dann ebenfalls Abschnitte gleicher Auftragsnummer nacheinander aus den Rohlingen ausgeschnitten werden. Diese Nebenbedingung entspricht einer der von Dyson und Gregory vorgestellten Bedingungen, die bei Verschnittoptimierungen in der Glasindustrie berücksichtigt werden sollten [Dys 1974]. Als weiteren Punkt führten die beiden Autoren eine Art "Dringlichkeit der Herstellung" ein. Scheiben, die dringend benötigt werden, sollen möglichst zu Beginn des Zuschnitts hergestellt werden. Dyson und Gregory faßten diese beiden Punkte — geordnete Aufträge und dringende Herstellung — unter dem Stichwort "pattern allocation" zusammen. Beim "pattern allocation"–Problem muß die Reihenfolge festgelegt werden, in der die Scheiben hergestellt werden sollen. Das Matchingverfahren könnte dieses Problem berücksichtigen, indem die Scheiben mit Dringlichkeitswerten versehen würden und die Kanten des Matchinggraphen nicht nur den Verschnitt einer Paarung bewerten, sondern auch deren Dringlichkeit angeben. Die Abschnitte — deren Dringlichkeitswert sich aus den Werten der enthaltenen Scheiben ergäbe — müßten dann den Werten nach absteigend aus den Rohlingen ausgeschnitten werden. Das "pattern allocation" Problem ist in der Praxis jedoch nicht von so großer Wichtigkeit wie die Verschnittoptimierung und wurde daher bislang nicht im Matchingprogramm implementiert.

Die in diesem Kapitel erläuterten zusätzlichen Nebenbedingungen können alle vom Matchingverfahren berücksichtigt werden. Zur Realisierung wären einige kleinere Änderungen des Programms notwendig, die im Rahmen dieser Arbeit jedoch nicht durchgeführt werden sollen.

Abschließend soll noch eine mögliche Änderung angegeben werden, die sich nicht aus einer zusätzlichen Anforderung ergibt, sondern die sich auf die Nachoptimierung der Abschnitte in Phase 3 bezieht. Wie sich aus der Ergebnistabelle 17 der Instanz Honsel_6 entnehmen läßt, kann die Laufzeit der Phase 3 sehr groß werden. Es wäre daher sinnvoll, ein künstliches Abbruchkriterium für die Nachoptimierungsphase einzuführen. Dies könnte beispielsweise eine fest vorgegebene Zeitspanne sein, die das Nachoptimieren höchstens in Anspruch nehmen darf. Wird jedoch während der gesamten Phase 3 eine gute Verringerung der Gesamtlänge erzielt, so wäre es nicht sinnvoll, diese Phase vorzeitig abzubrechen. Ein anderes Abbruchkriterium wäre daher, die Nachoptimierung zu beenden, wenn innerhalb einer gewissen Zeit kein Teilbaumtausch gefunden werden konnte, der die Gesamtlänge aller Abschnitte schrumpfen läßt. Ein zeitunabhängiges Abbruchkriterium ließe sich verwirklichen, indem die Verringerung genauer untersucht wird, die sich durch einen Teilbaumtausch ergibt. Liegen diese Verringerungen unter einem gewissen Grenzwert, so lassen sich anscheinend keine guten Verbesserungen mehr erzielen und die Nachoptimierungsphase kann beendet werden. Da selbst die längste Laufzeit bisher nicht mehr als 35 Minuten betrug und dies in der Praxis eine durchaus noch annehmbare Zeitspanne darstellt, wurde bisher noch kein künstliches Abbruchkriterium für Phase 3 im Matchingprogramm implementiert.

8 Zusammenfassung

Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, mit dem zweidimensionale Verschnittprobleme gelöst werden können. Dabei soll der entstehende Verschnitt minimiert und Nebenbedingungen berücksichtigt werden, wie sie sich für dieses Problem in der Praxis stellen. Insbesondere soll auf die Aufgabenstellungen in der Glasindustrie eingegangen werden.

Um einen Überblick über bisherige Lösungsansätze zu erhalten und somit die Besonderheiten des hier vorgestellten Verfahrens deutlich machen zu können, wurden zunächst einige typische Problemformulierungen vorgestellt und die bekannten Algorithmen zu deren Lösung erläutert. Um die Bedingungen in der Praxis zu berücksichtigen, wurde speziell auf die Herstellung und Verarbeitung von Glas eingegangen. Als Problemformulierung resultierte daraus das Glaszuschnitt–Problem 4.1. Aus großen Glastafeln sollen kleine Scheiben ausgeschnitten werden, wobei nur Guillotineschnitte zulässig sind. Das Hauptziel besteht in der Verschnittminimierung.

