

Armin Rothhaupt

*Modulares Planungssystem zur  
Optimierung der Elektronikfertigung*





Armin Rothhaupt

*Modulares Planungssystem zur  
Optimierung der Elektronikfertigung*

Herausgegeben von  
Professor Dr.-Ing. Klaus Feldmann,  
Lehrstuhl für  
Fertigungsautomatisierung und Produktionssystematik

**FAPS**



Carl Hanser Verlag München Wien

Als Dissertation genehmigt von der Technischen Fakultät  
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der Einreichung:	11. November 1994
Tag der Promotion:	10. Februar 1995
Dekan:	Prof. Dr. Dr. h. c. F. Durst
Berichterstatter:	Prof. Dr.-Ing. K. Feldmann Prof. Dr.-Ing. habil. W. Sauer

Die Deutsche Bibliothek - CIP - Einheitsaufnahme

**Rothhaupt, Armin:**

Modulares Planungssystem zur Optimierung der  
Elektronikfertigung / Armin Rothhaupt. - München : Hanser,  
1995

(Fertigungstechnik - Erlangen ; 44)

Zugl.: Erlangen, Nürnberg, Univ., Diss., 1995

ISBN 3-446-18307-8

NE: GT

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks  
und der Vervielfältigung des Buches oder Teilen daraus,  
vorbehalten.

Kein Teil des Werkes darf ohne schriftliche Genehmigung des  
Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein  
anderes Verfahren), auch nicht für Zwecke der Unterrichts-  
gestaltung - mit Ausnahme der in den §§ 53, 54 URG ausdrücklich  
genannten Sonderfälle -, reproduziert oder unter Verwendung  
elektronischer Systeme verarbeitet, vervielfältigt oder  
verbreitet werden.

© Carl Hanser Verlag München, Wien 1995

Herstellung: Gruner Druck GmbH, Erlangen-Eltersdorf

Printed in Germany

## **Vorwort**

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Assistent am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Herrn Prof. Dr.-Ing. K. Feldmann, dem Leiter dieses Lehrstuhls am Institut für Fertigungstechnik, danke ich für die großzügige Förderung bei der Durchführung der Arbeit.

Herrn Prof. Dr.-Ing. habil. W. Sauer, dem Leiter des Instituts für Elektroniktechnologie an der Technischen Universität Dresden, danke ich für die Übernahme des Koreferates.

Herrn Prof. Dr.-Ing. A. Weckenmann danke ich für die Bemühungen bei der Durchführung des Promotionsverfahrens.

Weiterhin danke ich allen meinen Kollegen, insbesondere der Gruppe für Elektronikproduktion, für die ständige Gesprächs- und Hilfsbereitschaft und die hervorragende Zusammenarbeit in den Jahren am Lehrstuhl. Ferner gilt mein Dank allen Studenten und wissenschaftlichen Hilfskräften, besonders Herrn Dipl.-Ing. M. Czech, die mich beim Erstellen dieser Arbeit unterstützt haben.

Zusätzlich bedanke ich mich bei Herrn Dr. rer. pol. Raimund Belz und Herrn Dipl.-Ing. Stefan Krug für die kritischen und wertvollen fachlichen Anregungen beim Auseinandersetzen mit meiner Arbeit.

Ein besonderer Dank gebührt meiner Frau Doris für eine Reihe guter gestaltungstechnischer Tips, das nötige Verständnis und das ausgleichende Umfeld, das zum Gelingen dieser Arbeit beigetragen hat.

Armin Rothhaupt



# **Modulares Planungssystem zur Optimierung der Elektronikfertigung**

## **– Inhaltsverzeichnis –**

<b>1</b>	<b>Einleitung .....</b>	<b>1</b>
<b>2</b>	<b>Analyse der Bestückung elektronischer Flachbaugruppen .....</b>	<b>4</b>
2.1	Analyse des Bestückungsprozesses .....	4
2.1.1	Handbestückung .....	4
2.1.2	Halbautomatische Bestückung .....	4
2.1.3	Vollautomatische Bestückung .....	5
2.2	Kinematische Klassifizierung der Bestückungsautomaten .....	8
2.3	Konfigurationsmöglichkeiten der Bestückungssysteme .....	10
2.4	Verkettung der Einzelmodule zu Bestückungslinien .....	12
<b>3</b>	<b>Planungssysteme in der Elektronikfertigung .....</b>	<b>13</b>
3.1	Rechnergestützte Planungssysteme – State of the art – .....	13
3.2	Defizite rechnergestützter Planungssysteme .....	16
3.3	Anforderungen an ein rechnergestütztes Planungssystem für die Elektronik .....	16
3.3.1	Datenhaltung und adaptiver Output .....	17
3.3.2	Rüstproblematik und Rüstanforderungen .....	18
3.3.3	Optimierungsfunktionalitäten .....	22
3.3.4	Kostenkalkulation und Zeitabschätzungen .....	23
3.3.5	Aufbau und Integration von Regelkreisen .....	24
3.3.6	Fertigungstransparenz und Informationsdarstellung .....	25
<b>4</b>	<b>Konzept eines modularen Planungssystems für die Elektronikfertigung .....</b>	<b>26</b>
4.1	Entwurf einer geeigneten Softwarearchitektur .....	27
4.2	Datenmodell zur Planungskomponente .....	29
4.2.1	Strukturierung der Daten .....	29
4.2.2	Entwicklung eines Datenbankkonzeptes .....	30
4.2.3	Entwicklung von Anwendereditoren .....	32

4.3	Entwicklung einer Hauptspeicherstruktur für schnelle Datenzugriffe .....	34
4.3.1	Entwurf eines allgemeingültigen Rüstmodells .....	36
4.3.2	Erstellung von Zeitmodellen für die verschiedenen Automatentypen .....	40
<b>5</b>	<b>Import– und Exportfunktionalitäten .....</b>	<b>47</b>
5.1	Datenformate und Schnittstellenproblematik der Elektronikfertigung .....	47
5.1.1	Gerber-Format .....	47
5.1.2	IGES 5.1 .....	48
5.1.3	IPC-D-35x .....	48
5.1.4	EDIF 3.0 .....	49
5.1.5	STEP .....	49
5.2	Anbindung an die Leiterplattenentwicklung .....	50
5.3	Anbindung der Prozeßebebene mittels eines flexiblen Postprozessors .....	52
<b>6</b>	<b>Entwicklung eines Design-for-Manufacturing-Systems .....</b>	<b>55</b>
6.1	Wissensrepräsentation .....	57
6.1.1	Regeln .....	58
6.1.2	Objektorientierte Programmierung .....	58
6.2	Wissensakquisition .....	59
6.2.1	Formalisierung der Fehlerhypothesen .....	60
6.2.2	Repräsentation und Gewichtung der Fehler .....	64
6.3	Erstellung eines DFM-Prototyps .....	66
6.4	Ablauf und Visualisierung des Design-Checks .....	70
<b>7</b>	<b>Entwicklung eines Modules zur Rüst- und Umrüstoptimierung .....</b>	<b>73</b>
7.1	Einteilung der Optimierungsprobleme .....	73
7.1.1	Bekannte modellierte Optimierungsprobleme der diskreten Optimierung .....	75
7.1.2	Optimierungsverfahren .....	76
7.2	Entwicklung eines Schichtenmodells zur Optimierung .....	87
7.3	Umrüstoptimierung und Chargenminimierung .....	89
7.4	Gleichmäßige Austaktung einzelner Bestückungslinien .....	94

7.5	Optimierung der Feederanordnungen	100
7.5.1	Spuroptimierung für Maschinenklasse III	102
7.5.2	Spuroptimierung für Maschinenklasse V	109
7.5.3	Spuroptimierung für Maschinenklasse VI	112
7.6	Optimierung der Bestücksequenzen	122
7.7	Einbindung der Optimierung für Automaten der Durchsteckmontage	125
7.8	Zusätzliche Optimierungsmöglichkeiten	127
7.8.1	Nutzenoptimierung	127
7.8.2	Ermittlung einer Basisrüstung	131
7.9	Visualisierung von Optimierungsergebnissen	133
<b>8</b>	<b>Entwicklung eines Moduls zur Auftragseinplanung</b>	<b>135</b>
8.1	Entwicklung eines Werkskalenders	135
8.2	Elektronische Plantafel zur Auftragsterminierung	137
8.2.1	Feinterminierung von Auftragslisten	137
8.2.2	Simulation von Umplanungen	138
8.3	Kalkulation der Fertigungskosten	139
8.4	Adaptive Erstellung von Fertigungsunterlagen	141
8.5	Aufbau einer BDE/MDE-Kopplung	142
<b>9</b>	<b>Testergebnisse des realisierten Optimierungssystems DOPE</b>	<b>143</b>
9.1	Beschreibung der Testparameter	143
9.2	Validierung der Optimierungs- und Zeitmodelle mittels Praxisergebnissen	144
9.3	Diskussion der Optimierungsläufe	147
9.4	Wirtschaftlichkeitsbetrachtungen und Einsparungspotentiale	149
<b>10</b>	<b>Zusammenfassung</b>	<b>151</b>
	<b>Literatur</b>	<b>153</b>

## **Abkürzungen**

ASCII	American Standard Code for Information Interchange
ANSI	American National Standard Institute
BE	Bauelement
BP	Bestückposition(en)
BGA	Ball Grid Array
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAP	Computer Aided Planning
COB	Chip on Board
DB	Datenbank
DFM	Design-for-Manufacturing
DXF	Data Exchange Format
ECAD	Electronic Computer Aided Design
EDIF	Electronic Design Interchange Format
ER	Entity Relationship
ESQL	Embedded Structured Query Language
FIFO	First In First Out
GA	Genetischer Algorithmus
IC	Integrated Circuit
IGES	Initial Graphics Exchange Specification
IPC	Institute for Interconnecting and Packaging Electronic Circuits
ISO	International Standards Organisation
LP	Leiterplatte
MID	Molded Interconnection Device
MIPS	Million Instructions Per Second
NP	Nicht Polynomial
OR	Operation Research
PDES	Product Data Exchange Specification
PPS	Produktionsplanung und -steuerung
SMD	Surface Mount Device
SMT	Surface Mount Technology
STEP	Standard for the Exchange of Product model data
SQL	Structured Query Language
THD	Through Hole Device
THT	Through Hole Technology
TSP	Traveling Salesman Problem



## 1 **Einleitung**

Vorherrschende Themen in der Elektronikproduktion sind die Reduzierung der Produktentwicklungszeiten, die Minimierung der Durchlaufzeiten und Bestände sowie die Sicherung der Qualität gekoppelt mit der Nutzung neuester Technologien [24]. Besonders problematisch sind in der Regel die komplexen und schwierig zu beherrschenden Prozesse, die sowohl technologischen als auch informationstechnischen Querbeziehungen der einzelnen Prozesse und ein unvollständiger und ineffizienter Daten- bzw. Informationsfluß. Neue Anforderungen an die Prozesse resultieren aus den speziell im Elektroniksektor stetig auftretenden Neuentwicklungen wie neuartigen Packungsformen von Bauelementen sowie die Miniaturisierung und Verringerung der Anschlußrastermaße [20, 88]. Auch Innovationen im Bereich der Prozeßtechnologien, wie beispielsweise die MID-Technologie, Leitlebetechnik oder Laserlötverfahren usw. steigern die Komplexität der Elektronikproduktion [25].

Häufige, aktuell auftretende Probleme in der Praxis sind hohe Liegezeiten der Aufträge in der Fertigung, ein hoher Anteil an Nebenzeiten durch verstärkte Rüst- und Umrüsttätigkeiten oder durch falsch disponiertes Material, aber auch Mehraufwand in einzelnen Bereichen mangels Informationen bzw. Absprachen unter den Abteilungen. Oft werden in der Entwicklung weitreichende Entscheidungen getroffen, die in nachfolgenden Prozeßschritten einen zusätzlichen Zeit- und somit Kostenaufwand verursachen.

Ziel muß es sein, Verbesserungspotentiale in einzelnen Bereichen sowie prozeß- und bereichsübergreifend zu erkennen und auszuschöpfen. Damit verkürzt sich die entscheidende "time to market". Leistungsfähige ECAD-Systeme zur Entwicklung der Produkte sowie die Entwicklung schnellerer und flexiblerer Bestückungssysteme im Prozeßbereich sind nur einige der Faktoren, die dazu beitragen [111]. Ein weiterer Ansatz ist der Einsatz rechnerunterstützter Planungssysteme, die den Anwender von der Produktentstehung über Auftragseinplanung bis zur Programmerzeugung für die Maschinen unterstützen und einen durchgängigen Informations- bzw. Datenfluß gewährleisten.

In der vorliegenden Arbeit wird eine rechnergestützte Planungsmethode vorgestellt, die von der Leiterplattenentwicklung bis zur Produktion eingesetzt wird. Basierend auf einer zentralen Datenbank, die ein Produkt- und ein Prozeßmodell beinhaltet, wurden mehrere Module entwickelt, die den Anwender in den verschiedenen Planungsaktivitäten unterstützen. Die Struktur des Gesamtsystems wurde so konzipiert, daß sämtliche Prozeßschritte der Flachbaugruppenfertigung integrierbar sind. Hauptaugenmerk lag dabei auf den Planungstätigkeiten im Bereich der Flachbaugruppenbestückung, insbesondere der SMD-Bestückung. Bedingt durch die große Bauelementevielfalt und der dicht gepackten, komplexen Leiterplatten ergeben sich gerade in diesem Sektor hohe Bestück- sowie Rüstzeitanteile. Dies führt oft dazu, daß dieser Prozeßschritt zum Flaschenhals der gesamten Prozeßkette wird und somit die

Durchlaufzeiten erhöht. Aus diesem Grund wird der Bestückbereich, zu berücksichtigende Parameter und die Zusammenhänge in Kapitel zwei genauer analysiert.

In dem folgendem Kapitel werden verfügbare Planungssysteme und die möglichen Einsatzgebiete einer kritischen Betrachtung unterzogen. Die Anforderungen seitens der Benutzer diverser Systeme, aber auch von potentiellen Anwendern wurden aufgenommen und ausgewertet. Die ermittelten Defizite dienten als Ausgangsbasis zum Entwurf des Planungssystems.

Kapitel vier und fünf erläutern sowohl die Struktur des Gesamtsystems als auch die Anbindungen an die Außenwelt. Beschrieben wird das Zusammenwirken der einzelnen Module, die Strukturierung sowie Realisierung der zentralen Datenbasis und die verwendete Softwarearchitektur. Weitere wichtige Punkte dieses Kapitels sind die Entwicklung einer Hauptspeicherstruktur zur detaillierten Modellierung der verschiedenen Bestückungssysteme. Da die Planungskomponente als Bindeglied fungieren soll und zur Aufgabe hat, einen durchgängigen Datenfluß zu gewährleisten, sind Schnittstellen zur Layoutentwicklung und zur Prozeßebene unabdingbar. Dazu werden in diesen Kapiteln die spezifischen Schnittstellenprobleme in der Elektronikfertigung erläutert und die gewählten bzw. entwickelten Lösungen vorgestellt.

Eine Komponente des Planungstools, vorgestellt in Kapitel sechs, ist ein Design-For-Manufacturing-System (DFM) zur Verbindung der Leiterplattenentwicklung und der Fertigung. Dieses Werkzeug ermöglicht es, bereits in der Entwicklungsphase einer Leiterplatte Anforderungen aus nachfolgenden Prozeßschritten, wie beispielsweise der Bestückung, des Lötens und des Testens zu berücksichtigen. Der erstellte Prototyp des Expertensystems kann, neben bereits integrierten Basiswissen, an die jeweilige Fertigung adaptiertes Fertigungs-Know-how aufnehmen. Zur Erweiterung bleibt die Struktur des Gesamtsystems unverändert, es sind lediglich im Regelteil die neuen Design-Rules einzugeben. Das spezifische Wissen über die Leiterplatten und das technische Equipment bezieht das DFM-System aus der zentralen relationalen Datenbank. Der rechnergestützte Regelkreis deckt frühzeitig Defizite und Fertigungsprobleme auf und unterstützt den Entwickler das Design fertigungs- bzw. testgerecht zu gestalten.

Einen Schwerpunkt der Arbeit bildet das Planungsmodul zur Rüst- und Umrüsto-optimierung, das in Kapitel sieben beschrieben wird. Ausgehend von technologischen Daten aus der Entwicklung und dispositiven Auftragsdaten, generiert dieses Modul für den kompletten Bestückungsbereich Lösungen, die einen optimalen Durchsatz der Flachbaugruppen gewährleisten. Je nach vorliegender Fertigung und Anwenderwunsch führt es verschiedene Optimierungsstrategien einzeln oder gekoppelt aus. Der Kern des Moduls besteht aus einem Schichtenmodell, das in eine bereichs-, linien- und maschinenbezogene Optimierung unterteilt ist. Die einzelnen Optimierungsschritte ermöglichen die Minimierung der gesamten Nebenzeiten innerhalb der Leiterplattenbestückung und die Reduzierung der Bearbeitungszeiten der Aufträge an den einzelnen Bestückungssystemen. Das zu produzierende Auftragspektrum

wird den verschiedenen Bestückungssystemen hinsichtlich technologischer und wirtschaftlicher Kriterien in einer umrüstoptimalen Reihenfolge zugeteilt. Aufgabe der linienbezogenen Optimierung ist es, unausgewogene Austaktungen der einzelnen Automaten innerhalb verketteter Bestückungslinien auszugleichen. Die maschinenorientierten Schichten errechnen zur Einsparung von Maschinenhauptzeiten in Abhängigkeit der einzelnen Automatentypen optimale Feederanordnungen und Bestücksequenzen. Eine maßgebliche Anforderung, die diese Komponente erfüllt ist, daß durch den allgemeingültigen, flexiblen Ansatz die Optimierung auf sämtliche am Markt verfügbare Bestückungssysteme anwendbar ist.

Lösungen für den Bereich der Arbeitsplanung sind in Kapitel acht beschrieben. Durch die Integration eines Werkskalenders in Verbindung mit den Resultaten der Optimierungskomponente sind detaillierte Kapazitätsplanungen und Feinterminierungen von Aufträgen durchführbar. So lassen sich neben Vorwärts- und Rückwärtsterminierung auch verschiedene Auftragsreihenfolgen und deren Auswirkungen auf Fertigungszeiten simulieren. Verknüpft mit den Maschinenstundensätzen der Bestückungssysteme stehen damit dem Anwender auch Aussagen über die zu erwartenden Fertigungskosten zur Verfügung. Zudem können innerhalb dieses Moduls Informationen über den laufenden Prozeß oder auch gespeicherte Auswertungen visualisiert werden. Es besteht die Möglichkeit, sich verschiedenartige Fertigungsdokumentationen (z.B. Rüstlisten, Zeitauswertungen etc.) passend zu den jeweiligen Fertigungsspezifika zu generieren.

Im letzten Kapitel wird eine Validierung der Ergebnisse vorgenommen. Hierzu wurden etliche theoretische Versuche durchgeführt, um die Ergebnisse zu diskutieren. Untermauert durch einige Tests in der laufenden Produktion, wurde auch die Funktionsfähigkeit des Prototypen inklusive Optimierungsergebnissen und Zeitsimulationen in der Praxis analysiert. Mit Hilfe dieser Daten und Auswertungen wurden Wirtschaftlichkeitsbetrachtungen durchgeführt und Rationalisierungspotentiale in diesem Bereich quantifiziert.

## **2      *Analyse der Bestückung elektronischer Flachbaugruppen***

### **2.1    *Analyse des Bestückungsprozesses***

Bei der Fertigung von Flachbaugruppen ist das Bestücken, d. h. das Fügen der Bauelemente auf die Leiterplatte, einer der wichtigsten Prozeßschritte und meist der zeitintensivste. Prinzipiell existieren hierfür zwei alternative Bestückungstechnologien, die Through Hole Technology (THT) und die Surface Mount Technology (SMT), wobei auch eine Mischform möglich ist. Seit Einführung der SMT hat diese durch eine Reihe von Vorteilen (höhere Packungsdichten, bessere Automatisierbarkeit, weniger Prozeßschritte etc.) die THT zunehmend verdrängt und liegt heute bei einem Anteil von ca. 50%, mit immer noch zunehmender Tendenz [9,88]. Sowohl in der THT, als auch in der SMT, werden abhängig von der Anzahl der Bauelemente, den zu bestückenden Bauelementetypen, den Leiterplattenvarianten und den zu produzierenden Stückzahlen verschiedene Bestückungsmethoden eingesetzt [97]:

- Handbestückung
- Halbautomatische Bestückung
- Vollautomatische Bestückung

#### **2.1.1   *Handbestückung***

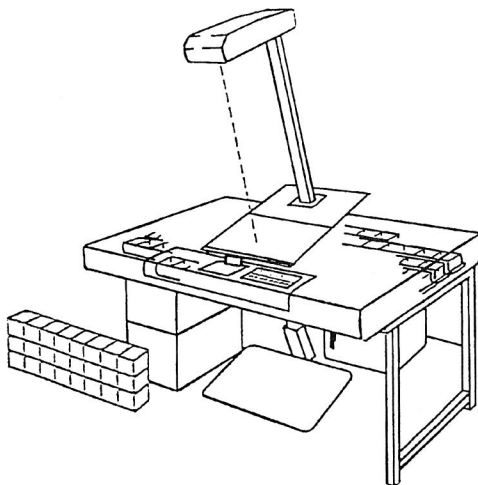
Handbestückung erfolgt im einfachsten Fall mittels Pinzette oder mit einem handgeführten Bestückungswerkzeug. Die Bauelemente stehen in Griffschalen auf einem Tisch oder Karussell bereit. Aufgrund der kleinen Bauformen sind bei der manuellen SMD-Bestückung optische Vergrößerungseinheiten notwendig.

Bei einfachen Platinen, optimalen Lichtverhältnissen, einer Handauflage und grifftechnisch günstig angeordneten Bauteilspendern kann mit der Handbestückung eine Bestückleistung von ca. 800 Bauelemente pro Stunde [48] erreicht werden. Eingesetzt wird die manuelle Montage hauptsächlich zu Reparaturzwecken und im Laborbetrieb für Musterleiterplatten.

#### **2.1.2   *Halbautomatische Bestückung***

Die Unterschiede der Halbautomatischen Bestückung zur Manuellen Bestückung liegen in der Bauelementebereitstellung und der Vorgabe der Bestückposition durch Markierungen. Die Bauelemente werden auf einem Karussell oder Endlosband in Griffschalen bereitgestellt. Dabei wird dem Bediener immer nur der Zugriff zu den aktuell zu bestückenden Bauelementen ermöglicht, da nur diese Griffschale vom Bestückungssystem freigegeben wird. Eine Projektionseinheit (Licht- oder Laserstrahl) zeigt die Bestückposition auf der Leiterplatte an (s. Bild 2-1). Halbautomatische Bestückungssysteme kommen in der THT zum Einsatz, bevorzugt bei kleinen bis mittle-

ren Serien oder für die Bestückung von Sonderbauelementen, die nicht automatisch bestückt werden können. Die eigentliche Montage erfolgt weiterhin per Hand, wobei die Bedienperson auf die richtige Polarität zu achten hat.



*Bild 2-1: Halbautomatischer Bestückungsautomat (Royonic)*

Die Bestückungsleistung bewegt sich je nach Komplexität der Leiterplatte zwischen 700 und 1000 Bauelementen in der Stunde. Die wesentlichen Vorteile gegenüber der Handbestückung sind:

- Verringerung der Fehlermöglichkeit durch programmgeführten Bauelementezugriff und Bestückpositionsmarkierung.
- Geringere Umgewöhnungszeit bei Änderung des Bestückprogramms.
- Anbindung an ECAD-Systeme möglich.

### **2.1.3 Vollautomatische Bestückung**

Für die vollautomatische Bestückung im Bereich der SMT, der schwerpunktmäßig untersucht wurde, existiert ein breites Spektrum an unterschiedlichen Bestückungssystemen.

Der Einsatz von SMDs begünstigt eine automatisierte Montage außerordentlich, da die Bauelemente nur auf die Leiterplatte aufgesetzt werden. Somit entfallen – im Gegensatz zur automatischen Bestückung von bedrahteten Bauelementen – bauteilspezifische Führungs- und Unterwerkzeuge für das Einfädeln und Umbiegen der Anschlußdrähte. Außerdem steht der aufgrund der Größe der Bestückwerkzeuge beim konventionellen Bestücken erforderliche Raum (Bestückschatten) zusätzlich als Bestückfläche zur Verfügung.

Bei der vollautomatischen Bestückung unterscheidet man zwei Verfahren [57]:

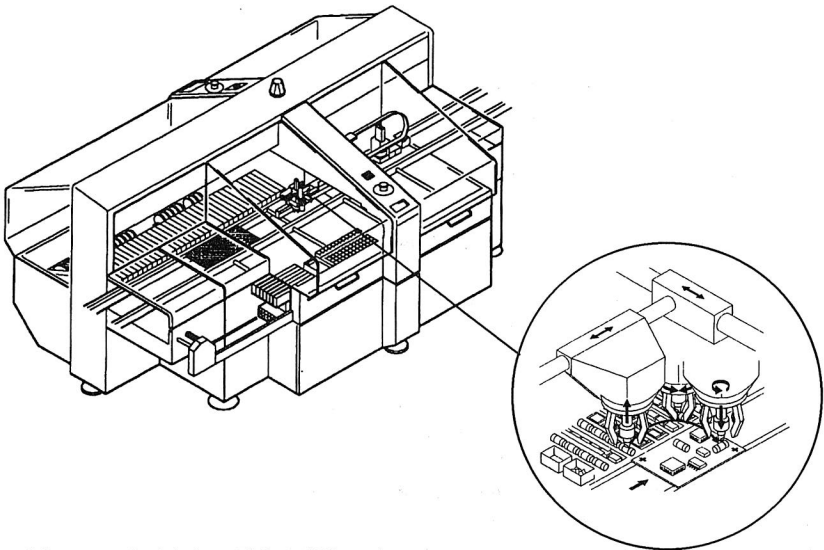
- Sequentielle Bestückung
- Simultane Bestückung

### Sequentielle Bestückung

Die sequentiellen Bestückungsautomaten sind am weitesten verbreitet. Sie lassen sich nach ihrer stündlichen Bestückleistung, aber auch nach ihrem kinematischen Aufbau klassifizieren [40].

Die Anforderungen an die vollautomatischen Bestückungsautomaten sind zum einen hohe Bestückleistung und zum anderen Verarbeitung immer kleiner werdender Bauelemente und verringerter Anschlußrastermaße. Diese in einem Maschinenkonzept schwer zu vereinbarenden Forderungen führten zu einer Diversifizierung der Automatentypen:

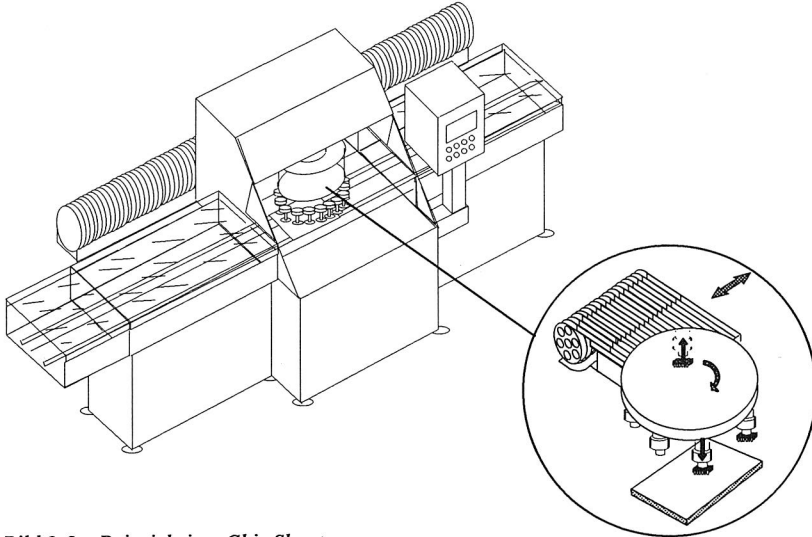
**Pick and Place-Bestückungsautomaten** (device placer, fine pitch placer) dienen zur Verarbeitung hochpoliger, großer Bauelemente. Innerhalb eines Bestückvorgangs wird ein Bauelement aus den Zuführsystemen gegriffen (pick), von dort auf die Leiterplatte verfahren und auf die Oberfläche gesetzt (place). Die Bewegung des Bestückkopfes kann dabei auf unterschiedliche Arten erfolgen.



*Bild 2-2: Beispiel eines Pick and Place-Automaten*

### Chip Shooter

sind für das Bestücken von kleinen, meist zweipoligen Bauelementen mit hohen Mengendurchsatz bestimmt. Die hohe Bestückleistung wird durch einen Revolverkopf mit mehreren Pipetten erreicht.



*Bild 2-3: Beispiel eines Chip Shooters*

Gerade für hohe Stückzahlen ist diese rein mengenorientierte Aufteilung der Bestückinhalte einer Leiterplatte von Vorteil. So setzt man einerseits spezialisierte Automaten für das sogenannte "Vogelfutter" mit hoher Bestückleistung ein und andererseits Automaten für präzises Bestücken von meist großen Gehäuseformen.

### Simultanbestückung

Die Simultanbestückung gibt es in zwei Ausprägungen :

- Automatische Sequentiell-/Simultanbestückung
- Automatische Simultanbestückung

Bei der automatischen Simultanbestückung (s. Bild 2-4) werden die Bauelemente gleichzeitig dem Förderbereich entnommen und bestückt. Mit diesen Bestückungssystemen können extrem hohe Bestückraten erzielt werden. Nachteilig ist jedoch der hohe Umrüstaufwand bei Leiterplattenwechsel, da die Automaten stark produktbezogen konfiguriert sind. Zudem muß bereits beim Leiterplattendesign die rasterförmige Anordnung des Bestückwerkzeugs berücksichtigt werden.

Bei der automatischen Sequentiell-/Simultanbestückung (s. Bild 2-4) werden die Bauelemente simultan aus den Zuführmodulen entnommen und sequentiell auf der Leiterplatte plaziert.

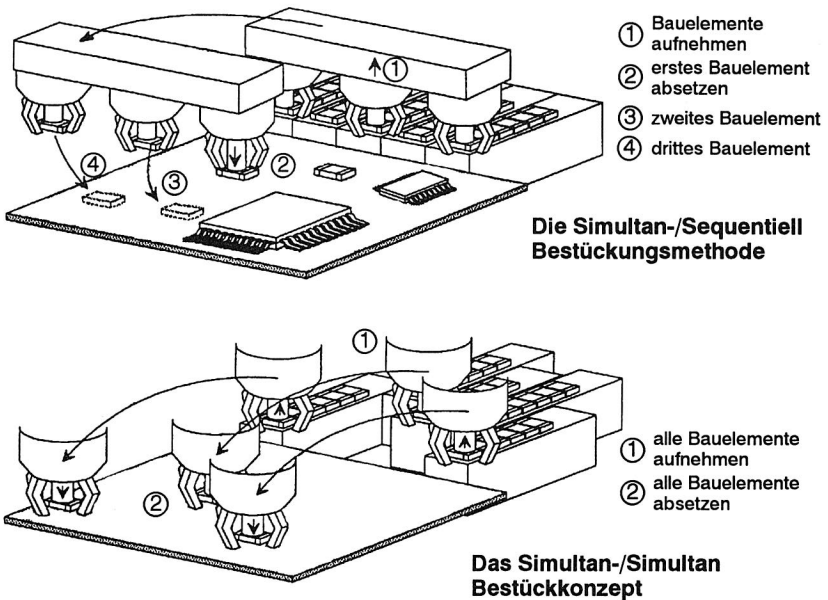


Bild 2-4: Simultanbestückung und Sequentiell-/Simultanbestückung

## 2.2 Kinematische Klassifizierung der Bestückungsautomaten

Zur Datenmodellierung der diversen Bestückungssysteme wurde eine Unterteilung in manuelle Bestückungssysteme, Automaten für die bedrahtete Bestückung und Automaten zur SMD-Bestückung zugrunde gelegt (s. Bild 2-5). Die Bestückungssysteme für die bedrahtete Technik werden entsprechend den Bauelementen in axiale, radiale, DIP und Sonderbestücker unterschieden.

