

Diss ETH Nr. 12316

Modulare Werkstattsteuerung einer Fluidzellpresse

Lösung eines sequentiellen 2D Bin Packing-Problems

Abhandlung zur Erlangung des Titels

Doktor der Technischen Wissenschaften

der

Eidgenössischen Technischen Hochschule Zürich

vorgelegt von

Beat Friedrich Steiner

Dipl. Betr.- und Prod.-Ing. ETH

geboren am 28. 4. 1966

von Schlossrued (AG)

Angenommen auf Antrag von

Prof. Dr. J. Reissner, Referent

Prof. Dr. H.-J. Lüthi, Korreferent

1997

Leer - Vide - Empty

Danksagungen

Die vorliegende Arbeit und die darauf basierende Software ‚FluidPlan‘ entstand in den Jahren 1993 bis 1997 während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Umformtechnik der ETH Zürich.

Dem Leiter dieses Instituts, Herrn Prof. Dr. Reissner, danke ich an dieser Stelle herzlich für die Unterstützung bei der Entstehung der Arbeit. Herrn Prof. Dr. Lüthi danke ich für die freundliche Übernahme des Korreferates. Ganz besonders möchte ich Herrn Dr. Maurice Cochand vom Institut für Operations Research für die vielen Anregungen und die persönlichen Beiträge zum behandelten Thema sowie für die ausführliche Durchsicht der Arbeit erwähnen. Frau Dr. Arlette Gaillard von demselben Institut danke ich für die zu Beginn der Arbeit geführten Diskussionen, die mir halfen, den richtigen Weg einzuschlagen.

Weiter danke ich Herrn Flüeler, Herrn Blättler sowie Herrn Pulfer von den Flugzeugwerken Pilatus AG in Stans für die fruchtbare Zusammenarbeit vor und während der Entstehung von FluidPlan.

Am meisten Dank gebührt Herrn Roman Bargezi, der mir während der gesamten Phase der Implementation mit Rat und Tat beistand. Seine konstruktiven Vorschläge zur Gestaltung des Programmes sowie die tiefgehenden Diskussionen über die eingesetzten Algorithmen haben viel zum erfolgreichen industriellen Einsatz von FluidPlan beigetragen. Von den Freunden und Kollegen am Institut für Umformtechnik, die direkt oder indirekt zum Gelingen dieser Arbeit beigetragen haben, möchte ich namentlich die beiden Herren Jörg Danzberg und Adrian Beer erwähnen, welche die abschliessende Durchsicht der Arbeit übernommen haben.

Einen besonderen Dank möchte ich auch meinen Eltern aussprechen, die mir die Zeit an der ETH überhaupt ermöglichten, und last but not least meiner Freundin Lena Meili für das Verständnis und Vertrauen, das sie mir während der Entstehung dieser Arbeit entgegengebracht hat.

Zürich, im August 1997

Inhaltsverzeichnis

| | |
|--|-----------|
| Danksagungen | iii |
| Inhaltsverzeichnis | v |
| Zusammenfassung | ix |
| Abstract | xi |
| Index | xiii |
| 1 Einleitung | 1 |
| 1.1 Motivation | 1 |
| 1.2 Innovation | 2 |
| 1.3 Aufbau | 3 |
| | |
| Teil I: Werkstattsteuerung einer Fluidzellpresse | 5 |
| | |
| 2 System- und Aufgabenbeschreibung | 7 |
| 2.1 Das System und dessen Umgebung | 7 |
| 2.1.1 Das System als Blackbox | 8 |
| 2.1.2 Input und Output des Systems | 8 |
| 2.1.3 Steuer- und Zielgrößen des gesamten Systems | 10 |
| 2.2 Systembeschreibung | 10 |
| 2.2.1 Materialfluss im System | 10 |
| 2.2.2 Typische Operationspläne | 11 |
| 2.2.3 Systemkomponenten | 12 |
| 2.2.4 Entscheidungen und Auswirkungen | 15 |
| 2.3 Das Nestingproblem der Fluidzellpresse | 16 |
| 2.3.1 Übersicht der Beschickungsregeln | 17 |
| 2.3.2 Problembeschreibung | 17 |
| 2.4 Aufgabenbeschreibung | 19 |
| | |
| 3 Der Umformprozess in einer Fluidzellpresse | 23 |
| 3.1 Fluidzelltechnologie | 23 |
| 3.1.1 Einordnung des Verfahrens | 23 |
| 3.1.2 Wirkungsweise und Einsatzgebiet der Fluidzelltechnik | 24 |
| 3.2 Einflusskriterien | 25 |
| 3.2.1 Betriebliche Optimierungsziele | 26 |
| 3.2.2 Einflüsse aus der Planungsebene | 27 |
| 3.2.3 Beschickungsregeln und aktuelle Betriebsdaten | 28 |
| 3.2.4 Entscheidungen und Auswirkungen | 29 |
| 3.2.5 Kommentar und Übersicht | 29 |
| 3.3 Beschickungsregeln | 31 |
| 3.3.1 Allgemeine Regeln | 31 |
| 3.3.2 Spezifische Regeln: Tabuzonen | 34 |

| | | |
|--|---|-----------|
| 4 | Konzept für eine modulare Werkstattsteuerung..... | 37 |
| 4.1 | Modellbildung..... | 37 |
| 4.1.1 | Einführung und Abgrenzung einiger Begriffe..... | 37 |
| 4.1.2 | Definition der benötigten Funktionselemente..... | 39 |
| 4.1.3 | Das Modell von $S(0)$ | 40 |
| 4.2 | Was ist ein Werkstattsteuerungssystem..... | 42 |
| 4.2.1 | Definitionen rund um die werkstattnahe Steuerung..... | 42 |
| 4.2.2 | Geschichte und Trends der Werkstattsteuerung..... | 45 |
| 4.3 | Entwicklungsstufen des Werkstattsteuerungskonzepts für $S(0)$ | 46 |
| 4.3.1 | Stufe 1: Lokale Einsatzplanung der Fluidzellpresse..... | 46 |
| 4.3.2 | Stufe 2: Einbezug der Wärmbehandlung..... | 48 |
| 4.3.3 | Stufe 3: Reihenfolgeplanung des Gesamtsystems..... | 50 |
| 4.4 | Weitere Elemente des Konzepts..... | 56 |
| 4.4.1 | Zielkriterium des Gesamtsystems..... | 56 |
| 4.4.2 | Auftragsmodellierung und Zuweisung zu den Systemelementen..... | 58 |
| Teil II: Sequentielles 2D Bin Packing | | 61 |
| II.1 | Das sequentielle 2D Bin Packing..... | 61 |
| II.2 | Klassifizierung nach Dyckhoff..... | 64 |
| II.3 | Ansätze aus der Literatur..... | 65 |
| 5 | Bestimmung von Teilesets | 69 |
| 5.1 | Anforderungen und Zielfunktion des Moduls ‚Teileset‘..... | 69 |
| 5.1.1 | Input..... | 69 |
| 5.1.2 | Output..... | 70 |
| 5.1.3 | Zielfunktion..... | 71 |
| 5.2 | Bestimmung eines Teilesets..... | 72 |
| 5.2.1 | Grundidee des Moduls ‚Teileset‘..... | 72 |
| 5.2.2 | Der Enumerationsbaum..... | 73 |
| 5.2.3 | Ablauf der Teilesetwahl..... | 76 |
| 5.2.4 | Berechnung von $f_{HM(j)}$ | 79 |
| 5.3 | Bestimmung einer Folge von Teilesets..... | 80 |
| 5.3.1 | Zielfunktion einer Teilesetfolge..... | 81 |
| 5.3.2 | Direktes Plazieren..... | 81 |
| 5.3.3 | Vorausschauendes Plazieren..... | 83 |
| 6 | Layoutgenerierung..... | 87 |
| 6.1 | Ansatz für die Layoutgenerierung..... | 87 |
| 6.1.1 | k-Teilung eines Layouts..... | 88 |
| 6.1.2 | Einzonenplatz..... | 93 |
| 6.1.3 | Mehrzonenplatz..... | 94 |
| 6.2 | Formalisierung des Ansatzes mit Hilfe gerichteter Graphen..... | 97 |
| 6.2.1 | Grundbegriffe der Graphentheorie..... | 99 |
| 6.2.2 | Graphen als Mittel zur Darstellung von Layouts..... | 100 |
| 6.2.3 | Extremalkoordinaten einer Matrize und ihre Berechnung..... | 101 |
| 6.2.4 | Algorithmen zum Finden freier Plätze..... | 103 |

| | | |
|----------|---|------------|
| 6.3 | Ablauf der Layoutgenerierung | 105 |
| 6.4 | Bewertungsfunktionen | 106 |
| 6.4.1 | Bewertung und Wahl eines Platzes. | 107 |
| 6.4.2 | Bewertung eines Layouts | 111 |
| 6.4.3 | Bewertung einer Layoutfolge | 113 |
| 6.5 | Manipulationen an den Digraphen | 113 |
| 6.5.1 | Einfügen einer Matrize in einen Einzonenplatz | 113 |
| 6.5.2 | Einfügen einer Matrize in einen Mehrzonenplatz | 114 |
| 6.5.3 | Entfernen einer Matrize und Behandlung leerer Zonen | 115 |
| 6.6 | Behandlung von Tabuzonen. | 118 |
| 6.6.1 | Platzsuche für eine Matrize mit Tabuzone | 119 |
| 6.6.2 | Platzsuche neben Matrizen mit Tabuzonen | 120 |
| 6.7 | Konzept des Moduls Layoutoptimierung. | 120 |
| 6.7.1 | Ausgangslage | 120 |
| 6.7.2 | Ansätze zur Optimierung einer Layoutfolge | 122 |
| 7 | Tests | 125 |
| 7.1 | Simulation der rollenden Planung | 125 |
| 7.1.1 | Vorgehen. | 125 |
| 7.1.2 | Auswertungen | 126 |
| 7.2 | Test des Kernmoduls | 130 |
| 7.2.1 | Zeitbedarf und Komplexität | 130 |
| 7.2.2 | Gebrauch von Mehrzonenplätzen | 132 |
| 7.2.3 | Weitere Aussagen zum Kernalgorithmus | 132 |
| 7.3 | Datengenerator | 133 |
| | Anhang | 137 |
| A1 | FluidPlan für den Benutzer | 137 |
| A2 | Literaturverzeichnis | 143 |
| A3 | Abbildungsverzeichnis | 147 |
| A4 | Tabellenverzeichnis | 151 |
| A5 | Lebenslauf | 153 |

Zusammenfassung

Die vorliegende Arbeit liefert einen Beitrag zur Werkstattsteuerung einer Fertigungsstrasse, in welcher Blechteile für die Flugzeugindustrie hergestellt werden. Die am Anfang ebenen Blechteile durchlaufen dabei mindestens einmal die Operationen Wärmebehandlung, Umformung und Handbearbeitung, bis sie im umgeformten Endzustand sind. Zu Beginn wird dieser Fertigungsbereich - unser System - sorgfältig beschrieben, wobei insbesondere eine Analyse der Abhängigkeiten für die zur Umformung dienende Fluidzellpresse erfolgt.

Darauf aufbauend wird ein mehrstufiges Konzept für die Werkstattsteuerung des Systems entwickelt. Es werden dafür Ziele definiert und derart aufgliedert, dass sich die gesamte Aufgabe in übersichtliche Teilaufgaben segmentieren lässt. Diese sind teilweise so komplex, dass es zum heutigen Zeitpunkt kaum denkbar ist, alle zusammen in einem einzigen Schritt rechnerisch zu lösen. Die Gliederung in Teilaufgaben liefert deshalb die notwendige Basis für einen modularen Aufbau des Werkstattsteuerungssystems. Dabei wird pro Teilbereich der Fertigung, in dem eine der obengenannten Operationen durchgeführt wird, ein Modul zur Lösung der dabei auftretenden Teilaufgabe konzipiert.

Ein weiterer, übergeordneter Modul gibt zu Beginn einer Reihenfolgeplanung für alle Operationen eines Auftrages Zeitintervalle vor, in welchen sie durchgeführt werden sollen. Diese zur Koordination der einzelnen Module dienenden Intervalle werden nicht als Restriktionen aufgefasst, sondern fließen nur in die Zielfunktion ein, da die Bearbeitungsdauern an den Handarbeitsstationen eher stochastischer als deterministischer Natur sind. Dadurch kann eine deterministische Reihenfolgeplanung selten eingehalten werden. Überdies gelangen in kurzen Abständen ständig neue Aufträge ins System, die abhängig von ihrer terminlichen Priorität sobald wie möglich eingeplant werden müssen. Die Reihenfolgeplanung muss also im Sinne einer rollenden Planung stets der neu vorliegenden Auftragsstruktur angepasst werden. Im ersten Teil der Arbeit wird ein mehrstufiges, modulares Konzept dieser Werkstattsteuerung vorgestellt.

Die erste Stufe des Konzepts wurde implementiert: Sie umfasst den komplexesten Modul, welcher unter Zuhilfenahme neuer, innovativer Ansätze den Einsatz einer Fluidzellpresse plant. Pro umzuformender Artikel existiert ein Werkzeug mit spezifischem, rechteckigem Grundriss, welches ohne Fixation auf einen Metalltisch gelegt wird, der in die Presse hinein und wieder herausgefahren werden kann. In der Fluidzellpresse wird eine Gummimembrane einem derart hohen Druck ausgesetzt, dass die Blechzuschnitte an die Werkzeuge gepresst und dadurch zu Blechteilen umgeformt werden.

Der Tisch kann pro Pressung mit verschiedenen Werkzeugen bestückt werden. Da die Fertigungsaufträge im Allgemeinen unterschiedliche Losgrößen haben, resultieren auf dem Tisch ständig wechselnde Werkzeugkombinationen für die einzelnen Pressungen. Im implementierten Modul wurden Methoden entwickelt, die bestimmen, welche Aufträge miteinander gefertigt werden. Hauptziel dieser Aufgabe ist eine möglichst hohe Pressenauslastung, unter Berücksichtigung von terminlichen Prioritäten. Für das Layout der Werkzeuge auf dem Tisch werden ausserdem Anordnungen vorgeschlagen, die verschiedene Faktoren berücksichtigen. Insbesondere soll der Aufwand zur Realisierung dieser Anordnungen zwischen zwei Pressungen minimal sein. Die Methoden und Algorithmen dieses Moduls werden im zweiten Teil der vorliegenden Arbeit vorgestellt.

In der Einleitung zum zweiten Teil wird für die oben beschriebene spezielle Problematik, welche an einer Fluidzellpresse besteht, eine adäquate formale Beschreibung vorgestellt, die „sequentielles 2D Bin Packing“-Problem genannt wird. Beim klassischen 2D Bin Packing geht

es darum, eine gegebene Anzahl kleiner Rechtecke derart grossen Rechtecken zuzuweisen, dass sie darin überschneidungsfrei angeordnet werden können und dass die benötigte Anzahl grosser Rechtecke minimal ist. Beim sequentiellen 2D Bin Packing kann einem grossen Rechteck (einer Pressung) pro Auftrag maximal ein kleines Rechteck (ein Blechteil) zugewiesen werden, da pro Auftrag nur ein Werkzeug vorhanden ist. Überdies müssen sämtliche Pressungen, in welchen ein Auftrag jeweils vorkommt, in einer aneinanderhängenden Sequenz abgearbeitet werden, da das entsprechende Werkzeug nur einmal auf den Pressentisch und wieder herausgehoben werden soll. In der Praxis auftretende, mit dem Bin Packing verwandte Probleme werden oft als Packing- oder Nestingprobleme klassifiziert.

Das Finden von interessanten Auftragskombinationen, die miteinander an der Presse gefertigt werden sollen, ist der kombinatorische Teil des Problems. Er wird durch eine geschickte Teilenumeration aller möglichen Auftragskombinationen effizient gelöst, da schnell interessante Kombinationen gefunden werden. Deren Anzahl wird zudem durch dynamische Schranken, die sich an der besten bisher gefundenen Lösung orientieren, ständig reduziert.

Der geometrische Teil des Problems behandelt das Plazieren der entsprechenden Werkzeuge einer Auftragskombination. Die relativen Positionen der Matrizen zueinander bzw. zum Rand des Pressentisches werden mit Hilfe eines Ungleichungssystems festgehalten, welches durch ein Paar von gerichteten und gewichteten Graphen formalisiert wird. Diese auf den ersten Blick umständliche Methode zum Festhalten einer geometrischen Anordnung von Rechtecken hat zwei wichtige Vorteile: Erstens werden den Werkzeugen beim Lösen des Ungleichungssystems nicht feste Koordinaten, sondern Koordinatenintervalle zugewiesen; durch Berechnen der Grösse dieser Intervalle ist bekannt, an welcher Stelle wieviel Platz vorhanden ist, um weitere Werkzeuge zu plazieren. Zweitens bleiben von Pressung zu Pressung die relativen Positionen derjenigen Werkzeuge erhalten, welche nicht ausgewechselt werden. Damit kann der Aufwand zur Neupositionierung der Werkzeuge zwischen zwei Pressungen tief gehalten werden. Einige der Methoden zum Auffinden freier Plätze werden in dieser Arbeit erstmals vorgestellt, ebenso sämtliche Methoden, die benutzt werden, um das Paar gerichteter Graphen nach dem Einfügen bzw. Entfernen von Werkzeugen entsprechend nachzuführen.

Im Weiteren wird eine Strategie für die Erzeugung einer Folge von Pressungen präsentiert, bei der Werkzeuge mit grosser Grundfläche bevorzugt behandelt werden. Auslöser dafür sind Tests einer Strategie ohne derartige Präferenzen, bei denen festgestellt wurde, dass die meisten Lücken sofort von kleinen Werkzeugen gefüllt werden und grosse nicht oder nur sehr spät zum Zuge kommen. Die Resultate mit der neuen Strategie sind sehr gut.

Abschliessend sei erwähnt, dass das aus dieser Arbeit entstandene Softwareprodukt seit Mai 1996 erfolgreich in der Praxis eingesetzt wird.

Abstract

The present work provides a contribution to the workshop control of a manufacturing process in which sheet components are produced for the aero industry. In this process the initially flat sheet parts pass at least once through the operations heat treatment, forming and manual working, until they reach the final formed condition. To begin with this manufacturing process - our system - is described in detail, whereby in particular an analysis is made of the factors affecting the fluid cell press used for the forming process.

On this basis a multistage concept is developed for the workshop control of the system. For this purpose targets are defined and arranged in such a way that the overall task can be subdivided into manageable subtasks. In part these are so complex that at present it is practically inconceivable that they should all be mathematically solved in a single step. For this reason the division into subtasks provides the required basis for a modular set-up of the workshop control system. In this way, for each step in the manufacturing process, in which one of the above-mentioned operations is performed, a module is conceived for the solution of the corresponding subtask.

A further superimposed module provides at the beginning a sequence plan, for all operations of an order, time intervals in which they should be performed. These intervals serve to provide a co-ordination of the individual modules. They are not to be understood as a restriction, but provide only an input to the target function, since the durations of the processing steps at the manual working stations are more of a stochastic than a deterministic nature. In this way it is seldom possible to maintain a deterministic sequence plan. In addition there is the frequent arrival of new tasks in the system. Depending on their schedule priority, they must be included in the planning as soon as possible. Hence within the framework of a rolling planning system the system must be continuously matched to the current task order structure. In the first part of the work a multistage, modular concept of this workshop structure will be introduced.

The first stage of the concept has been implemented: It constitutes the most complex module, which plans, with the help of new innovative solutions, the application of a fluid cell press. For each object to be formed there exists a tool with a specific rectangular layout form. This is laid, without fixation, on a metal table, which can be moved into the press and out again. In the fluid cell press a rubber membrane is exposed to a sufficiently high pressure to press the sheet blanks against the tool and hence result in forming of the sheet parts.

For each pressing operation the press table can be equipped with various tools. Since the manufacturing orders generally have different numbers of parts, this results in a continuous change in tool combinations on the press table for the individual pressing operations. In the implemented module methods have been developed which determine which orders should be manufactured together. The main aim of this optimisation is to achieve the highest possible usage factor for the press, while taking account of the schedule priorities. In addition arrangements are proposed for the layout of the tools on the press table, whereby attention is paid to various factors. In particular the effort to achieve these arrangements between two pressing operations should be minimised. The methods and algorithms of this module will be presented in the second part of the work.

In the introduction to the second part a sufficient formal description is presented for the special situation for a fluid press, as described above. This is referred to as the "sequential 2D bin packing problem". For classical 2D bin packing the requirement is to allocate a given number of small rectangles to larger rectangles in such a way that there are no intersections and the required number of large rectangles is as small as possible. For sequential 2D bin packing for each

task a small rectangle (one sheet component) can not be allocated to more than one large rectangle (one pressing operation), since only one tool is present for each pressing operation. In addition all pressing operations in which a specific order appears, should be dealt with in sequence, since the corresponding tool should only be lifted onto and removed from the press table once. Problems which arise in practice, in connection with bin packing, are often classified as packing or nesting problems.

The identification of interesting order combinations, which should be manufactured together on the press, constitutes the combinatorial part of the problem. It is efficiently solved through clever numbering of the parts for all possible order combinations, since in this way it is quickly possible to find interesting combinations. In addition the number of combinations is continuously reduced through the use of dynamic limits based on the best solutions currently identified.

The geometrical part of the problem deals with the placing of the appropriate tool for an order combination. The relative positions of the dies to one another and to the edge of the press table are determined with the help of a set of inequalities. This is formalised by a pair of directed and weighted graphs. At first sight this might seem to be a complicated method for the determination of a geometrical arrangement of rectangles. However it has two important advantages: Firstly, when the set of inequalities is solved, the tools are not allocated to fixed co-ordinates, but rather to co-ordinate intervals; by calculating the size of these intervals, it is known at which points space is available, in order to position further tools. Secondly, from one pressing operation to the next the relative positions of tools which will not be exchanged remains the same. In this way it is possible to minimise the effort for repositioning tools between pressing operations. Some of the methods for finding free space are presented for the first time in this work. The same applies for all methods used to update the positions of the pair of directed graphs after the inclusion or removal of tools.

In addition a strategy is presented for the generation of a sequence of pressing operations, with which tools with a large surface area can be preferentially treated. The need for this was seen in tests without such a preference, in which it was noted that most spaces were immediately filled by small tools and that large tools were accommodated either very late or not at all. The results with the new strategy are very satisfactory.

Finally it should be mentioned that the software product produced during the course of this work has been successfully applied in practice since May 1996.

Index

| | |
|---|----------|
| Abgeleitetes Layout | 80 |
| Adjazenzmatrix | 88 |
| Agenten | 53 |
| aktiver Knoten | 74 |
| Anfangsknoten | 99 |
| äquivalente k-Teilungen | 89 |
| Arbeitsgang | 9 |
| Arbeitsplätze | 9 |
| Artikel | 1, 8 |
| Attribut | 38 |
| Auftrag | 1, 8, 40 |
| Auftragspapiere | 8 |
| Aushärtung | 14 |
| Baustein | 39 |
| BDE | 10 |
| Bellmannsche Gleichungen | 103 |
| Beschickungsregeln | 13, 31 |
| Bin Packing | 61, 65 |
| Blechstrasse | 7 |
| Bogen | 99 |
| Bogenlänge | 100 |
| C&P-Probleme | 64 |
| Chargierregeln | 14 |
| Cutting Stock | 65 |
| Digraph | 99 |
| dir | 126 |
| dist _j | 111 |
| Durchlaufzeit | 57 |
| dynamisches Funktionselement | 38 |
| Einzonenplatz | 93 |
| Element | 37 |
| Endknoten | 99 |
| Fertigungstermin | 9 |
| FluidPlan | 1, 7 |
| Fluidzelltechnik | 23 |
| freie Plätze | 91 |
| freie Zonen | 91 |
| Funktionselement | 38 |
| Gerichteter Graph | 99 |
| Graph | 99 |
| Guillotineschnitt | 93 |
| Gummimembrane | 12 |
| HandlingValue(L _i , L _{i+1}) | 121 |
| Informationsfluss | 42 |
| Job shop scheduling | 52 |

| | |
|-------------------------------|-----------|
| Kanten | 88, 99 |
| Knapsack | 65 |
| Knoten | 88, 99 |
| Komponente | 10 |
| k-Teilung | 88 |
| Längste Wege | 102 |
| Layout | 17 |
| Layoutfolge | 29, 80 |
| LayoutValue(TS _j) | 71, 111 |
| leere Zone | 90 |
| Leitstand | 43 |
| Lexikographische Ordnung | 73 |
| Losgröße | 9 |
| Lösungsglühen | 14 |
| Marke | 39 |
| Materialfluss | 42 |
| Matrize | 1, 12, 41 |
| MaxPos | 91 |
| MaxX, MaxY | 101 |
| Mehrzonenplatz | 94 |
| Membrane | 12 |
| Metalltröge | 13 |
| MinPos | 91 |
| MinX, MinY | 101 |
| Modul | 39 |
| MoveValue(TS _j) | 112 |
| Nachbarschaft | 17 |
| Nachfolger | 100 |
| Nesting | 16 |
| Obersystem | 7 |
| Operation | 40 |
| Operationsplan | 9 |
| Pallet Loading | 65 |
| Pfeil | 99 |
| Platine | 1, 7 |
| PosValue(TS _j) | 112 |
| PPS | 9, 42 |
| präemptive Operation | 53 |
| Priorität | 9 |
| Prioritätsfaktor | 57 |
| Prioritätsklassen | 10 |
| Prioritätsliste | 9, 42 |
| Prioritätsregel | 56 |
| Pseudomatrize | 97 |
| Pufferlager | 10, 40 |
| Quelle | 100 |
| Rollende Planung | 53 |
| Senke | 100 |
| Sequentielles 2D Bin Packing | 62 |
| SetValue(TS _j) | 71 |

| | |
|-----------------------------|--------|
| Sicherheitsabstand | 17, 31 |
| Stammdaten | 9 |
| Station | 9 |
| statisches Funktionselement | 38 |
| Steuerungseinheit | 40, 42 |
| Subsystem | 7 |
| System | 1, 7 |
| Tabubogen | 120 |
| Tabuzonen | 17, 34 |
| Teilefamilie | 13 |
| Teileset | 13, 69 |
| Teilesetfolge | 70, 80 |
| Teilesetkandidat | 47, 69 |
| Teilesetliste | 69 |
| topologisch sortiert | 103 |
| Trennlinie | 88 |
| Trog | 12 |
| Umformprozess | 12 |
| Umformung | 41 |
| Umhüllende | 17, 32 |
| Umsysteme | 7 |
| Value(i,j) | 107 |
| Value(TS _j) | 72 |
| Verbindung | 39 |
| Verzögerungselement | 41 |
| Verzweigungstyp | 76 |
| vor | 126 |
| Vorgänger | 100 |
| Wärmebehandlung | 14, 40 |
| Wartezeit | 126 |
| Weichglühen | 14 |
| Werkstattsteuerung | 43 |
| Werkstattsteuerungssystem | 44 |
| Zeitabschnitt | 40 |
| Zeitpunkt | 40 |
| zulässiges Layout | 78 |
| Zustandskette | 38 |
| Zustandsübergang | 40 |

1 Einleitung

1.1 Motivation

In der Flugzeugindustrie werden die Blechteile für Rumpf und Flügel typischerweise in zwei sequentiellen Bereichen einer Fertigungsstrasse gefertigt. Im ersten dieser Bereiche werden die Blechteile aus Tafeln ausgeschnitten und im zweiten Bereich werden sie umgeformt. Die ausgeschnittenen Blechteile werden *Platinen*¹ genannt. Platinen sind die einzigen Produkte, welche in der Blechstrasse gefertigt werden. Im weiteren Verlauf dieser Arbeit wird für die Platinen auch der Begriff *Artikel* verwendet, welcher zusätzliche Informationen wie z.B. Ausgangsform, Endform, Dicke und Material enthält.

Im Zentrum der Arbeit steht der zweite Bereich einer derartigen Fertigungsstrasse eines Flugzeugherstellers, für den eine Werkstattsteuerung² entworfen werden soll. Die wichtigsten Komponenten dieses Bereichs - im Folgenden *System* genannt - sind zwei Wärmbebehandlungsofen sowie eine Fluidzellpresse, zu denen jeweils ein Pufferlager für die bereitliegenden Aufträge gehört. Daneben gibt es weitere Bearbeitungsstationen von geringerer Bedeutung. Geleitet wird das System von einem Vorarbeiter, für die einzelnen Systemkomponenten sind jeweils Bediener zuständig. Im System können mehrere verschiedene *Aufträge* gleichzeitig bearbeitet werden. Bei einem Auftrag wird im System zu einem im vornerein nicht bekannten Zeitpunkt eine bestimmte Menge eines Artikels bestellt, der unter gewissen zeitlichen Restriktionen produziert werden soll.

Die Werkstattsteuerung des Systems wird bis anhin vom Vorarbeiter wahrgenommen. Im Zentrum steht dabei die Einplanung der Aufträge auf die Fluidzellpresse. Auf dieser können stets mehrere Platinen gleichzeitig umgeformt werden. Da aber pro Artikel nur ein Werkzeug (eine sogenannte *Matrize*) existiert, kann pro Pressung und pro Auftrag nur eine Platine umgeformt werden. Und weil im Allgemeinen verschiedene Stückzahlen zu fertigen sind, ergeben sich ständig wechselnde Kombinationen von Matrizen auf der Presse. Zudem haben die Aufträge verschiedene terminliche Prioritäten, die es zu berücksichtigen gilt. Der Vorarbeiter hat also das kombinatorische Problem zu lösen, die vorliegenden Aufträge den folgenden Pressungen zuzuweisen.

Die Anordnung der Matrizen in der Presse erfolgt direkt an der Maschine durch die Bediener. Diese Belegung muss gewissen Regeln gehorchen, die die Geometrien der Platinen bzw. ihrer Umformwerkzeuge mit einbeziehen. Werden diese Regeln befolgt, erhöht sich einerseits die Prozesssicherheit und andererseits kann sich die Lebensdauer von zentralen Komponenten der Presse signifikant erhöhen. Die Bediener haben somit ein mehrstufiges zweidimensionales Packingproblem zu lösen, wofür in dieser Arbeit der Begriff *sequentielles 2D Bin Packing* eingeführt wird.

Anstoss zu dieser Arbeit war die Idee der Partnerfirma, dem Vorarbeiter und den Bedienern ein Planungshilfsmittel für ihre jeweilige Aufgabe an der Fluidzellpresse zur Verfügung zu stellen. Dieses rechnergestützte Planungstool mit dem Namen *FluidPlan* soll erstens dem Vorarbeiter Vorschläge für die Bearbeitungsreihenfolge liefern und zweitens für die einzelnen Pressungen mögliche Layouts graphisch anzeigen. Im Weiteren soll die vorgeschlagene Reihenfolge dazu gebraucht werden, den vorgehenden Arbeitsschritt im System zu planen. Dort werden die Pla-

-
1. Neue Begriffe sind jeweils kursiv gedruckt und werden ab dann in der erklärten Bedeutung verwendet
 2. Auf Begriffe aus dem Bereich der Produktionsplanung und -steuerung wird in Kapitel 4.2.1 eingegangen.
-

tinien einer Wärmebehandlung unterzogen, die die Werkstoffeigenschaften derart ändert, dass die Platinen umformbar werden. Dieser geänderte Werkstoffzustand hält allerdings nur eine gewisse Zeit an, weshalb zwischen den Arbeitsgängen eine zeitliche Kopplung von ungefähr einer Woche besteht.

Neben der Einsatzplanung der Fluidzellpresse soll FluidPlan also gemäss den Vorstellungen der Firma auch den Einsatz der Wärmebehandlungsöfen planen. Implizit ist somit eine dritte, übergeordnete Aufgabe gestellt worden, denn die beiden Planungsaufgaben müssen in irgend einer Weise koordiniert werden.

Folgende Gründe sprechen dafür, FluidPlan als modulares Werkstattsteuerungssystem aufzubauen. Dabei übernimmt ein übergeordneter Modul die Koordination von unabhängigen Submodulen, die die genannten Teilaufgaben lösen.

- Das sequentielle 2D Bin Packing-Problem an der Presse (Wahl der Aufträge sowie Zuweisung eines Platzes) gehört zur Klasse der sogenannten Nestingprobleme, auch Packungs- oder Verschnittprobleme genannt. Bereits relativ einfache Problemvarianten gelten als NP-schwere Optimierungsprobleme. Es ist kaum denkbar, für die gleichzeitige Lösung sämtlicher Teilprobleme einen einzigen Algorithmus zu formulieren.
- Die Entkopplung der Teilprobleme erleichtert eine Weiterentwicklung von FluidPlan, wie sie z.B. durch veränderte Bedingungen in der Fertigung (neue Maschinen oder Verfahren) notwendig wird. Ist die Aufgabenteilung und die Kommunikation zwischen den Modulen einmal definiert, können die einzelnen Module unabhängig voneinander verändert werden.
- Analog zur Aufgabenhierarchie, wie sie bei den Mitarbeitern des Systems besteht, kann eine Aufgabenhierarchie der Module aufgebaut werden. Wird pro Hierarchiestufe und pro Aufgabenbereich ein Modul entwickelt, besteht die Möglichkeit, für jeden Modul ein spezifisch ausgelegtes Interface zu entwerfen.

Der letzte Punkt spricht ein allgemeines Problem an, das auftritt, wenn Algorithmen Aufgaben übernehmen, die zuvor von Personen durchgeführt wurden. Erstens fällen Menschen Entscheide oft intuitiv und beeinflusst durch verschiedenste Einflüsse, zweitens ist die Kommunikation zwischen den beteiligten Personen meistens schwach bis gar nicht formalisiert. Beim Entwurf der Algorithmen fehlen dann Hinweise zum Setzen von Parametern. In dieser Arbeit werden aus diesem Grund oft Annahmen getroffen, die jedoch an Ort und Stelle begründet werden.

An FluidPlan werden firmenseitig zusätzliche Erwartungen geknüpft, die eher globaler Natur sind und betriebliche Aspekte enthalten wie:

- Verringerung der Herstellkosten durch eine erhöhte Auslastung der Maschinen, durch einen schonenden Einsatz der Fluidzellpresse sowie durch Entlastung qualifizierter Mitarbeiter
- Steigern der Termintreue beziehungsweise senken der Durchlaufzeiten

1.2 Innovation

In dieser Arbeit wird ein Konzept für die Werkstattsteuerung einer Fertigungsstrasse vorgestellt, in welcher mit Hilfe einer Fluidzellpresse Blechteile für die Flugzeugindustrie gefertigt werden. Die Werkstattsteuerung basiert auf einem modularen, stufenweise einföhrbaren Konzept und erfolgt rechnergestützt. Die einzelnen Teilbereiche der Fertigung können damit relativ autonom gesteuert werden, wobei von einem übergeordneten Modul, der eine kapazitätsbasierte

Reihenfolgeplanung des gesamten betrachteten Bereiches vornimmt, Eckdaten vorgegeben werden.

Im Zentrum der Arbeit steht der komplexeste Modul dieser Werkstattsteuerung, welcher ein sequentielles 2D Bin Packing auf heuristische Art löst. Dieses hier nach Wissen des Autors erstmals beschriebene Optimierungsproblem trifft man aufgrund spezieller Randbedingungen beim Einsatz von Fluidzellpressen an. Das Problem zählt zur Klasse der Nestingprobleme.

Um Nestingvorschläge zu generieren und darzustellen, wird ein Paar gerichteter und gewichteter Graphen eingesetzt, welche ein System von Ungleichungen repräsentieren. Damit werden geometrische Informationen über die relativen Positionen der Matrizen beschrieben. Dieser Gebrauch von Graphen ist beispielsweise aus dem VLSI-Design bekannt und in SCHLAG beschrieben, wo die Knoten VLSI-Bausteine symbolisieren und die Verbindungen für geometrische oder logische Relationen stehen.

In dieser Arbeit wird die derartige Anwendung von gerichteten Graphen erweitert: Es werden ad hoc entwickelte Methoden und Algorithmen aufgezeigt, welche es erlauben, die Graphen dynamisch wachsen bzw. schrumpfen zu lassen. Das heisst, es werden in bestehenden Graphen unter Einhaltung der geforderten geometrischen Relationen Knoten eingefügt beziehungsweise entfernt, die dem Plazieren bzw. Entfernen von Matrizen in der Presse entsprechen.

1.3 Aufbau

Die Arbeit ist in zwei Teile gegliedert, welche neben dieser kurzen Einführung 6 Kapitel umfassen.

Im ersten Teil wird die ganze Fertigungsstrasse beschrieben, darauf aufbauend werden die verschiedenen Probleme und die daraus resultierenden Aufgaben vorgestellt. Kapitel 2 beschreibt dabei den betrachteten Fertigungsbereich, während in Kapitel 3 näher auf den Prozess in einer Fluidzellpresse eingegangen wird. Gestützt auf diese Informationen wird in Kapitel 4 unser Konzept für eine modulare Werkstattsteuerung einer Fluidzellpresse vorgestellt.

Der zweite Teil befasst sich mit der Lösung des sequentiellen 2D Bin Packing. In der Einführung wird das sequentielle 2D Bin Packing formal eingeführt und einige verwandte Probleme erwähnt. Kapitel 5 behandelt den kombinatorischen Teil des vorliegenden Nestingproblems, der geometrische Teil wird in Kapitel 6 gelöst. Kapitel 7 zeigt Aufbau und Analyse der Tests, mit welchen die entworfenen Algorithmen überprüft wurden.

Im Anhang wird schliesslich das Programm sowie die Oberfläche der implementierten Module und deren Bedienung beschrieben.

Abschliessend sei erwähnt, dass die Informationen, welche für die Lösung der einzelnen Teilprobleme relevant sind, in den Kapiteln 2 und 3 vorliegen. Deshalb erfolgt in der Einleitung zum zweiten Teil eine zusammenfassende, formale Beschreibung des vorliegenden sequentiellen 2D Bin Packing-Problems, die erlauben sollte, den zweiten Teil möglichst unabhängig zu lesen.

***TEIL I: WERKSTATTSTEUERUNG EINER
FLUIDZELLPRESSE***

2 System- und Aufgabenbeschreibung

Dieses Kapitel liefert die wesentlichen Grundlagen zu den in dieser Arbeit vorgestellten Konzepten und Algorithmen. Zuerst wird das im Zentrum der Arbeit stehende System inklusive seiner Komponenten beschrieben. Gestützt auf diese Beschreibung werden die Erwartungen an diese Arbeit hinterfragt, worauf die einzelnen Teilaufgaben formuliert werden können.

Hauptfunktionen des Systems

Rumpf und Flügel von Flugzeugen werden aus Hunderten von Belchteilen, sogenannten *Platinen*, zusammengesetzt. Die Fertigung dieser Platinen erfolgt in der Regel mittels vier wesentlichen Operationen (1-4) in einer Fertigungsstrasse, der *Blechstrasse*. Zuerst werden die Platinen aus Blechtafeln ausgeschnitten (1) und erhalten so ihre Ausgangsform im ebenen Zustand. Anschliessend erfolgt eine Wärmebehandlung (2), um eine gute Formbarkeit des Werkstoffes für die Umformung sicherzustellen. Die Formgebung (3) erfolgt in einer Fluidzellpresse; darin werden die Platinen mit Hilfe einer mit hohem Druck beaufschlagten Gummimembrane gegen Formwerkzeuge gedrückt. Üblicherweise müssen die Platinen abschliessend manuell nachbearbeitet (4) werden, um die geforderte hohe Genauigkeit der Endform zu erreichen.

Der Bereich der Blechstrasse, in dem die Wärmebehandlung, die Umformung sowie die manuelle Bearbeitung erfolgt, wird ab jetzt *System* genannt. Für gewisse Aufgaben im System soll in dieser Arbeit ein computergestütztes Planungshilfsmittel entwickelt werden, welches *FluidPlan* genannt wird. Mit diesem Hilfsmittel soll erstens der Einsatz der Fluidzellpresse optimiert und zweitens die Werkstattsteuerung des Bereiches unterstützt werden.

Gliederung des Kapitels

Abschnitt 2.1 beschreibt das System als Blackbox, eingebettet in seine Umgebung. In Abschnitt 2.2 werden anschliessend die einzelnen Systemkomponenten näher betrachtet. Darauf aufbauend kann das klassische betriebliche Problem der Reihenfolgeplanung aufgezeigt werden. Im Mittelpunkt steht dabei die Frage, wann welche Platinen an den einzelnen Fertigungseinrichtungen bearbeitet werden sollen, also wie die Steuerung der Werkstatt aussehen soll.

Abschnitt 2.3 beschreibt das an der betrachteten Fluidzellpresse auftretende Optimierungsproblem detailliert genug, damit alle im Abschnitt 2.4 benötigten Informationen bekannt sind. Dort werden die an FluidPlan gestellten Erwartungen analysiert, worauf Teilaufgaben formuliert werden können.

2.1 Das System und dessen Umgebung

Das System wird in dieser Arbeit in verschiedenen Hierarchiestufen in entsprechendem Detaillierungsgrad gezeigt. Die Bezeichnung der Stufen wurde aus BÜCHEL1 übernommen. Gemäss dieser Terminologie orientieren sich die Hierarchiestufen an derjenigen Ebene, in der das betrachtete System selbst liegt. Diese Ebene wird mit $S(0)$ bezeichnet. $S(1)$ ist somit die Bezeichnung für das *Obersystem*, das aus $S(0)$ und seiner Umgebung, den sogenannten *Umsystemen*, gebildet wird. *Subsysteme* von $S(0)$ (Ofen, Presse etc.) werden $S(-1)_1$, $S(-1)_2$, ... genannt und sind mittels Indizes numeriert.

2.1.1 Das System als Blackbox

Abbildung 2-1 zeigt $S(0)$ als Blackbox eingebettet in das Obersystem $S(1)$. Als Blackbox lässt sich jeder aufgabenmässig abgrenzbare Bereich darstellen. Eine Blackbox hat gemäss HABERFELLNER ET AL. Material, Information und Energie als Hauptflüsse von Input und Output. In Anlehnung daran wird für die Inputs ‚Material‘ und ‚Information‘ in Abbildung 2-1 aufgezeigt, woher sie kommen und wie sie entstehen.

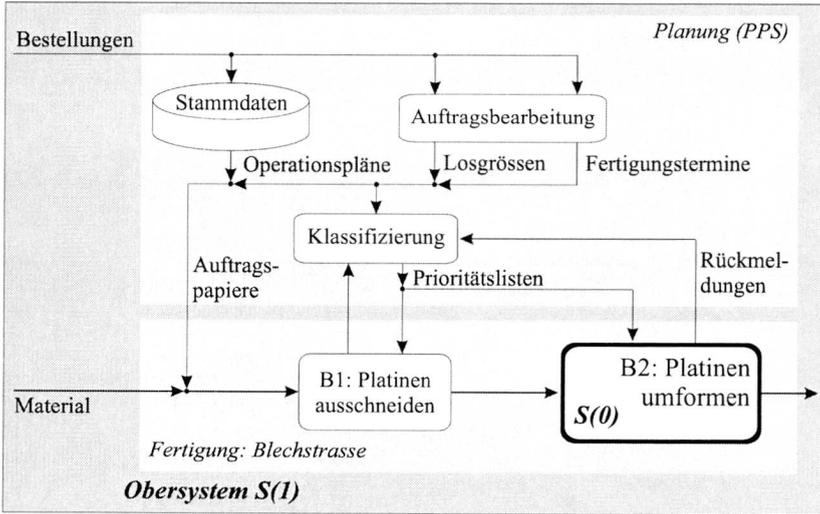


Abbildung 2-1: Das System und seine Umsysteme

2.1.2 Input und Output des Systems

Material

Die zentrale Aufgabe des Systems besteht darin, flache Platinen mit Hilfe einer Presse umzuformen. In Abschnitt 2.3 sowie in Kapitel 4 wird näher auf den Umformprozess eingegangen. An dieser Stelle begnüge man sich mit folgenden Feststellungen: Der materielle Input von $S(0)$ besteht aus flachen und der Output aus umgeformten Platinen. Eine Bestellung einer bestimmten Menge von identischen umgeformten Platinen wird *Auftrag* genannt.

Information

Jeder Auftrag wird aus seinem Weg durch Fertigung und Montage von sog. *Auftragspapieren* begleitet. Sie beschreiben u.a. detailliert, welche Operationen in welcher Reihenfolge zur Herstellung des betreffenden *Artikels*¹ durchzuführen sind. Diese Gliederung in einzelne Operatio-

1. Bei auftragsunabhängigen Daten von Produkten wird im betrieblichen Kontext generell der Begriff Artikel benutzt. Da Platinen die einzigen in dieser Arbeit betrachteten Artikel sind, können beide Begriffe verwendet werden.

nen wird *Operationsplan (OP)* genannt, ein Eintrag im OP heisst *Arbeitsgang*². Der OP für einen Artikel ist auftragsunabhängig. Die OP's aller Artikel werden als *Stammdaten* abgelegt.

Vervollständigt werden die Auftragspapiere durch auftragspezifische Daten, welche in der Planungsebene durch die Produktionsplanung und -steuerung (PPS) bestimmt werden. Dabei wird für die vorliegenden Bestellungen eine Stücklistenauflösung gemacht, woraus die zu fertigenden Platinen und ihre *Losgrösse* (Bestellmenge) resultieren. Dann werden diese Bestellungen durch Rückwärtsterminierung in die verschiedenen Arbeitsgänge grob eingeplant, woraus für jeden Auftrag ein *Fertigungstermin* pro Arbeitsgang resultiert, welcher in Wochen angegeben wird.

Jedem Auftrag wird eine terminliche *Priorität* zugewiesen, welche angibt, wie dringend die Fertigung eines Auftrages ist. Diese *Klassifizierung* kann sich während der Fertigung ändern, weshalb sie gesondert von den Auftragspapieren an die einzelnen *Arbeitsplätze*³ übermittelt wird. Die Priorität wird normalerweise automatisch vom Verzug auf den geplanten Fertigungstermin abgeleitet, kann aber - z. B. bei Eilaufträgen - auch durch einen Sachbearbeiter in der Planungsebene heraufgesetzt werden.

Um eine Übersicht über die Dringlichkeiten aller aktuellen Aufträge zu haben, wird jedem Arbeitsplatz periodisch eine sogenannte *Prioritätsliste* übergeben. Darin sind alle Aufträge, die am Arbeitsplatz anstehen, nach absteigender Priorität sortiert aufgeführt. Das Auffrischen der Prioritätslisten in der PPS erfolgt mit Hilfe der Rückmeldungen aus der Fertigung.

Die Rolle des Vorarbeiters im Informationsfluss

Geleitet wird $S(0)$ von einem *Vorarbeiter*, wie Abbildung 2-2 zeigt. Seine Hauptfunktion ist das Steuern des betrachteten Werkstattbereiches⁴. Genau bei dieser Tätigkeit soll er durch FluidPlan unterstützt werden, weshalb das Planungssystem in der Abbildung auch eingezeichnet wurde.

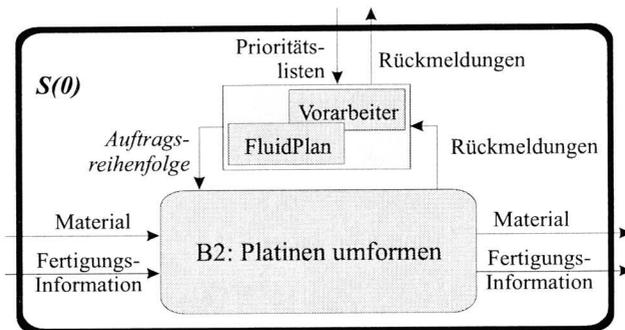


Abbildung 2-2: Das System mit den Steuerungsinstanzen ‚Vorarbeiter‘ und ‚FluidPlan‘

2. Nach SCHRÖDEL bildet ein Arbeitsgang „einen technologischen Abschnitt eines Arbeitsablaufs an einem Arbeitsplatz (bzw. Arbeitsort)“.
3. Ein Arbeitsplatz fasst alle *Stationen* - d.h. alle Maschinen, Apparate bzw. Einrichtungen - zusammen, an denen derselbe Arbeitsgang durchgeführt werden kann.
4. Bei Fertigungsbereichen wird meist der Begriff ‚Steuerung‘ verwendet, obwohl es sich eigentlich um ein Regeln des Bereiches handelt, da eine geschlossene Datenkette vorliegt (vgl. hierzu GEERING). Viele Arbeiten befassen sich explizit mit dem Thema ‚Regelung der Produktion‘, so z.B. RECH.

Der Vorarbeiter bestimmt aufgrund der Prioritätsliste, der Auslastung der einzelnen Arbeitsplätze und aufgrund der in $S(0)$ verfügbaren Mitarbeiter die Bearbeitungsreihenfolge der vorliegenden Aufträge. Die Aufgaben der Mitarbeiter werden in Abschnitt 2.2.4 aufgeführt. Abgeschlossene Arbeitsgänge werden mittels BDE⁵ abgemeldet, womit die PPS sowie der Vorarbeiter über den Arbeitsfortschritt der Aufträge informiert sind. Zusätzlich wird der Vorarbeiter von seinen Mitarbeitern über Störungen informiert, welche die geplante Auftragsreihenfolge gefährden oder gar verhindern.

2.1.3 Steuer- und Zielgrößen des gesamten Systems

In Abbildung 2-2 ist gut sichtbar, dass die Information in zwei Kategorien unterteilt werden kann. Zum einen ist Fertigungsinformation nötig, welche zum Auftrag gehört und das Material durch die ganze Fertigung begleitet. Zum anderen existiert Steuerungsinformation, welche die Bearbeitungsreihenfolge unter den Aufträgen in der Fertigung bestimmt.

Sind die Auftragspapiere (Fertigungsinformation) einmal generiert, kann von der Planungsebene (PPS) aus nur mit der Prioritätsliste Einfluss auf die Steuerung ausgeübt werden. Die Werkstattsteuerung selbst liegt dann alleine in den Händen des Vorarbeiters von $S(0)$.

Zu optimieren ist die Durchlaufzeit aller Aufträge in Funktion der Priorität, unter Berücksichtigung der im Folgenden beschriebenen Restriktionen. Je höher die Priorität ist, desto schneller muss ein Auftrag gefertigt werden. Eine messbare Zielgröße wird in Abschnitt 4.4.1 vorgestellt. Für die Auslastung der möglichen Kapazität des Systems wird eine untere Grenze gesetzt, die nicht unterschritten werden soll. Wie die verschiedenen *Prioritätsklassen*⁶ bezüglich ihrer Durchlaufzeiten untereinander zu gewichten sind, ist nicht vorgegeben. Beim Entwurf des Konzepts wird näher auf die Prioritätsklassen eingegangen und es wird ein in dieser Arbeit entwickeltes Modell aufgezeigt, wie die Klassen untereinander gewichtet werden können. Diese Gewichtung wird später auch zur Bewertung verschiedener Lösungen herangezogen.

2.2 Systembeschreibung

In diesem Abschnitt werden zuerst die Materialflüsse zwischen den *Komponenten*⁷ des Systems aufgezeigt. Anschliessend werden drei beispielhafte Operationspläne betrachtet; dabei sieht man, an welchen Systemkomponenten ein Auftrag typischerweise bearbeitet wird und in welcher Reihenfolge dies geschieht. Dann werden die Systemkomponenten soweit beschrieben, dass die zu lösenden Probleme verständlich sind. Am Schluss des Kapitels wird aufgezeigt, wo im System Entscheidungen getroffen werden und welche Auswirkungen sie haben.

2.2.1 Materialfluss im System

Abbildung 2-3 zeigt die drei zu Beginn des Kapitels aufgeführten Komponenten des Systems: Es sind die Arbeitsplätze Wärmebehandlung, Umformung und manuelle Bearbeitung. Zu den ersten beiden gehört je ein *Pufferlager*, worin alle zur Bearbeitung am betreffenden Arbeitsplatz bereit liegenden Aufträge enthalten sind. Bei der Wärmebehandlung sind zwei Verfahren auf-

5. Betriebsdatenerfassung; erfolgt i.a. elektronisch

6. Menge aller Aufträge mit gleicher Priorität

7. Für die ‚realen‘ Bausteine des Systems wird bewusst der Begriff Komponente verwendet. Die in der später folgenden Modellbildung definierten ‚idealen‘ Bausteine des Systems werden dann Elemente genannt.

gezählt (Weichglühen, Lösungsglühen). In Abschnitt 2.2.3 wird ausgeführt, wieso beide Verfahren notwendig sind; wichtig für den Materialfluss ist vor allem, dass nur Platinen das System verlassen können, die beim Arbeitsplatz Wärmebehandlung zuletzt lösungsgeglüht wurden.

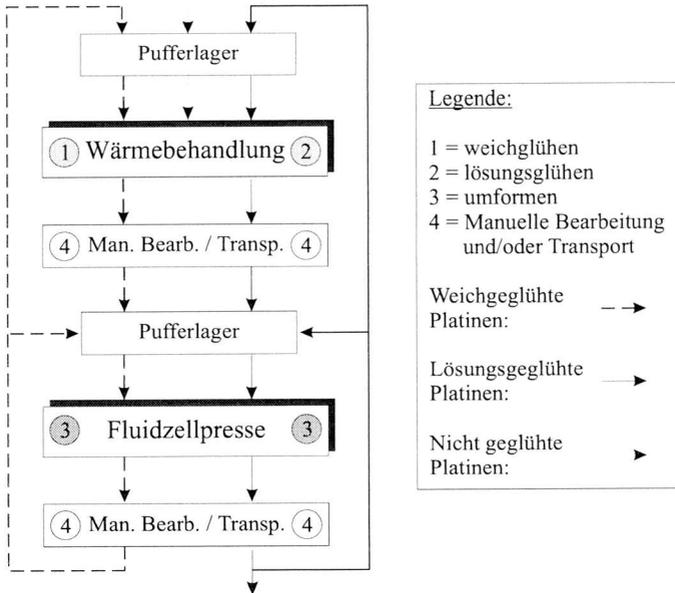


Abbildung 2-3: Möglicher Materialfluss im System

Zwischen den Arbeitsplätzen muss nicht immer eine manuelle Bearbeitung erfolgen. Um in Abbildung 2-3 diesbezüglich nicht jedesmal eine Fallunterscheidung machen zu müssen, sind die manuelle Zwischenbearbeitung und der Transport zusammengefasst worden. Ebenfalls aus Übersichtgründen wurde die manuelle Bearbeitung zweimal gezeichnet.

2.2.2 Typische Operationspläne

Der vorhin beschriebene mögliche Materialfluss zeigt ein „statisches“ Bild des Fertigungsreiches. Durch die Betrachtung einzelner Artikel bzw. ihrer Operationspläne erhält man ein dynamisches Bild konkreter Materialflüsse. In Abbildung 2-4 sind exemplarisch die einzelnen Arbeitsgänge von drei typischen Artikeln im System aufgeführt:

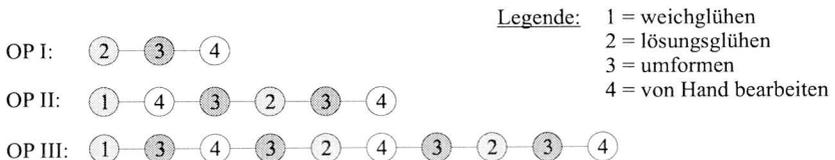


Abbildung 2-4: Aufbau von typischen Operationsplänen

Eine Analyse von 800 zufällig ausgewählten Operationsplänen (aus ca. 5000) zeigt, wie die Artikel prozentual auf die drei OP-Typen aus Abbildung 2-4 aufgeteilt sind:

- OP I: 77% der analysierten Artikel werden nur einmal umgeformt und vorgängig lösungsgeglüht
- OP II: 20% werden zweimal umgeformt; i.a. werden diese Artikel zuerst weich- und dann lösungsgeglüht
- OP III: 3% der betrachteten Artikel kommen mindestens dreimal in die Fluidzellpresse und zwei- bis dreimal zur Wärmebehandlung

Eine zwischenzeitliche Bearbeitung von Hand ist zwischen allen Arbeitsgängen an Öfen und Presse möglich, es gibt somit keinen ‚typischen‘ Operationsplan. Da also weder die Anzahl Einträge auf einem OP noch die Reihenfolge der verschiedenen Arbeitsplätze vorgegeben sind, muss für FluidPlan eine flexible Auftragsmodellierung konzipiert werden.

2.2.3 Systemkomponenten

Die Komponenten von S(0) sollen nun soweit vorgestellt werden, dass folgende Fragen am Schluss von Abschnitt 2.2 beantwortet sind:

- Was ist die Funktion der Komponente?
- Was sind Input und Output von Material und Information?
- Mit welchem (zeitlichen) Takt verändert sich der Zustand der Komponente, d.h. wie lange ist die Bearbeitungszeit für einen Auftrag?

Fluidzellpresse und Matrizen

In Kapitel 4 wird auf den Umformprozess in einer Fluidzellpresse, auf die Vor- und Nachteile des Verfahrens sowie auf die Einsatzgebiete eingegangen. Die Funktionsweise wird hier soweit erklärt, dass obige Fragen beantwortet werden können.

Mit Hilfe von Fluidzellpressen werden die Platinen umgeformt. Im Gegensatz zum konventionellen Tiefziehen von Blechteilen, bei dem mit einem Satz starrer Werkzeuge gefertigt wird, benutzt die Fluidzelltechnik eine Flüssigkeit zum Aufbringen der Umformkraft. Abbildung 2-5 zeigt schematisch den Endzustand des Prozesses.

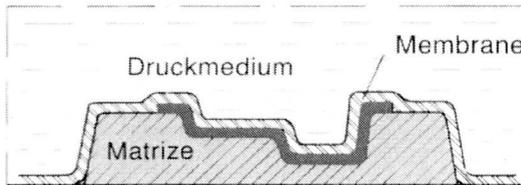


Abbildung 2-5: Prinzipielle Darstellung der Fluidzelltechnik (Bild: ABB)

Die formgebende Vorrichtung wird *Matrize* genannt. Die dunkel gezeichnete Platinenform wird auf sie gelegt und nötigenfalls fixiert. Die Matrize selbst liegt auf einer nicht dargestellten Metallplatte (dem sog. *Trog*), welche in die Presse hinein und daraus heraus gefahren werden kann. In der Presse befindet sich ein Ölbad als Druckmedium, welches durch eine *Gummimembrane* abgedichtet ist. In diesem Bad kann ein Druck von einigen hundert bar aufgebaut werden, worauf die Membrane an die Matrize gepresst wird und dadurch die Platinenform umformt.

Die in dieser Arbeit betrachtete Presse (vgl. Abbildung 2-6) besteht aus einem querliegenden Zylinder und aus zwei langen, rechteckigen *Metalltrögen*, die abwechselnd in die Presse eingefahren werden können. Pro Trog steht eine Fläche von max. 2.5 m^2 zur Verfügung. Ca. 70% der vorliegenden rund 5000 Matrizen belegen weniger als 0.1 m^2 Fläche und knapp 90% sind kleiner als 0.35 m^2 ; man könnte also über zwei Dutzend Matrizen auf einmal auf einen Trog legen. Im Normalfall sind es zwischen 10 und 15 Stück, da stets auch grössere Matrizen benutzt werden. Weil die Aufträge i.a. unterschiedliche Losgrößen haben, liegen ständig wechselnde Kombinationen von Matrizen miteinander auf einem Trog, denn pro Artikel existiert i.a. nur eine Matrize. Eine Gruppe von Matrizen, die miteinander auf den Trog kommt bzw. miteinander ein-geplant ist, wird *Teileset*⁸ genannt.



Abbildung 2-6: Bediener beim Beschicken eines Troges (Bild: ABB)

Ein Pressentakt eines Troges inklusive Umrüsten (Plazieren neuer Matrizen sowie Auswechseln der Platinen) dauert ca. 5 Minuten, was bei einer Arbeitszeit von 8.5 Stunden pro Tag ungefähr 100 Pressungen ergibt. Im Pufferlager liegen typischerweise 100 - 200 Aufträge bereit.

Über 90% der Artikel können mit einem Normdruck umgeformt werden. Die restlichen knapp 10% müssen mit andern Drücken gefertigt werden; Die Aufträge werden deshalb in nicht mischbare Gruppen von Artikeln, in sogenannte *Teilfamilien* (TF), eingeteilt. Innerhalb einer Teilfamilie gibt es keine Restriktionen bezüglich der Zusammensetzung des Teilesets. Es existieren aber Empfehlungen in Form von sogenannten *Beschickungsregeln*, die unter anderem auch die Zusammensetzung des Teilesets betreffen. Auf sie wird im Kapitel 4 eingegangen.

8. Der Name hat sich eingebürgert, obwohl eigentlich ein Matrizen- oder Auftragsset bestimmt wird.

Wärmebehandlungsöfen (und -verfahren)

Die im Flugzeugbau eingesetzten Werkstoffe der Platinen besitzen i.a. schlechte Umformeigenschaften. Die Platinen müssen deshalb vor der Umformung einer Wärmebehandlung unterzogen werden, um die gewünschten Eigenschaften zu erhalten.