Es sollen nun zusammenfassend die grundlegenden Ideen des hier entwickelten Verfahrens aufgezeigt werden:

Die herzustellenden Scheiben stellen eine Menge von Rechtecken mit bestimmten Ausdehnungen dar. Die Rechtecke werden iterativ gepaart. Das aus einer Paarung resultierende Metarechteck wird als neues Rechteck angesehen, welches im nächsten Iterationsschritt erneut gepaart werden kann. Durch das iterative Paaren entstehen Schnittmuster mit Guillotineschnitt–Struktur. Jedes Metarechteck beinhaltet einen gewissen Verschnitt, der sich beim Zusammensetzen der Rechtecke ergeben hat. Die Ausrichtung der Scheiben ist während des gesamten Verfahrens noch nicht festgelegt. Um alle sinnvollen Ausrichtungen zu berücksichtigen, werden die Formfunktionen der einzelnen Rechtecke bzw. Metarechtecke berechnet. Anhand der Sprungstellen der Formfunktion läßt sich ablesen, wie ein Rechteck zerschnitten werden muß und welcher Verschnitt sich aus dieser Art des Zerschneidens ergibt. Die Paarungen eines Iterationsschrittes werden mit Hilfe des Maximum–Weight–Matching bestimmt. Dazu werden die in einem Iterationsschritt vorliegenden Rechtecke als Knoten in einem ungerichteten gewichteten vollständigen Graphen interpretiert. Die Kanten des Graphen geben die möglichen Paarungen der Rechtecke an und werden mit einem Benefitwert versehen. Der Benefitwert gibt die Güte einer Paarung an und berechnet sich aus den Benefitfunktionen. Die Benefitfunktionen berücksichtigen die Verschnitte und Ausdehnungen des aus einer Paarung resultierenden Rechtecks, indem die Formfunktion dieses Rechtecks analysiert wird. Das Maximum–Weight–Matching wählt die bezüglich der Benefitwerte optimalen Paarungen aus. Die einzelnen Paarungen werden durch eine binäre Baumstruktur repräsentiert. Die Blätter der Bäume entsprechen den herzustellenden Scheiben. Werden zwei Rechtecke gepaart, so werden die Bäume, die diese Rechtecke repräsentieren, durch einen zusätzlichen Knoten kombiniert, wodurch ein neuer Baum entsteht. Zwei gepaarte Rechtecke besitzen somit einen gemeinsamen Vaterknoten. Die Formfunktion des entstandenen Rechtecks wird in der Wurzel des neu entstandenen Baumes gespeichert. Da die Bewertung der Paarungen mit zunehmender Größe der Bäume kompliziert wird, wird das Verfahren in vier Phasen unterteilt. Zunächst werden die Scheiben zu Rohlingsabschnitten gepaart. Dabei entspricht ein Rohlingsabschnitt der Gestalt eines Metarechtecks, die die Rohlingsbreite so gut wie möglich und die Rohlingslänge so wenig wie nötig beansprucht. Jeder Rohlingsabschnitt besitzt eine möglichst hohe Ausbeute, was einem möglichst geringen Verschnitt entspricht. Jeder Abschnitt ist eindeutig durch einen Baum repräsentiert. Durch Vertauschen gewisser Teilbäume werden die Abschnitte nachoptimiert. Das heißt, die Scheiben in den Abschnitten werden so umsortiert, daß sich die Gesamtlänge aller Abschnitte verringert. Abschließend werden die Rohlingsabschnitte den einzelnen Rohlingen zugeordnet, aus denen sie ausgeschnitten werden sollen. Grundlage dieser Zuordnung bildet der First–Fit–Decreasing Algorithmus. Da dieser jedoch keine befriedigenden Ergebnisse liefert, wird durch Vertauschen der Abschnitte zwischen den Rohlingen eine Verteilung gesucht, bei der in möglichst vielen Rohlingen die Länge komplett genutzt wird. Dadurch wird erreicht, daß von dem Rohling, dessen Länge am wenigsten genutzt wird, eine möglichst lange Restplatte abfällt. Diese kann dann in einer anderen Optimierung weiterverwertet werden. Am Ende der vierten Phase liegt eine Menge von Schnittbäumen vor, von denen jeder ein (oder mehrere) Schnittmuster für einen Rohling widerspiegelt. Die Schnittmuster lassen sich durch eine Top–Down–Traversierung der Bäume realisieren. Man sucht dazu in der Formfunktion der Wurzel eines Baumes die erste Sprungstelle mit y–Wert kleiner oder gleich der Rohlingsbreite. Die Schnittorientierung und die Schnittposition geben an, wie der Rohling in zwei Teile zerlegt wird. Diese Zerlegung entspricht einem Guillotineschnitt. Die weitere Zerlegung der entstandenen Teile läßt sich aus den Formfunktionen in den Söhnen der Wurzel ermitteln. Die Zerlegung wird so lange fortgesetzt, bis die bestellten Scheiben vorliegen. Die Daten des Schnittmusters eines Rohlings (Schnitt– und Plazierungskoordinaten) werden jeweils in einer Datei abgespeichert. Die Dateien sind dabei so allgemein gehalten, daß sie durch Konvertierungsprogramme einfach in ein anderes Format umgewandelt werden können.