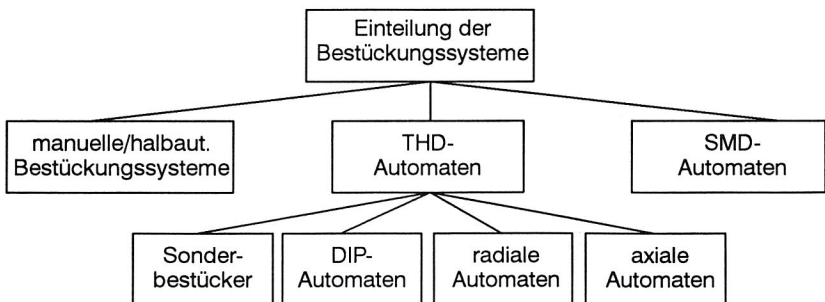
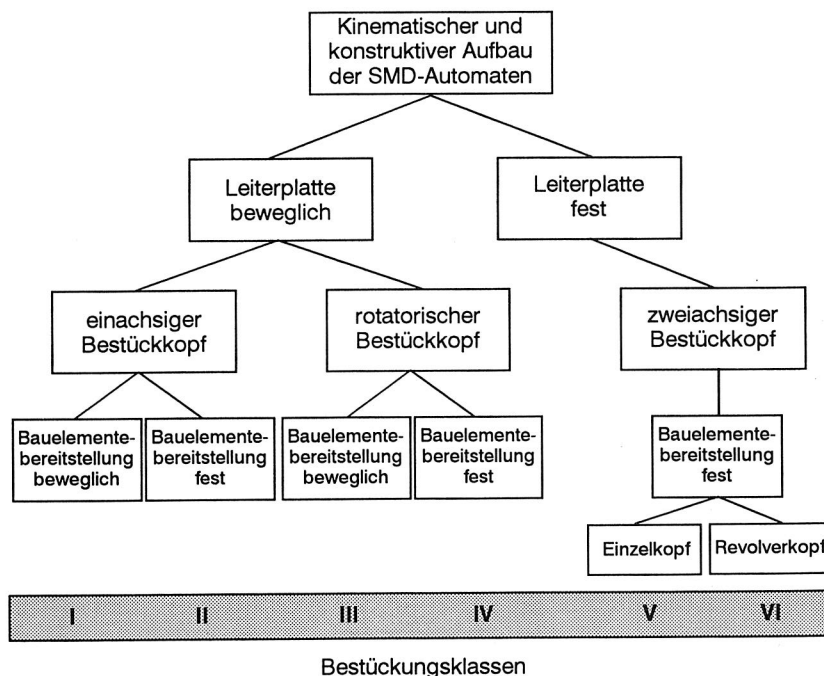


Bild 2-5: Einteilung der Bestückungssysteme zur Datenmodellierung



Die SMD-Automaten werden hinsichtlich ihrer Kinematik klassifiziert. Die Einteilung in verschiedene Bestückungsklassen ermöglicht gerade im Hinblick auf die erforderlichen Optimierungsalgorithmen eine zusammenfassende Betrachtungsweise für Automaten unterschiedlicher Hersteller (s. Bild 2-6).



*Bild 2-6: Klassifizierung der SMD-Automaten*

In Bild 2-7 sind die modellierten SMD-Bestückungssysteme, das kinematische Prinzip, die Leistungsklasse und einige Beispiele aufgeführt. Jede Bestückungsklasse erfordert ein eigenes Zeitmodell. Auf die erforderliche Zeit- und Rüstmodelle wird in den Kapiteln 4.3.1 und 4.3.2 noch näher eingegangen.

Zusätzlich existieren zu den eingeteilten Bestückungsklassen einige Automatentypen, die auf den ersten Blick nicht eindeutig einzuordnen sind, wie z.B. die Philips FCM oder die Siplace 80S von Siemens. Die Philips FCM besteht aus mehreren Pick and Place Köpfen, die segmentweise die Leiterplatte bestücken. Zur Modellierung kann dieser Automatentyp als eine Aneinanderreihung mehrerer Einzelköpfe der Klasse V aufgefaßt werden. Die Siplace 80S ist eine Doppelportalmaschine. Sie kann in der Form modelliert werden, daß zwei Komponenten der Klasse VI zusammenarbeiten, die synchronisiert werden müssen.

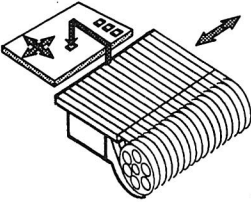
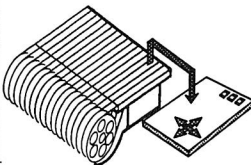
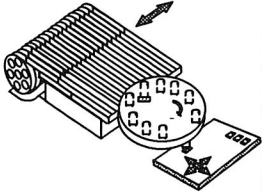
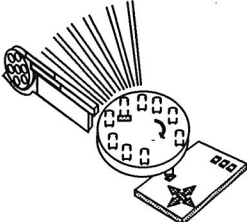
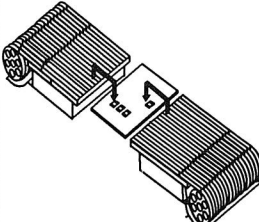
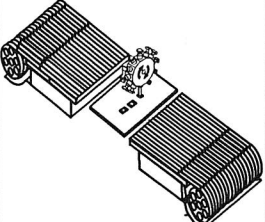
Klasse I	Klasse II	Klasse III
Leistungsklasse: bis ca. 6000 BE/h	Leistungsklasse: bis ca. 5000 BE/h	Leistungsklasse: bis ca. 25000 BE/h
Beispiele: Dynapert MPS 318 Dynapert MPS 525	Beispiele: Mydata TP-9 Fuji FIP I, FIP II Amistar FA-2001	Beispiele: Fuji CP III Universal HSP 4780 Panasonic Panasert Mv IIc
		
Klasse IV	Klasse V	Klasse VI
Leistungsklasse: bis ca. 14000 BE/h	Leistungsklasse: bis ca. 5000 BE/h	Leistungsklasse: bis ca. 7000 BE/h
Beispiele: TDK avimount RX-4260 Amistar SM-64 Universal Ultraplace 4715	Beispiele: Siemens HS-180 Philips CSM Eclipse Sanyo TCM-V500	Beispiele: Siemens SP-120 Siemens Siplace 80S Dynapert MPS 109
		

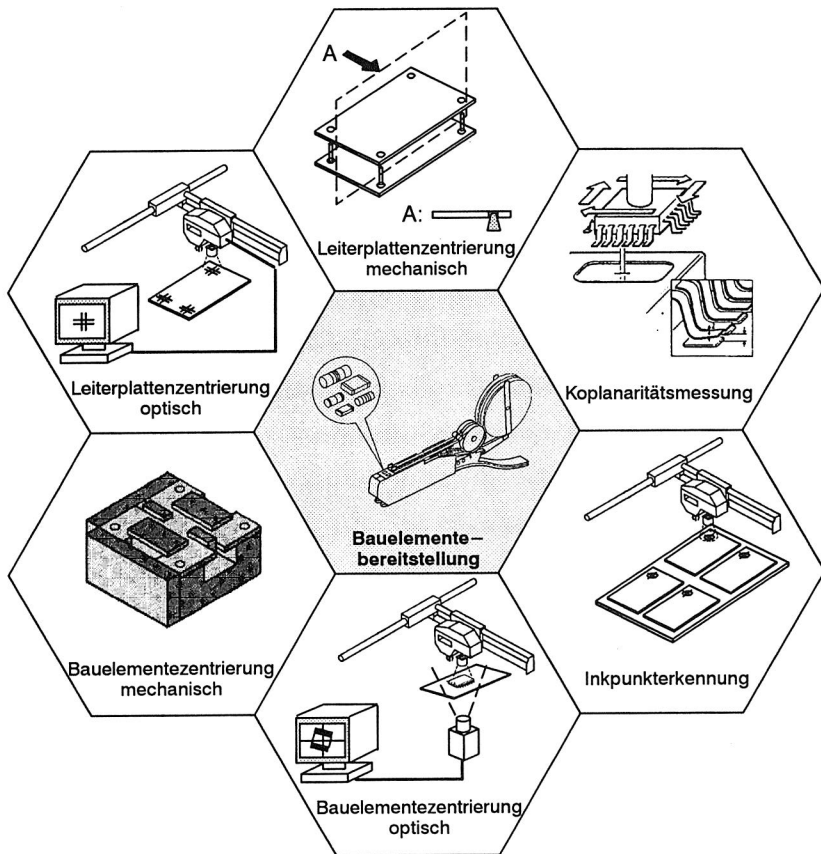
Bild 2-7: Bestückprinzipien der verschiedenen Bestückungsklassen

### 2.3 Konfigurationsmöglichkeiten der Bestückungssysteme

Neben den kinematischen Grundprinzipien der Automaten existieren eine Reihe von Zusatzeinrichtungen der Bestückungssysteme, die den Bestückungsprozeß beeinflussen bzw. wichtige technische Randbedingungen der jeweiligen Automatentypen darstellen (s. Bild 2-8). Ebenso wie die Bestückungsklasse sind der Aufbau und die Einzelkomponenten der Bestückungssysteme zu modellieren.

Wichtige zu modellierende Kriterien dieser Komponenten sind:

- Welche Zuführeinrichtungen sind für welchen Automaten vorhanden
- Platzbedarf
- Auf- und Abrüstzeit
- Wechselwirkungen mit anderen Komponenten
- Zeit für die einzelnen Funktionen
- Stellmöglichkeiten



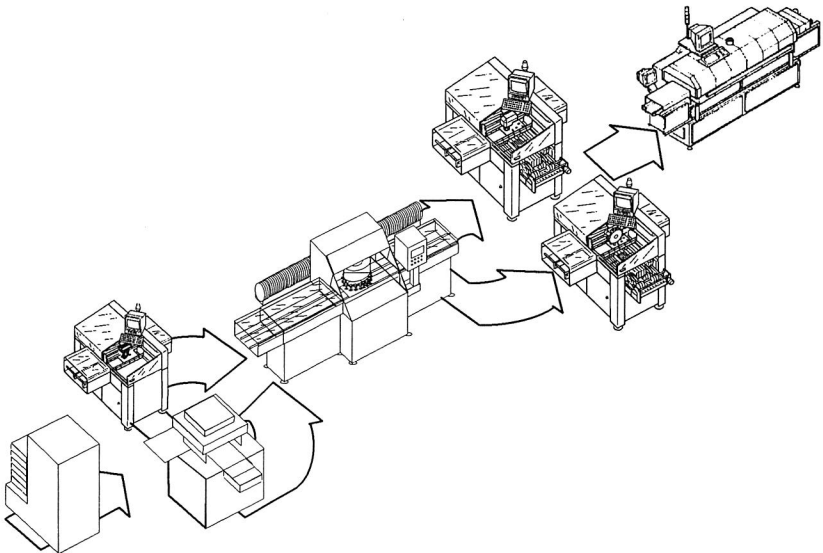
*Bild 2-8: Übersicht zu den modellierenden Stationskonfigurationen mit Schwerpunkt der Bauelementebereitstellung*

## 2.4 Verkettung der Einzelmodule zu Bestückungslinien

Die Zusammensetzung einer Bestückungslinie bestimmt sich aus den Zielvorstellungen des Anwenders, aus dem zu verarbeitenden Produktspektrum und aus den räumlichen Gegebenheiten. Entsprechend diesen Anforderungen lassen sich die einzelnen Systemkomponenten zu einer Linie verketteten. Sind die hintereinanderliegenden Fertigungsstationen direkt durch ein Transportsystem verbunden und laufen die Leiterplatten sequentiell durch jede Station, spricht man von einer in-line verketteten Bestückungslinie. Verkettungsmodule der Bestückungsautomaten sind:

- Transportbänder
- FIFO-Puffer
- Siebdrucker
- Ein- und Ausgabestationen
- Quertransportstationen
- Leiterplattenwendestationen
- Reflowöfen

Wichtig für die Datenmodellierung ist es, die Verkettung der Automaten, die Verkettungsmodule und die Reihenfolgenalternativen abzubilden.



*Bild 2-9: Modularer Aufbau einer Bestückungslinie*

### **3 Planungssysteme in der Elektronikfertigung**

Die Arbeitsplanung als ein integraler Bestandteil der Arbeitsvorbereitung übernimmt die Planungsmaßnahmen, die unter Berücksichtigung der Wirtschaftlichkeit die fertigungsgerechte Herstellung eines Erzeugnisses sichern [22, 4]. Die einzelnen Aufgaben der Arbeitsplanung, unabhängig von der betrachteten Sparte, sind [67]:

- Arbeitsablaufplanung
  - Erstellung neutraler Arbeitspläne
  - Erstellung auftragsbezogener Arbeitspläne
  - Kostenkalkulation
  - Varianten- und Ähnlichkeitsplanung
  - Erstellen von Standardarbeitsplänen
- Prozeßplanung
  - NC-Programmierung
  - Vorgabezeitermittlung
  - Maschinenauswahl
  - Betriebsmittelauswahl
  - Erstellung der Fertigungsdokumentation
- Stücklistenverarbeitung

Die Arbeitsplanung bezeichnet man auch als auftrags- und terminneutrale Planung. Die Arbeitssteuerung als weitere Funktion der Arbeitsvorbereitung hat demgegenüber die Aufgabe, das Produktionsprogramm in einer optimalen und termingerechten Ablauffolge einzubinden [119].

Meist liegt in der Praxis jedoch keine so starke Trennung und Gliederung der Aufgabenstellung vor. Die Aufgaben der Arbeitsvorbereitung sind je nach Organisationsform des Unternehmens verschiedenen Abteilungen zugeordnet.

Speziell in der Elektronikfertigung sind im Bereich der Arbeitsplanung eine Reihe komplexer Entscheidungen zu treffen. Der sehr innovative Elektronikmarkt erfordert stetig Änderungen und Anpassungen an neue Technologien und zu verarbeitende Bauelementetypen. Jeder neue Trend hat Auswirkungen auf das Fertigungsequipment und beeinflusst die Arbeitsvorgangsfolge. Weitere Charakteristika des Elektronikbereiches sind die Vielzahl der zu verarbeitenden Komponenten bzw. -typen, die enorme Variantenvielfalt, die stark unterschiedlichen Losgrößen und der hohe durchschnittliche monetäre Wert der Produkte [46, 58].

#### **3.1 Rechnergestützte Planungssysteme – State of the art –**

Der Einsatz rechnergestützter Planungssysteme in der Arbeitsplanung ist im Vergleich zu Konstruktion und Fertigung weniger ausgeprägt. Die meisten der vorhandenen Systeme unterstützen das Verwalten und Editieren von Arbeitsplänen sowie die Kosten- und Zeitkalkulation [110].

Speziell für die Elektronikproduktion existieren keine dedizierten Systeme, die die klassischen Aufgaben für die Arbeitsplanung (Arbeitsplangenerierung, -verwaltung, Zugriff auf ähnliche Arbeitspläne und Variantenerzeugung, Änderungsstücklisten, Varianten- und Versionsbildung etc.) unterstützen.

Anders verhält es sich im CAM-Bereich. Hier existieren eine Reihe auf die Elektronikbelange ausgerichteten Systeme, deren Leistungsspektren einen weiten Bereich überspannen. Er reicht von einfachen Visualisierungsprogrammen für diverse Datenformate bis zu komplexen CAM-Systemen, die z. B. die Aufbereitung von Foto-plotterdaten, die Erstellung von Fräs- und Bohrdaten und die Optimierung von Bestückprogrammen beinhalten. Viele der Systeme sind auf Teilbereiche spezialisiert oder bieten ausschließlich diese Funktionalität an.

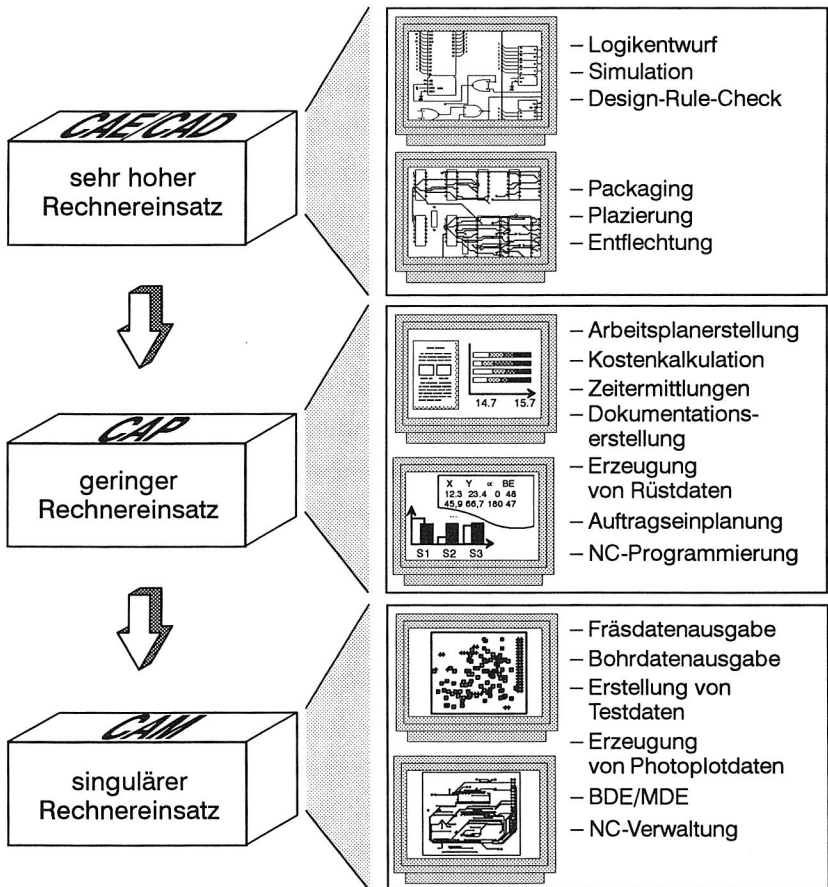


Bild 3-1: CA-Techniken in der Elektronik

So werden von einigen Maschinenherstellern Softwarelösungen zu ihren Automaten-typen angeboten [95]. Das Leistungsspektrum reicht von Bestücksequenzoptimierung und Optimierung der Feederanordnungen bis hin zum Line Balancing. Die Optimierungsmöglichkeiten sind rein maschinenspezifisch und beziehen sich meist auf einen Leiterplattentyp. Komplexere Problemstellungen, wie das Einlasten mehrerer Aufträge für eine heterogene Infrastruktur an Bestückungssystemen, werden von diesen Lösungen nicht abgedeckt.

Auch einige Leiterplattenlayoutsysteme haben CAM-Module in ihrem Leistungsumfang integriert. Neben Leiterplattendesignfunktionalitäten bestehen Möglichkeiten zur Bestückprogrammoptimierung und z. T. zum Manufacturing und Design-Rule-Check der Leiterplatten.

Weitere Lösungen im Bereich der Planung für die Elektronik stammen von kleineren bis mittleren Softwarehäusern, die z.B. singuläre Postprozessoren mit Optimierungsalgorithmen anbieten, Möglichkeiten der Visualisierung und des Datenaustauschs für verschiedene Datenformate zur Verfügung stellen oder auch für einige Teilprozesse (Fräsen, Bohren, Bestücken, Testen) Daten aus dem Entwicklungsbereich in die entsprechenden Maschinenformate transformieren [71], [74]. Weit verbreitet im Bereich der Planung, auch speziell in der Elektronikfertigung, ist die Simulationstechnik [101]. Zum Einsatz kommt die Simulationstechnik sowohl in der Bestimmung optimaler Fertigungsreihenfolgen als auch in der Auslegung von Bestückanlagen und deren Peripherie [92, 107].

Im Bereich Forschung und Entwicklung ist ein starker Trend in Richtung wissensbasierter Systeme für die Arbeitsplanung zu erkennen. Sowohl im Bereich Mechanik [110], Montage [109, 115] als auch in der Elektronikfertigung [13, 58] sind Entwicklungen im Gange, das breite und komplexe Fachwissen des Arbeitsplaners mit Hilfe von Expertensystemen abzubilden. Mit Hilfe wissensbasierter Systeme sollen die Planungsaktivitäten (Variantenplanung, Neuplanung) hinsichtlich Planungsgenauigkeit und Zeitaufwand unterstützt und verbessert werden.

Eine Gemeinsamkeit und ein wesentlicher Bestandteil der Systeme sind Produkt- und Prozeßmodelle zur Beschreibung der Werkstücke in der Mechanik und der Leiterplatten in der Elektronik. Die Produktmodelle beschreiben die Objekte auf der Basis technischer Elemente, der sogenannten features. Hiermit lassen sich neben Geometrieinformationen sämtliche technologischen Informationen, die zur Erstellung der Arbeitspläne nötig sind, abbilden. Die Prozeßmodelle beschreiben die für die Fertigung des jeweiligen Produktes anfallenden Prozeßschritte. Sie beinhalten Informationen über die einzelnen Prozeßschritte, die Abarbeitungsreihenfolgen, die Prozeßparameter und die zugeordneten Maschinen bzw. Arbeitsplätze.

Ein weiterer Forschungsschwerpunkt ist die Integration der Arbeitsplanung in das gesamte EDV-Umfeld [23, 113, 124, 116]. Die Arbeitsplanung verknüpft innerhalb eines Unternehmens den dispositiven Informationsfluß der Produktionsplanung und

-steuerung mit der technologischen Informationskette. Der Einbindung einer rechnerunterstützten Arbeitsplanung und der datentechnischen Verknüpfung der beiden Bereiche zur Gewährleistung eines durchgängigen Informationsflusses wird ein großer Stellenwert eingeräumt [53].

### **3.2 Defizite rechnergestützter Planungssysteme**

Innerhalb der CAX-Welten eines Elektronikunternehmens erfolgen die meisten Planungsaktivitäten separat. Die Spezialisten der einzelnen Abteilungen pflegen nur ihre Softwarewerkzeuge. Dies hat zur Folge, daß es oft zu einer redundanten Datenhaltung und einer geringen Nutzung des rechnerintegrierten Informationsflusses kommt. Viele der bereits im Rechner verfügbaren Daten bleiben aufgrund von Abteilungsbarrieren und Schwierigkeiten des prozeßübergreifenden Datentransfers in den Insellösungen. Es fehlt eine einheitliche Datenplattform zur Aufnahme technischer und dispositiver Daten. Für jeden einzelnen Benutzer sollte eine bereichs- bzw. abteilungsübergreifende Möglichkeit der Interaktion und Informationsgewinnung (mit unterschiedlichen Zugriffsmöglichkeiten) bestehen.

Viele CAP/CAM-Produkte sind auf singuläre Prozeßschritte der Elektronikfertigung spezialisiert. Defizite gibt es auch im Bereich der Optimierungssoftware für Bestückungsautomaten. Dedizierte Systeme, die eine herstellerunabhängige Auftrags-einlastung, Optimierung und Simulation für den kompletten Elektronikbereich vornehmen, existieren nicht. Oft wird in der Praxis die rechnergestützte Simulation als Optimierung bezeichnet. Diese Systeme erlauben zwar die vergleichende Bewertung von vorgegebenen Alternativen, generieren aber in keinem Fall automatisch situationsspezifische Vorschläge [38].

Ein weiteres, oft diskutiertes Problemfeld ist der fehlende oder mangelhafte Informationsaustausch zwischen Entwicklung und Fertigung [66, 100]. Dadurch, daß eine frühzeitige Einbindung und Berücksichtigung der Fertigungsbelange fehlt, kommt es zu kostenintensiven Fertigungsmaßnahmen und im Extremfall zum Redesign der Leiterplatte.

### **3.3 Anforderungen an ein rechnergestütztes Planungssystem für die Elektronik**

Aus den erörterten Defiziten und mehreren Analysen bei Produzenten von Flachbaugruppen wurden eine Reihe von Anforderungen zusammengestellt. Um den Anspruch eines allgemeingültigen Systems Rechnung zu tragen, sind sämtliche Prozeßschritte der Flachbaugruppenfertigung zu berücksichtigen (s. Bild 3-2). Die Daten sollten zentral verwaltet werden und allgemein verfügbar sein. Wichtig ist es, die Informationen entsprechend den Anforderungen des jeweiligen Benutzers aufzubereiten und übersichtlich grafisch zu visualisieren. In den folgenden Unterka-



piteln wird auf einige Detailanforderungen und Auswirkungen auf ein elektronikspezifisches Planungssystem näher eingegangen. Prinzipiell sollte ein Planungssystem für die Elektronik vier Aufgabenschwerpunkte abdecken [65]:

- Datenextraktion (aus verschiedenen Schnittstellen)
- Datenverwaltung
- Erzeugung von Fertigungsunterlagen und Arbeitsplänen
- Steuerung der Kommunikation und Information

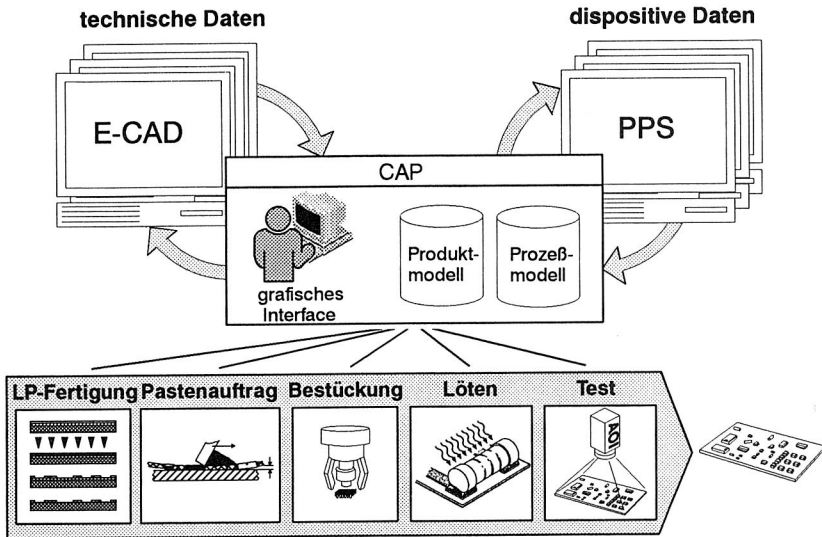


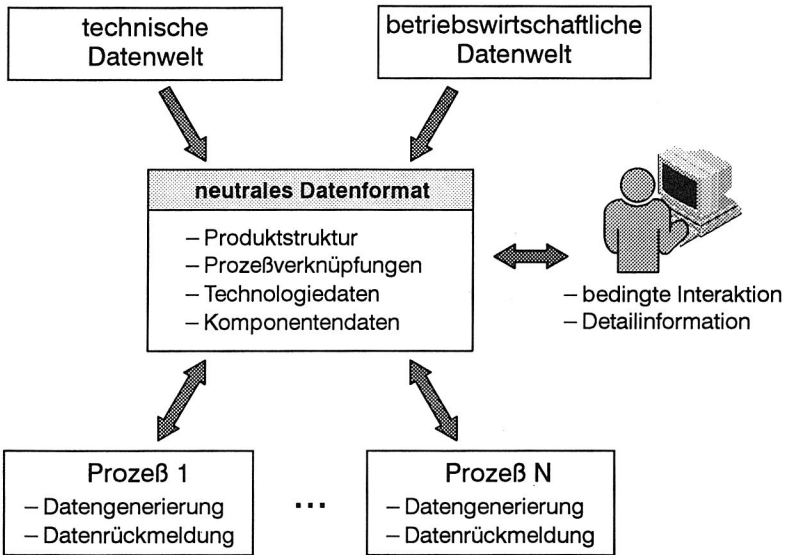
Bild 3-2: CAP-System für sämtliche Prozeßschritte in der Elektronikfertigung

### 3.3.1 Datenhaltung und adaptiver Output

Innerhalb des Produktionsprozesses einer Flachbaugruppe ist eine einheitliche, allgemein zugängliche Datenbasis anzustreben. In dem neutralen Datenformat sind Informationen zu sämtlichen Prozeßschritten strukturiert abzulegen. Den Anwendern in den einzelnen Prozeßstufen soll somit eine entsprechende Sicht auf die Daten zur Informationsgewinnung und zur Interaktion ermöglicht werden. So kann z.B. der Maschinenbediener mit einer grafischen Visualisierung der Leiterplatte oder Rüstung vor Ort schnell und einfach die für ihn relevanten Detailinformationen abrufen.

Die Datenhaltung muß mittels klar definierter Schnittstellen Import- und Exportfunktionalitäten sowohl zu den Entwicklungswerkzeugen und zu den Systemen der betriebswirtschaftlichen, planerischen Datenwelt als auch zur kompletten Prozeßwelt beinhalten. Der Datenoutput soll flexibel und einfach an unterschiedliche Belange

des Unternehmens anpaßbar sein. Gerade die Verknüpfung der betriebswirtschaftlichen mit den technischen Daten und die Möglichkeiten, sämtliche Systeme der Prozessebene (Leiterplattenfertigung, Pasten-/Kleberauftrag, Bestücken, Löten, Testen) aus einer einheitlichen Datenquelle zu speisen, bergen große Rationalisierungspotentiale.



*Bild 3-3: Neutrales Datenformat als Kern für ein Planungssystem*

### 3.3.2 Rüstproblematik und Rüstanforderungen

Die Rüstzeitverkürzung ist sowohl im Bereich Mechanik als auch in der Elektronik ein ständig aktuelles Thema. Technisch anspruchsvolle und teure Maschinensysteme sind nur dann wirtschaftlich einsetzbar, wenn unproduktive Nebenzeiten möglichst gering ausfallen. Flexible Produktion und geringe Losgrößen erfordern eine Reduzierung der Rüstzeitanteile, um weiterhin wettbewerbsfähig produzieren zu können. Das Rüsten selbst ist nach REFA [89] folgendermaßen definiert:

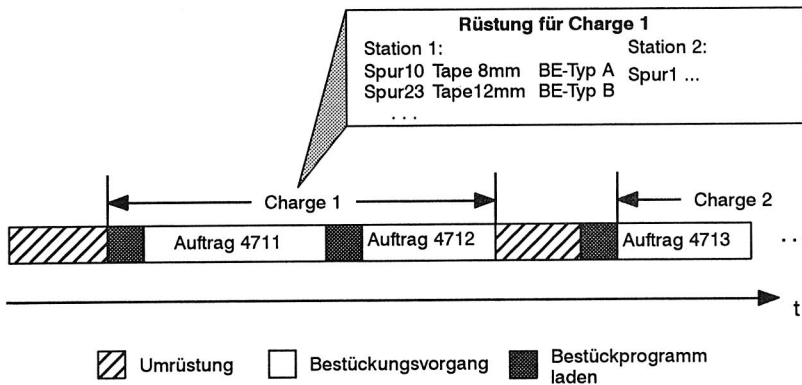
"Rüsten ist das Vorbereiten des Arbeitssystems für die Erfüllung der Arbeitsaufgabe sowie – soweit erforderlich – das Rückversetzen des Arbeitssystems in den ursprünglichen Zustand. Das Rüsten kommt im allgemeinen einmal je Auftrag vor"

Diese Definition beschreibt den Rüstprozeß an der Maschine. Innerhalb dieser Arbeit, fokussiert auf den Elektronikbereich, wird im folgenden zwischen "Rüsten"

und "Umrüsten" und somit auch zwischen Rüstopтимierung und Umrüstopтимierung unterschieden.