Das am häufigsten eingesetzte Verfahren ist das *Lösungsglühen*. Dabei werden die Platinen je nach Dicke 30 bis 60 Minuten auf einer (vom Material abhängigen) Temperatur von ca. 500° Celsius gegläht und anschliessend in einem Wasserbad abgeschreckt. Der so erreichte Gefügestand ist instationär; gleich nach dem Abschrecken beginnt die sogenannte *Aushärtung*, bei der sich der alte, schlecht formbare Zustand wieder einstellt. Je tiefer die Temperatur während des Aushärtens, desto langsamer läuft dieser Prozess ab. Werden die Platinen im Pufferlager der Presse in Kühlboxen mit Temperaturen um den Nullpunkt gelagert, lässt sich die Umformbarkeit auf ca. eine Woche erhöhen.

Die notwendige Kühlung der Platinen sowie die zeitlich befristete Umformbarkeit sind auf den ersten Blick Nachteile des Verfahrens. Da aber bei der Montage der ursprüngliche Gefügestand erwünscht ist, entpuppt sich zumindest der zweite Punkt schliesslich als Vorteil.

Sind die Umformgrade zu hoch bzw. die Formen zu komplex, reicht die durch Lösungsglühen erreichbare Umformbarkeit nicht aus. Ungefähr ein Fünftel der Artikel muss deshalb anders behandelt werden, meist wird dafür das *Weichglühen* eingesetzt. Die Prozesstemperatur liegt hier je nach Material bei 300 bis 460° Celsius und wird 0,5 bis 4 Stunden gehalten. Weichgeglühte Platinen härten nicht aus und können deshalb bei Raumtemperatur gelagert werden und haben keine zeitliche Kopplung mit der Umformung. Nach der Verformung muss der alte Gefügestand wieder hergestellt werden, wofür das bereits bekannte Lösungsglühen eingesetzt wird. Der beim Abschrecken unvermeidbare Verzug macht es nötig, die Teile ein zweites Mal auf der Presse umzuformen. Artikel, die weichgeglüht werden müssen, werden also immer mindestens zweimal umgeformt, wobei dieselben Matrizen verwendet werden.

Auch an den Öfen werden die anstehenden Aufträge zu nicht mischbaren Teilfamilien zusammengefasst, indem pro Verfahren nach Glühzeit und -temperatur unterschieden wird. Diese Teilfamilien decken sich nicht mit jenen an der Presse.

Es stehen zwei Öfen zur Verfügung, und die zwei vorgestellten Wärmebehandlungsverfahren können auf beiden Öfen durchgeführt werden. Die Platinen liegen jeweils auf herausnehmbaren Gittern, wobei je nach Verfahren unterschiedliche *Chargierregeln* (Regeln zur Anordnung der Platinen) gelten. Aufgrund ihrer Dimensionen können ca. 5% der Platinen nicht in beiden Öfen gegläht werden, was aus dem Operationsplan durch die Angabe der zugelassenen Stationen ersichtlich wird. Die Kapazität der Öfen beträgt 1.8 bzw. 3 m². Analog zur Umformung existiert auch vor der Wärmebehandlung ein Pufferlager.

Handarbeits- und Richtstationen

Die Platinen müssen manchmal vor oder nach dem Umformen an Handarbeitsstationen bearbeitet werden. Da die engen Masstoleranzen beim Pressen oft nicht eingehalten werden können, müssen nach der letzten Umformung die Platinen generell an den sogenannten Richtstationen nachgerichtet werden, bevor sie das System verlassen.

Diese Arbeitsgänge sind auf dem Operationsplan eingetragen, die Operationsdauer kann aber im Gegensatz zur Dauer an den beiden andern Arbeitsplätzen nur geschätzt werden. Für diesen Arbeitsplatz soll bei der Werkstattsteuerung kein Modul entworfen werden, aber da diese Stationen oft zwischen den geplanten Modulen „Umformung“ und „Wärmebehandlung“ liegen, sind diese ungenauen Bearbeitungszeiten Störgrössen, die es zu beachten gilt.

2.2.4 Entscheidungen und Auswirkungen

Nach der Vorstellung des Materialflusses sowie der Systemkomponenten interessieren folgende Fragen:

- Was passiert in welchem zeitlichen Rahmen im System?
- Welche Ereignisse können innerhalb des Systems entschieden werden (Steuergrößen)?
- Wie wird die Leistung der Systemkomponenten beurteilt (Zielgrößen) und wie kann diese Leistung durch die Steuergrößen beeinflusst und optimiert werden?

Ausgangslage

Folgende Kenngrößen der Komponenten bzw. Arbeitsplätze des Systems sind bekannt:

Umformung

- Fluidzellpresse mit 2 Trögen à 2,5 m²
- Es gibt insgesamt etwa 5000 Matrizen, wobei 70% weniger als 0.1 m² Grundfläche belegen
- Taktzeit (inkl. Umrüsten) ca. 5 Minuten pro Trog
- Pro Auftrag steht eine einzige Matrize zur Verfügung
- 10 - 15 Matrizen pro Pressung auf dem Trog
- Nicht mischbare Teilefamilien aufgrund des Druckes
- Pufferlager mit 100 - 200 Aufträgen

Wärmebehandlung:

- Zwei Öfen mit 1.8 resp. 3 m² Kapazität
- Zwei Verfahren werden eingesetzt, beide in beiden Öfen durchführbar
- Über die verfahrensabhängigen Chargierregeln erhält man die beanspruchte Gesamtfläche eines Auftrages
- Die Prozessdauer beträgt je nach Verfahren 0.5 bis 4 Stunden
- Jeweils ca. 5% der Artikel können aufgrund der Dimension nur in einem Ofen geglüht werden
- Pufferlager mit 100 - 200 Aufträgen

Handarbeitsplätze: - Die Bearbeitungsdauer an den Handarbeitsplätzen ist nur geschätzt.

Ablauf

Der Vorarbeiter des Systems bestimmt aufgrund der Prioritätslisten, welche Aufträge an den verschiedenen Arbeitsplätzen in der nächsten Zeit (ca. ein halber bis ein ganzer Tag) bearbeitet werden sollen. Somit bestimmt er auch, welche Teilefamilie bearbeitet wird. Für die einzelnen Arbeitsplätze heisst das konkret:

1. Der Vorarbeiter sichtet die Aufträge im Pufferlager der Wärmebehandlung, worauf er provisorisch die Aufträge bestimmt, die in den nächsten Chargen geglüht werden sollen. Erst wenn die Verfügbarkeit der Pressenmatrizen gewährleistet ist, werden die Aufträge definitiv für diese Chargen vorgesehen. Für die Einteilung der einzelnen Chargen und die Bedienung der Öfen ist ein Mitarbeiter zuständig, der sich diesbezüglich nicht mehr mit dem Vorarbeiter abspricht.
2. Aufträge, die an Handarbeitsplätzen bearbeitet werden müssen, werden durch den Vorarbeiter an die dort zuständigen Mitarbeiter verteilt.

3. Beim Pufferlager der Presse bestimmt wiederum der Vorarbeiter, welche Aufträge in den nächsten Stunden zu fertigen sind. Die Feinsteuerung übernehmen jedoch die Bediener der Presse selber. Sie bestimmen also, welche Aufträge miteinander umgeformt werden und wie die zugehörigen Matrizen auf dem Trog verteilt werden.

Diese Vorgehensweise ist flexibel genug, die täglich ca. fünf Aufträge mit manuell erhöhter Priorität schneller als gewöhnliche Aufträge zu behandeln. Der Grund liegt darin, dass sie früh genug bekannt sind, um insbesondere die Verfügbarkeit der entsprechenden Matrizen zu prüfen. An den Öfen und an der Presse müssen dann zu gegebener Zeit die zugehörigen Teilefamilien gewählt werden.

Bei diesem Ablauf tritt folgendes Problem auf: Die Prioritätsliste liefert zwar einen nach Wichtigkeit geordneten Überblick über die anstehenden Aufträge, jedoch fehlen darauf geometrische Informationen über die Matrizen. Diese Informationen sind auch sonst auf keinem dem Vorarbeiter und seinen Mitarbeitern zugänglichen EDV-System abrufbar, und über 5000 Matrizenformen kann sich niemand merken. Einzig über die Form der Platinen kann die Form der Matrizen abgeschätzt werden, was allerdings recht aufwendig ist, da in diesem Fall alle Aufträge im Pufferlager angeschaut werden müssen.

Steuer- und Zielgrößen innerhalb des Systems

Gemäss Ablauf wird innerhalb des Systems zweistufig über die Auftragsreihenfolge an den einzelnen Arbeitsplätzen entschieden.

- Die erste Steuergrösse ist die durch den Vorarbeiter bestimmte grobe Auswahl der in den nächsten Stunden zu bearbeitenden Aufträge. Sie basiert bei der Presse auf den vorliegenden Prioritätslisten und den momentanen Bedingungen an den einzelnen Arbeitsplätzen. Darauf basierend wird abgeleitet, welche Aufträge für die Wärmebehandlung vorzusehen sind. Es handelt sich dabei um eine kapazitätsbasierte Einplanung, die insbesondere die Koordination der Arbeitsplätze übernimmt. Die Anweisungen werden den Mitarbeitern an den einzelnen Arbeitsplätzen mündlich mitgeteilt.
- An den Arbeitsplätzen wird die genaue Reihenfolge und Position (Lage auf dem Gitter bei der Wärmebehandlung resp. Lage im Trog bei der Umformung) bei der Bearbeitung konkret festgelegt. Im Vordergrund steht bei dieser Aufgabe nicht die Kapazität, obwohl diese natürlich auch berücksichtigt wird. Beide Male muss primär ein sogenanntes Nestingproblem (siehe unten) unter Berücksichtigung von Chargier- bzw. Beschickungsregeln gelöst werden.

Bezüglich Zielgrößen gelten die für das Gesamtsystem gemachten Bemerkungen in Abschnitt 2.1.3 auch für die Komponenten: Zu optimieren ist die Durchlaufzeit aller Aufträge in Funktion der Priorität. Je höher die Priorität eines Auftrags ist, umso schneller muss er gefertigt werden. Eine Gewichtung der verschiedenen Prioritätsklassen untereinander muss im Konzept vorgeschlagen werden, da keine konkreten Vorgaben bestehen. Für die kapazitive Auslastung soll eine ebenfalls vorgegebene untere Grenze eingehalten werden.

2.3 Das Nestingproblem der Fluidzellpresse

Muss man Blech, Stoff, Glas etc. aus einer Grundform ausschneiden, spricht man von einem Verschnittproblem. Müssen Maschinen in einer Halle angeordnet oder eine Leiterplatte bestückt werden, spricht man von einem Packungsproblem. In beiden Fällen soll man möglichst viele vorgegebene „kleine Teile“ möglichst wenigen „grossen Teilen“ zuordnen. Bei Problemstellungen aus diesem Gebiet wird auch der Begriff *Nesting(problem)* verwendet.

Bevor in Abschnitt 2.4 die Aufgaben detailliert beschrieben werden können, muss Punkt 3 des oben beschriebenen Ablaufs genauer beschrieben werden. Denn das Problem des Vorarbeiters - also eine grobe Reihenfolgeplanung für die verschiedenen Arbeitsplätze - ist aus der Beschreibung des Ablaufs ersichtlich, nicht aber das Problem der Leute an der Presse⁹. Im ersten Teil dieses Abschnitts werden deshalb einige Begriffe eingeführt, die dann im zweiten Teil bei der Beschreibung des vorliegenden Nestingproblems verwendet werden.

2.3.1 Übersicht der Beschickungsregeln

Eine *Matrize M* kann - unter Einhaltung der hier vorgestellten Beschickungsregeln - an beliebiger Stelle auf dem Trog einer Fluidzellpresse plaziert werden, insbesondere muss eine Matrize nicht auf dem Trog fixiert werden.

Spezifische Beschickungsregeln

Für einige wenige Matrizen (< 3%) existieren „verbotene Zonen“. Das sind Bereiche des Troges, in die diese Matrizen nicht gelegt werden dürfen, da sonst die Prozessicherheit nicht mehr gewährleistet ist. In Abschnitt 3.3.2 wird ein Modell eingeführt, wie diese sogenannten *Tabuzonen* behandelt werden können.

Allgemeine Beschickungsregeln

Matrizen dürfen nicht beliebig nahe beieinander liegen, da sie sich bei einer Pressung gegenseitig beschädigen könnten. Auch zwischen dem Trogrand und den Matrizen muss ein minimaler Abstand eingehalten werden. Dieser minimal notwendige Abstand kann mit allgemeingültigen Regeln für alle Matrizen angegeben werden und wird *Sicherheitsabstand* genannt.

Da die überwiegende Anzahl der Matrizen eine annähernd rechteckige Grundform aufweist, werden alle Matrizen vereinfachend als Rechtecke betrachtet. Zu diesem kleinsten umhüllenden Rechteck wird der Sicherheitsabstand hinzuaddiert, woraus die *Umhüllende* einer Matrize *M* resultiert. Die Umhüllenden benachbarter Matrizen dürfen sich beim Nesten berühren.

Weiterhin werden an das Teileset, also an die Matrizen, welche miteinander auf dem Trog zu plazieren sind, Anforderungen gestellt, die die Unterhaltskosten der Presse signifikant senken helfen. Hohe Matrizen sollten nicht nebeneinander liegen. In einem vorgeschlagenen *Layout*¹⁰ müssen also die *Nachbarschaften* überprüft werden, d.h. die relativen Positionen der Matrizen untereinander. Aus dieser Forderung lassen sich Bedingungen für die Zusammensetzung des Teilesets ableiten, deren Einhaltung überprüft werden kann, bevor dann die Layoutvorschläge zusammengestellt werden.

2.3.2 Problembeschreibung

Für die Bediener der Presse stellt sich folgende Aufgabe: Gegeben ist ein *Trog T* sowie eine Menge von *Aufträgen A*. Die Bediener der Presse müssen nun aus den bereitliegenden Aufträgen jene Matrizen bestimmen, die die bereits im Trog liegenden aus ihrer Sicht am optimalsten zu einem neuen Teileset ergänzen. Im Allgemeinen werden einige Aufträge bereits bearbeitet,

-
9. Das Nestingproblem bei der Chargierung der Öfen wird nicht näher behandelt, da keine wesentlichen Restriktionen vorliegen, die dem Ofenbediener Mühe bereiten. Erwartet wird lediglich eine Kapazitäts-Analyse (Reihenfolge und Inhalt der Chargen).
 10. Die geometrische Anordnung eines Teilesets auf dem Trog wird *Layout* genannt.

so dass ihre Matrizen bereits auf dem Trog liegen. Erst wenn alle Platinen eines Auftrags umgeformt sind, wird die entsprechende Matrize entfernt und es entsteht Platz für neue Matrizen. Im Folgenden werden die gegebenen Attribute (Inputdaten, Tabelle 2-1), die Zielfunktion sowie die Restriktionen aufgelistet.

Input

| Input | Attribute | Bemerkungen |
|---------------------------|--|---|
| Menge A: | $= \{A_1, \dots, A_d\}$ | Aufträge, die vor der Presse bereit liegen oder bereits eingeplant bzw. in Bearbeitung sind. |
| Pro A_i : | w_i Pri_i Termin _i M_i | Anzahl notwendiger Pressungen von A_i Priorität von A_i Spätester Umformtermin von A_i (falls lösungsgeglüht) Zu A_i gehörende Matrize M_i , inklusive Angaben über die Umhüllende, die Tabuzonen etc. |
| Trog T1: (dito für T2) | L_{T1}, B_{T1} $TS = \{TS_j, \dots, TS_n\}$ | Länge (L_{T1}) und Breite (B_{T1}) des Troges T1 Bereits geplante Folge von Teilesets TS von T1 |
| Pro TS_j | $= \{M_{j,1}, \dots, M_{j,m}\}$ | Menge der TS_j zugewiesenen Matrizen, inklusive deren Anordnung auf T1 |

Tabelle 2-1: Wichtigste Inputdaten des Nestingproblems

Zielfunktionen

Die Vorgabe für die Bediener der Presse lautet in etwa: Der Trog soll möglichst gut ausgefüllt werden, wobei Aufträge mit höherer Priorität bevorzugt werden sollen. Nestingprobleme bestehen aus einem kombinatorischen und einen geometrischen Teil. Gestützt auf die Bemerkungen aus Abschnitt 2.1.3, wurde die Zielfunktion für die vorliegende Arbeit wie folgt aufgeteilt:

- 1. Kombinatorisches Problem:** *Es ist eine Folge von zulässigen Teilesets zu bestimmen. Die Teilesets werden in der gegebenen Reihenfolge abgearbeitet. Zulässig ist ein Teileset dann, wenn die zugewiesenen Matrizen unter Einhaltung der Restriktionen miteinander im Trog plaziert werden können. Die „durchschnittliche Priorität“ der Aufträge dieser Teilesets soll möglichst hoch sein.*
- 2. Geometrisches Problem:** Das Finden eines legbaren Layouts für ein gegebenes Teileset ist Ziel des geometrischen und zugleich Randbedingung des kombinatorischen Problems. In Kapitel 6 wird eine Zielfunktion zur Bewertung der gefundenen Layouts vorgestellt. Wichtiger Bestandteil wird die hier verbal formulierte Forderung sein: *Der bei einer Layoutänderung anfallende Handlingsaufwand soll minimal sein.* Das bedeutet, dass die bei einem Layoutwechsel im Trog verbleibenden Matrizen nicht unnötigerweise verschoben werden sollen. Sonst steigen die durchschnittlichen Durchlaufzeiten der Aufträge, während die Akzeptanz gegenüber einem derartigen Planungssystem sinkt. Denn die Pressenbediener werden Layoutvorschläge, die offensichtlich einen unnötig hohen Handlingsaufwand mit sich bringen, bald nicht mehr beachten!

Die folgende Punkte müssen bei der Behandlung des eben gezeigten Nestingproblems ebenfalls berücksichtigt werden.

Restriktionen und Sekundärziele

- Die Umhüllenden der Matrizen müssen parallel zum Trog und untereinander überschneidungsfrei im Trog platziert werden können.
- Lösungsgeglühte Platinen müssen wegen der Aushärtung innerhalb einer Woche umgeformt werden.
- f_{MinAusl} soll nicht unterschritten werden. Das Erreichen der Minimalauslastung kann nicht als Restriktion behandelt werden, weil man nicht garantieren kann, dass immer Lösungen gefunden werden, die dieser Anforderung genügen.
- Beschickungsregeln zur Schonung der Presse sollen wenn möglich eingehalten werden und können als Sekundärziele in die Zielfunktion einfließen. Regeln, die die Prozesssicherheit gewährleisten, müssen eingehalten werden, sind also Restriktionen für die Zulässigkeit eines Layouts.

2.4 Aufgabenbeschreibung

Den Anstoss zu dieser Arbeit gab der Wunsch, einerseits den Vorarbeiter und andererseits die Bediener der Fluidzellpresse bei ihren Aufgaben zu unterstützen. In erster Linie werden von einem rechnergestützten Planungssystem graphische Vorschläge für Pressenlayouts erwartet, welche durch den Vorarbeiter interaktiv am System generiert werden. Nicht genehme Vorschläge sollen verworfen und nach Änderung der Parameter neu berechnet werden können.

Durch konkrete Layoutvorschläge sollen v.a. erfahrene Mitarbeiter an der Presse entlastet werden, die bisher durch diese oft routinemässige Arbeit stark belastet sind. Dadurch könnten sie vermehrt freigestellt werden für anspruchsvollere Arbeiten wie z.B. das Durchführen von Probepressungen neuer Artikel (zur Bestimmung der Arbeitsgänge des Operationsplanes).

Es wurde auch die Idee aufgeworfen, die Chargierung bei der Wärmebehandlung basierend auf den Vorschlägen für die Presse ebenfalls rechnergestützt zu planen. Es würde allerdings genügen, die Chargen zusammenzustellen, d.h. es werden keine Layoutvorschläge erwartet.

Analyse

Die Kopplung der Umformung mit der Wärmebehandlung wird vom Vorarbeiter im Ansatz bereits angewendet, denn die Bestimmung der Aufträge für die nächsten Chargen basiert auf einer (rein gedanklich) vorgesehenen Reihenfolge der Aufträge an der Presse. Ein Problem dieser gekoppelten Planung bilden die Handarbeitsstationen, die oft zwischen Wärmebehandlung und Umformung durchgeführt werden müssen. Da die Startzeiten an diesen Stationen nicht festgelegt und die Bearbeitungszeiten nur geschätzt sind, sind auch die Endzeitpunkte nicht bekannt. Sonst auftretende Störfälle erschweren eine Einplanung über mehrere Stationen ebenfalls. Derartige Störfälle werden beispielsweise durch Maschinenausfälle verursacht, oder durch Matrizen, welche zum geplanten Zeitpunkt nicht verfügbar sind oder die während einer Pressung beschädigt wurden.

Bereits relativ einfache Varianten eines Nestingproblems gelten als NP-schwere Optimierungsprobleme, wie z.B. HEISTERMANN ausführt.

Eine planerische Kopplung der Arbeitsplätze macht zusätzlich zur Zuteilung des Problems einen übergeordneten Programmteil notwendig, welcher die Auftragsverwaltung und die Koordination zwischen den Arbeitsgängen übernimmt.

Aufgrund der eben aufgelisteten Erwartungen und Kommentare sind für FluidPlan zwei unterschiedliche Ausprägungen denkbar. Im ersten Fall wird lediglich die Kernaufgabe an der Presse gelöst, ohne rechnerische Kopplung mit der Wärmebehandlung („FluidPlan light“). Bei einem relativ kurzen Zeithorizont werden aus den an der Presse bereitliegenden Aufträgen Layouts zusammengestellt. Im zweiten Fall übernimmt FluidPlan die Werkstattsteuerung des gesamten Systems. Das heisst, es erfolgt erstens eine grobe Reihenfolgeplanung für die Arbeitsplätze ‚Wärmebehandlung‘, ‚Umformung‘ und ‚Zwischenbearbeitung‘, und zweitens wird für jeden dieser Arbeitsplätze eine Feinplanung durchgeführt, welche von der Grobplanung Ecktermine übernimmt. In diesem Fall müssen für die Arbeitsgänge an den Handarbeitsstationen genügend grosse Bearbeitungsdauern vorgesehen werden (die nach Inbetriebnahme des Programmes exakter bestimmt werden können). Die vorgeschlagene Variante, Öfen und Presse ohne die Zwischenbearbeitung zu planen, macht wenig Sinn, da die notwendigen Ecktermine für die Planung auf diese Weise nicht bekannt sind.

Fasst man den zweiten Fall (Werkstattsteuerung) ins Auge, sprechen mehrere Gründe für einen modularen Aufbau von FluidPlan:

- *Effektive Lösungsansätze:* Es ist kaum denkbar, mit einem einzigen Algorithmus alle Entscheidungsvariablen sämtlicher Teilprobleme gleichzeitig zu behandeln. Denn neben dem klassischen Reihenfolgeplanungsproblem für eine Werkstattfertigung muss u.a. auch das NP-schwere Nestingproblem gelöst werden. Werden die Teilprobleme einzeln behandelt, können darauf ausgelegte Methoden eingesetzt werden.
- *Effiziente schrittweise Entwicklung:* Es können drei Entwicklungsstufen definiert werden: Erstens „FluidPlan light“, zweitens die ansonsten verworfene Variante Öfen/Presse ohne Zwischenbearbeitung und drittens die Werkstattsteuerung als Endstufe. Für die zweite und dritte Stufe können dann die Module der bestehenden Stufe(n) mit kleinem Änderungsaufwand übernommen werden.
- *Flexibilität:* Die Modularisierung erleichtert eine spätere Weiterentwicklung von FluidPlan. Diese kann ausgelöst werden durch erkannte Schwachstellen einer Version oder durch veränderte Bedingungen in der Fertigung (neue Maschinen oder Verfahren). Ist die Aufgabenteilung und die Kommunikation zwischen den Modulen einmal definiert, können die einzelnen Module unabhängig voneinander verändert werden.
- *Übersichtlichkeit:* Parallel zur Aufgabenhierarchie der Module existiert eine Aufgabenhierarchie bei den Mitarbeitern des Systems. Wird pro Hierarchiestufe und pro Aufgabenbereich ein Modul entwickelt, kann pro Modul ein spezifisch ausgelegtes Interface entworfen werden.
- *Autonomie der Arbeitsplätze:* Die einzelnen Module können auf verschiedenen Rechnern laufen, die über ein lokales Netzwerk miteinander verbunden sind. So haben die Mitarbeiter eines Arbeitsplatzes die Möglichkeit, ihren Teilbereich selbst zu organisieren und zweitens können Störungen lokal behoben werden, ohne andere Arbeitsplätze dadurch zu behelligen.

Im Weiteren ist zu überlegen, welche Entscheidungskompetenzen dem zu entwerfenden Programm übergeben werden sollen. Dies betrifft Fragen wie z.B. die Anzahl eingesetzter Tröge an der Presse oder die Wahl der Teilefamilien, die abgearbeitet werden sollen.

Folgerungen

Aus den obenstehenden Überlegungen wurden folgende Teilaufgaben abgeleitet:

1. Entwurf eines Konzeptes für die Steuerung des Systems $S(0)$, d.h. für den oben beschriebenen, übergeordneten Programmteil zur Reihenfolgeplanung. Es muss u.a. eine Aufgabenhierarchie entworfen werden, die definiert, auf welcher Systemebene welche Aufgaben gelöst werden. Zudem müssen im Gesamtkonzept die erwähnten Entscheidungskompetenzen geregelt werden.
2. Entwurf eines Lösungskonzeptes für das Nestingproblem an der Fluidzellpresse, wie es in Abschnitt 2.3 vorgestellt wurde. Dieses Konzept muss graphische Layoutvorschläge generieren, die interaktiv abgelehnt oder akzeptiert werden können. Es sollen Vorschläge für ca. einen Arbeitstag berechnet werden können.
3. Entwurf eines Lösungskonzeptes bzw. eines Ansatzes für das Kapazitätsproblem bei den Wärmebehandlungsöfen.
4. Schrittweise Implementation des Konzeptes.

3 Der Umformprozess in einer Fluidzellpresse

Nach einer Einführung in die Fluidzelltechnologie werden die wichtigsten Kriterien aufgezeigt, welche das Geschehen an einer Fluidzellpresse beeinflussen. Zentral sind dabei die in Abschnitt 2.3.1 eingeführten Beschickungsregeln, welche die Zusammensetzung der Teilesets sowie die Layouts wesentlich prägen. Für diese nicht formalisierten Regeln werden allgemeine Anweisungen formuliert, die für alle Layouts Gültigkeit besitzen, und ein Modell zur Behandlung der Tabuzonen vorgestellt.

Gliederung des Kapitels

In Abschnitt 3.1 wird zuerst die Fluidzelltechnologie eingeordnet, dann werden Wirkungsweise und Einsatzort, inklusive Vor- und Nachteile des Verfahrens, vorgestellt. Abschnitt 3.2 zeigt auf, von welchen Kriterien Layoutvorschläge direkt oder indirekt beeinflusst werden. Diese Einflüsse werden bis zu den allgemeinen Optimierungszielen einer Fertigung (Kosten, Termine, Qualität) gezeigt. In Abschnitt 3.2.5 folgt eine Zusammenfassung der gemachten Aussagen. Zentrale Einflusskriterien, die analysiert werden müssen, bilden die Beschickungsregeln, da sie an der Presse interpretiert werden müssen. Sie werden in Abschnitt 3.3 aufgezeigt.

3.1 Fluidzelltechnologie

3.1.1 Einordnung des Verfahrens

Die Fluidzelltechnik ist ein Sonderverfahren des Tiefziehens. In Abbildung 4-1 ist die Gliederung der Tiefziehverfahren (nach DIN 8584) auszugsweise abgebildet.

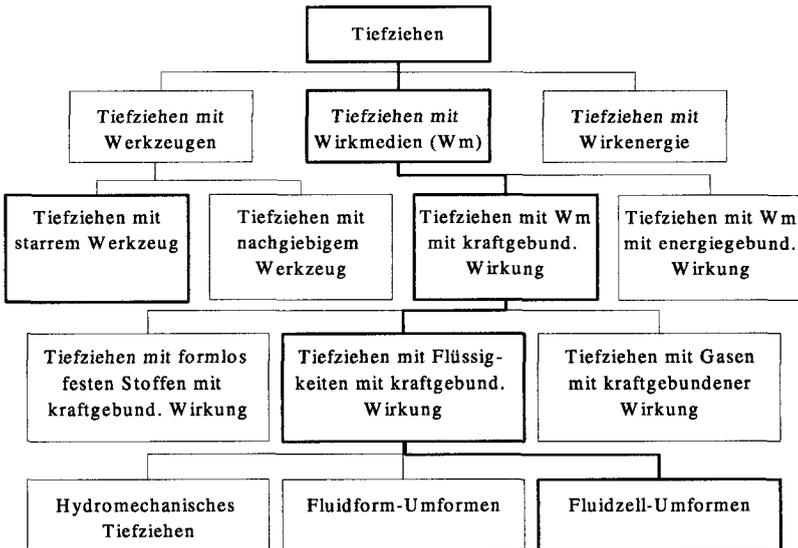


Abbildung 3-1: Gliederung der Tiefziehverfahren (Auszug)

3.1.2 Wirkungsweise und Einsatzgebiet der Fluidzelltechnik

Vor- und Nachteile sowie Einsatzgebiete der Fluidzelltechnik lassen sich im direkten Vergleich mit dem weit verbreiteten konventionellen Tiefziehen mit starren Werkzeugen aufzeigen. Beim konventionellen Tiefziehen, das in Abbildung 3-2 schematisch abgebildet ist, besteht der Werkzeugsatz typischerweise aus einer Matrize, einem Niederhalter (oder Blechhalter) sowie aus einem Stempel. Das Engineering und die Fertigung solcher Werkzeuge ist relativ teuer, da diese starren Teile präzise aufeinander abgestimmt werden müssen. Zudem fließen von der Presse, in welche die Werkzeuge eingebaut werden sollen, geometrische und physikalische Randbedingungen ein. Im Weiteren sind die erreichbaren Ziehverhältnisse im Vergleich zu Tiefziehverfahren mit Wirkmedien relativ niedrig, was mehrere Umformstufen nötig macht.

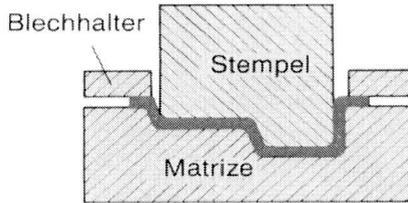


Abbildung 3-2: Konventionelles Tiefziehen (Bild: ABB)

Die Umrüstzeiten bei solchen Tiefziehpressen sind aufgrund der notwendigen Werkzeugabstimmung entsprechend hoch, doch sind die Taktzeiten während der Fertigung relativ klein. So zählt das konventionelle Tiefziehen zu den wichtigsten Verfahren zur Massenfertigung von metallischen und nichtmetallischen Hohlkörpern.

Als Alternativen bieten sich das Tiefziehen mit nachgiebigem Werkzeug (Gummikissenpresse) oder verschiedene Verfahren des Tiefziehens mit Wirkmedien an. Dazu gehört auch die Fluidzelltechnik, deren Prinzip bereits in Abbildung 2-5 gezeigt ist. Als Vorteile dieser Verfahren gegenüber dem konventionellen Tiefziehen nennen LANGE sowie HABERSTOCK & BERG:

- Kleinere Anzahl Umformstufen aufgrund grösserer möglicher Ziehtiefen
- Verarbeitung schwer umformbarer Teile (z.B. mit Hinterschneidungen)
- Hohe Form- und Massgenauigkeit
- Niedriger Eigenspannungszustand
- Umformen verschiedener Werkstoffe mit demselben Werkzeug, insbesondere auch umformen beschichteter Werkstoffe
- Unterschiedliche Materialdicken mit demselben Werkzeug umformbar
- Geringere manuelle Nacharbeit
- Kurze Lieferzeiten realisierbar

Neben diesen Vorteilen vereinfacht und verbilligt sich der Werkzeugbau bei der Fluidzelltechnik, denn das Werkzeug besteht lediglich aus einer Matrize, die ohne Fixation auf eine Metallplatte gelegt werden kann. Erstens entfällt dadurch die Anpassung an andere Werkzeugbestandteile und als Material können kostengünstige Werkstoffe wie Aluminium oder Hartholz eingesetzt werden. Zweitens sind, weil die Matrizen lediglich auf den Trog gelegt werden müssen, die Umrüstzeiten beim Werkzeugwechsel relativ gering. Da der Druck von einigen hundert bar nur langsam aufgebaut werden kann, liegt die Taktzeit allerdings bei mehreren Minuten.

Fluidzellpressen werden deshalb eingesetzt, wenn viele verschiedene Blechteile in kleinen Losen und mit teilweise komplexen Formen umgeformt werden sollen, und wenn zudem hohe Anforderungen an die Teile bestehen. Diese Situation trifft insbesondere für den Flugzeugbau zu, da Rumpf und Flügel der Flugzeuge aus vielen, vor allem kleinen Blechteilen bestehen. Im Weiteren werden Fluidzellpressen beim Prototypenbau in der Automobilindustrie eingesetzt. In HJELMQWIST sind verschiedene Pressentypen sowie deren häufigste Anwendungsgebiete aufgeführt.

Als Nachteile des Verfahrens führen HABERSTOCK & BERG u.a. auf:

- Kleine Radien nicht immer realisierbar
- Prägestellen nicht möglich
- Preis der Anlage sehr hoch

3.2 Einflusskriterien

Die an der Presse getroffenen Entscheidungen zur Wahl der Teilefamilie, der Aufträge und der Anordnung der Werkzeuge, basieren erstens auf Vorgaben aus der Planungsebene (Teile- und Werkzeugkonstruktion, Arbeitsvorbereitung), zweitens auf den aktuellen Betriebsdaten und drittens auf den Beschickungsregeln (vgl. Abbildung 3-3).

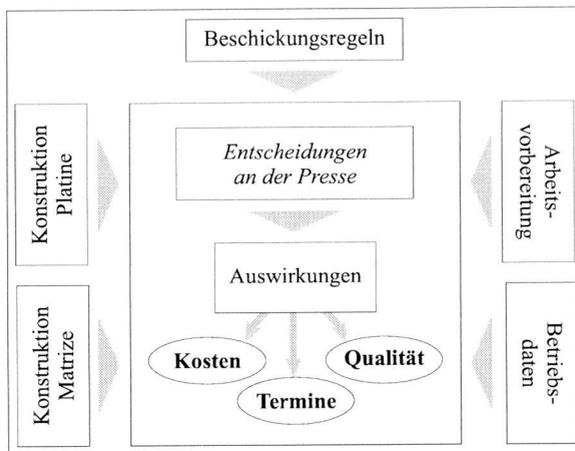


Abbildung 3-3: Ursprung der Einflussgrößen bei einer Fluidzellpresse

Für uns sind v.a. diejenigen Faktoren relevant, welche die Entscheidungen an der Presse beeinflussen, bzw. durch die Entscheidungen an der Presse beeinflusst werden. Sie werden in den folgenden Abschnitten beschrieben. Überdies wird gezeigt, welche Kriterien des Wirksystems Einfluss haben auf die drei klassischen betrieblichen Optimierungsziele, also auf das Erreichen der Ziele tiefe *Kosten*, kurze Lieferzeiten (*Termine*) und erforderliche *Qualität*. Abbildung 3-4 zeigt die Einflussmatrix der Hauptgruppen der Einflusskriterien.

Zuerst werden die oben genannten Optimierungsziele für die Fluidzellpresse formuliert. Dann werden die Kriterien der einzelnen Hauptgruppen in der Reihenfolge gemäss Abbildung 3-4, inklusive ihrer qualitativen Verknüpfungen untereinander, vorgestellt. Werden dabei Kriterien er-

wähnt, die erst später erläutert werden, wird mit dem entsprechenden Buchstaben (A-G) auf die Hauptgruppe verwiesen. In Abschnitt 3.2.5 folgt eine Zusammenfassung sämtlicher Einflüsse, welche in Abbildung 3-7 dargestellt ist.

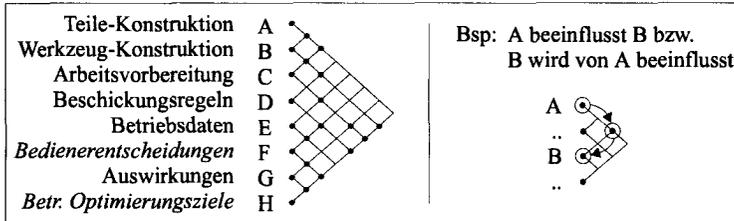


Abbildung 3-4: Einflussmatrix der Kriteriengruppen

3.2.1 Betriebliche Optimierungsziele

In diesem Abschnitt wird gezeigt, von welchen der später ausgeführten Kriterien die betrieblichen Optimierungsziele vor allem abhängen.

Kosten

Zwei Kostenarten werden hier unterschieden: Die *Betriebskosten* hängen von der Anzahl Pressungen (G) ab, die für die Fertigung einer bestimmten Anzahl von Aufträgen benötigt wird. Die Benutzung der Presse und die Lohnkosten der Bediener müssen hierbei berücksichtigt werden.

Die *Unterhaltskosten* für die Presse machen den zweiten Teil der Kosten aus. Der Grossteil dieser Kosten wird verursacht, wenn die Druckmembrane, welche das Ölbad abdichtet, reißt. Bis die Membrane ersetzt, die Anlage gereinigt und mit neuem Öl aufgefüllt ist, vergehen einige Tage, was Kosten von mehreren zehntausend Franken verursacht. Membranenrisse sind normale Abnützungserscheinungen und können nicht verhindert werden. Durch das Einhalten der Beschickungsregeln (D) aber kann die Lebensdauer der Membran signifikant erhöht werden (von ca. 6 auf ca. 12 Monate).

Termine: Tiefe Durchlaufzeiten und Termineinhaltung

Das Kriterium hängt v.a. ab von den Rüstzeiten (G), den Anzahl Pressungen (G) sowie von der Prozesssicherheit, denn schlechte Pressungen bzw. der Ausfall der Anlage bedeuten Zeitverzögerungen für die Aufträge. Eine weitere wichtige Einflussgrösse ist die Einteilung der Teilefamilien, denn von ihr hängt die Auslastung und somit die Anzahl notwendige Pressungen ab.

Qualität

Die Qualität produzierter Artikel ist dann gut, wenn es keinen Ausschuss gibt, wenn also eine hohe Prozesssicherheit vorhanden ist. Diese ist abhängig vom Druck (C), von der Schmierung zwischen Platine und Matrice (C) sowie vom Sicherheitsabstand zwischen den Matrizen (D).

Die drei aufgeführten Ziele beeinflussen sich auch untereinander. Beispielsweise steigen bei sinkender Qualität durch den anfallenden Ausschuss die Materialkosten. Im Rahmen dieser Arbeit werden derartige Zusammenhänge nicht betrachtet. Wir konzentrieren uns auf diejenigen Aspekte von Einflussgrössen, welche die Entscheidungen an der Presse beeinflussen und auf deren Auswirkungen.

3.2.2 Einflüsse aus der Planungsebene auf Entscheidungen an der Presse

Teilekonstruktion (A)

In der Teilekonstruktion werden u.a. *Werkstoff* und *Endform* einer Platine bestimmt. Durch die Endform sind implizit auch die *Dicke* sowie der *kleinste Radius* an der Platine gegeben. In Abbildung 3-5 zeigt sich, dass mit diesen Vorgaben der minimale Umformdruck (C) gegeben ist.

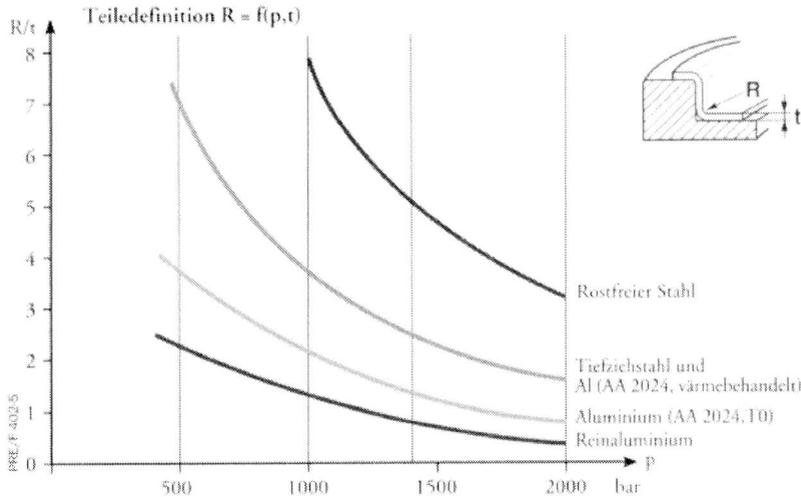


Abbildung 3-5: Verhältnis von Blechdicke, Radius, Werkstoff und Umformdruck (Bild: ABB)

Werkzeugkonstruktion (B)

Aus der Endform ist der *Zuschnitt* der Platine ableitbar. Den Grad der Umformung des Zuschnitts zur Endform nennt man *Umformgrad* einer Platine, welcher in (C) bei der Erstellung des Operationsplanes wieder benötigt wird. Hauptaufgabe der Werkzeugkonstruktion ist die Definition der *Matrizenform*, die sich im Wesentlichen aus dem Zuschnitt der Platine und ihrer Endform ableiten lässt. Wichtige Teilinformationen der Form sind der *Grundriss* sowie die *Randhöhe* der Matrize zur Bestimmung der Umhüllenden (D). Ebenfalls benötigt wird später die *Gesamthöhe* von Matrize und Platine.

Arbeitsvorbereitung (C): Einteilung der Teilfamilien

In der Arbeitsvorbereitung wird der Operationsplan eines Artikels erstellt und sämtliche Arbeitsanweisungen (Maschineneinstellungen etc.) definiert. Der in Abbildung 3-5 gezeigte Zusammenhang legt einen Minimaldruck fest, der bei einem hohen Umformgrad noch erhöht werden muss; auf diese Weise wird der *Umformdruck* festgelegt. Werkstoff und Endform der Platine, Matrizenform und Umformdruck sind Einflusskriterien der Reibung zwischen Platine und Matrize. Ist die Reibung zu gross, kann sie durch Schmierung herabgesetzt werden. *Reibung/Schmierung* bilden ein Kriterienpaar, das voneinander abhängt.

Aus dem notwendigen Umformgrad und den Reibungsverhältnissen lässt sich bestimmen, welches *Wärmebehandlungsverfahren* eingesetzt werden muss (vgl. die Erklärung zu den Verfahren in Abschnitt 2.2.3). Dicke und Werkstoff bestimmen Zeit und Temperatur des Verfahrens.

Die *Anzahl und Einteilung der Teilefamilien* für die Presse wird durch zwei Faktoren bestimmt. Die erste und steuerbare Entscheidung erfolgt durch die Festlegung der zur Wahl stehenden Drücke, mit denen das gesamte Teilespektrum umgeformt werden muss. Der Grossteil der Artikel wird dabei mit einem Normdruck umgeformt. Dieser Druck ist ein Kompromiss zwischen dem Zwang nach einem Minimaldruck (vgl. Abbildung 3-5) und dem Wunsch nach einem möglichst tiefen Druck: Erst ab einem bestimmten Wert ist ein formgenaues Umformen sichergestellt, eine Erhöhung des Wertes zeigt keine signifikante Verbesserung der Qualität mehr. Auf der anderen Seite steigt mit dem Druck erstens die Prozesszeit (linearer Druckaufbau in der Presse), zweitens erhöht sich die Membranenbelastung und drittens müssen die Matrizen robuster gebaut werden, um Beschädigungen zu vermeiden. In Anbetracht dieser Argumente wäre eine möglichst feine Einteilung, d.h. eine Vielzahl von zur Auswahl stehender Drücke, sinnvoll. Allerdings ist dann bei einem relativ kleinen Produktionsprogramm die Trogauslastung zu klein. Der Normdruck sowie die Spezialdrücke basieren auf Erfahrungswerten, da die theoretische Ermittlung der Anzahl und Einteilung der Umformdrücke sehr aufwendig wäre.

Teilefamilien können zudem nach einem zweiten Gesichtspunkt zusammengestellt werden, indem die Maximalhöhe der Matrize - inklusive Dicke der Platine - betrachtet wird. Diese ist bei den meisten Artikeln niedriger als die im Trog zur Verfügung stehende Einsatzhöhe. Deshalb wird wenn möglich eine sogenannte Zwischenplatte in den Trog gelegt, um diesen voll auszufüllen. Die Zwischenplatte senkt die Belastung der Gummimembrane deutlich.¹

3.2.3 Beschickungsregeln und aktuelle Betriebsdaten

Beschickungsregeln (D)

Auf die in Abschnitt 2.3.1 eingeführten Beschickungsregeln wird in Abschnitt 3.3 näher eingegangen. Sie werden hier vorgestellt, damit die Einflusskriterien in diesem Abschnitt komplett aufgezählt sind. Bereits bekannt sind die *Tabuzonen* einer Matrize, die v.a. aus der Matrizenform abgeleitet werden können. Weitere Regeln sollen die Membranenbelastung reduzieren und betreffen insbesondere die hohen Matrizen. Es sind dies die Beschränkung der *Anzahl hoher Matrizen* in einem Layout, die *Nachbarschaften hoher Matrizen* sowie die Anweisung, *hohe Matrizen in die Mitte* des Troges zu legen.

Mit Hilfe der Randhöhe einer Matrize wird im Weiteren ein sog. *Sicherheitsabstand* definiert, aus dem in Kombination mit dem Grundriss der Matrize die *Umhüllende* berechnet wird.

Betriebsdaten (E)

Hier interessieren in erster Linie die im Pufferlager der Presse bereit liegenden *Aufträge*, insbesondere ihre Priorität und Losgrösse. Eine weitere Teilmaterie ist der spätestmögliche *Termin* für die Fertigung, resultierend aus dem Lösungsglöhnen. Aus den vorliegenden Aufträgen und der getroffenen Einteilung der Teilefamilien ergibt sich die *aktuelle Grösse der Teilefamilien*.

1. Auch im betrachteten Betrieb wird eine Zwischenplatte eingesetzt. Für die Planung ist es nicht entscheidend, ob sich zwei Teilefamilien aufgrund des Druckes oder aufgrund der Zwischenplatte unterscheiden.

3.2.4 Entscheidungen und Auswirkungen

Bedienerentscheidungen (F)

Die erste Entscheidung, die der Bediener - bzw. der Vorarbeiter - trifft, ist die *Wahl der Teilefamilie*. Sie basiert auf den oben aufgeführten Betriebsdaten (Wahl und aktuelle Grösse einer Teilefamilie sind strenggenommen ein Kriterienpaar, denn eine jetzt getroffene Wahl beeinflusst die nachher vorhandene Grösse der Teilefamilien). Aufgrund der auf diese Weise ausgewählten Aufträge und unter Einbezug der Beschickungsregeln wird ein *Layoutvorschlag* generiert. Werden mehrere (sich folgende) Layouts akzeptiert, entsteht eine *Layoutfolge*. Eine wichtige Teilinformation der Layoutfolgen ist z.B. die *relative Position der Matrizen* im Trog, die von der Bewertung der Nachbarschaften abhängt. Die *absolute Position der Matrizen* hingegen hängt von den Tabuzonen und von der Anweisung ab, hohe Matrizen in die Mitte des Troges zu legen.

Auswirkungen (G)

Die Layoutfolgen sowie die dabei notwendige Schmierung gewisser Matrizen haben Einfluss auf den *Handlingsaufwand* und sind somit ein Teil der *Rüstzeiten*. Für einen Layoutvorschlag kann bezüglich der zur Verfügung stehenden Gesamtfläche des Troges eine *Auslastung* berechnet werden. Aus den Auslastungen einer Layoutfolge lässt sich schliesslich die *Anzahl Pressungen* ableiten, die für die Abarbeitung einer gegebenen Anzahl Aufträge notwendig ist. Von dieser Anzahl hängen u.a. die Betriebskosten ab, die als Teil der betriebliche Optimierungsziele in Abschnitt 3.2.1 genannt wurden.

3.2.5 Kommentar und Übersicht

Bei der Konstruktion der Blechteile (A) stehen Form, Flugeigenschaften und Gewicht eines Flugzeugs im Vordergrund, fertigungstechnische Aspekte sind zweitrangig. Die aus (A) stammenden Kriterien haben auf Entscheide an der Presse nur indirekten Einfluss, indem sie die Rahmenbedingungen für die in Abbildung 3-6 aufgeführten Kriterien der Werkzeugkonstruktion (B) und der Arbeitsvorbereitung (C) setzen.

| | | |
|------|-------------------------------|--|
| (A:) | Werkstoff | |
| | Endform Platine | |
| | Platinendicke | |
| | kleinster Radius | |
| (B:) | Zuschnitt Platine | |
| | Umformgrad | |
| | Matrizenform | |
| | Grundform Matrize (Grundriss) | |
| | Randhöhe Matrize | |
| | Gesamthöhe Matrize + Platine | |
| (C:) | Druck | |
| | Reibung / Schmierung | |
| | Wärmebehandlungsverfahren | |
| | TF: Anzahl, Einteilung | |

Abbildung 3-6: Einfluss der Teilekonstruktion

Zentraler Output von (B) ist die Matrizenform; zusammen mit dem Umformgrad prägt sie praktisch alle folgenden Kriterien aus der Arbeitsvorbereitung, und die Beschickungsregeln beziehen sich i.a. auf die vorliegenden Formen (vgl. Abbildung 3-7).

Die Einteilung der Teilefamilien (C) hat direkten Einfluss auf die Lebensdauer der Membrane sowie auf die Durchlaufzeiten und ist somit für die Wirtschaftlichkeit einer Blechstrasse von zentraler Bedeutung. Trotzdem wird sie nur unzureichend hinterfragt. Eine Optimierung dieser Einteilung ist allerdings nur mit einer relativ aufwendigen Simulation der Materialflüsse möglich (vgl. auch den zweitletzten Absatz in Abschnitt 3.2.2). Aus Abbildung 3-7 ist ersichtlich, dass die Kriterien aus (C) direkten Einfluss haben auf die Betriebsdaten (E) einerseits und auf die betrieblichen Optimierungsziele (H) andererseits (v.a. auf die Prozesssicherheit).

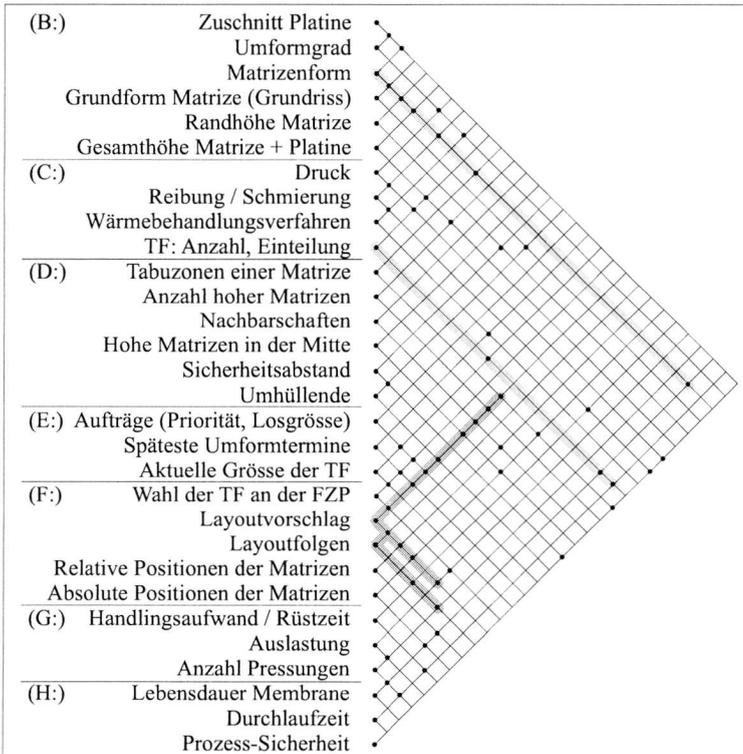


Abbildung 3-7: Einflussmatrix (alle Kriterien ausser Werkzeugkonstruktion)

Die Entscheidungen an der Presse (F) werden direkt von den Beschickungsregeln (D) und den Betriebsdaten (E) beeinflusst. Dabei sind die im nächsten Abschnitt behandelten Beschickungsregeln diejenigen Einflussgrössen, deren Anwendung in den Händen der Mitarbeiter an der Presse liegt. Wir werden sehen, dass die Regeln v.a. aus Empfehlungen mit einigem Spielraum zusammengesetzt sind, nur einige davon sind Restriktionen.

3.3 Beschickungsregeln

Die Beschickungsregeln werden in allgemeine und in spezifische Regeln eingeteilt. Allgemeine Regeln können für alle Matrizen definiert werden und beziehen sich stets auf die Dimensionen der Matrizen. Spezifische Regeln betreffen „Spezialbehandlungen“, die für einzelne Matrizen gelten und sind meist im Operationsplan des betreffenden Artikels vermerkt. In dieser Arbeit werden im Rahmen der spezifischen Regeln nur die Tabuzonen betrachtet. Eine weitere spezifische Regel könnte z.B. aus der Schmierung der Matrize an der Kontaktfläche zur Platine abgeleitet werden. Die Schmierung dient aber lediglich der Prozesssicherheit und hat keinen direkten Einfluss auf das zu verbessernde Planungsergebnis.

3.3.1 Allgemeine Regeln

Sicherheitsabstand

Oft ragen Platinen über den Rand der Matrize hinaus. Bei der Umformung dürfen diese Platinen nicht mit anderen Matrizen oder mit dem Trogrand kollidieren. An hohen Rändern, über die keine Platine hinausragt, kann mit Hilfe von Gummistücken eine Treppe gebildet werden. Abbildung 3-8 veranschaulicht diese Fälle. Bei der links gezeichneten Matrize erübrigt sich die Treppengbildung, weil der freie Rand abgeflacht ist. Die Treppe dient dazu, die Dehnung der Gummimembrane zu reduzieren.

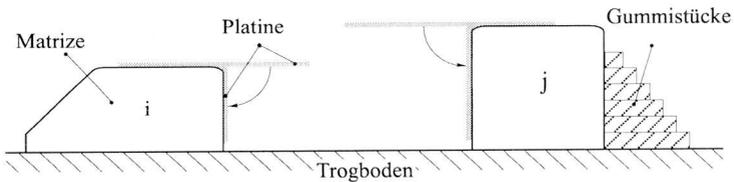


Abbildung 3-8: Mögliche Ausprägungen von Matrizen

Zwischen den Matrizen untereinander sowie zwischen den Matrizen und dem Rand muss deshalb eine gewisse Distanz eingehalten werden, die pro Matrize im Sicherheitsabstand s_a festgehalten wird. In Abbildung 3-9 sind die massgebenden Grössen eingezeichnet

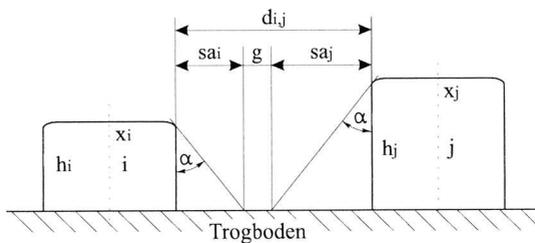


Abbildung 3-9: Sicherheitsabstand zwischen zwei Matrizen

sa_i ist der Sicherheitsabstand einer Matrize M_i , der sich aus einem gegebenen Winkel α und der Randhöhe h_i ergibt, g ist ein Grundabstand, der zwischen allen Matrizen eingehalten werden soll, und $d_{i,j}$ ist der Abstand zwischen den beiden Matrizen M_i, M_j . Es soll gelten:

$$d_{i,j} \geq sa_i + sa_j + g \tag{3-1}$$

Um diese Restriktion zu erfüllen, wird jeder Matrize M_i eine *Umhüllende* u_i, v_i zugeordnet. u_i, v_i ergeben sich aus dem kleinsten umhüllenden Rechteck um M_i , aus dem Sicherheitsabstand sa_i sowie aus g . Ungleichung [3-1] führt dann zur Restriktion, dass sich die Umhüllenden zwar berühren, nicht aber überschneiden dürfen. Für u_i, v_i gilt:

$$\begin{aligned} u_i &= x_i + 2 \times sa_i + g \\ v_i &= y_i + 2 \times sa_i + g \end{aligned} \tag{3-2}$$

Im Rahmen dieser Arbeit beschränken wird uns darauf, für die Berechnung von u_i, v_i die in der Praxis üblichen Werte $g = 0$ und $\alpha = 45^\circ$ einzusetzen. Somit erhält man für sa_i :

$$sa_i = h_i \times \tan \alpha = h_i \tag{3-3}$$

Wir sind uns bewusst, dass durch eine exaktere Bestimmung der Parameter g und α die effektive Trogauslastung um einige Prozent gesteigert werden könnte, ohne die Prozesssicherheit zu gefährden (die effektive Auslastung ist die Summe der Matrizenflächen, im Gegensatz zur Summe der Umhüllenden aller Matrizen eines Layout). Bei den kleinen Matrizen, welche die Mehrheit der Matrizen bilden, ist die Fläche der Umhüllenden z.T. doppelt so gross wie die effektive Matrizenfläche. Minimalwerte von g und α könnten z.B. gefunden werden mit Hilfe einer Simulation des Prozesses, welche das Verständnis fördern würde, was in der Presse während der Umformung genau passiert. Eine mögliche Fragestellung ist, ob die Matrizen zu Beginn des Prozesses auf dem Trog bewegt werden (durch die Membrane, die sich über die Matrizen legt) und dadurch kollidieren können. Mit einer genaueren Modellierung der Matrizen könnte die Fläche der Umhüllenden ebenfalls verringert werden. Erstens ist die Randhöhe einer Matrize i.a. nicht auf allen vier Seiten gleich hoch, was nicht festgehalten wird. Zweitens kann $d_{i,j}=0$ gesetzt werden, d.h. M_i und M_j können aneinander geschoben werden, wenn folgende Bedingungen erfüllt sind: M_i und M_j haben auf je einer Seite, über die die Platinen *nicht* ragen, ähnliche Randhöhen und ähnliche Kantenlängen. Abbildung 3-10 veranschaulicht diesen Fall:

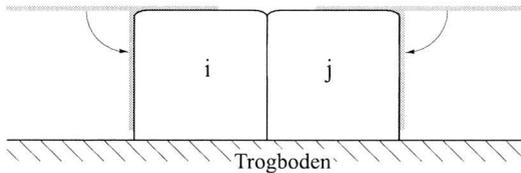


Abbildung 3-10: Zusammengeschobene Matrizen ($d_{i,j} = 0$)

Lage der Matrizen im Trog

Bei der Umformung dehnt sich die Gummimembrane über die Matrizen. Nach einigen Tausend Zyklen wird die Membrane aufgrund dieser Belastung reißen. Abbildung 3-11 zeigt die Membrane in zwei Stadien der Umformung (Platinen nicht eingezeichnet). In der Abbildung ist der Druck in der Fluidzelle gerade so hoch, dass die Membrane auf den Matrizen i, j aufliegt. Dabei

wird die Annahme getroffen, dass die Membrane bis jetzt gleichmässig über die ganze Fläche gedehnt wurde. Wir nehmen weiter an, dass sich die Membrane von diesem Stadium an nur noch im Bereich zwischen den Matrizen auf die *Endlänge* a_2 dehnt (unteres Bild). Das heisst, dass aufgrund der hohen Reibung keine Relativbewegung zwischen Matrizenoberseite und Membrane besteht.

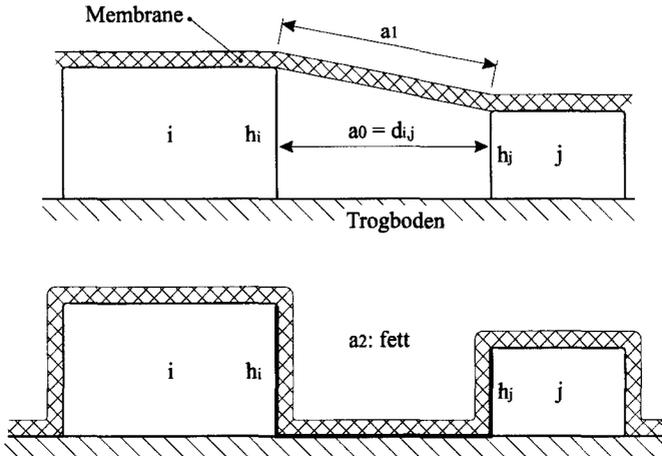


Abbildung 3-11: Dehnung der Membrane zwischen zwei Matrizen i, j

Für a_0 , a_1 und a_2 gilt unter Einbezug der Gleichungen [3-1] und [3-3]:

$$a_0 = d_{i,j} = s_{a_i} + s_{a_j} = h_i + h_j \quad [3-4]$$

Die Längen a_1 , a_0 und $(h_i - h_j)$ bilden ein rechtwinkliges Dreieck:

$$a_1 = \sqrt{a_0^2 + (h_i - h_j)^2} = \sqrt{(h_i + h_j)^2 + (h_i - h_j)^2} = \sqrt{2 \cdot (h_i^2 + h_j^2)} \quad [3-5]$$

Und für a_2 erhält man:

$$a_2 = a_0 + h_i + h_j = 2 \cdot (h_i + h_j) \quad [3-6]$$

Die Membrane wird bei einer Pressung gemäss Abbildung 3-11 von der Länge a_1 auf die Länge a_2 gedehnt. In Tabelle 3-1 ist das Dehnungsverhältnis dieser beiden Grössen angegeben. Dabei ist h_i konstant 100 mm, was einer relativ hohen Matrize entspricht; h_j wandert von 10 bis 100 mm (bei einer durchschnittlichen Matrizenhöhe von 30 mm). Damit lassen sich a_0 , a_1 und a_2 sowie das erwähnte Dehnungsverhältnis berechnen.