Das Matchingverfahren wurde als C–Programm implementiert. Anhand von sieben Testinstanzen, bei denen es sich um echte Bestellungen von Scheiben aus einem glasverarbeitenden Betrieb handelt, wurde das Matchingprogramm getestet. Als Rechner stand dabei ein mit 33 MHz getakteter Computer mit 486er Prozessor zu Verfügung. Anhand der berechneten Lösungen wurden die Qualität des iterierten Matching und die Auswirkungen der Nachoptimierung untersucht. Als Bewertungsgrundlage für die Güte der berechneten Schnittmuster diente die reelle Rohlingszahl (Definition 4.1). Die Ergebnisse des vierphasigen Matchingprogramms für die sieben Instanzen wurden mit denen eines kommerziellen Programms verglichen. Als Fazit der Testergebnisse läßt sich folgendes angeben:

Das Matchingprogramm bildet innerhalb kurzer Zeit (für die sieben Testinstanzen durchschnittlich 60 Sekunden) eine Menge von Rohlingsabschnitten. Die Ausbeuten dieser Abschnitte betragen in der Regel über 95 Prozent. Das Nachoptimieren gemäß Phase 3 bewirkt (abgesehen von der Instanz Honsel_6) im Mittel eine Verringerung der Gesamtlänge der Abschnitte um circa 56 Zentimeter. Die hierfür benötigte Zeit liegt in fünf der sieben Fälle unter 43 Sekunden. Durch Ermitteln der Lösungen, die ohne eine Nachoptimierung entstehen, zeigte sich, daß aufgrund des gierigen Verhaltens des Matchingalgorithmus eine Nachoptimierung sinnvoll ist. Da in ungünstigen Fällen die Laufzeit für die Nachoptimierung extrem hoch sein kann, ist die Einführung eines künstlichen Abbruchkriteriums für Phase 3 sinnvoll. Die Rohlingsabschnitte werden mit Hilfe des MFFD–Verfahrens auf die Rohlinge verteilt. Aus den Ergebnissen dieser Verteilung läßt sich die reelle Rohlingszahl berechnen. Die Mindestrohlingszahl spiegelt den idealisierten Fall der Verteilung wider, bei dem alle Rohlinge bis auf einen in ihrer ganzen Länge genutzt werden. Die Abweichung der reellen von der Mindestrohlingszahl beträgt für die sieben Instanzen im Mittel 0.031. Dies entspricht einem mittleren Längenunterschied von knapp 18.5 Zentimetern zwischen der reellen und der idealisierten Verteilung der Abschnitte. Bei der Berechnung dieser Mittelwerte wurde die abweichende Rohlingslänge der Instanz Honsel_3 berücksichtigt. Die vom MFFD benötigte Zeit liegt — abgesehen von Instanz Honsel_7 — bei höchstens einer Sekunde. Der Vergleich zwischen den Lösungen des Matchingprogramms und denen des kommerziellen Programms ergab, daß die Matchingergebnisse besser sind als die anderen. Für die sieben Instanzen wurde eine Gesamtglasfläche von circa 2.16 m^2 durch das Matchingprogramm eingespart. Ein von dem Matchingprogramm berechneter Rohling beinhaltet im Mittel einen Verschnitt von nur 4.84 Prozent.

Das iterierte Matchingverfahren berechnet hochwertige Lösungen für zweidimensionale Verschnittprobleme, die in vielen Fällen sogar die Qualität der Lösungen eines kommerziellen Programms übertreffen.