Unter Rüstung versteht man die genaue Vorgabe (Bauelement, Förderer, Zuführspur) für ein Bestückungssystem, mit der eine komplette Charge ohne eine einzige Veränderung des Rüststandes bearbeitet werden kann (s. Bild 3-4). Als Charge wird die Zusammenfassung mehrerer Aufträge (die kleinste Charge besteht aus einem Auftrag) bezeichnet.

Das Umrüsten bezeichnet die Änderung eines Rüststandes für ein Bestückungssystem bei Chargenwechsel. Das Umrüsten ist ein zeitlicher Vorgang.



*Bild 3-4: Zeitlicher Ablauf des Umrüstens und Bestückens*

Im Mechanikbereich wurden zum Thema Rüstproblematik und -analyse bereits eine Reihe von Untersuchungen durchgeführt [123, 120], wobei sich einige Parallelen zur Elektronik ziehen lassen (s. Bild 3-5). Neben den Unterteilungen in organisatorische, personalbezogene und technische Maßnahmen lassen sich prinzipiell drei Schwerpunkte der Rüstzeitreduzierung unterscheiden [33]:

- rüstzeitverkürzende,
- rüstzeitverlagernde und
- rüstzeitvermeidende Konzepte.

Unter rüstzeitverkürzenden Konzepten versteht man Maßnahmen direkt am Arbeitsplatz wie z.B. Schnellspannsysteme [122].

Bei der Rüstzeitverlagerung versucht man parallel zur Maschinenhauptzeit für den nächsten Arbeitsgang zu rüsten. Flexible Fertigungssysteme, Wechselpalettenssysteme oder, im Bereich Elektronik, externe Rüstwagen sind Beispiele hierzu.

Das Konzept der Rüstzeitvermeidung konzentriert sich hauptsächlich auf das Werkstück. Rüstgerechte Konstruktion und Komplettbearbeitung sind hier die entscheidenden Faktoren.

Ein weiterer wichtiger Aspekt, neben der Rüstzeitreduzierung ist die ermittelte Rüstung an der Maschine selbst. Durch optimales Einrüsten der Maschine bzw. der Systeme, mit Hilfe von Werkzeugbereitstellung oder der Anordnung der Bauelementezuführungen, aber auch durch optimierte NC- bzw. Bestückprogramme lassen sich zusätzlich erhebliche Einsparungspotentiale erreichen. Strategien zum optimalen Einrüsten der Maschinen erzielen im Gegensatz zu den vorgestellten Konzepten zur Verringerung der Maschinennebenzeiten eine Reduzierung der Maschinenhauptzeit.

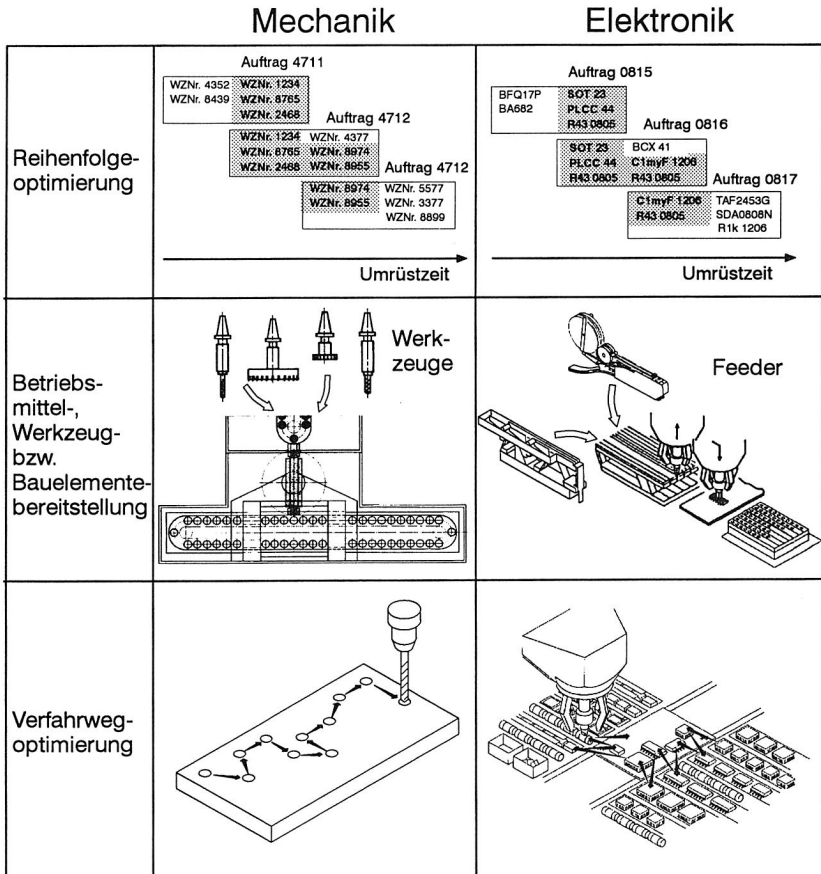


Bild 3-5: Beispiele zur Rüstproblematik für Mechanik und Elektronik

Ein wesentlicher Aspekt neben dem globalen Optimierungsziel Reduzierung der Auftragsdurchlaufzeiten ist die Berücksichtigung von Anforderungen und Restriktionen

der Anwender. Nach Analysen bei einigen Flachbaugruppenbestückern (sowohl bei Auftragsfertigern wie auch bei Massenproduzenten) wurden einige Anforderungen zur Rüstproblematik speziell für die Elektronikfertigung aufgenommen:

### **Berücksichtigung von Festrüstanteilen**

Der Wunsch, alle im Produktionsbereich aktuellen Flachbaugruppentypen mit einer fixen Rüstung zu produzieren, ist meist aufgrund der Komplexität der Leiterplatte und der großen Variantenvielfalt der Bauelemente nicht möglich. Besonders häufig vorkommende Komponententypen oder Komponenten mit hoher Umrüstzeit sollten jedoch auf bestimmten Rüstplätzen fixiert werden können.

### **Vorgabe von Mehrfachrüstungen**

Für Bauelementtypen mit hoher Bestückhäufigkeit pro Auftrag sollte es möglich sein, diese mehrfach auf den Bestückungssystemen zu rüsten. In diesem Zusammenhang ist auch die Bauelementedominanz zu beachten. Besitzt ein Bauelementtyp für einen Auftrag einen höheren Bestückanteil als sämtliche anderen Typen dieses Auftrags, so ist dieser Bauelementtyp dominant. Diese Komponenten sollten erkannt und automatisch eine Mehrfachrüstung generiert werden.

### **Ausschlüsse und Bindungen**

Neben technologischen Ausschlüssen oder Bindungen von Bauelementen, z. B. in Folge benötigter Zusatzeinrichtungen (Zentriersysteme etc.), sind auch Restriktionen dieser Art vom Benutzer zu beachten. Maßnahmen hierfür sind etwa das Plazieren häufig nachzurüstender Zuführsysteme an gut zugänglichen Stellen des Bestückungssystems, oder der Ausschluß von Bauelementen an bestimmten Automatentypen, da diese dort aus Erfahrungswissen häufig Bestückfehler verursachen.

### **Berücksichtigung von Ausgangsrüstständen**

Rüststände, die bei Beginn einer Neu- oder Umplanung der Auftragseinlastung vorliegen, sollen in die Planungsmethode mit eingehen.

### **Ausgabe von Bearbeitungs- und Umrüstzeiten**

Detaillierte Abarbeitungs- und Umrüstzeiten sollen zur Kostenkalkulation der Aufträge ausgegeben werden.

### **Kalibrierung der Automatentypen**

Da sich baugleiche Automatentypen in ihrer Bestückleistung oder Verfügbarkeit durchaus unterscheiden, soll jeder Automat als Individuum betrachtet werden und benutzergesteuert kalibrierbar sein. Der Anwender hat somit die Möglichkeit, entsprechend seiner Anforderungen die Lastverteilung innerhalb der Linie zu steuern.

### 3.3.3 Optimierungsfunktionalitäten

Bei der Entwicklung von Optimierungsstrategien gehen eine Reihe von Faktoren ein, welche die Generierung eines Optimums erschweren. Neben den primären Zielfunktionen Reduzierung der Rüst- und Bestückzeit an den Automaten und hoher Maschinenauslastung spielen auch Benutzer und Systemanforderungen eine gewichtige Rolle.

Allein durch die große Vielzahl an Systemen im Bereich der Bestückungstechnik ist ein hoher Grad an Variationsmöglichkeiten gegeben. So beleben diesen Markt eine Reihe von Herstellern, die zum Teil recht unterschiedliche Automatenkonzepte verfolgen und für ihre Maschinen eine Reihe von Zusatzmodulen und Peripherieeinheiten anbieten.

Aus den analysierten Teilproblemen und Benutzeranforderungen lassen sich für den Optimierungsbereich folgende Zielkriterien zusammenfassen:

- Senkung der Maschinennebenzeiten durch kostenoptimale Zuteilung der einzuplanenden Aufträge auf die vorhandenen Bestückungssysteme. Berücksichtigung aller technologischen Randbedingungen, der Benutzerausschlüsse und des momentanen Auslastungsgrades bzw. der Rüststände der Bestückungssysteme.
- Errechnung einer umrüstopimalen Abarbeitungsreihenfolge der Aufträge und Minimierung der Chargen durch Zusammenfassen von Aufträgen. Beachtung einer vorgegebenen Prioritätsstaffelung, der Ausgangsrüststände, maximaler Belegungskapazitäten und benutzer- bzw. maschinenabhängiger Umrüstzeiten. Ausgabe von Umrüstanweisungen für das Rüstpersonal.
- Reduzierung der Maschinenhauptzeiten durch eine gleichmäßige Austaktung der einzelnen Automaten innerhalb der Bestückungslinien für die generierten Chargen.
- Minimierung der Bestückzeiten für die unterschiedlichen Automatentypen durch Generierung einer optimalen Feederanordnung und einer reihenfolgeoptimierten Bestücksequenz.
- Bestimmung von optimierten Festrüstanteilen für ein vorgegebenes und gewichtetes Spektrum an aktuellen Leiterplattentypen.
- Durch Benutzervorgaben gesteuerte Betonung des Optimierungsziels, z.B. Optimierung für Variantenfertigung oder Massenfertigung.

### 3.3.4 Kostenkalkulation und Zeitabschätzungen

Wichtige Steuergrößen der Fertigung sind die Kenngrößen Durchlaufzeiten und Termintreue. Die grobe, bereichsübergreifende Planung zur Ein- und Ausgangssteuerung der Aufträge übernehmen in den meisten Unternehmen PPS-Systeme. Die Vorteile bereichsbezogener Zeiten- und Terminplanung sind der hohe Detaillierungsgrad, die Möglichkeiten zur Simulation und Umplanung in sektoralen Fertigungsbereichen und das in den einzelnen Bereichen vorhandene Know-how zur Optimierung der Einplanung. Im Idealfall kann in einem Regelkreis durch ein Feedback der feinterminierten Auftragsdaten eine Verbesserung der Planungsvorgaben des PPS-Systems erzielt werden.

Weitverbreitete Fragestellungen im Bereich der Auftragseinplanung sind:

- Wieviel Zeit benötigt ein bestimmter Auftrag auf den unterschiedlichen Bestückungssystemen?
- Wann ist ein Auftrag spätestens einzuplanen? (Rückwärtsterminierung)
- Welche Termine ergeben sich bei einer bestimmten Einplanung oder Umplanung?
- Welche Terminverschiebungen ergeben sich beim Einschieben eines Eilauftrages oder Ausfall einer Maschine?

Um diesen oder ähnlichen Anforderungen gerecht zu werden, sind als Grundlage verschiedene Zeitmodelle für die Systeme einer Flachbaugruppenfertigung zu entwerfen.

Ein weiterer Planungsfaktor ist die Bestimmung der zu erwartenden Kosten. Die Herstellkosten einer Leiterplatte setzen sich aus den Material- und Fertigungskosten zusammen [17]. Die Fertigungskosten werden wiederum aus den Fertigungseinzel-, Fertigungsgemeinkosten und Sondereinzelkosten der Fertigung gebildet (s. Bild 3-6).

In der Elektronikfertigung ist diese Unterscheidung stark vom Produktprogramm, von der Losgröße und der Variantenhäufigkeit abhängig. Bei einer Massenfertigung mit geringer Variantenvielfalt werden meistens die gesamten Fertigungskosten nur über die Fertigungsgemeinkosten, die den Maschinenstundensatz darstellen, kalkuliert. Bei einem sehr vielseitigen Produktprogramm, stark unterschiedlichen Auftragsmengen von Leiterplatten und einer sehr hohen Vielfalt von Varianten hingegen erfolgt eine Trennung in Fertigungseinzelkosten. Diese präsentieren sich in Form von Fertigungslöhnen gemäß der Fertigungszeit.

Um nun eine monetäre Leistungszuordnung zu einzelnen Aufträgen vornehmen zu können, ist eine prozeß- und tätigkeitsbezogene Kostenzuteilung unabdingbar. Um der Anforderung einer Kostenkalkulation im Bereich der Fertigungskosten gerecht zu werden, ist eine Gliederung des Produktes, analog des zu Grunde liegenden Prozeß- und Produktmodells, in einzelne Prozeßschritte erforderlich. Addiert man verursachungsgerecht die einzelnen Teilprozesse mit Hilfe der Zeitermittlungen und z.B.

der Prozeßkostenrechnung bzw. dem Maschinenstundensatz, so erhält man eine Kalkulationsgrundlage für die Fertigungskosten [45].

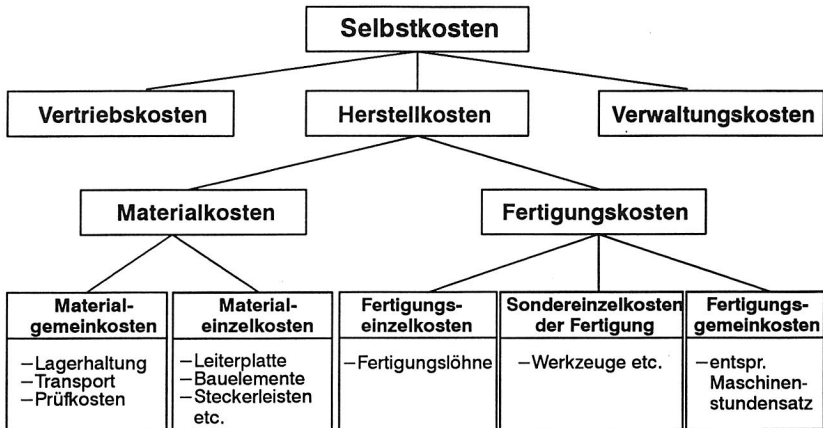


Bild 3-6: Kostenstruktur am Beispiel einer Flachbaugruppe [17]

### 3.3.5 Aufbau und Integration von Regelkreisen

Ein großes Problem vieler Unternehmen ist die fehlende Kommunikation zwischen den einzelnen Bereichen, insbesondere der Entwicklung und der Fertigung. Oft ist zusätzlich eine regionale Trennung der verschiedenen Abteilungen vorhanden. Hauptaugenmerk des Entwicklungsingenieurs ist der Entwurf einer funktionsorientierten Schaltung. Fertigungs-, test- und montagegerechte Gesichtspunkte werden oft außer acht gelassen. Aus dieser Vorgehensweise resultieren kostenintensive Fertigungsmaßnahmen, und im Extremfall ist ein Redesign der Leiterplatte erforderlich.

Ziel ist es, ein System im Unternehmen zu etablieren, das die Verknüpfung der Entwicklung mit den Belangen der einzelnen Teilprozesse ermöglicht. Der Entwickler sollte beim Design unterstützt werden, die Unmenge an erforderlichen Kenntnissen aus den Fertigungsprozessen zu berücksichtigen. Im Zeitalter des Concurrent Engineering sollte ein Design-for-Manufacturing-System den interdisziplinären Informationsaustausch nicht ersetzen, sondern als zusätzlicher Quercheck fungieren und den Designer von einer Reihe von Routineüberprüfungen entlasten.

Eine weitere Möglichkeit eines effizienten Feedbacks ist das Zusammenspiel von Feinterminierung und Kapazitätsplanung mit übergeordneten Steuerungssystemen (PPS).



### 3.3.6 Fertigungstransparenz und Informationsdarstellung

Der Wunsch vieler Fertigungsunternehmen ist es, ein umfassendes Informationssystem zwischen Produktionsplanung und Fertigung aufzubauen. Erst die grafische Darstellung komplexer Sachverhalte und die bereichsübergreifende Verfügbarkeit von Informationen ergeben die gewünschte Transparenz der Produktionsabläufe und ermöglichen eine optimale Auftragssteuerung. Fertigungsleitstandssysteme als Bindeglied der Produktionsplanung und der eigentlichen Produktion sollen diese Aufgabe erfüllen [83]. Im Planungsbereich existieren Anforderungen wie die Simulation bestimmter Fertigungsabläufe, Unterstützung der Disposition, Erkennung von Engpässen etc. Im Controllingbereich ermöglichen BDE/MDE-Kopplungen die Zustandsüberwachung der Prozesse, frühzeitige Fehlererkennung mit evtl. abgeleiteten Fehlerdiagnosen oder die Erstellung von Qualitätsstatistiken.

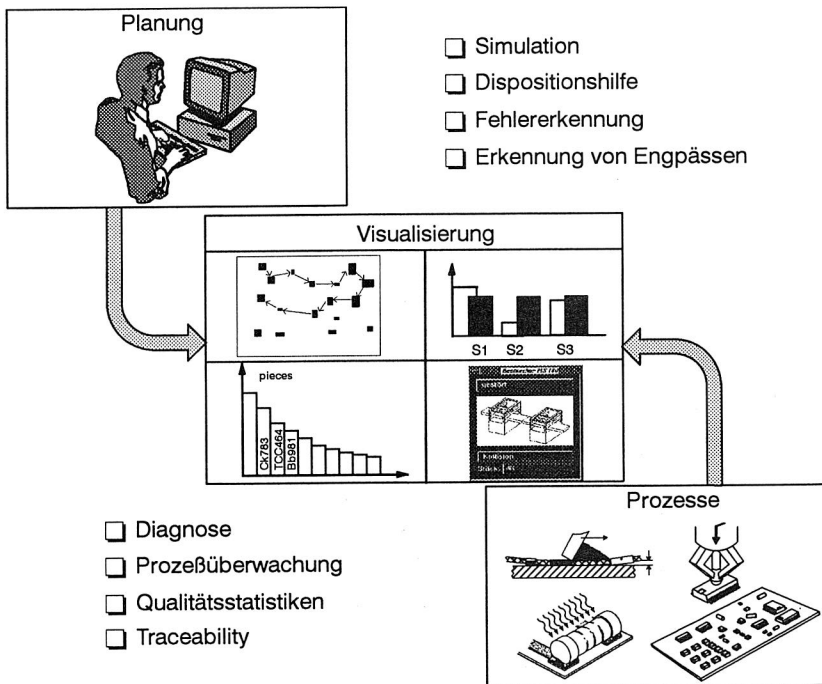


Bild 3-7: Grafische Informationsdarstellung als Planungs- und Controllingwerkzeug

#### 4 Konzept eines modularen Planungssystems für die Elektronikfertigung

Die Gesamtstruktur des entwickelten Softwaresystems zur Unterstützung der verschiedenen Planungsaktivitäten in der Elektronikfertigung ist in Bild 4-1 dargestellt. Das Konzept basiert auf einem modularen Ansatz mit einer zentralen Datenbasis als Kern. Die funktional voneinander unabhängigen Module beziehen ihre Informationen aus diesen neutralen Datenkern und legen dort ihre Ergebnisse ab, so daß sie allgemeingültig zur Verfügung stehen [27]. Die zentrale Datenbasis ist so strukturiert, daß sie Informationen über alle Prozessschritte aufnimmt. Begünstigt durch den modularen Aufbau sind jederzeit prozeßspezifische Erweiterungsmodule ohne Einfluß auf das Gesamtsystem realisierbar. Teilweise decken die entwickelten Modulen (Exportfilter, DFM-System) auch prozeßübergreifende Funktionalitäten ab.

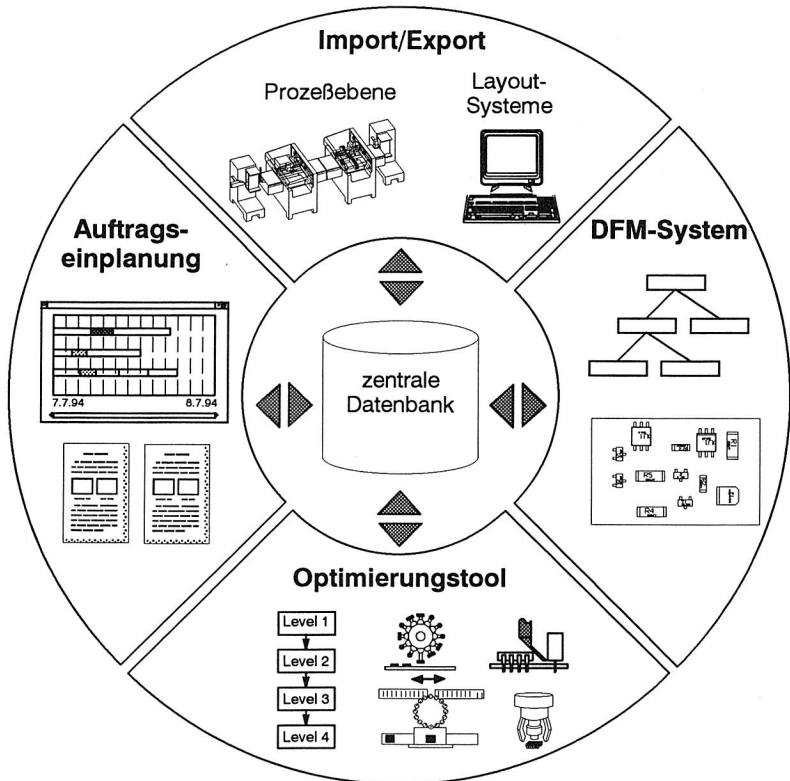


Bild 4-1: Struktur des Planungssystems

Die zentrale Datenbasis enthält sämtliche Daten zur Beschreibung der einzelnen Produkte (Flachbaugruppen) sowie Informationen über die Prozesse, das technische Equipment und die Bauelemente. Die Datenbasis hat die Aufgabe, sämtliche erforderlichen Parameter der Produkte und Prozesse aufzunehmen, zu verwalten und den verschiedenen Modulen zur Verfügung zu stellen.

Das Import-/Exportmodul nimmt die erforderlichen technischen Daten aus dem Entwicklungsbereich, d.h. aus den Layoutsystemen, über ein Standardformat oder spezifische Formatanpassungen auf. Weitere Daten sind aus den dispositiven Bereichen zu entnehmen. Mit den zur Verfügung stehenden Editoren ist auch ein interaktiver Input des Benutzer möglich. Die Exportkomponente überspielt die erforderlichen Maschinendaten in dem entsprechenden Format an die Prozeßebene direkt in den Maschinenrechner. Ebenso können auch in der Gegenrichtung Daten aus der Maschine in die Planungskomponente übernommen werden.

Das DFM-Modul dient zur fertigungsspezifischen Diagnose der Leiterplattenlayouts. Zur Unterstützung des Entwicklungsingenieurs überprüft es verschiedene Layouts nach festlegbaren Regeln und bewertet diese.

Die optimale Einplanung der Aufträge und die Minimierung der Rüst- und Bestückzeiten erfolgt mit dem Optimierungsmodul. In einem hierarchischen Modell werden mehrere Optimierungsschichten durchlaufen und verschiedene Optimierungsziele verfolgt.

Das Modul zur Arbeitsplanung ermöglicht mittels der Integration eines Werkskalenders und verschiedener Kostenfunktionen eine detaillierte Terminierung der Aufträge, Zeitsimulationen und eine Abschätzung der Fertigungskosten. Zudem können für die Arbeitsplanung relevante Dokumentationen erstellt und benötigte Informationen zu Planungs- oder Prozeßdaten abgerufen werden.

Die Funktionsweisen und verfolgten Ziele der einzelnen Module werden in späteren Kapiteln detailliert erläutert.

#### **4.1 Entwurf einer geeigneten Softwarearchitektur**

Die Realisierung der gesamten Planungskomponente erfolgte mit den Standardwerkzeugen Ingres (Datenbanksystem), C bzw. C++ (Programmiersprachen), PROKAPPA (Expertensystemshell) und X11 Motif (Benutzerdialog) [11, 72, 73]. Einen Überblick über die Softwarearchitektur und das Zusammenwirken der einzelnen Werkzeuge gibt Bild 4-2.

Zur Verarbeitung der großen Datenmengen und als neutraler Datenspeicher zu externen Anwendungen wurde eine relationale Datenbank gewählt. Diese bietet die Vorteile, die große Menge an zu verwaltenden Informationen unkompliziert und fehler tolerant zu pflegen und vor allem konsistent zu halten. Zudem erlaubt sie durch die

Standarddatenbanksprache SQL (Structured Query Language) einen universellen Einsatz des Systems auf verschiedenen Hardwareplattformen und ein schnelles Anpassen der Datenbankstrukturen an die betriebspezifischen Gegebenheiten.

Die Gestaltung des grafisch orientierten und benutzerfreundlichen Dialogs wurde in X11 realisiert. Der Anwenderdialog erfolgt zum größten Teil direkt mit der Datenbank, in einigen Fällen fungiert eine Zwischenschicht als temporäre Speicherstruktur.

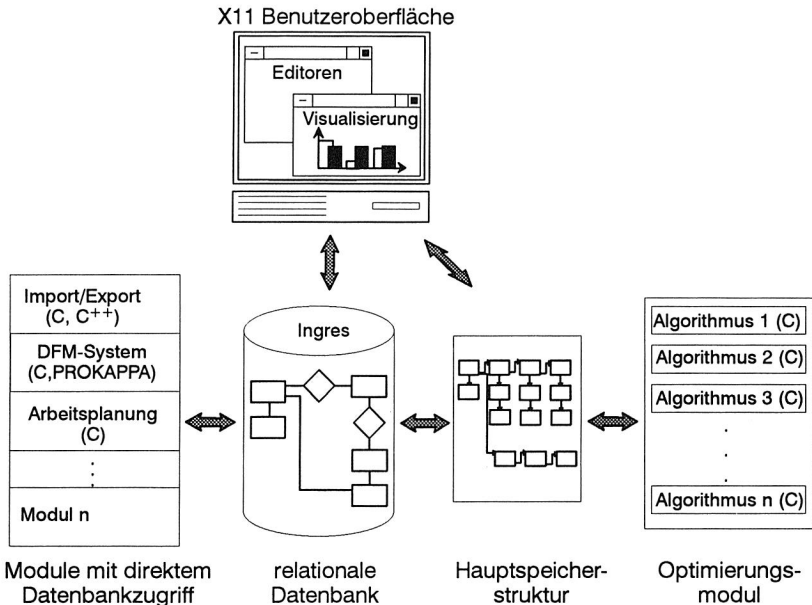


Bild 4-2: Softwarearchitektur

Bedingt durch die komplexen Optimierungsvorgänge und die damit verbundenen häufigen Datenzugriffe wurde für das Optimierungsmodul zwischen Datenbank und den Optimierungsalgorithmen eine Zwischenschicht in Form eines sogenannten Blackboards erstellt [21, 34]. Diese für mehrere Subsysteme gemeinsam zugängliche Datenstruktur wird aus Performancegründen im Hauptspeicher des Rechners gehalten. Sie wird zu Beginn eines Optimierungslaufes mit den relevanten Daten aus der Datenbank gefüllt. Am Ende des Optimierungslaufes werden die Ergebnisse aus der Hauptspeicherstruktur wieder in die Datenbank übertragen. Die aktivierten Optimierungsalgorithmen (Subsysteme) greifen somit nur auf die gemeinsame Hauptspeicherstruktur zu und können Teilergebnisse austauschen sowie Endergebnisse eintragen.

## **4.2 Datenmodell zur Planungskomponente**

Zum Entwurf der zentralen Datenbank war eine systematische Strukturierung der betrachteten Produkte, der Prozesse und Subprozesse der Elektronikfertigung erforderlich. Es waren Datenmodelle zu konzipieren, um die geometrische Eigenschaften, Prozeßdaten und auch Prozeßinterdependenzen der gesamten Prozeßkette beschreiben zu können. Der komplette Entwurf des zentralen Datenbanksystems erfolgte in drei Phasen:

- Phase 1: Datenanalyse, Strukturierung der Daten, Aufbau der Modelle
- Phase 2: Umsetzung in ein Entitäten-Beziehungsmodell
- Phase 3: Überführung des Entitäten-Beziehungsmodells in ein relationales Datenbankschema

### **4.2.1 Strukturierung der Daten**

In der ersten Phase der Datenanalyse wurden sämtliche Anforderungen aufgenommen, die Querbeziehungen herausgearbeitet und zu Datenmodellen zusammengefaßt. Das Gros der Daten stammt aus den benötigten Parametern des Optimierungsmoduls und den zu verarbeitenden Benutzeranforderungen. Weitere Informationen waren aus den Schnittstellen zu externen Systemen zu extrahieren. Aus der analysierten und strukturierten Datenmenge wurde folgende Gliederung der Daten vorgenommen:

#### **Geometrische Daten**

Hiermit werden alle geometrischen Daten der Flachbaugruppen und ihrer Komponenten beschrieben, die während des Planungsvorgangs relevant sind.

#### **Produktconstraints**

Diese Datenklasse dient dazu, die Verknüpfungen der Produkte und der Komponenten abzubilden.

#### **Technologische Daten**

Hierunter versteht man eine Sammlung sämtlicher technologischer und materialtechnischer Eigenschaften der Produkte und aller Einzelkomponenten.

#### **Steuerdaten**

Diese Datenmenge beinhaltet die vom Benutzer gestellten Randbedingungen und Vorgaben (Regeln) sowie Steuerungsdaten, die den Ablauf der Planungsaktivitäten bestimmen.

#### **Systemdaten**

Unter den Systemdaten sind die technischen Parameter des gesamten technischen Equipments und die Verknüpfungen der einzelnen Module gespeichert.

### Prozeßparameter

Diese Datenmenge kennzeichnet die Funktion und die einstellbaren bzw. gegebenen Parameter der einzelnen Prozesse bezogen auf das verwendete Equipment und die vorliegenden technischen Restriktionen.

### Prozeßverknüpfungen

Mit den Prozeßverknüpfungen werden die Interdependenzen, die Reihenfolgen und eventuelle Alternativen der einzelnen Prozeßschritte beschrieben.

### Ergebnisdaten

Unter den Ergebnisdaten sind sämtliche Zwischen- und Endresultate der Planungsmodule zusammengefaßt.

Aus der strukturierten und gegliederten Datenmenge ergaben sich als Datenbasis ein verzahntes Produkt- und Prozeßmodell (s. Bild 4-3). Diese sind wiederum aus einer Reihe von Teilmodellen zusammengesetzt.