Werden die Sicherheitsabstände eingehalten, fällt die unterschiedliche Dehnung der Membrane in Abhängigkeit der Matrizenhöhe auf den ersten Blick gering aus; die Differenz zwischen Fall 3 und Fall 10 aus Tabelle 3-1 beträgt knapp 24 %. Unter der Voraussetzung, dass sich die Lebensdauer der Membrane umgekehrt proportional zur Belastung verhält, ist diese Differenz jedoch sehr gross! Wir postulieren deshalb folgende allgemeine Beschickungsregeln:

- Die vorhandenen hohen Matrizen sollen gleichmässig auf die Layouts verteilt werden, damit wird auch die erhöhte Membranenbelastung verteilt.
- Die einem Layout zugewiesenen hohen Matrizen sind derart auf der Trogfläche zu plazieren, dass dazwischen flache Matrizen liegen (relative Position der Matrizen)
- Da die Membranenbelastung am Trogrand am grössten ist, gilt es dort keine hohen Matrizen zu plazieren (absolute Position der Matrizen).

| Nr | h_i | h_j | a_0 | a_1 | a_2 | a_1/a_2 |
|----|-------|-------|-------|-------|-------|-----------|
| 1 | 100 | 10 | 110 | 142.1 | 220 | 154.8 % |
| 2 | 100 | 20 | 120 | 144.2 | 240 | 166.4 % |
| 3 | 100 | 30 | 130 | 147.6 | 260 | 176.1 % |
| 4 | 100 | 40 | 140 | 152.3 | 280 | 183.8 % |
| 5 | 100 | 50 | 150 | 158.1 | 300 | 189.7 % |
| 6 | 100 | 60 | 160 | 164.9 | 320 | 194.0 % |
| 7 | 100 | 70 | 170 | 172.6 | 340 | 197.0 % |
| 8 | 100 | 80 | 180 | 181.1 | 360 | 198.8 % |
| 9 | 100 | 90 | 190 | 190.3 | 380 | 199.7 % |
| 10 | 100 | 100 | 200 | 200 | 400 | 200.0 % |

Tabelle 3-1: Dehnungsverhältnisse bei unterschiedlichen Matrizenhöhen

3.3.2 Spezifische Regeln: Tabuzonen

Für einige Matrizen (< 3 Prozent) existieren Bereiche im Trog, in welche sie aus Gründen der Prozesssicherheit nicht gelegt werden dürfen. Um diese Fälle beim Nesting abzudecken, werden sogenannte *Tabuzonen* eingeführt, die für die Modellierung dieser „verbotenen Zonen“ benutzt werden. Hierfür wird ein Trog in neun gleich grosse Felder eingeteilt, die in Abbildung 3-12 abgebildet sind.

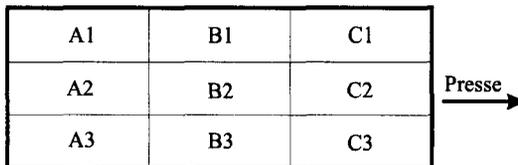


Abbildung 3-12: Tabufelder eines Troges

Nach Überprüfung der vorkommenden ‚verbotenen Zonen‘ hat sich gezeigt, dass sechs der insgesamt $2^3 - 1 = 511$ möglichen Kombinationen ausreichen, um die Regeln einzuhalten. Diese sechs Varianten V1 bis V6 sind in Abbildung 3-13 zusammengestellt.

Die Variante V1 z.B. ist aus folgendem Grund aufgeführt: Beim Einfahren des Trogs in die Presse wird eine Schutzmembrane über den Trog abgerollt, so dass die zweite Membrane, welche das Ölbad abdichtet, nicht direkt auf die Platinen zu liegen kommt (beide Membranen sind aus Gummi). Liegt eine relativ hohe Matrize am Trogrand in Richtung Presse, so kann es passieren, dass die Schutzmembrane gehoben und somit zwischen Presse und Trog eingeklemmt wird. Analog dazu dürfen andere Matrizen nicht am hinteren Trogrand liegen (Variante V2).

Die Varianten V3, V4 und V6 ergeben sich daraus, dass gewisse Matrizen nicht am Längs- oder Querrand des Troges liegen dürfen, da sonst die Prozesssicherheit nicht gewährleistet ist. Andere Matrizen sind zu schwer oder zu unhandlich, um sie in die Mitte legen zu können, weshalb auch Variante V5 für FluidPlan eine Tabuzone ist.

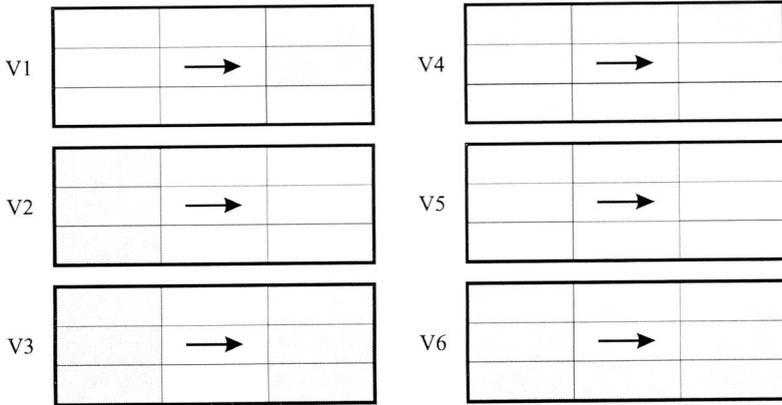


Abbildung 3-13: Mögliche Tabuzonen (grau)

Die Tabuzonen werden nicht als strikte Restriktionen aufgefasst. Eine Überlappung einer Matrize und ihrer Tabuzone ist bis zu einem gewissen Grad erlaubt, allerdings nicht erwünscht (unzulässig ist sicher die Situation, dass eine Matrize vollständig in ihrer Tabuzone liegt). In Abschnitt 6.6 wird gezeigt, wie die Tabuzonen im Zusammenhang mit dem Nestingproblem benutzt werden.

4 Konzept für eine modulare Werkstattsteuerung

Zu Beginn dieses Kapitels wird der Teilbereich der Blechteilefertigung, der das betrachtete System bildet, modelliert. Im zweiten Abschnitt wird der Begriff ‚Werkstattsteuerung‘ diskutiert und gegen andere Begriffe abgegrenzt, die alternativ dafür verwendet werden. Anschliessend wird ein Konzept für die modulare Werkstattsteuerung einer Fluidzellpresse vorgestellt.

Gliederung des Kapitels

Der erste Abschnitt des Kapitels 4 beginnt mit einer Einführung in eine Terminologie, welche häufig zur Systembeschreibung benutzt wird. Darauf aufbauend werden diejenigen Begriffe vorgestellt, die in der anschliessenden Systemmodellierung verwenden werden.

In Abschnitt 4.2 folgen Definitionen rund um die werkstattnahe Steuerung. Die betrachteten Begriffe (PPS, Leitstand etc.) sind allgemein bekannt, werden aber unterschiedlich ausgelegt. Sie werden hier definiert, um bei ihrer Verwendung keine Unklarheiten aufkommen zu lassen. Dann wird kurz auf Geschichte und Trends der Werkstattsteuerung eingegangen. Neben einigen Stichworten zum Thema wird auf Literatur verwiesen, die sich eingehender damit befasst.

Im dritten Abschnitt wird ein Konzept für eine modulare Werkstattsteuerung des Systems vorgestellt, das auf drei Entwicklungsstufen basiert (vgl. Abschnitt 2.4). Zuerst wird FluidPlan für die Einsatzplanung der Fluidzellpresse alleine entworfen, dann folgt der Einbezug der Wärmebehandlung und die dritte Stufe soll schliesslich den Einsatz aller drei Arbeitsplätze im System planen. Neben den Entwicklungsstufen werden die Zielkriterien vorgestellt und es wird gezeigt, wie die Synchronisation zwischen den Modulen vonstatten geht.

4.1 Modellbildung

Will man einen Fertigungsbereich computergestützt steuern, ist es sinnvoll, dieses System mit Hilfe einer Beschreibungssprache anwendungsorientiert zu modellieren. Dafür haben die in Abschnitt 4.1.1 zitierten MERTINS ET AL., SCHULTE, sowie BECKER ET AL. jeweils Terminologien eingeführt, wobei denselben Begriffen oft unterschiedliche Bedeutungen zugewiesen werden. Diese Begriffe werden teilweise wiedergegeben, um sie untereinander zu positionieren.

In Abschnitt 4.1.2 folgen eigene Definitionen, um die zitierten zu präzisieren bzw. zu ergänzen, die dann in der Modellierung in Abschnitt 4.1.3 verwendet werden. Diese beiden Abschnitte sind ausführlicher gehalten, als es für diese Arbeit nötig wäre: Damit soll die Basis geschaffen werden, das modellierte System in einen globalen Kontext bezüglich weiteren modellierten Fertigungssystemen zu stellen.

4.1.1 Einführung und Abgrenzung einiger Begriffe

In der Systembeschreibung in Abschnitt 2.2 wurde der Begriff Komponente informell verwendet und es wurde darauf hingewiesen, dass die modellierten Komponenten hier *Elemente* genannt werden. Das modellierte System setzt sich also aus einer Menge von Elementen zusammen. Im Weiteren wird jedem Element ein Typ sowie ein Zustand zugewiesen. Vorher müssen einige allgemeine Begriffe eingeführt werden, um die in dieser Arbeit benötigten Elemente definieren zu können.

Funktionselemente

Funktionselemente bezeichnen den Typ eines Elements (in der objektorientierten Terminologie existieren die analogen Begriffe „Klasse“ für Funktionselement und „Objekt“ bzw. „Instanz“ für Element). Es werden statische und dynamische Funktionselemente unterschieden. *Statische* Funktionselemente sind während der ganzen Lebenszeit eines Modells existent. *Dynamische* Funktionselemente haben eine (ev. wiederholte) begrenzte Existenz innerhalb eines Modells.

Attribute

Mit Hilfe von *Attributen* wird der Zustand eines Funktionselements beschrieben. Attribute nehmen eindeutig definierte Werte an, die sich unter klar definierten Bedingungen ändern. Ein Funktionselement kann eine endliche Menge von Zuständen annehmen, die i.A. zyklisch besucht werden. Diese sogenannten *Zustandsketten* sind für statische Funktionselemente geschlossene Ketten, für dynamische Funktionselemente offene Ketten.

Die vorgegebene zyklische Änderung für statische Funktionselemente der Attribute kann unterbrochen werden durch nicht erwünschte und nicht planbare Ereignisse, welche den Funktionselementen einen Ausnahmezustand zuweisen (Bei einem Element ‚Maschine M‘ mit der geschlossenen Zustandskette ‚M wartet - M wird umgerüstet - M bearbeitet einen Auftrag A - M wartet‘ wäre dies beispielsweise der Ausnahmezustand ‚M ist defekt‘.) Die Behandlung derartiger Ausnahmezustände ist im vorliegenden Fall nicht vorgesehen, da für das Erkennen und Beheben solcher Zustände die Mitarbeiter des Systems besser geeignet sind als ein computergestütztes Programm.

Baukasten, Bausteine und Baugruppen

In MERTINS ET AL. wird eine Werkstattsteuerung nach dem Baukastenprinzip aufgebaut. Der *Baukasten* umfasst dabei u.a. ein „Set von vorgefertigten, standardisierten Grundelementen und Verbindungen.“ Die Grundelemente sind sogenannte *Bausteine*, die innerhalb des Systems einfache funktionale Einheiten mit definierten Ein- und Ausgängen bilden. Die Verbindungen sind die möglichen Material- und Datenflüsse. Bausteine können zu *Baugruppen* zusammengefügt werden, auch sie bilden funktionale Einheiten, aber „auf einer höheren Aggregationsstufe“, also auf einer höheren (System-)Hierarchiestufe.

Auch SCHULTE beschreibt Fertigungsprozesse durch Bausteine und deren Verbindungen. Er trennt die Bausteine in *Basis-* und *Komplexbausteine* auf. Komplexbausteine können aus Basisbausteinen oder weiteren Komplexbausteinen bestehen. Durch diesen Mechanismus wird eine Hierarchiebildung ermöglicht, die ohne die Unterscheidung zwischen Bausteinen und Baugruppen auskommt und beliebig viele Hierarchiestufen zulässt. Im Weiteren werden Grenzbausteine, die wiederum aus Basisbausteinen bestehen, eingeführt. Sie werden zur Abbildung der Systemgrenzen gebraucht.

Marken

Neben den oben aufgeführten statischen Funktionslementen werden bei SCHULTE auch dynamische Funktionslemente eingeführt, sogenannte *Marken*. Als Beispiele werden Ressourcen und die herzustellenden Produkte genannt.

Module

Oft wird von modularer Bauweise gesprochen, sei es bei komplexen Werkzeugen oder bei Software. Darüber hinaus wird der Begriff auch als Synonym zum oben eingeführten Komplexbaustein verwendet.

Man findet jedoch selten eine explizite Gegenüberstellung der Begriffe ‚Modul‘ und ‚Baustein‘ oder ‚Baugruppe‘. BECKER ET AL. sprechen von ‚Bausteinen oder Modulen‘, und bei MERTINS ET AL. heisst es, ‚Programm-Bausteine (...) werden auch Module genannt‘. Und im Zusammenhang mit dem Begriff ‚Modularisierung‘ werden dort sowohl Bausteine als auch Baugruppen genannt.

4.1.2 Definition der benötigten Funktionselemente

Die Funktionselemente zur Modellierung des Systems sind in Anlehnung an SCHULTE definiert (vgl. oben). Da er ein allgemeines Modell zur Werkstattsteuerung vorstellt, also ohne eine konkrete Fertigungsstruktur vor Augen, sind im folgenden Modell nicht alle Funktionselemente übernommen und andere abgeändert worden.

Baustein

Bausteine sind einfache funktionale Einheiten mit definierten Ein- und Ausgängen. Sie werden als Blackbox betrachtet, d.h. ihre innere Struktur ist nicht bekannt. Bausteine sind statische Funktionselemente.

Modul

Module sind funktionale Einheiten, die aus Bausteinen oder aus weiteren Modulen bestehen und definierte Ein- und Ausgänge haben. Die Zustände von Modulen werden durch die Zustände ihrer Bestandteile sowie mit Hilfe von eigenen Attributen beschrieben. Module sind statische Funktionselemente.

Marke

Marken sind dynamische Funktionselemente. Mit Hilfe ihrer offenen Zustandskette werden sie zu jedem Zeitpunkt ihrer Existenz einem Baustein oder einem Modul zugewiesen.

Verbindung

Bausteine und Module werden durch Verbindungsoperatoren miteinander verbunden. Sie beschreiben die zugelassenen Wege, auf welchen sich Marken (Material) oder Zustandsbeschreibungen bzw. Anweisungen (Information) bewegen dürfen. Verbindungen können gerichtet oder ungerichtet sein.

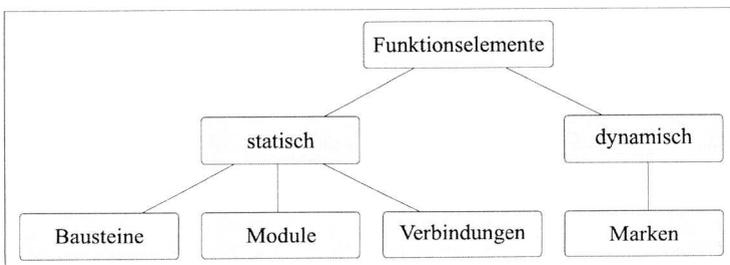


Abbildung 4-1: Hierarchie der Funktionselemente

Der Zustand eines Elements wird wie gezeigt durch Attribute beschrieben. Der Übergang eines Elements von einem Zustand in einen andern Zustand muss sich daher immer in einer Änderung mindestens eines Attributwerts des Elements ausdrücken. Es muss bei der Modellierung definiert werden, wann ein derartiger *Zustandsübergang* eintritt bzw. welche Bedingungen dazu erfüllt sein müssen und was den Zustandsübergang auslöst. Dieser Vorgang wird *Operation* genannt, wofür in der Regel folgendes benötigt wird: Ein zu veränderndes Element, ein Element, welches die Veränderung durchführt sowie die Erlaubnis, die Operation durchzuführen.

4.1.3 Das Modell von S(0)

Analog zu den Zuständen und den Zustandsübergängen müssen für die Modellierung von S(0) *Zeitabschnitte* t_i sowie *Zeitpunkte* dt_i eingeführt werden:

Als Zeitabschnitt t_i wird die Zeitdauer definiert, in der im System kein Attribut eines Elements eine Änderung erfährt. Zum Zeitpunkt dt_i , der den Übergang von t_i zu t_{i+1} beschreibt, ändert im System mindestens ein Attribut eines Elements seinen Wert.

Die Länge (Betrag) eines Zeitabschnittes t_i hängt gemäss dieser Definition davon ab, wann ein Attribut ändert, wann also ‚etwas passiert‘. Um diese Änderungen auszulösen, bedarf es einer sogenannten *Steuerungseinheit*:

Das Systemelement ‚Steuerungseinheit‘ repräsentiert einerseits einen oder mehrere Entscheidungsträger, die im Modell Ereignisse auslösen oder beenden, und andererseits eine Uhr, die zeitlich definierte Ereignisse startet bzw. beendet.

Beispiel: Die Bearbeitung des Auftrags A an der Maschine M wird ereignisgesteuert (durch einen Entscheidungsträger) ausgelöst. Nach der im Operationsplan angegebenen Operationszeit wird die Auftragsbearbeitung automatisch (durch die Uhr) beendet. Beide Male werden die Zustände und somit gewisse Attribute des Auftrags A und der Maschine M geändert.

(Diese Uhr ist nur virtueller Bestandteil des Systems, denn wenn die Zeit als Attribut bei einzelnen Elementen geführt würde, könnte strenggenommen kein Element einen Zustand ohne Attributsänderung für eine Zeitdauer grösser Null einnehmen.)

Abbildung 4-2 zeigt das modellierte System mit folgenden Elementen E1 bis E10:

- E1: *Aufträge* (Marke). In S(0) befindet sich eine Menge von Aufträgen $A = \{A_1, A_2, \dots, A_n\}$. Die Zustandskette eines A_i entspricht den Einträgen im Operationsplan des zugehörigen Artikels (also den Arbeitsgängen). Es dürfen jederzeit beliebig viele Aufträge existieren. Sie werden generiert, wenn sie das erste Mal auf einer Prioritätsliste erscheinen; sie werden gelöscht, wenn im Operationsplan alle Operationen an E3 und E6 abgeschlossen sind.
- E2: *Pufferlager* vor den Öfen (Baustein). E2 besteht aus allen Aufträgen, die für die Wärmebehandlung bereit sind. Pufferlager sind wichtige Bestandteile einer Fertigung, denn sie entkoppeln die Module, an denen die Aufträge bearbeitet werden. D.h., die Aufträge müssen nicht ständig bearbeitet werden; sie können i.A. beliebig lange in einem Pufferlager bleiben. Die Kapazität von E2 ist unbeschränkt.
- E3: *Wärmebehandlung* (Modul). Die Öfen bilden parallele Stationen und haben je eine Identifikationsnummer sowie eine Kapazität. Im Weiteren gibt es pro Ofen eine Menge von Chargen $C = \{C_1, C_2, \dots, C_k\}$, die die geplanten bzw. sich in Bearbeitung befindlichen Chargen beschreibt. Durch die Teilfamilie ist die Bearbeitungsdauer eines Auftrags A_i bzw. einer Charge C_k bekannt. Jeder Ofen kann alle Teilfamilien glühen, aber einige

Aufträge aus A können nur in einem der beiden Öfen gegläht werden. Für Öfen i lautet die Zustandskette: Öfen i glüht - Öfen i wartet - Öfen i wird eingerichtet - Öfen i wartet. Die beiden Öfen können ihre Zustände unabhängig voneinander ändern.

- E4: *Verögerungselement* (Baustein). Die Handarbeits- und Richtstationen werden in einem Element zusammengefasst. Die Bearbeitungsdauern der Aufträge sind nur als Schätzwerte bekannt. Da das Pufferlager vor den Handarbeitsstationen ebenfalls zu E4 gerechnet wird, ist die Kapazität von E4 unbegrenzt.

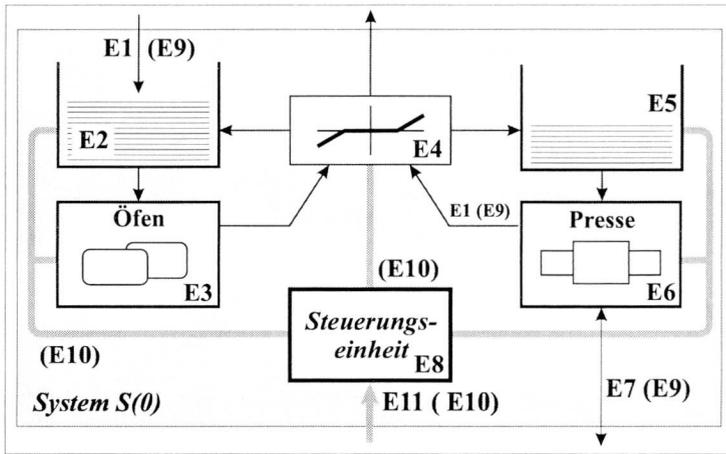


Abbildung 4-2: Modelliertes System (Systemebene 0)

- E5: *Pufferlager* vor der Fluidzellpresse (Baustein). Hier gelten dieselben Bemerkungen wie für E2 mit der Einschränkung, dass für lösungsgeglühte Aufträge der spätestmögliche Startzeitpunkt der Bearbeitung bekannt ist, womit sie nicht beliebig lange in E5 liegen bleiben können.
- E6: *Umformung* (Modul). Den beiden Tröge der Fluidzellpresse wird je eine Identifikationsnummer sowie eine nutzbare Fläche zugeordnet. Im Weiteren gibt es pro Trog eine Menge von Layouts $L = \{L_1, L_2, \dots, L_l\}$, die die geplanten bzw. sich in Bearbeitung befindlichen Layouts beschreibt. Die Bearbeitungsdauer eines Auftrags A_i ergibt sich aus der als konstant angenommenen Taktzeit und der Losgröße von A_i . Auch der Modul E6 besteht aus zwei parallelen Stationen unterschiedlicher Kapazität¹, sie greifen jedoch auf eine gemeinsame Ressource, die Presse zu. Für Trog i lautet die Zustandskette: Trog i presst - Trog i wartet - Trog i wird eingerichtet - Trog i wartet. Es kann sich jeweils nur ein Trog im Zustand ‚Trog i presst‘ befinden.
- E7: *Matrizen* (Marke). Pro Auftrag existiert genau eine Matrize, deren spezifische Eigenschaften zur Ablaufplanung an E6 benötigt werden. Eine solche Marke existiert genau dann, wenn der betreffende Auftrag existiert.

1. Die Troggröße kann verändert werden

- E8: Die *Steuerungseinheit* (Modul) bestimmt, welche Attribute an welchen Elementen geändert werden und steuert so den Materialfluss durch das System.
- E9: *Materialfluss* (Verbindung): Für die Aufträge wird beschrieben, welche Verbindungen zwischen den Elementen E2 bis E6 zulässig sind. Ein davon unabhängiger zweiter Materialfluss besteht zwischen E6 und dem nicht zum System gehörenden Matrizenlager.
- E10: *Informationsfluss* (Verbindung): Von den Elementen E2 bis E6 fließen die Informationen, in welchem Zustand sich diese Elemente und die zugewiesenen Aufträge befinden, zur Steuerungseinheit. Umgekehrt erhalten sie über E10 von E8 die Weisungen, welche Attribute als nächstes zu ändern sind. Im zweiten eingezeichneten Informationsfluss erhält E8 von ausserhalb des Systems die Prioritätslisten.
- E11: *Prioritätslisten* von E3 und E6. Darin sind alle Aufträge inkl. Priorität und Losgrösse aufgelistet, die in den entsprechenden Elementen bereit liegen. Die Angabe der Priorität dient der Steuerungseinheit zur Bestimmung der Auftragsreihenfolge

Im Materialfluss des Modells gelangen alle Aufträge nach E3 und E6 immer zu E4. In den Operationsplänen von Abbildung 2-4 sieht man jedoch, dass einige Aufträge direkt von E3 zu E5 bzw. von E6 zu E2 gelangen. Das ist kein Widerspruch, muss allerdings beim Entwurf des Konzepts berücksichtigt werden.

4.2 Was ist ein Werkstattsteuerungssystem

4.2.1 Definitionen rund um die werkstattnahe Steuerung

Für die hier vorgestellten Begriffe rund um die werkstattnahe Steuerung finden sich fast so viele Definitionen wie Bücher, deshalb werden die Begriffe mit Hilfe zitierter Definitionen untereinander abgegrenzt.

PPS (Produktionsplanung und -steuerung) und PPS-Systeme

In BÜCHEL2 wird die PPS hierarchisch in die Ebenen der Produktionsplanung und der Produktionssteuerung eingeteilt, die sich funktional jeweils in die drei Bereiche Materialwirtschaft, Terminplanung und -steuerung und Datenorganisation gliedern. Die Materialwirtschaft befasst sich v. a. mit der Bedarfsermittlung und der sich daraus ergebenden Materialbeschaffung, was gemäss Aufgabenstellung in Kapitel 2.4 in dieser Arbeit nicht betrachtet wird. Wichtig ist hingegen die Terminsteuerung, zu der die Werkstattsteuerung gerechnet wird. Ebenso zentral ist die Datenorganisation, zu der u. a. die Verwaltung der Grund-, Auftrags- und Zustandsdaten gezählt werden. Eine umfassende Definition des Begriffes PPS-System liefert STORM, mit welcher auch die Funktionalität der PPS gut aufgezeigt wird:

„PPS-Systeme unterstützen die organisatorische Planung, Steuerung und Überwachung der Produktionsabläufe von der Angebotsbearbeitung bis zum Versand nach Mengen-, Termin- und Kapazitäts Gesichtspunkten. Abgedeckt werden häufig die Funktionen der Fertigungsprogrammplanung mit Stamm- und Bewegungsdatenverwaltung, Bedarfs- und Stücklistenauflösung, Termin- und Kapazitätsplanung sowie der Produktionsauftragsbildung, -veranlassung und -überwachung.“

STORM beschreibt weiter, dass die (zentralen) PPS-Systeme in den Anfängen nur Listen mit den Planungsinformationen ausgaben, bevor die Informationsbereitstellung durch Bildschirme erfolgte. Bis dahin wurden PPS-Systeme gemäss MERTINS ET AL. mit dem Anspruch eingesetzt,

die ganze Kette der Auftragsbearbeitung zu planen und zu steuern, also auch die Steuerung in der Werkstatt abzudecken.

Laut BULLINGER & HIRSCH wurden PPS-Systeme im operativen Tagesgeschäft aber oft als störend empfunden. Als Gründe werden u.a. die aufwendige Verwaltung, die Schwerfälligkeit im Arbeitsablauf und die unzureichende Flexibilität genannt. Für die Steuerung der einzelnen Bereiche wurden deshalb sogenannte Plantafeln verwendet (eine ausführliche Beschreibung von Plantafeln findet sich bei STRACK).

Leitstand

Aufgrund der Erfahrung, dass PPS-Systeme v.a. für die lang- und mittelfristige Termin- und Kapazitätsplanung sowie für die Materialwirtschaft zu nutzen sind, wurden elektronische Fertigungsleitstände entwickelt. Die ersten elektronischen Leitstände waren gemäss HÄMMERLE denn auch im Wesentlichen den damals eingesetzten Plantafeln nachempfunden. Dass nun schon seit längerer Zeit werkstattnahe Steuerungssysteme im Einsatz sind, die sich ursprünglich an klassischen Plantafeln orientierten, beschreibt auch SCHRÖDEL. Diese Leitstände fungieren gemäss dieser Beschreibung als zentrale Fertigungssteuerung, können jedoch sehr unterschiedlich ausgeprägt sein. In diesem Sinne wird ein Leitstand in MERTINS ET AL. wie folgt definiert:

Ein Leitstand ist „ein grafikfähiger Bildschirm, der weitgehend die Möglichkeiten bietet, welche früher in der Werkstattsteuerung durch Plantafeln realisiert wurden. Die auf dem Bildschirm dargestellten Objekte werden per Tastatur oder durch die Maus und den Bildschirmlcursor handhabbar, d.h. anwählbar, greifbar und verschiebbar.“

Für ROHDE sind Leitstände immer auch interaktive Systeme, die - wie ein PPS-System - Planungslogiken verwenden: „Die Güte des Planungsergebnisses eines Leitstands hängt neben den formalisierbaren Logiken und den erfassten und strukturierten Informationen der Leitstanddatenbasis auch vom aktuellen Wissen und der Erfahrung des Leitstandbedieners ab.“

Werkstattsteuerung (WSS)

Die WSS wird wie gezeigt funktional der PPS zugewiesen, gemäss den vorhergehenden Ausführungen aber auf eigenständigen Systemen implementiert (oder von den Mitarbeitern in den Bereichen der Fertigung ohne Rechnerunterstützung durchgeführt). MERTINS ET AL. führen aus: „Steuern und planen im Bereich der Fertigung wird von REFA (Verband für Arbeitsstudien und Betriebsorganisation e.V.) als Werkstattsteuerung definiert“. Sie umfasst sowohl planerische als auch auslösende Tätigkeiten und wird in eine dispositive, eine administrative und eine operative WSS unterteilt. Das sei, „im Gegensatz zur langfristigen groben Planung des Produktionsprogrammes, und im Gegensatz zur mittelfristigen Fertigungsprogrammplanung, der Horizont der täglichen, schichtweisen, oder einen Zeitraum von Stunden umfassenden Feinsteuerung der Produktion, inklusive vorausgehender Feinplanung“. Konkreter wird SCHRÖDEL mit folgender Definition:

„Die Werkstattsteuerung soll für einen kurzfristigen Planungszeitraum, zum Beispiel eine Schicht oder kürzer, aus einem vorgegebenen Fertigungsprogramm von n Aufträgen entsprechend den Arbeitsplänen nach einem vorgegebenen Optimierungsziel den Auftragsdurchlauf entscheiden.“

Zur Durchführung dieser Aufgabe sind Tätigkeiten der Planung (Auftragsverwaltung, Reihenfolgeplanung, Kapazitätsfeinplanung), der Steuerung (Material- und Werkzeugflusssteuerung) und der Überwachung (Soll-Ist-Abgleich, Fortschrittsüberwachung) notwendig. Eine relativ

detaillierte Gliederung der im Rahmen der WSS zu erfüllenden Aufgaben wird in DIN/KCIM vorgeschlagen.

Hält man sich an die bis jetzt gemachten Definitionen, wird klar: Ein Leitstand deckt nicht die ganze Funktionalität ab, die von einer Werkstattsteuerung erwartet wird.

Werkstattsteuerungssystem (WSS-System)

Häufig verwendet wird auch der Ausdruck Werkstattsteuerungssystem. In der in diesem Kapitel zitierten Literatur fehlt leider eine kurze und prägnante Definition. MERTINS ET AL. füllen z.B. einige Seiten mit Anforderungen an ein WSS-System, andere gebrauchen den Begriff, ohne ihn einzuführen. Für diese Arbeit gilt deshalb eine eigene Definition:

Ein Werkstattsteuerungssystem ist ein computergestütztes Entscheidungsunterstützungsinstrument zur Werkstattsteuerung und ist auf einer autonomen (Rechner-) Plattform funktionsfähig.

Die Definition ist zu recht einfach gehalten, denn für den Funktionsumfang eines WSS-Systems kann auf die Definition der WSS zurückgegriffen werden. Die Forderung nach der Autonomie des WWS-Systems drückt aus, dass es kein PPS-Modul ist; allerdings ist eine Kopplung daran erwünscht und gemäss BULLINGER & HIRSCH auch zumeist realisiert. Eher selten sei hingegen eine Integration in Richtung Fertigung, also z.B. zur Werkzeug- oder NC-Programmverwaltung.

Laut MERTINS ET AL. werden an WSS-Systeme zwei Hauptaufgaben gestellt:

- Bereichsübergreifendes Leiten, im Sinne eines Durchführens verschiedener Objekte (Informationen und Material) durch die verschiedenen Stationen² der Fertigung hindurch
- Steuern, im Sinne eines Lenkens der einzelnen Stationen zum Zwecke der Ausführung ihrer Teilaufgaben.

Es hängt dabei von der konkreten Art und Organisation der Fertigung ab, in welchem Verhältnis die beiden Aufgaben der WSS zueinander stehen:

- ein mehr umfassendes, und verschiedene Kapazitätseinheiten koordinierendes Leiten von Objekten durch die Fertigungsstationen, oder eher
- das Steuern dieser Stationen, die als einzelne Kapazitätseinheiten äusserst komplexe Objekte darstellen können.

Als Ziel nennen BULLINGER & HIRSCH bei der Einführung einer rechnergestützten WWS für die Unternehmen u.a. eine Verbesserung der Termineinhaltung und eine Verkürzung der Durchlaufzeiten; für die Mitarbeiter steht der Abbau von Stresssituationen, die Erhöhung der Entscheidungskompetenzen und die Erweiterung der Handlungsspielräume im Vordergrund.

Unterscheidung der Begriffe Leitstand, zentrale und dezentrale Werkstattsteuerung

WEBER unterscheidet die dezentrale WSS, welche eine manuelle Organisationsform sei, von der zentralen WSS als EDV-gestützte Organisationsform. Dazu führt er folgendes aus: „Bei einer zentralen Werkstattsteuerung liegt die Dispositionskompetenz bei der Fertigungssteuerung in Form einer Fertigungsleitzentrale, die in der Regel als Leitstand bezeichnet wird.“ Ähnlich argumentiert STRACK: Für ihn sind die Begriffe „zentrale Werkstattsteuerung“ und „Leitstand“

2. Hier wird unter einer Station eine Funktionseinheit wie z.B. eine Fertigungszelle verstanden, im Gegensatz zur in Kapitel 2.1.2 erfolgten Definition einer Station des betrachteten Systems.

Synonyme. Aber während er den Leitstand eine Organisationsform der WSS nennt, ist für WEBER wie oben zitiert die zentrale resp. dezentrale WSS eine Organisationsform.

Wir halten uns bei diesem Gewirr an MERTINS ET AL., die explizit schreiben: „*Ein Leitstand ist kein Werkstattsteuerungssystem.*“ Ein Leitstand ist für sie eine „Systemkomponente eines Werkstattsteuerungssystems“ und die weiter oben aufgeführte Definition eines Leitstands stammt von eben diesen Autoren. Weiter wird von der Möglichkeit gesprochen, mehrere dezentrale, computergestützte WSS-Systeme in einem Betrieb einzusetzen; d.h., der Begriff „dezentrale WSS“ beschränkt sich nicht auf manuelle Hilfsmittel sondern präzisiert lediglich, dass das WSS-System nicht für die ganze Fertigung, sondern nur für einen Bereich zuständig ist.

4.2.2 Geschichte und Trends der Werkstattsteuerung

In SCHERER wird ein guter geschichtlicher Überblick aufgezeigt, der hier nicht in diesem Masse wiedergegeben werden soll. Im Zentrum der Arbeiten im Bereich des Industrial Engineering und auch des Operations Researchs stand lange das Problem der Reihenfolgeplanung als eine Kernaufgabe der WSS. Dieses auch „job shop scheduling“ genannte Problem wird beispielsweise in GEORGI ausführlich behandelt. Erst neuere Ansätze berücksichtigen verstärkt die dynamische und reaktive Natur der Reihenfolgeplanung. SCHERER führt aus, dass seit einigen Jahren viele Arbeiten in Zusammenarbeit von Hochschulen und Industriebetrieben entstanden sind. Er zählt dabei vier Richtungen auf, die analytisch unterschieden werden können:

- Entwicklung von „Elektronischen Leitständen“: Antrieb war dabei die Entwicklung neuer Formen des Technikeinsatzes und der Arbeitsorganisation in der Produktion. Oft wurden dabei neue Softwareprototypen entwickelt, z.B. bei MERTINS ET AL., HÄMMERLE und SCHRÖDEL.
- Arbeitsorientierte Konzepte, die im Bereich der arbeitspsychologischen sowie industriesoziologischen Forschung entstanden sind. Die Motivation dieser Arbeiten bestand in der Erkenntnis, dass die Einführung neuer Informatiksysteme oft ohne eine adäquate Gestaltung der Arbeitsorganisation vor sich ging.
- Ansätze zur umfassenden Dezentralisierung ganzer Unternehmungen: Bei diesen Arbeiten ist der Fokus nur beschränkt auf der Ebene der WSS, da sie eher globalere Fragestellungen berücksichtigen; als Stichwort sei hier die „fraktale Fabrik“ gegeben.
- „Human scheduling“: Im Zentrum steht hier die Frage, wie Menschen das Problem der WSS, insbesondere die Reihenfolgeplanung lösen. Dabei wird untersucht, wie Menschen optimal in ein informatikgestütztes Entscheidungssystem eingebunden werden können.

Auf eine Vorstellung verschiedener WSS-Systeme wird im Rahmen dieser Arbeit verzichtet. In BULLINGER & HIRSCH werden ausgewählte Leitstände (die nach der in dieser Arbeit geltenden Definition eher WSS-Systeme genannt würden) beschrieben und methodisch bewertet.

Die Tendenz, dezentrale WSS-Systeme zur Steuerung von Fertigungsbereichen einzusetzen, wird von verschiedenen Autoren kritisch hinterfragt. Sie führen aus, dass solche Systeme für die einzelnen Bereiche auf der operativen Ebene einen sicheren Spielraum für die kurzfristige Planung schaffen und sich so von der durch deterministische Denkansätze inspirierten „Totalplanung“ unterscheiden. Die gewonnene operative Freiheit wird immer noch durch den gesetzten Rahmen (des Bereiches) beschränkt, aber auch stabilisiert. Die globale Dynamik der gesamten Fertigung ist dadurch allerdings eingeschränkt. Denn nachdem die Ecktermine fixiert sind, erweist sich der Ansatz für das gesamte System als relativ unflexibel. Das Ziel muss daher gemäss SCHERER eine „Koordinierte Autonomie“ sein, deren Autonomiegrenzen sich aus den strukturellen und situativen Bedingungen des Systems immer wieder neu ergeben.

4.3 Entwicklungsstufen des Werkstattsteuerungskonzepts für S(0)

Es liegt ein spezielles Werkstattsteuerungsproblem vor: Nur drei Arbeitsplätze sind zu koordinieren, welche jedoch bezüglich Ablaufplanung komplexer sind als Maschinen, wie sie z.B. in der spanenden Fertigung vorkommen. Denn auf E3 und E6 können jeweils mehrere Aufträge auf einmal bearbeitet werden. Während bei E3 die Anfangs- bzw. Endzeitpunkte der Bearbeitung dieser Aufträge übereinstimmen, sind diese bei E6 im Allgemeinen verschieden.

Es ist deshalb sinnvoll, FluidPlan modular aufzubauen und stufenweise zu entwickeln (vgl. Abschnitt 2.4). In der letzten der drei geplanten Stufen werden die von MERTINS ET AL. zitierten zwei Hauptaufgaben eines WSS-Systems von verschiedenen Modulen wahrgenommen: Ein Hauptmodul übernimmt das koordinierende Leiten der Aufträge durch S(0), drei separate Module werden für das Steuern von E3, E4 und E6 benötigt. Der dreistufige Aufbau wird in diesem Kapitel vorgestellt.

Da in dieser Arbeit ein WSS-System primär als *Instrument zur Entscheidungsunterstützung* betrachtet wird, soll FluidPlan in erster Linie statistische Informationen bzw. Vorschläge bereitstellen, und nicht selbst Entscheidungen treffen. MERTINS ET AL. schreiben Mensch-Maschine-Systemen in diesem Zusammenhang ein Potential zur vorausschauenden gedanklichen Simulation von Handlungsketten (als Ursachen) und zur Bewertung ihrer Ergebnisse (als Wirkung) zu. Wichtig ist bei WSS-Systemen die Flexibilität, welche durch die Kompetenz zum Ausregeln des Regelkreises und durch die Fähigkeit zur Eigenregulation geschaffen wird. Letzteres wird durch eine hohe Interaktivität von FluidPlan erreicht, die Kompetenz muss vom Bediener eingebracht werden. Bei FluidPlan ist primär der Vorarbeiter als Bediener vorgesehen.

SCHERER zeigt ebenfalls auf, dass die Planung möglichst flexibel sein soll und ruft dazu auf, den Menschen als zentrales Flexibilitätspotential wahrzunehmen. Denn die Produktion verhält sich in vielen Fällen eher chaotisch, da auch geringe Abweichungen in einer Situation zu grossen Abweichungen gegenüber dem geplanten Zustand führen können. Dieser Ansatz steht der traditionellen Sichtweise gegenüber, welche ein deterministisches Bild der Produktion vermittelt. Unsicherheiten werden dabei aufgefangen durch Sicherheitsfaktoren, doch Störungen verunmöglichen trotzdem oft die Durchführung der geplanten Vorgaben.

4.3.1 Stufe 1: Lokale Einsatzplanung der Fluidzellpresse

Aufgabe des Planungsmoduls ‚Nesting‘ ist erstens, dem Bediener den Zustand der Systemelemente E5 und E6 anzuzeigen und zweitens aufgrund der Bedienerentscheide eine der folgenden Funktionen auszuführen: einen Layoutvorschlag berechnen, eine Folge von Layoutvorschlägen berechnen oder eine Folge von Layoutvorschlägen optimieren. Der Modul ist mehrere Ebenen unterteilt, wie Abbildung 4-3 zeigt.

Nestingfenster

Aufgabe des (Teil-)Moduls ‚Nestingfenster‘ ist die Präsentation statistischer Daten, die dem Bediener als Entscheidungsgrundlage dienen. Die wichtigsten Daten sind:

- Anzahl Aufträge pro Teilefamilie in E5 und E6
- Verteilung der Aufträge pro Teilefamilie auf die Prioritätsklassen
- Geschätzte Anzahl Pressungen, die zur Abarbeitung einer Teilefamilie notwendig sind (für die Berechnung dieses Wertes vgl. Abschnitt 5.2.4).
- Bereits geplante Layouts pro Trog (Grafik sowie Liste der dazugehörenden Matrizen)

Der Vorarbeiter benötigt weitere Informationen, die nicht im System sind, bevor er die folgenden Entscheidungen trifft. Er weiss z.B., ob Probepressungen neuer Matrizen einen Trog belegen, er kennt die personelle Situation an den einzelnen Arbeitsplätzen etc. Gestützt auf alle Informationen entscheidet er,

- wie spezifische Parameter (beispielsweise die geforderte Minimalauslastung der Layouts) des Moduls gesetzt werden sollen,
- wieviele Tröge eingesetzt werden,
- und welche Teilefamilien auf welchem Trog bearbeitet werden. Hierzu gibt es verschiedene Varianten:
 - Layouts für eine neue Teilefamilie berechnen (Bedingung: Trog ist leer)
 - Layouts einer Teilefamilie, die bereits eingeplant sind, ergänzen um weitere Layouts (beispielsweise soweit, dass Layouts für 100 Pressungen geplant sind). Die Strategie zur Generierung von Layoutfolgen wird in Kapitel 5.3 erläutert.
 - Eingeplante Layouts optimieren.

Je nach Wahl übergibt das ‚Nestingfenster‘ den Moduln in $S(-1)_6$ bzw. in $S(-2)_6$ die notwendigen, richtig gefilterten Daten. Die dort berechneten Vorschläge kann der Bediener akzeptieren oder ablehnen (und nach dem Ändern spezifischer Parameter die Berechnung neu starten). Der Modul dient also zur Kommunikation zwischen den Rechenmodulen und dem Bediener.

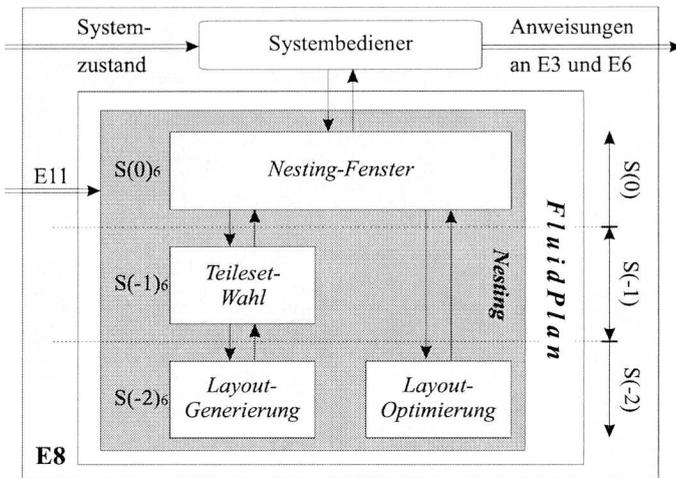


Abbildung 4-3: Erste Entwicklungsstufe: Planungsmodul ‚Nesting‘

Teilesetwahl

Der Modul ‚Teilesetwahl‘ muss eine Untermenge von in E5 bereitliegenden Aufträgen generieren (sog. Teilesetkandidaten), die gleichzeitig gepresst werden müssen. Dazu wird ihm vom ‚Nestingfenster‘ eine Liste mit Aufträgen übergeben. Die Aufträge müssen derselben Teilefamilie angehören und momentan E5 oder E6 zugewiesen sein. Wie aus den bereitliegenden Aufträgen der Teilesetkandidat bestimmt wird, ist Thema des Kapitel 5.

Input des Moduls ‚Teilesetwahl‘ sind also die E1 in E5 und E6, Output ein Teilesetkandidat.

Layoutgenerierung

Der Modul ‚Layoutgenerierung‘ hat die Aufgabe, die Matrizen der von der ‚Teilesatzwahl‘ zusammengestellten Aufträge zu einem *Layout* zu packen. Wie Layouts generiert und bewertet werden, ist in Kapitel 6 detailliert beschrieben.

Input des Moduls ‚Layoutgenerierung‘ ist also ein Teilesetkandidat, Output (wenn möglich) ein Layout für die entsprechenden Matrizen.

Layoutoptimierung

Aufgabe dieses Moduls ist es, eine vorliegende Layoutfolge in einem von der Layoutgenerierung unabhängigen Schritt zu optimieren (vgl. Abschnitt 6.7). Die Zusammensetzung der Teilesets sowie ihre geplante Reihenfolge wird dabei nicht verändert. Optimiert werden in diesem Modul lediglich die Positionen der einzelnen Matrizen in den verschiedenen Layouts. Das heisst, es wird der Handlingsaufwand minimiert, der benötigt wird, um alle Layouts einer Layoutfolge im Trog zu legen. Diese Aufgabe in einem unabhängigen Modul und typischerweise nach Generierung einer ganzen Folge von Layouts zu lösen, ist sinnvoll, da Layouts sequentiell generiert werden (wo der Handlingsaufwand nur bezüglich des direkt vorhergehenden Layouts - also lokal - optimiert wird).

4.3.2 Stufe 2: Einbezug der Wärmebehandlung

In der zweiten Entwicklungsstufe wird die Wärmebehandlung in das Steuerungselement E8 eingebunden. Der Modul ‚Chargierung‘ plant den Einsatz der Elemente E2 und E3 und ist analog zum Modul ‚Nesting‘ konzipiert, wie aus Abbildung 4-4 ersichtlich ist:

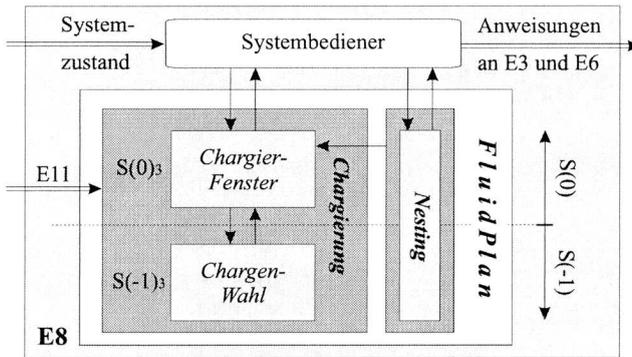


Abbildung 4-4: Zweite Entwicklungsstufe

Die beiden Module ‚Chargierung‘ und ‚Nesting‘ sind miteinander gekoppelt, gemäss Abbildung 4-4 findet eine Rückwärtsterminierung von der Presse zu den Öfen statt. Möglich wäre auch eine Vorwärtsterminierung. Da aber im realen Beispiel ebenfalls die Presse mit ihren wesentlich ‚härteren‘ Randbedingungen bestimmend ist, wird die Rückwärtsstrategie in dieser Stufe nicht hinterfragt. Es sei hier auf BERNATZKI verwiesen, der in seiner Arbeit in analoger Weise auf die beiden Feinterminierungsstrategien näher eingeht und diesen eine dritte, gemischte Strategie gegenüberstellt.

Ein Planungszyklus z (für einen halben bis ganzen Tag) verläuft wie folgt:

1. Neuste Prioritätsliste einlesen (vom Bediener ausgelöst)
2. Für den Planungszyklus z mit dem Nestingprogramm definitive Layoutvorschläge berechnen, Input sind die Aufträge in E5
3. Für den Planungszyklus $z+1$ mit dem Nestingprogramm provisorische Layoutvorschläge berechnen, Input sind alle Aufträge im System (mit geschätzter Inputzeit in E5)
4. Aufgrund der in Schritt 3 berechneten Layouts für den Planungszyklus $z+1$ werden für die Wärmebehandlung Vorschläge für die Chargen berechnet (siehe unten).

Man bemerkt, dass für diese Funktionalität am Nestingfenster aus Stufe 1 Anpassungen notwendig sind: Der Bediener muss angeben können, wie lange ein Planungszyklus dauert und für welchen Zyklus Layoutvorschläge zu berechnen sind.

Im Weiteren kann manchmal auf Schritt 2 verzichtet werden: Ist der Input von Schritt 2 beim Zyklus z identisch mit dem Input von Schritt 3 beim Zyklus $z-1$, können die provisorischen Vorschläge aus $z-1$ in z definitiv übernommen werden. Typischerweise muss aber Schritt 2 durchgeführt werden, weil ein Auftrag normalerweise zwischen zwei Stationen an einer Handarbeitsstation bearbeitet wird, wofür weder der Bearbeitungsbeginn noch die genaue Bearbeitungsdauer bekannt sind. Somit wird eine Umsetzung der geplanten Layouts oft nicht möglich.

Chargierfenster

Auch dieses Fenster dient der Kommunikation zwischen Rechenmodul und Bediener. Als Input dient hier aber nicht primär die Prioritätsliste von E3, sondern v.a. die provisorischen Pressenlayouts (aus oben erklärtem Schritt 3). Der Bediener entscheidet,

- wie spezifische Parameter des Moduls gesetzt werden sollen,
- und welche Teilefamilien in welchen Öfen geglüht werden sollen.

Der Modul ‚Chargenwahl‘ stellt Vorschläge zur Zusammensetzung der Chargen zusammen und gibt zugleich an, ob mit diesen Vorschlägen die provisorischen Layouts wie vorgeschlagen gepresst werden können. Einerseits müssen die Aufträge rechtzeitig an der Presse ankommen, andererseits darf die nach dem Lösungsglühen einsetzende Aushärtung noch nicht zu weit fortgeschritten sein. Diese Vorschläge kann der Bediener akzeptieren oder ablehnen (und nach dem Ändern spezifischer Parameter die Berechnung neu starten).

Chargenwahl

Bei der Chargierung der Öfen müssen nur die Chargen zusammengestellt, aber keine Vorschläge für die Lage der Platinen gemacht werden. Dieser Modul wird eine grössere Funktionalität haben als der Modul ‚Teilesetwahl‘, da er die ganze Aufgabe (ohne zusätzliches Modul auf einer tieferen Systemebene) löst. In Kapitel 2.2.4 sind die Ausgangsdaten zusammengestellt und werden hier nicht mehr wiederholt.

Das vorliegende Chargierproblem gehört zur Klasse der *eindimensionalen³ Packingprobleme*. Zu dieser Thematik ist bereits eine Fülle von Arbeiten veröffentlicht worden, weshalb hier nicht näher darauf eingegangen wird. Erwähnt sei DYCKHOFF & FINKE; darin wird eine ausführliche Einteilung von Packingproblemen vorgenommen, überdies wird eine umfangreiche Liste mit Veröffentlichungen zu den einzelnen Problemvarianten angegeben.

3. Von Öfen und Platinen sind nur die Flächen bekannt, da nur Chargen zusammengestellt werden müssen.

Die Wirkungsweise der 2. Stufe, die sowohl provisorische als auch definitive Layouts berechnet, wird in Abbildung 4-5 veranschaulicht:

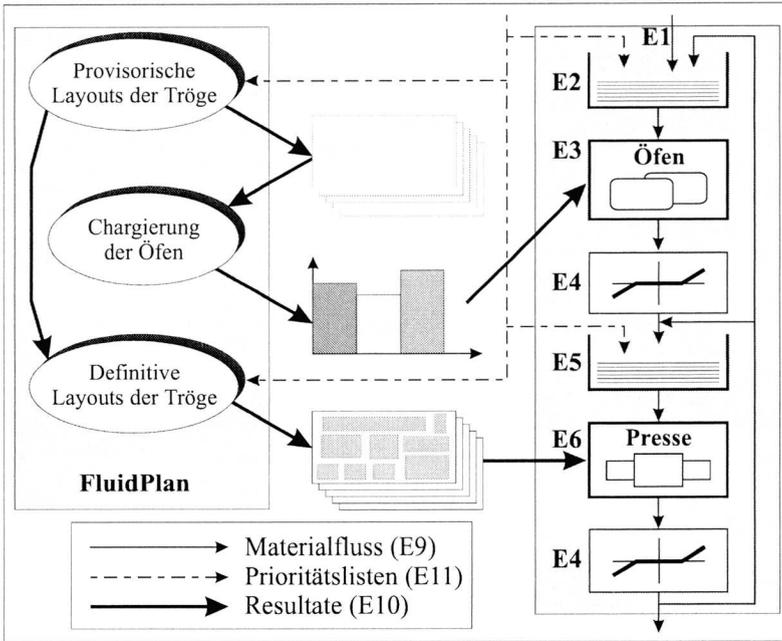


Abbildung 4-5: Ablauf der zweiten Entwicklungsstufe

4.3.3 Stufe 3: Reihenfolgeplanung des Gesamtsystems

In der oben vorgestellten zweiten Stufe werden die provisorischen Layoutvorschläge als Randbedingung des Chargiermoduls benutzt. Bei der dritten Stufe wird das Zusammenspiel der Module ‚Chargierung‘ und ‚Nesting‘ durch einen übergeordneten Modul synchronisiert. Dieser Modul trägt den Namen ‚Synchro‘. Der Ansatz macht nur dann Sinn, wenn auch die Zwischenbearbeitungen eingeplant werden, wozu der Baustein ‚Handarbeit‘ eingeführt wird. Somit muss pro Auftrag die Bearbeitungsdauer einer Zwischenbearbeitung bekannt sein, ebenso die Anzahl maximal zur Verfügung stehender Arbeitsplätze. Das Konzept für die dritte Ausbaustufe von E8 ist in Abbildung 4-6 aufgezeigt.

Die Aufgabe, zulässige Ablaufpläne zu generieren, ist erfahrungsgemäss kein Problem (der Vorarbeiter stellt ja auch ohne computergestütztes Hilfsmittel Ablaufpläne zusammen). Die einzige zeitliche Restriktion, gegeben durch das Lösungsglühen, ist i.A. unproblematisch, da der Zeitrahmen von einer Woche relativ gross ist. Ziel der Planung ist also, im Sinne einer Zielfunktion oder von Gütekriterien optimale Ablaufpläne zu generieren.

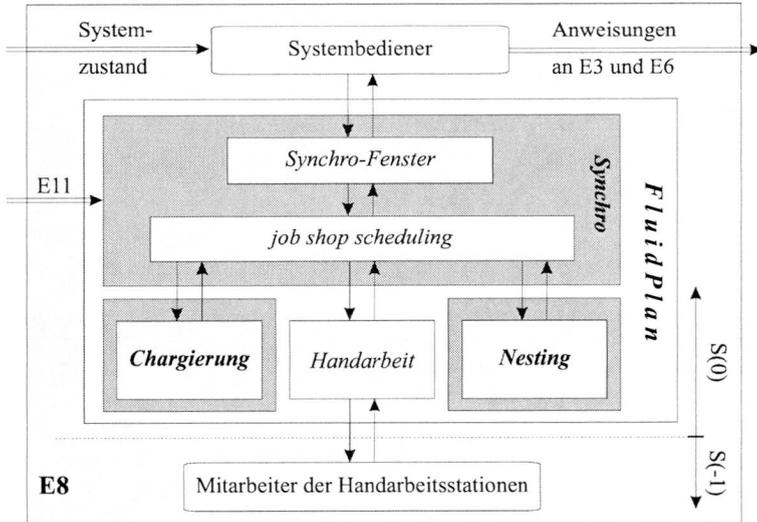


Abbildung 4-6: Dritte Entwicklungsstufe

Ziel des Moduls ‚Synchro‘

In Abschnitt 2.1.3 wurde bereits ein verbales Zielkriterium für das gesamte System formuliert: Zu optimieren sind die Durchlaufzeiten aller Aufträge in Funktion ihrer Priorität. Eine formale Zielfunktion für diese Aussage wird in Abschnitt 4.4.1 vorgestellt, an dieser Stelle werden zwei Regeln genannt, mit welchen der Submodul ‚Synchro‘ dieses Ziel erreichen soll:

- Regel 1: Die Einplanung der Aufträge in das System erfolgt nach Prioritätsklassen. Das heisst, die Klassen mit höchster Dringlichkeit (von Hand heraufgesetzt) werden zuerst eingeplant. Die verbleibenden Kapazitäten an den einzelnen Stationen werden mit Aufträgen aus tiefer eingestuftem Prioritätsklassen aufgefüllt.
- Regel 2: Bei diesem Auffüllen soll v.a. darauf geachtet werden, dass in E5 (Puffer vor der Presse) stets ein „guter“ Auftragsmix vorhanden ist. Das heisst, es soll eine Menge von Aufträgen bereit liegen, mit deren Matrizen erstens eine hohe Auslastung möglich ist und zweitens die Membranenbelastung niedrig gehalten werden kann. Insbesondere soll die Ankunft hoher und grosser Matrizen in E5 gesteuert werden.

Diese Regeln sind nur deshalb erfolgsversprechend, weil i.A. relativ wenige Aufträge hoher Priorität im System sind. Das heisst, nach Einplanung dieser Aufträge ist normalerweise noch genügend Kapazität vorhanden, um den Ablaufplan gemäss Regel 2 aufzufüllen.

Wir werden nun mit folgenden Punkten die Funktionalität des Moduls ‚Synchro‘ beschreiben:

- Funktionsweise des ‚Synchrofensters‘: Welche Information wird dem Bediener gezeigt, was kann er bestimmen und welche Berechnungen kann er auslösen
- Besonderheiten des Submoduls ‚job shop scheduling‘: Zu zeigen ist, in welchen Punkten sich die hier gestellte Aufgabe von der klassischen Reihenfolgeplanung unterscheidet.

Resultat des job shop scheduling ist ein Plan, der die Arbeitsgänge der aktuellen Aufträge für gewisse Zeitintervalle den Arbeitsplätzen zur Bearbeitung zuweist.

- Dann wird auf die Synchronisation der drei Submodule bzw. Bausteine ‚Chargierung‘, ‚Handarbeit‘ und ‚Nesting‘ eingegangen. Zentrale Frage ist hier, wie die Submodule reagieren, wenn sie die Aufträge innerhalb der vorgegebenen Zeitfenster nicht bearbeiten können. Wir zeigen verschiedene Möglichkeiten auf, dieses Problem zu lösen.

Bei der Diskussion des dritten Punktes werden auch Ansätze erwähnt, welche mit sogenannten *Agenten* operieren. Auf diesen zur Zeit sehr geläufigen Begriff wird näher eingegangen, bevor dann Ansätze zur Synchronisation vorgestellt werden.

Am Schluss dieses Abschnitts wird das Konzept für den Baustein ‚Handarbeit‘ aufgezeigt.

Synchrofenster

Auch hier liefert das Fenster des Moduls ein „Schaltpult“ zum Starten, Begutachten und Akzeptieren bzw. Verwerfen der Berechnungen. Neben einer Gesamtübersicht aller Aufträge im System können weiterhin die Fenster der Submodule ‚Chargierung‘ und ‚Nesting‘ sowie des Bausteins ‚Handarbeit‘ betrachtet werden.