Literatur

[Can 1979] P. de Cani, "Packing problems in theory and practice", Department of Engineering Production, University of Birmingham, März 1979

[Chv 1983] V. Chvátal, "Linear Programming", Freeman and Company, New York/San Francisco, 1983

[Cof 1980] E.G. Coffman, M.R. Garey, D.S. Johnson und R.E. Tarjan, "Performance Bounds for Lavel–Oriented Two–Dimensional Packing Algorithms", SIAM Journal on Computing, Vol. 9, November 1980, Seiten 808–826

[Coo 1971] S.A. Cook, "The complexity of theorem-proving procedures", Proc. 3rd Annual ACM Symposium on Theory of Computing, Association of Computing Machinery, New York 1971, Seiten 151-158

[Dys 1974] R.G. Dyson und A.S. Gregory, "The Cutting Stock Problem in the Flat Glas Industry", Operational Research Quaterly, Vol. 25, 1974, Seiten 41–53

[Eis 1957] K. Eisemann, "The Trim Problem", Management Science, Vol. 3, Nr. 3, April 1957, Seiten 279–284

[Gab 1973] H. Gabow, "Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs", Ph.D. Thesis, Stanford University, 1973

[Gar 1979] M.R. Garey und D.S. Johnson, "Computers and Intractability", Freeman, San Francisco, 1979

[Gil 1961] P.C. Gilmore und R.E. Gomory, "A Linear Programming Approach to the Cutting-Stock Problem", Operations Research, Vol. 9, 1961, Seiten 849–859

[Gil 1965] P.C. Gilmore und R.E. Gomory, "Multistage cutting stock problems of two and more dimensions", Operations Research, Vol. 13, 1965, Seiten 94–120

[Ill 1983] H.J. Illig, "ABC Glas", Autorenkollektiv, VEB Deutscher Verlag für Grundstoffindustrie, Leipzig, 1983

[Kak 1976] Y. Kakazu, M. Furukawa und N. Okino, "The Optimum Trimming of Rectangular Plates", Bulletin of the Japan Society of Precision Engineering, Vol. 9, Nr. 5, März 1976, Seiten 127–132

[Kan 1939] L.V. Kantorovich, "Mathematical Methods of Organizing and Planning Production", veröffentlicht an der Staatlichen Universität zu Leningrad 1939, nachträglich in Management Science, Vol. 6, 1960, Seiten 366–422

[Kop 1991] H. Kopka, "LaTeX— Eine Einführung", Addison Wesely, 1991

[Mue 1991] R. Müller, "Hierarchisches Floorplaning mit integrierter globaler Verdrahtung", Bericht Nr. 81 im Fachbereich Mathematik–Informatik der Universität–Gesamthochschule–Paderborn, Februar 1991

[Ott 1983] R.H.J.M. Otten, "Efficient Floorplan Optimization", In Proceedings of the International Conference on Computer Design: VLSI in Computers, IEEE, 1983, Seiten 499–502

[Pap 1982] C.H. Papadimitiou und K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity", Prentice–Hall Inc., New Jersey, 1982, Seiten 247–266

[Saa 1975] T.L. Saaty und J.M. Alexander , "Optimization and the Geometry of Numbers: Packing and Covering", SIAM Review, Band 17, Nr. 3, Juli 1975, Seiten 475–519

[Sto 1983] L. Stockmeyer, "Optimal Orientations of Cells in Slicing Floorplan Design", Information and Control, 1983, Nr. 57, Seiten 91–101

[Was 1975] erschienen in Data Systems , Teil 16 , Mai 1975, Seiten 24–25

[Whi 1977] C. Whitlock und N. Christofides, "An Algorithm for Two–Dimensional Cutting Problems", Operations Research, Vol. 25, Nr. 1, Januar–Februar 1977, Seiten 30–44

Danksagung

An dieser Stelle möchte ich Herrn Professor Dr. Peter Brucker und Herrn Professor Dr. Oliver Vornberger danken, die mich während dieser Diplomarbeit betreut haben. Ebenfalls danke ich Herrn Wägener von der Firma Glas Honsel KG, der mir einen Einblick in die Glasverarbeitung gab und die Instanzen zum Test des entwickelten Programms zur Verfügung stellte.

Erklärung

Ich versichere, daß ich diese Arbeit selbständig angefertigt habe und keine außer den genannten Hilfsmitteln und Quellen verwendet habe. Den Quellen wörtlich oder sinngemäß entnommene Stellen sind als solche kenntlich gemacht.

Georgsmarienhütte, den 04.01.1994

Andreas Fritsch