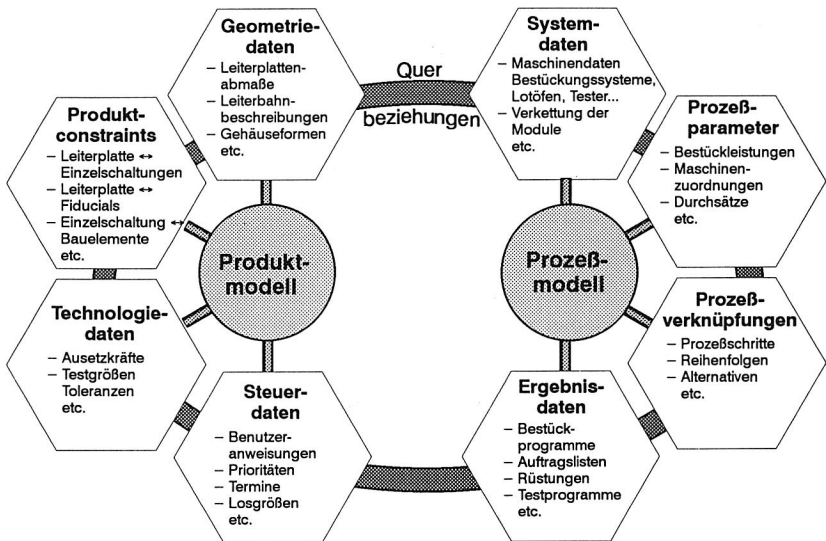


Bild 4-3: Zentrales Datenmodell der Planungskomponente

### 4.2.2 Entwicklung eines Datenbankkonzeptes

In der zweiten Phase wurde das erstellte Datenmodell umgesetzt in ein sogenanntes Entitäten-Beziehungsmodell (entity relationship model), abgekürzt auch ER-Modell genannt. Dieses Modell wurde im Jahre 1976 von P. Chen vorgestellt und hat sich in letzter Zeit im Software-Engineering als Standard etabliert [15]. Es ist ein semanti-

ches Modell, d. h. ein Beziehungen darstellendes Modell. Das ER-Modell basiert auf der Methode, Mengen und Beziehungen in Form eines bipartiten Graphen darzustellen [108], dessen Kanten mit den Beziehungstypen beschriftet werden. Rechtecke symbolisieren hierbei die Entitätsklassen und Rhomben die Beziehungsmengen (s. Bild 4-4).

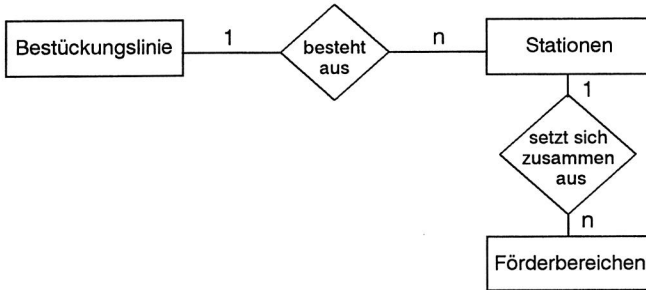


Bild 4-4: Beispiel ER - Diagramm für Linien und Stationen

Einen Auszug für das zentrale Datenmodell gibt Bild 4-5.

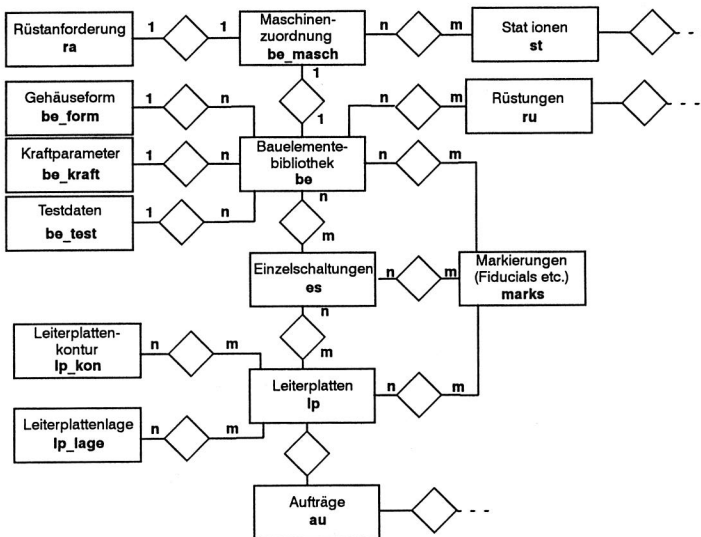


Bild 4-5: Ausschnitt der ER-Graphen zur Datenbank

Im dritten Schritt erfolgte die Überführung der ER-Graphen in ein relationales DB-Schema. Die Entitäts- und Beziehungsmengen des ER-Diagrammes wurden jeweils in Tabellen übersetzt. Dies erfordert in erster Linie das Festlegen von Primärschlüsseln für die Entitätsmengen. Diese gehen dann als Fremdschlüssel in die Tabellen ein, die aus den mit ihnen verbundenen Beziehungsmengen erstellt werden [69]. Mit

der Programmierschnittstelle ESQL, die die verwendete Datenbank Ingres zur Verfügung stellt, wurden anschließend die Anwenderprogramme realisiert.

#### 4.2.3 Entwicklung von Anwendereditoren

Zur Generierung der Anwendereditoren wurde X11 Motif verwendet. Soweit wie möglich wurde dabei auf die Einhaltung von Richtlinien zur ergonomischen Benutzergestaltung geachtet. Durch einen einheitlichen Aufbau und dem damit verbundenen "Look and Feel"-Prinzip, sollte es dem Benutzer möglich sein, die Editoren in kürzester Zeit zu bedienen. Die visuelle Gestaltung der Bildschirmmasken spielt in diesem Zusammenhang eine große Rolle [5]. Sie sollte vier Informationsklassen enthalten:

- Statusinformationen
- Arbeitsinformationen
- Steuerinformationen und
- Meldungen

Eine Statusinformation ist beispielsweise die Modusangabe (Lesen, Ändern, Löschen etc.). Unter Arbeitsinformationen versteht man die eigentlichen Daten, die eingegeben, modifiziert oder geändert werden können. Menüleisten bzw. Buttons entsprechen den Steuerinformationen und unter Meldungen werden Hinweise, Warn- und Fehlermeldungen zusammengefaßt.

Entsprechend den Anforderungen wurden zwei verschiedene Typen von Editoren entwickelt:

**Systemeditoren**, die es ermöglichen, mit Kenntnis der komplexen Datenbankstruktur auf jede einzelne Relation und somit auf jedes einzelne Attribut zuzugreifen. Diese Eingabeschnittstelle ist notwendig zur Eingabe von Testdaten, zu Reparaturzwecken und für Systemdaten, die dem Anwender verborgen bleiben.

**Anwendereditoren**, die eine komfortable und strukturierte Datenpflege gewährleisten. Diese Editoren beziehen sich auf logisch zusammengehörige Relationen. Der Benutzer benötigt keinerlei Kenntnis über die interne Datenbankstruktur. So sind z. B. alle relevanten Bauelementedaten von einer Maske aus zu bedienen, der interne Zugriff auf sechs bis acht verschiedene Relationen bleibt verborgen.

Durch hinterlegte Integritätsbedingungen der einzelnen Eingabemasken (Frames) wird die wichtige Konsistenz der Daten sichergestellt. Syntaktische Fehleingaben gerade bei Attributen, deren Werte in anderen Relationen definiert sind, wurden weitgehend durch die Verwendung von Auswahltabellen oder Auswahlfeldern vermieden. Entscheidend ist auch die Konsistenzgewährleistung über Relationsgrenzen hinweg. Wird z.B. ein neues Bauelement eingegeben, muß der Eintrag mindestens eines Maschinentyps sichergestellt sein. Beim Löschen eines Förderertyp, muß das System sämtliche Bauelemente, die diesen als Defaultförderer hinterlegt haben, ak-



tualisieren. Für die Sicherstellung der semantischen Integrität werden entsprechende Eingabedaten mit technisch oder physikalisch sinnvollen Grenzen verglichen.

**Steuerdaten - Editor**

Optimierungs-Namen:

Bemerkung:

Genauigkeit:

Leiterplattengrösse berücksichtigen: ☐ ja ☐ nein

maximale Optimierungszeit (s):

betroffene Aufträge:

- Auftrag1
- Auftrag2
- Auftrag4
- Auftrag4711
- Auftrag4712
- Auftrag4713
- Auftrag4714
- Auftrag4715
- Auftrag5
- Auftrag5711
- Auftrag5712

Auftrag löschen

betroffene Linien:

- Linie1
- Linie2
- Linie4711

Linie löschen

Optimierungsläufe

Linien-Name	Restlast	Datum
Linie2	0	25.10.93

Optimierungslauf löschen

Ausgangsruelstände

Linien-Name	Ruecklage-Name
Linie2	Linie2_fert

Ausgangsruelstand löschen

vorhandene Aufträge:

- Auftrag0
- Auftrag3

Auftrag löschen

vorhandene Linien:

- Linie3
- Linie4
- Linie4712
- Linie5
- Linie6

Linie löschen

Speichern

Cancel

SCHLIESSEN

Bild 4-6: Steuerdateneditor

Als Beispiele des Aufbaus der Editoren und der Bedienung sind der Steuerdateneditor des Optimierungstools (s. Bild 4-6) und der Editor für Bestückungsmaschinen (s. Bild 4-7) abgebildet. Mit Hilfe des Steuerdateneditors können mehrere Steuerdateien generiert werden. Eine Steuerdatei enthält sämtliche Informationen zur Durchführung einer Optimierung, einzeln interaktiv oder für mehrere im Batchbetrieb. Aus den einzuplanenden Aufträgen und den zur Verfügung stehenden Bestückungslinien wird über Mausklick der zu optimierende Auftragspool zusammengestellt. Weitere Eingabemöglichkeiten zu den einzelnen Optimierungsalgorithmen sind relevante Zeitschranken oder Gütevorgaben. Optional editierbar sind Restlasten der Linien, zu berücksichtigende Ausgangsrüststände und über eine weitere Eingabemaske Anwenderrestriktionen.

Bild 4-7: Editor für Maschinendaten (SMD- Bestückungsautomat)

Die Eingabemasken des Stationseditors sind an die unterschiedlichen Automatentypen entsprechend der Bestückungsklassen angepaßt. Editierbar sind sämtliche erforderliche Maschinenparameter zu den modellierbaren Automatentypen.

### 4.3 Entwicklung einer Hauptspeicherstruktur für schnelle Datenzugriffe

Speziell zur Durchführung der Optimierungsläufe wurde zur Beschleunigung des Datenzugriffs eine Hauptspeicherstruktur (Blackboard) entwickelt, die die notwendigen Informationen über die Fertigungssysteme beinhaltet. Ziel war es, eine allgemeingültige Struktur zu entwerfen, die es ermöglicht, alle Maschinentypen, die zugehörigen Förderertypen, die Zusatzkomponenten, Auftrags- und Leiterplattendaten, die Benutzeranforderungen etc. abzubilden. So wird z.B. vor der Durchführung eines Optimierungslaufes entsprechend den Vorgaben der Steuerdatei das Blackboard mit den Parametern aus der Datenbank gefüllt (s. Bild 4-8).

Der Ausgangspunkt für alle Optimierungen ist das Hauptspeicherabbild der Steuerdatei. Von dieser Basisstruktur aus lassen sich alle weiteren Daten, die für einen spe-

ziellen Optimierungslauf nötig sind, erschließen. Ermittelt wird z.B. die zu optimierende Linie/Station, welche Restriktionen bzw. Ausgangsrüstungen vorliegen und welche Zeit bzw. Güteschranken die Optimierung hat.

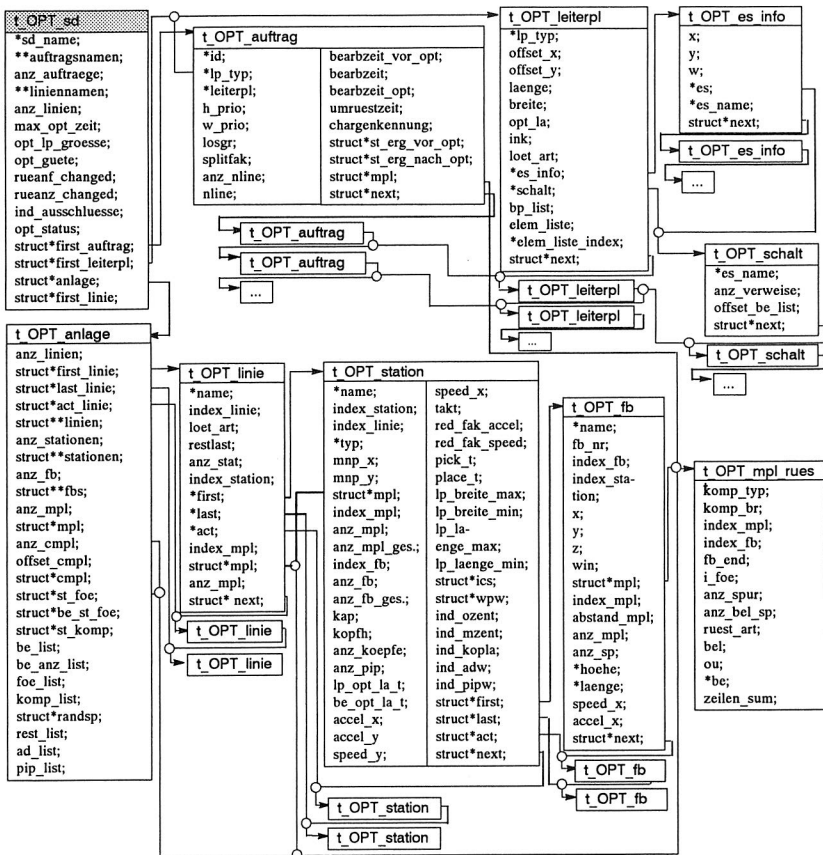


Bild 4-8: Auszug aus Hauptspeicherabbild

Eine weitere wichtige Datenstruktur stellt die Anlage dar. Sie gibt Auskunft über die betroffenen Linien und Stationen, über die dort gerüsteten Zusatzkomponenten sowie über alle Bauelemente, Förderertypen und deren mögliche Kombinationen bzw. Maschinentypzuordnungen.

Unterhalb der Steuerdatei sind in verzweigten Strukturen die Daten der Aufträge, wie beispielsweise die zu fertigenden Stückzahlen der Leiterplatten, abgelegt. Ebenso

sind zu jeder Leiterplatte die Menge der Bauelementetypen, die Bestückkoordinaten und Bestückschwerpunkte bestimmt.

Für die Ergebnisse der einzelnen Optimierungsalgorithmen sind ebenfalls Datenstrukturen vorhanden, die mit Fortschritt der Optimierung anwachsen. Die erforderliche dynamische Datenhaltung innerhalb der kompletten Hauptspeicherstruktur wurde mit verketteten Listen realisiert. Die für die einzelnen Module benötigten Zeit- und Rüstmodelle zu den verschiedenen Maschinensystemen sind ebenfalls Bestandteile der Hauptspeicherstruktur. Das Rüstmodell (s. Kap. 4.3.1) ist in dem allgemeingültigen Teil des Blackboards integriert. Die Zeitmodelle (s. Kap. 4.3.2) sind Bestandteile der jeweiligen Optimierungsalgorithmen. Zusammen bilden sie eine eigene Substruktur (s. Bild 4-9).

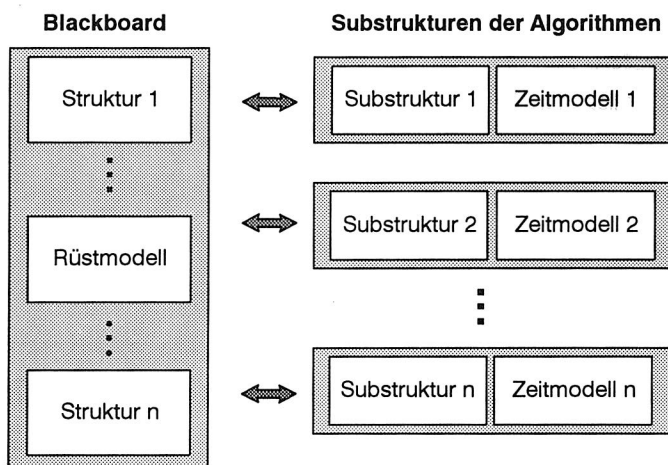


Bild 4-9: Eingliederung des Rüstmodells und der Zeitmodelle in die Hauptspeicherstruktur

### 4.3.1 Entwurf eines allgemeingültigen Rüstmodells

Ein wesentlicher Bestandteil der Datenstrukturen für das Optimierungsmodul ist das Rüstmodell und die Abbildung der Rüstanforderungen. Die Daten für einen Optimierungslauf bezüglich des Rüstmodells sind im einzelnen:

#### Rüstanforderungen

- Zuordnung Bauelementtyp zu Förderertyp in Abhängigkeit vom Maschinentyp
- Anzahl der zu rüstenden Bauelemente-Förderer-Kombinationen
- Platzbedarf des Förderers auf dem jeweiligen Maschinentyp
- Wechselwirkungen der Bauelement-Förderer-Kombination
- Zeitliche Bewertung des Umrüstaufwands der einzelnen Förderer

### Restriktionen

- Technologisch mögliche Zuordnung der Bauelement-Förderer-Kombination auf den zur Verfügung stehenden Rüstplätzen
- Benutzerbedingte Zuordnung der Bauelement-Förderer-Kombination auf den zur Verfügung stehenden Rüstplätzen

### Rüststände

- Abbildung eines möglichen Rüststandes
- Abbildung von Festrüstanteilen und Ausgangsrüstungen

Bei der Zuteilung von Aufträgen auf Linien und bei der Zusammenfassung von Aufträgen zu Chargen ist die Überprüfung der Rüstbarkeit ein fundamentales Kriterium. Die Rüstbarkeit ergibt sich zum einem aus technologischen Randbedingungen und zum anderen aus dem schwer zu kontrollierenden Vergleich der Rüstkapazität mit der aufzuteilenden Bauelementemenge. Im letzterem Fall sind zwei Arten von Fehlern möglich:

### Überschreiten der Rüstkapazität

Bei dieser Fehlerart werden Aufträge einer Linie oder einer Station bzw. zu viele Aufträge zu einer Charge zusammengefaßt, obwohl alle benötigten Bauelementetypen nicht auf einmal gerüstet werden können. Fehler dieser Art sind auf jeden Fall zu vermeiden, da sie maschinennahe Optimierungsschritte, wie z.B. die Optimierung der Feederkonfiguration, vor unlösbare Problem stellen.

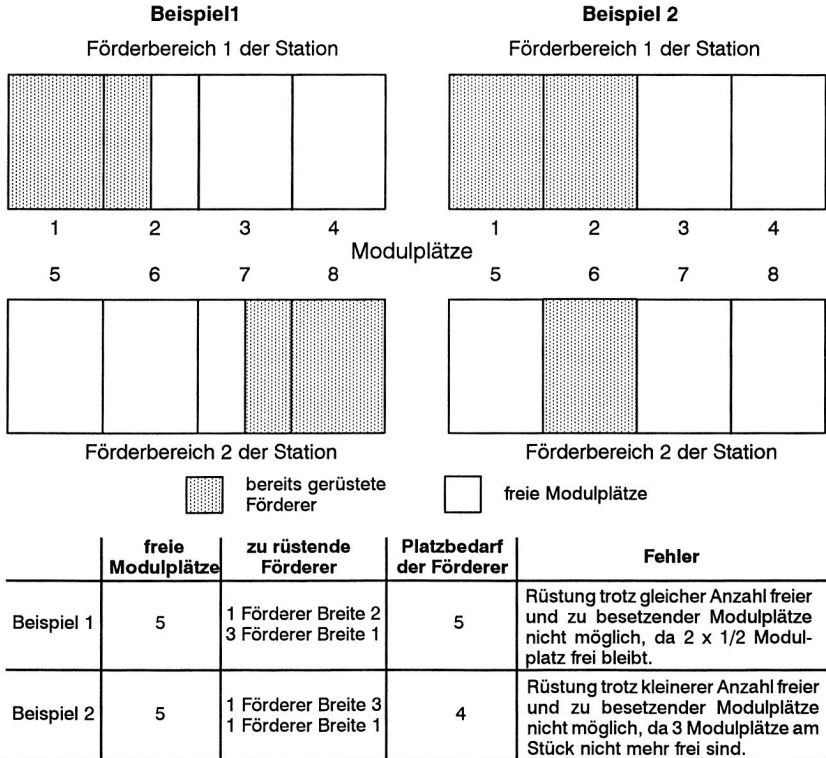
### Auslastungsfehler

Optimiert man mit zu großen Sicherheitsreserven, werden zu kleine Chargen gebildet, und es ist noch Stellkapazität auf der Linie bzw. Station vorhanden. Die Folge ist, daß häufiger als nötig umgerüstet werden muß. Da dieses Umrüsten teilweise sehr zeitaufwendig ist, macht die Optimierung alle möglichen Verbesserungen wieder zunichte. Diese Fehlerart ist nicht so gravierend wie die Überschreitung der Rüstkapazität, sollte aber auch so gering wie möglich gehalten werden.

Zur Überprüfung der Rüstbarkeit wurden mehrere Verfahren (Abzählen der Bauelemente, Ermitteln des Platzbedarfs der Bauelementetypen mit und ohne Reserven) getestet. Aufgrund der Komplexität der Förderer bei einigen Maschinentypen war bei keinem dieser Verfahren gewährleistet, Fehler beider Art komplett auszuschließen.

Ein Beispiel hierzu sind die Zuführmodule der Siemens Bestückungsautomaten. Die meisten dieser Förderer belegen ein Vielfaches von Modulplätzen (1;2;3 ...), wobei auch Förderer existieren, die zusätzlich noch halbe Modulplätze (1,5; 2,5; ...) beanspruchen. Die freigebliebene Hälfte des Modulplatzes kann im Anschluß nur ein Förderer belegen, der auch einen halben Modulplatz beansprucht, ansonsten bleibt der halbe Modulplatz ungenutzt. Ein Beispiel der Stellproblematik und der Fehler, die hierbei auftreten können, sind in Bild 4-10 zu sehen.

Zur Lösung dieses Problems wurde ein Rüstalgorithmus entwickelt, der mittels Spalten- und Zeilensummen mögliche Rüstungen berechnet. Diese Rüstmethode wird eingesetzt um gesicherte Aussagen über die Rüstbarkeit von Aufträgen bzw. Chargen zu treffen. Des weiteren wird diese Methode verwendet, um im Verlauf der Optimierungsalgorithmen ein Zuteilen, Verschieben und Vertauschen von Bauelementen zwischen den Stationen vorzunehmen.



*Bild 4-10: Beispiele für Fehler, die zur Überschreitung der Rüstkapazität führen*

Das folgende einfache Beispiel dient zur Erläuterung des Rüstalgorithmus: Aufgabe ist es, drei Bauelemente auf drei Stellplätze zu rüsten. Die Sichtweise des Förderers wird zunächst nicht berücksichtigt. Die Stellplätze, auf denen Bauelemente (z. B. aus technologischen Gründen) nicht gerüstet werden dürfen, sind grau hinterlegt.

In diesem Beispiel (Tabelle 4-1) ist noch auf einen Blick zu erkennen, daß alle drei Bauelemente gerüstet werden können. Aber es zeigt sich, daß es einige Lösungsansätze gibt, die ein Rüsten aller Bauelemente unmöglich macht. So ist es z. B. falsch,

Bauelement 2 auf Stellplatz 2 oder 3 zu rüsten, da sonst entweder auf Bauelement 1 oder 3 kein gültiger Stellplatz mehr entfällt.

	Bauelement 1	Bauelement 2	Bauelement 3
Stellplatz 1	rüstbar	rüstbar	
Stellplatz 2	rüstbar	rüstbar	rüstbar
Stellplatz 3		rüstbar	

*Tabelle 4-1: Beispiel zur Rüstbarkeit von Bauelementen*

Mit Hilfe der Spalten- und Zeilensummen lassen sich nun die günstigsten Rüstoppositionen ermitteln (s. Tabelle 4-2). Die Rüstreihenfolge ergibt sich aus den Minima der Spalten- und Zeilensummen, in diesem Fall 1. Ist die Spaltensumme für ein Bauelement 0 und ist dies noch nicht gerüstet, kann das Bauelement nicht mehr gerüstet werden (Abbruchkriterium). Ist dagegen die Zeilensumme 0, bedeutet dies, daß kein Bauelement vorhanden ist, das man auf diesen Stellplatz rüsten kann. Dies kann jedoch nur dann ein Abbruchkriterium sein, wenn die restlichen Bauelemente mehr Platz benötigen als noch auf der Linie vorhanden ist.

	Bauelement 1	Bauelement 2	Bauelement 3	Zeilensumme
Stellplatz 1	rüstbar	rüstbar		2
Stellplatz 2	rüstbar	rüstbar	rüstbar	3
Stellplatz 3		rüstbar		1
Spaltensumme	2	3	1	

*Tabelle 4-2: Beispiel zur Rüstbarkeit von Bauelementen mit Spalten- und Zeilensumme*

Durch die Hinzunahme der Fördererproblematik, der Bauelemente-Maschinentypzuordnung, Mehrfachrüstungen etc. wurde dieses Grundkonzept noch erheblich erweitert und modifiziert (Bild 4-11):

- Für jeden Förderertyp zu einem Bauelement wird eine Spalte angelegt
- Die Spaltensumme gibt die Anzahl der gleichzeitig stellbaren Förderer wieder, sofern nicht schon Förderer gerüstet wurden, auf denen das Bauelement gerüstet werden kann.
- Falls auf einem schon gerüsteten Förderer noch Platz für ein Bauelement ist (die Anzahl ist nicht entscheidend), beträgt sein Anteil zur Spaltensumme eins.
- Für mehrfach zu rüstende Bauelemente werden nur die Bauelemente-Förderer-Kombinationen und die zugehörigen Zeiger auf die Spalte kopiert.
- Die Restriktionsspalte wird über alle Linien erstellt, d.h. es ist eine Aussage über die Rüstbarkeit aller Maschinentypen möglich.
- Bei Fördereralternativen wird für jede Spalte eine Summe gebildet. Aber nur wenn die maximale Spaltensumme dieser alternativen Förderer das Minimum der gesamten Spalten- und Zeilensummen bildet, wird das Bauelement gerüstet.





Schritt 3: Fahrt zur Sonderstation (z.B. Zentrierstation)

Schritt 4: Absetzen des Bauelements

Schritt 5: Sonderaktion

Schritt 6: Aufnahme des Bauelements

Schritt 7: Fahrt zur Bestückposition

Schritt 8: Absetzen des Bauelements

Ein Zyklus für diesen Automatentyp setzt sich immer aus den Arbeitsschritten 1, 2, 7 und 8 zusammen (s. Bild 4-12). Die Vorgänge 3-6 sind optional in Abhängigkeit vom Bauelementtyp zu beachten und können je nach Automatentyp variieren. So ist es bei einigen Pick and Place-Automaten z.B. nicht erforderlich, das Bauelement auf der Zentrierstation abzusetzen.

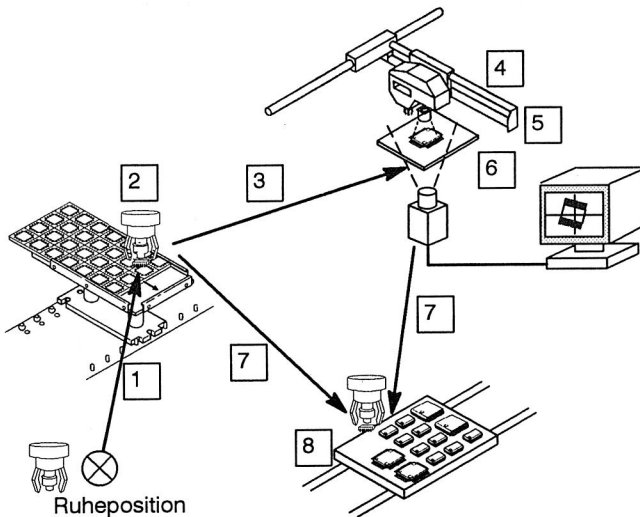


Bild 4-12: Bestückzyklus bei Pick and Place Automaten

Die Bestückzeit  $t_{BZ}$  eines Auftrags setzt sich folgendermaßen zusammen:

$$t_{BZ}(\text{Auftrag}) = \sum_{i=1}^{\text{Losgr.}} \sum_{\text{Betyp } 1}^{\text{Betyp } N} \sum_{j=1}^{\text{Anz. BP}} t_{\text{Pick}}(i) + t_{\text{Weg}}(i) + t_{\text{Place}}(i) + t_{\text{Sonder}}(i)$$

$t_{\text{Pick}}$  = Aufnahmezeit eines Bauelements

$t_{\text{Place}}$  = Absetzzeit eines Bauelements

$t_{\text{Weg}}$  = Fahrweg von Abholposition zu Bestückposition

$t_{\text{Sonder}}$  = Sonderzeiten z.B. Zentrieren, Koplanaritätsmessung etc.

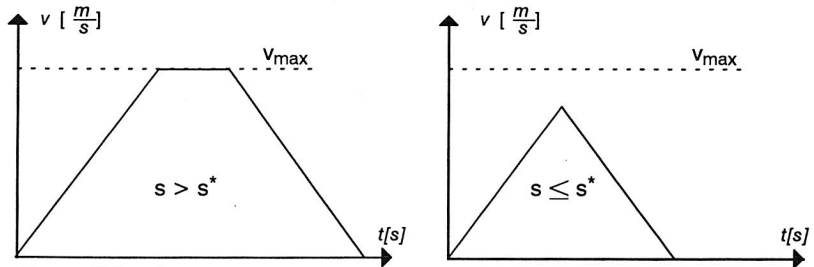
Für die Berechnung der Fahrzeiten gilt es zu beachten, daß die meisten Bestückungsautomaten die Achsen nicht synchronisieren. Die Fahrzeit richtet sich somit nach der langsameren Achse:

$$t_{\text{Weg}} (\text{Strecke}) = \max (t_{\text{Strecke}} (\text{x-achse}); t_{\text{Strecke}} (\text{y-achse}))$$

Für die Achsen gliedert sich der Bewegungsablauf grundsätzlich in die Abschnitte:

- Beschleunigung,
- Fahrt mit konstanter Geschwindigkeit und
- Abbremsen,

wobei die zweite Phase bei kurzen Wegen entfällt. Näherungsweise wird angenommen, daß konstante Beschleunigung sowohl beim Anfahren als auch beim Abbremsen vorherrscht. Zur exakten Positionierung der Bauelemente weisen die Automaten in der Praxis ein degressives Abbremsverhalten auf. In Abhängigkeit vom Bestückweg ergeben sich folgende Geschwindigkeits-Zeit-Diagramme und Fahrzeiten (s. Bild 4-13).



Gesamtfahrzeiten:

$$t_{\text{ges}} = \left( \frac{v_{\text{max}}}{a} + \frac{s_{\text{ges}}}{v_{\text{max}}} \right)$$

$$t_{\text{ges}} = 2 \sqrt{\frac{s_{\text{ges}}}{a}}$$

$s^*$  = Strecke, bei der Maximalgeschwindigkeit erreicht wird

$v_{\text{max}}$  = Maximalgeschwindigkeit der Achse

$s_{\text{ges}}$  = zurückzulegender Fahrweg

$a$  = Beschleunigung

Bild 4-13: Geschwindigkeits-Zeit-Diagramme

### Zeitmodell zu einem Chip-Shooter (Klasse III)

Für Automatentypen mit Bestückkopfkarrussell, bewegtem Bauelementeschlitten und Leiterplatte wurden, wie auch bei alle anderen Typen, drei unterschiedliche Zeitmodelle erstellt. Ein grobes Modell zur Abschätzung für Linienkapazitäten, ein feineres Modell zur Austaktung dieses Typs innerhalb der Linie und ein genaues Modell zur Spur- bzw. Bestücksequenzoptimierung und der detaillierten Bestimmung von Bestückzeiten.