Im Zentrum der Planung steht immer noch die Fluidzellpresse, weshalb für sie die wichtigsten Randbedingungen gesetzt werden, bevor die Reihenfolgeplanung beginnt. Das sind die

- Anzahl der eingesetzten Tröge,
- Teilefamilie pro Trog (ev. mehrere in einer Reihenfolge),
- bevorzugte Teilefamilie an den Öfen und
- maximal verfügbare Anzahl der Handarbeitsplätze.

Diese Angaben ergeben zusammen mit den Prioritätslisten des Systems die Rahmenbedingungen für die Planung des im Folgenden beschriebenen Submodul.

job shop scheduling

Die Synchronisation der Module basiert auf den Kapazitäten der drei Arbeitsplätze und stellt einen Spezialfall des job shop (scheduling)-Problems dar. Zu dessen Lösung besteht bereits eine Vielzahl von Arbeiten, weshalb hier nicht näher darauf eingegangen wird. Als Resultat liefert der Modul ‚job shop scheduling‘ für jeden Auftrag disjunkte Zeitintervalle, während denen sie einem der Arbeitsplätze zur Bearbeitung zugewiesen sind.

Der Submodul eines Arbeitsplatzes muss diese Zuweisungen überprüfen, indem für den Arbeitsplatz selbst ein detaillierter Ablaufplan der zugewiesenen Aufträge berechnet wird (beispielsweise eine Folge von Layoutvorschlägen für die Presse). Zentral ist dabei die Frage, ob mit den gegebenen Zeitintervallen zulässige Ablaufpläne für die Submodule gefunden werden.

Erwähnenswert ist die Arbeit von GEORGI, der verschiedene Lösungsverfahren vorstellt, sie mittels Testdatensätzen simuliert und die Resultate nach unterschiedlichen Zielfunktionen bewertet. Gemäss Definition von GEORGI unterscheidet sich das vorliegende Problem in folgenden Punkten vom klassischen job shop-Problem:

- Im Startzeitpunkt dt_0 der Planung können alle vorliegenden Aufträge eingeplant werden, da Aufträge erst bekannt sind, wenn sie im System auftauchen. Es existieren also keine frühesten Freigabezeitpunkte der Aufträge für das Gesamtsystem bzw. diese sind für alle Aufträge dt_0 . (Es wäre natürlich von Vorteil, wenn Aufträge bereits vor ihrem erstmaligen Auftauchen in E2 bekannt wären.)

- Fälligkeitspunkte, welche durch das Aushärten lösungsgeglühter Platinen gegeben sind, müssen als Restriktionen aufgefasst werden. Fälligkeitspunkte aufgrund der provisorischen Einplanung auf den nächsten Arbeitsgang fliessen - wie auch manchmal im klassischen Fall - nur in die Zielfunktion ein. GEORGI erwähnt, dass viele Ansätze oft keine zulässigen Lösungen generieren, wenn Fälligkeitszeitpunkte als Restriktionen gelten.
- Restriktionen der Nichtgleichzeitigkeit beschränken die Anzahl (gleicher) Operationen, die gleichzeitig stattfinden können. Bezüglich Nichtgleichzeitigkeit unterscheidet sich das vorliegende Problem in zwei Punkten vom Klassischen (bei welchem pro Operation nur eine Maschine zur Verfügung steht, auf welcher nur ein Auftrag auf einmal bearbeitet werden kann):
 - Parallele Maschinen: Zwei oder mehrere Kapazitätseinheiten, auf denen die gleichen Operationen durchgeführt werden können, werden parallele Maschinen genannt. Diese Definition trifft auf die Öfen und auf die Handarbeitsplätze zu
 - Bei den Öfen sowie bei der Presse können mehrere Aufträge gleichzeitig bearbeitet werden. Die beschränkende Grösse ist demnach nicht die Anzahl gleicher Operationen, die zusammen durchgeführt werden können, sondern die Kapazität der Maschinen
- Bei der Wärmebehandlung ist es möglich, dass ein sehr grosser Auftrag nicht in einer einzigen Charge geglüht werden kann. Dann muss die Operation sequentiell (im gleichen Ofen) oder parallel (auf beide Öfen) gesplittet werden. Dieses job splitting ist bei der klassischen Problemstellung nicht vorgesehen
- Kann eine Operation an einem Auftrag auf einer Maschine unterbrochen werden, spricht man von *präemptiven Operationen*, was bei den Arbeitsplätzen Handarbeit sowie Umformung möglich ist. Diese Möglichkeit wird nur angewendet, wenn ein Auftrag mit Priorität PriA im System erscheint (vgl. Tabelle 4-1). Dieser Fall trifft selten ein, deshalb muss er von FluidPlan nicht abgedeckt werden; solche Aufträge werden vom Vorarbeiter ‚informell‘ durch das System geleitet

Da ständig neue Aufträge im System auftauchen, die noch nicht eingeplant sind, muss eine *rollende Planung* vorgenommen werden. Das heisst, dass z.B. alle halben oder ganzen Tage die Prioritätslisten nachgeführt werden und die Planung neu durchgeführt wird. Dabei werden alle geplanten Reihenfolgen neu berechnet. Auf Stufe PPS wird auch mit einer rollenden Planung gearbeitet, allerdings mit wochenweisen Intervallen. Es liegen also vergleichsweise kurze Intervalle vor.

Agenten

Gemäss FISCHER ist ein *Agent* eine abstrakte Berechnungseinheit mit bestimmten Problemlösungsfähigkeiten. Er führt weiter aus, dass im deutschsprachigen Raum noch keine gültige Definition zu finden ist. Es wird aber eine Hierarchie von Agentenmodellen zitiert: Die unterste Stufe bildet der ‚primitive Agent‘ (z.B. Sensor-Aktor-Systeme), gefolgt vom ‚technischen Agenten (programmgesteuerte Systeme wie Roboter, Werkzeugmaschinen usw.). Die weiteren Stufen sind der ‚technisch-intelligente Agent‘, der ‚kognitive Agent‘ und der ‚soziale Agent‘.

SCHULTE führt agentenbasierte Verfahren bei der Vorstellung von Verfahren zur Werkstattsteuerung unter dem Kapitel ‚Näherungsverfahren‘ ein. Er beschreibt sie als Konzept, mit dem die betrieblichen Planungs- und Ablaufstrukturen modelliert werden können. Agenten sind dabei autonome Softwareobjekte mit definierter Schnittstelle zur Umwelt, eigenen Zielen und eigenen Verhaltensregeln. Die Agenten werden als Steuerungsmechanismus den verschiedenen Objekten eines physischen Modells zugeordnet (z.B. Fertigungslosen, Ressourcen oder Mitarbeitern).

Die Kommunikation der Agenten basiert nach FISCHER z.B. auf Kontraktnetzen. Dieses Konzept definiert ein Protokoll, mit dem das Problemlösen in einer Multiagentenumgebung gesteuert werden kann. Ein Agent kann dabei die Rolle eines Managers, eines Anbieters oder eines Kontraktnehmers übernehmen. Eine weitere Möglichkeit der Kommunikation wird mittels sogenannten ‚verteilten Blackboards‘ wahrgenommen; diese funktionieren wie Börsen, wo Angebote und Nachfragen beispielsweise nach Ressourcen kenntlich gemacht werden.

KALDICH gebraucht Agenten als spezielle Management-Applikationsprozesse, die in einer lokalen Systemumgebung eine Reihe von Objekten verwalten. Das Thema der Arbeit von KALDICH ist das Management von Kommunikationsnetzen.

Bei KIMMLINGEN, der sich mit der Einbettung eines Prozessmodells in eine konkrete Organisation befasst, sind Agenten ausführende Einheiten, welche „die Schritte eines Softwareentwicklungsprozesses ausführen. Konzeptionell können dies Menschen oder auch computergestützte Werkzeuge sein“. Durch vorgeschriebene Tätigkeiten können sie ihren bzw. den Zustand von anderen Agenten verändern.

Abschliessend soll zum Thema ‚Agenten‘ noch einmal SCHULTE zum Wort kommen, der 1995 schrieb: „Die Modellierung eines realen Systems mit einem solchen Ansatz ist anschaulich und überschaubar, da sie sich an den Objekten der Realität orientiert. Die Optimierungsalgorithmen sind verteilt in den verschiedenen Agenten und Kommunikationsmechanismen implementiert und daher ebenfalls anschaulich. (...) Wie jedoch die Realität zeigt, sind diese Abläufe nur bedingt geeignet, im Sinne der Fertigungssteuerung optimale Ergebnisse zu liefern. Besonders die Annahme, dass alle Entscheidungen grundsätzlich am besten lokal getroffen werden, führt in seltenen Fällen zu einem globalen Optimum des gesamten Prozesses. (...). Insgesamt befinden sich die agentenbasierten Verfahren zur Zeit in einem frühen Forschungsstadium, das ihren Einsatz in der industriellen Werkstattsteuerung noch nicht zulässt.“

Modulsynchronisation: Deterministische Ansätze

Deterministische Ansätze behandeln die Fälligkeitszeitpunkte der vom job shop scheduling berechneten Zeitintervalle (welche auf den Kapazitäten der einzelnen Arbeitsplätze beruhen) als Restriktionen. Diese Zeitintervalle dienen als Input bei der Berechnung der detaillierten Ablaufpläne für die einzelnen Submodule. Als Output wird folgendes erwartet:

- Beim Submodul ‚Chargierung‘: Liste der Chargen, unter Angabe des geplanten Zeitpunktes sowie einer Zusammenstellung der Aufträge pro Charge
- Beim Submodul ‚Nesting‘: Liste der Folge von berechneten Layoutvorschlägen, mit einer Liste der Aufträge pro Pressung inklusive Position der entsprechenden Matrize im Trog
- Beim Baustein ‚Handarbeit‘: Liste mit den zu bearbeitenden Aufträgen, geordnet nach (Arbeits-)Station sowie nach Anfangszeitpunkt der Bearbeitung

Erhält man diesen Output tatsächlich, muss keine Modulsynchronisation im eigentlichen Sinne durchgeführt werden, da der übergeordnete Ablaufplan bereits synchrone Zeitfenster liefert. Kann aber unter Beachtung der Fälligkeitszeitpunkte eines der Submodule keinen zulässigen Ablaufplan generieren, muss eine Synchronisation gestartet werden, bis lauter zulässige Detailpläne vorliegen. Für eine derartige Modulsynchronisation sind verschiedene Ansätze denkbar:

1. Aufträge, die nicht innerhalb der vorgegebenen Schranken liegen, werden durch Aufträge mit ähnlichen Operationsplänen ausgetauscht. Es ist nicht garantiert, dass dadurch zulässige Lösungen entstehen. Deshalb muss die Anzahl Austauschschritte begrenzt und dann ein anderer Ansatz eingesetzt werden, falls keine zulässige Lösung gefunden wird.

2. Die Einplanung von Aufträgen, die nicht innerhalb der vorgegebenen Schranken liegen, wird verzögert. Das heisst, im Startzeitpunkt d_0 einer neuen Reihenfolgeplanung für das Gesamtsystem werden nicht mehr alle vorliegenden Aufträge berücksichtigt.
3. Ansatz 2 kann auch „umgekehrt“ werden: Bei der Reihenfolgeplanung für das Gesamtsystem werden nur so viele Aufträge berücksichtigt, dass die Auslastung der einzelnen Arbeitsplätze relativ tief ist, beispielsweise 70% bei der Fluidzellpresse (zuerst eingeplant werden dabei gemäss Regel 1 Aufträge mit hoher Priorität). Die Planung wird dann mit immer mehr Aufträgen solange wiederholt, bis keine zulässigen Lösungen mehr resultieren. Die zuletzt erhaltene zulässige Lösung wird dem Benutzer vorgeschlagen.
4. Ein agentenbasierter Ansatz stützt sich auf die bei FISCHER vorgestellten Kontraktnetze: Die Submodule berechnen für die Aufträge, welche nicht innerhalb ihres Zeitintervalls eingeplant werden können, alternative Intervalle und schlagen sie den vorgelagerten Submodulen vor. Diese überprüfen die Vorschläge und akzeptieren, verändern oder verworfen sie (es findet also ein eigentliches „Feilschen“ statt!). Dieser Ansatz impliziert erstens, dass im Modul ‚Synchro‘ ein Agent implementiert wird, der das Management dieses Feilschens übernimmt. Und zweitens müssen die Submodule selbst zu Agenten erweitert werden, damit sie überhaupt feilschen können. Es geht dabei nicht mehr primär um eine hohe Auslastung der Arbeitsplätze, sondern um das Einhalten der Zeitschranken unter Berücksichtigung der oben formulierten Regel 2.

Die vorgestellten deterministischen Ansätze werden aus verschiedenen Gründen für ungeeignet zur Lösung des vorliegenden Problems erachtet:

1. Unter Umständen sind sehr viele Rechendurchgänge nötig, bis unter Einsatz eines oben aufgeführten Ansatzes die Modulsynchronisation eine zulässige und zufriedenstellende Reihenfolgeplanung vorschlägt. Die zur Verfügung stehende Rechenzeit ist aber durch die Intervalle der rollenden Planung relativ kurz, wenn man bedenkt, dass die Berechnung von Layoutvorschlägen für einen Tag bei der jetzigen Konfiguration mindestens 30 Minuten reine Rechenzeit beansprucht (falls jeder Vorschlag akzeptiert wird).
2. Es wird auf die bereits zitierte Bemerkung von SCHERER verwiesen, wonach bereits geringe Störungen in der Produktion zu grossen Abweichungen gegenüber dem geplanten Zustand führen können. Das vorliegende System mit den immer wieder auftretenden Störfällen und den nicht exakt festlegbaren Bearbeitungszeiten bei den Handarbeitsstationen wäre geradezu prädestiniert, eine deterministische Planung ad absurdum zu führen.

Modulsynchronisation: Alternativer Ansatz

Eine Alternative zu deterministischen Ansätzen basiert auf derselben Reihenfolgeplanung des Gesamtsystems, unterscheidet sich aber in wesentlichen Punkten:

- Die beim job shop scheduling erhaltenen Zeitintervalle werden bei den Submodulen nicht als Restriktionen behandelt, sondern fliessen dort nur in eine Zielfunktion ein
- Der Output der einzelnen Submodule ist formal derselbe wie bei den deterministischen Ansätzen. Inhaltlich kann er aber recht verschieden sein, da aufgrund des erstgenannten Punktes auch Outputs resultieren können, bei denen die Zeitintervalle nicht eingehalten werden. Die Submodule behandeln derartige Fälle wie folgt:
 - Chargierung: Da hier wie beim job shop scheduling nur eine kapazitive Zuteilung der Aufträge zu den Chargen erfolgt, können die zeitlichen Vorgaben im Rahmen des Planungshorizontes immer umgesetzt werden (falls die Aufträge zur geplanten Zeit verfügbar sind). In diesem Zusammenhang sei bemerkt, dass ein Auftrag innerhalb

eines Planungsintervalls an maximal zwei Arbeitsplätzen bearbeitet wird, bis der nächste Planungsintervall beginnt

- Nesting: Dieser Modul muss Aufträge mit hoher Priorität und mit bald erreichtem Fälligkeitstermin (vom Lösungsglühen) bevorzugen. Da in den Pufferlagern Arbeit für durchschnittlich drei Tage bereitliegt, hat man bei einer Aushärtungszeit von sieben Tagen genügend Spielraum. Treten trotzdem Abweichungen von den vorgegebenen Zeitintervallen auf, werden diese an den Modul ‚Synchro‘ gemeldet und im nächsten Planungsintervall berücksichtigt. Da meist nur ein Trog der Presse im Einsatz ist, kann vor der Meldung auch der Einsatz des zweiten Troges geprüft werden
- Handarbeit: siehe unten

Durch den Modul ‚Synchro‘ wird also lediglich eine Grobplanung als übergeordneter Rahmen für die Submodule festgelegt. Da unter den gegebenen Umständen eine relativ aufwendige deterministische Detailplanung selten wirklich umgesetzt werden kann, ergibt der hier vorgestellte Ansatz ein gutes Verhältnis von Aufwand und Nutzen. Nicht vergessen werden soll, dass die Mitarbeiter des Systems heute schon viele Störungen situativ beheben und dies natürlich auch mit computergestützter Reihenfolgeplanung machen können.

Handarbeit

Die Mitarbeiter der Handarbeitsstationen können die Feinsteuerung selbst übernehmen. An einem Bildschirm wird ihnen angezeigt, welche Aufträge bereitstehen und wann ihre Fälligkeitstermine sind. Durch Angabe der Anzahl Stationen, an denen momentan Aufträge bearbeitet werden, kann der Baustein ‚Handarbeit‘ überprüfen, ob die Fälligkeitstermine eingehalten werden können. Trifft dies nicht zu, wird wie folgt reagiert:

- Meldung an die Bediener, wieviele Stationen mindestens bedient sein müssten, um die Vorgaben zu erfüllen. Als Antwort erwartet der Baustein, ob diese Forderung erfüllt wird oder nicht
- Kann die Anzahl bediente Stationen nicht erhöht werden, erfolgt eine Meldung an den Modul ‚Synchro‘, der diese Information im nächsten Berechnungsschritt verarbeitet
- Als ‚informelle‘ Möglichkeit können die Bediener die Bearbeitung eines Auftrags unterbrechen und einen anderen, dringlichen Auftrag vorziehen (wie bereits erwähnt, werden an den Handarbeitsplätzen präemptive Operationen durchgeführt)

4.4 Weitere Elemente des Konzepts

4.4.1 Zielkriterium des Gesamtsystems

BRÜGGEMANN beschreibt verschiedene mögliche Zielkriterien, die alle zumindest implizit mit der Zeit verhängt sind: Es kann beispielsweise der Produktionsfluss oder die Kapazitätsauslastung optimiert werden. Er spricht beim Auftreten solcher konkurrierender Ziele vom bekannten „Dilemma der Ablaufplanung“ mit den unterschiedlichen Zielsetzungen ‚Maximierung der Kapazitätsauslastung‘ und ‚Minimierung der Durchlaufzeit‘. Um die geforderten Ziele zu erreichen, werden gemäss SCHULTE in der industriellen Praxis meist sog. *Prioritätsregeln* als lokale Steuerungsstrategien eingesetzt, die die Reihenfolge von Aufträgen vor einer oder mehreren Arbeitsstationen bestimmen. Hierzu zählen im weiteren Sinne auch die beiden Regeln, nach denen der Modul ‚Synchro‘ planen soll.

Nun wird gezeigt, auf welche Weise verschiedene Vorschläge für einen Ablaufplan gemäss dem Zielkriterium aus Abschnitt 2.1.3 untereinander bewertet werden können, weshalb an dieser Stelle näher auf die Prioritätsklassen eingegangen wird. Sie bezeichnen unterschiedliche Dringlichkeitsstufen, für welche die in der Tabelle 4-1 aufgelisteten verbalen Regeln gelten.

| Prioritätsklasse | Verbale Regel |
|------------------|---|
| PriA | Fertigung im nächsten Zyklus einer Maschine. Das heisst, dass unter Umständen die Bearbeitung anderer Aufträge unterbrochen wird. |
| PriB | Der Auftrag muss im nächstmöglichen Zyklus gefertigt werden, es werden aber keine Bearbeitungen anderer Aufträge unterbrochen. |
| PriC | Im Prinzip wie PriB, allerdings wird bei Konflikten zwischen Aufträgen mit PriB und PriC der erste bevorzugt. |
| Pri9 - Pri0 | Mit diesen Aufträgen soll die verbleibende Kapazität unter Einhaltung der Restriktionen soweit möglich gefüllt werden. Bevorzugt werden die Aufträge höherer Priorität. |

Tabelle 4-1: Einfluss der Priorität auf die Behandlung eines Auftrages

Nicht im Verzug stehenden Aufträgen ist die Priorität *Pri0* zugeordnet. Je Verzugswoche erhöht sich diese Klassifizierung um eine Stufe, allerdings höchstens bis zu *Pri9*. Die manuell setzbaren Stufen sind *PriA*, *PriB* und *PriC*. Generell gilt, dass Aufträge mit manuell gesetzten Prioritäten auf jeden Fall sobald wie möglich zu bearbeiten sind. Die restlichen Aufträge dienen in diesem Fall dazu, die Maschinen unter Einhaltung der Belegungsrestriktionen zu füllen.

GEORGI stellt verschiedene Zielfunktionen für das job shop scheduling vor, einige davon haben für jeden Auftrag einen Gewichtungsfaktor. Analog dazu wird nun eine Zielfunktion formuliert, in der die *Durchlaufzeit* d_i eines Auftrags A_i (Zeit in Stunden, in welcher A_i in $S(0)$ ist) in Abhängigkeit seiner Priorität gewichtet wird, und zwar mit einem sog. *Prioritätsfaktor* pri_i . Die Güte eines Ablaufplanes P wird mit der Funktion $Z(P)$ bewertet, die möglichst klein werden soll (a entspricht dabei der Anzahl Aufträge, die in P eingeplant sind):

$$Z(P) = \sum_{i=1}^a (pri_i \times d_i) \quad [4-1]$$

In Tabelle 4-2 sind die im Rahmen der Algorithmen benutzten Werte für pri_i angegeben.

| Priorität von A_i | Wert von pri_i | Priorität von A_i | Wert von pri_i |
|---------------------|------------------|-----------------------|------------------|
| Priorität PriA | 10 | Priorität Pri9 - Pri5 | 3 |
| Priorität PriB | 7 | Priorität Pri1 - Pri4 | 2 |
| Priorität PriC | 5 | Priorität Pri0 | 1 |

Tabelle 4-2: Mögliche Werte des Prioritätsfaktors eines Auftrages

4.4.2 Auftragsmodellierung und Zuweisung zu den Systemelementen

Die Aufträge müssen derart modelliert werden, dass sie jederzeit einem der Systemelemente E3, E4 oder E6 zugeordnet werden können. Zudem werden eindeutige Regeln darüber benötigt, unter welchen Bedingungen sich die den Auftragszustand beschreibenden Attribute ändern. In Abbildung 4-7 ist das Konzept eines Auftragsmodells aufgezeigt. Im Kopf des Modells sind allgemeine, auftragspezifische Daten aufgeführt. Dazu gehören z.B. der Identifikationsschlüssel (Auftragsnummer) und aus der Auftragsbearbeitung resultierende Angaben wie z.B. die Losgröße.

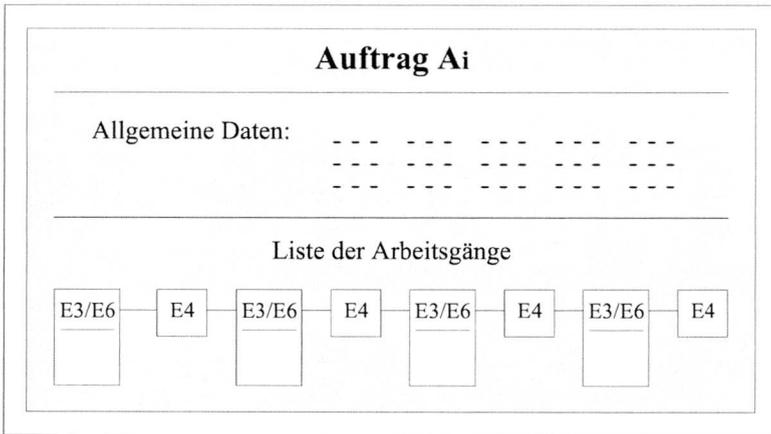


Abbildung 4-7: Struktur eines modellierten Auftrags

Aus dem zum Auftrag gehörenden Operationsplan wird zudem eine Liste der Arbeitsgänge generiert. Da wie gezeigt der Aufbau der Operationspläne nicht einheitlich ist, muss eine flexible Darstellungsform gewählt werden. Pro Arbeitsgang werden Kopfdaten benötigt, zusätzlich müssen für die Wärmebehandlung bzw. die Umformung spezifische Daten gespeichert werden. Deshalb werden in Abbildung 4-7 die Arbeitsgänge für E3 resp. E6 (inklusive Puffer) als Rechtecke gezeichnet, diejenigen für E4 als Quadrate. Die Rechtecke sind sog. ‚aktive‘ Arbeitsgänge (Attribut *AG-Typ*), die Quadrate ‚passive‘ Arbeitsgänge. Erfolgt bei einem Auftrag in E4 eine Zwischenbearbeitung, steht das Quadrat zudem für einen ‚echten‘ Arbeitsgang (Attribut *EchterAG*), im Gegensatz zu einem ‚falschen‘ Arbeitsgang, bei welchem keine Bearbeitung erfolgt: Bei dem ‚echten‘ Arbeitsgang muss gemeldet werden, wann der Auftrag fertig bearbeitet ist, ein falscher Arbeitsgang wird automatisch nach einer gewissen Zeit (für Transport etc.) fertig gemeldet.

Der Grund für die Unterscheidung in aktiv/passiv wird klar beim Betrachten der beiden folgenden Tabellen. Tabelle 4-3 zeigt die Attribute, mit welchen der Auftragsfortschritt eines Auftrages festgehalten wird.

| Attribut | Werte | Erklärung |
|------------|-----------------------|---|
| AG-Status | Wait InUse Done | AG nicht in Bearbeitung AG in Bearbeitung AG abgeschlossen |
| AG-Typ | Aktiv Passiv | AG ist Element E3 oder E6 AG ist Element E4 |
| EchterAG | Ja Nein | Beendigung des AG muss gemeldet werden ,Virtueller' AG, der nach bestimmter Zeit automatisch als beendet gemeldet wird |
| Eingeplant | Nein Ja | AG ist noch nicht eingeplant AG ist eingeplant |

Tabelle 4-3: Den Auftragsfortschritt kennzeichnende Attribute

Das Attribut ‚AG-Status‘ zeigt den Bearbeitungsstatus eines einzelnen Arbeitsganges an und hilft, den Auftrag zu lokalisieren. Der Auftrag befindet sich beim ersten Arbeitsgang in der Liste, dessen AG-Status nicht ‚Done‘ ist. Mit dem Attribut *Eingeplant* wird verhindert, dass ein Auftrag für einen Arbeitsgang zweimal eingeplant wird. Die Parameter ‚AG-Typ‘ und ‚EchterAG‘ werden für die Entscheidungstabelle benötigt, welche die Änderung des AG-Status regelt. Tabelle 4-4 gibt dazu eine Übersicht.

| Typ | Echt | Status | Ereignis |
|--------|------|--------|---|
| Aktiv | Ja | Wait | Ist der Status aller vorhergehenden AG's ‚Done‘, ist dieser AG zur Einplanung bereit. Eine Statusänderung (Bearbeitungsbeginn) muss gemeldet werden. |
| Aktiv | Ja | InUse | Eine Statusänderung (Done) muss gemeldet werden. |
| Aktiv | Ja | Done | Dieser AG ist abgeschlossen. |
| Aktiv | Nein | - | (Nicht möglich) |
| Passiv | - | Wait | Ist der Status aller vorhergehenden AG's ‚Done‘, wird der Status dieses AG's automatisch auf ‚InUse‘ gesetzt. |
| Passiv | Ja | InUse | Eine Statusänderung (Done) muss gemeldet werden. |
| Passiv | Nein | InUse | Eine Statusänderung (Done) erfolgt automatisch nach einer bestimmten Frist. |
| Passiv | - | Done | Dieser AG ist abgeschlossen. |

Tabelle 4-4: Entscheidungstabelle für die Attributsänderung

Ist in einem Feld ein Strich gesetzt, kann dort für das Attribut ein beliebiger (zulässiger) Wert eingesetzt werden. Gemäss dieser Entscheidungstabelle ist garantiert, dass maximal ein Arbeitsgang eines Auftrages den Status auf ‚InUse‘ gesetzt hat. Alle vorherigen AG's sind ‚Done‘, alle nachfolgenden ‚Wait‘. In Tabelle 4-5 ist beispielhaft die Attributsänderung einzelner Ar-

beitsgänge eines Auftrags gezeigt. In der ersten Spalte sind die Zeitabschnitte t_i angegeben, in welchen sich die betreffenden Elemente E im gezeigten Zustand befinden.

| t_i | E | Status | Eingeplant | Bemerkungen |
|-------|----|--------|------------|---|
| 1 | E3 | InUse | ja | Auftrag ist an den Öfen in Bearbeitung |
| 1 | E4 | Wait | - | Nächster Zw.arbeitsgang ist noch inaktiv |
| 2 | E3 | Done | ja | Wärmebehandlung ist abgeschlossen |
| 2 | E4 | InUse | - | Auftrag ist dem Zw.arbeitsgang zugewiesen |
| 3 | E4 | Done | - | Auftrag hat Zw.arbeitsgang verlassen |
| 3 | E5 | Wait | nein | Auftrag liegt bei der Presse bereit |
| 4 | E5 | Wait | ja | Auftrag ist bei der Presse eingeplant (Einplanung kann rückgängig gemacht werden!) |
| 5 | E6 | InUse | ja | Auftrag ist an der Presse in Bearbeitung |

Tabelle 4-5: Beispiel eines Auftragsfortschritts

Am Schluss ist der Auftrag an der Presse im selben Zustand, wie zu Beginn bei der Wärmebehandlung. Die Zeilen für den nicht mehr gezeigten Zeitabschnitt t_6 sehen aus wie jene des Zeitabschnittes t_2 (mit E6 statt E3).

Alle Aufträge aus der Menge A, deren Attribute sich im Zustand gemäss t_3 befinden, werden auf diese Weise der Presse für eine Einplanung zugewiesen. Dort werden sie dem Bediener getrennt nach Teilefamilien aufgelistet, worauf er gemäss den in den Entwicklungsstufen gezeigten „Fenster“-Modulen seine Wahl trifft.

TEIL II: SEQUENTIELLES 2D BIN PACKING

Teil II dieser Arbeit behandelt die Lösung des Nestingproblems entsprechend der in Abschnitt 2.3 vorgenommenen Teilung des Problems in eine kombinatorische und geometrische Aufgabe.

In Kapitel 5 wird eine Lösung für das kombinatorische Problem vorgestellt, mit welcher Kombinationen von Aufträgen - sogenannte Teilesets - zusammengestellt werden, deren Matrizen miteinander auf dem Trog der Fluidzellpresse plaziert werden sollen. Im Weiteren wird eine Strategie zur Zusammenstellung ganzer Folgen von Teilesets vorgestellt, die sicherstellen soll, dass die Leistung des Systems relativ konstant und möglichst hoch bleibt.

Kapitel 6 beschreibt die Lösung des geometrischen Teils des Nestingproblems. Es wird ein Ansatz vorgestellt, welcher die oben erwähnten Matrizen im Trog mit Hilfe eines Systems von Ungleichungen relativ zueinander positioniert. Formalisiert wird dieses Ungleichungssystem durch gerichtete und gewichtete Graphen. Dann werden Methoden aufgezeigt, die basierend auf diesen Graphen in einer bestehenden Anordnung freien Platz für zu legende Matrizen finden und nach Einfügen oder Entfernen von Matrizen die betreffende Anordnung entsprechend nachführen.

Kapitel 7 beschreibt und kommentiert die Tests an denjenigen implementierten Modulen, die in den Kapiteln 5 und 6 vorgestellt wurden.

Die folgenden Abschnitte dieser Einleitung zu Teil II gehen näher auf das Nestingproblem ein: Zuerst wird es durch den unten neu eingeführten Problemtyp des *sequentuellen 2D Bin Packings* formal beschrieben. Anschliessend werden DYCKHOFF & FINKE zitiert, die eine Typologie von Packungs- und Verschnittproblemen (vgl. unten) vorstellen, in der verschiedene Problemvarianten formal definiert werden. Von dieser Typologie wird nur ein Ausschnitt gezeigt, der vier Problemtypen beschreibt. Basierend darauf werden im dritten Abschnitt einige Ansätze aus der Literatur vorgestellt und bezüglich ihrer Brauchbarkeit bei der Lösung des vorliegenden Problems beurteilt.

Das Nestingproblem wird der Klasse von sogenannten Packungsproblemen zugeordnet, bei denen es darum geht, eine Menge von „kleinen“ Elementen in einem oder mehreren „grossen“ Elementen überschneidungsfrei anzuordnen. Beispiele dazu sind die Anordnung von Anlagen in Fabrikhallen, das Zusammensetzen eines Puzzles oder das Bestücken von Leiterplatten. In Analogie zu diesem Problemtyp existieren sogenannte Verschnittprobleme, bei denen möglichst viele vorgegebene Objekte aus einem Grundobjekt ausgeschnitten werden sollen. Praktische Verschnittprobleme sind das Zuschneiden von Blechen, Textilien oder Leder.

II.1 Das sequentielle 2D Bin Packing

„Klassisches“ 2D Bin Packing

Gegeben sind zwei Mengen von zusammenhängenden 2D-Elementen:

- Menge $T = \{T_1, \dots, T_\infty\}$ von „grossen“ Elementen T_i (mit den Dimensionen $T_i = L * B$)
- Menge $M = \{M_1, \dots, M_m\}$ von „kleinen“ Elementen.

Eine zulässige Lösung des 2D Bin Packing ist eine Partition

$$M = P_1 \cup P_2 \cup \dots \cup P_k, \quad P_i \cap P_j = \emptyset, \quad 1 \leq i < j \leq k,$$

sodass für jedes i die Elemente von P_i ohne Überschneidung im Element T_i platziert werden können.

Die zu minimierende Zielfunktion k ist die Anzahl grosser Elemente, die benötigt werden, um alle Elemente von M überschneidungsfrei zu packen (vgl. Abbildung II-1).

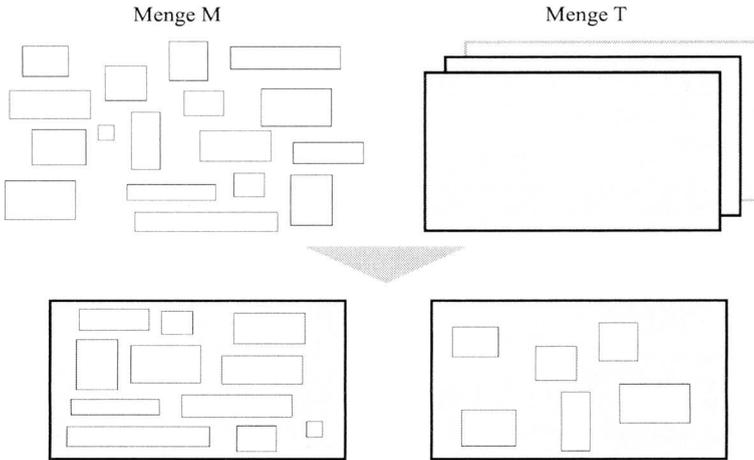


Abbildung II-1: Klassisches 2D Bin Packing

Sequentielles 2D Bin Packing

Bei der Einsatzoptimierung einer Fluidzellpresse trifft man auf die in den Kapiteln 2 und 3 gezeigte Problemstellung, für die Aufträge im Pufferlager eine Folge von Layouts zu generieren. Führt man sich diese spezielle Problemstellung vor Augen, wird offensichtlich: Die Formulierung des klassischen 2D Bin Packing reicht nicht aus, um das vorliegende Problem adäquat zu beschreiben. Deshalb wird an dieser Stelle das *sequentielle 2D Bin Packing* definiert. Gegeben sind zwei Mengen:

- Menge $T = \{T_1, \dots, T_\infty\}$ von „grossen“ Elementen T_i (mit den Dimensionen $T_i = L \cdot B$)
- Menge $A = \{A_1, \dots, A_r\}$, wo $A_j = \{M_j(1), \dots, M_j(n_j)\}$ eine Menge identischer kleiner Elemente ist. Sei $M = A_1 \cup A_2 \cup \dots \cup A_r$.

Eine zulässige Lösung des sequentiellen 2D Bin Packing ist eine geordnete Partition

$$M = P_1 \cup P_2 \cup \dots \cup P_k, \quad P_i \cap P_j = \emptyset, \quad 1 \leq i, j \leq k,$$

sodass gilt:

- Für jedes i können die Elemente von P_i ohne Überschneidung im Element T_i platziert werden.
- $|A_j \cap P_i| \leq 1, \quad \forall i, j$
- Für jedes i müssen die Elemente von A_i einer konsekutiven Folge von Elementen aus P zugeordnet sein. Formal geschrieben heisst das:

$$M_j(1) \in P_i \Rightarrow M_j(1+s) \in P_{i+s}, \quad 1 \leq s < n(j)$$

Bezogen auf die vorliegende Arbeit repräsentiert die Menge T den Trog, T_i entspricht der i -ten Pressung, A_j einem Auftrag mit $n(j)$ Pressungen und P_i den Matrizen der Aufträge, welche T_i zugeordnet sind. $M_j(k)$ entspricht der k -ten Pressung der Matrize von A_j . Abbildung II-2 zeigt die Mengen A und T .

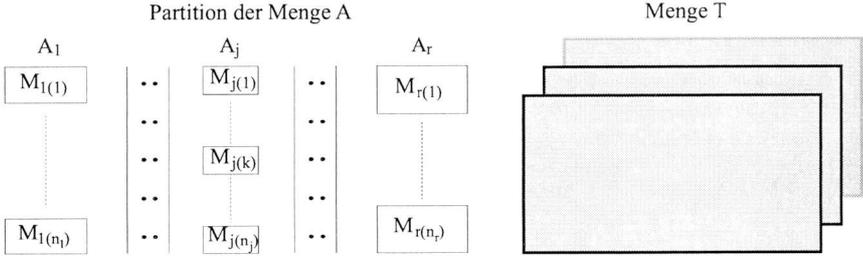


Abbildung II-2: Ausgangslage beim sequentiellen 2D Bin Packing

Die Zielfunktion: „Minimiere die Anzahl notwendiger Elemente T_p “ vom klassischen 2D Bin Packing ist hier bedeutungslos, da im betrachteten Fall eine rollende Planung vorliegt. Das heißt, bevor alle $M_j(k)$ einem T_i zugewiesen sind, könnte die Menge A um weitere Aufträge A_{r+1}, \dots, A_{r+x} ergänzt werden (in keinem Fall wird ein bereits existierender Auftrag A_j erweitert, n_j ist also konstant). An dieser Stelle können für das vorliegende Problem anstelle einer Zielfunktion lediglich fünf Gütekriterien formuliert werden (vgl. Abschnitt 2.3.2).

- Die Auslastung soll einen gegebenen Prozentsatz nicht unterschreiten (Auslastung: Fläche aller $M_j(k) \subset T_i$ im Verhältnis zur Fläche von T_i).
- Jedem A_j ist eine (terminliche) Gewichtung p_{rj} zugewiesen. Elemente $M_j(k)$ mit hohem p_{rj} müssen bevorzugt den P_i mit kleinem i zugewiesen werden.
- Die Positionen von $M_j(k)$ in T_i und $M_j(k+1)$ in T_{i+1} sollen möglichst identisch sein und die Orientierung soll wenn möglich erhalten bleiben. Abbildung II-3 zeigt mögliche (zulässige) Positionen und Orientierungen einer Matrize M_j .
- Es existieren Paare A_j, A_h , deren Matrizen möglichst nicht nebeneinander zu positionieren sind. Solche Anordnungen sind zwar zulässig, aber nicht erwünscht (hohe Matrizen).
- Es existieren Aufträge A_j , deren Matrizen möglichst wenig in gewisse Zonen von T_i hineinragen sollten (Tabuzonen).

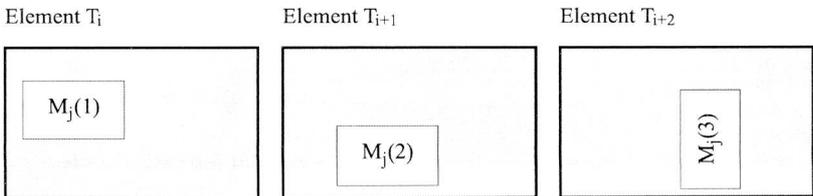


Abbildung II-3: Erlaubte Verschiebungen und Rotationen einer Matrize M_j

II.2 Klassifizierung nach Dyckhoff

In DYCKHOFF & FINKE wird eine ausführliche Einteilung von Packungs- und Verschnittproblemen vorgenommen, in der von der englischen Bezeichnung „Cutting- and Packing problems“ abgeleitet der Begriff *C&P-Probleme* benutzt wird. Die Autoren haben mit ihrer Arbeit drei Ziele verfolgt: Schaffen einer konsistenten Terminologie für Packungs- und Verschnittprobleme; Entwicklung einer allgemeinen Charakterisierungsmethode solcher Probleme, um aus unterschiedlichsten Anwendungsproblemen die logische Struktur des Problems herauszuschälen; Schaffen eines systematischen Literaturüberblickes.

Die Charakterisierung von C&P-Problemen basiert auf über einem Dutzend Attributen. Es wurde festgestellt, dass die benutzten Attribute nicht ausreichen, um das vorliegende Problem vollständig zu beschreiben. Insbesondere für die oben aufgeführten speziellen Restriktionen fanden sich keine Attribute, was die Definition des neuen Problemtyps ‚sequentielles 2D Bin Packing‘ rechtfertigt.

Im Folgenden wird aus der genannten Arbeit eine Typologisierung von C&P-Problemen zitiert, die auf wenigen Attributen beruht und welche erste Informationen zu Lösungsstrategien der einzelnen Typen liefert. Abbildung II-4 zeigt die Typologisierung (kleine Elemente werden ‚Teile‘ genannt, grosse Elemente heissen ‚Körper‘):

| Problemtypen | | Zuweisungstyp | |
|----------------|-----------|--------------------------|--------|
| | | Typ III oder IV | Typ II |
| Teilesortiment | homogen | | PL |
| | heterogen | wenige Teile pro Gestalt | BP |
| | | viele Teile pro Gestalt | CS |
| | | | KS |

Abbildung II-4: Typologisierung von C&P-Problemen nach DYCKHOFF & FINKE

Es werden vier Zuweisungstypen unterschieden, die für ein C&P-Problem angeben, wieviele der gegebenen Teile und Körper jeweils benutzt werden. Möglich sind die folgenden Kombinationen: alle Teile allen Körpern zuweisen (I), eine Auswahl der Teile allen Körpern zuweisen (II), alle Teile einer Auswahl der Körper zuweisen (III) und eine Auswahl der Teile einer Auswahl der Körper zuweisen (VI). Bei der vorliegenden rollenden Planung wird implizit (wegen der geforderten hohen Auslastung) eine Auswahl der Aufträge möglichst wenigen Pressungen zugewiesen, was dem Typ IV entspricht.

Das Teilesortiment bei der Fluidzellpresse ist heterogen, weil die Matrizen i.A. unterschiedliche Dimensionen aufweisen. Die Frage ‚wenige oder viele Teile pro Gestalt‘ wird an die Frage gekoppelt, ob genug ähnliche Formen vorhanden sind, um daraus repräsentative Gruppen zu bilden. Dies ist nicht der Fall (es liegen also wenige Teile pro Gestalt vor), womit das betrachtete Problem zur Gruppe der Bin Packing-Probleme (BP) gehört. Die einzelnen Problemtypen beschreiben DYCKHOFF & FINKE wie folgt:

Bin Packing (BP): Viele kleine heterogen geformte Teile, unter denen nur wenige die gleiche Gestalt haben, werden ganz oder teilweise einer Auswahl von Körpern zugewiesen. Diese Zuweisung ist i.A. komplex und wird als NP-komplett bezeichnet. Meist werden schnelle und einfache Heuristiken eingesetzt.

Cutting Stock (CS): Viele Teile mit wenigen unterschiedlichen Formen werden wiederum ganz oder teilweise einer Auswahl von Körpern zugewiesen. Im Gegensatz zum BP-Problem können die Teile in relativ wenige Gruppen identischer Gestalt eingeteilt werden. Daraus resultieren oft sich wiederholende Muster im Layout der Lösung. Diese Muster können einmal berechnet und dann des öfteren gebraucht werden. Die Problemkomplexität kann dann soweit reduziert werden, dass oft ein exaktes Verfahren eingesetzt werden kann, wenn Integralitätsbedingungen vernachlässigt werden.

Knapsack (KS): Aus einer grossen Menge von heterogen gestalteten Teilen muss eine (relativ kleine) Auswahl getroffen werden, die einer limitierten Menge von Körpern zugewiesen werden sollen. Alle verfügbaren Körper sollen verwendet werden. Oft werden Branch and Bound-Verfahren oder solche der dynamischen Programmierung zur Lösung von KS-Problemen eingesetzt.

Pallet Loading (PL): Hier muss eine Auswahl der Teile von homogener Gestalt verschiedenen Körpern zugewiesen werden. Da die Körper meist ebenfalls von homogener Gestalt sind, reicht eine repräsentative Lösung für diese Aufgabe.

II.3 Ansätze aus der Literatur

HEISTERMANN teilt Veröffentlichungen zum Thema ‚Nesting‘ in drei Kategorien ein:

- Theoretische Arbeiten ohne Anspruch auf praktische Relevanz
- Heuristiken mit dem Anspruch auf praktische Anwendbarkeit und
- Übersichtsarbeiten für den gesamten Bereich Packen und Verschnitt

Die vorliegende Arbeit wird eindeutig der zweiten Kategorie zugeteilt. Umfangreiche Übersichten finden sich genügend in Arbeiten aus der ersten und insbesondere aus der dritten Kategorie, zu welcher auch diejenige von DYCKHOFF & FINKE zählt. Deshalb werden in diesem Abschnitt nur wenige in der Literatur gezeigten Ansätze vorgestellt und bezüglich ihrer Brauchbarkeit für das sequentielle 2D Bin Packing beurteilt.

Zu Beginn wird eine Arbeit zitiert, welche das 2D Bin Packing Problem anders definiert als in dieser Arbeit. Im Verlaufe der nächsten Absätze wird aber klar, dass die gewählte Einteilung sinnvoll ist.

NELISSEN beschreibt unter dem zweidimensionalen Bin Packing folgendes Problem (die Mengen T, M sind gleich definiert wie beim klassischen Bin Packing): T besteht aus einem einzigen Rechteckstreifen mit gegebener Breite und nicht bestimmter (unendlicher) Höhe. Die Aufgabe besteht darin, die Menge M derart in T anzuordnen, dass die Höhe von T minimal wird. Es wird nicht erwähnt, ob in M identische Elemente vorkommen. Weiter wird ein Cutting Stock Problem definiert: Dabei sollen aus einem gegebenen Rechtecktyp T_r verschiedene (kleine) Rechtecktypen M_j in Stückzahlen n_j ausgeschnitten werden. Es sind nur orthogonale Schnitte zugelassen und es sollen möglichst wenig Grundplatten zerschnitten werden.

Bezüglich diesen Beschreibungen deckt sich das betrachtete Problem eher mit dem von NELISSEN ‚Cutting Stock‘ genannten Problem, allerdings werden die Restriktionen, welche zur Defi-

nition eines sequentiellen Problems führten, nicht berücksichtigt. Zudem verwenden die von NELISSEN gezeigten Lösungsmethoden ebenfalls den Ansatz, kleine Elemente identischer Gestalt zusammenzufassen und damit sich oft wiederholende Muster zu generieren. Im Folgenden werden Ansätze zum Packen eines unendlich langen Rechtecks T gezeigt, anschliessend Ansätze zum Packen einer Anzahl endlicher Rechtecke T_i .

Heuristiken zum Packen eines unendlich langen Rechtecks T

Unter diesen Heuristiken werden hier Methoden verstanden, welche Layouts wie folgt generieren: „Sortiere erstens die Matrizen nach einem Kriterium x “ (beispielsweise nach der Priorität von A_j oder nach der Länge der Matrize M_j); Zweitens werden die M_j sequentiell gelegt, indem der verbleibende Platz mit Hilfe eines einfachen Musters abgesucht wird: „Plaziere die nächste noch nicht gelegte Matrize M_j an der erstmöglichen Stelle, welche gemäss Muster y gefunden wird“. Beispiele solcher Heuristiken sind:

Bottom Up - Left Justified - Heuristiken (BL): M_j wird im Streifen so weit unten wie möglich und auf dieser Höhe so weit links wie möglich plaziert wird. Typischerweise werden die Elemente in M vor dem Legen nach auf- oder absteigender Höhe (bzw. Länge) oder Breite sortiert, was den jeweiligen Verfahren den Namen gibt (beispielsweise BLDH für Bottom up, Left justified Decreasing Height).

Verfeinerte BL-Heuristiken: Diese Heuristiken arbeiten oft mit sogenannten Leveln oder Shelves. T wird dabei in (waagrechte) Segmente unterteilt und M_j muss vollständig in einem solchen Segment plaziert werden können. Die Höhe dieser Segmente wird beispielsweise vom ersten darin gelegten Element bestimmt oder aufgrund einer Auswertung sämtlicher in M vorhandenen Elemente berechnet.

*Mehrstufige Algorithmen*¹: Oft werden mehrere einfache Strategien nacheinander eingesetzt. NELISSEN beispielsweise zitiert eine Methode, bei welcher die Elemente M nach geometrischen Kriterien in fünf Untermengen eingeteilt werden, dann wird jede dieser Untermengen nach einem eigenen Algorithmus plaziert.

Diese Strategien können auch zum Füllen mehrerer Rechtecke endlicher Höhe angewendet werden, indem man das unendliche Rechteck unter Benützung der Troglänge in „Hauptlevel“ unterteilt. Man findet mit derartigen Strategien sicherlich nicht auf Anhieb gute Lösungen. Da sie sehr einfach sind, können damit aber in kurzer Zeit viele verschiedene Layouts berechnet werden, bei denen durch einen geeigneten Vergleich der Vorschläge der „beste“ ausgewählt werden kann. Unterschiedliche Vorschläge entstehen durch Änderung der Reihenfolge, in der die Matrizen plaziert werden. Eine geeignete Methode zur wiederholten und zielgerichteten Änderung dieser Reihenfolge ist zum Beispiel das Simulated Annealing (vgl. hierzu zum Beispiel HEISTERMANN).

Keine dieser Methoden berücksichtigt die Positionen derjenigen Matrizen, die bereits in der vorhergehenden Pressung im Trog lagen. Genau aus diesem Grund ist ihre Anwendung auf das vorliegende Problem ungeeignet.

1. Mehrstufige Algorithmen werden überall verwendet, wo sich ein Problem in Teilprobleme unterteilen lässt. Vgl. TERNO ET AL.

Heuristiken zum Packen mehrerer endlicher Rechtecke T_i

Deterministisches sequentielles Anordnen: TERNO ET AL. erwähnen eine Methode, bei der aus der vorsortierten Menge der noch nicht plazierten Elemente für das nächste Element M_j mehrere mögliche Anordnungen in T untersucht werden und die ‚beste‘ dieser Anordnung in T gewählt wird. Eine Erweiterung dieser Methode arbeitet mit einem Vorausschauparameter k , bei dem für die nächsten k noch nicht plazierten Elemente mehrere mögliche Anordnungen geprüft werden.

Dieser Ansatz wird oft eingesetzt zur Lösung von Packungsproblemen, insbesondere auch beim Packen beliebiger Formen in eine Grundfläche (wo durch Drehen eines untersuchten Elements an derselben Stelle in T bereits unterschiedliche Anordnungen entstehen).

Viele Probleme sind gemäss den oben ausgeführten Definitionen Cutting Stock-Probleme und werden normalerweise mit dem bereits erwähnten Ansatz gelöst, bei dem oft sich wiederholende Muster eingesetzt werden. Diese Muster werden beispielsweise mit dem eben beschriebenen sequentiellen Anordnen positioniert.

Eine interessante Methode stellt HEISTERMANN vor, bei dem die Grundfläche allerdings kein Rechteck ist, sondern eine Tierhaut, welche zu Leder verarbeitet werden soll. An die Lederstücke, welche man mit sogenannten Schablonen aus der Haut schneidet, werden unterschiedliche Qualitätsanforderungen gestellt, was beim Plazieren der Schablonen berücksichtigt werden muss. Beim präsentierten Ansatz werden die Schablonen ebenfalls sequentiell gelegt, und zwar immer an den Rand der noch freien Fläche. Diese wird durch mindestens ein Polygon beschrieben, welches nach dem Plazieren einer Schablone jeweils neu bestimmt werden muss. HEISTERMANN bestimmt mit Hilfe des Polygons zuerst eine Zone, in welche die nächste Schablone gelegt werden soll. Dann werden für diese Zone einige Schablonen ausgewählt und probeweise plaziert. Im dritten Schritt wird eine Schablone definitiv bestimmt und plaziert, worauf das Polygon nachgeführt werden muss.

Die in diesem Abschnitt vorgestellten Verfahren prüfen verschiedene Varianten, bevor ein Element plaziert wird. Dieses gezielte Plazieren scheint auch für das sequentielle 2D Bin Packing ein sinnvoller Ansatz zu sein, v.a. bezüglich des Handlingsaufwands, welcher bei der Presse möglichst gering gehalten werden soll. Der Aufwand kann beim Bewerten verschiedener möglicher Plätze berücksichtigt werden, ebenso die in Abschnitt 3.3 formulierten Beschickungsregeln.

Alle gefundenen Methoden weisen den Elementen fixe Plätze zu. Das heisst, man hat i.A. nicht die Möglichkeit, Elemente ein wenig zu verschieben, um andern Elementen Platz zu machen. Manchmal existieren relativ aufwendige Nachoptimierungen, um Anordnungen kompakter zu machen, allerdings werden diese erst nach Abschluss der eigentlichen Plazierungsalgorithmen eingesetzt. Da beim sequentiellen 2D Bin Packing oft Matrizen aus der letzten Pressung übernommen werden müssen, die nicht stark verschoben werden sollen, wird deshalb in Kapitel 6 ein Ansatz vorgestellt, der kleine Verschiebungen ohne zusätzlichen Rechenaufwand durchführt.

Die Auswahl der Elemente, welche zusammen gepackt werden sollen, ist in den zitierten Ansätzen nie als eigenes Thema behandelt worden, weil i.A. alle vorliegenden Elemente dem zu packenden oder einem der nächsten grossen Elemente zugewiesen werden müssen.

Bemerkung: Das vorliegende Problem kann auch als 3D Packungsproblem definiert werden, wenn zusätzlich folgende Restriktion eingeführt wird: Die Lage und Orientierung von $M_{j(k)}$ in T_i muss identisch sein mit der Lage und Orientierung von $M_{j(k+1)}$ in T_{i+1} für $1 \leq k \leq n(j)$. Das heisst, eine Matrize darf während der ganzen Bearbeitungszeit nicht verschoben oder gedreht

werden (vgl. Abbildung II-3, wo diese Änderungen erlaubt sind). Legt man gedanklich alle T_1 gemäss ihrer Reihenfolge übereinander (und versehe sie mit einer Dicke), erhält man einen Grundquader, in welchen kleine Quader (die Aufträge) gepackt werden müssen. In diesem Fall soll die Packungsdichte maximiert werden, bzw. die Höhe des Grundquaders minimiert werden, wenn keine rollende Planung vorliegt.

Für eine reduzierte Problembeschreibung dieses 3D Packungsproblems führte GOBET an den Instituten für Operations Research und für Umformtechnik der ETH Zürich eine Semesterarbeit durch. Die gewählte Problembeschreibung hat die Menge der Lösungen zu stark beschränkt, weshalb der Ansatz in der erwähnten Arbeit hier nicht verwendet werden kann.

5 Bestimmung von Teilesets

In diesem Kapitel werden zuerst die Funktionen des Moduls ‚Teilesetwahl‘ behandelt. Es wird gezeigt, wie aus den im Pufferlager bereitliegenden Aufträgen eine Untermenge zusammengestellt wird, deren Matrizen miteinander auf die Presse kommen und die die bereits auf dem Trog liegenden Matrizen für die nächsten Pressungen ergänzen. Eine solche Untermenge wird Teileset TS genannt und mit Hilfe einer Bewertung aus einer Familie von Untermengen von Aufträgen ausgewählt. Im Anschluss an diese Ausführungen werden zwei Strategien gezeigt, wie eine Folge von Teilesets bestimmt wird. Diese Funktion ist dem ‚Nestingfenster‘ zugeordnet.

Gliederung des Kapitels

In Abschnitt 5.1 werden zuerst Input und Output des Moduls ‚Teilesetwahl‘ (siehe Abbildung 4-3) aufgezeigt, um dann die Zielfunktion herzuleiten. Darauf aufbauend wird in Abschnitt 5.2 der gewählte Lösungsansatz vorgestellt: Die vorliegenden Aufträge werden nach gewissen Kriterien geordnet, worauf Untermengen von Aufträgen, sogenannte *Teilesetkandidaten TSK*, gebildet werden. Da eine vollständige Enumeration aller möglichen TSK sinnlos und nicht realisierbar wäre, steht folgende Frage im Vordergrund: Wie kann die Zahl jener TSK klein gehalten werden, die generiert und eventuell genestet werden, um eine gute Lösung zu finden? Zu diesem Zweck werden Teile eines Enumerationsbaumes gebildet, bei welchem jeder Knoten einen TSK darstellt. Aufgrund der gebildeten Ordnung und mit Hilfe von Zulässigkeitschranken kann die Suche auf interessante Teile des Baumes beschränkt werden.

Abschnitt 5.3 zeigt, wie eine Folge von Teilesets - und somit eine Folge von Layouts - bestimmt wird. Es wird gezeigt, dass die Systemleistung (d.h. das Erreichen einer hohen Auslastung mit Aufträgen hoher Priorität) mit der Zeit sinkt, wenn jede entstandene Lücke im Trog¹ sofort gefüllt wird. Als Alternative wird ein Ansatz präsentiert, der Aufträge mit hoher Priorität und grosser Grundfläche vorausschauend plaziert und somit die Systemleistung über längere Zeit relativ konstant bleiben lässt. Das Generieren von Layoutfolgen wird vom Modul ‚Nestingfenster‘ aus gesteuert (vgl. Abbildung 4-3).

5.1 Anforderungen und Zielfunktion des Moduls ‚Teileset‘

Die Teilesetwahl wird dann ausgelöst, wenn der Vorarbeiter Vorschläge für Layouts generieren will. Im Allgemeinen sind für einen Trog bereits einige Layouts geplant. Entweder sollen diese geplanten Layouts ergänzt, d.h. mit neuen Aufträgen aufgefüllt werden, oder es sollen Vorschläge für neue Layouts gemacht werden. Eine geplante Folge von Layouts bzw. von Teilesets wird in der sogenannten *Teilesetliste TSL* festgehalten.

5.1.1 Input

Gemäss Abschnitt 4.3.1 zeigt der Modul ‚Nestingfenster‘ dem Vorarbeiter statistische Angaben über die bereitliegenden Aufträge im System. Daraufhin entscheidet er, welche Teilfamilie auf welchem Pressentrog einzuplanen ist und legt fest, wieviele Pressungen vorgeschlagen werden sollen. Aufgrund dieser Vorgaben stellt das ‚Nestingfenster‘ pro Trog die in Frage kommenden Aufträge zusammen und übergibt diese der ‚Teilesetwahl‘. Dort wird genau ein Teileset zusam-

1. Lücken entstehen, wenn Matrizen herausgenommen werden, weil ihre Aufträge fertig bearbeitet sind.

mengestellt und wieder an das ‚Nestingfenster‘ zurückgegeben. Sollen mehrere Teilesets nacheinander (eine sog. *Teilesefolge*) erzeugt werden, erfolgt der Aufruf der ‚Teilesewahl‘ mehrere Male. Auf diesen Ablauf wird in Abschnitt 5.3 eingegangen.

Die Inputdaten sind in Tabelle 5-1 zusammengefasst: Überwiesen werden der ‚Teilesewahl‘ neben Angaben zum Trog T je eine Menge von Aufträgen $A_{(E5)}$ und $A_{(E6)}$. Alle Aufträge der Untermengen gehören derselben Teilefamilie an und sind E5 (Puffelager) bzw. E6 (Presse) zugewiesen. Ein weiterer Input ist TS_j , also das Teileset, welchem Aufträge aus $A_{(E5)}$ zugewiesen werden sollen. Hält TS_j bereits Aufträge (früherer Pressungen), sind die entsprechenden Matrizen sowie ihre Positionen im Trog bekannt.

| Input | Attribute | Bemerkungen |
|--|--|--|
| $A_{(E5)}$: $A_{(E6)}$: | $= \{A_{1(E5)}, \dots, A_{a(E5)}\}$ $= \{A_{1(E6)}, \dots, A_{a(E6)}\}$ | Aufträge der gewählten TF, die E5 zugewiesen sind Aufträge der gewählten TF, die E6 zugewiesen sind |
| Pro $A_i(E5)$ bzw. pro $A_i(E6)$: | M_i w_i Pri_i $Termin_i$ Eingeplant: ja/nein | Zu A_i gehörende Matrize M_i Anzahl notwendige Pressungen von A_i Priorität von A_i Spätester Umformtermin von A_i Zeigt, ob A_i bereits eingeplant ist. |
| Pro M_i : | u_i, v_i h_i Tabuzonen TZ_i | Umhüllende der Matrize M_i Höhe der Matrize M_i Tabuzonen von M_i |
| Trog T: | L_T, B_T $f_{MinAusl}$ $TSL = \{TS_1, \dots, TS_n\}$ TS_j | Länge (L_T) und Breite (B_T) des Troges Geforderte minimale Auslastung Teilesetliste: Bereits geplante Folge von Teilesets TS_j , das zusammengestellt bzw. ergänzt werden soll |
| Von TS_j | $\{M_1, \dots, M_m\}$ L_j $AnzP_j$ | Menge der Matrizen, welche TS_j zugewiesen sind Zu TS_j gehörendes Layout L_j (Positionen der Matrizen) Anzahl Pressungen des Teilesets TS_j |

Tabelle 5-1: Input des Moduls ‚Teilesewahl‘

Bemerkung: Das in der Tabelle aufgeführte Auftragskriterium ‚Termin‘ wird bei den ersten beiden Entwicklungsstufen des in Kapitel 4 vorgestellten Konzepts alleine durch das Lösungsglücken gesetzt. Bei der dritten Stufe ergeben sich durch die provisorische Einplanung über alle Arbeitsgänge im System Fälligkeitzeitpunkte, die ebenfalls berücksichtigt werden müssen.