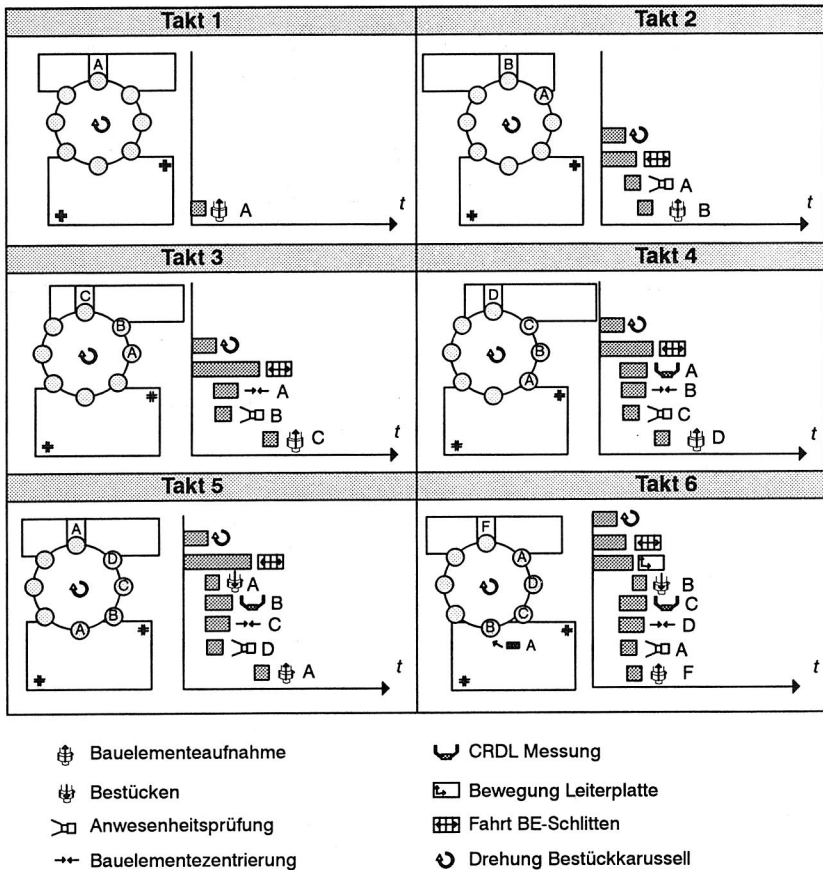


Bild 4-14: Taktablauf bei einem Karussellautomat

Exemplarisch soll das Zeitmodell zur Linienausstattung vorgestellt werden. Aus dem Bestückvorgang (s. Bild 4-14) dieses Automatentypen wird ersichtlich, daß der ausschlaggebende Faktor für die Berechnung der Gesamtzeiten einer Station der längste Anteil am Takt ist. Bestimmt wird der Taktzyklus aus den parallel ablaufenden Einzelaktionen:

- Bauelement abholen, d.h. Fahrt des Bauelementeschlittens
- Bauelement bestücken, d.h. Positionieren der Leiterplatte unter Bestückposition
- Sonderaktionen (z.B. Zentrieren, reduzierte Karussellgeschwindigkeit, Prüfung des elektrischen Wertes)

Berechnet wird der Takt aus dem Maximum der Durchschnittswerte der drei Zeiteile. Im detaillierten Zeitmodell der Spur- und Sequenzoptimierung wird hingegen mit fest vorgegebenen Zeitanteilen pro Bauelement gerechnet.

$$\text{Stationstakt} = \max \left\{ \begin{array}{l} \text{Durchschnitt der Leiterplattenbewegungen} \\ \text{Durchschnitt der Schlittenbewegungen} \\ \text{Durchschnitt der Sonderfunktionen} \end{array} \right.$$

Zur Berechnung des Durchschnittswertes für die Bestückung (d.h. für die Summe an Bewegungen der Leiterplatten) wird zu Beginn für jeden Leiterplattentyp eine grob optimierte Bestücksequenz errechnet. Für die Berechnung wurde ein sehr schneller Algorithmus verwendet. Zudem ist die Leiterplatte in Felder aufgeteilt (s. Bild 4-15). Dies hat zur Folge, daß eine Neuberechnung der Sequenz nur dann erforderlich ist, wenn ein neues Randfeld hinzukommt oder wegfällt. Dieser Fall tritt ein, wenn der Station ein Bauelement zugeteilt wird, das außerhalb der bestehenden Randfelder liegt oder wenn das letzte Element aus einem Randfeld gestrichen wird. Zur Bestimmung der Gesamtzeit werden aus den Einzelstrecken (Bestückposition  $n$  zu Bestückposition  $n + 1$ ) mittels Achsgeschwindigkeiten und Achsbeschleunigungen die Einzelzeiten aufsummiert und gemittelt.

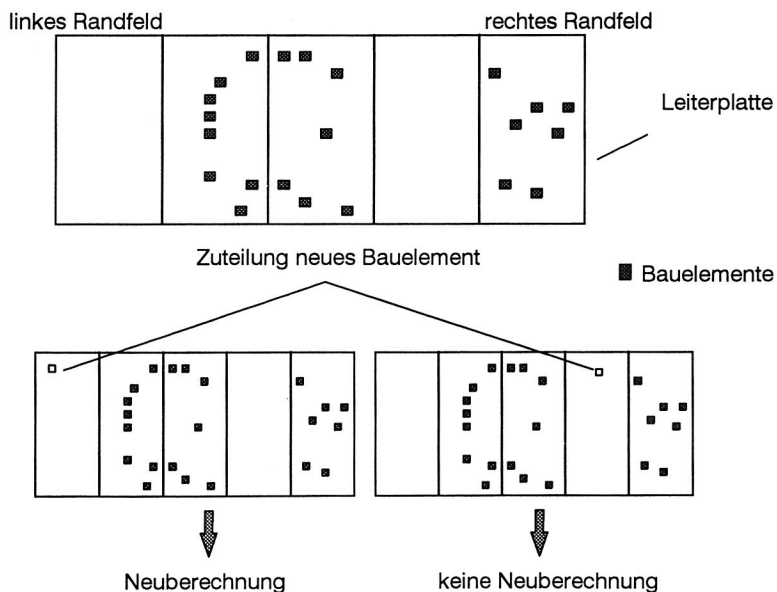


Bild 4-15: Berechnung der Bestückanteile der Leiterplatten über Randfelder

Der Durchschnittswert der Leiterplattenbewegungen errechnet sich demzufolge als:

$$\emptyset t_{LP} = \left[ \sum_{j=0}^{\text{Anz. LP}} \left[ \frac{\sum_{i=0}^{\text{Anz. BP}} t_{BE_{ij} \rightarrow BE_{i+1,j}} + t_{Place}}{\text{Anz. BP}} \right] \times \text{Lsgr}(LP_j) \right] \div \sum_{j=0}^{\text{Anz. LP}} \text{Lsgr}(LP_j)$$

Anz. LP = Anzahl der Leiterplatten in dieser Charge

Anz. BP = Anzahl der Bestückpositionen pro Leiterplatte

$t_{BE_{ij} \rightarrow BE_{i+1,j}}$  = Fahrzeiten von Bestückposition zu Bestückposition

$\text{Lsgr}(LP_j)$  = Losgröße der Leiterplatte j

$t_{Place}$  = Zeit zum Absetzen eines Bauelements

Die Zeit des Transportschlittens berechnet sich aus der durchschnittlichen Fahrzeit der Feederbank in Abhängigkeit von der Bestückreihenfolge und aus der Aufnahmezeit.

$$\emptyset t_{Schlitten} = \left[ \sum_{j=0}^{\text{Anz. LP}} \left[ \frac{\sum_{i=0}^{\text{Anz. BP}} t_{BS_{ij} \rightarrow BS_{i+1,j}} + t_{Pick}}{\text{Anz. BP}} \right] \times \text{Lsgr}(LP_j) \right] \div \sum_{j=0}^{\text{Anz. LP}} \text{Lsgr}(LP_j)$$

$t_{BS_{ij} \rightarrow BS_{i+1,j}}$  = Zeit der Feederbank von Abholspur zu Abholspur

$t_{Pick}$  = Zeit zur Aufnahme eines Bauelements aus Feederbank

Zudem werden für die Bauelemente die Maxima der Sonderaktionen aufsummiert und wiederum der Durchschnitt gebildet. Da sämtliche Zusatzfunktionen zeitparallel ablaufen, ist nur die längste Aktion ausschlaggebend. Bei den meisten Maschinen ist die Schlitten- oder die Leiterplattenbewegung der entscheidende taktgebende Faktor. Was auf jeden Fall beachtet werden muß, sind Vorgaben zur reduzierten Beschleunigung bei einigen Gehäuseformen. So erzwingt ein Bauelement, das nur mit geringerer Beschleunigung bestückt werden kann, eine langsamere Karussell-drehung, solange sich das Bauelement auf dem Bestückkopf befindet. Entscheidend ist aber die Auswirkung auf die Leiterplatte. Ist die erste Gehäuseform mit diesen Eigenschaften auf der Leiterplatte bestückt, kann ab diesem Zeitpunkt die Leiterplatte nur noch in reduzierter Geschwindigkeit verfahren werden.

Der aus diesen vorgestellten Schritten ermittelte Stationtakt bestimmt in Abhängigkeit vom zugeteilten Bestückvolumen die Gesamtzeit der Station.

### Rüst- und Umrüstzeiten

Ein letztes Beispiel der Zeitmodelle sind die Rüst- bzw. Umrüstzeiten, relevant für die Bestimmung kompletter Auftragszeiten und bei der Berücksichtigung von Ausgangsrüstständen. Das Planungssystem entscheidet anhand des zeitlichen Umrüstaufwands, ob es besser ist eine Bauelemente-Förderer-Kombination am Ausgangsort zu belassen, oder eine Umrüstung vorzunehmen. Diese Zeiten sind sehr fertigungsspezifisch und abhängig vom Maschinentyp. Bei einigen Automatentypen kann hauptzeitparallel gerüstet werden, so daß diese Zeiten nicht anfallen.

Zur Ermittlung praxisnaher Zeiten wurden in einigen Fertigungen Messungen durchgeführt und in Abhängigkeit vom Förderertyp und Rüsttätigkeit folgende Zeiten bestimmt.

Abrüstzeit: Abrüsten des kompletten Moduls  
 Aufrüstzeit: Aufrüsten des kompletten Moduls (d.h. Förderer bereits mit Bauelement bestückt)  
 Bauelementewechselzeit: Wechsel des Bauelementetyps innerhalb des Moduls, z.B. Gurt aus- und einfädeln.

Fördererart	Abrüstzeit [s]	Aufrüstzeit [s]	BE-Wechselzeit [s]
Gurtförderer (8/12/16/24mm)	10	20	300
Linearförderer	10	40	30
Schüttgutförderer	10	20	180
Flächenmagazin	20	30	--

Tabelle 4-3: Ermittelte Rüst- bzw. Umrüstzeiten

## **5 Import– und Exportfunktionalitäten**

Da die Planungskomponente als Bindeglied zwischen Entwicklungs- und Prozeßebene fungieren soll, ist ein Datenimport aus den Layoutsystemen sowie ein Datenexport in die entsprechenden Maschinenformate der Prozeßebene zu realisieren. Die folgenden Unterkapitel geben einen Überblick über die vorherrschenden Formate zum Datenaustausch in der Elektronikfertigung und die vorliegenden Probleme. Zudem werden Lösungsvorschläge für die Planungskomponente sowohl zum Datenimport als auch -export vorgestellt.

### **5.1 Datenformate und Schnittstellenproblematik der Elektronikfertigung**

Ein großes Problem der Datenhaltung und des Datenaustauschs in der Elektronikfertigung ist, daß viele Daten in nicht kompatiblen Formaten oder sogar redundant gespeichert werden. Zur Verbindung der CAX-Inseln werden oft individuelle Schnittstellen zum Datenaustausch programmiert [31]. Insbesondere bei Einbindung neuer Systeme wird die Ineffizienz dieser Vorgehensweise deutlich, da Fremddaten nur unzureichend weiterverarbeitet oder durch die Datenkomplexität Systeme nicht synchronisiert werden können. Doch gerade die Leiterplattenindustrie hat im Gegensatz zu anderen Industriezweigen große Vorteile, da viele produktbeschreibende Informationen von Beginn an als Datensätze im Rechner vorliegen. Ziel ist es, sich auf ein einheitliches Datenformat festzulegen, das die rechnerunterstützte Durchgängigkeit der Daten für alle Prozeßschritte sicherstellt [70].

Durch den gestiegenen Informationsbedarf und die Datenvielfalt wird die Informationsverarbeitung immer mehr zu einem bestimmenden Element. Die Forderung liegt nahe, automatisierte Produktionsinseln und komplexe Systemwelten der technischen und dispositiven Bereiche zu einem rechnerintegrierten Produktionssystem zu vereinen. Vorangetrieben durch verschiedene Gremien, aber auch durch enorme Marktdurchdringung bestimmter Systeme, sind einige Formate zum Datenaustausch im Bereich der Elektronik entstanden. Ein einheitliches Datenformat zur Integration aller Maschinentypen des Fertigungsprozesses, zur Beschreibung der Daten für elektrische Tests und optische Inspektionssysteme und zur Beschreibung der kompletten Leiterplattendaten hat sich jedoch noch nicht etabliert. Im folgenden werden einige Datenformate vorgestellt, die momentan und in der Zukunft im Elektronikbereich eingesetzt werden.

#### **5.1.1 Gerber-Format**

Das Gerber-Format wurde ursprünglich als Treibersprache für XY-Plotter entwickelt. Durch den großen Marktanteil an Gerber Plottern fand das Gerber-Format schnell weite Verbreitung und wird auch heute noch von den meisten CAD-Systemen unterstützt [112].

Vorteile:

- weit verbreiteter de facto-Standard
- ASCII-Format
- leicht lesbar

Nachteile:

- Darstellung im wesentlichen beschränkt auf Leiterbahnen
- lange Übertragungszeiten

### 5.1.2 IGES 5.1

Die Entwicklung von IGES (**I**nitial **G**raphics **E**xchange **S**pecification) begann 1979, angeregt durch die Nachfrage vieler CAD/CAM-Benutzer nach einem einheitlichen Datenaustauschformat. Die Grundlage bildeten Vorarbeiten von Boeing und General Electric, die von einem Komitee unter Aufsicht des amerikanischen NBS (**N**ational **B**ureau of **S**tandards) weiterentwickelt wurden. Im Juli 1989 wurde IGES Version 4.0 als ANSI-Standard anerkannt und liegt seit September 1991 in der Version 5.1 vor [87].

Vorteile:

- anerkannter Standard
- ASCII-Format
- ausgereifter Entwicklungsstand

Nachteile:

- geringe Verbreitung im Elektronikbereich
- starres Tabellenformat

Als Datenbasis eines Prozeßplanungstools bietet IGES viele Möglichkeiten, es wird derzeit aber hauptsächlich im Mechanikbereich eingesetzt.

### 5.1.3 IPC-D-35x

Das IPC (**I**nstitute for **I**nterconnecting and **P**ackaging **E**lectronic **C**ircuits) ist eine Handelsorganisation von Leiterplattenherstellern mit derzeit über 1600 Mitgliedsunternehmen. Es hat seit 1972 eine Reihe von Standards bzw. Entwürfen veröffentlicht, die der Darstellung und dem Austausch von Leiterplattendaten zum Zwecke der Fertigung dienen [112]. Die für die Belange der Arbeitsplanung relevanten Formate IPC-D-350 bis 357 umfassen Beschreibungen für Leiterbahnen, für schematische Zeichnungen, für Design-Rules, für elektrische sowie optische Testdaten, für Bestückdaten, für Bibliotheken und zum Bare Board [30].

Vorteile:

- anerkannter Standard
- ASCII-Format
- speziell auf Belange der Elektronikindustrie ausgerichtet



Nachteile:

- hoher Speicherbedarf
- starre Speicherstruktur (aus Lochkartenära)

#### 5.1.4 EDIF 3.0

Aus einer Initiative der Halbleiterhersteller Motorola, National und Texas Instruments sowie der Workstation- und Softwarehersteller Daisy, Mentor und Tektronix heraus begann 1983 die Arbeit an EDIF (**E**lectronic **D**esign **I**nterchange-**F**ormat), einem Beschreibungsformat, das ursprünglich zum Datenaustausch zwischen Werkzeugen zum Design von ICs und Leiterplatten gedacht war. Die Weiterentwicklung erfolgte seitdem durch verschiedene Technische Subkomitees unter der Führung der EIA (**E**lectronic **I**ndustries **A**ssociation) [75]. Bei der formalen Informationsmodellierung wurde hauptsächlich die Beschreibungssprache EXPRESS bzw. deren graphische Version EXPRESS-G verwendet. Im November 1993 wurde die Version 3.00 EIA-618 freigegeben und im Januar 1994 als ANSI-Standard anerkannt. Die Arbeit an EDIF findet inzwischen eine breite Unterstützung durch die Industrie.

Vorteile:

- anerkannter Standard
- breite Unterstützung seitens der Elektronikindustrie
- speziell auf Belange der Elektronikindustrie ausgerichtet
- ASCII-Format

Nachteile:

- einige wichtige Informationen noch nicht darstellbar
- schwer lesbares Format

#### 5.1.5 STEP

Im Dezember 1983 begann die ISO (**I**nternational **S**tandards **O**rganisation) mit der Entwicklung eines Standards, der sämtliche Informationen, die während des Lebenszyklus eines Produkts anfallen, abdecken sollte [96]. STEP (**S**Tandard for the **E**xchange of **P**roduct **M**odel **D**ata), eine ursprünglich europäische Initiative, wurde vom Subkomitee 4 (SC4-Industrial Data and Global Manufacturing Programming Languages) des technischen Komitees 184 (TC184-Industrial Automation Systems and Integration) der ISO getragen und erhielt die ISO-Kennung 10303 [47]. Parallel dazu begann 1986 die amerikanische IPO (**I**GES/**P**DES **O**rganisation) mit der Entwicklung des vergleichbaren Standards PDES (**P**roduct **D**ata **E**xchange **S**pecification). 1990 wurde beschlossen, beide Standards zu integrieren, wobei die Abkürzung PDES eine Umdeutung in "Product Data Exchange using STEP" erfuhr.

Vorteile:

- umfassende Produktbeschreibung
- Belange aller Sparten abbildbar
- ASCII-Format

Nachteile:

- viele Teile noch in Entwicklung

Über diese Formate hinaus gibt es noch eine Reihe weiterer Standards, wie z.B. DXF (Data Exchange Format) oder VDAFS (Verband der deutschen Automobilindustrie Flächen-Schnittstelle) [10]. Diese sind jedoch auf Beschreibung der äußeren Form eines Produktes beschränkt und daher für die Anforderungen der Elektronikproduktion ungeeignet [10].

## **5.2 Anbindung an die Leiterplattenentwicklung**

Für die Aufnahme der technischen Daten der Leiterplatten wurde ein Importfilter für das IPC-D-355-Format implementiert. Ausschlaggebend für diese Entscheidung waren die ausreichenden Möglichkeiten, die Belange der Bestückdaten abbilden zu können, die speziell für die Optimierungssoftware benötigt werden. Zur Vervollständigung der Verfahrenskette wurde exemplarisch für ein ECAD-System (PCDS) eine Ausgabeprogramm für das IPC-Format entwickelt.

Der Datentransfer aus dem IPC-Format in die Datenbank wurde mit Hilfe der Unix-Werkzeuge Lex und Yacc und einer zusätzlichen Interaktionsmöglichkeit für den Benutzer realisiert [103]. Den gesamten Umsetzungsprozeß zeigt Bild 5-1.

### **Lexikalische Analyse**

Der entwickelte Scanner übernimmt die lexikalische Analyse des Eingabestroms. Bestimmte Textmuster werden in diesem Dateneingabestrom erkannt und in spezielle Zeichenketten, sogenannten Token, umgewandelt.

### **Syntaktische Analyse**

Die aus dem Scanner übermittelten Token werden vom Parser in eine grammatische Struktur umgesetzt. Innerhalb des Parsers sind die Eingabestrukturen gemeinsam mit den Aktionen (Programmanweisungen) spezifiziert, die durchgeführt werden, wenn eine solche Struktur erkannt wird. Die formulierte Beschreibung der Eingabestrukturen werden zusammen mit den Aktionen in ein Ausgabeprogramm der Wirtssprache C umgesetzt.

### **Semantische Analyse**

Der vom Parser erzeugte Syntaxbaum wird mit der semantischen Analyse interpretiert, Plausibilitäts-, bzw. Konsistenzüberprüfungen parallel dazu durchgeführt und in eine interne Datenstruktur umgesetzt.

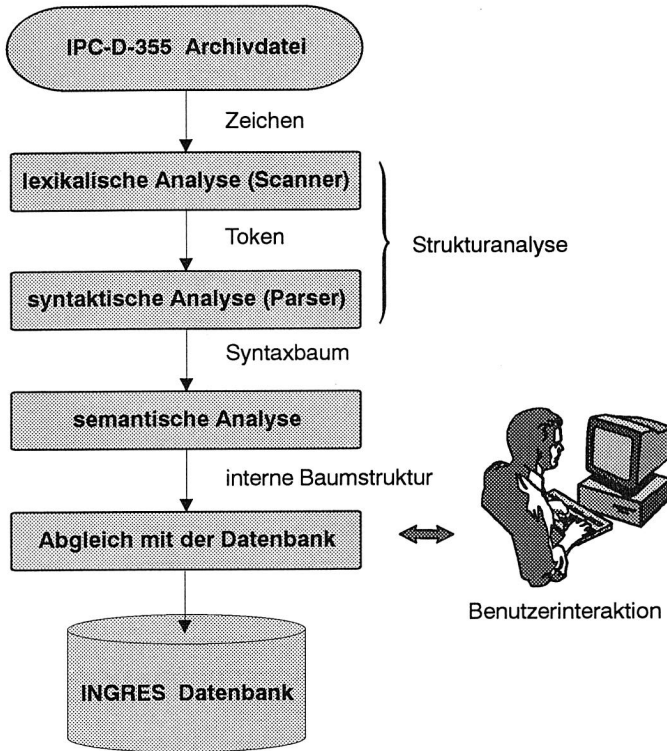


Bild 5-1: Gesamtablauf des Datentransfers

### Abgleich mit der Datenbank

In diesem Schritt wird die eingelesene Datenstruktur in die vorgesehenen Relationen der Datenbank übertragen. Von Vorteil in diesem Zusammenhang ist, daß die Datenbankstruktur und die semantische Struktur des IPC-Files ähnlich angelegt wurden. Beim Übertrag der Daten in die Datenbank ist bei auftretenden Inkonsistenzen eine Interaktion mit dem Benutzer zwingend erforderlich, z.B. muß der Benutzer beim Vorkommen eines noch nicht vorhandenen Bauelementetyps darauf hingewiesen werden und dann entscheiden, ob dieses Bauelement in die Bibliothek aufgenommen wird oder ob es sich nur um einen Namenskonflikt handelt.

### 5.3 Anbindung der Prozeßebene mittels eines flexiblen Postprozessors

Zur Erzeugung herstellerspezifischer Maschinenformate wurde ein flexibler Postprozessor entwickelt, der eine einfache und problemlose Anbindung der Datenbank an die Prozeßebene ermöglicht. Bild 5-2 zeigt den kompletten Ablauf von der Erstellung einer Beschreibungsdatei bis zur Ausgabe des individuellen Maschinenfiles.

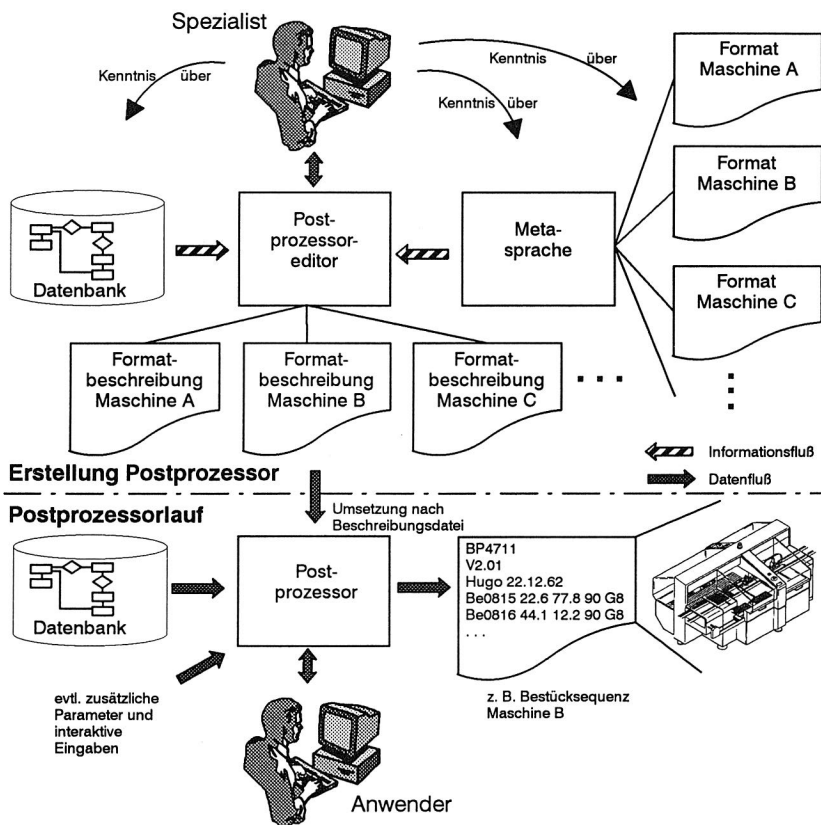


Bild 5-2: Gesamtstruktur des flexiblen Postprozessors

Nach Analysen verschiedener Maschinenformate wurde eine Metasprache, bestehend aus sechs Grundelementen, konzipiert, mit der die diversen Datenformate beschrieben werden können. In einer grafischen Oberfläche stehen die Grundelemente der Metasprache mit einer Reihe von Zusatzparametern zur Verfügung. Der Anwender kann ähnlich einem Baukastensystem das erforderliche Maschinenformat gra-

fisch am Bildschirm nachbilden. Nach Abschluß dieses Vorgangs wird eine Beschreibungsdatei im Format des Postprozessorgenerator angelegt. Diese Datei beinhaltet und beschreibt die Zusammenhänge der Daten aus der Datenbank, eventuell erforderliche Zusatzparameter wie Konstanten oder interaktive Eingaben mit der strukturellen Anordnung in dem entsprechenden Maschinenformat.

Für den eigentlichen Postprozessorlauf spezifiziert man die gewünschte Umsetzung, d.h. eine erzeugte Beschreibungsdatei. Entsprechend des Inhalts der Datei generiert der Postprozessorlauf nach den vorgegebenen Umsetzungsregeln nun die gewünschte Maschinendatei.

Entscheidende Vorteile dieser Entwicklung sind, daß durch die Metasprache ein neuer Postprozessor schnell und einfach erstellt werden kann und daß keinerlei Änderungen am Sourcecode und an der Struktur des flexiblen Postprozessors anfallen.

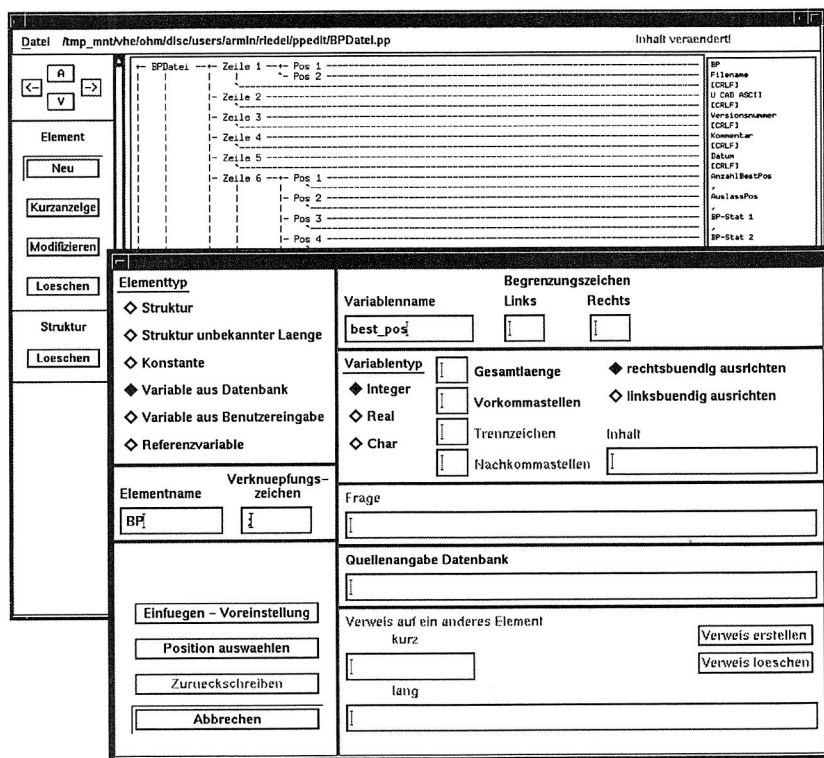


Bild 5-3: Postprozessoreditor für die Erstellung einer Beschreibungsdatei

Der Editor des Postprozessors ermöglicht die Erstellung, Verwaltung und Modifizierung der Beschreibungsdateien. Im Bild 5-3 ist auf dem hinteren Fenster ein kleiner

Ausschnitt einer erstellten Maschinenstruktur zu erkennen. Das Fenster im Vordergrund zeigt den Einfügemodus der Grundelemente und die einstellbaren Parameter. Die Elemente der Beschreibungssprache sind:

### **Struktur**

Dieses Grundelement enthält außer dem symbolischen Namen und dem evtl. nötigen Verknüpfungszeichen zum nächsten Element keine weiteren Daten. Es dient zur Aufnahme eines weiteren Elements nach rechts und unten (analog einem binären Baum) und damit dem Aufbau der komplexen Struktur.

### **Struktur unbekannter Länge**

Ein analoges Element zur Struktur mit dem Unterschied, daß zur Laufzeit des Postprozessors entschieden wird, wie oft diese Struktur durchlaufen wird (z.B. die Summe aller Bestückpositionen für ein Bestücksequenzfile).

### **Konstante**

Eine Konstante beinhaltet alle aufeinanderfolgenden ASCII-Zeichen, die ohne Veränderungen der zu erstellenden herstellerspezifischen Konfigurationsdatei erscheinen. Die Trennzeichen zwischen den einzelnen Elementen einer Zeile (meist Kommas, Klammern etc.) oder zwischen den Zeilen (CRLF) könnte man auch als Konstante definieren. Aufgrund ihrer Häufigkeit wurden sie jedem Element als Verknüpfungszeichen mitgegeben.

### **Variable aus Benutzereingabe**

Für diesen Wert wird der Typ der Variablen angegeben, wieviel Zeichen sie breit sein darf oder muß, ob rechts oder linksbündig ausgerichtet wird und ob der freie Platz mit einem bestimmten Zeichen aufzufüllen ist. Zusätzlich wird eine Textzeile eingelesen, die im Postprozessorlauf als Frage an den Benutzer erscheint.

### **Variable aus Datenbank**

Für dieses Grundelement muß die genaue Quelle aus der Datenbank spezifiziert werden. Während des Postprozessorlaufs werden die Verbindung aufgebaut, die Daten aus der Datenbank ausgelesen und im Herstellerformat spezifisch aufbereitet.

### **Zählvariable**

Diese Variable enthält einen Verweis auf eine vom Programm verwaltete Variable, in der z. B. die Anzahl der gelesenen Zeilen aus der Datenbank gezählt wird.

Allen Elementen gemeinsam ist, daß jedes Element einen symbolischen Namen besitzt, der mit der Maus bzw. Cursor selektierbar ist, eine Variable, die mit einem Verknüpfungszeichen besetzt werden kann und außerdem einen Zeiger in vier Richtungen (left, right, up, down). Die Verwaltung der komplexen Elementstruktur wurde mit einem objektorientierten Ansatz in C++ realisiert. Vorteil dieses Ansatzes ist, daß ein Hinzufügen bzw. Modifizieren eines bestehenden Elementes ohne Änderung der bestehenden Dateien des Postprozessors geschehen kann.

## 6 Entwicklung eines Design-for-Manufacturing-Systems

Die Bildung interdisziplinärer Teams beim Entwurf neuer Produkte gilt als eine der wichtigsten Voraussetzungen, um die Markteinführung so schnell und so unproblematisch wie möglich zu gestalten [16]. Dieses Prinzip des Concurrent Engineering bedeutet, vom Standpunkt der Fertigung aus gesehen, Produkte, in diesem Fall die Flachbaugruppen, zu entwerfen, die einfach bestückt, gelötet und getestet werden können. Design-for-Manufacturing, Design-for-Assembly oder Design-for-Testing-Systeme sind rechnergestützte Lösungen hierzu. Im Bereich der DFM-Systeme gibt es eine Reihe von Entwicklungen, die oft einzelne Prozeßschritte, z.B. das Testen oder Prozeßteilbereiche wie das Bestücken bedrahteter Bauelemente, unterstützen [61, 77, 68, 99]. Die wichtigsten Gründe für die Einführung und Nutzung von DFM-Systemen sind [41]:

- Die steigende Komplexität der Produkte.
- Die zunehmenden Anforderungen hinsichtlich ökonomischer und ökologischer Gesichtspunkte.
- Die Minimierung der Fertigungskosten. Test und Fallstudien haben gezeigt, daß DFM-Systeme die Bestückkosten um 25% und die Leiterplattenkosten um 35% senken können [42, 44].
- Im Entwicklungsprozeß eines elektronischen Produktes werden 60% der Fertigungskosten in den ersten Stufen des Designprozesses bestimmt, wohingegen bis zu diesem Zeitpunkt erst 35% der Entwicklungskosten angefallen sind [41].

Ziel war es, den Prototyp eines Expertensystems zu entwerfen, der nicht auf einen Prozeß der Flachbaugruppenfertigung fixiert ist, sondern der sich an unterschiedliche Fertigungsbelange flexibel anpassen läßt. Die datentechnische Struktur der Leiterplatten, des technischen Equipments und der Prozeßspezifika sollte so allgemeingültig wie möglich gehalten werden, so daß bei der Eingabe oder Änderung von Fertigungsregeln kein Modifizieren der Basisstruktur des Systems nötig ist. Bei der Entwicklung des DFM-Systems und zur Entscheidung, welche Expertensystemtechnik sich am besten eignet, wurde nach folgender Vorgehensweise verfahren [85]:

- Einteilung der konkreten Probleme in Problemlösungstypen
- Zuordnung der Problemlösungstypen zu Problemlösungsstrategien
- Abbildung der Problemlösungsstrategien in Wissensrepräsentationen und zugehörige Kontrollstrategien
- Implementierung der Wissensrepräsentation und der Kontrollstrategien

Mögliche Problemlösungstypen sind Diagnostik, Konstruktion und Simulation. Die Entwicklung eines DFM-Systems zum fertigungs- bzw. testgerechten Leiterplattendesign ist sicherlich dem Bereich der Diagnostik zuzuordnen, wobei nicht im klassischen Sinne ein Fehler erkannt oder eine Lösung aus einer Menge vorgegebener Alternativen bestimmt werden soll, sondern alle Fehler einer vorgegebenen Menge von Manufacturing-Rules aufzuzeigen sind. Weiterhin kann von einer vollständigen Pro-

blemsbeschreibung ausgegangen werden. Informationen, die nicht durch das vollständig beschriebene geometrische Modell einer Leiterplatte sowie der zugehörigen Fertigungsbeschreibung gegeben sind, gibt der Benutzer im Dialog ein. Genauer betrachtet liegt sogar ein Spezialfall der Diagnostik dahingehend vor, daß Symptome, also Verletzungen der Manufacturing-Rules, lediglich zu erkennen, nicht jedoch zu begründen sind. Wie sollte man auch einen Fehler des Layouters begründen?

Der nächste Schritt ist die Auswahl von Problemlösungsstrategien, welche dem Problemlösungstyp am besten zuzuordnen sind. Im Bereich der Grundtechniken der Wissensrepräsentation gibt es eine Reihe von Gemeinsamkeiten, die sich gut oder weniger gut für einen bestimmten Problemlösungstyp eignen (s. Bild 6-1).

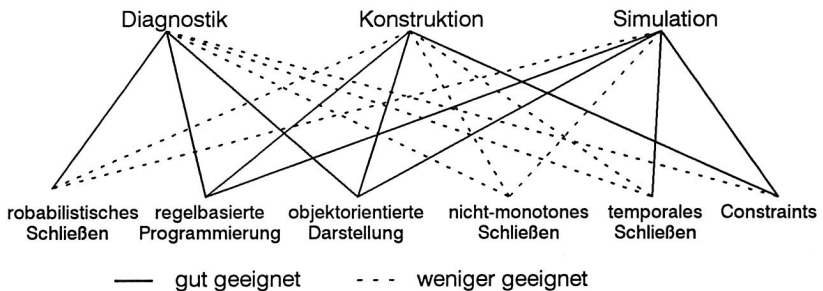


Bild 6-1: Zuordnung Problemlösungstypen zu Problemlösungsstrategien [85]

Im vorliegenden Fall kann davon ausgegangen werden, daß probabilistisches Schließen höchstens im Sonderfall der Datenerhebung im Benutzerdialog sinnvoll sein könnte. Das Verarbeiten einer "Nullwert"-Antwort ("weiß nicht") könnte dies notwendig machen [7]. Sonst sind Regelverletzungen eindeutig bestimmbar oder aus dem geometrischen bzw. Fertigungsmodell berechenbar. So gilt es, zwischen regelbasierter Programmierung und objektorientierter Wissensrepräsentation abzuwägen.

Der prinzipielle Aufbau eines Expertensystems besteht im wesentlichen aus vier Komponenten:

- Der Problemlösungskomponente, die das Expertenwissen mit dem fallspezifischem Wissen zur Lösung des vom Benutzer spezifizierten Problems benutzt.
- Der Interviewerkomponente, welche den Benutzerdialog führt, fallspezifische Daten ermittelt oder automatisch Werte, z. B. aus einer Datenbank, einliest.
- Der Erklärungskomponente, die gefundene Lösungen dem Benutzer gegenüber erklärt und dem Experten die Vorgehensweise, also das Anwenden der Regeln z.B. bei Fehlersuche, transparent machen kann.
- Die Wissenserwerbskomponente, welche zur Änderung und Erweiterung des Wissens dient und mehr oder weniger ausgeprägt sein kann.



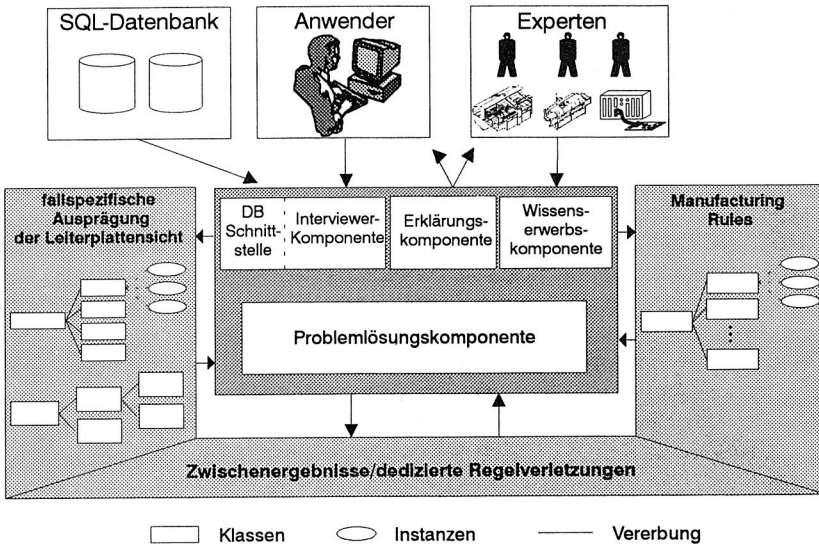


Bild 6-2: Architektur des DFM-Systems

In Hinblick der vorliegenden Problemstellung ergibt sich die in Bild 6-2 dargestellte Architektur für das DFM-System. Hierbei ist die Erklärungskomponente als schwach ausgeprägt zu verstehen, da eine erkannte Verletzung der Manufacturing-Rules nicht interpretiert, sondern lediglich in übersichtlicher Form visualisiert werden muß. Die schematisierte Sichtweise auf die Leiterplatte sowie die Modellierung der Fertigungsanlagen, erfolgt aus der Sicht des Experten. Das fallspezifische Wissen ist die Ausprägung einer solchen schematisierten Sicht für eine aktuell zu untersuchende Leiterplatte. Dieses Wissen wird im wesentlichen aus der geometrischen und fertigungsspezifischen Darstellung der relationalen Datenbank erhoben. Der Benutzerdialog ist optional zu verstehen, falls Angaben z. B. funktionaler Art benötigt werden, die jedoch nicht aus CAD-Daten zu berechnen oder abzuleiten sind. Ein weiterer Teil des Expertenwissens ist die Menge der zu überprüfenden Manufacturing-Rules. Entscheidend ist für diesen Teil die Änderungsfreundlichkeit und leichte Erweiterbarkeit, um den Adaptierungsmöglichkeiten an verschiedene Fertigungen Rechnung zu tragen.

## 6.1 Wissensrepräsentation

Wie bereits erwähnt, sind die geeigneten Wissensrepräsentationen für den vorliegenden Problemfall im Bereich der Diagnostik die regelbasierte Programmierung und die objektorientierte Darstellung.

### 6.1.1 Regeln

Eine Regel in regelbasierter Wissensrepräsentation besteht aus einer Vorbedingung und einem Aktionsteil [82]. Die Vorbedingung kann als boolesches Prädikat aufgefaßt werden, welches, falls als wahr bewertet, den Aktionsteil der Regel zur Ausführung freigibt. Regeln werden gern in der Formulierung *Wenn <Vorbedingung> Dann <Aktion>* angegeben. Die Verwendung von Regelmengen für die Wissensrepräsentation ist deswegen so beliebt, weil Experten ihr Wissen bei der Wissensakquisition oftmals intuitiv in dieser Form strukturieren. Die Aufteilung von komplexem Wissen in quasi eigenständige Wissensstücke macht eine Wissensbasis sehr flexibel, da man neue Regeln einfach zur bestehenden Regelmenge hinzufügen kann. Eine regelbasierte Wissensbasis besteht aus gültigen Fakten, Regeln zur Herleitung neuer Fakten und einem Inferenzmechanismus zur Steuerung des Herleitungsprozesses. Fakten können hierbei prinzipielle Gültigkeit besitzen oder fallspezifisch erhoben und dem System hinzugefügt werden. Das Expertenwissen ist meistens der Teil der Regeln, welcher zur Herleitung neuer Fakten von der Inferenzmaschine benutzt wird. Man kann zwischen zwei prinzipiellen Möglichkeiten zur Herleitung neuer Fakten unterscheiden:

- **Vorwärtsverkettung (Forward Chaining):** Ausgehend von einer vorhandenen Menge von Regeln und Fakten wird diejenige Menge von Regeln bestimmt, deren Vorbedingungsteil erfüllt ist. Eine dieser Regeln wird durch eine Auswahlstrategie bestimmt und ihr entsprechender Aktionsteil ausgeführt. Dieser Prozeß wird solange wiederholt, bis keine Regel mehr anwendbar ist [8].
- **Rückwärtsverkettung (Backward Chaining):** Ausgehend von einer oder mehreren Endregeln (Goals) werden alle Regeln überprüft, deren Aktionsteil zur Evaluierung der Vorbedingungen der Endregeln beitragen. Die Teile der Vorbedingung, welche nicht durch Fakten als "wahr" erkannt werden, versucht der Inferenzmechanismus durch erneute Regelanwendungen aus der Menge der Fakten abzuleiten [8].

### 6.1.2 Objektorientierte Programmierung

Im Gegensatz zu regelbasierten Systemen, in denen Wissen und Fakten unstrukturiert als Menge gespeichert sind, ist in der objektorientierten Welt die Strukturierung des Wissens das Wesentliche [62]. Die Welt bzw. deren Modell stellt sich als Menge von Objekten dar, die sich mit der Zeit ändern kann [12]. Eine erste Strukturierung wird erreicht, indem man die Aussagen über ein Objekt sammelt und in Datenstrukturen zusammenfaßt. Ein weiterer Abstraktionsschritt besteht darin, Objekte gleicher Struktur zu klassifizieren. Mit Objekten gleicher Struktur verbinden sich auch Funktionen und Operationen auf diesen Objekten. Mit diesem Tripel, der Menge von Objekten gleicher Struktur, einer Menge darauf anwendbarer Funktionen und der Menge der anwendbaren Operationen, ist im wesentlichen die Klassendefinition der objekt-

orientierten Sicht erklärt. Derart definierte Klassen können natürlich vollständig in anderen Klassen enthalten sein [104].

Objektorientierte Programmierung beruht nun auf der Vorstellung, es mit einer Gemeinschaft kommunizierender Objekte zu tun zu haben. Solchermaßen geartete Objekte kann man sich auch als separate Rechner mit bestimmten Funktionalitäten vorstellen. Wichtig ist, daß Objekte sowohl erzeugt als auch zerstört werden können. Dies bedeutet, daß eine Klasse zumindest einen Konstruktor und einen Destruktor in der Menge der Funktionen anbieten muß.

Ein weiteres typisches Merkmal für einen objektorientierten Entwurf ist die Vererbung. Wie schon angesprochen, können Klassen wiederum Klassen enthalten. Sowohl aus Sicht der funktionalen Schnittstelle als auch aus Sicht der Struktur, kann eine Klasse A spezifischer sein als eine Klasse B. Es bietet sich an, die Klasse A aus B abzuleiten. Die Klasse A ererbt dabei alle vererbbaaren Eigenschaften und Funktionalitäten von B und besitzt darüber hinaus selbst zusätzliche Eigenschaften und/oder Funktionalitäten. Man erreicht so eine Strukturierung der Menge der Klassen und eine weitere Verminderung von Redundanzen, vorausgesetzt, die gemeinsamen Eigenschaften werden so weit wie möglich oben im "Vererbungsbaum" beschrieben. Eine Klasse kann nicht nur von einer Klasse Eigenschaften ererben, sondern kann eine Menge sogenannter Parents besitzen. Ein Bauelement beispielsweise besitzt eine Menge spezifischer Bauelementeeigenschaften. Weiterhin besitzt eine konkrete Instanz sicherlich eine geometrische Beschreibung. Ein Bauelement mit gleichen elektrischen Eigenschaften kann möglicherweise in verschiedenen Bauformen mit unterschiedlichen geometrischen Beschreibungen verfügbar sein. Hier bietet sich die Möglichkeit, geometrische Beschreibungen und Bauelementebeschreibungen zu trennen und eine konkrete Instanz den entsprechenden Parents zuzuordnen.

## 6.2 Wissensakquisition

Die zu erwartende Regelmenge für ein DFM-System für mehrere Prozeßschritte ist sehr vielschichtig und umfangreich. Zur Entwicklung des Prototypen wurden durch Befragungen einiger Fertigungsexperten aus verschiedenen Firmen eine Reihe von Regeln in umgangssprachlicher Form zusammengestellt. Für die Bewertung eines Leiterplattendesigns wurden von den Fachleuten Fehlergewichtungen von 0-100 angegeben. 0 bedeutet völlige Irrelevanz des Fehlers, 100 die maximale Gewichtung. Im folgenden ein exemplarischer Auszug der aufgenommenen umgangssprachlichen Fehlerhypothesen:

1. Bauelemente müssen einen bestimmten Abstand zum Leiterplattenrand in Abhängigkeit von den zu durchlaufenden Maschinen einhalten  
Gewichtung: 100
2. Bauelemente müssen untereinander einen Mindestabstand einhalten (in Abhängigkeit von den zu durchlaufenden Maschinen).  
Gewichtung: 100

3. Bestimmte Bauelemente sollten in Bezug auf die Transportrichtung mit bestimmter Orientierung plaziert werden. (Bei Schwallöten 90° also quer zur Transportrichtung).  
Gewichtung: 70-80
4. Je nach Lötverfahren (Schwall-/ Reflowlöten) für die Leiterplattenseiten sollten Bauelemente bevorzugt auf Ober- oder Unterseite bestückt werden.  
Gewichtung: 100
5. Eine Sperrzone für Barcodelabel, in der nicht bestückt werden darf, muß vorgesehen werden.  
Gewichtung: 80
6. Schnelle Logikbausteine sollten kurze Verbindungen zur Steckerleiste aufweisen.  
Gewichtung: 50-60
7. Pads müssen untereinander und zur angebundenen Bleizinnstruktur Mindestabstände einhalten, um ungewollte Lotbrücken vor allem beim Schwallöten, zu verhindern.  
Gewichtung: 100

### 6.2.1 Formalisierung der Fehlerhypothesen

Die aufgeführten Fehlerhypothesen mit den zugehörigen Gewichtungen sind anwenderspezifische Beispiele. Die entwickelte Struktur des DFM-Prototypen erlaubt es, adaptierbar an die Gegebenheiten des Anwenders, Regeln und Gewichtungen einzugeben. Einige bei der Befragung aufgenommene Design-Rules wurden zur Entwicklung und zur Demonstration der Funktionsweise des Systems herangezogen. Mit Hilfe eines konkreten Beispiels (Regel 1 und Regel 5) soll das Datenmodell der Wissensrepräsentation in einem semantischen Netz und das Regelwerk dokumentiert werden.

#### Regel 1:

**Bauelemente müssen einen bestimmten Abstand zum Leiterplattenrand in Abhängigkeit von den zu durchlaufenden Maschinen einhalten.**

Begriffe sind: Bauelemente, Abstand\_Leiterplattenrand, Maschinen

Aus dem allgemeinen Wissen kommt hinzu:

- Maschinen sind standalone oder in Linien angeordnet.
- Leiterplatten werden auf Bestückungssystemen bestückt.
- Leiterplatten bestehen aus einer oder mehreren Einzelschaltungen.
- Einzelschaltungen können Bauelemente auf Ober- und Unterseite aufweisen.
- Leiterplatten haben eine geometrische Form.
- Bauelemente haben eine geometrische Form.

- Geometrische Formen haben einen minimalen Abstand zueinander.
- Maschinen benötigen Mindestabstände der Bauelemente zur Leiterplatte.

Die Abbildung in einem semantischen Netz zeigt Bild 6-3.

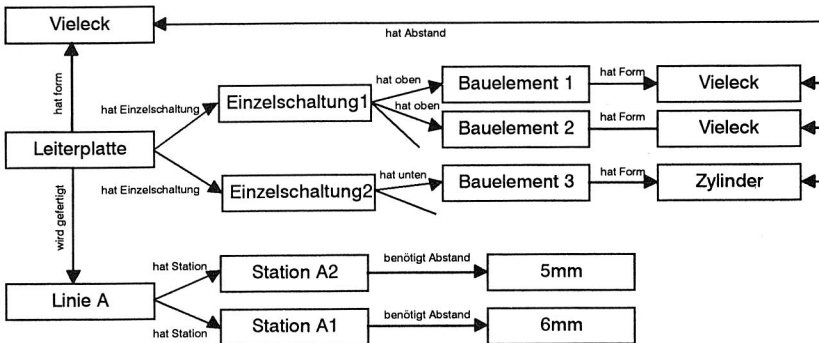


Bild 6-3: semantisches Netz zu Regel 1

Die zugehörige Regel in Pseudocode:

```
IF      {Abstand Bauelementeform_Leiterplattenrand
        < max {benötigter Abstand (Stationen)}
THEN    Regelverletzung (Regel 1)
```

### Regel 5:

**Sperrzone für Barcodelabel, in der nicht bestückt werden darf, muß vorgesehen werden.**

Begriffe sind: Barcodelabel, Leiterplatte, Sperrzone

Aus dem allgemeinen Wissen kommt hinzu:

- Leiterplatten haben Einzelschaltungen.
- Leiterplatten haben Barcodelabel auf Ober- und Unterseite.
- Einzelschaltungen haben Barcodelabel auf Ober- und Unterseite.
- Einzelschaltungen haben Bauelemente auf Ober- und Unterseite.
- Barcodelabel haben geometrische Beschreibung.
- Bauelemente haben geometrische Beschreibung.

Die zugehörige Regel in Pseudocode:

```
IF      {hat_Überdeckung (Bauelementeform, Barcodelabelform)}
THEN    Regelverletzung (Regel 5)
```

Die Abbildung in einem semantischen Netz zeigt Bild 6-4.

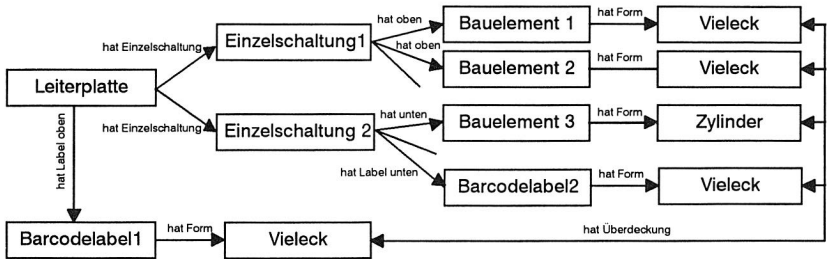


Bild 6-4: semantisches Netz zu Regel 2

Die Umsetzung der Regeln in der beschriebenen Weise und die disjunkte Vereinigung der semantischen Netze führen zu einem gemeinsamen Datenmodell für die Wissensrepräsentation, das sich auch für die Erweiterung des Regelsystems eignet (s. Bild 6-5). Zu erkennen ist, daß das generelle Wissen einen erheblichen Umfang der Wissensrepräsentation ausmacht und sich bei der Beschreibung jedes Regelbeispiels durch ein semantisches Netz große Redundanzen ergeben.

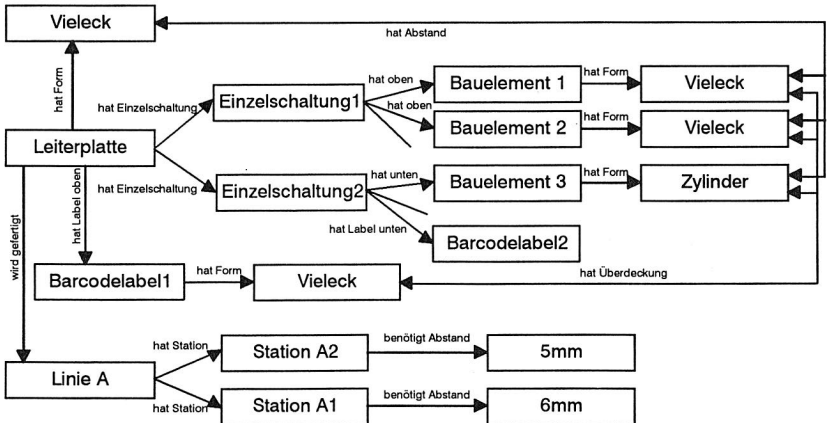


Bild 6-5: disjunkte Vereinigung der semantischen Netze aus Beispiel 1 und 2

Sowohl die direkte Umsetzung des vereinigten semantischen Netzes in eine objektorientierte Darstellung als auch die rein regelbasierte Präsentation ergäben einige Probleme. Die rein objektorientierte Darstellung würde bei einer größeren Menge an Rule-Checks zu einer Überflutung der Objekte mit Attributen führen, bei einer rein regelorientierten Vorgehensweise läge demhingegen eine große Menge des Faktenwissens in unstrukturierter Form vor. Aus diesen Gründen wurde eine hybride Lösung favorisiert, eine objektorientierte Darstellung des Faktenwissens verbunden mit einem Regelsystem. Zudem wurde im Sinne objektorientierter Datenmodelle innerhalb des semantischen Netzes eine Klassifizierung der Objekte durchgeführt (s. Bild 6-6).

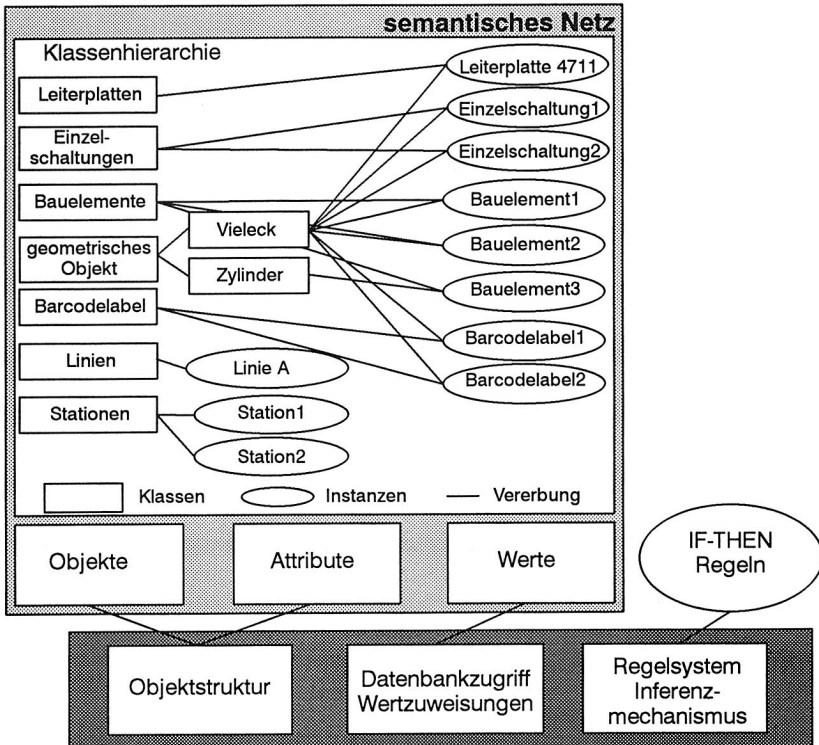


Bild 6-6: Implementierung des formalisierten Regelwissens

Das semantischen Netzes zur Repräsentation des Faktenwissens wurde wie folgt klassifiziert: Knoten des semantischen Netzes, welche strukturell mehrfach vorkommen oder von denen ein mehrfaches Vorkommen denkbar wäre, führen zu einer Klassendefinition. Knoten, die lediglich Werte repräsentieren, gehen über in den Wert eines Attributes in der entsprechenden Instanz. Beispielsweise enthält die Instanz **Station A2** das Attribut **benötigt Abstand** mit Wert **5mm**. Die von einem Knoten des Netzes fortführenden Kanten werden in der Klassendefinition als für ein Objekt gemeinsame Menge von Attributen definiert. Exakter: Die disjunkte Vereinigung der von einer Menge gleichartiger Objekte fortführenden Kanten bildet in der Klassendefinition die Menge der Attribute. Die Objekte, auf welche die Kanten zeigen, sind jetzt entweder einfache Werte einer Instanz oder eine Referenz auf eine andere Instanz. Für Einzelschaltungen würde in der Klassendefinition das Vorhandensein der Attribute **hat oben** und **hat unten** vereinbart. Das Attribut **hat oben** der Instanz **Einzelschaltung1** erhält den Wert **Bauelement1**. Hierbei ist der Wert des Attributes als Referenz auf die Instanz **Bauelement1** zu verstehen. Es wird allerdings vorausgesetzt,

daß die Möglichkeit besteht, einem Attribut auch mehrere Werte zuzuordnen, da Einzelschaltungen sicher mehrere Bauelemente auf Ober- und Unterseite besitzen. Interessant ist das Ausnutzen der Mehrfachvererbung. Bei der Instanziierung, also der Falldatenerhebung, kann jedem Objekt ein geometrisches Outline zugeordnet werden. Hierbei ererbt die jeweilige Instanz z.B. die Eigenschaften eines Vielecks. Unter diesen Eigenschaften könnte eine Methode sein, welche es einem Objekt mit geometrischem Outline ermöglicht, den Abstand zu einem anderen geometrischen Objekt zu bestimmen. Die Definition einer solchen Methode ist am besten in der Klasse **geometrisches Objekt** vorzunehmen, damit alle Geometrieobjekte diese Eigenschaft ererben können. Auf diese Weise sind die bidirektionalen Kanten **Abstand\_zu** und **hat\_Überdeckung** aus Bild 6-4 gut modellierbar. Eine Überflutung der Instanzen mit Attributen und Werten kann so vermieden werden.

### 6.2.2 Repräsentation und Gewichtung der Fehler

Um nun ein Leiterplattendesign zu überprüfen, sind noch einige Vorgaben zu berücksichtigen:

- Eine Diagnose soll hinsichtlich der zu überprüfenden Regeln konfigurierbar sein.
- Am Ende einer Diagnose soll eine Bewertung des Leiterplattendesigns in Abhängigkeit der gefundenen Fehlern erstellt werden.

Ein gefundener Fehler in diesem Zusammenhang ist eine Ausprägung einer Regelverletzung. Fehler werden durch den Inferenzmechanismus des Regelsystems mit Hilfe des vorgestellten objektorientierten Faktenwissens aufgedeckt. Ausgehend von dem eingegebenen Faktenwissen, werden alle durch die Diagnose konfigurierten (IF-THEN) Regeln ausgewertet und Schlußfolgerungen gezogen.

Die Repräsentation der gefundenen Fehler resultiert wiederum in einer objektorientierten Darstellung. Es wurde hierzu eine Oberklasse **Fehler** mit folgenden Attributen angelegt:

Gewicht	Attribut
Assertkomponente	?
Konstruktor	Methode
Destruktor	Methode
Erklärungskomponente	Methode

Das **Gewicht** ist die Grundlage für die zu erstellende Bewertung. Werte für die einzelnen Fehlerhypothesen wurden in Kapitel 6.2 angegeben. Da an dieser Stelle die tool-spezifischen Möglichkeiten, das objektorientierte Faktenwissen in das Regelsystem zu transferieren, unbekannt sind, wird in Anlehnung an das in vielen regelbasierten Systemen existierende ASSERT Statement zum Hinzufügen von Fakten zu einem Regelsystem eine allgemeine, nicht näher spezifizierte Assertkomponente vorgesehen. **Konstruktor** und **Destruktor** legen Ausprägungen an und löschen sie wieder (falls



eine neue Diagnose in derselben Sitzung stattfindet). Dem **Konstruktor** kommt im Einzelfall auch das Erstellen der individuellen Fehlergewichtung einer anzulegenden Instanz zu. Die **Erklärungskomponente** dient zur Anzeige der gefundenen Regelverletzung. Hier sollten mehrere fehlerspezifische Möglichkeiten vorgesehen werden. Ein Rule-Check aus der in Kapitel 6.2 zusammengestellten Auflistung stellt eine spezifischere Regelverletzung dar. Daher muß für jeden Rule-Check eine eigene Klasse aus der oben spezifizierten verallgemeinerten Klasse abgeleitet werden. Zur Erläuterung wird dies anhand des Beispiels 1 aus Kapitel 6.2 durchgeführt.