Dem Modul ‚Nestingfenster‘ wird die ganze TLS übergeben und nicht nur TS_j : Erstens muss für $j>1$ auch TS_{j-1} übergeben werden. Zweitens wird im nächsten Abschnitt gezeigt, dass durch ein als TS_j akzeptiertes TSK im Allgemeinen die folgenden Teilesets auch verändert werden.

5.1.2 Output

Wurde für ein TSK ein zulässiges Layout gefunden und dieser Vorschlag von der Entscheidungsinstanz in $S(0)$ als nächstes Teileset TS_j akzeptiert, wird der Input wie folgt geändert und an $S(0)$ zurückgegeben:

- Bei Aufträgen aus $A_{(E5)}$, die neu einem TS_j zugewiesen sind, ändert das Attribut ‚Eingepplant‘ von ‚nein‘ auf ‚ja‘. Der Benutzer kann dieses Attribut wieder auf ‚nein‘ setzen, der betreffende Auftrag ist dann keinem Teileset mehr zugeteilt. Nur wenn für ein oder mehrere Layouts angegeben wird, dass sie effektiv in Bearbeitung sind, werden die entsprechenden Aufträge aus $A_{(E5)}$ entfernt und $A_{(E6)}$ zugeteilt.
- In TSL wird TS_j neu eingefügt bzw. angepasst. Dazu gehören die Menge der zu den Aufträgen gehörenden Matrizen $\{M_1, \dots, M_m\}$, das entsprechende Layout L_j sowie die Anzahl Pressungen $AnzP_j$. Auch die auf TS_j folgenden Teilesets $\{TS_{j+1}, \dots, TS_n\}$ und die entsprechenden Layouts $\{L_{j+1}, \dots, L_n\}$ müssen neu eingefügt bzw. angepasst werden

Die Änderung der auf TS_j folgenden Teilesets und ihrer Layouts hat folgenden Grund: Jedesmal, wenn eine Matrize aus dem Trog entfernt werden kann, liegt ein neues Teileset vor. Für einen noch leeren Trog werde beispielsweise das erste Layout L_1 gefunden und in TS_1 seien m Aufträge enthalten. Haben alle Aufträge in TS_1 dieselbe Losgrösse, wird nur L_1 generiert. Haben aber alle Aufträge in TS_1 unterschiedliche Losgrößen, so werden m-1 weitere, von TS_1 abgeleitete Teilesets und Layouts generiert.

5.1.3 Zielfunktion

Ziel des Gesamtsystems ist, die Durchlaufzeiten der Aufträge - gewichtet nach Prioritäten - zu minimieren. Für einen Layoutvorschlag wurden folgende verbalen Gütekriterien genannt:

- Die durchschnittliche Priorität der Aufträge eines Teilesets soll möglichst hoch sein.
- Der Handlingsaufwand soll möglichst klein sein.

Im Weiteren wurden Beschickungsregeln aufgezeigt, welche sich auf die Lage hoher Matrizen im Trog beziehen. Das ergibt für die Zielfunktion beim Nestingmodul folgende *Anforderungen*:

1. Ein Teileset muss in Funktion der Prioritäten seiner Aufträge bewertet werden.
2. Die Auslastung muss ebenfalls in die Bewertung einfließen, um die Durchlaufzeit der Aufträge tief zu halten.
3. Die Anzahl hoher Matrizen in einem Teileset muss berücksichtigt werden.
4. Die Positionen derselben Matrize in den aufeinanderfolgenden Layouts müssen betrachtet werden, um den Handlingsaufwand minimieren zu können.
5. Bei hohen Matrizen muss ihre absolute Position zum Trog sowie ihre relative Position zu andern hohen Matrizen berücksichtigt werden.

Mit diesen Anforderungen wird nun eine Zielfunktion für ein einzelnes TS_j definiert. Auf eine Zielfunktion für eine Folge von genesteten Teilesets wird in Abschnitt 5.3.1 eingegangen.

Zielfunktion eines Teilesets TS_j :

Die Zielfunktion eines genesteten Teilesets setzt sich aus zwei jeweils nicht negativen Teilen zusammen: $SetValue(TS_j)$ berücksichtigt die ersten drei oben aufgezählten Anforderungen, hängt folglich nur vom Teileset TS_j ab und nicht von einem zugehörigen Layout. Deshalb der Name (Teile-)SetValue. Der Wert ist umso grösser, je besser ein Teileset ist. Der zweite Teil der Zielfunktion heisst $LayoutValue(TS_j)$ und berücksichtigt die oben aufgeführten Punkte 4 und 5. Für dessen Berechnung werden Layoutvorschläge benötigt, weshalb dieser Teil der in Gleichung [5-1] gezeigten Zielfunktion erst in Kapitel 6 ausführlich behandelt wird.

$\text{LayoutValue}(\text{TS}_j)$ ist eine Penaltyfunktion, denn dieser Wert ist gross für Layouts, welche die gestellten Anforderungen schlecht erfüllen. Den Gesamtwert $\text{Value}(\text{TS}_j)$ eines Teilesets erhält man, indem $\text{LayoutValue}(\text{TS}_j)$ von $\text{SetValue}(\text{TS}_j)$ subtrahiert wird.

$$\text{Value}(\text{TS}_j) = \text{SetValue}(\text{TS}_j) - \text{LayoutValue}(\text{TS}_j) \quad [5-1]$$

Auf diese Weise liegt ein Kriterium vor, mit welchem der relativ aufwendige und deshalb zeitraubende Nestingalgorithmus nicht für jeden TSK aufgerufen werden muss. Dieser Fall tritt bei folgender Situation ein: Für das zu generierende Teileset TS_j wurde bereits mindestens eine zulässige Lösung gefunden. Die beste dieser Lösungen wird TSK_{best} genannt, und es wird ein weiteres Teileset TSK_{neu} überprüft. Nun wird der Teilwert $\text{SetValue}(\text{TSK}_{\text{neu}})$ ermittelt. Falls

$$\text{SetValue}(\text{TSK}_{\text{neu}}) \leq \text{Value}(\text{TSK}_{\text{best}}) \quad [5-2]$$

ist, dann gilt wegen $\text{LayoutValue}(\text{TS}_j) > 0$ auf jeden Fall

$$\text{Value}(\text{TSK}_{\text{neu}}) < \text{Value}(\text{TSK}_{\text{best}}) \quad [5-3]$$

und für TSK_{neu} muss der Modul ‚Layoutgenerierung‘ nicht aufgerufen werden. Somit bildet $\text{SetValue}(\text{TSK}_{\text{best}})$ eine *untere Schranke*, die erfüllt sein muss, damit überhaupt Layouts gesucht werden.

Die Definition von $\text{SetValue}(\text{TS}_j)$ ist in Gleichung [5-4] gegeben. pri_i ist der Prioritätsfaktor eines Auftrags A_i (vgl. Tabelle 4-2), welche jeweils mit der Fläche der Umhüllenden von A_i multipliziert wird. Die davon gebildete Summe wird $\text{PriValue}(\text{TS}_j)$ genannt. Der Faktor $f_{\text{HM}(j)}$ ist eine Kennzahl, welche die Anzahl hoher Matrizen (HM) in einem Teileset berücksichtigt.

$$\text{PriValue}(\text{TS}_j) = \sum_{i=1}^m (\text{pri}_i \times u_i \times v_i) \quad [5-4]$$

$$\text{SetValue}(\text{TS}_j) = f_{\text{HM}(j)} \times \text{PriValue}(\text{TS}_j)$$

Matrizen mit grosser Grundfläche werden also höher bewertet, da sie schwieriger zu plazieren sind als solche mit kleinerer Grundfläche. Der erste Teil von [5-4] wird später als Abbruchkriterium benutzt. Sämtliche Dimensionen sind in mm angegeben.

Zur Berechnung von $f_{\text{HM}(j)}$ wird eine Abschätzung gemacht, wie viele hohe Matrizen durchschnittlich pro Pressung im Trog liegen, bis alle vorliegenden Aufträge abgearbeitet sind. Je näher der reale Anteil hoher Matrizen eines Teilesets TS_j an diesem Schätzwert ist, umso grösser ist $f_{\text{HM}(j)}$. Die Berechnung des Faktors $f_{\text{HM}(j)}$ kann in Abschnitt 5.2.4 nachgelesen werden. Es gilt $f_{\text{HM}(j)} \leq 1$.

5.2 Bestimmung eines Teilesets

5.2.1 Grundidee des Moduls ‚Teileset‘

Aus den q *Aufträgen* der Untermenge $A_{(E5)}$ soll ein Teileset TS_j zusammengestellt bzw. ergänzt werden. Ergänzt heisst, dass in TS_j noch nicht abgearbeitete Aufträge aus TS_{j-1} vorliegen, welche im Teileset TS_{root} zusammengefasst werden. Im letzten Abschnitt wurde gezeigt, nach wel-

chen Kriterien die Wahl von TS_j erfolgt, insbesondere wurde eine untere Schranke bei der Bewertung möglicher Teilesets eingeführt.

Mit der Bedingung, dass ein Auftrag A_i aus $A_{(E5)}$ maximal einmal im gesuchten TS_j enthalten sein kann, ergeben sich unabhängig von den Losgrößen der Aufträge aus $A_{(E5)}$ $2^q - 1$ mögliche Auftragskombinationen, die sogenannten Teilesetkandidaten TSK. Diese TSK werden jeweils mit den Aufträgen aus TS_{root} ergänzt und - falls sie die untere Schranke überbieten ([5-2] und die minimale Auslastung) - zu schachteln versucht. Mit der hier vorgestellten Grundidee verfolgt man das Ziel, die Zahl der geprüften und eventuell genesteten TSK bei der Suche nach bestmöglichen TSK aus folgenden Gründen zu beschränken:

1. Das Zusammenstellen eines TSK braucht verhältnismässig wenig Zeit, bei einem typischerweise zwischen 100 und 200 liegenden q muss aufgrund der Menge möglicher TSK die Anzahl zusammengestellter TSK trotzdem tief gehalten werden.
2. Ein Nestingaufruf braucht relativ viel Zeit, weshalb die Anzahl genesteter TSK deutlich tiefer sein soll, als die Menge der überprüften TSK.

Es wird deshalb nach folgendem Ansatz vorgegangen:

- Alle $2^q - 1$ möglichen TSK können gedanklich als Knoten eines Enumerationsbaumes dargestellt werden. In der Praxis wird immer nur ein Knoten betrachtet, zudem sind Regeln zur Generierung von Nachbarsknoten bekannt. Durch eine geschickte lexikographisch basierte Ordnung der Aufträge, welche die Position eines TSK im Baum festlegt, kann eine Region des Baumes bestimmt werden, in der sich TSK mit potentiell hohen Gesamtwerten $Value_{TSK}$ befinden.
- Die geforderte Minimalauslastung und die zulässige Maximalauslastung (100%) ergeben Schranken. Es genügt deshalb, auf jedem Ast des Baumes ein Intervall zu betrachten.
- Der Teilwert $SetValue(TSK_k)$ eines TSK_k wird mit dem Gesamtwert $Value(TSK_{best})$ des besten bisher gefundenen TSK verglichen. Trifft Ungleichung [5-2] zu, muss TSK_k nicht genestet werden.

Gewählt wird schliesslich derjenige TSK, welcher den besten Gesamtwert $Value(TSK_{best})$ aufweist. Ein TSK_k kann mehrere Male genestet werden, wobei dafür gesorgt wird, dass verschiedene Layoutvorschläge resultieren können. Auf diesen Punkt wird in Kapitel 6 näher eingegangen.

5.2.2 Der Enumerationsbaum

Lexikographische Ordnung des Baumes

Die Aufträge in $A_{(E5)}$ werden mittels einer lexikographischen Ordnung, welche auf drei Kriterien beruht, linear geordnet:

1. Kriterium: Priorität (absteigend). Das heisst, zuoberst in der Auftragsliste befinden sich die Aufträge mit der höchsten Priorität
2. Kriterium: Der Fälligkeitszeitpunkt. Trifft ein, sofern die Auftragsbearbeitung am Tage der Berechnung gestartet werden muss (Termin = ‚heute‘)
3. Kriterium: Fläche der Umhüllenden (absteigend). Grosse Matrizen sind schwieriger zu platzieren als kleine

Die Kriterien 1 und 3 tragen auch zum $\text{SetValue}(\text{TS}_j)$ bei und sorgen dafür, dass bekannt ist, in welcher Region des Enumerationsbaumes TSK mit potentiell hohen Werten zu finden sind. Kriterium 2 soll erstens verhindern, dass Aufträge mit kleinen Matrizen und anfänglicher Priorität $\text{Pri}0$ nicht erst eingeplant werden, wenn ihre Priorität auf $\text{Pri}1$ heraufgesetzt wird. Zweitens muss bei lösungsgeglühten Platinen die Umformung stattfinden, bevor die Platinen wieder ausgehärtet sind (dieses Kriterium kommt selten zum Tragen, muss aber trotzdem berücksichtigt werden).

Zur Veranschaulichung kann die Menge aller Kombinationen von $A_{(E5)}$ als Baumstruktur dargestellt werden, wie Abbildung 5-1 zeigt. Die Knotennummer ist zusammengesetzt aus den Auftragsnummern, die der Knoten repräsentiert, wobei die Auftragsnummern jeweils der Position des Auftrags in der lexikographisch geordneten Auftragsliste entsprechen. Beispielsweise besteht Knoten 125 aus den Aufträgen $\{A_1, A_2, A_5\}$. In der ersten Ebene des Baumes liegen alle TSK, die einen Auftrag enthalten, in der zweiten Ebene jene mit zwei Aufträgen und der TSK in der q -ten Ebene entspricht der Menge $A_{(E5)}$.

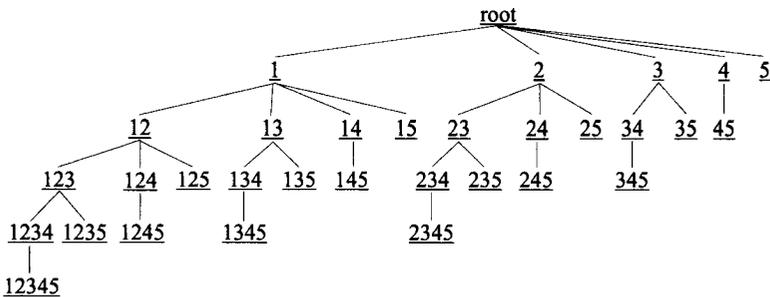


Abbildung 5-1: Lexikographisch geordneter Enumerationsbaum mit $q = 5$

Der Baum selbst wird gar nicht aufgebaut, es wird immer nur ein Knoten k betrachtet, der *aktiver Knoten* genannt wird. Um von einem Knoten zum nächsten zu gelangen, wird eine von drei später vorgestellten Verzweigungsregeln angewendet. Die Wahl der Regel wird durch verschiedene Merkmale des aktiven Knotens bestimmt: Konnte bereits ein zulässiges Layout gefunden werden, liegt der aktive Knoten innerhalb oder ausserhalb der Zulässigkeitsschranken, wie hoch ist sein Teilwert $\text{SetValue}(\text{TSK}_k)$ etc.

Schranken im Enumerationsbaum

Da die Aufträge primär nach ihrer Priorität geordnet sind, kennt man in horizontaler Richtung die Region des Baumes, in der die diesbezüglich guten TSK liegen: Es sind die Äste möglichst weit links im Enumerationsbaum. Mit Hilfe der Gesamtfläche der Teilesets kann zudem in vertikaler Richtung bestimmt werden, wo die zulässigen TSK liegen, da aus den Auslastungsgrenzen obere und untere Schranken gebildet werden.

Abbildung 5-2 zeigt einen Teil des Enumerationsbaumes mit eingezeichneten Auslastungsschranken. Nur die TSK innerhalb der Schranken sind zulässig, d.h. nur sie werden näher geprüft (auf den Ablauf der Teilesetwahl wird im nächsten Abschnitt eingegangen).

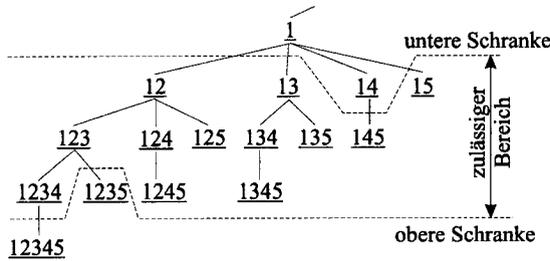


Abbildung 5-2: Enumerationsbaum mit Auslastungsschranken

Bei der Überprüfung der Schranken werden die Aufträge in TSK_k und die Aufträge in TS_{root} zur Menge $TSK_{(k,root)}$ zusammengefasst. In TS_{root} seien m Aufträge enthalten, in TSK_k n Aufträge. Damit ist die gesamte Fläche der Umhüllenden im Trog:

$$F_{TSK(k,root)} = \sum_{i \in TSK(k,root)} (u_i \times v_i)$$

Die *untere Schranke* wird erstens durch die minimal geforderte Auslastung $f_{MinAusl}$ gebildet, zweitens durch die in Gleichung [5-1] enthaltene, bereits diskutierte Forderung:

$$(F_{TSK(k,root)} \geq F_{Trog} \cdot f_{MinAusl}) \cap (\text{SetValue}(TSK_{(k,root)}) > \text{Value}(TSK_{best})) \quad [5-5]$$

$$\text{Pri Value}(TSK_{(k,root)}) > \text{Value}(TSK_{best}) \quad [5-6]$$

[5-6] wird später als Abbruchbedingung verwendet, womit die Anzahl hoher Matrizen keinen Einfluss auf den Abbruchentscheid hat. Die *obere Schranke* für ein TSK_k (inklusive TS_{root}) wird alleine durch die maximale Auslastung gesetzt, und ist in Bedingung [5-7] aufgezigt:

$$F_{TSK(k,root)} \leq F_{Trog} \quad [5-7]$$

Navigation im Baum

Da wie erwähnt der Enumerationsbaum gar nie aufgebaut wird sondern nur zur Veranschaulichung dient, beziehen sich die Begriffe „Navigation“ und „Verzweigungstyp“ auf dieses anschauliche Modell. Strenggenommen sind die Verzweigungstypen „Regeln zur Generierung des neuen TSK “.

Bei der Suche nach dem bestmöglichen Teilesetkandidaten werden die Knoten nach lexikographischer Ordnung besucht: Die Äste werden also von „links nach rechts“ und von „oben nach unten“ untersucht. Gestartet wird beim root-Knoten, und es wird $\text{Value}(TS_{root}) = 0$ gesetzt.

Um von einem Knoten TSK_k zum nächsten Knoten TSK_{k+1} zu gelangen, wird einer der in Tabelle 5-2 aufgeführten Verzweigungstypen eingesetzt. Welcher Typ angewendet wird, hängt von der Position des aktiven Knotens TSK_k im Baum und vom Ergebnis der Abfragen bezüglich TSK_k ab. Bei diesen Abfragen werden, wie in den Gleichungen [5-5] bis [5-7] bereits gezeigt, die Werte des zusammengefassten $TSK_{(k,root)}$ verwendet.

In untenstehender Tabelle wird gezeigt, aus welchen Aufträgen TSK_{k+1} besteht und wann welcher Verzweigungstyp angewendet wird. Bei den Erklärungen ist jeweils erwähnt, ob für TSK_k

vor der Generierung von TSK_{k+1} eine *Layoutgenerierung LG* durchgeführt wird oder nicht. Nach einer LG wird ev. TSK_{best} geändert.

| Typ | TSK_k | TSK_{k+1} | Erklärung (bezogen auf TSK_k) |
|-----|--|---------------------------|---|
| I | $\{g, \dots, i, j\}$ (mit $j \neq q$) | $\{g, \dots, i, j, j+1\}$ | a) [5-7] erfüllt, [5-5] nicht erfüllt; ohne LG weiter b) [5-7], [5-5] erfüllt; Die LG ergibt ein zulässiges Layout Man bewegt sich im Baum nach „unten“. |
| II | $\{g, \dots, i, j\}$ (mit $j \neq q$) | $\{g, \dots, i, j+1\}$ | a) [5-7] nicht erfüllt, [5-6] erfüllt; ohne LG weiter b) [5-7], [5-5] erfüllt; Die LG ergibt <i>kein</i> zulässiges Layout Die Nachkommen von TSK_k werden nicht betrachtet, man bewegt sich im Baum nach „rechts“. |
| III | $\{g, \dots, i, j, q\}$ (mit $j \neq q$) | $\{g, \dots, i, j+1\}$ | TSK_k hat keinen Sohn (Ende eines Astes). Falls [5-7] und [5-5] erfüllt sind, wird eine LG durchgeführt. Unabhängig vom Resultat muss Typ III angewendet werden. |

Tabelle 5-2: Erklärung zu den Verzweigungstypen

Man beachte erstens, dass Typ III immer durchgeführt werden muss, wenn man am Ende eines Astes angelangt ist. Zweitens wird [5-6] nur dann anstelle von [5-5] geprüft, wenn [5-7] nicht erfüllt wurde, weil nun u.U. die Teilesetwahl abgebrochen wird. Zur Illustration sind die drei Verzweigungstypen in Abbildung 5-3 eingezeichnet:

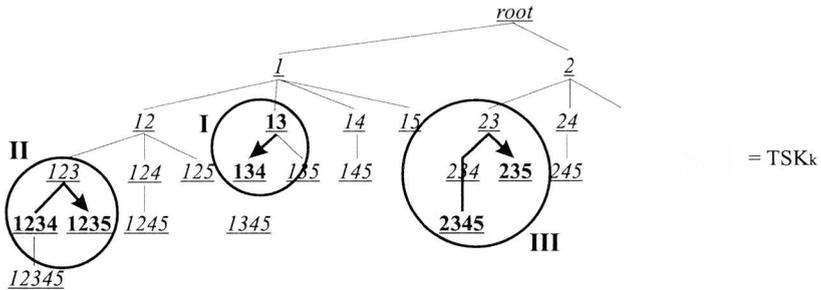


Abbildung 5-3: Verzweigungstypen im Baum

Nachdem gezeigt wurde, wie die Menge aller Teilesetkandidaten TSK als Enumerationsbaum dargestellt werden kann, wie in diesem Baum schnell potentiell gute TSK lokalisiert werden können und wie die Navigation im Baum vor sich geht, kann im nächsten Abschnitt auf den Ablauf der Teilesetwahl eingegangen werden.

5.2.3 Ablauf der Teilesetwahl

Zusammenfassend seien an dieser Stelle folgende Punkte aufgezählt:

1. Startknoten ist der root-Knoten des Enumerationsbaumes.
2. Dank den Schranken [5-5] bis [5-7] werden nicht alle der 2^q-1 potentiellen Knoten besucht. Vor allem wegen der oberen Schranke werden viele Knoten ausgeschlossen.

3. Von den besuchten Knoten werden diejenigen genestet, die innerhalb der Schranken liegen. Darüber hinaus wird mit Hilfe der vorgestellten Enumeration dafür gesorgt, dass mit ‚guten‘ TSK die Suche gestartet wird. Deshalb ist die untere Schranke von Beginn weg hoch, welche u.a. vom besten gefundenen TSK abhängt.
4. Die Navigation im Baum erfolgt mit den Regeln I bis III aus Tabelle 5-2. Vom Startknoten aus wird immer mit der Regel I gestartet, da TS_{root} entweder leer oder garantiert legbar ist.
5. Ein Abbruch erfolgt prinzipiell dann, wenn alle zulässigen Knoten besucht worden sind. Darüber hinaus können verschiedene Abbruchkriterien formuliert werden. Für FluidPlan sind solche Kriterien notwendig, da in der Fertigung nur relativ kurze Rechenzeiten zur Verfügung stehen, bis ein Resultat erwartet wird. Mögliche Abbruchkriterien sind:
 - Zeitlimite, die nicht überschritten werden darf (inklusive Nestingalgorithmus). Denkbare Startpunkte für die Uhr sind: der Start des Moduls ‚Teilesatzwahl‘; der Moment, in dem zum ersten Mal ein TSK_{best} gefunden wird; jedesmal, wenn ein neues TSK_{best} gefunden wird, etc.
 - Anzahl genesteter Knoten
 - Weder Bedingung [5-7] noch Bedingung [5-6] werden erfüllt

Bemerkung zum letzten Punkt: Hier nimmt man in Kauf, dass ev. noch nicht besuchte TSK existieren, die beide Bedingungen erfüllen. Aufgrund der Enumeration darf man allerdings vermuten, dass der PriValue für ein Astende eine obere Schranke für die SetValue der TSK ist, welche weiter rechts im Baum liegen. Abbildung 5-4 zeigt den Ablauf der Teilesatzwahl schematisch auf.

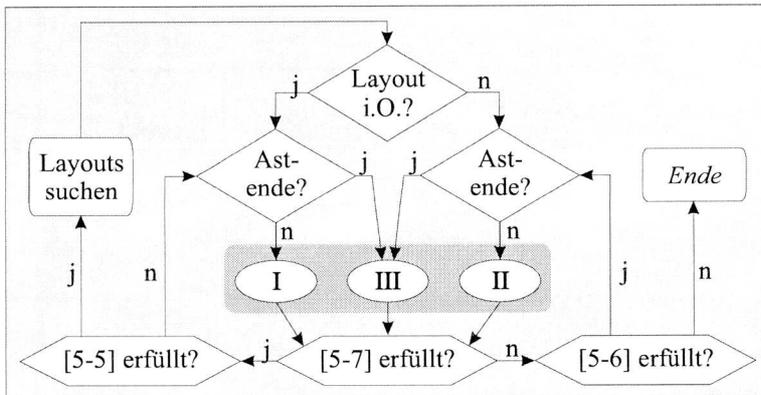


Abbildung 5-4: Grobablauf der Teilesatzwahl

Gemäss dem oben aufgeführten Punkt 4 wird mit einem Verzweigungstyp I begonnen (die Verzweigungstypen sind dunkelgrau hinterlegt). Ein neu zusammengestelltes TSK wird jeweils zuerst daraufhin untersucht, ob die Schranken eingehalten werden (Bedingungen [5-7] und [5-5]). Nur wenn beide Bedingungen erfüllt sind, wird via ‚Layouts suchen‘ der Modul ‚Layoutgenerierung‘ gestartet. Sonst wird direkt zum nächsten Verzweigungstyp übergegangen. Liegt ein TSK beispielsweise nur innerhalb der oberen, aber ausserhalb der unteren Schranke, wird nur geprüft, ob der TSK am Baumende liegt; entsprechend wird dann der Verzweigungstyp I oder III angewendet.

Der in Abbildung 5-5 gezeigte Submodul ‚Layouts suchen‘ bereitet den mehrmaligen Aufruf des Moduls ‚Layoutgenerierung‘ vor. Resultiert aus einem dieser Aufrufe ein Layout, in dem alle Aufträge unter Einhaltung der Beschickungsregeln platziert werden können, liegt ein *zulässiges Layout* für TSK_k vor. Darauf bezieht sich die Abfrage ‚Layout i.O.?‘:

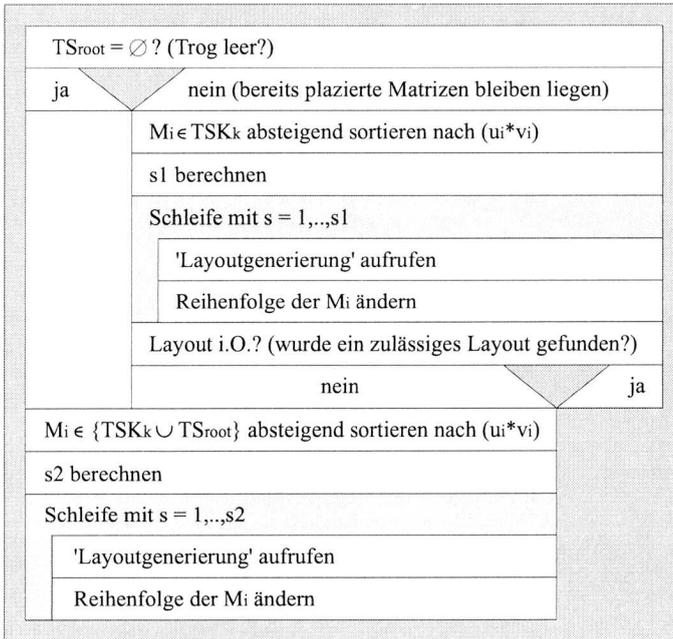


Abbildung 5-5: Submodul ‚Layouts suchen‘

Es werden zwei verschiedene Aufrufe des Moduls ‚Layoutgenerierung‘ unterschieden:

1. *Auffüllen*: Wenn TS_{root} nicht leer ist, wird zuerst versucht, die Matrizen aus TS_{root} im Trog zu belassen und das Layout mit den Matrizen des TSK_k aufzufüllen. Damit soll der Handlingsaufwand minimal gehalten werden, um vom Layout L_{j-1} zum Layout L_j umzurüsten.
2. *Neu füllen*: Kann durch das Auffüllen kein zulässiges Layout generiert werden, wird der Trog gedanklich geleert. Die Matrizen aus TSK_k und aus TS_{root} werden zusammengelegt und es wird versucht, auf diese Weise ein zulässiges Layout von Anfang an zu generieren. Es ist zu erwarten, dass ein auf diese Weise gefundenes Layout einen relativ hohen Handlingsaufwand aufweist. Ist TS_{root} (d.h. der Trog) noch leer, wird direkt der zweite Aufruf gestartet.

Beide Arten von Aufrufen werden für einen TSK unter Umständen mehrere Male durchgeführt, wobei sich für dieselbe Menge von Matrizen andere Layouts ergeben können. Das ist möglich, weil der Output der Layoutgenerierung abhängig von der Reihenfolge der Matrizen in TSK ist. Nach jedem Aufruf wird die Reihenfolge der ersten s Matrizen aus dem TSK mittels zyklischer Permutation geändert. Die Anzahl Aufrufe s_1 bzw. s_2 ist abhängig von der Anzahl Matrizen mit

grosser Grundfläche. Als Matrizen mit grosser Grundfläche werden diejenigen Matrizen M_i bezeichnet, für welche gilt:

$$u_i \cdot v_i > F_{\text{Grenz}} = \frac{1}{9} \cdot F_{\text{Trog}} \quad [5-8]$$

F_{Grenz} entspricht der Grösse einer Tabuzone. Für s_1, s_2 gilt:

- s_1 = Anzahl Matrizen aus TSK_k , welche Ungleichung [5-8] erfüllen
- s_2 = Anzahl Matrizen aus TSK_k und aus TS_{root} , welche Ungleichung [5-8] erfüllen

Die Reihenfolge der Matrizen wird jeweils wie folgt geändert. Ist s die Anzahl grosser Matrizen, so wird für $s > 0$ der Modul ‚Layoutgenerierung‘ s mal aufgerufen. Das erste Mal erfolgt der Aufruf in der nach der Grundfläche sortierten Reihenfolge $\{M_1, \dots, M_m\}$. Vor dem nächsten Aufruf wird M_1 an die Position der Matrize M_s verschoben, M_2 bis M_s rücken um eine Position vor. Man betrachtet also $\{M_2, M_3, \dots, M_s, M_1, M_{s+1}, \dots, M_m\}$, $\{M_3, M_4, \dots, M_s, M_1, M_2, M_{s+1}, \dots, M_m\}$ etc.

5.2.4 Berechnung von $f_{HM(j)}$

Der in Gleichung [5-4] verwendete Faktor $f_{HM(j)}$ wurde eingeführt, damit nicht zuviele hohe Matrizen miteinander auf dem Trog sind. Dazu wird ein Durchschnittswert berechnet, der angibt, wieviele hohe Matrizen theoretisch pro Pressung auf einem Trog liegen müssen, bis alle vorliegenden Aufträge abgearbeitet sind. Es wird erstens angenommen, dass keine neuen Aufträge zu $A_{(E6)}$ hinzukommen und dass zweitens die Auslastung bei der Abarbeitung konstant ist. Als Auslastung wird die geforderte Minimalauslastung eingesetzt, um die Anzahl notwendiger Pressungen zu schätzen. Je näher der reale Anteil hoher Matrizen eines Teilesets k diesem Schätzwert ist, umso grösser ist $f_{HM(j)}$, zu dessen Berechnung folgende Variablen nötig sind:

| Variable | Erklärung |
|---------------------------|---|
| W_{theor} | Theoretisch notwendige Anzahl Pressungen zur Abarbeitung aller Aufträge in $A_{(E6)}$, bei einer Auslastung von 100% |
| W_{Sch} | Schätzung der notwendigen Anzahl Pressungen bei der Minimalauslastung |
| $B = \{B_1, \dots, B_b\}$ | Untermenge aus $A_{(E6)}$; für B_k gilt $h_k > \text{Grenzhöhe}$ (\Rightarrow hohe Matrizen) |
| $HM(A_{(E6)})$ | Durchschnittliche Anzahl hoher Matrizen pro Pressung |
| HM_j | Anzahl hohe Matrizen im Teileset j |

Tabelle 5-3: Variablen zur Berechnung von $f_{HM(j)}$

W_{theor} erhält man, indem man für alle Aufträge in $A_{(E6)}$ das Produkt aus Grundfläche ($u_i \cdot v_i$) mal Anzahl Pressungen w_i bildet und diesen Wert durch die Trogfläche teilt:

$$W_{\text{theor}} = \frac{1}{L_T \times B_T} \times \sum_{i=1}^{a(E6)} (u_i \times v_i \times w_i) \quad [5-9]$$

Und für den Schätzwert W_{Sch} gilt mit f_{MinAusl} als geforderte Minimalauslastung:

$$W_{\text{Sch}} = \frac{W_{\text{theor}}}{f_{\text{MinAusl}}} \quad [5-10]$$

Die durchschnittliche Anzahl hoher Matrizen erhält man, indem die Anzahl Pressungen aller Aufträge in B aufsummiert und anschliessend durch den Schätzwert dividiert wird:

$$\overline{\text{HM}}(A_{(E_6)}) = \left(\sum_{i=1}^b w_i \right) / W_{\text{Sch}} \quad [5-11]$$

Die Differenz dieses Schätzwertes von der realen Anzahl hoher Matrizen eines Teilesets j fliesst gemäss Gleichung [5-12] in $f_{\text{HM}(j)}$ ein. Pro Matrize, um die diese Werte voneinander abweichen, sinkt der Wert eines genesteten Teilesets um 10% (vgl. auch Gleichung [5-4]).

$$f_{\text{HM}(j)} = \max\{0, 1 - 0.1 \times |\overline{\text{HM}}(A_{(E_6)}) - \text{HM}_j|\} \quad [5-12]$$

Möglich wäre auch, die reale Anzahl hoher Matrizen mit einem fixen „Idealwert“ (z.B. drei hohe Matrizen pro Pressung) zu vergleichen. Dabei wird aber folgende Situation in der Fertigung nicht berücksichtigt; Es können sich z.B. immer mehr hohe Matrizen (im Vergleich zu flachen Matrizen) vor der Presse ansammeln, wenn relativ viele hohe Matrizen gefertigt werden müssen. Ein situativ angepasster Faktor verhindert eine derartige Fehlentwicklung.

5.3 Bestimmung einer Folge von Teilesets

Sind in TS_1 Aufträge unterschiedlicher Losgrösse enthalten, wurde bereits mit TS_1 eine Teilesetfolge bzw. eine Layoutfolge generiert. Denn jedesmal, wenn die Matrize eines fertig bearbeiteten Auftrags vom Trog genommen wird, entsteht ein neues Layout. L_{j+1} ist in diesem Fall jeweils eine Untermenge von L_j und wird *abgeleitetes Layout* genannt. Abbildung 5-6 zeigt eine derartige Layoutfolge: In der Abszisse sind die Layoutnummern eingetragen, in der Ordinate die prozentuale Auslastung des Troges (zusätzlich sind in L_1 die Auslastungen der einzelnen Matrizen eingetragen).

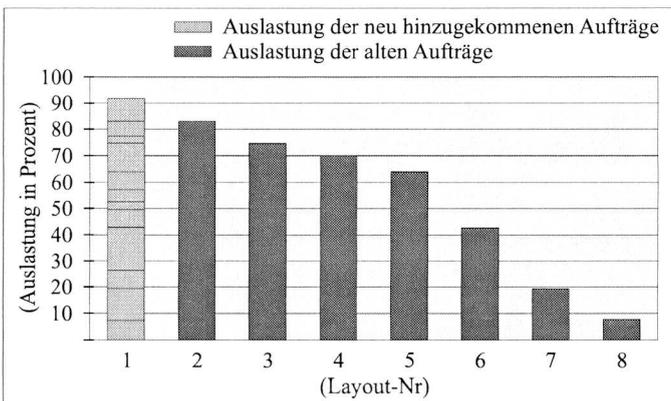


Abbildung 5-6: Abgeleitete Layoutfolge eines einzigen Teilesets

5.3.1 Zielfunktion einer Teilesetfolge

Die in Gleichung [5-1] aufgeführte Zielfunktion für ein einzelnes Teileset TS_j kann auch für eine Folge von Teilesets $TSL = \{TS_1, \dots, TS_n\}$ benutzt werden. Auch wenn die Funktion [5-13] in dieser Arbeit nicht benutzt wird, wird sie hier aufgeführt, weil sie ein möglicher Schätzwert für den Modul ‚Synchro‘ ist, welcher in Abschnitt 4.3.3 vorgestellt wurde.

Bei der hier besprochenen Zielfunktion wird der Wert $SetValue(TS_j)$ eines Teilesets TS_j pro durchgeführte Pressung p_i mit einem Gewichtungsfaktor $g(p_i)$ multipliziert. Für ein TS_j , welches bei der Pressung p_i eingeplant ist, wird die Schreibweise $TS(p_i)$ verwendet. Je früher ein TS_j in TSL eingeplant ist, umso mehr soll sein Wert zum Gesamtwert der Teilesetfolge beitragen. Auf diese Weise wird eine Folge von Teilesets TSL, in der ‚gute‘ TS_j früh eingeplant sind, hoch bewertet. Die Zielfunktion heisst (mit \bar{n} für die Anzahl Pressungen):

$$Value(TSL) = \sum_{i=1}^{\bar{n}} \{g(p_i) \cdot SetValue(TS(p_i))\} \quad [5-13]$$

Für den Gewichtungsfaktor $g(p_i)$ gilt:

$$g(p_i) = \frac{\bar{n} - i/2}{\bar{n}} \quad [5-14]$$

Für $i = 1$ erhält man somit $g(p_1) \approx 1$ und für $i = \bar{n}$ ist $g(p_{\bar{n}}) = 0.5$.

5.3.2 Direktes Plazieren

Eine einfache Strategie zur Erzeugung von Layoutfolgen ist, jedesmal ‚Nesting‘ neu zu starten, wenn (mindestens) eine Matrize vom Trog entfernt wird. In Abbildung 5-6 wird somit TS_2 zu TS_{root} und es wird versucht, das immer noch zu über 80 Prozent ausgelastete Layout aufzufüllen.

Abbildung 5-7 zeigt eine Layoutfolge, die mit dieser „direktes Plazieren“ genannten Strategie erzeugt wurde. Die dabei verwendeten Daten sind in mehrfacher Hinsicht speziell, wurden aber bewusst gewählt, um möglichst gut zu zeigen, weshalb das „direkte Plazieren“ eine ungeeignete Strategie ist. Der Datensatz

- hat wenige Aufträge (75),
- eine kleine Umhüllende pro Matrize (bis zu 30 Matrizen können in einem Layout plaziert werden)
- und wurde mit einer hohen geforderten Minimalauslastung geschachtelt (90 Prozent)

Bis zum elften Layout (total 32 Pressungen) konnte die geforderte Auslastung eingehalten werden, dann fällt sie stark zusammen. Sie fällt in den folgenden 21 Layouts (112 Pressungen) bis auf ca. 50 Prozent. Dieser Leistungsabfall ist auch für diese Strategie sehr früh, und hängt damit zusammen, dass von den wenigen vorhandenen Matrizen jeweils eine grosse Anzahl plaziert werden kann.

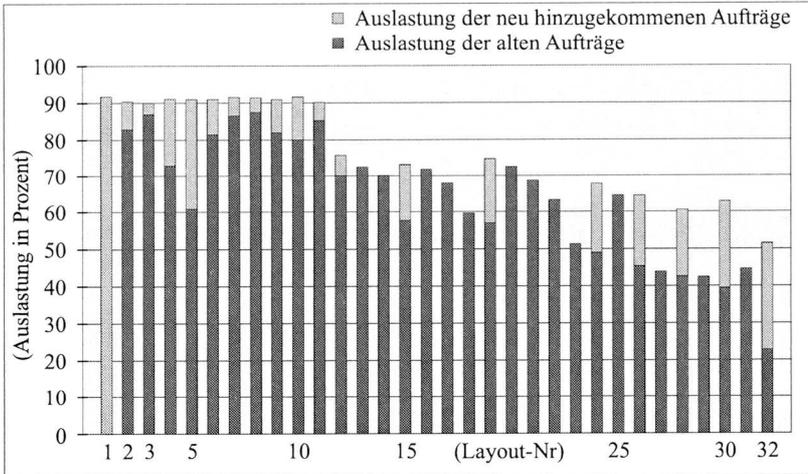


Abbildung 5-7: Auslastungsverteilung einer Layoutfolge (in Prozent)

Abbildung 5-8 zeigt die zugehörige Verteilung der Aufträge nach Prioritätsgruppen geordnet (Ordinate: Anzahl Aufträge; Abszisse: Layoutnummer). Dabei fällt auf, dass die Anzahl der Aufträge in den ersten 11 Layouts - also bei quasi konstanter Auslastung - stets steigt. Das heisst, dass der durch eine entfernte Matrize freigewordene Platz meist durch mehrere Matrizen aufgefüllt wurde.

Hier liegt das Problem beim direkten Plazieren: Grosse Matrizen bleiben liegen, bis keine kleinen Matrizen mehr vorhanden sind und deswegen genügend grosse Lücken entstehen können. Ab Layout 15 erfolgt jeder Auslastungszuwachs mit einer einzigen neuen Matrize, und ab Layout 23 liegen 10 oder weniger Matrizen im Trog (was für diesen Datensatz wenig ist).

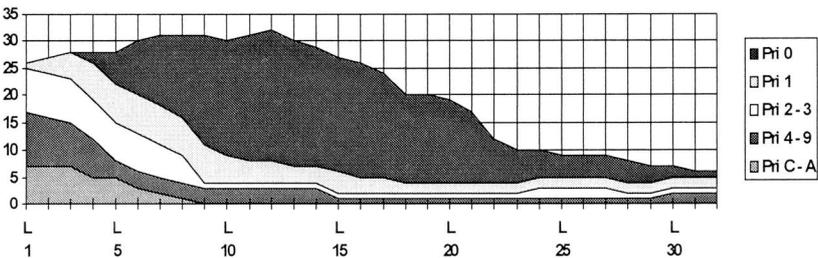


Abbildung 5-8: Verteilung der Anzahl Aufträge

Positiv zu vermerken ist die Zusammensetzung der Teilesets bezüglich den Prioritäten: Fast alle Aufträge der oberen Prioritätsklassen wurden im ersten Layout plaziert. Das illustriert, wie der Modul ‚Teilesetwahl‘ diesbezüglich effizient arbeiten kann. Hingegen ist die Strategie, in wel-

cher der Modul eingesetzt wird, langfristig nicht gut. Als Alternative bietet sich die Strategie des vorausschauenden Plazierens an.

5.3.3 Vorausschauendes Plazieren

Grundidee der Strategie:

Die Matrize mit der grössten Grundfläche aus der höchsten nichtleeren Prioritätsklasse muss im nächstmöglichen Layoutvorschlag enthalten sein. Entstehen zwischen zwei auf diese Weise generierten Vorschlägen Layouts mit einer tieferen Auslastung, werden diese Layouts mit der Strategie des direkten Plazierens aufzufüllen versucht.

Es wird also gefordert, dass der erste Auftrag (A_1) der Auftragsliste $A_{(E5)}$ im nächsten vorgeschlagenen Layout plaziert wird ($A_{(E5)}$ ist nach absteigender Priorität und nach absteigender Fläche geordnet). Beim vorausschauenden Plazieren werden somit nur diejenigen TSK geprüft, die an einem Teilst des Knotens mit dem $TSK_1 = \{A_1\}$ hängen.

Graphisches Beispiel

Abbildung 5-9 zeigt eine Layoutfolge, die aus einem einzigen Layout-Aufruf generiert wurde². Beim Start der Teilesetwahl wird $TS_{root} = TS_2$ gesetzt. Wird kein TSK (mit A_1) gefunden, der mit TS_{root} zusammen ein zulässiges Layout ergibt, wird $TS_{root} = TS_3$ gesetzt und erneut ein zulässiges Layout gesucht. Auf diese Weise ist garantiert, dass A_1 im frühestmöglichen Layout eingeplant wird.

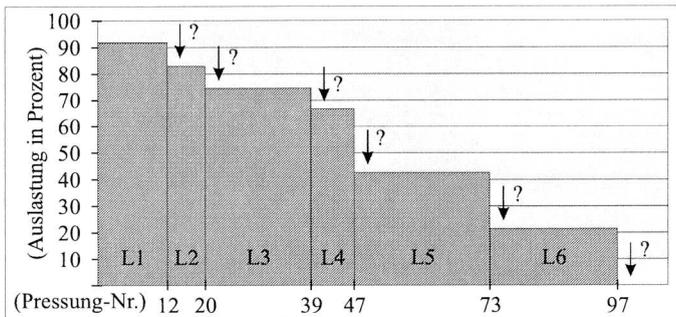


Abbildung 5-9: Mit einem einzigen Layout-Aufruf generierte Layoutfolge

Falls A_1 erst in L_4 plaziert werden kann, wie beispielsweise in Abbildung 5-10 gezeigt, entsteht in der Auslastung eine Lücke zwischen L_1 und L_4 . Diese Lücke kann bzw. soll gefüllt werden.

2. Im Gegensatz zur Layoutfolge aus Abbildung 5-6 sind hier in der Abszisse die Anzahl Pressungen eingetragen. Das heisst, dass die Breite eines Layouts L_j proportional zu seiner Anzahl notwendiger Pressungen $AnzP_j$ steht.

Die Anzahl Pressungen der dabei berücksichtigten Aufträge muss kleiner oder gleich der Anzahl Pressungen der Auslastungslücke sein.

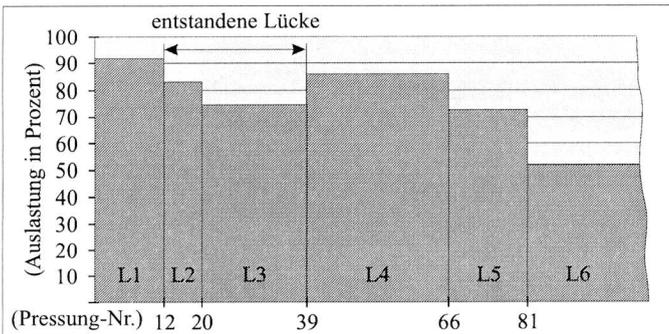


Abbildung 5-10: Layoutfolge nach Aufruf des vorausschauenden Plazierens (L_4 ergänzt)

Steuerung des vorausschauenden Plazierens

Die Steuerung der vorgestellten Strategie liegt nicht beim Modul ‚Teilesetwahl‘, sondern beim Modul ‚Nestingfenster‘. Folgende Gründe sprechen für diesen Entscheidung:

- Es ist nicht nötig, die in Abschnitt 5.2 beschriebene Funktion des Moduls ‚Teilesetwahl‘ zu erweitern. Damit werden die in Abschnitt 2.4 aufgezählten Argumente eingehalten, weshalb FluidPlan modular aufgebaut werden soll (Schrittweise Entwicklung; Einfache Anpassung bei Strategieänderungen; Abbildung der Aufgabenhierarchie).
- Der Modul ‚Nestingfenster‘ filtert die Systemdaten und präsentiert sie dem Benutzer. Darüber hinaus werden Funktionen zur Verfügung gestellt, um Ereignisse wie z.B. die Teilesetwahl auszulösen. Werden die Ereignisse vom Benutzer nach einer festgelegten Strategie ausgelöst, kann man diesen Vorgang automatisieren. Es ist zweckmässig, diese Automatisierung an der Stelle durchzuführen, an der sonst die Eingaben getätigt würden.

Ablauf des vorausschauenden Plazierens

Nach Vorgabe des Benutzers von Trog, Teilefamilie und Anzahl zu generierende Pressungen p_{soll} sind die meisten Inputdaten aus Tabelle 5-1 gegeben. Indem das ‚Nestingfenster‘ Teilmen-gen aus $A_{(E5)}$ sowie L_j mehrere Male verschieden bestimmt und als weitere Inputdaten an den Modul ‚Teilesetwahl‘ übergibt, steuert es die Berechnung der Layouts. Im unten beschriebenen Ablauf geht man davon aus, dass bereits ein akzeptiertes Layout L_0 sowie von L_0 abgeleitete Layouts L_j mit $j = 1, \dots, n$ existieren. Unter der Annahme, dass noch nicht genügend Pressungen vorgeschlagen sind, werden folgende Schritte durchgeführt:

1. A_1 wird im frühestmöglichen Layout $L_j \in \{L_1, \dots, L_n\}$ zu plazieren versucht. Kann A_1 in keinem Layout aus $\{L_1, \dots, L_n\}$ plaziert werden, wird $L_j = L_{n+1}$ generiert. $A_{(E5)}$ ist nach Priorität und nach Grundfläche der Matrizen geordnet, A_1 ist der erste Auftrag in $A_{(E5)}$.
2. Für A_1 wurde in L_j ein Platz gefunden. Es werden zwei Fälle unterschieden:
 - $j=1$: Es sind keine Lücken entstanden. Direkt zu Punkt 3.

- $j > 1$: Es sind Lücken entstanden. Die Layoutgenerierung wird aufgerufen für alle Layouts L_i aus $\{L_1, \dots, L_{j-1}\}$, wobei folgende Restriktionen bzw. Bemerkungen gelten:
 - Die TSK_k werden aus einer Untermenge $A_{(ES)}$ von $A_{(E5)}$ zusammengestellt. In $A_{(ES)}$ sind nur Aufträge enthalten, deren Losgrösse nicht grösser ist als die Summe der Anzahl Pressungen der Layouts $\{L_1, \dots, L_{j-1}\}$. Das heisst, diese Aufträge müssen vor L_j abgearbeitet sein.
 - Durch neu aufgenommene Aufträge wird ev. L_i oder ein folgendes Layout in zwei oder mehrere Layouts aufgeteilt. Das heisst, die Anzahl Layouts in $\{L_1, \dots, L_{j-1}\}$ kann wachsen, die Summe der Anzahl Pressungen bleibt hingegen konstant.
3. Die Layoutgenerierung wird für L_j aufgerufen. Die entsprechenden TSK_k werden aus $A_{(E5)}$ zusammengestellt, wobei A_1 in TSK_k enthalten sein muss.
4. Es wird geprüft, ob bereits genug Layoutvorschläge gemacht worden sind. Das heisst, die Summe der Pressungen aller vorgeschlagenen Layouts muss gleich oder grösser dem vorgegebenen Wert p_{soll} sein. Die vom letzten generierten Layout L_j abgeleiteten Layouts werden dabei nicht berücksichtigt. Abhängig von dieser Abfrage wird folgendes durchgeführt:
- p_{soll} nicht erreicht: $L_0 = L_j$. Zurück zu Schritt 1.
 - p_{soll} erreicht: Ende der Bestimmung einer Layout- bzw. Teilesetfolge.

Der Auftrag A_1 wird im ersten möglichen Layout plziert, ohne zu prüfen, ob die Zielfunktion der Teilesetfolge besser erfüllt wäre, wenn A_1 in einem späteren Layout plziert würde. Es wird also eine *greedy lookahead*-Strategie verfolgt (vorauschauend Plzieren, ohne Alternativen zu prüfen). Der Grund dafür liegt im hohen Rechenaufwand, der resultiert, wenn man mehrere verschiedene Varianten von Teilesetfolgen berechnen will.

In Kapitel 7 wird beschrieben, wie die beiden hier vorgestellten Strategien getestet wurden.

6 Layoutgenerierung

In diesem Kapitel werden alle notwendigen Elemente vorgestellt, um das sequentielle 2D Bin Packing Problem zu behandeln. Es geht also darum, für einen vom Modul ‚Teileselektwahl‘ zusammengestellten Teileselektkandidaten TSK ein zulässiges Layout zu finden. Zulässig ist ein Layout dann, wenn alle Matrizen des TSK unter Einhaltung der Restriktionen im Trog platziert werden können. Es wird gezeigt, wie man mit Hilfe gerichteter Graphen Layouts darstellt, wie in solchen Layouts freie Plätze für noch nicht platzierte Matrizen gefunden werden und wie beim Einfügen resp. Entfernen von Matrizen die Graphen nachgeführt werden müssen.

Gliederung des Kapitels

In Abschnitt 6.1 wird eine Darstellungsform für Layouts definiert und es werden Methoden vorgestellt, wie mit diesem Ansatz in den Layouts freie Plätze für zusätzliche Matrizen gefunden und belegt werden können. In Abschnitt 6.2 folgt eine formale, mathematische Beschreibung dieses Ansatzes mit Hilfe gerichteter Graphen.

Der gesamte Ablauf der Layoutgenerierung wird in Abschnitt 6.3 beschrieben, Input und Output des Moduls sind an dieser Stelle ebenfalls aufgeführt.

In Abschnitt 6.4 werden verschiedene Bewertungsfunktionen vorgestellt, welche einerseits bei der Wahl eines Platzes für eine Matrize, andererseits bei der Bewertung eines ganzen Layouts benutzt werden. Dabei werden auch hohe Matrizen behandelt. Abschnitt 6.5 beschreibt, welche Manipulationen an den gerichteten Graphen durchgeführt werden müssen, wenn Matrizen in ein Layout platziert bzw. entfernt werden. Wie Tabuzonen im Rahmen des präsentierten Konzeptes behandelt werden, zeigt Abschnitt 6.6.

Im siebten und letzten Abschnitt dieses Kapitels wird der Modul ‚Layoutoptimierung‘ behandelt, der in Abschnitt 4.3.1 eingeführt wurde. Dabei geht es um die Frage, wie eine Folge bestehender Layouts optimiert werden kann, ohne die Reihenfolge der Teilesets zu ändern. Die verschiedenen Positionen, welche den einzelnen Matrizen von Layout zu Layout zugewiesen sind, sollen dabei über die ganze Folge von Teilesets optimiert werden, sodass der Handlungsaufwand für die gesamte Layoutfolge minimiert wird.

6.1 Ansatz für die Layoutgenerierung

Der Ansatz für die Layoutgenerierung für ein TSK = $\{M_1, \dots, M_m\}$ ist sequentiell (d.h. die Matrizen M_1, M_2, \dots, M_m werden einzeln in der gegebenen Reihenfolge auf den Trog T gelegt).

M_1, \dots, M_k seien bereits auf T platziert und man versucht, M_{k+1} zu positionieren. Dazu werden die „freien Plätze“ betrachtet, welche zwischen M_1, \dots, M_k existieren. Typischerweise müssen M_1, \dots, M_k oder eine Untermenge davon auf T bewegt werden, damit M_{k+1} platziert werden kann. Um die unendliche Menge der möglichen Bewegungen zu beschränken und um sicherzustellen, dass sich die Matrizen dabei nicht überschneiden, braucht es ein konzeptionelles Werkzeug, welches jetzt eingeführt wird.

Das Konzept der k-Teilung wird benutzt, um die relativen Positionen der Matrizen zu beschränken und zu beschreiben. Die Forderung nach Überschneidungsfreiheit der Matrizen wird mittels einem System von Ungleichungen garantiert, welches die Position und die Größe der Matrizen mit einbezieht. Mit Hilfe gerichteter Graphen werden beide erwähnten Aspekte veranschaulicht und effizient behandelt.

6.1.1 k-Teilung eines Layouts

Gegeben sei ein rechteckiger, zweidimensionaler Trog T. Dann gilt die Definition:

Eine k-Teilung (des Troges T) ist eine Partition P von T in k achsenparallele Rechtecke R_1, \dots, R_k , wobei sich nie vier Rechtecke von P in einem Punkt treffen.

Abbildung 6-1 zeigt eine k-Teilung für $k = 7$, also eine 7-Teilung. Eine k-Teilung entspricht einem planaren Gridgraphen G_p (d.h. einer Menge von achsenparallelen Geradensegmenten. Diese Segmente werden Kanten genannt, deren Extremitäten Knoten.). Die Knoten von G_p sind die Eckpunkte der Rechtecke R_1, \dots, R_k . Die Kanten von G_p sind die für die Inklusion minimalen Seitensegmente, die zwei Knoten miteinander verbinden. Beispiele von Kanten sind die Verbindungen der Knotenpaare (1,2) und (5,9). Die Eckpunkte von T haben Grad 2, alle andern Knoten haben Grad 3, weil sich nie vier Rechtecke von P in einem Punkt treffen (Die Anzahl Kanten, die sich in einem Knoten treffen, wird Grad des Knotens genannt).

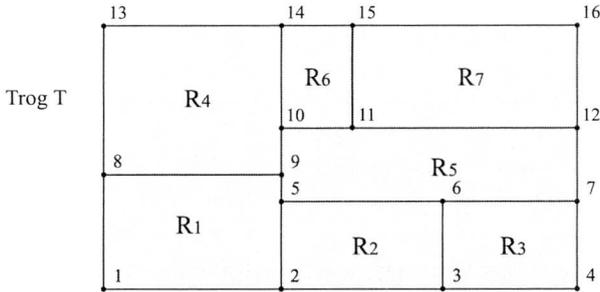


Abbildung 6-1: k-Teilung von T mit $k = 7$.

Weiter gilt die Definition:

Eine Trennlinie einer k-Teilung P ist ein für die Inklusion maximales Geradensegment von G_p .

Beispiele für Trennlinien sind die Segmente [2,14], [11,15], [10,12]. Die Restriktion in der Definition der k-Teilung, dass ein Knoten von G_p höchstens Grad 3 hat, wurde eingeführt, damit für eine k-Teilung eine Familie von sich nicht kreuzenden Trennlinien resultiert. Wie man sehen wird, ist diese Restriktion rein technischer Natur. Die Adjazenzmatrix A_P einer k-Teilung P ist eine $k \times k$ -Matrix mit den Komponenten $a_{ij} \in \{1, r, u, o, \emptyset\}$, für die gilt:

- $a_{ij} = 1$, falls die rechte Seite von R_i und die linke Seite von R_j auf derselben Trennlinie liegen (R_i liegt „links“ von R_j);
- $a_{ij} = r$, $\Leftrightarrow a_{ji} = l$;
- $a_{ij} = u$, falls die obere Seite von R_i und die untere Seite von R_j auf derselben Trennlinie liegen (R_i liegt „unterhalb“ von R_j); [6-1]
- $a_{ij} = o$, $\Leftrightarrow a_{ji} = u$;
- $a_{ij} = \emptyset$ sonst.

Für das gezeigte Beispiel gilt somit: $A_p =$

$$\begin{bmatrix} \emptyset & l & \emptyset & u & l & l & \emptyset \\ r & \emptyset & l & r & u & \emptyset & \emptyset \\ \emptyset & r & \emptyset & \emptyset & u & \emptyset & \emptyset \\ o & l & \emptyset & \emptyset & l & l & \emptyset \\ r & o & o & r & \emptyset & u & u \\ r & \emptyset & \emptyset & r & o & \emptyset & l \\ \emptyset & \emptyset & \emptyset & \emptyset & o & r & \emptyset \end{bmatrix}$$

Es gilt die Definition:

Zwei k-Teilungen sind äquivalent, falls sie dieselbe Adjanzenzmatrix besitzen (eventuell nach Renumerierung der Rechtecke).

Durch paralleles Verschieben von Trennlinien (ohne parallele Trennlinien zu überspringen) kann also eine k-Teilung in eine äquivalente k-Teilung überführt werden, wie beispielsweise Abbildung 6-2 zeigt. Ab jetzt werden nicht am Rand liegende Trennlinien in der gezeigten Art gezeichnet und die Knoten des Gridgraphen G_p weggelassen.

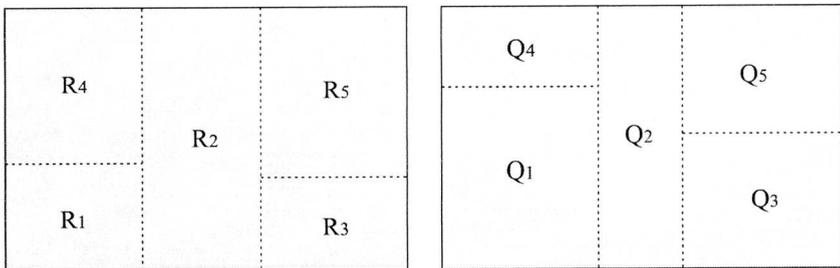


Abbildung 6-2: Äquivalente k-Teilungen R_i und Q_i

Annahme: Die Ecke unten links des Troges sei der Ursprung des benutzten Koordinatensystems. Weiter seien L,B die Gesamtlänge resp. die Gesamtbreite des Troges. Die Ecke oben rechts hat also die Koordinaten (L,B).

Gegeben sei eine k-Teilung P, jedes R_j sei durch vier Trennlinien begrenzt. Dann heißen

$x_L, x_R, x_1, \dots, x_b$ die Abszissen der vertikalen Trennlinien,

$y_U, y_O, y_1, \dots, y_c$ die Ordinaten der horizontalen Trennlinien.

x_L, x_R, y_U, y_O entsprechen den Randtrennlinien. Abbildung 6-3 zeigt die 4 Trennlinien von R_j .

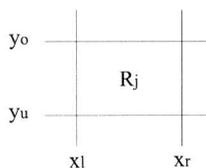


Abbildung 6-3: Trennlinien von R_j mit lokaler Notation für Abszissen und Ordinaten

Es werden für jedes R_j die Ungleichungen

$$\begin{aligned} x_l &\leq x_r \\ y_u &\leq y_o \end{aligned} \tag{6-2}$$

eingeführt, wo x_l, x_r, y_u, y_o die Koordinaten der Trennlinien sind, die R_j bestimmen. Zusätzlich sind $x_L = y_U = 0$ und $x_R = L, y_O = B$. Dieses System von Ungleichungen charakterisiert die Funktion der k -Teilungen äquivalent zu P .

Gegeben sei weiter eine Familie M von Rechtecken M_1, \dots, M_r mit $M_i = u_i * v_i$, eine k -Teilung $P: R_1, \dots, R_k$ mit $R_j = a_j * b_j$ und $k \geq r$ sowie eine Zuordnung z

$$z: \{1, \dots, r\} \rightarrow \{1, \dots, k\}, \quad z(i) \neq z(j) \quad \forall i \neq j$$

Es gilt:

Das Paar (P, z) heisst eine k -Teilung für die Familie M , falls und nur falls

$$\begin{aligned} u_i &\leq a_{z(i)} \\ v_i &\leq b_{z(i)} \quad i = 1, \dots, r. \end{aligned} \tag{6-3}$$

Das heisst, für $i = 1, \dots, r$ können die Rechtecke M_i in die Rechtecke $R_{z(i)}$ gelegt werden. Dieses Konzept wird benutzt, um die relativen Positionen der Matrizen einer Familie M zu beschreiben und folglich ein Layout zu generieren. In Abbildung 6-4 ist ein Layout einer Familie M mit $r=7$ abgebildet, welche in eine 8-Teilung plazierte wurde. Ein Rechteck wie das gezeigte Rechteck R_8 , das keine Matrize enthält, wird *leere Zone* genannt. Leere Zonen werden in Abschnitt 6.5.3 näher betrachtet, wo das Entfernen von Matrizen aus einem Layout diskutiert wird.

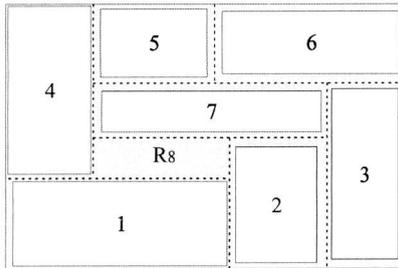


Abbildung 6-4: 8-Teilung für eine Familie M_1, \dots, M_7

Gegeben sei eine k -Teilung (P, z) für M , die möglichen Positionen für die Trennlinien werden aus [6-2] und [6-3] abgeleitet. Unter der Voraussetzung, dass $M_i = u_i * v_i$ in $R_{z(i)}$ liegt, gilt:

$$\begin{aligned} x_l + u_i &\leq x_r \\ y_u + v_i &\leq y_o \end{aligned} \tag{6-4}$$

wo x_l, x_r, y_u, y_o die Koordinaten der Trennlinien sind, welche $R_{z(i)}$ bestimmen.

Abbildung 6-5 illustriert dieses System von Ungleichungen. Man beachte, dass die zwei Systeme von Ungleichungen (in x-Richtung und in y-Richtung) voneinander unabhängig sind.

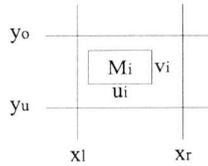


Abbildung 6-5: Trennlinien von $R_{z(i)}$ mit lokaler Notation für Abszissen und Ordinaten

Bevor der Ansatz weiter ausgeführt werden kann, müssen sogenannte *freie Zonen* und *freie Plätze* definiert werden. Gegeben ist eine k -Teilung (P, z) für eine Familie M von Matrizen M_1, \dots, M_r . Sei \mathfrak{S} die Familie der k -Teilungen (P', z) äquivalent zu (P, z) , dann gilt:

Das grösste Rechteck $R_{z(i)}$ in einer k -Teilung der Familie \mathfrak{S} , in welches M_i zugeordnet ist, wird freie Zone von M_i genannt.

Bemerkung: Die Position der freien Zone von M_i bezüglich des Troges T ist absolut.

Abbildung 6-6 zeigt eine k -Teilung eines Layouts, bei der die freie Zone von M_4 eingezeichnet ist. M_4 ist in ihrer *Minimalposition* *MinPos* (möglichst weit „unten links“) sowie in ihrer *Maximalposition* *MaxPos* (möglichst weit „oben rechts“) gezeichnet, die sie innerhalb ihrer freien Zone einnehmen kann.

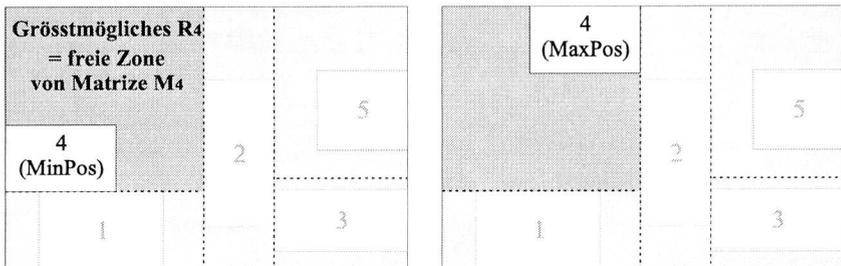


Abbildung 6-6: Freie Zone der Matrize M_4

Gegeben ist die freie Zone R von M_i . Die vier maximalen rechteckigen Bereiche von R , in die man neben M_i eine Matrize legen kann, werden *freie Plätze* von M_i genannt. In Abbildung 6-7 sind die beiden möglichen freien Plätze in x - und in y -Richtung der Matrize M_4 eingezeichnet (übernommen aus dem Beispiel aus Abbildung 6-6). Es gelten die folgenden (allgemeinen) Notationen:

- $p1_x(i)$ = freier Platz *nach* M_i in x -Richtung (rechts von M_i)
- $p1_y(i)$ = freier Platz *nach* M_i in y -Richtung (links von M_i)
- $p2_x(i)$ = freier Platz *vor* M_i in x -Richtung (unterhalb von M_i)
- $p2_y(i)$ = freier Platz *vor* M_i in y -Richtung (oberhalb von M_i)

Sofern nicht unterschieden werden muss, ob von einem freien Platz vor oder nach M_i die Rede ist, wird oft der Ausdruck $p_x(i)$ verwendet, der die freien Plätze $p1_x(i)$ und $p2_x(i)$ zusammenfasst. Analog existiert der Ausdruck $p_y(i)$ für die freien Plätze $p1_y(i)$ und $p2_y(i)$.

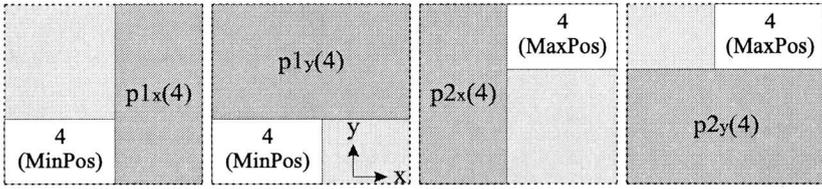


Abbildung 6-7: Freie Plätze neben M_4 (in der freien Zone von M_4)

Mit dem Ansatz der k-Teilung kann die Grundidee für die Layoutgenerierung präzisiert werden:

- Die Matrizen werden sequentiell eingeführt, sodass im Schritt r eine k-Teilung P_{alt} für die Matrizen M_1, \dots, M_r gegeben ist, welche die relativen Positionen für M_1, \dots, M_r festhält.
- Um M_{r+1} zu plazieren, ist eine Teilung für die Matrizen M_1, \dots, M_{r+1} zu finden. Dazu werden ein oder mehrere Rechtecke $R_{j(alt)}, \dots, R_{s(alt)}$ der Teilung P_{alt} aufgeteilt in neue Rechtecke $R_{j(neu)}, \dots, R_{s(neu)}$, R_{r+1}, \dots . Die Matrizen M_j, \dots, M_s werden den Rechtecken $R_{j(neu)}, \dots, R_{s(neu)}$ zugewiesen, M_{r+1} wird R_{r+1} zugewiesen. Die neu erhaltene Teilung P_{neu} liefert eine Teilung für M_1, \dots, M_r, M_{r+1} , falls das System von Ungleichung [6-4] für die Familie von k-Teilungen äquivalent zu P_{neu} zulässig ist.

Fundamental für den vorgestellten Ansatz ist, dass während des Algorithmus kein festes Layout, sondern nur die relativen Positionen der Matrizen behandelt werden. Die Garantie, dass die Beschreibung der relativen Matrizenpositionen zulässig ist, wird durch eine k-Teilung für diese Matrizen mit dem entsprechenden Ungleichungssystem gegeben (die Beschreibung ist zulässig, wenn ein festes Layout existiert, das diese relativen Positionen realisiert). Bei jeder Änderung - Einführen oder Entfernen einer Matrize - ist es demnach zentral, die Änderung der entsprechenden k-Teilung durchzuführen.

In Abschnitt 6.1.2 wird gezeigt, wie durch die Teilung eines einzelnen Rechteckes $R_{j(alt)}$ in zwei Rechtecke $R_{j(neu)}$ und R_{r+1} eine neue und zulässige Teilung P erzeugt wird. Ein auf diese Weise gefundener Platz für M_{r+1} wird *Einzonenplatz* genannt. Abbildung 6-8 zeigt, wie durch Teilung von R_4 die Rechtecke R_4 und R_6 entstehen.

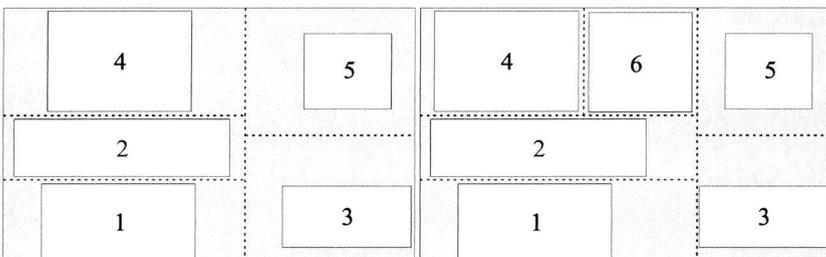


Abbildung 6-8: Beispiel eines Layouts vor und nach Plazieren von M_6 (M_{r+1})

In Abschnitt 6.1.3 wird erklärt, wann mehrere Rechtecke geteilt werden, um R_{r+1} zu erzeugen. Ein auf diese Weise gefundener Platz für M_{r+1} wird *Mehrzonensplatz* genannt.

6.1.2 Einzonensplatz

In diesem Abschnitt wird gezeigt, wie mit Hilfe von Einzonensplätzen Layouts generiert werden. Beim dabei verwendeten Algorithmus werden die Matrizen $\{M_1, M_2, \dots, M_m\}$ des Teilesetkandidaten TSK sequentiell von M_1 bis M_m gelegt. Die Reihenfolge der Matrizen innerhalb des TSK ist durch den Modul ‚Teilesetwahl‘ bestimmt worden. Bei anfänglich leerem Trog kann M_1 sofort gelegt werden, woraus eine 1-Teilung des Troges resultiert. Dann wird (m-1) mal derselbe, im letzten Abschnitt bereits gezeigte Schritt durchgegangen:

- Gegeben ist eine k-Teilung für M_1, \dots, M_k .
- Gesucht ist eine (k+1)-Teilung für M_1, \dots, M_{k+1} .

Hierfür wird *iterativ* für alle Matrizen $M_i \in \{M_1, \dots, M_k\}$ geprüft, ob M_{k+1} in einem der vier freien Plätze von M_i platziert werden kann, ev. nach Drehung um 90° . Alle auf diese Weise gefundenen freien Plätze für M_{k+1} werden bewertet (auf die Bewertung wird in Abschnitt 6.4 eingegangen). Am Schluss wird das Rechteck $R_j = R_{z(i)}$ mit der höchsten Bewertung in $R_{j(\text{neu})}$ und R_{k+1} aufgeteilt. Die entsprechenden Änderungen der k-Teilung und der Adjazenzmatrix sind trivial.

Eine aus lauter Einzonensplätzen generierte Teilung P hat folgende Eigenschaften:

- Jedem Rechteck in P ist eine Matrize zugewiesen (keine leeren Zonen).
- Die Einzonens-Platzierung trennt eine freie Zone stets im *Guillotineschnitt*, womit auch ein gesamtes Troglayout stets aus Guillotineschnitten aufgebaut ist.

Bei einem Layout aus lauter Guillotineschnitten existiert immer mindestens eine Trennlinie, welche von einem Rand zum andern führt. Teilt man dieses Layout der Trennlinie entlang in zwei Sublayouts, existiert wiederum pro Sublayout mindestens eine derartige Trennlinie. Diese Teilung kann sooft durchgeführt werden, bis in jedem Sublayout nur noch ein Rechteck R_j liegt. Die Abbildungen 6-6 und 6-8 zeigen derartige Layouts.

Im Weiteren gilt für Einzonensplätze, dass die Vereinigung der neuen freien Zonen von M_j und M_{k+1} gleich der alten freien Zone von M_j (vgl. Abbildung 6-9) ist.

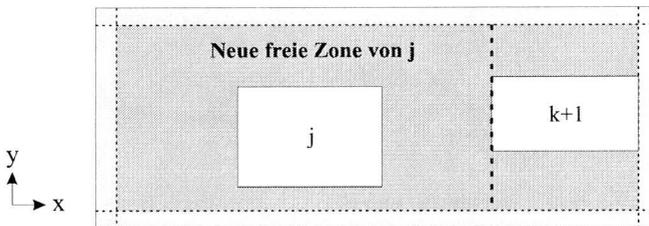


Abbildung 6-9: Neue freie Zone der Matrize M_j .

Können alle Matrizen aus TSK platziert werden, wird ein festes Layout erzeugt und gemäß Abschnitt 6.4 bewertet und der Wert wird an das übergeordnete Modul gemeldet. Konnten nicht alle Matrizen platziert werden, wird dies ebenfalls gemeldet, allerdings ohne Bewertung. Diese Information wird bei der Teilesetwahl (Abbildung 5-4) benötigt.

6.1.3 Mehrzonenplatz

Abbildung 6-10 zeigt eine Situation, in der bereits fünf Matrizen gelegt wurden. Für M_6 wird mit der oben beschriebenen Einzonenplatz-Methode der offensichtlich vorhandene Platz nicht gefunden, da die jeweiligen freien Plätze zu klein sind. Zur Veranschaulichung sind die Projektionen der freien Plätze $p_y(4)$, $p_y(2)$ und $p_y(5)$ am oberen Rand eingezeichnet (wo M_6 , um 90° gedreht, Platz fände).

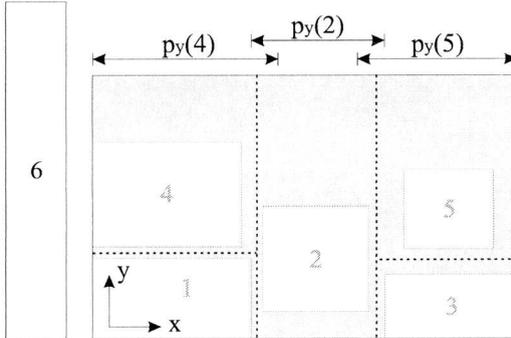


Abbildung 6-10: Versagen der Einzonenplatz-Methode

Um M_6 plazieren zu können, müssen die freien Plätze von M_2 , M_4 und M_5 zusammengelegt werden. In Abbildung 6-11 ist jeweils einer der beiden in y -Richtung freien Plätze von M_2 und M_4 eingezeichnet. Vom zweiten freien Platz ist am linken Rand die Projektion eingezeichnet.

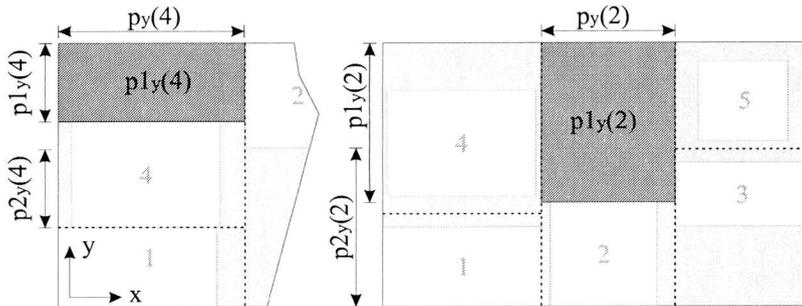


Abbildung 6-11: Benachbarte freie Plätze der Matrizen 2 und 4

Die beiden Plätze werden zusammengelegt, indem in ihre (flächenmässige) Vereinigung das grösstmögliche Rechteck gelegt wird. Für solche Rechtecke wird die Notation $p_{1y}(2,4), \dots, p_{4y}(2,4)$ eingeführt.

Abbildung 6-12 zeigt die Zusammenlegung der freien Plätze von M_2 und M_4 (der rechte Trog- rand ist jeweils abgeschnitten). Im oberen Teil des Bildes sind die freien Plätze der Matrizen eingezeichnet, im unteren Teil die resultierenden Rechtecke. Kann M_6 in einer der gemeinsa-

men freien Plätze $p_{1y}(2,4)$ bis $p_{4y}(2,4)$ gelegt werden, wird die Platzsuche an dieser Stelle abgebrochen. Ansonsten werden diejenigen freien Plätze $p_{y'}(2,4)$, die für M_6 noch breit genug sind, mit jenen von M_5 zusammengelegt, woraus die neuen freien Plätze $p_{y'}(2,4,5)$ resultieren.

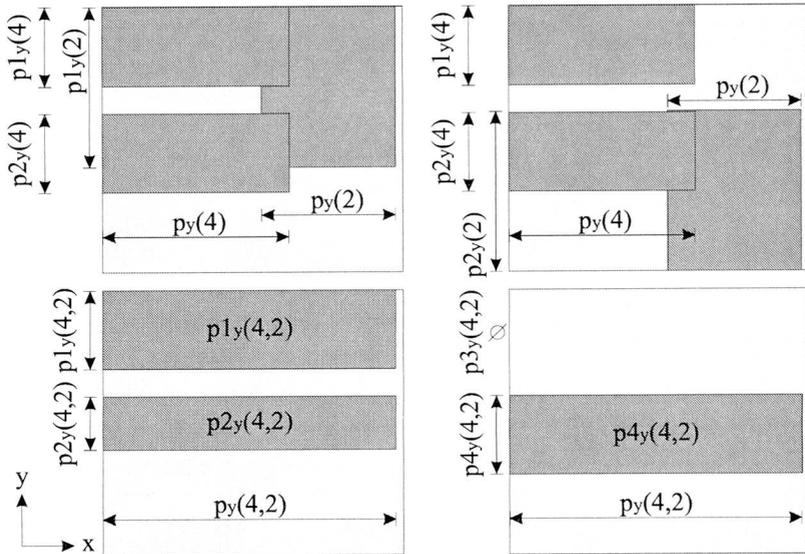


Abbildung 6-12: Zusammenlegung der benachbarten freien Plätze

Mehrzonenplätze generieren unter Umständen Layouts, die nicht aus Guillotineschnitten aufgebaut sind. Ein Beispiel dafür ist in Abbildung 6-13 zu sehen; Das Layout links besteht aus Guillotineschnitten. Lediglich die Matrizen M_2 und M_5 haben noch freie Plätze, und in diesen Mehrzonenplatz $p_{y'}(2,5)$ wird die neue Matrize M_7 eingefügt (Layout rechts). Nun liegt kein Layout aus reinen Guillotine-Schnitten mehr vor.

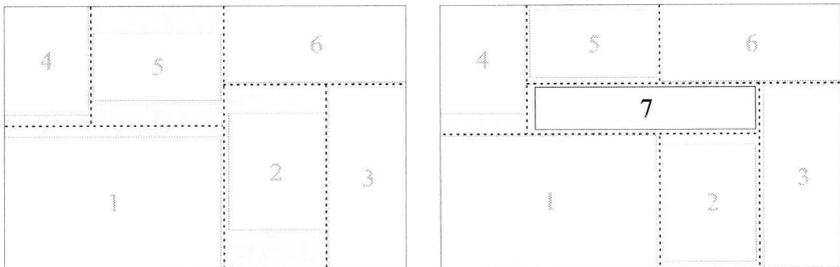


Abbildung 6-13: Beispiel eines Layouts ohne Guillotine-Schnitt (rechts)

Änderung der k-Teilung beim Plazieren von M_{k+1} in einen Mehrzonenplatz

Gegeben ist eine k-Teilung P_{alt} mit einer Folge $F: R_1, R_2, \dots, R_d$ von Rechtecken von P_{alt} , für die gilt:

- $a_{i,i+1} = 1; \quad 1 \leq i < d.$
- Für die Menge der Rechtecke F ist der Durchschnitt ihrer Projektionen auf die y-Achse nicht leer (M_{k+1} wird in eine gemeinsame freie Zone $p_y(1,2,\dots,d)$ gelegt).

Die Figur links in Abbildung 6-14 zeigt eine derartige Folge von Rechtecken F :

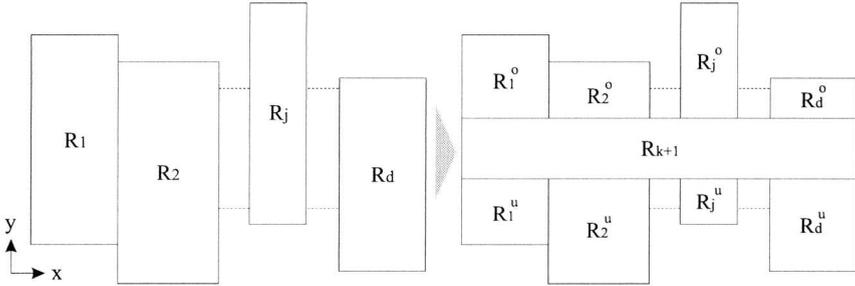


Abbildung 6-14: Aufteilung der Rechtecke bei einem Mehrzonenplatz

Eine k-Teilung P_{neu} wird erzeugt, indem die Folge F von d Rechtecken in $2d+1$ neue Rechtecke aufgeteilt wird. Hierfür wird jedes Rechteck R_j aus F dreigeteilt: Zwei dieser Teile von R_j werden zu den Rechtecken R_j^o und R_j^u , aus der Vereinigung aller (mittleren) dritten Teile der $R_j \in F$ entsteht R_{k+1} . Die Figur rechts in Abbildung 6-14 zeigt diese Aufteilung.

In der Adjazenzmatrix $A_{P_{neu}}$ werden dementsprechend für die neuen Rechtecke die unten aufgeführten Relationen eingetragen. Um die Lesbarkeit zu vereinfachen, wird an dieser Stelle für a_{ij} die Schreibweise $a(R_i, R_j)$ benutzt. Für $1 \leq i, j \leq d$:

$$\begin{aligned}
 a(R_i^o, R_j^o) &= a(R_i, R_j) \\
 a(R_i^u, R_j^u) &= a(R_i, R_j) \\
 a(R_i^o, R_j^u) &= \emptyset \\
 a(R_{k+1}, R_j^o) &= u \\
 a(R_{k+1}, R_j^u) &= o
 \end{aligned}$$

Und für ein Rechteck R von P_{alt} , das nicht zur Folge F gehört, gilt:

$$\begin{aligned}
 \text{Falls } a(R, R_1) &= 1 \Rightarrow a(R, R_1^o) = 1; a(R, R_1^u) = 1; a(R, R_{k+1}) = 1 \\
 \text{Falls } a(R_d, R) &= 1 \Rightarrow a(R_d^o, R) = 1; a(R_d^u, R) = 1; a(R_{k+1}, R) = 1 \\
 \text{Falls } a(R, R_j) &= u \Rightarrow a(R, R_j^u) = u \\
 \text{Falls } a(R_j, R) &= u \Rightarrow a(R_j^o, R) = u
 \end{aligned}$$

Sonst ist $a(R_j^u, R) = a(R_j^o, R) = a(R_{k+1}, R) = \emptyset.$

Falls für ein M_i sein $R_{z(i)}$ zu F gehört, dann muss M_i entweder $R_{z(i)}^o$ oder $R_{z(i)}^u$ zugeordnet werden. Oft ist [6-3] nur für eines dieser beiden Rechtecke erfüllt, womit die Zuweisung von M_i

klar ist. Ist [6-3] für $R_{z(i)}^o$ und für $R_{z(i)}^u$ erfüllt, erfolgt die Wahl gemäss den in Abschnitt 6.2.4 beschriebenen Algorithmen. M_{k+1} wird R_{k+1} zugeordnet.

Damit wurden d Rechtecke eingeführt, denen keine Matrizen zugeordnet sind. Aus technischen Gründen wird in jedes dieser Rechtecke eine (*Pseudo*-)Matrize mit Länge und Breite = 0 gelegt. Auch wenn sie real nicht existieren, haben sie trotzdem eine benutzbare freie Zone. Ob solche leeren Zonen und Pseudomatrizen entfernt werden können oder nicht, wird in Abschnitt 6.5.3 behandelt. Die in diesem Fall notwendigen Anpassungen in A_p werden ebenfalls dort beschrieben.

Bemerkung: Die freien Plätze einer Pseudomatrize entsprechen ihrer freien Zone.

(Gilt für die Folge von Rechtecken R die Beziehung $a_{i,i+1} = u$, so sind die Änderungen in A_p analog den oben beschriebenen durchzuführen).

6.2 Formalisierung des Ansatzes mit Hilfe gerichteter Graphen

Die Behandlung des Ungleichungssystems [6-4] ist der Kern des Ansatzes. Die Entscheidung, ob ein System von Ungleichungen für reale Variablen zulässig ist, ist im Allgemeinen keine triviale Aufgabe. Das hier betrachtete Ungleichungssystem ist sehr speziell und wird mittels gerichteten Graphen behandelt.

Die Variablen im System [6-4] entsprechen Trennlinien. Eine zulässige Lösung für [6-4] legt eine k -Teilung fest und ordnet jede Matrize M_i einem Rechteck $R_{z(i)}$ zu (wo M_i Platz findet). Eine Matrize M_i kann irgendwo in $R_{z(i)}$ gelegt werden. Eine zulässige Lösung für [6-4] liefert also Koordinaten, genaugenommen ein Intervall von möglichen Koordinaten, für die Position der Matrizen. Aufgrund des Handlingsaufwandes, der bei einem Layoutwechsel resultiert, werden die Matrizen wahrscheinlich nicht in eine Ecke von $R_{z(i)}$ gelegt, da bei einem Layoutwechsel oft nur kleine Änderungen im Koordinatenintervall auftreten und die Matrizen dann ständig um wenige Zentimeter verschoben werden müssten. Neben den Koordinaten für die Trennlinien werden nun auch für die Matrizen Koordinaten eingeführt: (X_i, Y_i) seien die Koordinaten der linken unteren Ecke einer Matrize M_i . Im bisher verwendeten Kontext ist (X_i, Y_i) also derjenige Punkt in M_i , welcher dem Koordinatenursprung am nächsten liegt (mit $X_i, Y_i \geq 0, \forall i$).

Damit nicht ständig zwei Typen von Variablen (für die Trennlinien sowie für die Matrizen) behandelt werden müssen, wird jetzt aus [6-4] ein System von Ungleichungen für die Matrizenvariablen abgeleitet. Zur Vereinfachung wird angenommen, dass $z(i) = i$, dass also M_i im Rechteck R_i liegt.

Ungleichungssystem für (X_i, Y_i)

Gegeben eine k -Teilung für M_1, \dots, M_k , wird für die Koordinaten (X_i, Y_i) der Matrizen folgendes System von Ungleichungen erzeugt:

$$\begin{array}{ll}
 X_i, Y_i \geq 0 & \\
 X_i + u_i \leq L & \\
 Y_i + v_i \leq B & i = 1, \dots, k \qquad [6-5] \\
 \text{Falls } a_{ij} = l, \text{ dann} & X_i + u_i \leq X_j \\
 \text{Falls } a_{ij} = u, \text{ dann} & Y_i + v_i \leq Y_j
 \end{array}$$

Eine zulässige Lösung für [6-4] liefert eine zulässige Lösung für [6-5], wobei jede Matrize M_i möglichst unten links in $R_{z(i)}$ platziert wird. Umgekehrt beachte man, dass eine zulässige Lösung für [6-5] eine zulässige Lösung für [6-4] liefert, wie Abbildung 6-15 illustriert: Gegeben sei eine Trennlinie t mit der Koordinate x_t und der Familie der Rechtecke $R_1, R_2, \dots, R_k, \dots, R_m$, die eine Seite auf t haben (R_1, R_2, \dots, R_k liegen auf der linken Seite von t und R_{k+1}, \dots, R_m auf der rechten Seite von t (also $a_{ij} = 1$ für $1 \leq i \leq k, k+1 \leq j \leq m$)). Dann gilt die Definition

$$x_t = \max_{i=1, \dots, k} (X_i + u_i) \quad [6-6]$$

Es ist leicht zu sehen, dass damit eine zulässige Lösung für [6-4] entsteht.

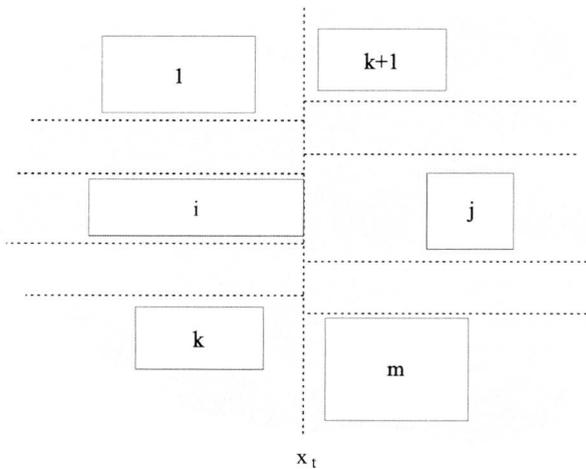


Abbildung 6-15: Zusammenhang der Ungleichungen [6-4] und [6-5]

Es stellt sich nun die Frage, wie das System von Ungleichungen [6-5] behandelt wird. Die Aufgaben, welche damit erfüllt werden sollen, sind hier zusammenfassend aufgelistet. Als *Zustandsbeschreibung* soll folgende Informationen über ein Layout bzw. über die darin platzierten Matrizen vorhanden sein:

1. Relative Position einer Matrize gegenüber andern platzierten Matrizen
2. Absolute Position einer Matrize im Trog
3. Freie Plätze p_x, p_y neben einer Matrize
4. Zusammengelegte freie Plätze p_x resp. p_y neben mehreren Matrizen

Die *Beschreibung der Zustandsänderung* - oder exakter: das Nachführen einer Zustandsänderung zur Beschreibung des neuen Zustandes - verlangt Regeln für

5. das Platzieren einer Matrize in einen Einzonenplatz,
6. das Platzieren einer Matrize in einen Mehrzonenplatz und
7. das Behandeln leerer Zonen, welche beim Entfernen von Matrizen von ihrem Platz oder beim Generieren von Mehrzonenplätzen entstehen können.

In Abschnitt 6.1 wurde das Konzept der k -Teilung eingeführt, um die relativen Positionen der Matrizen zu beschreiben. Der Vorteil dieses Ansatzes ist, dass die k -Teilungen sowie die entsprechenden Ungleichungssysteme augenfällig garantieren, dass ein überschneidungsfreies Layout vorliegt. Aus zwei Gründen wird jetzt eine neue Modellierungsform (nämlich Digraphen) für Layouts eingeführt:

- Digraphen liefern einfache Algorithmen zur Behandlung der Ungleichungssysteme.
- Digraphen sind geeigneter als k -Teilungen, um Restriktionen für die absolute Position von Matrizen im Trog zu beachten (z.B. Tabuzonen).

Man wird sehen, dass durch die Benützung gerichteter Graphen die ersten beiden oben aufgeführten Punkte quasi automatisch erfüllt sind, und für den dritten Punkt kann man auf bestehende Methoden der Graphentheorie zurückgreifen. Für die Erfüllung der restlichen Punkte müssen eigene Methoden entworfen werden. Bevor auf diese Methoden eingegangen wird, werden einige Begriffe der Graphentheorie vorgestellt.

6.2.1 Grundbegriffe der Graphentheorie

Gemäss NEUMANN & MORLOCK ist das Gebiet der Graphen und Netzwerke „neben der linearen Optimierung dasjenige Teilgebiet des Operations Research, dessen Methoden in der Praxis am häufigsten angewendet werden“. Entsprechend gibt es viel Literatur zum Thema, die hier nur soweit zitiert wird, wie es zum Verständnis dieser Arbeit notwendig ist. ZIMMERMANN definiert Graphen wie folgt:

Ein Graph $G(V,E)$ besteht aus einer Menge V (Knoten, vertices) und einer Menge E (Kanten, edges) mit $V \neq \emptyset, V \cap E = \emptyset$ sowie einer auf E definierten Abbildung Ψ (Inzidenz), die jedem $e \in E$ eine Menge von zwei, möglicherweise identischen, Elementen $v, v' \in V$ zuordnet. Ist das jedem $e \in E$ zugeordnete Paar von Elementen aus V geordnet, so sind die Kanten gerichtet und man spricht von einem gerichteten Graphen (Digraph). Anderenfalls handelt es sich um einen ungerichteten Graphen.

Weiter gelten folgende Vereinbarungen aus ZIMMERMANN:

- Die Knoten seien numeriert (indiziert)
- Ist eine Kante e ungerichtet, schreibt man: $e = (i,j)$
- Ist eine Kante e gerichtet, schreibt man: $e = \langle i,j \rangle$ und bezeichnet sie als *Bogen* oder *Pfeil*
- Bei gerichteten Kanten bezeichne v_i (kurz i) den *Anfangsknoten* und v_j (kurz j) den *Endknoten* einer Kante
- Durch eine Abbildung $d: E \rightarrow \mathbb{R}^n$ werden den Kanten eines Graphen Längen oder Gewichte zugeordnet
- Annahmen:
 - $G(V,E)$ hat endlich viele Kanten und Knoten
 - $G(V,E)$ hat weder Schlingen noch parallele Kanten. Eine Kante wird als Schlinge bezeichnet, wenn sie mit einem einzigen Knoten inzidiert. Sind zwei Kanten inzident zu denselben zwei Knoten, spricht man von parallelen Kanten
- Existiert in einem Digraphen von einem beliebigen Knoten i aus in Pfeilrichtung *kein* Kantenzug (geschlossene Bogenfolge), durch den man wieder zu i gelangt, liegt ein *zyklenfreier* Digraph vor

Ein Bogen $e = \langle i, j \rangle$ und seine Knoten werden üblicherweise wie in Abbildung 6-16 dargestellt:

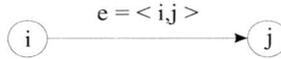


Abbildung 6-16: Bogen eines Digraphen mit Anfangsknoten i und Endknoten j

i wird (unmittelbarer) Vorgänger (engl.: Predecessor) von j genannt, und j (unmittelbarer) Nachfolger (engl.: Successor) von i . Es lassen sich damit zwei wichtige Mengen definieren:

$$P(j) = \text{Menge der Vorgänger von } j = \{i \in V \mid \exists e \in E \text{ mit } e = \langle i, j \rangle\}$$

$$S(i) = \text{Menge der Nachfolger von } i = \{j \in V \mid \exists e \in E \text{ mit } e = \langle i, j \rangle\}$$

Ein Knoten i , der keine Vorgänger hat, wird *Quelle* q genannt, somit gilt: $P(q) = \emptyset$. Ein Knoten j , der keine Nachfolger hat, wird *Senke* s genannt, somit gilt: $S(s) = \emptyset$.

6.2.2 Graphen als Mittel zur Darstellung von Layouts

Zur Darstellung der geometrischen Beziehungen eines Layouts mit gegebener k -Teilung wird pro Trogdimension (x, y) je ein zyklensfreier Digraph benutzt. Die beiden Digraphen G_x, G_y werden beschrieben durch

$$\begin{aligned} G_x &= \langle V, E_x, d_x \rangle \\ G_y &= \langle V, E_y, d_y \rangle \end{aligned} \tag{6-7}$$

G_x, G_y haben dieselbe Knotenmenge $V = \{q, v_1, v_2, \dots, v_m, s\}$. v_i (kurz i) sei die im Trog liegende Matrize i (geometrisch: ihre Ecke mit den tiefsten Koordinaten). Die Quelle q sei die Trogecke am Koordinatenursprung, die Senke s sei die gegenüberliegende Ecke. E_x ist definiert durch

$$\begin{aligned} E_x = \quad & \{ \langle i, j \rangle \mid a_{ij} = 1 \} \cup \\ & \{ \langle q, i \rangle \mid R_i \text{ am linken Trogrand} \} \cup \\ & \{ \langle i, s \rangle \mid R_i \text{ am rechten Trogrand} \} \end{aligned} \tag{6-8}$$

Die Bogenlänge d_x wird wie folgt definiert: $d_x(\langle i, j \rangle) = u_j$, $d_x(\langle q, i \rangle) = 0$, $d_x(\langle i, s \rangle) = u_i$. E_y wird analog definiert. Eine senkrechte Trennlinie zwischen zwei benachbarten Matrizen ist äquivalent zu einem Bogen in G_x , eine waagrechte Trennlinie zu einem Bogen in G_y . Eine mögliche Darstellung von G_x ist in Abbildung 6-17 gegeben. Im Trog befinden sich die Matrizen M_i und M_j , deren Längen in x -Richtung u_i, u_j betragen.

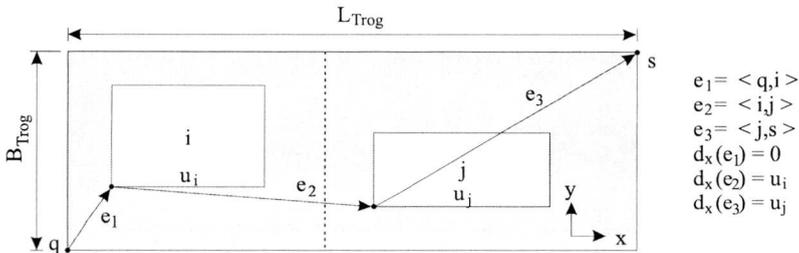


Abbildung 6-17: Digraph G_x eines Layouts

Ein Bogen $e_x \in G_x$ ist äquivalent zu folgender Ungleichung (X_i, X_j seien die noch nicht festgelegten x-Koordinaten der entsprechenden Matrizen):

$$e_x = \langle i, j \rangle_x \Leftrightarrow X_i + u_i \leq X_j \quad [6-9]$$

Die Koordinaten von q und s sind fest. Es gilt:

$$\begin{aligned} (X_q, Y_q) &= (0, 0) \\ (X_s, Y_s) &= (L, B) \end{aligned} \quad [6-10]$$

Das Ungleichungssystem [6-9], [6-10] entspricht genau dem x-Teil des Ungleichungssystems [6-5]. Entsprechend ist G_y äquivalent zum y-Teil des Ungleichungssystems aus [6-5]. Die zwischen den Digraphen G_x, G_y und der k-Teilung eines Layouts bestehende Äquivalenz ist in Abbildung 6-18 verdeutlicht.

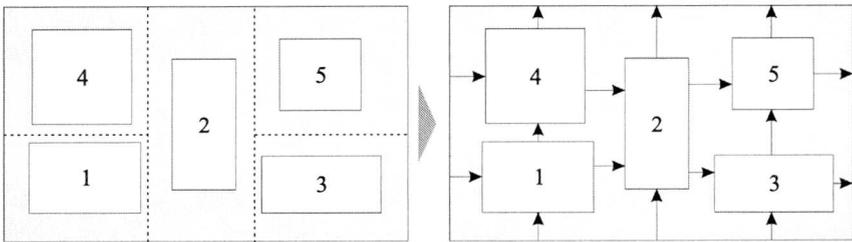


Abbildung 6-18: Analogie zwischen k-Teilung und Digraphen

Die Behandlung des Ungleichungssystems [6-9] beruht auf folgendem Satz:

Für einen zyklischen Digraphen G_x mit [6-9] und [6-10] gilt: [6-9], [6-10] hat eine zulässige Lösung, falls und nur falls der längste Weg zwischen q und s eine Länge $\leq L_{\text{Trog}}$ hat.

Sei für jeden Knoten i:

$\text{Min}X_i$ = Länge des längsten Weges zwischen q und i

$\text{Max}X_i$ = $L -$ (Länge des längsten Weges zwischen i und s),

dann sind die Vektoren $(\text{Min}X_i)_{i=1,\dots,k}$, $(\text{Max}X_i)_{i=1,\dots,k}$ und allgemeiner $(\lambda * \text{Min}X_i + (1-\lambda) * \text{Max}X_i)_{i=1,\dots,k}$ für $0 \leq \lambda \leq 1$ zulässige Lösungen von [6-9] und [6-10].

6.2.3 Extremalkoordinaten einer Matrize und ihre Berechnung

Wie gezeigt, kann sich eine plazierte Matrize M_j unter Einhaltung von [6-5] beliebig in ihrer freien Zone bewegen. Die folgende Notation wurde oben eingeführt: Die Koordinaten (X_j, Y_j) von M_j seien in der Minimalposition ihrer freien Zone ($\text{Min}X_j, \text{Min}Y_j$) und analog ($\text{Max}X_j, \text{Max}Y_j$) in der Maximalposition. Dann sind auch die Eckkoordinaten der freien Zone gegeben (vgl. Abbildung 6-19).

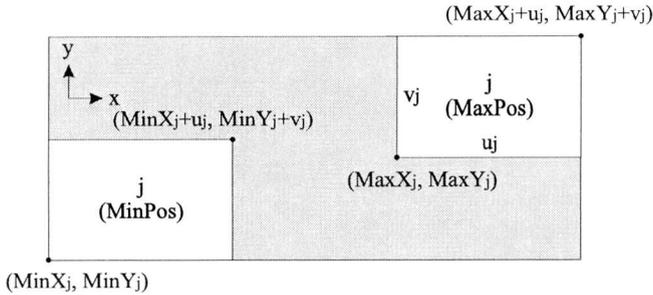


Abbildung 6-19: Koordinaten einer Matrize M_j und ihrer freien Zone

Für die Koordinaten (X_j, Y_j) von M_j gelten somit die Intervalle:

$$\begin{aligned} \text{Min}X_j &\leq X_j \leq \text{Max}X_j \\ \text{Min}Y_j &\leq Y_j \leq \text{Max}Y_j \end{aligned} \quad [6-11]$$

Berechnung der Koordinaten durch Berechnung der längsten Wege

Nach jedem Einfügen einer Matrize in ein Layout müssen die Extremalwerte für die Koordinaten der Matrizen nachgeführt werden. Mit Hilfe der in Abbildung 6-19 gezeichneten Koordinatenpunkte können dann für eine Matrize M_j die freien Plätze $p_x(j)$ und $p_y(j)$ berechnet werden. Für die Bestimmung der Extremalwerte müssen in den beiden Digraphen jeweils die *längsten Wege* berechnet werden, und zwar von der Quelle zu jeder Matrize M_j und von M_j zur Senke. In Abbildung 6-20 sind die längsten Wege für M_5 in x-Richtung eingezeichnet. Für eine Matrize M_j ergibt dabei

- der längste Weg von der Quelle q zu M_j die Minimalwerte $(\text{Min}X_j, \text{Min}Y_j)$, und
- der längste Weg von M_j zur Senke s die Maximalwerte $(\text{Max}X_j, \text{Max}Y_j)$ des für die Koordinaten X_j, Y_j von M_j zulässigen Intervalls.

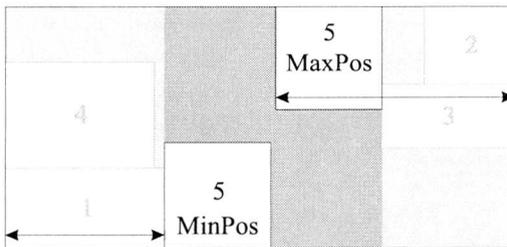


Abbildung 6-20: Längste Wege von M_5 in x-Richtung

In gewichteten Digraphen sind meist die kürzesten Wege zwischen zwei Knoten von Interesse, weshalb für deren Berechnung eine Reihe effizienter Algorithmen bekannt sind. Die Grundglei-

chungen, auf die sich viele dieser Verfahren stützen, stammen laut JUNGnickel von *Bellmann* (vgl. unten). Einer der bekanntesten Algorithmen zur Lösung dieser Gleichungen stammt von *Dijkstra*. *Zimmermann* beschreibt, dass einige Verfahren zur Bestimmung der kürzesten Wege auch zur Ermittlung der längsten Wege verwendet werden, indem z.B. die Pfeillängen mit -1 multipliziert werden. Dabei wird auch eine Modifikation des Verfahrens von *Dijkstra* aufgezählt, dessen Verfahren die kürzesten (bzw. längsten) Wege von einem beliebigen Knoten zu allen andern Knoten bestimmt.

In *Neumann & Morlock* werden ebenfalls einige Verfahren zur Berechnung kürzester Wege vorgestellt, unter anderem auch das Verfahren von *Bellmann*, mit welchem kürzeste Wege in zyklensfreien und *topologisch sortierten* Digraphen berechnet werden können. In einem topologisch sortierten Digraphen G sind die Knoten $v_j \in G$ derart indiziert, dass für jeden Vorgänger $v_i \in P(j)$ gilt: $i < j$.

Auch das *Bellmannsche* Verfahren ist mittels kleiner Modifikationen geeignet für die Berechnung längster Wege. Da hier die Berechnung eingeschränkt werden kann auf die längsten Wege zwischen der Quelle q und den Knoten j (bzw. zwischen den Knoten j und der Senke s), ist dieses Verfahren ideal für die vorliegende Anwendung. Im Folgenden werden die *Grundelemente des Verfahrens* zur Berechnung der Minimal- und Maximalwerte in G_x ($\text{Min}X_j, \text{Max}X_j$) erklärt; für nähere Erläuterungen sei auf die bereits erwähnte Literatur verwiesen. Unter der Annahme, dass die Knoten $v_j \in G_x$ topologisch sortiert seien, gilt:

1. Die Werte $\text{Min}X_j$ werden initialisiert: $\text{Min}X_q = 0$; $\text{Min}X_j = -1$ für $j = 1, \dots, k$.
2. Für $j = 1, \dots, k$ wird iterativ die *modifizierte Bellmannsche Gleichung* angewendet:

$$\text{Min}X_j = \max_{i \in P(j)} (\text{Min}X_i + u_i) \quad [6-12]$$

Für alle j , die am Trogrand bei der Quelle q liegen, ist $P(j) = \{q\}$ und somit $\text{Min}X_j = 0$. Am Schluss des Schleife von Punkt 2 kann überprüft werden, ob immer noch ein zulässiges Layout vorliegt: Bei Anwendung von Gleichung [6-12] auf die Senke s muss gelten: $\text{Min}X_s \leq L$.

Zur Berechnung der Maximalwerte von G_x muss in Schritt 1 $\text{Max}X_j = \infty$, $\text{Max}X_s = L$ gesetzt werden. In Schritt 2 muss der Zähler j rückwärts laufen: $j = k, \dots, 1$; die *Bellmannsche Gleichung* lautet nun:

$$\text{Max}X_j = \min_{i \in S(j)} (\text{Max}X_i - u_i) \quad [6-13]$$

Die Überprüfung des Layouts durch Anwendung von Gleichung [6-13] auf die Quelle q muss ergeben: $\text{Max}X_q \geq 0$. Die Werte für G_y werden analog ermittelt.

6.2.4 Algorithmen zum Finden freier Plätze

Einzonenzplatz

Sind die Extremalwerte der k bereits plazierten Matrizen aus TSK bekannt, ist für eine Matrize M_j ($j \leq k$) die Grösse ihrer freien Plätze $p_x(j)$, $p_y(j)$ bestimmt durch [6-14] (und [6-15]):

$$p_x(j) = (dX_j, dY_j + v_j); \quad p_y(j) = (dX_j + u_j, dY_j), \quad \text{mit} \quad [6-14]$$

$$dX_j = \text{Max}X_j - \text{Min}X_j; \quad dY_j = \text{Max}Y_j - \text{Min}Y_j \quad [6-15]$$

Bei der Erweiterung einer k -Teilung durch die Matrize $M_i (= M_{k+1})$ werden die freien Plätze $p_x(j)$, $p_y(j)$ jeder plazierten Matrize M_j daraufhin untersucht, ob M_i darin Platz findet. Hierzu werden folgende Bedingungen [6-16] (für $p_x(j)$) und [6-17] (für $p_y(j)$) überprüft:

$$(dX_j \geq u_i) \wedge (dY_j + v_j \geq v_i) \tag{6-16}$$

$$(dX_j + u_j \geq u_i) \wedge (dY_j \geq v_i) \tag{6-17}$$

Mehrzonenzplatz

Kann für M_i neben der plazierten Matrize M_j mit [6-16] bzw. [6-17] kein Einzonenzplatz gefunden werden, versucht man, M_i mit der Methode zum Auffinden von Mehrzonenzplätzen zu plazieren. In Abschnitt 6.1.3 wurde gezeigt, dass ein Mehrzonenzplatz aus einer Folge $F: R_1, R_2, \dots, R_d$ von Rechtecken generiert wird, für die $a_{j,j+1} = 1, 1 \leq j < d$, gilt. Es wird nun gezeigt, wie eine solche Folge mit Hilfe der Digraphen gefunden werden kann. Der Eintrag $a_{j,j+1} = 1$ in der Adjazenzmatrix A_P ist äquivalent zu einem Bogen $\langle j, j+1 \rangle_x$. Das heisst, man legt freie Plätze p_y von Matrizen zusammen, die auf einem Pfad in G_x liegen. Abbildung 6-21 veranschaulicht das Verfahren (mit $i = 6$).

(Analog dazu können Mehrzonenzplätze in G_y gesucht werden, wozu man freie Plätze p_x zusammenlegt. Die äquivalenten Einträge in A_P sind dann $a_{j,j+1} = u$).

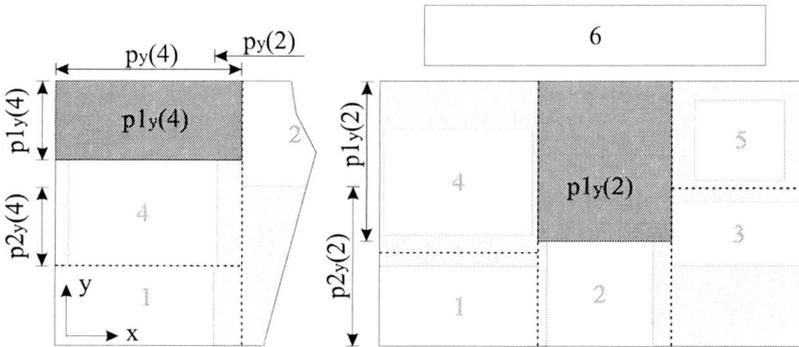


Abbildung 6-21: Suche eines Mehrzonenzplatzes für M_6

Gesucht ist wie in Abschnitt 6.1.3 bereits erklärt ein freier Platz $p_y(j, \dots, k-1, k)$, dessen Breite grösser gleich v_i ist und für dessen Länge gelten muss: $\text{Max}X_k + u_k - \text{Min}X_j \geq u_i$. Dieser freie Platz entspricht dem Pfad $j \rightarrow \dots \rightarrow k-1 \rightarrow k$ in G_x . Nun wird der Baum von Nachfolgern von M_j in G_x rekursiv nach einem solchen Pfad abgesucht.

Als *Startbedingung* für die Suche eines Mehrzonenzplatzes (in y -Richtung) muss die Überprüfung von [6-17] folgendes Resultat ergeben:

- $dX_j + u_j \geq u_i$: *nicht erfüllt* \Rightarrow Matrize M_i zu lang
- $dY_j \geq v_i$: *erfüllt* \Rightarrow Matrize M_i schmal genug.

Die rekursive Funktion $\text{Rek}(P)$ benutzt als Parameter einen Pfad $P: j \rightarrow \dots \rightarrow p$ in G_x . Beim Aufruf von $\text{Rek}(P)$ gilt:

- $N(p)$ ist die Liste der Nachfolger von p in G_x
- Es existiert mindestens ein Streifen $p_y(j, \dots, p-1, p)$, dessen Breite grösser gleich v_i ist
- $\text{Max}X_p + u_p - \text{Min}X_j < u_i$ (der Streifen ist nicht lang genug für M_i)

$\text{Rek}(P)$ wird initialisiert mit $P = j$ und $N(p) = S_x(j)$ und verläuft gemäss folgenden Schritten:

- Solange $N(p) \neq \emptyset$ ist:
 - $r =$ erstes Element in $N(p)$
 - Entferne r aus $N(p)$
 - Berechne $p1_y(r)$, $p2_y(r)$ und bilde alle Streifen $p1_y(j, \dots, p, r)$, $p2_y(j, \dots, p, r), \dots$
 - Falls mindestens ein Streifen existiert, dessen Breite $\geq v_i$ ist:
 - Prüfe $\text{Max}X_r + u_r - \text{Min}X_j \geq u_i$:
 - Erfüllt: Platz gefunden, Output ist der zulässige Streifen (Abbruch der Rekursion)
 - Nicht erfüllt: $\text{Rek}(j \rightarrow \dots \rightarrow p \rightarrow r)$

In der Implementation wird $\text{Rek}(P)$ nach dem ersten gefundenen Platz abgebrochen. Bricht man an dieser Stelle nicht ab, findet die Rekursion sämtliche Mehrzonenplätze mit M_j als Startelement.

Die Suche nach einem Mehrzonenplatz in x -Richtung verläuft analog.

6.3 Ablauf der Layoutgenerierung

Nachdem gezeigt wurde, wie Plätze für Matrizen gefunden werden, wird in diesem Abschnitt der Ablauf der Layoutgenerierung zusammenfassend aufgezeigt. Die dazu notwendigen Daten sind in Tabelle 6-1 zusammengestellt (plazierte Matrizen werden in diesem Abschnitt mit M_j bezeichnet).

| Input | Attribute | Bemerkungen |
|-------------|---|---|
| TSK | $= \{M_1, \dots, M_n\}$ | Teilesatkandidat mit den Matrizen M_i . Die Reihenfolge der M_i ist durch den Modul ‚Teilesatzwahl‘ gegeben. |
| Pro M_i : | w_i u_i, v_i h_i Tabuzonen TZ_i | Anzahl notwendige Pressungen von M_i Umhüllende von M_i Höhe der Matrize M_i Tabuzonen von M_i |
| Trog T: | L_T, B_T L_t | Länge (L_T) und Breite (B_T) des Troges Zu füllendes Layout |
| Von L_t | $\text{Anz}P_t$ $TS_t = \{\overline{M}_1, \dots, \overline{M}_m\}$ Position von \overline{M}_i in T | Anzahl Pressungen des Layouts L_t Teilesatz TS_t , zu welchem L_t gehört Lage der Matrizen aus TS_t im Trog, gegeben durch G_x, G_y |

Tabelle 6-1: Input des Moduls ‚Layoutgenerierung‘

Der Ablauf ist in Abbildung 6-22 aufgezeigt. Pro zu plazierte Matrize M_i wird bei jeder bereits plazierte Matrize \overline{M}_j geprüft, ob ein Einzonenplatz vorhanden ist. Falls nicht, wird geprüft, ob ein Mehrzonenplatz gefunden werden kann. Diese Platzsuche wird mit gedrehter Ma-

trize M_i wiederholt. Alle gefundenen Plätze werden bewertet und aufgrund dieser Bewertung wird am Schluss der inneren Schleife M_i an den besten Platz gelegt.

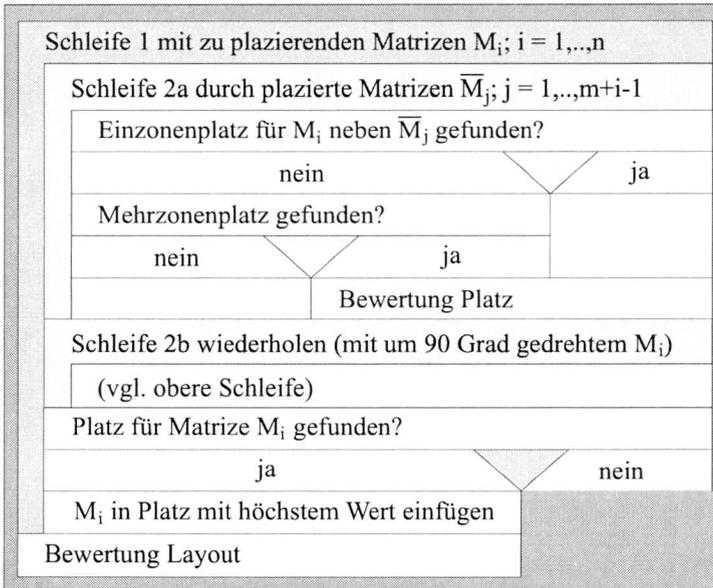


Abbildung 6-22: Ablauf der Layoutgenerierung

Können alle n Matrizen des TSK plaziert werden, wird ein zulässiges Layout bestimmt, indem die Matrizen in die Mitte ihrer freien Zonen gelegt werden. Das heisst für M_j :

$$\begin{aligned}
 X_j &= (\text{Min}X_j + \text{Max}X_j)/2 \\
 Y_j &= (\text{Min}Y_j + \text{Max}Y_j)/2
 \end{aligned}
 \tag{6-18}$$

Dieses neue Layout wird bewertet und zusammen mit seinem Wert an den Modul ‚Teilesetwahl‘ übergeben. Konnte eine Matrize nicht plaziert werden, wird die ganze Layoutgenerierung für dieses Teileset TS_i abgebrochen.

6.4 Bewertungsfunktionen

In diesem Abschnitt werden zuerst die Bewertungen für Einzonen- und Mehrzonenplätze vorgestellt. In einem ersten Schritt wird nicht zwischen den beiden freien Plätzen $p1_y(j)$ und $p2_y(j)$ in der gleichen Richtung unterschieden, da nur die Dimension des freien Platzes betrachtet wird. Die Wahl zwischen den beiden freien Plätzen erfolgt nach einer feineren Analyse.

Anschliessend wird die Bewertung ganzer Layouts erläutert, welche in die Zielfunktion eines Teilesets (vorgestellt in Abschnitt 5.1.3) einfließt. Darauf aufbauend erfolgt die Bewertung ganzer Layoutfolgen, wie sie insbesondere bei der Optimierung von Layoutfolgen benötigt wird, zu der in Abschnitt 6.7 einige Bemerkungen gemacht werden.

6.4.1 Bewertung und Wahl eines Platzes

Einonenplatz

Falls M_i in einem freien Platz von M_j Platz findet, dann wird dieser mögliche Platz mit der Funktion $Value(i;j)$ bewertet. Für einen Platz in $p_y(j)$ für M_i gilt:

$$Value(i;j) = ValueX(i;j) + ValueY(i;j) + ValueZ(i;j) \quad [6-19]$$

Der Gesamtwert eines Platzes $Value(i;j)$ setzt sich aus drei Werten zusammen. Die zu legende Matrize M_i und ihr Nachbar M_j werden bezüglich den Dimensionen x , y ($ValueX$, $ValueY$) sowie der Anzahl Pressungen ($ValueZ$) miteinander verglichen. Die einzelnen Terme aus [6-19] werden gemäss Tabelle 6-2 berechnet.

| Bereiche pro Dimension | Entsprechende Werte | Term |
|--|---|---------|
| $u_i < u_j$ $u_i = u_j$ $u_j < u_i < u_j + dX_j$ $u_i = u_j + dX_j$ | $ValueX(i;j) = u_i / u_j$; $ValueX(i;j) = 2$; $ValueX(i;j) = (u_i - u_j) / dX_j$; $ValueX(i;j) = 2$; | [6-19a] |
| $v_i < dY_j$ $v_i = dY_j$ | $ValueY(i;j) = v_i / dY_j$; $ValueY(i;j) = 2$; | [6-19b] |
| $w_i < w_j$ $v_i = w_j$ $w_i > w_j$ | $ValueZ(i;j) = w_i / w_j$; $ValueZ(i;j) = 2$; $ValueZ(i;j) = w_j / w_i$; | [6-19c] |

Tabelle 6-2: Teilwerte eines gefundenen Platzes $p_y(j)$

Gut ist die Platzauswertung beispielsweise, wenn die Umhüllenden zweier benachbarter Matrizen M_i und M_j gleich lange parallele Kanten haben. Denn damit kann aus den Umhüllenden (u_j, v_j) und (u_i, v_i) die gemeinsame Umhüllende $(u_j + u_i, v_j)$ bzw. $(u_j, v_j + v_i)$ werden. Ebenfalls hoch bewertet wird, wenn M_i die gesamte Länge bzw. Breite eines freien Platzes $p_y(j)$ ausfüllt. Und haben zwei benachbarte Matrizen M_i , M_j die gleiche Anzahl (verbleibende) Pressungen, werden am Schluss ihrer Bearbeitung zugleich zwei benachbarte Plätze frei.