### Regel 1:

**Bauelemente müssen einen bestimmten Abstand zum Leiterplattenrand in Abhängigkeit von den zu durchlaufenden Maschinen einhalten.**

Zusätzlich erforderliche Attribute der Fehlersubklasse:

Bauelementebezeichnung	Attribut
Position	Attribut

Unter der **Bauelementebezeichnung** ist eine Gehäuseform oder ein Typname zu verstehen. Die **Position** ist notwendig, falls mehrere gleiche Typen oder Gehäuseformen auf der Leiterplatte vorhanden sind. Aus rein objektorientierter Sicht wäre eine Referenz auf das betreffende Bauelement natürlich völlig ausreichend.

Für jeden Rule-Check aus der Liste aus Kapitel 3 ist in dieser Weise eine ausreichend beschreibende Struktur auszuarbeiten. Diese Strukturen bilden dann die aus der verallgemeinerten Fehlerklasse abgeleiteten Subklassen (s. Bild 6-7).

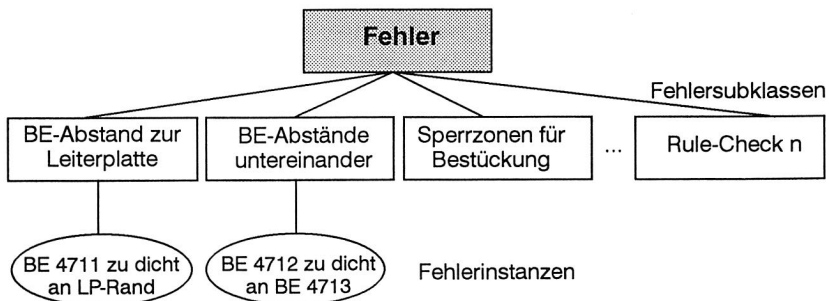


Bild 6-7: Fehlerklassenhierarchie mit Instanzen

Als Grundlage für die Bewertung eines Leiterplattendesigns dient die Menge der aufgedeckten Regelverletzungen, also alle erzeugten Instanzen der einzelnen Fehlerklassen einer Diagnose. Jede dieser Instanzen besitzt ein Gewicht, das entweder ererbt oder individuell berechnet wurde. Ererbte Gewichte repräsentieren eine Art Ja/Nein-Fehler, bei welchem eine Regelverletzung besteht oder nicht. Individuelle Be-

wertungen entstehen im Falle einer möglichen Einschätzung einer Regelverletzung. Ein Bauelement kann beispielsweise in einer Orientierung bestückt werden, die der vorzuziehenden Ausrichtung zwar nicht genügt, jedoch nur eine kleine Winkelabweichung darstellt. Ein derartiger Fehler wird festgestellt, eine Instanz angelegt, das Gewicht basierend auf dem Grundgewicht der Fehlerklasse allerdings individuell berechnet. Eine große Winkelabweichung führt zu einer hohen Gewichtung, eine geringe Abweichung lediglich zu einem kleinen Gewicht in Relation zum Basisgewicht der Fehlerklasse.

Zur Gütebewertung wurde die Dichtefunktion aus der Stochastik verwendet:

$$G = e^{-\alpha \sum_{i=1}^{\text{Anz. Inst.}} g_i} \quad \begin{array}{l} \alpha = \text{Skalierungsfaktor} \\ g_i = \text{Gewichtung der Fehlerinstanz} \end{array}$$

Ein fehlerfreies Design im Sinne der Rule-Checks wird mit 1 bewertet. Gefundene Fehler führen je nach Gewicht zu einer Abwertung. Mit dem Skalierungsfaktor kann die Steigung der Abwertung justiert werden.

Zum kompletten Ablauf einer Diagnose legt der Benutzer mittels der Interviewerkomponente die zu bewertende Leiterplatte fest und konfiguriert die Diagnose, indem er alle oder nur einige Fehlerhypothesen spezifiziert. Das Datenbankinterface liest aus der Datenbank die entsprechenden technologischen Parameter der Leiterplatte, der Bauelemente und des technischen Equipments. Entsprechend der objektorientierten Repräsentation des Faktenwissens werden daraufhin die Instanzen hierzu gebildet. Mittels des Assertstatements wird das Faktenwissen dem Regelsystem zur Kenntnis gebracht. Bei Regelverletzungen bildet das System die entsprechenden Fehlerinstanzen analog zur objektorientierten Fehlerrepräsentation. Nach Durchlauf aller Fehlerhypothesen wird das Design bewertet und mit Hilfe der Erklärungskomponente gefundene Fehlerinstanzen angezeigt.

### 6.3 Erstellung eines DFM-Prototyps

Implementiert wurde der Prototyp unter der Entwicklungsumgebung PROKAPPA, die eine objektorientierte, regelbasierte und hybride Form der Wissenrepräsentation unterstützt. Die Hauptkomponenten des Entwicklungstools sind ein Regelsystem mit vor- und rückwärtsverkettbaren Inferenzmechanismus, eine objektorientierte Wissensrepräsentation, eine Interface-Workbench zur objektorientierten Entwicklung grafischer Benutzeroberflächen unter X11 und ein ANSI C Interpreter. Wichtige Voraussetzung war die Möglichkeit, von PROKAPPA aus auf die zentrale Datenbank zugreifen zu können.

Zur Realisierung des Prototypen wurde die an den Beispielen vorgestellte Vorgehensweise zur Formalisierung des Regelwissens eingehalten. Die Anwendung selbst ist in drei Teile gegliedert: **DFMmeta**, **DFMrumpf** und **DFMfehler** (s. Bild 6-8).

## DFMmeta

Die Applikation **DFMmeta** enthält die komplette Ablaufsteuerung des Systems und führt den Benutzerdialog der Diagnoseanwendung. Innerhalb dieses Moduls erfolgt die Kommunikation mit der zentralen Datenbank, d.h., mit Hilfe der eingebetteten SQL-Anweisungen werden die erforderlichen technischen Daten in das DFM-System eingelesen.

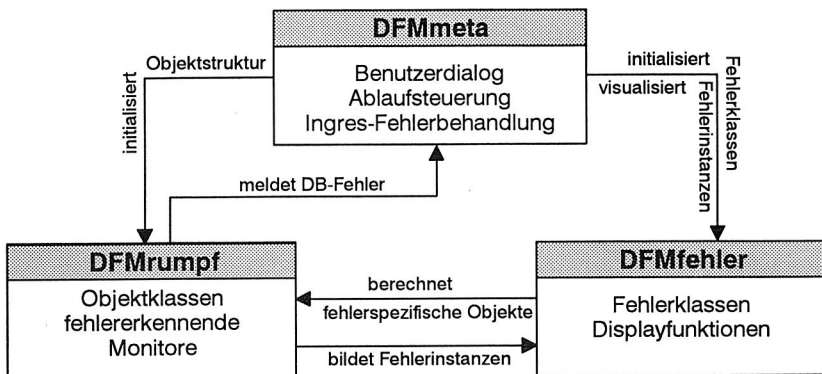


Bild 6-8: Struktur des implementierten Prototyps

## DFMrumpf

Das Modul **DFMrumpf** ist die Repräsentation des Faktenwissens. Innerhalb dieses Moduls wird die aktuell zu untersuchende Leiterplatte zusammen mit der Sicht auf die Fertigungsmöglichkeiten dargestellt. Initial existieren hier nur Klassendefinitionen mit ihren Ableitungshierarchien. Eine zu untersuchende Leiterplatte wird als Instanz, genauer als Menge aller Instanzen, die für eine Beschreibung notwendig sind, angelegt. Die Sicht auf eine Leiterplatte entspricht der in Kapitel 6.2 beschriebenen Modellierung semantischer Netze. Ein Beispiel zu den implementierten Klassen bzw. Subklassen ist die Klasse Geometrieobjekt:

Klasse Geometrieobjekt

Slot	Werte
abstand_zu	default
drehung	default
liegt_in	default
lies_ref_punkt	Geometrieobjekt_liegt_in
ref_punkt	(0,0)
setze_ref_punkt	Geometrieobjekt_setze_ref_punkt

Die Attribute der einzelnen Klassen sind in den sogenannten Slots abgebildet. Die Slots **abstand\_zu**, **drehung** und **liegt\_in** enthalten als Wert lediglich Defaultfunktio-

nen, deren Bezeichner keine Rolle spielt. Die Slots wurden nur deshalb in der subsumierenden Klasse definiert, um ihre Existenz in allen abgeleiteten Klassen zu erzwingen. In den abgeleiteten Klassen besitzen diese Slots dann Werte, die für das entsprechende geometrische Objekt die geforderten Funktionalitäten bereitstellen. Hierbei berechnet die Funktion im Slot **abstand\_zu** den Abstand des jeweiligen Objektes zu einem beliebigen anderen geometrischen Objekt. Der Slot **drehung** enthält Funktionen, die das aktuelle geometrische Objekt um einen vorgegebenen Punkt rotieren. Parameter sind der x-Wert und der y-Wert des Punktes, um welchen die Rotation erfolgen soll, sowie eine Winkelangabe für die Rotation. Das Senden einer Message an den Slot **liegt\_in** mit einem zusätzlichem Parameter als Referenz auf ein weiteres geometrisches Objekt stellt Überdeckungen zwischen den geometrischen Beschreibungen der beiden Objekte fest. Über diese Methode kann entschieden werden, ob das aktuelle Objekt innerhalb einer durch das zweite Objekt beschriebenen Zone liegt oder nicht. Das Flag dient dazu, festzulegen, ob auf vollständiges Umgeben geprüft werden soll oder nicht. Setzen des Flags auf 1 bewirkt die Überprüfung auf vollständiges Umgeben, der Flag 0 überprüft einfache Überdeckungen. Alle in dieser Anwendung betrachteten Geometrieobjekte besitzen einen Referenzpunkt, auf welchen sich die relative geometrische Beschreibung in der Datenbank bezieht. Der Referenzpunkt eignet sich auch dazu, ein geometrisches Objekt zu identifizieren. Defaultwert für diesen Referenzpunkt ist (0,0). Die Methoden **lies\_ref\_punkt** und **setze\_ref\_punkt** dienen zum Lesen bzw. Setzen des Referenzpunktes. Abgeleitete Klassen dieses Beispiels sind die Subklassen Polygonzug, Baumstruktur, Kreis, Kreisring etc., auf die im Detail nicht eingegangen wird.

### DFMfehler

Das Modul **DFMfehler** enthält die Fehlerklassen und damit die Repräsentation des Regelwissens. Die gesamte Anwendung ist so ausgelegt, daß hier beliebige neue Fehlerklassen hinzugefügt werden können. Einzige Bedingung ist, daß neue Klassen nicht als Top-Level Klassen, sondern als Ableitungen der subsumierenden Klasse Fehler angelegt werden. Allen Fehlerklassen gemeinsam sind die Eigenschaften, welche in der Klasse **Fehler** vereinbart sind:

Klasse Fehler:

Slot	Werte
Gewichtung	0
Lies_gewicht!	fehler_Lies_gewicht
Display!	default
Test!	default
Bilde_Instance!	default
Löschen!	Fehler_löschen

Alle Fehlerklassen besitzen ein vom Benutzer einstellbares Grundgewicht. Die Zuweisung eines Gewichtes erfolgt in der Relation Fehler der Datenbank. Der Metho-

denslot **lies\_gewicht** enthält eine rekursive Funktion, die ausgehend von der Klasse **Fehler** versucht, die Gewichte der Fehlerklassen über einen Datenbankzugriff zu bestimmen. Sollte dies nicht möglich sein, muß angenommen werden, daß der Fehler irrelevant für die Bewertung ist. Ein solcher Fehler erbt das Defaultgewicht der Klasse **Fehler**. Eine ebenfalls rekursive Methode, stellt der Wert des Slots **Loeschen** dar. Das Senden einer Message an den Slot **Loeschen** der Klasse **Fehler**, bewirkt das Löschen aller existierenden Fehlerinstanzen. Diese Aktion ist zur Initialisierung bei einer neu konfigurierten Diagnose nötig. Die Slots **Display**, **Test** und **Bilde\_Instance** müssen in allen abgeleiteten Fehlerklassen vorhanden sein und werden aus diesem Grund in der subsumierenden Klasse vereinbart. Werte können hier natürlich nur Defaultfunktionen sein, die im Grunde keine Rolle spielen. In den abgeleiteten Klassen müssen diesen Slots dann fehlerspezifische Werte zugeordnet werden. Der **Bilde\_Instance**-Slot muß eine Funktion enthalten, die abhängig von der jeweiligen Fehlerklasse als Konstruktor für Fehlerinstanzen fungiert. Wird im Verlauf der Diagnose ein Fehler erkannt, so wird über diese Methode eine Instanz erzeugt. Der **Test**-Slot muß eine Funktion enthalten, die einen Fehler im Faktenwissen erkennen kann oder die Fehlererkennung zum Beispiel mit Hilfe des Regelsystems anstößt. Dieser Slot entspricht der in Kapitel 6.2 angesprochenen Assertkomponente. Der **Display**-Slot muß eine Funktion enthalten, die der Erklärungskomponente entspricht, z.B. eine textuelle Erklärung im **Fehler\_Display\_Fenster** des Moduls DFMmeta, kombiniert mit dem Leiterplatteneditor zur Visualisierung des Fehlers. Als ein Beispiel zu den einzelnen Rule-Checks sollen die zusätzlichen Werte der Subklasse Bauelement-Bauelement Abstand aufgezeigt werden:

Klasse BE\_BE\_Abstand:

Slot	Werte
be_be_min_dist	?
be_namen	?
seite	?
pos	?

Der Slot **be\_be\_min\_dist** enthält, nach dem Einlesen einer Leiterplatte den für die Fehlerklasse relevanten Vergleichswert, eingetragen durch den Monitor **Stationen\_be\_be\_dist\_monitor\_1** aus **DFMrumpf**. Ist der Abstand zweier Bauelemente kleiner als der Vergleichswert, so werden deren Namen im Slot **be\_namen** der erzeugten Instanz eingetragen. Als weitere Information werden beim Anlegen der Instanz die Positionen der betroffenen Bauelemente im Slot **pos** (zwei zweistellige Listen (x-Wert, y-Wert)) und die Seite der Leiterplatte, auf welcher die Bauelemente bestückt wurden, im Slot **seite** eingetragen.

Das wesentliche Merkmal der vorgestellten Architektur ist die leichte Erweiterbarkeit hinsichtlich neuer Manufacturing-Rule-Checks zur Anpassung an die unterschiedli-

chen Fertigungsbelange. Die hierzu notwendigen Schritte können in drei Phasen gegliedert werden:

- Formalisierung der Regel, wie in Kapitel 6.2 beschrieben
- Ableiten einer neuen Fehlerklasse aus der Top-Level Klasse Fehler in **DFMfehler**
- Eventuell Erweiterungen des Datenmodells für die Faktenwissensrepräsentation in **DFMrumpf**

#### 6.4 Ablauf und Visualisierung des Design-Checks

Zur Durchführung einer Diagnose bestimmt der Anwender in einem aus **DFMmeta** gesteuerten Dialogfenster die zu untersuchende Leiterplatte und das technische Equipment. Die erforderlichen Daten werden aus der zentralen Datenbank des Planungssystems eingelesen und dem DFM-System zur Verfügung gestellt.

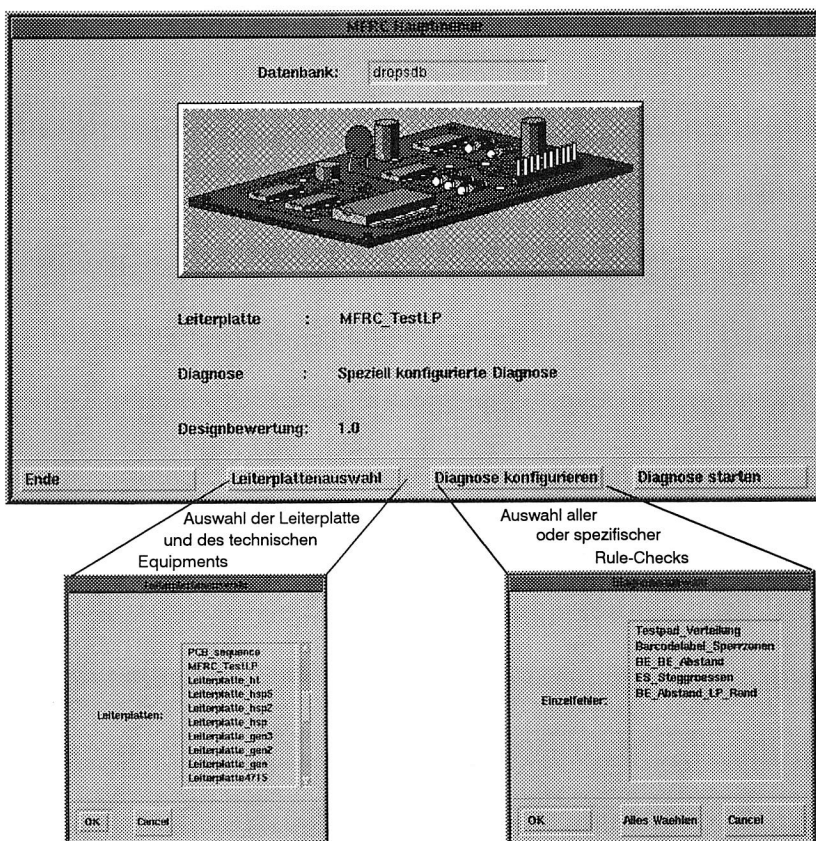


Bild 6-9: Start und Konfiguration des Diagnoselaufs

Des weiteren hat der Benutzer die Möglichkeit, die Diagnose zu konfigurieren. Dies bedeutet, er kann alle oder ausgewählte Rule-Checks spezifizieren, nach denen das Leiterplattendesign untersucht werden soll.

Im nächsten Schritt werden entsprechend der objektorientierten Repräsentation des Faktenwissens aus **DFMrumpf** alle relevanten Instanzen zu den eingelesenen technischen Parametern der Leiterplatte und des technologischen Equipments angelegt. Der Anwender hat hierbei die Möglichkeit, die Instanzenbildung und damit die datentechnischen Zusammenhänge für Leiterplatte und Equipment auf dem object browser mitzuverfolgen.

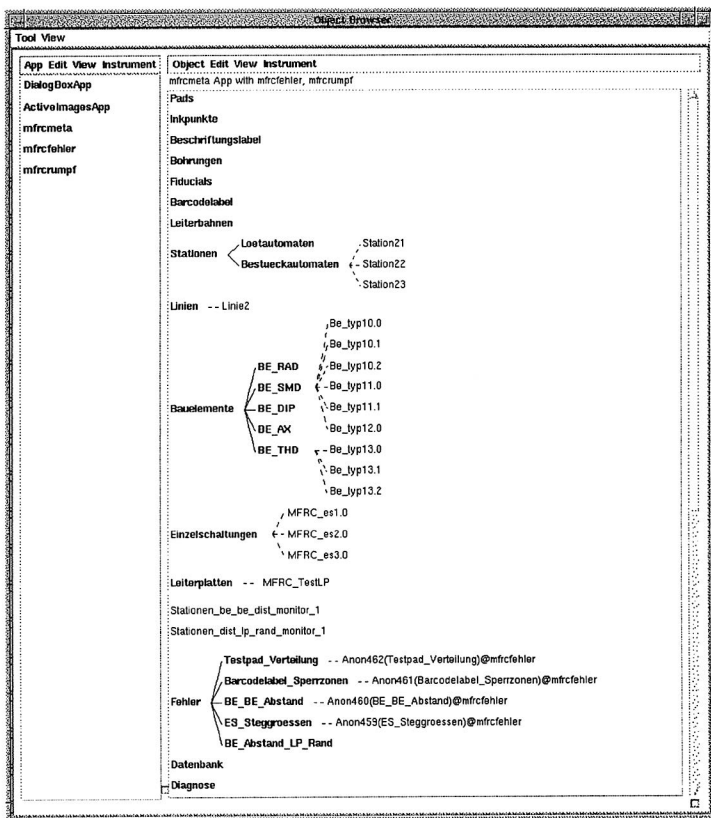


Bild 6-10: Objectbrowser zur Instanzenbildung des Faktenwissens

Nach dem erfolgreichen Aufbau der Instanzen aus **DFMrumpf** werden alle konfigurierten Rule-Checks durchlaufen. Wird eine Regelverletzung entdeckt, wird eine Fehlerinstanz hierzu angelegt. Nach Beendigung aller Rule-Checks werden mit dem

Displayslot der Fehlerinstanzen die gefundenen Designverletzungen ausgegeben. Dies geschieht zum einen mit einem Messagefenster, das den Fehlertext, evtl. Positionswerte, Typnamen etc. enthält, und zum anderen mit einem entwickelten Leiterplatteneditor, der einen grafischen Bezug des Fehlers zur Leiterplatte ausgibt. Die in Kapitel 6.2 beschriebene Gütebewertung für das untersuchte Design wird auf dem Startfenster in das dafür vorgesehene Feld eingetragen.

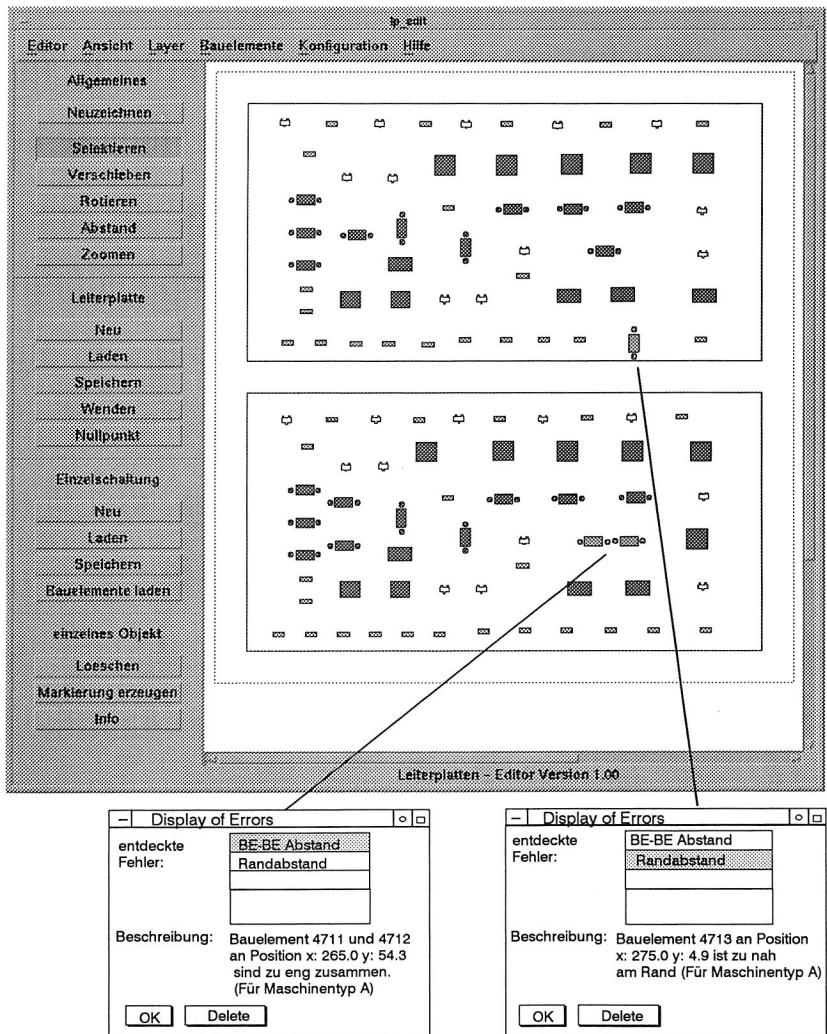


Bild 6-11: Erklärung und Visualisierung gefundener Fehler



## **7      *Entwicklung eines Modules zur Rüst- und Umrüstoptimierung***

Gegenstand des folgenden Kapitels ist die Analyse der vorliegenden Optimierungsprobleme in der Rüst- und Umrüstoptimierung und die Entwicklung geeigneter Lösungsstrategien. Die Aufgabe besteht darin, die Summe an Nebenbedingungen zu filtern, das Problem zu abstrahieren und in ein mathematisches Modell umzuformen. Auf dieses Modell können dann bekannte Optimierungsverfahren oder maschinen-nahe Eigenentwicklungen angewandt werden.

### **7.1    *Einteilung der Optimierungsprobleme***

Die technische Welt erfordert sehr oft Entscheidungen, deren meist weitreichende Auswirkungen zum Zeitpunkt der Entscheidung oft nur sehr schwer zu überschauen sind. Entscheidungsfreudige Personen sind demzufolge sehr gefragt, tragen aber auch oft ein sehr großes Verantwortungsrisiko. Die Aufgabe besteht meist darin, im Rahmen seiner Entscheidungsmöglichkeiten, alle möglichen Alternativen abzuwägen und die Beste auszusuchen.

Entscheidungsfindungen im technisch-wirtschaftlichen Bereich bezeichnet man auch als sogenannte Operations Research-Probleme. Eine mögliche Charakterisierung dieser Problemstellungen lautet: Operations Research bedeutet die Suche nach einer bestmöglichen (optimalen) Entscheidung unter Berücksichtigung von Nebenbedingungen [76].

Fragestellungen im OR-Bereich begegnen uns sowohl im täglichen Leben als auch im Bereich der Elektronikproduktion. So ist die Optimierungsaufgabe, in den Weihnachtsferien alle im Land verstreuten Verwandten in einem vorgegebenen Zeitraum einmal zu besuchen und möglichst wenige Kilometer zurückzulegen vergleichbar mit dem Problem einen Pick and Place-Bestückungsautomaten so zu programmieren, daß der Bestückkopf zwischen den Abholpositionen und Bestückkoordinaten auf der Leiterplatte den geringsten Weg zurücklegt.

Ein entscheidender Teil der Aufgabe besteht darin, die Nebenbedingungen zu abstrahieren und das eingebettete Problem zu erkennen, um hierfür Lösungsverfahren zu generieren. Die Fülle an Nebenbedingungen (z.B. beschränkte Aufnahmekapazität der Bestückungssysteme, unterschiedliche Ausmaße der Förderer, feste Pipettenzuordnung zu den Bauelementen etc.) ist häufig so groß, daß das Abprüfen und Abbilden aller Randbedingungen die eigentliche Optimierungsaufgabe an Aufwand übersteigt.

Eine Klassifizierung der Optimierungsprobleme und -methoden kann auf unterschiedliche Arten erfolgen. Gegliedert werden sie z.B. nach ihrer zeitlichen Abhängigkeit [80], d.h. danach, ob die Aufgabe einer zeitlichen Änderung unterliegt, in zwei Sparten:

- diskrete und kombinatorische Optimierung
- dynamische Optimierung

Bei der diskreten Optimierung geht es um die Maximierung bzw. Minimierung einer Güte-/Ziel-/Kostenfunktion für mathematisch modellierte Probleme, welche von der Anordnung (Permutation) oder Zusammenstellung (Kombination) gegebener Elemente abhängig ist [81]. In abstrakter mathematischer Form ausgedrückt gilt für alle diskreten Optimierungsprobleme:

$$\min (\max) f(x) \quad x \in F$$

wobei  $f$  die Gütefunktion,  $x$  die Optimierungsvariablen und  $F$  den möglichen Raum der Variablen darstellen.

Die dynamische Optimierung behandelt Probleme, denen ein in der Zeit ablaufender Prozeß zugrunde liegt. Des öfteren wird auch noch der Begriff stochastische Optimierung verwendet. Diese Bezeichnung verwendet man, wenn Entscheidungsvariablen, die in der Praxis mit großen Unsicherheiten verbunden sind, mit Zufallswerten belegt werden.

Hinsichtlich des Berechnungsaufwandes unterscheidet man verschiedenen Problemklassen. Man spricht von sogenannten P-Problemen, wenn der Berechnungsaufwand (Rechenzeit und Speicherplatz) in Abhängigkeit der Anzahl der Elemente durch ein Polynom begrenzt wird. Ein Nicht Polynomiales Problem liegt vor, wenn der Rechenaufwand steiler als polynomial (z.B. exponentiell oder mit der Fakultät) ansteigt. P-Probleme werden nach dieser Klassifizierung auch als leicht, NP-Probleme als schwer lösbar bezeichnet [55].

Einige Beispiele, wie die Rechenzeit in Abhängigkeit der Rechenoperationen wächst, zeigt Tabelle 7-1. Auf Basis einer RISC Workstation mit ca. 60 MIPS (Millionen Instructions Per Second) wurde für eine Rechenoperation 10 ns angenommen. Zur Abschätzung des Problems wurde idealisierter Weise die Anzahl der Elemente gleich der Rechenoperationen gesetzt. In der Praxis werden mehr Rechenoperationen benötigt als Elemente vorhanden sind.

Anzahl Rechenoperationen Rechenaufwand	20	50	100	1000	
$n$	$2 \times 10^{-7} \text{ s}$	$5 \times 10^{-7} \text{ s}$	$10^{-6} \text{ s}$	$10^{-5} \text{ s}$	P-Probleme
$n^2$	$4 \times 10^{-6} \text{ s}$	$2,5 \times 10^{-5} \text{ s}$	$10^{-4} \text{ s}$	$10^{-3} \text{ s}$	
$n^3$	$8 \times 10^{-5} \text{ s}$	$1,25 \times 10^{-3} \text{ s}$	$10^{-3} \text{ s}$	10	
$2^n$	$10^{-3} \text{ s}$	130 Tage	$4 \times 10^{14}$ Jahre	$339,7 \times 10^{282}$ Jahre	NP-Probleme
$n!$	770 Jahre	$10^{49}$ Jahre	$3 \times 10^{142}$ Jahre		

Tabelle 7-1: Abschätzung des Rechenaufwands

### **7.1.1 Bekannte modellierte Optimierungsprobleme der diskreten Optimierung**

Die Problemfälle der Rüst- und Umrüstoptimierung befinden sich ausschließlich im Bereich der ganzzahligen Optimierung. Im Rahmen dieser Optimierungsfälle gibt es eine Reihe unterschiedlicher Problemfälle, die teilweise die Forscher und Entwickler schon über zwei Jahrhunderte beschäftigen. Einige Beispiele zu relevanten Optimierungsproblemen sollen im folgenden kurz vorgestellt werden.

#### **Traveling Salesman-Problem**

Eines der bekanntesten Beispiele für ein NP-Problem ist das TSP-Problem, auch als Problem des Handlungsreisenden bekannt. Die Aufgabe des Handlungsreisenden besteht darin, eine Reihe von Städten zu besuchen und hierbei die kürzeste Reiseroute zu wählen, wobei jede Stadt nur einmal in der Reiseroute vorkommen darf. Als Zielfunktion fungiert in diesem Fall die Gesamtwegstrecke, die es zu minimieren gilt. Ein möglicher Anwendungsfall dieses Problems für die Elektronikproduktion ist die Bestimmung der Bestückreihenfolge der Bauelemente auf den Leiterplatten.