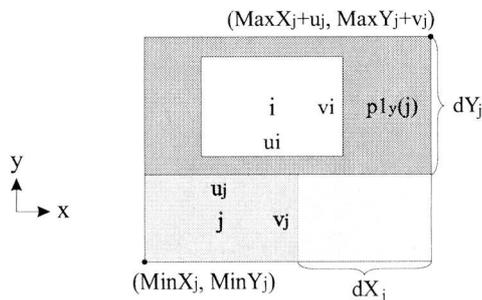


Abbildung 6-23: Grafik zur Bewertung eines Einonenplatzes

Mehrzonenzplatz

Die Bewertung eines Mehrzonenzplatzes verläuft analog zu jener eines Einzonenzplatzes. Für einen gefundenen Platz $p_y(j, \dots, p)$ gilt:

$$\text{Value}(i; j, \dots, p) = \text{ValueX}(i; j, \dots, p) + \text{ValueY}(i; j, \dots, p) + \text{ValueZ}(i; j, \dots, p) \quad [6-20]$$

mit den zwei Termen (und \bar{n} für die Anzahl Nachbarn)

$$\text{ValueY}(i; j, \dots, p) = \frac{1}{\bar{n}} \cdot \sum_{\kappa=j}^p \text{ValueY}(i; \kappa) \quad [6-21]$$

$$\text{ValueZ}(i; j, \dots, p) = \frac{1}{\bar{n}} \cdot \sum_{\kappa=j}^p \text{ValueZ}(i; \kappa) \quad [6-22]$$

Für die Berechnung von ValueX wird der Term [6-19a] verwendet, allerdings mit einer modifizierten Länge \bar{u}_i anstelle von u_i , wie sie in Abbildung 6-24 dargestellt ist:

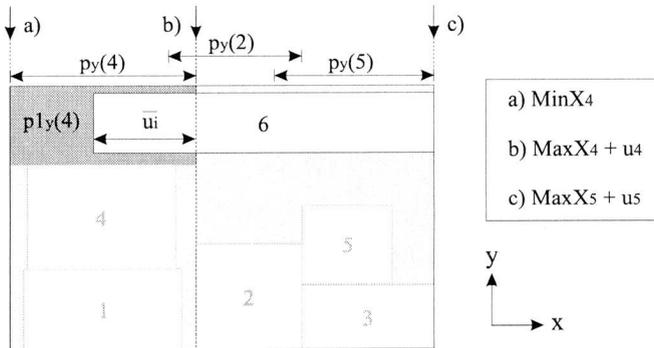


Abbildung 6-24: Modifizierte Länge bei der Bewertung eines Mehrzonenzplatzes (Beispiel)

Hierfür wird u_i um denjenigen Betrag gekürzt, um den $p_y(j, \dots, p) (= \text{MaxX}_p + u_p - \text{MinX}_j)$ länger ist als $p_y(j) (= \text{dX}_j + u_j = \text{MaxX}_j + u_j - \text{MinX}_j)$:

$$\bar{u}_i = u_i - (\text{MaxX}_p + u_p - (\text{MaxX}_j + u_j)) \quad [6-23]$$

Mit dieser Definition von \bar{u}_i werden bei Mehrzonenzplätzen dieselben „Konstellationen“ gut bewertet, wie bei Einzonenzplätzen.

Im oben gezeigten Beispiel sind die Nachbarn (j, \dots, p) der neu zu legenden Matrize M_6 die Matrizen mit den Nummern (4,2,5).

Ist der zuletzt dazugekommene Nachbar M_p in x -Richtung relativ lang, können \bar{u}_i und somit auch ValueX negativ werden. Dieser Wert wird bewusst negativ belassen, denn es existiert mindestens ein weiterer Ein- oder Mehrzonenzplatz für M_i , und zwar derjenige der jetzigen Nachbarn ohne die erste Matrize M_j , der dadurch besser bewertet wird.

Wahl zwischen den Plätzen $p_{1y}(j)$ und $p_{2y}(j)$ und Behandlung hoher Matrizen

Für M_i sei neben M_j der Platz $p_{y}(j)$ gewählt worden, d.h. der Platz hat gemäss [6-19] die höchste Bewertung erzielt. Der Entscheid, ob M_i in $p_{1y}(j)$ oder in $p_{2y}(j)$ gelegt wird, basiert auf der untenstehenden Entscheidungstabelle, in der die folgenden Fälle unterschieden werden:

- Fall I: Die Matrizen M_i, M_j waren beide im vorhergehenden Layout noch nicht plaziert.
- Fall II: Eine oder beide Matrizen M_i, M_j waren im vorhergehenden Layout bereits plaziert. Somit muss für mindestens eine Matrize ein Handlingsaufwand berechnet werden (vgl. unten).

In Abschnitt 3.3.1 wurden Beschickungsregeln postuliert, von denen sich einige insbesondere auf die Höhe einer Matrize beziehen (für jede Matrize muss einmal von einem Bediener in FluidPlan die Gesamthöhe eingegeben werden. Ist diese grösser als ein einstellbarer Wert, gilt sie als hohe Matrize). Deshalb wird bei der Wahl zwischen $p_{1y}(j)$ und $p_{2y}(j)$ auch beachtet, ob die Matrizen M_i, M_j flach oder hoch sind:

- Relation 1: M_i, M_j sind flache Matrizen
- Relation 2: Eine der Matrizen ist hoch, die andere flach
- Relation 3: M_i, M_j sind hohe Matrizen

Aus den Fällen und Relationen ergeben sich 6 Kombinationen für die Entscheidungstabelle, bei denen die unten aufgeführten Plazierungsregeln in unterschiedlicher Auswahl und Reihenfolge angewendet werden (die Regeln B und C sind aus Beschickungsregeln abgeleitet):

Die Wahl zwischen den Plätzen $p_{1y}(j)$ und $p_{2y}(j)$ basiert auf der Tabelle 6-3. Sind für eine Kombination die Regeln A, C angegeben, heisst das: Wahl gemäss Regel A; falls kein Entscheid nach Regel C; falls kein Entscheid, wird M_i in $p_{1y}(j)$ plaziert.

| | Relation 1 | Relation 2 | Relation 3 |
|-----------------|------------|------------|------------|
| Fall I: | - | B, C | C |
| Fall II: | A | B, A, C | A, C |

Tabelle 6-3: Entscheidungstabelle bei der Plazierung von M_i neben M_j

- Regel A: Platzwahl so, dass der Handlingsaufwand für diejenigen Matrizen M_i, M_j , die bereits im Trog liegen, minimal wird (die Berechnung des Handlingsaufwands wird später erklärt)
- Regel B: Platzwahl so, dass diejenige Matrize M_i, M_j , die hoch ist, möglichst nicht am Trogrand liegt
- Regel C: Die Vorgänger und Nachfolger des freien Platzes von M_j werden ebenfalls betrachtet. Platzwahl so, dass die Anzahl hoher Matrizen, welche nebeneinander liegen, minimal ist.
- sonst: M_i wird in $p_{1y}(j)$ plaziert.

Bei *Mehrzonenplätzen* wird analog zur hier beschriebenen Methode vorgegangen (falls für den gefundenen Platz $p_y(j, \dots, p)$ überhaupt mehrere Varianten möglich sind). Die Verschiebungswerte sämtlicher Matrizen i, j, \dots, p , welche bereits in der vorhergehenden Pressung im Trog lagen, werden aufsummiert. Bei Regel C werden nur die Vorgänger und Nachfolger der ersten Matrize M_j betrachtet, um die Anzahl Abklärungen und Fallunterscheidungen gering zu halten (im Bewusstsein, dadurch manchmal auch „schlechte“ Plazierungen zu bestimmen).

Um die Berechnung des Handlingsaufwandes zu erklären, wird ein Beispiel mit dem Fall II gezeigt, bei welchem M_1 neu dem Trog zugewiesen wird und M_2 bereits in der letzten Pressung im Trog lag. M_1 wird im Layout L_t einmal in $p_{1y}(j)$ und einmal in $p_{2y}(j)$ platziert und für M_2 wird jeweils die neue freie Zone berechnet. Vergleicht man diese Zonen mit der alten freien Zone von M_2 (im Layout L_{t-1}), erhält man ein Mass für den notwendigen Verschiebungsaufwand. In Abbildung 6-25 und Abbildung 6-26 betrachte man die Situation, wo erstens M_1 und M_3 entfernt werden und zweitens $M_8 (= M_1)$ in eine freie Zone neben $M_2 (= M_j)$ platziert werden soll (zum Entfernen von Matrizen vgl. Abschnitt 6.5.3).

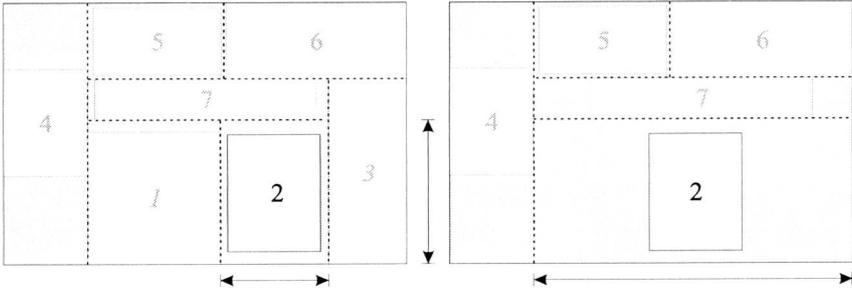


Abbildung 6-25: Freie Zonen von M_2 in Layout L_{t-1} (links) und nach Entfernen von M_1 und M_3

Bei der Ergänzung des Layouts durch Matrizen aus TSK wird die freie Zone von M_2 wieder verkleinert. Dabei sind zwei verschiedene Positionen für die neue freie Zone von M_2 möglich, welche in Abbildung 6-26 zu sehen sind. Durch die Grafik wird deutlich, dass die Position im linken Bild weniger Handlingsaufwand verursacht (M_8 wird also in $p_{2x}(2)$ gelegt).

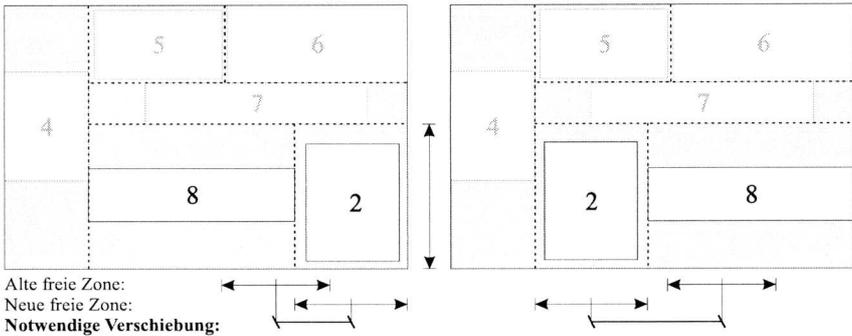


Abbildung 6-26: Freie Zonen von M_2 in L_{t-1} und bei verschiedener Platzwahl für M_8

Da bei einem Layout die Extremalpositionen der Ecke mit den tiefsten Koordinaten (Bezugspunkt) festgehalten werden, wird der Handlingsaufwand aus diesen Werten abgeleitet. Für die freie Zone einer Matrize M_j ist die gemittelte Lage (X_j, Y_j) des Bezugspunktes gegeben durch:

$$\begin{aligned} X_j &= (\text{Min} \bar{X}_j + \text{Max} X_j) / 2 \\ Y_j &= (\text{Min} Y_j + \text{Max} Y_j) / 2 \end{aligned} \quad [6-24]$$

Auf diese Weise wird die oben gezeigte notwendige Verschiebung dist_j berechnet. Diejenige freie Zone mit der kleineren Verschiebung wird gewählt. Gleichung [6-25] zeigt die Berechnung von dist_j . (X_j, Y_j) ist dabei die Lage des Bezugspunktes in L_t , (\bar{X}_j, \bar{Y}_j) die Lage des Bezugspunktes in L_{t-1} .

$$\text{dist}_j = \sqrt{(X_j - \bar{X}_j)^2 + (Y_j - \bar{Y}_j)^2} \quad [6-25]$$

Mit diesem einfachen Ansatz zur Berechnung der Verschiebung wird ein Mittelweg begangen. Abbildung 6-27 zeigt mögliche Alternativen auf. Die gezeichneten Rechtecke spannen den Bereich auf, in welchem der Bezugspunkt (X_j, Y_j) einer Matrize M_j liegen kann.

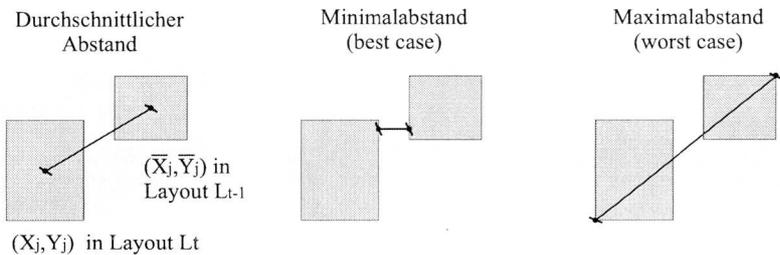


Abbildung 6-27: Mögliche Ansätze, die Verschiebung zu berechnen

Erste Alternative zum gewählten Ansatz wäre ein ‚best case‘-Ansatz, bei dem geprüft wird, was die minimale notwendige Verschiebung zwischen der alten und einer neuen freien Zonen ist. Zweitens bietet sich der ‚worst case‘-Ansatz an, bei dem die entsprechenden maximalen Verschiebungen betrachtet werden. Folgender Grund spricht für die getroffene Wahl: Ohne weitere Information über die spätere Lage der Matrize ist es vernünftiger, davon auszugehen, dass die Matrizen üblicherweise tatsächlich im zur Verfügung stehenden Platz eingemittelt werden.

6.4.2 Bewertung eines Layouts

Für die Bewertung eines gesamten Teilesets gilt Gleichung [5-1]:

$$\text{Value}(TS_t) = \text{SetValue}(TS_t) - \text{LayoutValue}(TS_t) \quad [6-26]$$

In Kapitel 5 wurde ausgeführt, wie man den ersten Teil dieser Gleichung ($\text{SetValue}(TS_t)$) erhält. In diesem Abschnitt wird gezeigt, wie der zweite Teil der Gleichung berechnet wird. Der $\text{LayoutValue}(TS_t)$ eines Teilesets TS_t ist so ausgelegt, dass er „schlechte“ Layouts, d.h. solche mit unerwünschten Merkmalen, bestraft. Deshalb wird er in Gleichung [6-26] vom Gesamtwert abgezogen. Für die Berechnung von $\text{LayoutValue}(TS_t)$ gilt:

$$\text{LayoutValue}(TS_t) = \text{MoveValue}(TS_t) + \text{PosValue}(TS_t) \quad [6-27]$$

Pro TS_t werden bei der Teilesetwahl gemäss Abbildung 5-5 i.A. mehrere Layouts zusammengestellt. Das beste dieser Layouts ist dasjenige, für welches die Gleichung [6-27] minimal ist. Im Folgenden werden die beiden Teile dieser Gleichung näher vorgestellt.

Verschiebungswert $MoveValue(TS_t)$

Beim Verschiebungswert $MoveValue(TS_t)$ eines ganzen Layouts L_t wird analog zum Verschiebungswert $dist_j$ einer einzelnen Matrize M_j vorgegangen. Verschiebungen werden nun gemäss [6-28] quadratisch bestraft und zudem mit der Wurzel aus der Grundfläche F_j von M_j multipliziert. Und muss eine Matrize gedreht werden beim Wechsel von L_{t-1} zu L_t , soll dies ebenfalls bestraft werden. Hierfür wird in Gleichung [6-28] die Grundfläche F_j zum $MoveValue$ addiert.

$$MoveValue(TS_t) = \sum_{j \in TS_t \cap TS_{t-1}} \left\{ \frac{dist_j^2 \cdot \sqrt{F_j}}{100} + dreh_j \cdot F_j \right\} \quad [6-28]$$

Für $dreh_j$ gilt:

$$\begin{aligned} dreh_j &= 1, \text{ wenn } M_j \text{ gedreht wird von } L_{t-1} \text{ zu } L_t; \\ &= 0, \text{ sonst.} \end{aligned}$$

Bemerkung: Sämtliche Dimensionen sind in mm angegeben.

Nicht berücksichtigt wird das Gewicht der Matrizen, denn im Allgemeinen sind die normalerweise aus Hartholz gefertigten Matrizen relativ leicht. Selten vorkommende schwere Matrizen können auf der glatten Trogplatte ohne Anheben verschoben werden. Ebenso wird das pro Auftrag einmalige Hineinheben und Herausnehmen der Matrizen nicht berücksichtigt. Vgl. hierzu Abbildung 2-6, bei der im Hintergrund ein Zwischenlager für Matrizen zu sehen ist; der Weg vom Zwischenlager zur Presse ist einiges grösser als die Distanz vom Trogrand zur Trogmitte.

Positionswert $PosValue(TS_t)$: Bewertung der Nachbarpaare

Alle Matrizen eines Layouts TS_t werden bezüglich ihrer direkten Nachbarn bewertet. Die Grundidee ist dieselbe wie bei der Bewertung eines Platzes aus Abschnitt 6.4.1: Für zwei Nachbarmatrizen wird geprüft, ob die parallelen Kanten ähnlich lang und die Anzahl verbleibender Pressungen ähnlich hoch sind. Zuerst muss jedoch überprüft werden, ob sich die beiden Matrizen überhaupt berühren. Der Ablauf der Positionsbewertung verläuft in folgenden drei Schritten:

- Suche alle Nachfolger M_i von M_j ; $i \in S_x(j)$, $i \in S_y(j)$.
- Prüfe für ein Nachbarspaar M_i, M_j , ob sich die Rechtecke $R_{z(i)}, R_{z(j)}$ auf der gemeinsamen Trennlinie berühren. Falls ja: M_i, M_j sind *direkte Nachbarn*. Nur solche Paare werden bewertet. M_i, M_{k+1} in Abbildung 6-15 sind beispielsweise keine direkten Nachbarn.
- Mit einer an Tabelle 6-2 angelegten Skala werden direkte Nachbarn derart bewertet, dass eine Penaltyfunktion erzeugt wird. Bestraft wird auch, wenn sowohl M_i als auch M_j hohe Matrizen sind.

Schliesslich wird an dieser Stelle auch die Überschneidung einer Matrize mit ihrer Tabuzone bestraft.

6.4.3 Bewertung einer Layoutfolge

Werden für dieselben bereitliegenden Aufträge verschiedene Layoutfolgen (bzw. versch. Teilesefolgen TS) generiert und soll dann die beste davon bestimmt werden, kann die Zielfunktion aus Abschnitt 5.3.1 adaptiert werden:

$$\text{Value(TSL)} = \sum_{t=1}^{\bar{n}} \{g(p_t) \cdot \text{Value(TS}(p_t))\} \quad [6-29]$$

Dabei ist p_t die Nummer der Pressung und $\text{TS}(p_t)$ das Teileseset, welches bei der Pressung p_t auf dem Trog liegt. \bar{n} ist die Gesamtzahl geplanter Pressungen, und $g(p_t)$ ist ein Gewichtungsfaktor, der $\text{TS}(p_t)$ je stärker gewichtet, umso früher es eingeplant ist. Für $g(p_t)$ gilt:

$$g(p_t) = \frac{\bar{n} - t/2}{\bar{n}} \quad [6-30]$$

6.5 Manipulationen an den Digraphen

Beim Einfügen sowie beim Entfernen einer Matrize M_i müssen - analog zu den Änderungen in der k -Teilung - die Verbindungen der entsprechenden Knoten im Digraphenpaar G_x, G_y nachgeführt werden. Anschliessend muss man die Extremalwerte der im Digraphenpaar verbleibenden Matrizen neu berechnen. Beim Einfügen muss beachtet werden, ob der gewählte Platz ein Ein- oder ein Mehrzonenplatz war. Beim Entfernen eines Knotens spielt dieser Punkt keine Rolle. Das Entfernen einer Matrize wird zusammen mit der Behandlung von leeren Zonen betrachtet.

6.5.1 Einfügen einer Matrize in einen Einzonenplatz

M_i werde in den Einzonenplatz $p1_y(j)$ der Matrize M_j eingefügt. Aus Abbildung 6-28 wird deutlich, wie dabei die Bogen in den Digraphen G_x, G_y umgehängt werden müssen:

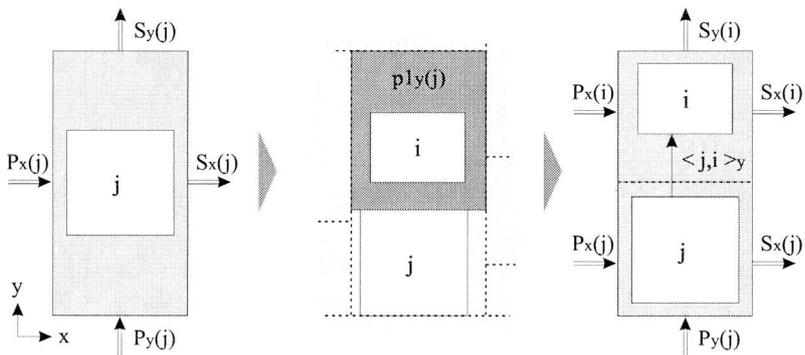


Abbildung 6-28: Einfügen in einen Einzonenplatz

Im Digraphen G_x gilt somit für die Plätze $p1_y(j)$ und $p2_y(j)$:

$$\begin{aligned} P_x^{\text{neu}}(i) &= P_x^{\text{neu}}(j) = P_x^{\text{alt}}(j) \\ S_x^{\text{neu}}(i) &= S_x^{\text{neu}}(j) = S_x^{\text{alt}}(j) \end{aligned} \quad [6-31]$$

Im Digraphen G_y gilt für $p1_y(j)$:

$$\begin{aligned} P_y^{\text{neu}}(i) &= \{j\} \\ S_y^{\text{neu}}(i) &= S_y^{\text{alt}}(j); \quad S_y^{\text{neu}}(j) = \{i\} \end{aligned} \quad [6-32]$$

Wird M_i in $p2_y(j)$ eingeführt, dann gilt:

$$\begin{aligned} P_y^{\text{neu}}(i) &= P_y^{\text{alt}}(j); \quad P_y^{\text{neu}}(j) = \{i\} \\ S_y^{\text{neu}}(i) &= \{j\} \end{aligned} \quad [6-33]$$

Wird M_i in $p1_x(j)$ oder $p2_x(j)$ plaziert, müssen die Gleichungen [6-31] bis [6-33] entsprechend abgeändert werden.

Berechnung der längsten Wege

Die Berechnung der längsten Wege basiert auf einer topologischen Sortierung der Knoten. Man beachte:

1. Bei der Einführung einer Matrize M_i neben M_j kann Knoten i sofort neben Knoten j in der topologischen Ordnung eingefügt werden (ohne die Ordnung neu zu berechnen).
2. Falls $u_i \leq u_j$ und falls M_i in $p_y(j)$ plaziert wurde, müssen in G_x die Extremalwerte nicht neu berechnet werden (vgl. Abbildung 6-28).
3. Für das Layout in Abbildung 6-20 ist (1,4,5,2,3) eine mögliche topologisch sortierte Reihenfolge der Matrizen für G_x , und M_5 sei die soeben plazierte Matrize. Es ist leicht einsehbar, dass bei der neuen Berechnung der Extremalwerte die MinX-Werte von M_1 und M_4 nicht verändert werden, und die MaxX-Werte von M_2 und M_3 ebenfalls nicht. Diese Bemerkung gilt allgemein für Vorgänger und Nachfolger.

6.5.2 Einfügen einer Matrize in einen Mehrzonenplatz

In Abschnitt 6.1.3 wurde die Änderung der k -Teilung beim Plazieren einer Matrize M_i in einen Mehrzonenplatz erklärt. Es wurde gezeigt, wie die entsprechenden Anpassungen in der Adjazenzmatrix A_P (von $A_{P_{\text{alt}}}$ zu $A_{P_{\text{neu}}}$) aussehen. Unter Einbezug der Gleichungen [6-5] und [6-8] können die dort beschriebenen Anpassungen von A_P zu Änderungen am Digraphenpaar G_x, G_y transformiert werden. Es wird darauf verzichtet, die Änderungen in G_x, G_y wie im letzten Abschnitt zu erklären, da für Mehrzonenplätze zuviel Fallunterscheidungen gemacht werden müssten (die bereits in Abschnitt 6.1.3 gemacht wurden).

Abbildung 6-29 zeigt, wie eine Matrize M_6 in einen Mehrzonenplatz neben M_1 und M_2 eingefügt wird. Um die Bilder übersichtlich zu halten, wurden anstelle der Bögen aus G_x, G_y die dazu äquivalenten Trennlinien eingezeichnet. Bild links zeigt die Trennlinien von $A_{P_{\text{alt}}}$, Bild rechts diejenigen von $A_{P_{\text{neu}}}$. Gemäss Anweisungen aus Abschnitt 6.1.3 resultieren neben dem Recht-

eck für M_6 auch leere Zonen für die Pseudomatrizen $M_{1'}$ und $M_{2'}$, auf deren Behandlung im nächsten Abschnitt eingegangen wird.

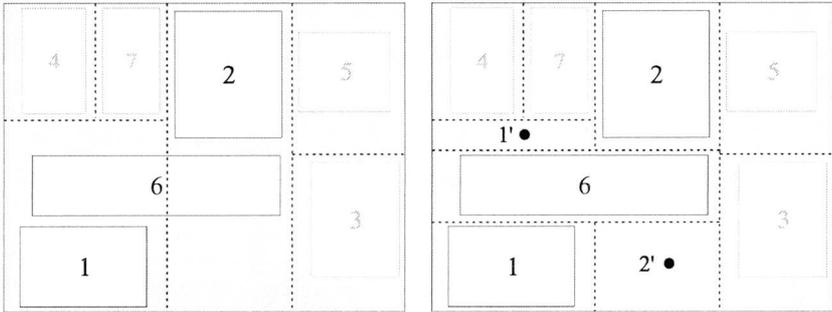


Abbildung 6-29: Layout nach dem Einfügen von M_6

6.5.3 Entfernen einer Matrize und Behandlung leerer Zonen

Leere Zonen entstehen in zwei Situationen:

- Beim Einfügen einer Matrize in einen Mehrzonenplatz werden mehrere leere Zonen eingeführt.
- Beim Entfernen einer Matrize M_i vom Trog wird in den Darstellungen (k-Teilung und Digraph) M_i durch eine Pseudomatrize ersetzt.

Diese dimensionslose Pseudomatrize wird benötigt, um die dazugehörige leere Zone - insbesondere den Platz, den sie repräsentiert - mit Hilfe des Digraphenpaares handhaben zu können. Durch das fortlaufende Einfügen und Entfernen von Matrizen aus dem Trog werden mit der Zeit immer mehr leere Zonen vorhanden sein. Deshalb stellen sich die beiden Fragen

- Welche leeren Zonen können entfernt werden?
- Wie werden leere Zonen entfernt?

Auffinden leerer Zonen, die entfernt werden dürfen

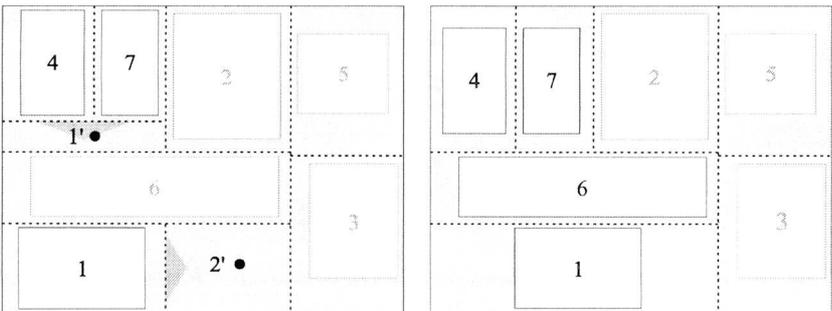


Abbildung 6-30: Nicht notwendige leere Zonen

Abbildung 6-30 zeigt im linken Bild eine k-Teilung eines Layouts mit zwei Pseudomatrizen. Im Bild rechts ist eine dazu äquivalente k-Teilung abgebildet, in der die Rechtecke der Pseudomatrizen die Fläche Null haben (indem die eine Trennlinie auf die gegenüberliegende Trennlinie gelegt wird). Offensichtlich kann die ganze Fläche, die diese Rechtecke einnehmen, von einem oder mehreren anderen Rechtecken übernommen werden. Das heisst, es existiert ein identisches Layout mit einer vereinfachten Teilung, bei der die Pseudomatrizen und ihre leeren Zonen entfernt worden sind.

Abbildung 6-31 zeigt eine k-Teilung, für die keine äquivalente k-Teilung existiert, in der die leere Zone der Pseudomatrix M_5 , die Fläche Null hat. Diese leere Zone kann deshalb nicht entfernt werden.

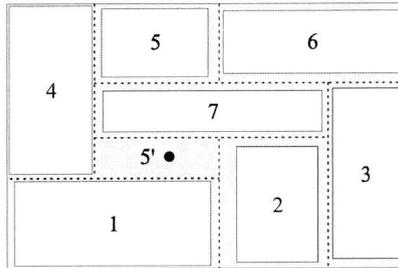


Abbildung 6-31: Layout mit leerer Zone, die nicht entfernt werden darf

Formal wird mit „Entfernen von leeren Zonen“ folgende Aufgabe gemeint:

Gegeben eine k-Teilung P_{alt} für M_1, \dots, M_r ($k > r$), gesucht ist eine k' -Teilung P_{neu} für M_1, \dots, M_r , sodass $k' < k$ und sodass für alle verbleibenden nicht leeren R_i gilt:

$$\text{Falls } a^{alt}_{i,j} \neq \emptyset, \text{ dann } a^{neu}_{i,j} = a^{alt}_{i,j}.$$

Falls eine zulässige Lösung für [6-4] existiert, wo parallele Trennlinien, die eine leere Zone begrenzen, übereinander geschoben werden können, dann ist das der graphische Beweis, dass eine einfachere Teilung für dasselbe Layout existiert. Um solche Situationen zu erkennen, werden jetzt die möglichen Lagen der Trennlinien betrachtet, die eine leere Zone R_i begrenzen:

- Fall 1: Eine Seite von R_i ist eine Trennlinie (Zeichnung links in Abbildung 6-32).
- Fall 2: Keine Seite von R_i ist eine Trennlinie. Man sieht, dass die relativen Lagen der Trennlinien, die R_i begrenzen, wie in Abbildung 6-32 (rechts) „turbinenartig“ aussehen.



Abbildung 6-32: Fälle leerer Zonen

Im Fall 1 wurde diese Seite als Guillotineschnitt erzeugt und es ist augenfällig, dass sie mit der gegenüberliegenden Linie in Deckung gebracht werden kann, womit R_i entfernt werden darf. Solche Fälle sind einfach erkennbar, indem R_i einziger Vorgänger seiner Nachfolger bzw. einziger Nachfolger seiner Vorgänger in G_x oder G_y ist. Sind die Extremalkoordinaten berechnet worden, können solche Fälle mit Hilfe von einer der folgenden Gleichungen erkannt werden:

$$\begin{aligned} & (\text{Min}X_j = \text{Min}X_i; \forall j \in S_x(i)) \cup \\ & (\text{Max}X_j + u_j = \text{Max}X_i; \forall j \in P_x(i)) \cup \\ & (\text{Min}Y_j = \text{Min}Y_i; \forall j \in S_y(i)) \cup \\ & (\text{Max}Y_j + v_j = \text{Max}Y_i; \forall j \in P_y(i)) \end{aligned} \quad [6-34]$$

Mit den ersten beiden Bedingungen können senkrechte Trennlinien identifiziert werden, waagrechte Trennlinien mit der dritten und vierten. Die zweite Bedingung entspricht der Situation der leeren Zone 2' in Abbildung 6-30, die dritte derjenigen von 1'.

Die Bedingungen in Formel [6-34] überdecken nicht alle Fälle, wo eine freie Zone entfernt werden kann, wie Abbildung 6-33 zeigt. Die dünne Linie zeigt eine mögliche Position, wo die beiden senkrechten Trennlinien zur Deckung gebracht werden können. Somit kann die leere Zone in der Mitte entfernt werden.

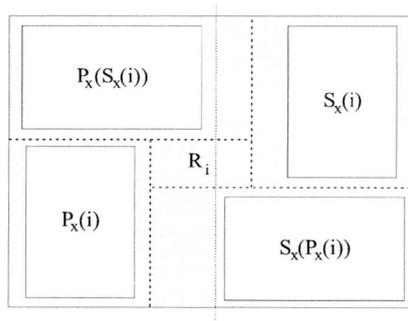


Abbildung 6-33: Entfernbar leere Zone R_i und neue Trennlinie

Die Bedingung, dass parallele Trennlinien, die eine leere Zone begrenzen, übereinander geschoben werden können, wird allgemein durch die Ungleichung [6-35] für senkrechte Trennlinien ausgedrückt. Für waagrechte Trennlinien gilt die analog die Ungleichung [6-36].

$$\text{Min}_{j \in S_x(i)} \{ \text{Min}X_j \} \leq \text{Max}_{j \in P_x(i)} \{ \text{Max}X_j + u_j \} \quad [6-35]$$

$$\text{Min}_{j \in S_y(i)} \{ \text{Min}Y_j \} \leq \text{Max}_{j \in P_y(i)} \{ \text{Max}Y_j + v_j \} \quad [6-36]$$

Entfernung der leeren Zone von M_i

Die Änderungen in den Digraphen, die der Änderung der k -Teilung entsprechen, sind folgende:

- Fallunterscheidung: Die Trennlinien, die übereinander gelegt werden können, sind:

- Senkrecht: Alle Vorgänger $P_x(i)$ von i werden Vorgänger aller Nachfolger $S_x(i)$ von i . Zudem werden die Knoten in der Menge $P_x(S_x(i))$ - also die Vorgänger der Nachfolger von i - Vorgänger der Knoten in der Menge $S_x(P_x(i))$ - also der Nachfolger der Vorgänger von i . Vgl. Abbildung 6-33.
- Waagrecht: Analoges Vorgehen
- Alle Bogen in G mit i als Anfangs- oder Endknoten werden entfernt
- Im Digraphenpaar G wird der Knoten i entfernt

Bemerkung: Im Fall 1 (Seite 116) ist eine der Mengen $P_x(S_x(i))$ oder $S_x(P_x(i))$ gleich dem Knoten i , der anschliessend entfernt wird. Die entsprechende Modifikation ist also irrelevant.

Leere Zonen bei einem Mehrzonenplatz

In Abschnitt 6.1.3 wurden bei der Einführung einer Matrize in einen Mehrzonenplatz $p_y(i, \dots, d)$ d leere Zonen eingeführt. Für $1 \leq i \leq d$ ist R_i^u oder R_i^o eine leere Zone. Man nehme an, dass $d > 2$ ist: Dann gehört für $1 < i < d$ eine solche leere Zone zum Fall 1, da zwei parallele Seiten ihre Extremitäten im inneren der gleichen Trennlinie haben (auf der Seite von R_{k+1}). Folglich ist eine solche leere Zone nicht turbinenartig und darf entfernt werden. Somit müssen höchstens zwei leere Zonen eingeführt werden.

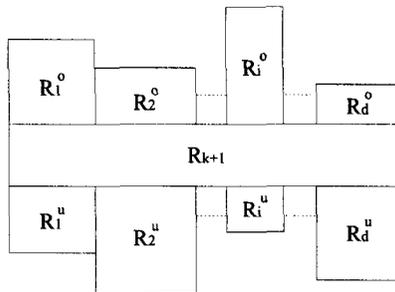


Abbildung 6-34: Einföhren d leerer Zonen bei einem Mehrzonenplatz

6.6 Behandlung von Tabuzonen

Auf Matrizen mit Tabuzonen muss in drei Situationen geachtet werden: Erstens, wenn eine solche Matrize plaziert werden soll; zweitens, wenn sie bereits plaziert ist, und für weitere Matrizen Plätze gesucht werden; drittens, wenn andere Matrizen entfernt werden.

Es sei noch einmal erwähnt, dass der Anteil an Matrizen mit Tabuzonen unter 3% aller Matrizen liegt. Das heisst, das Auftreten solcher Matrizen in einem Teilesetkandidaten TSK ist nicht der Normal-, sondern ein Spezialfall. Aus diesem Grund wurde in früheren Abschnitten darauf nicht eingegangen.

Falls Tabuzonen als strikte Restriktionen behandelt werden, genügt es, zusätzliche Bogen einzuföhren, die den Abstand einer Matrize zum Rand bzw. zu den Rändern begrenzen. Dieser Ansatz wird auch bei der verwendeten Modellierung angewendet, aber aus zwei Gründen verfeinert: Erstens ist das Ziel der Modellierung, solche Matrizen nicht gänzlich in ihre Tabuzonen zu legen, deshalb wird eine Überlappung der Umhüllenden und der Tabuzone bis zu einer gegebenen

Schranke S zugelassen. Zweitens haben v.a. Matrizen mit grossen Umhüllenden Tabuzonen, weshalb sie bei restriktiver Auslegung unter Umständen gar nicht ohne in die Tabuzone zu ragen platziert werden könnten.

6.6.1 Platzsuche für eine Matrize mit Tabuzone

Für die Matrize M_i (mit Tabuzonen) werde ein Platz gesucht. Wie in Abschnitt 6.2.4 beschrieben, werden hierfür die freien Plätze neben allen im Trog liegenden Matrizen überprüft. Man nehme an, die Wahl falle auf $p_x(j)$. Im Allgemeinen wird erst nach der Wahl dieses besten Platzes entschieden, ob M_i in $p_{1_x}(j)$ oder in $p_{2_x}(j)$ zu liegen kommt. Ist M_i aber eine Matrize mit Tabuzone, so werden $p_{1_x}(j)$ und $p_{2_x}(j)$ bereits vor der Bewertung näher untersucht. Je nach Lage eines freien Platzes und je nach Variante der Tabuzone von M_i entscheidet sich dabei, ob dieser Platz für unzulässig erklärt wird oder wie er zu bewerten (zu bestrafen) ist.

Zu Beginn wird M_i im freien Platz (gedanklich) eingemittet und es wird berechnet, wieviel Prozent der Umhüllenden in der Tabuzone liegt. Falls dieser Prozentsatz p höher als die Schranke S ist, wird der betreffende freie Platz für unzulässig erklärt. Sonst wird p in die Bewertung des freien Platzes mit einbezogen, sodass Plätze mit $p = 0$ immer höher bewertet werden als Plätze mit $p > 0$. Bei der Wahl zwischen $p_{1_x}(j)$ und $p_{2_x}(j)$ wird zusätzlich der Platz mit grösserem p als unzulässig erklärt.

Abbildung 6-35 zeigt die freien Plätze in x -Richtung neben M_i , in denen M_i platziert werden kann. Ebenso sind verschiedene mögliche Tabuzonen von M_i eingezeichnet, wie sie in Abbildung 3-13 vorgestellt wurden. Man nehme an, M_i sei genau so gross wie die gezeichneten freien Plätze und es gelte $S = 60\%$.

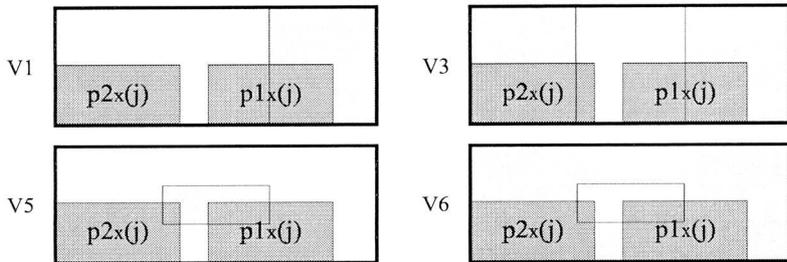


Abbildung 6-35: Trog mit Tabuzone von M_i und freien Plätzen $p_{1_x}(j)$, $p_{2_x}(j)$

Mit den vorgestellten Regeln gilt folgendes :

- Variante V1: Für $p_{1_x}(j)$ ist $p = 50\%$, also kleiner als S . Da für $p_{2_x}(j)$ $p = 0\%$ ist, wird $p_{1_x}(j)$ für unzulässig erklärt.
- Variante V3: Für $p_{2_x}(j)$ ist $p = 85\%$ und wird deshalb für unzulässig erklärt.
- Variante V6: Für beide Plätze ist $p > S$, somit sind beide unzulässig.

Die Variante V5 bildet einen Spezialfall, der auf relativ einfache Art behandelt wird: Für Matrizen mit solchen Tabuzonen sind nur freie Plätze zulässig, die in einer der vier Ecken des Troges liegen.

6.6.2 Platzsuche neben Matrizen mit Tabuzonen

Wie oben gesagt, werden beim Einführen einer Matrize M_j mit Tabuzonen zusätzliche Bogen (sog. *Tabubogen*) in die Graphen eingefügt. Sie müssen garantieren, dass die freie Zone von M_j so reduziert wird, dass nicht mehr als S Prozent von M_j in der entsprechenden Tabuzonen liegen.

Damit besteht das Risiko, „tote“ Zonen einzuführen, indem der abgeschnittene Teil in der freien Zone von M_j unter Umständen nicht mehr als freier Platz erkannt wird (z.B. wenn zwischen diesem Teil und dem Rand keine andere Matrize liegt). Dementsprechend muss man mit der Einführung von Tabubogen auch die aktuelle k -Teilung ändern und zusätzlich leere Zonen einführen, wie Abbildung 6-36 illustriert.

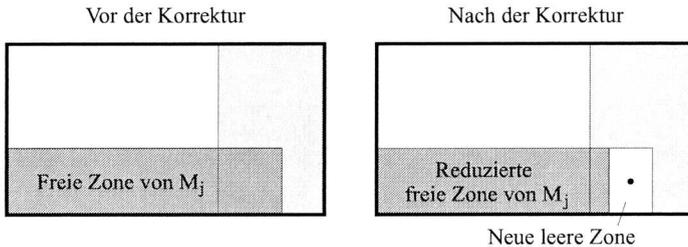


Abbildung 6-36: Korrektur der freien Zone von M_j aufgrund ihrer Tabuzone

Auf analoge Weise werden die Varianten V2, V3, V4 und V6 behandelt. V5 kann nicht mit Hilfe von Tabubogen behandelt werden. Für M_j mit Tabuzone V5 wird wie folgt garantiert, dass sie immer in einer der vier Ecken liegt: Bei der Untersuchung ihrer freien Plätze werden diejenigen zwei, die an einem Rand liegen, für unzulässig erklärt. Dadurch werden nie andere Matrizen zwischen M_j und den Rand plaziert.

Werden beim Plazieren eines M_j mit Tabuzonen wie oben erwähnt stets Tabubogen mit entsprechenden leeren Zonen eingeführt, muss M_j beim Entfernen anderer Matrizen nicht mehr speziell beachtet werden. Allerdings muss dafür beim Entfernen leerer Zonen geprüft werden, ob sie wegen eines M_j eingeführt wurden.

6.7 Konzept des Moduls Layoutoptimierung

In diesem Abschnitt werden einige Ansätze aufgezeigt, wie der Autor sich den Modul ‚Layoutoptimierung‘ vorstellt, welcher in Abschnitt 4.3.1 eingeführt wurde. Zuerst wird ausgeführt, wieso eine Optimierung einer mit den oben beschriebenen Methoden erzeugten Layoutfolge als sinnvoll erachtet wird, dann werden einige Ideen zur Behandlung des Problems vorgestellt.

6.7.1 Ausgangslage

Gegeben ist eine Folge von Teilesets in Form einer Teilesetliste $TSL = \{TS_1, \dots, TS_n\}$. Für jedes dieser Teilesets TS_j existiert ein zulässiges Layout. Es werden alternative Layouts für die einzelnen TS_j gesucht, um den Handlungsaufwand über die ganze Folge von Teilesets zu minimieren, ohne ihre Reihenfolge in TSL zu ändern.

Motivation

Die Vermutung, dass eine vorliegende Layoutfolge bezüglich ihrem Handlingsaufwand verbessert werden kann, stützt sich auf folgende Punkte:

- Bei der Layoutgenerierung steht die Suche nach zulässigen Lösungen im Vordergrund. Die in Kapitel 5 beschriebene Zielfunktion für ein Teileset TS_j berücksichtigt v.a. den $SetValue(TS_j)$ und somit die Priorität sowie die Gesamtfläche der in TS_j enthaltenen Aufträge. Der Handlingsaufwand wurde eigentlich nur durch den $MoveValue$ grob geschätzt. Das macht in diesem Moment durchaus Sinn: Durch eine zu starke Gewichtung des Handlingsaufwandes würden u.U. Layouts mit beispielsweise 60% Auslastung solchen mit 90% vorgezogen (Wer je selber eine Zielfunktion mit so vielen Parametern ‚justieren‘ musste, kann dieses Problem wohl sehr gut nachvollziehen)
- Die Layouts werden sequentiell generiert. Dabei wird der Handlingsaufwand nur bezüglich des direkt vorhergehenden Layouts - also lokal - optimiert. Auch diese Strategie ist während der Layoutgenerierung sinnvoll: Da die Planung wenige Stunden vor der Umsetzung erfolgt, müssen die Vorschläge relativ schnell generiert sein. Insbesondere müssen aufgrund der vorgeschlagenen Teilesets die Matrizen der neu hinzugekommenen Aufträge herbeigeschafft werden
- Bei einer gegebenen zulässigen Teilung wird ein festes Layout ohne Berücksichtigung des vorhergehenden Layouts bestimmt

Aus dem zweiten Punkt kann abgeleitet werden, in welchem Zeitintervall eine Layoutoptimierung durchgeführt werden kann: Sie wird gestartet, wenn die vorgeschlagene TSL akzeptiert wurde, und muss beendet sein, wenn mit der Abarbeitung der darin enthaltenen Teilesets begonnen wird. Man beachte, dass für die Layoutoptimierung deutlich mehr Zeit zur Verfügung steht (Stunden), als für die Layoutgenerierung (Minuten).

Zielfunktion für die Optimierung einer Layoutfolge

Um den Handlingsaufwand zu bestimmen, der aus einer Folge von Layouts L_1, \dots, L_m resultiert, muss man zuerst eine Funktion $HandlingValue(L_i, L_{i+1})$ definieren, die für zwei sich folgende Layouts L_i, L_{i+1} die Arbeit der Pressenbediener quantifiziert. Zwei natürliche Erwartungen an diese Funktion sind:

- $HandlingValue(L_i, L_i) = 0$
- $HandlingValue(L_i, L_{i+1}) + HandlingValue(L_{i+1}, L_{i+2}) \geq HandlingValue(L_i, L_{i+2})$

Die Funktion muss also erstens die notwendigen Manipulationen an den Matrizen abschätzen können und zweitens diese Manipulationen in Aufwand (Zeitbedarf) transformieren.

Daraus leitet sich eine erste Erwartung an den Modul ‚Layoutoptimierung‘ ab: Gegeben sind zwei sich folgende Layouts L_i, L_{i+1} sowie m Matrizen, welche in beiden Layouts vorkommen. Gesucht ist eine Reihenfolge, in der diese Matrizen verschoben werden müssen, um den dazu notwendigen Aufwand minimal zu halten. Eine erste Aufgabe des Moduls ‚Layoutoptimierung‘ besteht darin, diese Reihenfolge dem Bediener vorgeschlagen. Zwei Arten von Aufwänden müssen dafür berücksichtigt werden. :

- Anzahl Matrizen, welche von den Pressenbedienern in die Hände genommen werden müssen. Da alle zu bewegenden Matrizen beim Umstellen von L_i zu L_{i+1} mindestens einmal berührt werden, ist eine Verschiebungsreihenfolge gesucht, bei der möglichst wenige Matrizen mehrere Male bewegt werden müssen

- Im Weiteren soll durch eine geschickte Reihenfolge vermieden werden, dass eine Matrize über eine andere gehoben werden muss, um an ihren neuen Platz gelegt zu werden

Für eine Folge $TSL = \{TS_1, \dots, TS_n\}$ sowie einer Layoutliste $LS = \{L_1, \dots, L_n\}$, in welcher die entsprechenden, zulässigen Layouts eingetragen sind, wird folgende Zielfunktion definiert:

$$\text{HandlingValue}(TSL) = \sum_{i=1}^{n-1} \text{HandlingValue}(L_i, L_{i+1}) \quad [6-37]$$

6.7.2 Ansätze zur Optimierung einer Layoutfolge

Ansatz 1: Optimierung einer Layoutfolge ohne Änderung der k-Teilungen

Ziel dieses Ansatzes ist es, ohne Änderungen der k-Teilungen Matrizen so zu legen, dass sie möglichst selten und möglichst wenig verschoben werden müssen.

Gegeben sei eine k-Teilung für TS_j . Üblicherweise wird dann für die feste Position einer Matrize die Mitte ihrer freien Zone ermittelt, indem Gleichung [6-18] angewendet wird:

$$\begin{aligned} X_j &= (\text{Min}X_j + \text{Max}X_j)/2 \\ Y_j &= (\text{Min}Y_j + \text{Max}Y_j)/2 \end{aligned} \quad [6-38]$$

Es stellt sich die Frage, ob über eine gesamte Folge von Layouts für jede Matrize M_j dasselbe Koordinatenpaar X_j, Y_j behalten werden kann. Das heisst: Gesucht wird eine Folge von Layouts, bei denen die auf dem Trog verbleibenden Matrizen nicht verschoben werden müssen. Diese Layoutfolge existiert genau dann, falls die Ungleichungssysteme [6-5] für die Folge von k-Teilungen für TS_1, \dots, TS_n in einem einzigen System zusammengestellt werden und falls dieses System eine zulässige Lösung besitzt. Es wird vermutet, dass dies nicht der Fall ist für die Layoutfolge, die generiert wurde (sonst hätte man eine 3D Packing-Lösung). Dann sollte man für einige wenige Matrizen mehr als ein Koordinatenpaar (X_j, Y_j) zulassen (z.B. für die Matrize M_j die Paare (X_j^1, Y_j^1) für die Layouts L_1, \dots, L_r und (X_j^2, Y_j^2) für die Layouts L_{r+1}, \dots, L_n). Ein möglicher Ansatz, um zu bestimmen, wann neue Koordinaten eingeführt werden müssen, wäre folgender: Finde eine maximale Anzahl konsekutiver Layouts, deren zusammengestelltes Ungleichungssystem zulässig ist.

Ansatz 2: Optimierung der Layoutfolge mit Änderung der k-Teilungen

Für jedes TS_j werden m Layouts L_j^1, \dots, L_j^m generiert. Nun kann man einen gewichteten, gerichteten Graphen $G(V, E)$ erzeugen mit

$$\begin{aligned} V &= \{L_j^r; j = 1, \dots, n, ; r = 1, \dots, m\} \cup \{Q, S\} \\ E &= \{(Q, L_j^r), (L_j^r, L_{j+1}^s), (L_n^s, S); j = 1, \dots, n-1; r, s = 1, \dots, m\} \end{aligned} \quad [6-39]$$

Für jedes der $m \times n$ Layouts wird also ein Knoten erzeugt, zusätzlich existiert eine Quelle Q und eine Senke S . Die Länge der Kanten (L_j^r, L_{j+1}^s) werde definiert als $\text{HandlingValue}(L_j^r, L_{j+1}^s)$. Q ist mit allen m Layouts des TS_1 verbunden, alle Bogen $\langle Q, L_1^r \rangle$ ($r=1, \dots, m$) haben die Kanten-

länge Null. Weiter sind alle m Layouts des TS_n mit der Senke S verbunden, alle Bogen $\langle L_n^i, S \rangle$ haben die Kantenlänge Null. Abbildung 6-37 zeigt den oben beschriebenen Graphen G .

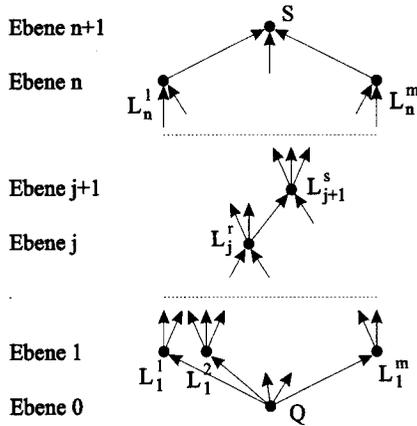


Abbildung 6-37: Graph G mit den Layouts der Teilesets TS_j

Die Layoutfolge mit dem kleinsten Handlingsaufwand kann nun ermittelt werden, indem im Graphen G der kürzeste Weg von Q nach S gesucht wird. Die Bestimmung kürzester Wege in einem endlichen, schlichten Digraphen kann aus Abschnitt 6.2.3 abgeleitet werden, wo gezeigt wurde, wie für solche Graphen die längsten Wege berechnet werden.

7 Tests

Dieses Kapitel beschreibt die Testreihen, mit welchen die implementierten Module von FluidPlan getestet wurden, und den Datengenerator, welcher dabei benutzt wurde.

7.1 Simulation der rollenden Planung

Um das Verhalten beider Strategien (direktes und vorausschauendes Plazieren) zu vergleichen, wurde entschieden, die rollende Planung für eine Folge von ungefähr 500 Pressungen zu simulieren, was der Planung einer Layoutfolge von ca. einer Woche entspricht.

Diese Aufgabe wurde mit 5 verschiedenen Datensätzen, die unabhängig voneinander generiert wurden, durchgeführt. Der verwendete Datengenerator ist in Abschnitt 7.3 beschrieben. In den unten gezeigten Statistiken sind die Ergebnisse aller Testläufe pro Strategie zusammengefasst.

7.1.1 Vorgehen

Für die gesamte Testreihe wurde also wie folgt vorgegangen:

- Generierung 5 verschiedener Datensätze, bestehend aus jeweils 400 zufällig zusammengestellten Aufträgen
- Pro Datensatz wurde je ein Testlauf mit der Strategie des direkten Plazierens und mit der Strategie des vorausschauenden Plazierens durchgeführt
- Pro Testlauf wurden mindestens 500 Pressungen generiert (exakterweise $500+t$ Pressungen, falls die 500. Pressung zusammen mit t weiteren Pressungen generiert wurde)
- Die Pressungen wurden in 10 Serien zu je 50 Pressungen geplant, wobei pro Serie nie mehr als 100 Aufträge zur Einplanung freigegeben waren. Eine Serie entspricht den Pressungen für ca. einen halben Arbeitstag

Ein Testlauf wurde in folgenden Schritten durchgeführt:

1. Einlesen des Datensatzes. Beim Start der Simulation wurden 100 der 400 Aufträge zur Einplanung freigegeben (Zähler n für die Serien initialisieren: $n=1$)
2. Generierung von soviel Layouts, bis mindestens n mal 50 Pressungen vorliegen. Ausschlaggeben ist dabei die letzte Pressung des zuletzt generierten Layouts, ohne Einbezug der davon abgeleiteten Layouts. Für die einzelnen Strategien heisst das:
 - Direktes Plazieren (Iteration):
 - Layoutvorschlag generieren (und direkt akzeptieren)
 - Zur ersten Pressung des ersten davon abgeleiteten Layouts wechseln
 - Falls dessen Pressungsnummer kleiner gleich $n*50$: Iteration wiederholen
 - Vorausschauendes Plazieren: Der Ablauf dieser Strategie ist in Abschnitt 5.3.3 detailliert beschrieben (mit $p_{\text{soll}} = n*50$)
3. $n < 10$: Von den provisorisch gestoppten Aufträgen sovieler freigeben, dass wieder 100 Aufträge eingeplant werden können. $n = n+1$. Zurück zu Schritt 2

Verwendete Parameter in FluidPlan: Timeout = 1 Minute, Mindestauslastung = 80%.

7.1.2 Auswertungen

Für die Auswertungen wurden bei den mit demselben Datensatz durchgeführten Testläufen dieselbe Anzahl Pressungen (500+t) berücksichtigt. Dafür wurden für beide Strategien die Nummer der letzten generierten Pressung eruiert und die tiefere davon gewählt.

In den folgenden Tabellen wird für die Strategie des direkten Plazierens die Bezeichnung *dir* verwendet, entsprechend für das vorausschauende Plazieren die Bezeichnung *vor*.

Auslastungen

In Tabelle 7-1 sind die wichtigsten Kennzahlen zu den erzielten Auslastungen der Testläufe zusammengefasst. Die fünf Werte derselben Strategie sind dafür gemittelt worden.

| Strategie | Auslastung | Standardabw. | Maximalwert | Minimalwert |
|-----------|------------|--------------|-------------|-------------|
| direkt | 87.3% | 3.6% | 93.9% | 72.4% |
| voraus | 83.7% | 10.0% | 95.2% | 22.5% |
| Differenz | 3.6% | 6.4% | 1.3% | 49.9% |

Tabelle 7-1: Zusammenfassung der Auslastungsdaten

Bei einer durchschnittlich geforderten Minimalauslastung von 80% liefern beide Strategien Resultate, die den Anforderungen genügen. Die um 3.6% tiefere Auslastung beim vorausschauenden Plazieren werden aufgrund der untenstehenden weiteren Kennzahlen in Kauf genommen, ebenso die implizite Information, dass mit dieser Strategie insgesamt weniger Aufträge gefertigt wurden (aber mit höherer durchschnittlicher Priorität!).

Der tiefe Minimalwert der Auslastung beim vorausschauenden Plazieren sowie die hohe Standardabweichung ergeben sich durch folgende Situation: Die Matrize des zu plazierenden Auftrags ist derart gross, dass nur wenige andere Matrizen daneben gelegt werden können. Der Trog muss also praktisch leer sein für die Plazierung. Beim Füllen der entstandenen Lücke sinkt dann die Trogauslastung u.U. kurzzeitig stark, da relativ wenige Aufträge kleine Losgrößen (unter 10) haben. In der Praxis wird diese Situation von den Bedienern umgangen, beispielsweise durch den kurzzeitigen Einsatz des zweiten Troges.

Wartezeiten

Im Zusammenhang mit der Auslastung ist auch interessant, wie lange die Aufträge zur Einplanung freigegeben waren, bis sie tatsächlich eingeplant wurden. Diese Frist heisst *Wartezeit* und wurde in Anzahl Pressungen gemessen. Tabelle 7-2 zeigt eine Übersicht der durchschnittlichen Wartezeiten, aufgeteilt nach Prioritätsklassen und nach der eingesetzten Strategie. Im Weiteren sind die Standardabweichungen und die Maximalwerte der Wartezeiten angegeben, ebenso die betreffende Anzahl Aufträge.

Bemerkung: Ein Auftrag eines Testdatensatzes wird bei Anwendung unterschiedlicher Strategien nicht zwingend für dieselbe Serie freigegeben. Ursache ist der Algorithmus, welcher am Schluss einer Serie jeweils neue Aufträge freigibt. Dieser Effekt entspricht der Realität, denn es wird aufgrund der Situation an der Presse entschieden, wieviele Aufträge bei der Wärmebehandlung berücksichtigt werden bzw. wieviele Chargen zusammengestellt werden. Deshalb sind die Anzahl Aufträge bei den Strategien unterschiedlich.

Im Kopf der Tabelle 7-2 werden folgende Abkürzungen verwendet: Anzahl = Anzahl Aufträge, Schnitt = Durchschnittliche Wartezeit, Stdabw. = Standardabweichung der Wartezeit, Max. = Maximalwert der Wartezeit.

| Pri-Klasse | Anzahl direkt | Anzahl voraus | Schnitt direkt | Schnitt voraus | Stdabw. direkt | Stdabw. voraus | Max. direkt | Max. voraus |
|------------|---------------|---------------|----------------|----------------|----------------|----------------|-------------|-------------|
| PriB | 45 | 39 | 13.0 | 5.2 | 41.1 | 7.5 | 261 | 30 |
| PriC | 52 | 50 | 22.0 | 11.7 | 48.3 | 16.1 | 207 | 72 |
| Pri9 | 14 | 13 | 32.8 | 15.0 | 55.8 | 11.4 | 181 | 30 |
| Pri8 | 18 | 16 | 27.7 | 16.4 | 52.7 | 27.2 | 191 | 111 |
| Pri7 | 24 | 21 | 11.3 | 6.9 | 29.9 | 9.2 | 149 | 25 |
| Pri6 | 60 | 52 | 24.5 | 18.3 | 67.6 | 14.2 | 490 | 48 |
| Pri5 | 60 | 59 | 26.9 | 31.0 | 55.4 | 39.1 | 331 | 198 |
| Pri4 | 74 | 78 | 31.8 | 35.2 | 61.4 | 49.1 | 387 | 288 |
| Pri3 | 110 | 118 | 36.1 | 44.5 | 60.9 | 56.4 | 348 | 339 |
| Pri2 | 170 | 162 | 43.8 | 76.4 | 69.0 | 78.8 | 480 | 398 |
| Pri1 | 294 | 249 | 66.1 | 103.0 | 88.3 | 113.6 | 478 | 525 |
| Pri0 | 407 | 256 | 101.4 | 140.6 | 112.8 | 131.0 | 499 | 573 |

Tabelle 7-2: Verteilung und durchschnittliche Wartezeiten der eingeplanten Aufträge

In Abbildung 7-1 sind die durchschnittlichen Wartezeiten der sechs höheren Prioritätsklassen, inklusive Standardabweichung, eingezeichnet.

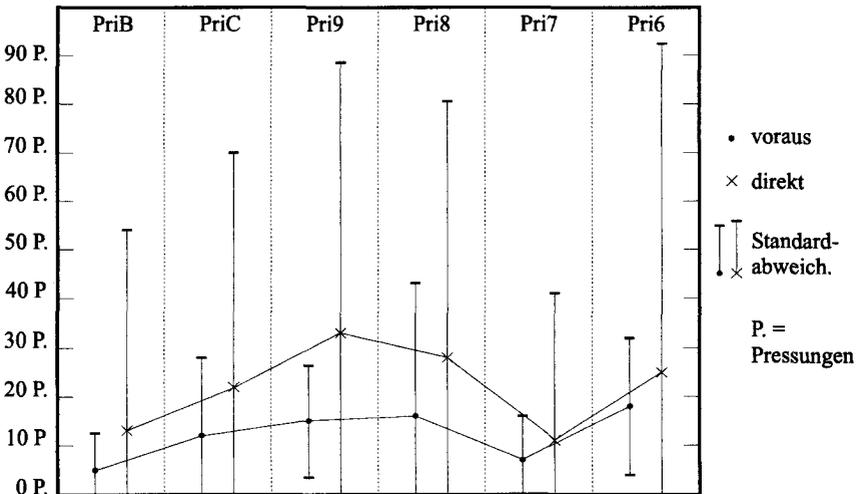


Abbildung 7-1: Durchschnittliche Wartezeit von Aufträgen der oberen Prioritätsklassen

Man stellt fest, dass beim direkten Plazieren die Wartezeiten der oberen Prioritätsklassen durchschnittlich etwa doppelt so hoch sind wie beim Vorausschauenden. Insbesondere sind die Maximalwerte viel höher.

Tabelle 7-3 zeigt in Ergänzung zu den oben aufgeführten Daten die durchschnittlichen Wartezeiten (bis 500+) derjenigen Aufträge, die im Verlaufe eines Testlaufes freigegeben, aber nie eingeplant wurden.

| Pri-Klasse | Anzahl direkt | Anzahl voraus | Schnitt direkt | Schnitt voraus | Stdabw. direkt | Stdabw. voraus | Max. direkt | Max. voraus |
|------------|---------------|---------------|----------------|----------------|----------------|----------------|-------------|-------------|
| PriB | 7 | - | 192.9 | - | 135.8 | - | 398 | - |
| PriC | 6 | - | 313.0 | - | 151.0 | - | 443 | - |
| Pri9 | 3 | - | 383.0 | - | 160.5 | - | 500 | - |
| Pri8 | 1 | - | 348.0 | - | - | - | 348 | - |
| Pri7 | 1 | - | 247.0 | - | - | - | 247 | - |
| Pri6 | 6 | - | 141.2 | - | 79.3 | - | 247 | - |
| Pri5 | 6 | - | 281.2 | - | 145.8 | - | 448 | - |
| Pri4 | 11 | 1 | 352.6 | 45.0 | 131.3 | - | 500 | |
| Pri3 | 18 | 4 | 348.5 | 316.3 | 142.8 | 240.5 | 500 | 45 |
| Pri2 | 35 | 10 | 331.5 | 172.3 | 154.7 | 184.6 | 500 | 514 |
| Pri1 | 72 | 76 | 338.9 | 307.8 | 139.7 | 188.4 | 500 | 514 |
| Pri0 | 222 | 287 | 333.5 | 328.0 | 148.2 | 162.5 | 500 | 514 |

Tabelle 7-3: Verteilung und durchschnittliche Wartezeiten der nicht eingeplanten Aufträge

Es wird deutlich, dass beim direkten Plazieren Aufträge der höchsten Prioritätsklassen u.U. nie plaziert werden. Diese Aufträge haben allesamt Matrizen mit grosser Grundfläche.

Anzahl Layouts

Weiter wurde die Anzahl generierter Layouts ausgewertet, die Ergebnisse sind in Tabelle 7-4 auszugsweise wiedergegeben. Die Minimal- und Maximalwerte sind ebenfalls angegeben, um zu zeigen, dass die Abweichungen bei den einzelnen Testläufen pro Strategie relativ gering waren.

| Strategie | Extremwerte | Durchschnitt | Genormter Schnitt |
|------------|-------------|--------------|-------------------|
| direkt | 141/161 | 155.6 | 117.2 |
| voraus | 70/81 | 75.2 | 62.6 |
| Verhältnis | - | 2.07 | 1.73 |

Tabelle 7-4: Zusammenfassung der Anzahl Layouts

Der ‚genormte Schnitt‘ entspricht der Anzahl Layouts, die nötig sind, um 1000 Aufträge einzuplanen (werden die Anzahl Aufträge in Tabelle 7-2 aufsummiert, ergeben sich für das direkte Plazieren 1328 Aufträge gegenüber 1113 beim vorausschauenden Plazieren).

Handlingsaufwand

Die doppelte Anzahl benötigter Layouts beim direkten Plazieren widerspiegelt sich auch im Handlingsaufwand, welcher proportional zur Anzahl notwendiger Verschiebungen angenommen wird. Diese sind in Tabelle 7-5 für sämtliche Testläufe einer Strategie aufsummiert. Die Werte wurden in Gruppen aufgeteilt, wobei man davon ausgehen kann, dass die Verschiebungen der ersten Gruppe nicht durchgeführt werden, da sie klein sind: Eine Matrize muss von den Bedienern nicht millimetergenau positioniert werden, insbesondere ist ihre Lage innerhalb der Umhüllenden nicht exakt definiert. Die Grenze zur Unterteilung der letzten beiden Gruppen wurde gleich der halben Troglänge gesetzt.

| Gruppe | Verschiebung v in mm | direkt | voraus | dir/vor |
|--------|------------------------|--------|--------|---------|
| 1 | $v \leq 50$ | 3346 | 1030 | 3.25 |
| 2 | $50 < v \leq 150$ | 854 | 345 | 2.48 |
| 3 | $150 < v \leq 300$ | 1181 | 451 | 2.62 |
| 4 | $300 < v \leq 800$ | 2081 | 768 | 2.71 |
| 5 | $800 < v \leq 1300$ | 886 | 292 | 3.03 |
| 6 | $1300 < v$ | 543 | 211 | 2.57 |
| | Summe (ohne Gruppe 1) | 5545 | 2067 | 2.68 |

Tabelle 7-5: Verteilung der notwendigen Verschiebungen

Auch ohne Einbezug der ersten Gruppe spricht das Verhältnis zwischen den Strategien deutlich gegen das direkte Plazieren. Pro Gruppe sind 2.5 bis 3 mal mehr Verschiebungen durchzuführen. Das ‚genormte‘ Verhältnis der Verschiebungen (wiederum bezogen auf 1000 Aufträge) beträgt 2.25:1.

Aufträge pro Layout

Schliesslich wurde auch die Anzahl Aufträge, die sich durchschnittlich pro Layout auf dem Trog befanden, berechnet. Diese Zahl ist identisch mit der Anzahl Matrizen, welche pro Pressung auf dem Trog waren. Da pro Testlauf die durchschnittliche Auslastung bekannt ist, kann bestimmt werden, wie gross im Schnitt die Umhüllenden der plazierten Matrizen waren. Die Resultate dieser Auswertung sind in Tabelle 7-6 zu sehen, wobei die Grösse der Umhüllenden in Prozent der Trogfläche angegeben sind.

| Strategie | Extremwerte der Anzahl Auftr. | Durchschnitt der Anzahl Auftr. | Durchschnittl. Auslastung | Durchschnittl. Matrizenfläche |
|-----------|-------------------------------|--------------------------------|---------------------------|-------------------------------|
| direkt | 14.2/15.9 | 15.2 | 87.3% | 5.74% |
| voraus | 11.1/12.3 | 11.7 | 83.7% | 7.17% |

Tabelle 7-6: Anzahl und Grösse der Aufträge pro Layout

Wiederum sind die minimalen und maximalen Werte der durchschnittlichen Anzahl Aufträge pro Layout angegeben. Es zeigt, dass die Unterschiede innerhalb einer Strategie deutlich kleiner sind als die Unterschiede zwischen den beiden Strategien.

Die grössere Anzahl Aufträge mit kleinen Matrizen beim direkten Plazieren deckt sich mit den Beobachtungen, welche bereits in Abschnitt 5.3 gezeigt wurden: Offensichtlich werden frei ge-

wordene Plätze zu Beginn eines Testlaufes tendenziell mit mehr Matrizen wiederaufgefüllt als beim vorausschauenden Plazieren (in beiden Fällen pendelt sich der Schnitt nach einigen Dutzend Pressungen ein). Damit kann erklärt werden, weshalb beim direkten Plazieren Aufträge mit grossen Matrizen trotz hoher Priorität u.U. gar nicht eingeplant werden (vgl. Tabelle 7-3).

Aufgrund der Auswertungen der durchgeführten Simulation wird das vorausschauende Plazieren als die bessere Strategie erachtet. Der Nachteil der etwas geringeren Auslastung wird mehr als wettgemacht durch die deutlich kleineren Wartezeiten und durch den nicht einmal halb so grossen resultierenden Handlungsaufwand.

Verhalten beim leeren Trog

Den Abschluss dieses Abschnittes bildet Tabelle 7-7, in welcher aufgelistet ist, aus welchen Prioritätsklassen die Aufträge der ersten Pressung eines Testlaufes jeweils gewählt wurden. Beide Strategien bestimmen für diese Pressung dasselbe Teileset. Die Zusammenstellung zeigt die Verteilung der bereitliegenden Aufträge auf die Prioritätsklassen (A. total), die gewählten Aufträge (A. gewählt) sowie die Differenz dieser Werte. Die Auslastung lag dabei durchschnittlich bei 90.2%.

| Pri-Klasse | B | C | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|----|----|---|---|---|----|----|----|----|----|-----|-----|
| A. total | 19 | 29 | 6 | 4 | 8 | 16 | 18 | 23 | 40 | 67 | 100 | 167 |
| A. gewählt | 19 | 29 | 4 | 1 | 6 | 4 | 2 | 2 | 4 | 2 | 0 | 0 |
| Differenz | 0 | 0 | 2 | 3 | 2 | 12 | 16 | 21 | 36 | 65 | 100 | 167 |

Tabelle 7-7: Im ersten Layout gewählte Aufträge

Das heisst, dass die entwickelten Methoden der Teilesetwahl sehr gut arbeiten.

7.2 Test des Kernmoduls

Der Modul Layoutgenerierung wird als Kernmodul betrachtet und wurde hinsichtlich zweier Faktoren untersucht. Erstens interessierte der Zeitbedarf des Algorithmus, zweitens wurde untersucht, ob überhaupt Mehrzonenplätze geprüft und v.a. gewählt wurden.

7.2.1 Zeitbedarf und Komplexität

Für die Messungen wurden mit dem Datengenerator (vgl. Abschnitt 7.3) verschiedene Datensätze generiert, wobei dieselben Einstellungen verwendet wurden wie für die Simulation der rollenden Planung. Ein Testlauf wurde wie folgt gemessen:

- Startpunkt der Zeitmessung war der Moment, in welchem der Modul ‚Layoutgenerierung‘ einen Teilesetkandidaten TSK erhielt
- Um Schwankungen bei der Zeitdauer zu glätten, generierte der Modul die Layouts 10 mal hintereinander, ohne das Resultat jeweils an den Modul ‚Teilesetwahl‘ zurückzugeben
- Endpunkt der Zeitmessung war der Moment, in welchem der Modul ‚Layoutgenerierung‘ ein zulässiges Layout an den Modul ‚Teilesetkandidat‘ zurückgab

Um auch TSK mit über 50 Matrizen zu schachteln, wurde die Grösse des Troges mehrmals heraufgesetzt. Auf diese Weise erhielt man breiter abgestützte Daten für die Analyse.

Die erhaltenen Daten wurden mit Hilfe eines Tabellenkalkulationsprogrammes als Kurve dargestellt und es wurde automatisch eine Trendlinie mit potentieller Funktion generiert. Abbildung 7-2 zeigt die Kurve (dicke Linie) inklusive Trendlinie (dünn). Die erhaltenen Werte sind durch 10 geteilt, zeigen also den Zeitbedarf einer einzigen Layoutgenerierung bei der entsprechenden Anzahl Matrizen.

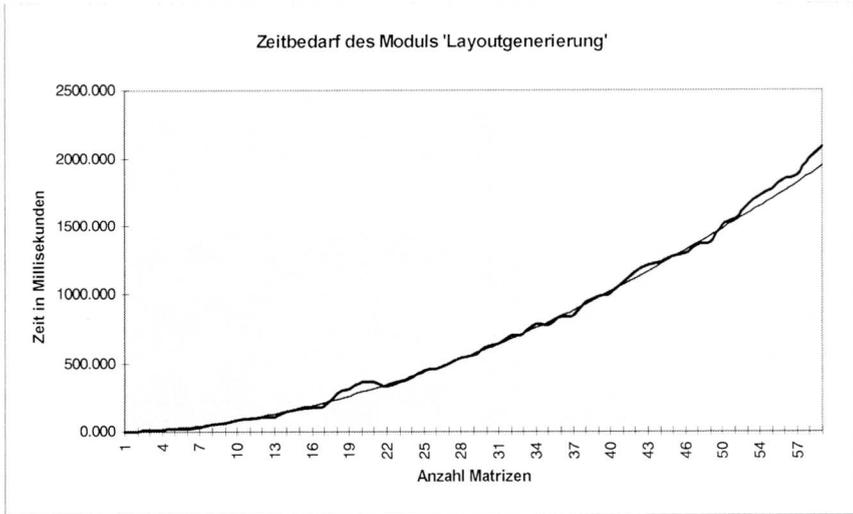


Abbildung 7-2: Zeitbedarf der Schachtelung inklusive Trendlinie

Abbildung 7-8 zeigt einige ausgewählte Werte der obenstehenden Kurve. Die Tests wurden auf einem PC mit einem mit 90 Megahertz getakteten Pentium-Prozessor durchgeführt.

| | | | | | | | | | |
|-----------------|-----|-----|------|------|------|------|------|------|------|
| Anzahl Matrizen | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Zeit in ms | 4.7 | 8.5 | 13.6 | 18.8 | 24.9 | 35.1 | 50.2 | 69.0 | 79.1 |
| Anzahl Matrizen | 12 | 14 | 16 | 18 | 20 | 25 | 30 | 35 | 40 |
| Zeit in ms | 105 | 140 | 163 | 278 | 357 | 440 | 615 | 785 | 983 |

Tabelle 7-8: Zeitbedarf der Layoutgenerierung in Abhängigkeit der Anzahl Matrizen

Die potentielle Funktion, auf welcher die automatisch generierte Trendlinie aufbaut, stützt folgende Aussage bezüglich der *Komplexität* der Layoutgenerierung:

Bei Einzonenplätzen wächst der Aufwand zur Layoutgenerierung quadratisch mit der Anzahl zu platzierender Matrizen.

Um die Matrize M_{k+1} zu platzieren, werden die freien Zonen der k bereits auf dem Trog liegenden Matrizen M_1, \dots, M_k geprüft. Um n Matrizen zu positionieren, müssen also insgesamt $1+2+3+\dots+(n-1) \approx n^2/2$ freie Zonen geprüft werden.

Die Aussage gilt nicht für Mehrzonenplätze. Da aber Mehrzonenplatzierungen weniger als einen Sechstel der gesamten Untersuchungen ausmachen (vgl. den folgenden Abschnitt), kann man erwarten, dass der Aufwand auch damit quadratisch bleibt. Diese Erwartung wird durch Abbildung 7-2 untertützt.

7.2.2 Gebrauch von Mehrzonenplätzen

Bei der Simulation der rollenden Planung wurde auch gemessen, wie oft bei der Platzsuche Ein- und Mehrzonenplätze gefunden wurden. Weiter wurde registriert, welcher Typ schliesslich beim Plazieren gewählt wurde. Abbildung 7-9 zeigt die über alle Testläufe aufsummierten Daten, wobei wie in Abschnitt 7.1 eine Trennung nach Strategien vorgenommen wurde. Diese Auftrennung zeigt, dass die Verhältnisse zwischen Ein- und Mehrzonenplätzen unabhängig von der Strategie sind. Die verwendeten Abkürzungen bedeuten: EZ = Einzonenplatz; MZ = Mehrzonenplatz; b = bewertet (also zulässig); a = ausgewählt; Verh = Verhältnis.

| Strategie | EZ-b | MZ-b | Verh | EZ-a | MZ-a | Verh |
|-----------|-----------|---------|------|---------|--------|-------|
| direkt | 1'913'741 | 374'066 | 5.12 | 488'902 | 45'850 | 10.66 |
| voraus | 688'966 | 133'700 | 5.15 | 206'843 | 19'333 | 10.70 |
| Summe | 2'602'707 | 507'766 | 5.13 | 695'745 | 65'183 | 10.67 |

Tabelle 7-9: Verhältnis von gefundenen und gewählten Ein- und Mehrzonenplätzen

Jede elfte Matrize wird also in einen Mehrzonenplatz gelegt, was zeigt, dass diese Methode für die Platzsuche sinnvoll ist. Um zu schätzen, wieviel Handlungsaufwand dadurch vermieden wird, müssten zuviele unsichere Annahmen getroffen werden, weshalb darauf verzichtet wird.

7.2.3 Weitere Aussagen zum Kernalgorithmus

Hohe Matrizen

Die beiden Regeln zur Behandlung von hohen Matrizen (nicht am Rand, nicht neben einer weiteren hohen Matrize) werden in vielen Fällen verletzt. Der Grund liegt in der Tatsache, dass hohe Matrizen praktisch immer auch eine grosse Grundfläche haben; oft füllen hohe Matrizen den Trog in seiner ganzen Breite. Die Regeln können dann gar nicht eingehalten werden. In Abbildung A-5 des Anhangs sind beispielsweise zwei derart grosse Matrizen abgebildet, die nicht plaziert werden können, ohne die Regeln zu verletzen (unter der Annahme, dass es sich um hohe Matrizen handelt).

Beobachtungen beim Einsatz von FluidPlan zeigten, dass die Regeln nicht verletzt werden, sofern die Matrizengrösse dies zulassen. Tests mit Matrizen, deren Grundflächen so klein sind, dass die Regeln i.A. eingehalten werden können, haben für die Fluidzellpresse keine praktische Relevanz und wurden deshalb nicht durchgeführt.

Tabuzonen

Die vorliegende Version von FluidPlan behandelt Tabuzonen, allerdings nicht gemäss dem in dieser Arbeit vorgestellten Konzept. Deshalb wurden keine systematischen Tests bezüglich Tabuzonen durchgeführt. Das präsentierte Konzept wurde nicht implementiert, weil die Anforderung zur Behandlung derartiger Zonen erst sehr spät definiert wurde.

In Abbildung 7-3 sind zwei verschiedene Vorschläge für dasselbe Layout zu sehen, deren Teileset aus derselben Auftragsmenge zusammengestellt wurde. Einer dieser Matrizen wurde für den zweiten Vorschlag eine Tabuzone der Variante V5 zugewiesen, sie soll also nicht mehr im mittleren Bereich des Troges plaziert werden. Entsprechend wurde die Matrize im zweiten Layout an den Rand gelegt. In der Abbildung zeigt jeweils ein Pfeil auf die gewählte Matrize.

Die implementierten Methoden zur Behandlung von Tabuzonen wurden mit Tests dieser Art durchgeführt. Es wurde auch beobachtet, wo solche Matrizen plazierte wurden, wenn sie in einer vorhergehenden Pressung bereits auf dem Trog lagen. Eine Verletzung der in Kapitel 6 formulierten Regeln wurde nie festgestellt.

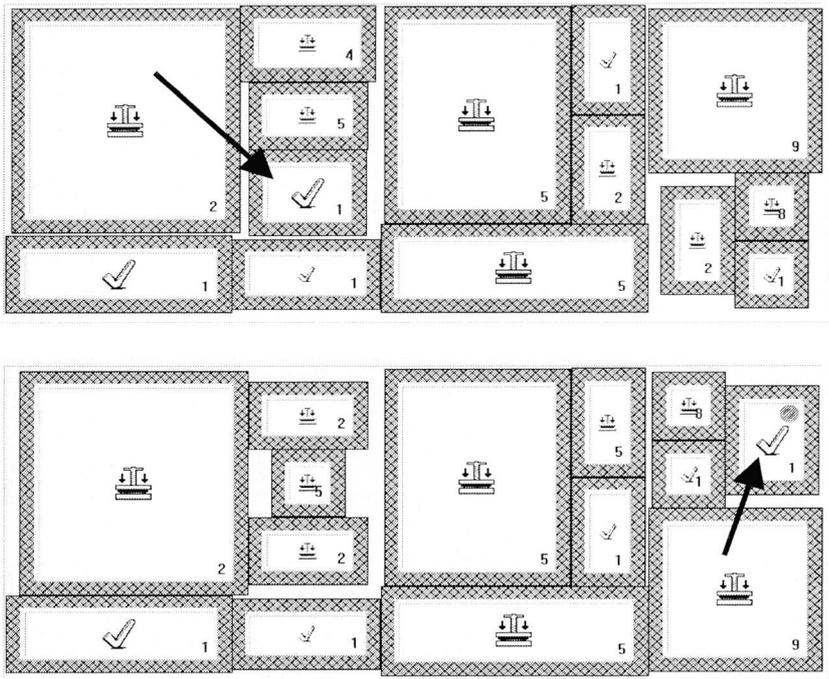


Abbildung 7-3: Platzierung derselben Matrize ohne und mit Tabuzone V5

Im Weiteren liegen zur Zeit keine zuverlässigen Zahlen vor, wieviele Matrizen tatsächlich Tabuzonen besitzen und wie sich diese Zonen auf die in Abschnitt 3.3.2 vorgestellten Varianten verteilen.

7.3 Datengenerator

Mit dem in diesem Abschnitt vorgestellten Datengenerator wurden sämtliche in den Tests verwendeten Datensätze erzeugt. Zwei Aspekte des Generators werden in diesem Abschnitt beleuchtet:

1. Welche Daten werden generiert und woher stammen die Verteilungen, mit welchen diesen Daten Werte zugewiesen werden
2. Welche Rahmenbedingungen können selber gesetzt werden

Zufällig generierbare Daten

Pro Auftrag werden folgende Angaben zufällig und unabhängig voneinander generiert:

- Losgrösse
- Priorität
- Grundfläche der entsprechenden Matrize

Um eine repräsentative Verteilung der Losgrössen und der Prioritäten zu erhalten, wurden fünf zufällig gewählte Prioritätslisten des Arbeitsplatzes ‚Fluidzellpresse‘ analysiert. Die Verteilungen wurden anschliessend im Datengenerator übernommen. Haben beispielsweise 4% aller Aufträge dieser Listen die Priorität Pri5, so werden durchschnittlich 4% aller generierten Aufträge ebenfalls Pri5 zugewiesen.

Für die Grundfläche existiert bei der Partnerfirma eine Liste mit 66 Flächenintervallen. Der kleinste Intervall geht von 0 bis knapp 100 cm², die obere Schranke des grössten Intervalls umfasst die grösstmögliche Trogfläche. Jede Matrize ist einem dieser Intervalle zugeordnet. Die anteilmässige Verteilung aller Matrizen auf dieser Liste wurde für den Datengenerator übernommen.

Nach der zufälligen Zuweisung einer Fläche wird für eine Matrize ihr Seitenverhältnis wie folgt bestimmt: Gegeben ist eine (für alle Flächen gültige) minimale Seitenlänge, die nicht unterschritten werden darf. Aus dieser und der Grundfläche lässt sich die maximale Seitenlänge bestimmen, womit ein Intervall vorliegt, in welchem die Werte liegen müssen. Eine Seitenlänge wird gleichverteilt einem Wert innerhalb dieses Intervalls zugewiesen, woraus sich die zweite Länge ableiten lässt.

Allgemeine Einstellungen

Abbildung 7-4 zeigt den Datengenerator. Anhand dieses Bildes wird kurz darauf eingegangen, welche Werte einstellbar sind, bevor ein Datensatz generiert wird.

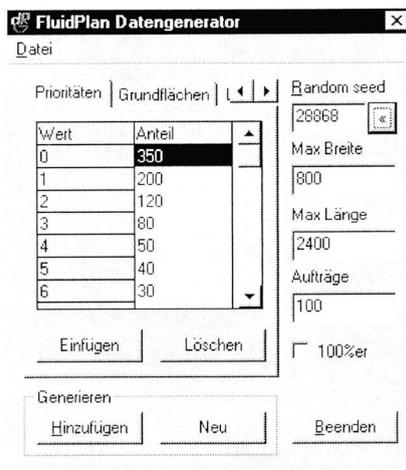


Abbildung 7-4: Printscreen des eingesetzten Datengenerators

Rechts oben kann der Startpunkt des Zufallsgenerators von Hand gesetzt werden (Random seed). Somit können einmal generierte Daten jederzeit reproduziert werden. Weiter können Maximalwerte für Länge und Breite einer Matrize eingegeben werden. Diese Werte fließen beim Bestimmen des Seitenverhältnisses einer Matrize ebenfalls als Grenzwerte ein. Im Weiteren muss angegeben werden, wieviele Aufträge generiert werden sollen. Zusätzlich kann bestimmt werden, ob die neuen Daten die bereits bestehenden ergänzen oder ersetzen sollen.

In der linken oberen Hälfte ist ein Teil der Tabelle zu sehen, welche den möglichen Prioritätswerten Anteile zuweist. Die oben beschriebenen geschätzten Werte aus der Praxis können also von Hand geändert werden. Die Häufigkeit, mit welcher ein Wert ausgewählt wird, ist proportional zum Verhältnis ihres Anteils an der Summe der Anteile aller Werte (die Gesamtsumme der abgebildeten Prioritätsanteile beträgt 1000, entsprechend werden 35% aller Aufträge Pri0 haben).

Die Einstellungen, mit welchen ein Testdatensatz generiert wurde, können gespeichert werden.

ANHANG

Gliederung des Anhangs

Im ersten Abschnitt des Anhangs wird die Sicht des Benutzers auf die implementierten Module des vorgestellten Konzepts beschrieben.

In den Abschnitten A2 bis A4 finden sich das Literaturverzeichnis sowie je eine Zusammenstellung der Abbildungen und Tabellen. Der Lebenslauf des Autors folgt im letzten Abschnitt.

A1 FluidPlan für den Benutzer

Implementiert wurde die erste Stufe des in Kapitel 4 vorgestellten Konzepts, mit Ausnahme des Moduls ‚Layoutoptimierung‘. Die Funktionsweise kann in Abschnitt 4.3.1 nachgelesen werden, in diesem Abschnitt werden deshalb lediglich die verschiedenen Ansichten des Moduls ‚Nestingfenster‘ gezeigt. In Abbildung A-1 ist skizziert, wie der Bildschirm aufgeteilt wird.

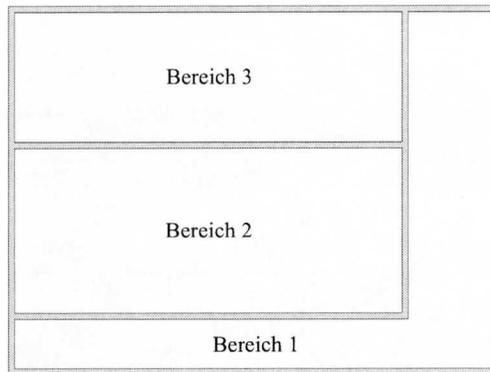


Abbildung A-1: Bildschirmaufbau von FluidPlan

In Bereich 1 sind die Bedientasten und Auswahlfelder untergebracht. Alle diese Bedienelemente wurden sofern möglich am rechten und am unteren Rand platziert, damit die Bedienung von FluidPlan beim Einsatz eines Touchscreen-Bildschirmes einfach und übersichtlich bleibt. Auf die einzelnen Elemente wird bei der Beschreibung der beiden anderen Bereiche eingegangen.

Beim Bereich 2 kann zwischen drei Fenstern hin- und hergeschaltet werden. Das erste Fenster, für welches in Abbildung A-2 ein Beispiel gegeben ist, zeigt eine Übersicht der anstehenden Aufträge, aufgeteilt nach Teilefamilien. Pro Teilefamilie wird eine Säule gezeichnet, deren Höhe proportional zur geschätzten Anzahl notwendiger Pressungen ist (vgl. Abschnitt 5.2.4). Die Zahl oben links in den Säulen entspricht dieser Schätzung. Der dunkel gefärbte Teil der Säulen zeigt an, wieviele Pressungen notwendig sind, wenn ausschliesslich Aufträge hoher Priorität gepresst werden. Welche Prioritätsklassen zu dieser Gruppe gehören, kann der Bedie-

ner via Eingabefenster (nicht abgebildet) selbst bestimmen. Durch Doppelklick auf eine Säule wird die entsprechende Teilefamilie detaillierter beschrieben, indem graphisch gezeigt wird, wieviele Aufträge pro Prioritätsklasse vorliegen (ebenfalls nicht abgebildet).

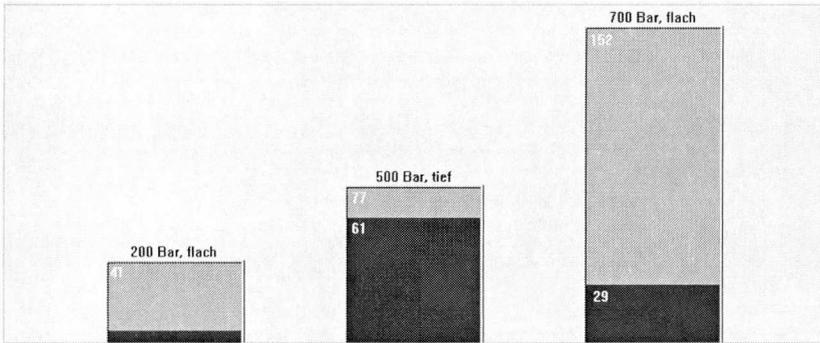


Abbildung A-2: Übersicht der Teilefamilien

Alternativ zu diesem Fenster kann im Bereich 2 einer der beiden Tröge der Fluidzellpresse eingeblendet werden. Dazu stehen dem Bediener die drei Schalter aus Abbildung A-3 zur Verfügung (die im Bereich 1 angeordnet sind).

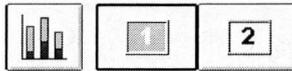


Abbildung A-3: Auswahl der Ansicht in Bereich 2

Bei einem leerem Trog kann der Bediener eine der bereitliegenden Teilefamilien auswählen und einen Nestingdurchgang (Teilesetwahl und Layoutgenerierung) starten. Akzeptiert der Bediener den Layoutvorschlag, liegt ein erstes generiertes Layout vor, wie beispielsweise in Abbildung A-4 gezeichnet.

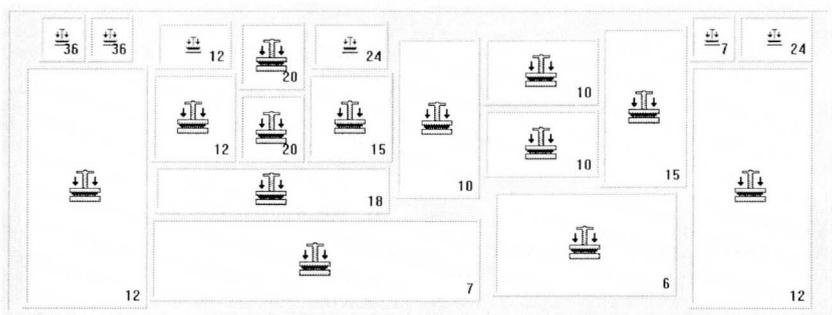


Abbildung A-4: Erstes akzeptiertes Troglayout einer Teilefamilie

Pro Matrize wird dabei die Umhüllende gezeichnet, zudem ist in der rechten unteren Ecke angegeben, wieviele Pressungen notwendig sind, bis der entsprechende Auftrag fertig bearbeitet ist. Das Pressensymbol zeigt überdies für jede Matrize an, dass sie bei der gezeigten Pressung zum ersten Mal im Trog liegt.

Waren beim Nestingaufruf dem betreffenden Layout bereits Matrizen zugewiesen, sieht ein generiertes Layout beispielsweise wie dasjenige in Abbildung A-5 aus. Man bemerkt, dass diese Matrizen dunkel eingefärbt sind.

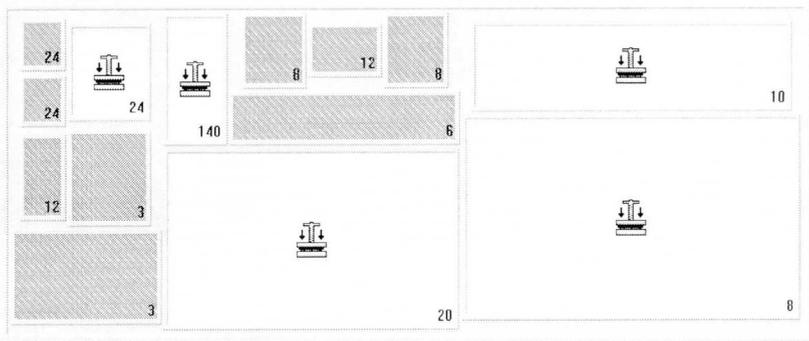


Abbildung A-5: Troglayout mit alten und neuen Matrizen

Das Layout aus Abbildung A-5 kann auf zwei verschiedenen Wegen entstanden sein:

1. Der Bediener hat selbst das Layout gewählt, welches ergänzt werden soll. Dazu wird ihm auf Knopfdruck im Bereich 3 des Bildschirms eine Auslastungsübersicht angezeigt, wie sie in Abbildung 5-9 zu sehen ist. Zudem steht ihm ein Schieber zur Verfügung, mit welchem er durch die einzelnen Layouts bzw. Pressungen „blättern“ kann. In Abbildung A-6 ist dieser Schieber gezeigt. Mit den einfachen Pfeilen rechts und links vom Schieberband kann durch die einzelnen Pressungen geblättert werden, mit den Doppelpfeilen hüpft der Schieber direkt zu einer Pressung, bei der eine Änderung durchgeführt werden muss (also zum Anfang oder Ende eines Layouts). Das auf diese Weise gewählte Layout wird ergänzt, wenn im Bereich 1 der Schalter ‚Neues Layout‘ gedrückt wird (vgl. Abbildung A-7)

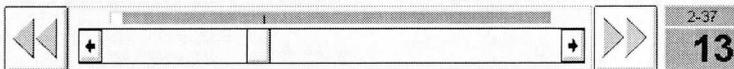


Abbildung A-6: Layoutschieber

2. Auf dem Bildschirm wird irgend ein Layout im betreffenden Trog angezeigt, und der Benutzer betätigt den ‚Vorschlag‘-Schalter im Bereich 1, welcher in untenstehender Abbildung am rechten Bildschirmrand zu sehen ist. Dadurch wird ein Nestingaufruf gemäss der Strategie des vorausschauenden Plazierens ausgelöst.

In Abbildung A-7 sieht man auch das ursprüngliche Layout, welches ergänzt werden soll (im eingblendeten Trog im zweiten Bereich). Zudem ist der Bereich 3 zu sehen. In dessen rechten Hälfte kann der Bediener einige Einstellungen wählen, wobei zwei davon nur bei leerem Trog

verändert werden können: Erstens die Teilefamilie und zweitens der Sicherheitsabstand (bei welchem 100% einem Winkel $\alpha = 45^\circ$ aus Gleichung [3-3] entsprechen). Der Sicherheitsabstand kann nicht vergrößert werden können, wenn einmal Layouts generiert sind, da sonst die Überschneidungsfreiheit nicht mehr garantiert werden kann.

In der linken Hälfte des dritten Bereiches ist der soeben generierte Layoutvorschlag gezeigt. Der Bediener kann mit diesem Bild aufgrund verschiedener Kriterien bestimmen, ob er den Vorschlag annehmen will: Vergleicht er die beiden Layouts aus den Bereichen 2 und 3, kann er selbst den notwendigen Handlungsaufwand abschätzen. Zudem wird oberhalb des Layouts die Troglastung eingblendet.

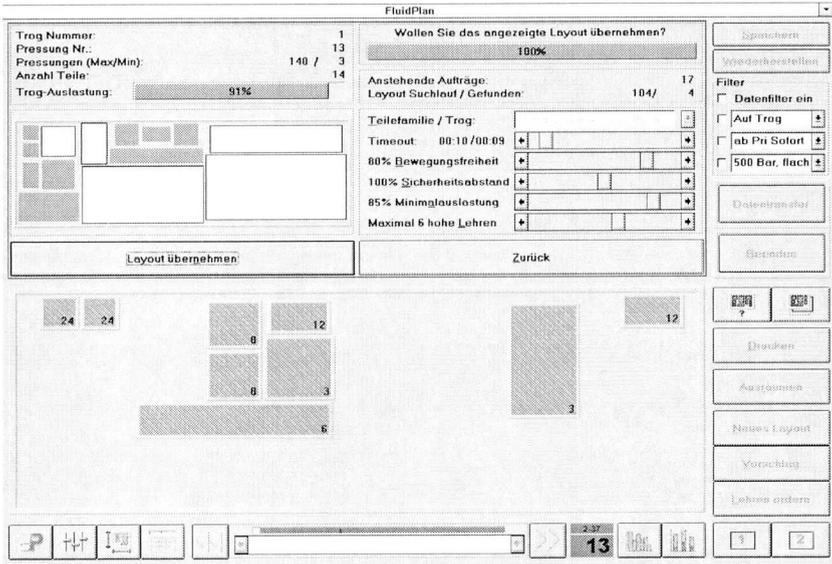


Abbildung A-7: Ansicht des ganzen Bildschirmes mit Layoutvorschlag

Im hier gezeigten Fall muss der Bediener jeden Vorschlag selbst annehmen. Er hat aber auch die Möglichkeit, dass FluidPlan das am höchsten bewertete Layout ohne nachzufragen akzeptiert.

Das beste bisher gefundene Layout wird bereits während der Berechnung in der linken Hälfte des dritten Bereiches angezeigt. Der direkt darunter liegende Schalter fungiert dann als Abbruchknopf, mit welchem man einen Nestingdurchgang frühzeitig beenden kann. Normalerweise wird ein Durchgang aber nach einer einstellbaren Zeit (Timeout-Schieber im dritten Bereich rechts) beendet, welche ab dem ersten gefundenen Vorschlag läuft.

Im Bereich 3 wird sonst, d.h. wenn keine Vorschläge berechnet werden, ein Fenster mit Auftragsdaten eingblendet, wie in Abbildung A-8 gezeigt. In der linken Hälfte befinden sich die eigentlichen Auftragsdaten wie Auftrags und Operationsplannummer, Arbeitsgang, Losgröße und die Priorität. Die Daten der im Operationsplan aufgeführten Matrize (Betriebsmittel) befinden sich in der rechten Hälfte. Deutlich erkennbar sind dabei die Schalter, mit welchen die Tabuzonen einer Matrize gekennzeichnet werden. Die gezeigte Matrize hat gemäss Definition in Abschnitt 3.3.2 eine Tabuzone der Variante V3.

Einige der in diesem Fenster eingetragenen Daten werden nicht von FluidPlan, sondern nur von den Bedienern verwendet, weshalb hier nicht näher darauf eingegangen wird.

| Auftragsnummer | Arbeitsgang | OPP-E Nr. | Betriebsmittel | Lehren Nr. | Füllmaterial | |
|----------------|--------------|-------------|-------------------------------|----------------------|--------------|-----------|
| 90000.0007 | 60 | 90007 | 90007 | 900.00-007.00.00.000 | Keines | |
| 90000.0006 | 60 | Artikel Nr. | 90007 | Länge | 682 | |
| 90000.0007 | | 9000000700 | Teile | 1 | Breite | 104 |
| 90000.0008 | | | Stufen | 1 | Höhe | 30 |
| 90000.0009 | | | tr | 0.010 | Randhöhe | 30 |
| Aufträge | 10 | Termin | te | 0.100 | Gewicht | 0 |
| Los Total | 111 | Los 1 | 111 | Anzahl | 1 | Schmieren |
| Ertig | 0 | Los 2 | 0 | Lage | 1&2 | Lehrentyp |
| Priorität | OPPA-Status | Druck | <input type="checkbox"/> Tief | | | einzel |
| 0 | Normal Aktiv | 700 | | | | |

Abbildung A-8: Fenster mit Auftragsdaten

Zum Schluss sei erwähnt, dass bei Betätigung des im Bereich 1 liegenden Schalters ‚Drucken‘ ein Druckjob gestartet wird, welcher das gerade eingeblendete Layout sowie eine Liste der darin enthaltenen Aufträge ausdruckt. Diese Papiere dienen den Bedienern der Presse als Vorlage.

A2 Literaturverzeichnis

- BECKER M., HABERFELLNER R., LIEBETRAU G.: *EDV-Wissen für Anwender*. Ein Informatik-Handbuch für die Praxis. Verlag Industrielle Organisation, Zürich, 1990.
- BERNATZKI K.-P.: *Entwicklung eines Feinterminierungsmodells als Kern der rechnergestützten operativen Zeitwirtschaft in der Produktionssteuerung eines Stahlwerks*. Fortschritt-Berichte VDI, Reihe 20, Nr. 170 (Dissertation). VDI Verlag GmbH, Düsseldorf, 1995.
- BRÜGGEMANN W.: *Ausgewählte Probleme der Produktionsplanung*. Modellierung, Komplexität und neuere Lösungsmöglichkeiten. Physica-Verlag, Heidelberg, 1995.
- BÜCHEL A.(1): *Betriebswissenschaftliche Methodik. Systemdenken* (Vorlesungsbeilage). Betriebswissenschaftliches Institut, ETH Zürich, 1990.
- BÜCHEL A.(2): *PPS und Logistik. Teil A: Einleitung* (Vorlesungsbeilage). Betriebswissenschaftliches Institut, ETH Zürich, 1990.
- BULLINGER H.-J., HIRSCH B. E. (Hrsg.): *Leitstände für die Werkstattsteuerung*. Erfahrungen, Konzepte und Realisierungsbeispiele. Schriftenreihe der Bundesanstalt für Arbeitsschutz, Fb 705. Wirtschaftsverlag NW, Bremerhaven, 1994.
- DIN/KCIM (Hrsg.): *Schnittstellen der rechnerintegrierten Produktion (CIM) - Fertigungssteuerung und Auftragsabwicklung*. DIN-Fachbericht 21. KOMMISSION COMPUTER INTEGRATED MANUFACTURING (Hrsg.). Beuth Verlag, Berlin, 1989.
- DYCKHOFF H., FINKE U.: *Cutting and Packing in Production and Distribution: A Typology and Bibliography*. Physica-Verlag, Heidelberg, 1992.
- FISCHER K.: *Verteiltes und kooperatives Planen in einer flexiblen Fertigungsumgebung*. Dissertation (an der TU München) zur Künstlichen Intelligenz, Band 26. Verlag infix, Sankt Augustin, 1993.
- GEERING H.P.: *Mess- und Regelungstechnik*. Mathematische Grundlagen, Entwurfsmethoden, Beispiele. Springer Verlag, Berlin, Heidelberg, 1988.
- GEORGI G.: *Job Shop Scheduling in der Produktion*. Lösungen für ein Kernproblem der Produktionsplanung und -steuerung bei mittleren Auftragszahlen und variierenden Einsatzbedingungen. Physica-Verlag, Heidelberg, 1995
- GOBET CH.: *Layoutplanung einer Fluidzellpresse*. Semesterarbeit am Institut für Operations Research und am Institut für Umformtechnik der ETH Zürich, Zürich, 1995.
- HABERFELLNER R., NAGEL P., BECKER M., BÜCHEL A., VON MASSOW H.: *Systems Engineering. Methodik und Praxis*. Verlag Industrielle Organisation, Zürich, 1994.
- HABERSTOCK R., BERG H.: *Von der Idee zum Fertigteil - Einsatzmöglichkeiten der Fluid-Zell-Pressstechnik*. Tagungsband „Zürcher Tagung 1995“. Institut für Umformtechnik der ETH Zürich, Zürich, 1995.

- HÄMMERLE E.: *Werkstatorientierte Systeme zur Arbeitsplanung und kurzfristigen Fertigungssteuerung*. Dissertation am Fachbereich Produktionstechnik der Universität Bremen, 1993.
- HEISTERMANN J.: *Effizientes rechnergestütztes Nesting*. Dissertation an der mathematisch-naturwiss. Fakultät der Rheinischen Friedrich-Wilhelms-Universität, Bonn, 1994.
- HJELMQWIST J.: *Quintus Flexform Presses*. Tagungsbericht „Flexauto 1992“. Zürich, Küssaberg, 1992.
- JUNGNICKEL D.: *Graphen, Netzwerke und Algorithmen*. 3. Auflage. BI Wissenschaftsverlag, Mannheim, Leipzig, Wien, Zürich, 1994.
- KALDICH R.: *Ein Konzept für die objektorientierte und verteilte Realisierung eines parallel arbeitenden OSI-Systems-Management-Agenten*. Dissertation an der technischen Fakultät der Universität Erlangen-Nürnberg, 1993.
- KIMMLINGEN F.: *Agenten-basiertes Prozessmodell mit Berücksichtigung von Kapazitätsrestriktionen*. GMD-Studien Nr. 284, GMD-Forschungszentrum Informationstechnik GmbH, Sankt Augustin, 1996.
- LANGE K. (Hrsg.): *Umformtechnik: Handbuch für Industrie und Wissenschaft*. Band 3: Blechbearbeitung. Springer Verlag, Berlin, 1990.
- MERTINS K., ALBRECHT R., STEINBERGER V.: *Werkstattsteuerung - Werkstattmanagement*. Wegweiser zur Einführung. Hanser Verlag, München, 1992.
- NELISSEN J.: *Die Optimierung zweidimensionaler Zuschnittprobleme*. Schriften zur Informatik und angewandten Mathematik, Bericht Nr. 150. Rheinisch-westfälische technische Hochschule Aachen, 1991.
- NEUMANN K., MORLOCK M.: *Operations Research*. Hanser Verlag, München, Wien, 1993.
- RECH K.: *Regelungsmodell zur Konzipierung der Informationsverarbeitung in der Produktionslogistik*. Dissertation an der Universität des Saarlandes, Schriftenreihe Produktionstechnik (Band 4), Saarbrücken, 1994.
- RENNER A.: *Kostenorientierte Produktionssteuerung*. Anwendung der Prozesskostenrechnung in einem datenbankgestützten Modell für flexibel automatisierte Produktionssysteme. Verlag Franz Vahlen, München, 1991.
- ROHDE B.: *Konzeption eines objektbasierenden Werkstattkommunikationssystems*. Fortschritt-Berichte VDI, Reihe 10, Nr. 348 (Dissertation). VDI Verlag GmbH, Düsseldorf, 1995.
- SCHERER E.: *Koordinierte Autonomie in verteilten, heterogenen Produktionssystemen*. Ein Beitrag zur Flexibilisierung der Werkstattsteuerung (Dissertation). BWI-Publikation im vdf Hochschulverlag AG an der ETH Zürich, Zürich, 1996.
- SCHLAG M.: *An algorithm for optimal two-dimensional compaction of VLSI layouts*. Elsevier Science Publishers B.V., North Holland, 1983.
- SCHRÖDEL O.: *Flexible Werkstattsteuerung mit objektorientierter Softwarestruktur*. Fertigungstechnik - Erlangen; 30. Hanser Verlag, München, 1992.

- SCHULTE: *Werkstattsteuerung mit genetischen Algorithmen und simulativer Bewertung*. Berichte aus dem IPA und dem IAO, Stuttgart, Band 220 (Dissertation). Springer Verlag, Berlin, 1995.
- STORM M. (Hrsg.): *Werkstattinformationssysteme*. Verlag TÜV Rheinland GmbH, Köln, 1993.
- STRACK M.: *Optimale Produktionssteuerung*. Organisation, Wirtschaftlichkeit und Einführung konventioneller, EDV-gestützter Leitstände. Verlag TÜV Rheinland GmbH, Köln, 1986.
- TERNO J., LNDEMANN R., SCHEITHAUER G.: *Zuschnittprobleme und ihre praktische Lösung*. Verlag Harri Deutsch, Thun, Frankfurt am Main, 1987.
- WEBER R.: *Werkstattorganisation und Werkstattsteuerung in der Lean Production*. expert verlag, Ehningen bei Böblingen, 1993.
- ZIMMERMANN H.-J.: *Methoden und Modelle des Operations Research*. Für Ingenieure, Ökonomen und Informatiker. Vieweg Verlag, Braunschweig, Wiesbaden, 1992.

A3 Abbildungsverzeichnis

| | | |
|-----------------|--|----|
| Abbildung 2-1: | Das System und seine Umsysteme | 8 |
| Abbildung 2-2: | Das System mit den Steuerungsinstanzen ‚Vorarbeiter‘ und ‚FluidPlan‘ . . . | 9 |
| Abbildung 2-3: | Möglicher Materialfluss im System | 11 |
| Abbildung 2-4: | Aufbau von typischen Operationsplänen | 11 |
| Abbildung 2-5: | Prinzipielle Darstellung der Fluidzelltechnik | 12 |
| Abbildung 2-6: | Bediener beim Beschicken eines Troges. | 13 |
| Abbildung 3-1: | Gliederung der Tiefziehverfahren. | 23 |
| Abbildung 3-2: | Konventionelles Tiefziehen | 24 |
| Abbildung 3-3: | Ursprung der Einflussgrößen bei einer Fluidzellpresse | 25 |
| Abbildung 3-4: | Einflussmatrix der Kriteriengruppen. | 26 |
| Abbildung 3-5: | Verhältnis von Blechdicke, Radius, Werkstoff und Umformdruck | 27 |
| Abbildung 3-6: | Einfluss der Teilekonstruktion | 29 |
| Abbildung 3-7: | Einflussmatrix | 30 |
| Abbildung 3-8: | Mögliche Ausprägungen von Matrizen. | 31 |
| Abbildung 3-9: | Sicherheitsabstand zwischen zwei Matrizen. | 31 |
| Abbildung 3-10: | Zusammengeschobene Matrizen. | 32 |
| Abbildung 3-11: | Dehnung der Membrane zwischen zwei Matrizen | 33 |
| Abbildung 3-12: | Tabufelder eines Troges | 34 |
| Abbildung 3-13: | Mögliche Tabuzonen. | 35 |
| Abbildung 4-1: | Hierarchie der Funktionselemente | 39 |
| Abbildung 4-2: | Modelliertes System (Systemebene 0) | 41 |
| Abbildung 4-3: | Erste Entwicklungsstufe: Planungsmodul ‚Nesting‘. | 47 |
| Abbildung 4-4: | Zweite Entwicklungsstufe. | 48 |
| Abbildung 4-5: | Ablauf der zweiten Entwicklungsstufe | 50 |
| Abbildung 4-6: | Dritte Entwicklungsstufe. | 51 |
| Abbildung 4-7: | Struktur eines modellierten Auftrags | 58 |
| Abbildung II-1: | Klassisches 2D Bin Packing | 62 |
| Abbildung II-2: | Ausgangslage beim sequentiellen 2D Bin Packing. | 63 |
| Abbildung II-3: | Erlaubte Verschiebungen und Rotationen einer Matrize | 63 |
| Abbildung II-4: | Typologisierung von C&P-Problemen nach Dyckhoff & Finke. | 64 |
| Abbildung 5-1: | Lexikographisch geordneter Enumerationsbaum mit $q = 5$ | 74 |
| Abbildung 5-2: | Enumerationsbaum mit Auslastungsschranken. | 75 |
| Abbildung 5-3: | Verzweigungstypen im Baum. | 76 |
| Abbildung 5-4: | Grobablauf der Teilesetwahl | 77 |
| Abbildung 5-5: | Submodul ‚Layouts suchen‘ | 78 |
| Abbildung 5-6: | Abgeleitete Layoutfolge eines einzigen Teilesets. | 80 |
| Abbildung 5-7: | Auslastungsverteilung einer Layoutfolge | 82 |
| Abbildung 5-8: | Verteilung der Anzahl Aufträge | 82 |
| Abbildung 5-9: | Mit einem einzigen Layout-Aufruf generierte Layoutfolge | 83 |
| Abbildung 5-10: | Layoutfolge nach Aufruf des vorausschauenden Plazierens. | 84 |

| | | |
|-----------------|---|-----|
| Abbildung 6-1: | k-Teilung von T mit $k = 7$ | 88 |
| Abbildung 6-2: | Äquivalente k-Teilungen | 89 |
| Abbildung 6-3: | Trennlinien mit lokaler Notation für Abszissen und Ordinaten | 89 |
| Abbildung 6-4: | 8-Teilung für eine Familie M_1, \dots, M_7 | 90 |
| Abbildung 6-5: | Trennlinien von $R_{z(i)}$ mit lokaler Notation für Abszissen und Ordinaten | 91 |
| Abbildung 6-6: | Freie Zone der Matrize M_4 | 91 |
| Abbildung 6-7: | Freie Plätze neben M_4 | 92 |
| Abbildung 6-8: | Beispiel eines Layouts vor und nach Platzieren von M_6 | 92 |
| Abbildung 6-9: | Neue freie Zone der Matrize M_j | 93 |
| Abbildung 6-10: | Versagen der Einzonenplatz-Methode | 94 |
| Abbildung 6-11: | Benachbarte freie Plätze der Matrizen 2 und 4 | 94 |
| Abbildung 6-12: | Zusammenlegung der benachbarten freien Plätze | 95 |
| Abbildung 6-13: | Beispiel eines Layouts ohne Guillotine-Schnitt | 95 |
| Abbildung 6-14: | Aufteilung der Rechtecke bei einem Mehrzonenplatz | 96 |
| Abbildung 6-15: | Zusammenhang der Ungleichungen [6-4] und [6-5] | 98 |
| Abbildung 6-16: | Bogen eines Digraphen mit Anfangsknoten i und Endknoten j | 100 |
| Abbildung 6-17: | Digraph G_x eines Layouts | 100 |
| Abbildung 6-18: | Analogie zwischen k-Teilung und Digraphen | 101 |
| Abbildung 6-19: | Koordinaten einer Matrize M_j und ihrer freien Zone | 102 |
| Abbildung 6-20: | Längste Wege von M_5 in x-Richtung | 102 |
| Abbildung 6-21: | Suche eines Mehrzonenplatzes für M_6 | 104 |
| Abbildung 6-22: | Ablauf der Layoutgenerierung | 106 |
| Abbildung 6-23: | Grafik zur Bewertung eines Einzonenplatzes | 107 |
| Abbildung 6-24: | Modifizierte Länge bei der Bewertung eines Mehrzonenplatzes | 108 |
| Abbildung 6-25: | Freie Zonen von M_2 | 110 |
| Abbildung 6-26: | Freie Zonen von M_2 in L_{t-1} | 110 |
| Abbildung 6-27: | Mögliche Ansätze, die Verschiebung zu berechnen | 111 |
| Abbildung 6-28: | Einfügen in einen Einzonenplatz | 113 |
| Abbildung 6-29: | Layout nach dem Einfügen von M_6 | 115 |
| Abbildung 6-30: | Nicht notwendige leere Zonen | 115 |
| Abbildung 6-31: | Layout mit leerer Zone, die nicht entfernt werden darf | 116 |
| Abbildung 6-32: | Fälle leerer Zonen | 116 |
| Abbildung 6-33: | Entfernbar leere Zone R_i und neue Trennlinie | 117 |
| Abbildung 6-34: | Einfügen d leerer Zonen bei einem Mehrzonenplatz | 118 |
| Abbildung 6-35: | Trog mit Tabuzone von M_i und freien Plätzen $p1_x(j)$, $p1_x(j)$ | 119 |
| Abbildung 6-36: | Korrektur der freien Zone von M_j aufgrund ihrer Tabuzone | 120 |
| Abbildung 6-37: | Graph G mit den Layouts der Teilesets TS_j | 123 |
| Abbildung 7-1: | Durchschnittliche Wartezeit von Aufträgen der oberen Prioritätsklassen | 127 |
| Abbildung 7-2: | Zeitbedarf der Schachtelung inklusive Trendlinie | 131 |
| Abbildung 7-3: | Plazierung derselben Matrize ohne und mit Tabuzone V5 | 133 |
| Abbildung 7-4: | Printscreen des eingesetzten Datengenerators | 134 |

| | | |
|----------------|--|-----|
| Abbildung A-1: | Bildschirmaufbau von FluidPlan. | 137 |
| Abbildung A-2: | Übersicht der Teilefamilien. | 138 |
| Abbildung A-3: | Auswahl der Ansicht in Bereich 2 | 138 |
| Abbildung A-4: | Erstes akzeptiertes Troglayout einer Teilefamilie. | 138 |
| Abbildung A-5: | Troglayout mit alten und neuen Matrizen | 139 |
| Abbildung A-6: | Layoutschieber | 139 |
| Abbildung A-7: | Ansicht des ganzen Bildschirmes mit Layoutvorschlag | 140 |
| Abbildung A-8: | Fenster mit Auftragsdaten | 141 |

A4 Tabellenverzeichnis

| | |
|---|-----|
| Tabelle 2-1: Wichtigste Inputdaten des Nestingproblems | 18 |
| Tabelle 3-1: Dehnungsverhältnisse bei unterschiedlichen Matrizenhöhen..... | 34 |
| Tabelle 4-1: Einfluss der Priorität auf die Behandlung eines Auftrages..... | 57 |
| Tabelle 4-2: Mögliche Werte des Prioritätsfaktors eines Auftrages..... | 57 |
| Tabelle 4-3: Den Auftragsfortschritt kennzeichnende Attribute..... | 59 |
| Tabelle 4-4: Entscheidungstabelle für die Attributsänderung..... | 59 |
| Tabelle 4-5: Beispiel eines Auftragsfortschritts | 60 |
| Tabelle 5-1: Input des Moduls ‚Teilesetwahl‘ | 70 |
| Tabelle 5-2: Erklärung zu den Verzweigungstypen | 76 |
| Tabelle 5-3: Variablen zur Berechnung von $f_{HM(j)}$ | 79 |
| Tabelle 6-1: Input des Moduls ‚Layoutgenerierung‘ | 105 |
| Tabelle 6-2: Teilwerte eines gefundenen Platzes $p_y(j)$ | 107 |
| Tabelle 6-3: Entscheidungstabelle bei der Plazierung von M_i neben M_j | 109 |
| Tabelle 7-1: Zusammenfassung der Auslastungsdaten | 126 |
| Tabelle 7-2: Verteilung und Wartezeiten der eingeplanten Aufträge | 127 |
| Tabelle 7-3: Verteilung und Wartezeiten der nicht eingeplanten Aufträge | 128 |
| Tabelle 7-4: Zusammenfassung der Anzahl Layouts | 128 |
| Tabelle 7-5: Verteilung der notwendigen Verschiebungen | 129 |
| Tabelle 7-6: Anzahl und Grösse der Aufträge pro Layout | 129 |
| Tabelle 7-7: Im ersten Layout gewählte Aufträge | 130 |
| Tabelle 7-8: Zeitbedarf der Layoutgenerierung in Abhängigkeit der Anzahl Matrizen | 131 |
| Tabelle 7-9: Verhältnis von geprüften und gewählten Ein- und Mehrzonenplätzen | 132 |

A5 Lebenslauf

Personalien

Name: Steiner, Beat Friedrich
Geburtsdatum: 28. April 1966
Heimatort: Schlossrued AG
Wohnadresse: Gemeindestrasse 7
8032 Zürich

Ausbildung

1973 - 1978: Primaschule in Reinach AG
1987 - 1982: Bezirksschule in Reinach AG
1982 - 1985: Matura Typ C an der Alten Kantonsschule Aarau
1986 - 1988: Grundstudium in Maschinenbau an der ETH Zürich
1988 - 1989: Praktikum und befristete Stelle bei der Zühlke Engineering AG, Schlieren
1989- 1992: Studium der Betriebs- und Produktionswissenschaften an der ETH Zürich
Vertiefungsrichtungen: Umformtechnik und Automatisierung
1993 - 1997: Assistenz und Doktorat am Institut für Umformtechnik der ETH Zürich