#### **Rucksackproblem**

Eine weitere typische Anwendung der diskreten Optimierung ist das sogenannte Rucksack-Problem, bei dem es darum geht, einen Rucksack mit möglichst viel Volumen bei gegebener Gewichtsbeschränkung zu füllen. Die Volumina und Gewichte der Füllelemente sind bekannt. Gesucht ist eine optimale Rucksackfüllung, d.h. diejenige Zusammenfassung von Gegenständen, bei der die Zielfunktion Volumen ihren maximalen Wert erreicht, das maximal erlaubte Gewicht aber nicht überschritten wird. Eine mögliche Anwendung ist das Zusammenfassen von Aufträgen zu Chargen, ohne dabei die Gesamtkapazität eines Bestückungssystems zu überschreiten. Ziel dabei ist, die Anzahl an Chargen zu minimieren.

#### **Verschnittprobleme**

Ein weiterer Problemfall industrieller Fertigungsvorgänge ist das Zuschneiden von Material. Gesucht ist ein sogenannter Verschnittplan, bei dem der Verschnitt minimiert wird oder die kleinste Menge an Ausgangsmaterial benötigt wird. Nach Anzahl der Abmessungen des Materials unterscheidet man ein-, zwei- und dreidimensionale Verschnittprobleme.

Eindimensionale Verschnittprobleme treten z.B. beim Sägen von Holzlatten auf, wobei nur die Länge der Einzelstücke von Interesse ist. Bei zweidimensionalen Verschnittproblemen werden Flächenstücke zugeschnitten, so daß Länge und Breite zu berücksichtigen sind. Das zweidimensionale Verschnittproblem ist in der Praxis am häufigsten zu finden, z.B. beim Zuschneiden von Blechtafeln. Im Elektronikbereich kann dieses Problem bei der Nutzenanordnung von Leiterplatten auftreten.

## Maschinenbelegungsplanung (Scheduling)

Schedulingprobleme befassen sich mit der optimalen Einplanung von Aufträgen, die auf verschiedenen Maschinen zu bearbeiten sind. Die zentrale Frage der Maschinenbelegungsplanung ist, wann welcher Auftrag auf welcher Maschine bearbeitet werden soll, so daß eine bestimmte Zielfunktion (meist die Gesamtdurchlaufzeit) minimiert wird [121].

### 7.1.2 Optimierungsverfahren

Alleine für das Traveling Salesman Problem, existieren eine Reihe bekannter Lösungsverfahren (s. Bild 7-1).

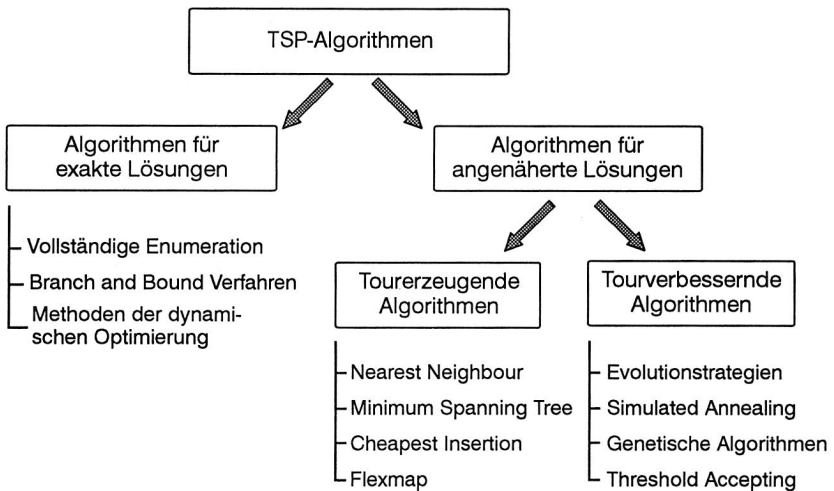


Bild 7-1: Optimierungsverfahren für TSP Probleme [56]

Eine Möglichkeit, TSP-Probleme zu lösen, wäre der Einsatz von Algorithmen, die zu exakten Lösungen führen. Diese sind jedoch aufgrund des hohen Aufwands nur in kleinen Lösungsräumen anwendbar.

#### Vollständige Enumeration

Die wohl aufwendigste Methode zur Lösung eines Optimierungsproblems stellt die vollständige Enumeration dar. Hierzu werden alle möglichen Kombinationen berechnet und die beste gewählt. Fokussiert man diese Methode auf das Problem der Bestücksequenzoptimierung, so ist man bei einer Leiterplatte mit 20 Bestückpositionen, bei heutiger Rechnerleistung, bereits bei einem Rechenaufwand von 770 Jahren. Demzufolge ist diese Methode für die vorhandenen Aufgabenstellungen nicht bzw. nur in einigen Spezialfällen einsetzbar.

### **Implizite Enumeration/Branch and Bound Verfahren**

Bei diesem Optimierungsverfahren wird versucht, sukzessive Teilmengen des Lösungsraums zu finden, in denen keine optimale Lösung zu erwarten ist, und diese auszusondern. Dies geschieht mit Hilfe eines Suchbaumes, der vom Wurzelknoten aus stufenweise in disjunkte Teilmengen aufgeteilt wird. Für jede dieser Mengen wird eine Kostenuntergrenze berechnet, die die Mindestkosten für alle in der Untermenge enthaltenen Lösungen angibt. Das Aufspalten in Teilmengen wird solange fortgesetzt, bis eine Lösung bekannt ist, deren Kosten nicht größer sind als die Kostenuntergrenze aller anderen nicht unterteilten Mengen. Diese Lösung ist dann die Optimallösung.

Der Name des Verfahrens kommt daher, daß das Aufspalten einer Lösungsmenge als Branching und das Berechnen der Grenze als Bounding bezeichnet wird. In ungünstigen Fällen kann dieses Verfahren zu einer vollständigen Enumeration führen, so daß diese Methode nicht immer praktikabel erscheint.

Da das TSP-Problem zur Klasse der NP-Probleme zu rechnen ist, wurden einige Algorithmen entwickelt, die in polynomialer Zeit suboptimale Lösungen liefern. Mit der Wahl einer Heuristik verbindet man den Wunsch, möglichst schnell ein nahezu optimales Ergebnis zu finden. Eine Ergebnisabschätzung bezogen auf das tatsächliche Optimum ist in den meisten Fällen unmöglich, die Endergebnisse können aber untereinander verglichen werden. Einzig und allein die Laufzeit kann man in vielen Fällen im voraus abschätzen.

Der Bereich der nicht exakten Lösungen wird in Eröffnungsverfahren (tour-erzeugende Algorithmen) und suboptimale Iterationsverfahren (tour-verbessernde Algorithmen) unterteilt. Im folgenden sollen einige bekannte Eröffnungsverfahren vorgestellt werden, wobei hier kein Anspruch auf Vollständigkeit erhoben wird.

### **Nearest Neighbour**

Eine der ältesten Methoden ist das Verfahren des besten Nachfolgers (Nearest Neighbour). Bei diesem Verfahren beginnt man mit einem beliebigen Ort, sucht für diesen den nächst gelegenen Folgeort und für diesen wiederum den besten Nachfolger unter den noch nicht verplanten Orten. Dies wird fortgesetzt bis zum letztem Ort.

### **Minimum Spanning Tree**

Diese Methode aus dem Bereich der Graphentheorie sucht nach dem sogenannten minimalen Spannbaum. Dieser wird gebildet, in dem man alle Knoten eines gewichteten Graphen so verbindet, daß die Summe der Kantengewichte wenigstens ebenso klein ist wie die Summe der Gewichte jeder beliebigen anderen Menge von Kanten, die alle Knoten verbindet [98]. Um nun von dem Baum auf eine für das TSP-Problem benötigte Tour zu kommen, durchforstet man mittels der "deep first" Methode den Baum von der Wurzel aus. An den Blättern des Baumes kehrt man auf demselben Pfad zurück bis zur Verzweigung eines unbehandelten Teilbaumes. In ei-

nem dritten Schritt eliminiert man die Rückwege, die entlang schon besuchter Städte liegen, indem man sich Abkürzungen sucht (s. Bild 7-2).

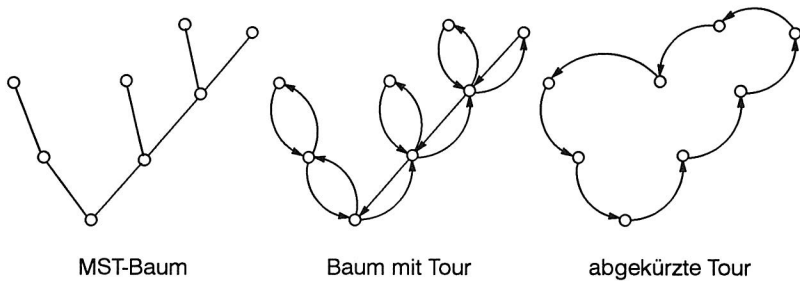


Bild 7-2: Vorgehensweise beim Minimum Spanning Tree

### Sukzessive Einbindung von Städten (Cheapest Insertion)

Dieses Verfahren beginnt mit einer Tour, die zwei in der Kostenmatrix benachbarte Städte, z.B. A und B verbindet. Zu diesen bildet man einen Zyklus  $A \rightarrow B \rightarrow A$  mit den entsprechenden Kosten. Nun sucht die Tour die nächstgelegene Stadt, welche daraufhin an die bestmögliche Stelle eingefügt wird, so daß entweder der neue Zyklus  $A \rightarrow C \rightarrow B \rightarrow A$  oder  $A \rightarrow B \rightarrow C \rightarrow A$  entsteht. Im Gegensatz zum Nearest Neighbour steigt mit dem Fortschritt des Verfahrens die Anzahl der Freiheitsgrade.

### Flexmap

Dieser Algorithmus beruht auf dem Modell von Kohonen, allerdings mit einer wachsenden Ringstruktur [51, 52]. Initial beginnt man bei einem Ring mit drei Zellen, wobei jede Zelle durch einen zweidimensionalen Positionsvektor beschrieben wird (s. Bild 7-3). Sukzessive werden dann neue Zellen eingefügt, wobei zwei Bedingungen beachtet werden:

- Die neue Position der Zelle ist in der Nähe von anzufahrenden Städten
- Nachbarzellen haben ähnliche Positionen

Die Vorgehensweise bei dem Verfahren der wachsenden Zellstrukturen ist wie folgt:

- Suche per Zufall eine Stadt c als Eingabesignal
- Die neu gebildete Zelle s fungiert als Erregerzentrum
- Adaption der neuen Zelle und ihrer topologischen Nachbarn in Richtung des Eingabesignals

Die wachsenden Zellen kommen bei fortschreitender Laufzeit des Algorithmus an die Gesamtzahl der zu verbindenden Städte heran. Wenn für jede Stadt die am nächsten gelegene Zelle eine andere ist, ist die Rundtour durch die Städte eindeutig durch die Zellen in der Ringstruktur definiert. Hervorzuheben bei diesem Algorithmus ist, daß der Rechenaufwand linear mit der Problemgröße wächst.

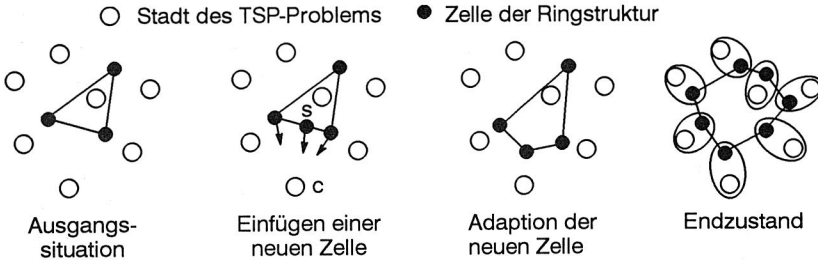


Bild 7-3: Wachsende Zellstrukturen am Beispiel des TSP Problems [32]

Die suboptimalen Iterationsverfahren starten mit einer bestehenden Tour, die aus einem der oben genannten Verfahren übernommen oder auch zufällig bestimmt werden kann. Durch Permutation von einzelnen Städten oder Gruppen wird versucht die bestehende Route zu verändern [37].

### Kanten-Permutationverfahren

Das bekannteste Verfahren von Lin und Kernighan [60] wählt eine variable Zahl von Kanten (Wege) zwischen Knoten (Städten). Innerhalb der Paare werden die Knoten ausgetauscht und die Kosten der neu entstandenen Kanten mit den ursprünglichen verglichen. Das von Lin und Kernighan vorgeschlagene Verfahren läßt die Anzahl der zu vertauschenden Kanten variabel. Man grenzt die Suche in der Regel aber auf zwei bis drei Paare ein.

### Städtefolge-Permutation

Bei diesem Verfahren, das der Kanten-Permutation sehr ähnlich ist, werden ganze Städtefolgen (Strings) bestimmt, ausgeklint und an anderer Stelle plaziert. Der Vergleich zwischen alten und neuen Kosten bestimmt die Einfügestelle. Im Lauf des Algorithmus wird zusätzlich die Länge der Kette verändert.

Die bisher vorgestellten Näherungsmethoden sind Verfahren, die sich nach deterministischen und probalistischen Regeln an das Optimum herantasten. Die Zielorientierung geschieht nach bestimmten Regeln, wobei Schrittweite und -richtung mit Hilfe einer Metrik definiert werden.

### Evolutionsstrategien

Angelehnt an die äußerst flexible Optimierungsstrategie der Natur, die Evolution, wurden vor einigen Jahren von Rechenberg [86] Mutations-Selektions-Strategien erfolgreich bei technischen Problemstellungen angewandt. Dieses wirkungsvolle Prinzip der biologischen Evolution hatte bereits vor mehr als einem Jahrhundert Charles Darwin erkannt. Er postulierte als treibende Kraft der Artenentwicklung das Wechselspiel von Mutation und Selektion. Übertragen auf technische Probleme bedeutet dies, daß man an einer existierenden Lösung eine beliebige Änderung (Mutation) vollzieht und das Ergebnis überprüft. Ergab sich eine Verschlechterung der Lösung

wird der Schritt wieder rückgängig gemacht, ansonsten wird die neue Lösung Ausgangspunkt für den nächsten Schritt (Selektion). Die Schritte selbst unterliegen einem gewissen Zufallsprozeß, der sowohl gewisse Schrittweiten als auch Schrittrichtungen bevorzugt [3].

Am Beispiel des TSP kann man neue Reihenfolgen auf zwei unterschiedlichen Arten von Mutationen erzeugen (s. Bild 7-4). Es besteht die Möglichkeit, Knoten neu zu positionieren (**P**ositions-Mutation) oder die Verbindungen zwischen den Knoten, die Kanten zu ändern (**K**anten-Mutation).

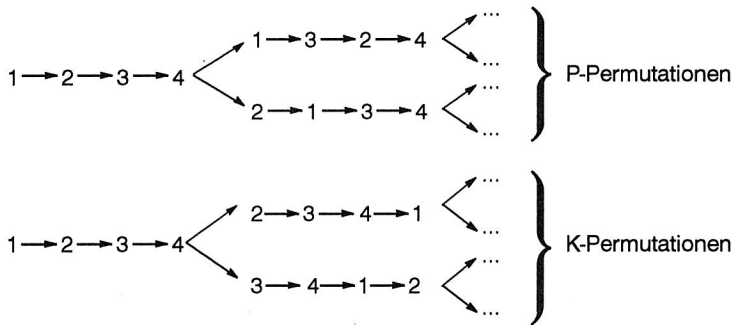


Bild 7-4: Mutationsarten für das TSP-Problem

Eine Eigenschaft der Evolution besteht darin, sich einem (Sub-) Optimum möglichst schnell zu nähern, wobei es zunächst kaum eine Möglichkeit gibt, diesem zu entkommen, da alle Mutationen, die von diesem Maximum (Minimum) wegführen, einen schlechteren Zielfunktionswert haben als das bisherige Optimum. Daher ist es notwendig, für eine gewisse (eventuell zufällig ermittelte) Anzahl von Mutationen auch "schlechtere" Mutationen zu selektieren.

Ein Suboptimum kann daran erkannt werden, daß es kaum noch oder gar keine neuen erfolgreichen Mutationen gibt, was an den Zählern für die einzelnen Mutationsarten abgelesen werden kann. Danach ist es am einfachsten sich das bisher beste Ergebnis zu merken und eine Phase einzuleiten, in der wahllos einige Mutationen durchgeführt werden, ohne dabei auf Verbesserung der Zielfunktion zu achten. Anschließend dürfte das Verfahren bei genügender Abweichung vom bisherigen Optimum auf ein anderes Sub- oder sogar das absolute Optimum zusteuern. Die Wechsel zwischen Stabilisierungs- (Annähern an Suboptima) und Destabilisierungsphasen (Zulassen von Verschlechterungen innerhalb bestimmter Grenzen) sind charakteristisch für Evolutionsstrategien (s. Bild 7-5).



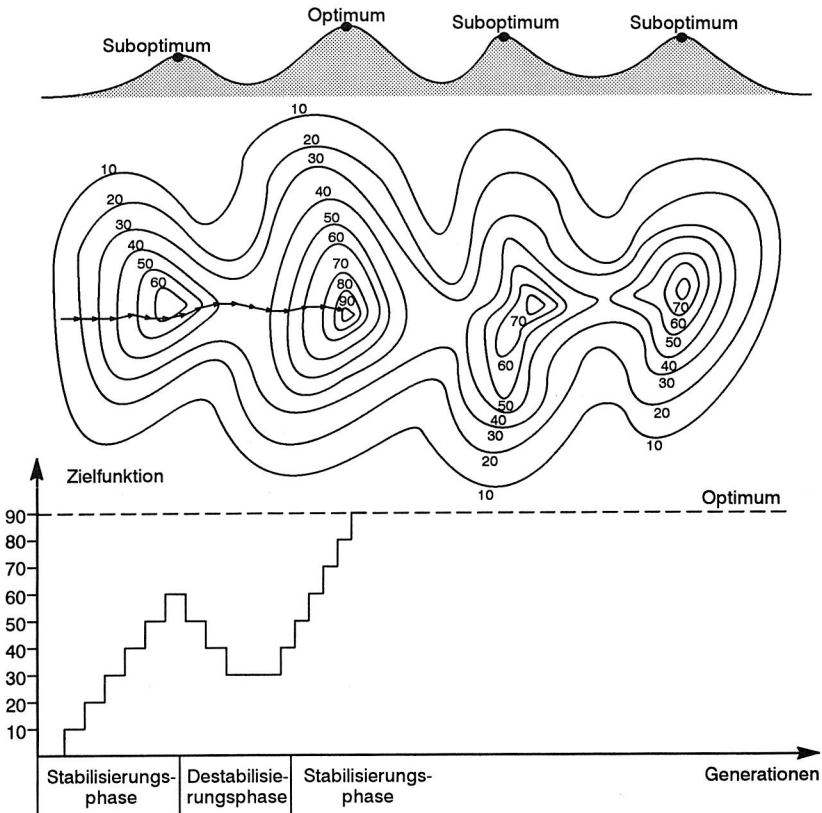


Bild 7-5: Stabilisierungs- und Destabilisierungsphasen der Evolutionsstrategie

### Simulated Annealing

Ein noch sehr junges Optimierungsverfahren stellten in den frühen 80' er Jahren Kirkpatrick, Gelatt und Vecchi [49, 50] und unabhängig davon, Cerny [14] vor. Die Grundidee dieses Optimierungsverfahrens besteht in einer Analogie zur Abkühlung und kristallinen Erstarrung von Flüssigkeiten in der Thermodynamik (Annealing = langsames Abkühlen ausgehend von einer hohen Anfangstemperatur) [1].

Bei hoher Temperatur bewegen sich die Moleküle einer Flüssigkeit sehr zahlreich und frei. Diese thermische Beweglichkeit verliert die Flüssigkeit im Laufe der Abkühlung. Je langsamer die Abkühlung vonstatten geht, desto größer ist die Wahrscheinlichkeit, daß sich kristalline Strukturen ausbilden, die sich durch große Ordnung in alle Richtungen über viele Moleküle hinweg auszeichnen. Diese Kristalle bilden den Zustand minimaler Energie für dieses System [6]. Wenn ein flüssiges Metall schnell

abgekühlt wird, dann erstarrt es zu einem polykristallinen oder amorphen Zustand mit höherer Energie.

Zur Erlangung des Zustandes minimaler Energie ist in jeder Temperaturstufe thermodynamisches Gleichgewicht notwendig, welches besagt, daß die Energiezustände der Boltzmann-Wahrscheinlichkeitsverteilung genügen:

$$P(E) \sim e^{(-E/kT)}$$

mit:  $P(E)$  = Wahrscheinlichkeit für Energiezustand  $E$   
 $E$  = Energie  
 $T$  = Temperatur  
 $k$  = Boltzmannkonstante

Aus diesem Prinzip entstand der Metropolis-Algorithmus [79], der besagt: Jede Energieerniedrigung wird mit der Wahrscheinlichkeit 1, jede Energieerhöhung mit der Wahrscheinlichkeit  $e^{(-E/kT)}$  vom System übernommen. Zur Vereinfachung setzt man  $k = 1$  und  $\Delta E$  bzw.  $T$  dimensionslos. Der so entstandene Algorithmus wird auch Orakel-Befragung genannt [1].

Um den Bezug zu einem kombinatorischen Optimierungsproblem herzustellen, trifft man man folgende Vereinbarungen:

- Lösungen eines kombinatorischen Optimierungsproblems sind äquivalent zu den Stati eines physikalischen Systems.
- Die Kosten einer Lösung sind äquivalent zum Energiestatus

Des weiteren wird ein sogenannter Kontrollparameter eingeführt, der die Rolle der Temperatur übernimmt. Der eigentliche Simulated Annealing-Algorithmus kann nun als Sequenz von Metropolis-Algorithmen betrachtet werden, die bei einer sinkenden Folge von abnehmenden Werten des Kontrollparameters  $T$  ausgeführt werden.

### Genetische Algorithmen

Die von Holland [43] entwickelten genetischen Algorithmen sind Suchalgorithmen, die das Adaptionverhalten biologischer Systeme simulieren. Sie stützen sich hierbei, wie auch schon die vorgestellten Evolutionsstrategien, auf folgende Arbeitsschemen der Natur:

- Survival of the fittest (Selektion)
- Zufallsgestützter Informationsaustausch (sexuelle Reproduktion)

Das erste Schema bestimmt, welche Informationen einer Population überleben, letzteres ermöglicht einen Austausch genetischer Information und kombiniert Gene der Eltern zu neuen Individuen.

Unterscheidungsmerkmale von genetischen Algorithmen zu Evolutionsstrategien sind die Art der Codierung und die Möglichkeit, neben reiner Mutation und Selektion auch Kreuzungen zwischen Genstrings durchzuführen [94]. Die Codierung stellt somit auch einen wesentlichen Parameter der Güte eines genetischen Algorithmus dar. Die Codierung erfolgt in Strings, wobei jeder String die Repräsentation eines Chromosoms darstellt. Ein Gen ist dementsprechend ein Bereich des Strings, der für ein Merkmal des Individuums (Lösung des Optimierungsproblems) spezifisch ist. Die Merkmalswerte, die ein String aufnehmen kann, bezeichnet man als Allele und die Stringposition als Locus. Die Populationsgröße bezeichnet die Anzahl der vorliegenden Strings einer Population. Die Bewertungsfunktion zur Aussage über die Lebensfähigkeit eines Individuums wird im Sprachgebrauch der genetischen Algorithmen als Fitneßfunktion bezeichnet. Allgemein gilt für genetische Algorithmen:

Genstring  $s$ :  $s \in V^m$       Länge des Strings:  $m \in \mathbb{N}$   
 Alphabet:  $V$  (z.B.  $V \in \{0, 1\}$ )  
 Population  $p$ :  $p \in s^n$       Population:  $n \in \mathbb{N}$

Für folgendes einfache Beispiel:

Stringlänge 6 ( $s=6$ ),

Populationsgröße 4 ( $n=4$ ),

binäres Alphabet, d.h.  $V \in \{0, 1\}$ ,

Optimierungsziel: Finde das Maximum der Funktion  $f(x) = x^2$

ergeben sich folgende Codierung und Fitneßwerte

	Stringposition 1 2 3 4 5 6	Decodierter Wert	Fitneßwert
Genstring 1	1 0 0 1 0 1	37	1369
Genstring 2	0 0 1 1 0 1	13	169
Genstring 3	0 1 1 1 0 0	28	784
Genstring 4	1 1 0 0 0 1	49	2401

Dekodierung und Fitneßberechnung z.B. für Genstring 1:

$$1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 37 \Rightarrow 37^2 = 1369$$

Zur Simulation der Vererbungsvorgänge verwenden genetische Algorithmen vier Grundtypen von Operatoren:

- Reproduktion
- Kreuzung
- Mutation
- Inversion

Bei der **Reproduktion** werden entsprechend der Populationsgröße Individuen ausgewählt. Das hierzu hauptsächlich verwendete Heiratsschema (Roulette Wheel) wählt die Kandidaten für die Erzeugung von Nachkommen mit einer Wahrscheinlich-

keit proportional zu ihrer Fitness aus. Dadurch soll erreicht werden, daß Nachkommen hochbewerteter Elemente einer Population in der nächsten Generation mit höherer Wahrscheinlichkeit vertreten sind als durchschnittliche Elemente [35]. Durch dieses Prinzip erhalten aber auch Individuen mit geringerer Fitness die Möglichkeit, sich fortzupflanzen. Zur Durchführung einer "Roulette Wheel Selection" werden alle Strings der Population auf 100% skaliert. Somit ergibt sich eine relative Fitness, die man als Sektor einer Kreisscheibe interpretieren kann. Um jetzt Individuen mit einer Verteilung zu ermitteln, die der Struktur der Fitnessverteilung entspricht, werden Zufallswerte erzeugt, die den gesamten Kreissektor überstreichen. Werden die Strings für die Bildung der neuen Population herangezogen, in deren Kreissektor die jeweiligen Zufallswerte fallen, so ergibt sich eine der Fitnessfunktion entsprechende Verteilung. Bild 7-6 zeigt ein Beispiel für drei Strings.

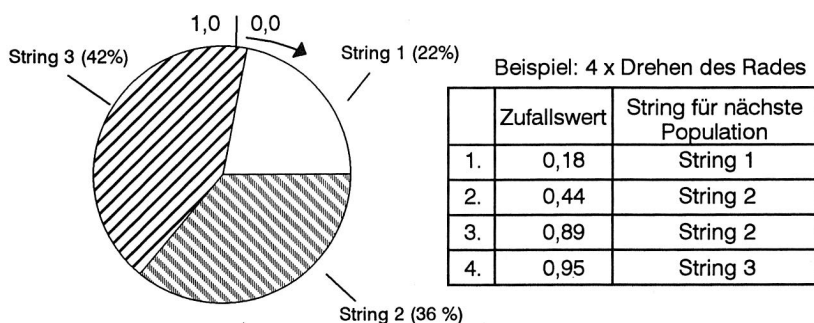


Bild 7-6: Roulette Wheel-Auswahl

Nach der Auswahl der Parentstrings findet der **Kreuzungs**-Prozeß statt, um neue Strings zu erzeugen. In der Theorie wurden verschiedene Operatoren vorgestellt und deren Einfluß für die evolutionäre Entwicklung untersucht [102]. Ein relativ einfacher Operator ist der "Single Crossover" (s. Bild 7-7). Bei diesem Operator wird zunächst eine Crossoverposition im Bereich der Stringlänge zufällig ermittelt, an der dann der eigentliche Crossover stattfindet. Dazu wird der Teil des ersten Parentstrings vom Anfang bis zu der Crossoverposition mit dem Stringteil des zweiten Strings, der von der Crossoverposition bis zum Ende des Strings reicht, zusammengesetzt. Danach ergeben sich zwei neue Strings, die teilweise weiterhin die Merkmale der Parentstrings tragen.

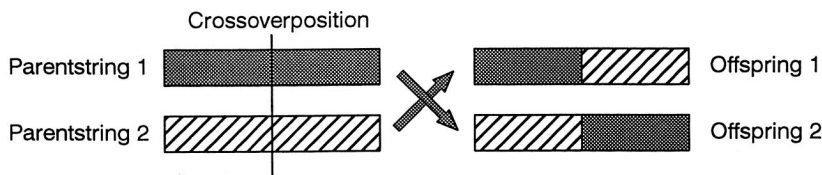


Bild 7-7: Single Crossoveroperator

Die **Mutation** stellt einen eher sekundären Operator dar. Mit einer bestimmter Wahrscheinlichkeit  $P$  wird für jede einzelne Position im Genstring ausgelost, ob ihr Wert mutiert wird oder gleichbleibt. Mutation bedeutet bei einer binären Codierung einen Wechsel der Allelewerte an einem bestimmten Locus von 0 auf 1 oder umgekehrt. Die Mutationsrate ist sowohl in der Natur als auch bei genetischen Algorithmen eher niedrig, um nicht die Gesamtpopulation durch sprunghafte Veränderungen zu gefährden.

Ein weiterer Operator zur Verbesserung der Konvergenzeigenschaft ist der **Inversionsoperator**. Hierbei wird ein Teilbereich eines Strings zwischen zwei Zufallszahlen invertiert eingefügt.

Weiterentwicklungen von Crossoveroperatoren, z.B. durch den Ansatz, die Operatoren Crossover und Inversion zu verbinden, führte zu einer Reihe weiterer Operatoren wie:

- partially matched crossover,
- order crossover,
- cycle crossover,

auf die im einzelnen nicht näher eingegangen wird. Erst nach der Analyse des Optimierungsproblems kann man entscheiden, welcher Operator erfolgversprechend eingesetzt werden kann oder ob eventuell ein spezieller, der Optimierungsaufgabe entsprechender Operator generiert werden muß.

Zusammenfassend ergibt sich für einen genetischen Algorithmus folgendes Grundschema (s. Bild 7-8).

Da genetische Algorithmen allein durch die Anwendung der Funktionen Auswahl, Kreuzung und Mutation blind operieren würden, vergleichen sie zum Erreichen des Optimierungsziels die Bitkombinationen der einzelnen Strings und geben den besseren den Vorzug. Für diese Vorgehensweise wurde von Holland die Schematheorie entwickelt. Schemata bilden eine Art Mustervorlage für eine Gruppe von Genstrings, die sich an einigen Stellen ähnlich sind. Genauer spezifiziert ist ein Schema ein Wort des dreielementigen Alphabets  $V = \{0, 1, *\}$ . Der Asterisk ist dabei ein Platzhalter für das Symbol 0 oder 1. Betrachtet man das Schema  $H = *0*10***$ , so ist der String  $A = 00110100$  ein Beispiel für das Schema  $H$ .

In einem Alphabet der Kardinalität  $k$  mit Wörtern der Länge  $L$  gibt es offensichtlich  $(k+1)^L$  Schemata. Ein Wort eines Alphabets stellt  $k^L$  Schemata dar, also sind in einer Population der Größe  $n$  bereits  $n * k^L$  Schemata vorhanden. Nimmt man z.B. wiederum das binäre Alphabet ( $k=2$ ) und eine Wortlänge von 5, so ergeben sich  $3^5 = 243$  Schemata.

Um Schemata qualifizieren zu können, werden die Begriffe Ordnung und Distanz eingeführt. Die Ordnung  $o(H)$  eines Schemas  $H$  ist die Anzahl der fixierten Positionen

des Schemas. Mit der Distanz  $\delta(H)$  wird ausgedrückt, wieviele Stellen zwischen der am weitesten links und der am weitesten rechts gelegenen fixierten Positionen innerhalb des Strings liegen. So ist z.B.  $o(*0****10) = 3$  und  $\delta(*0****10) = 5$ .

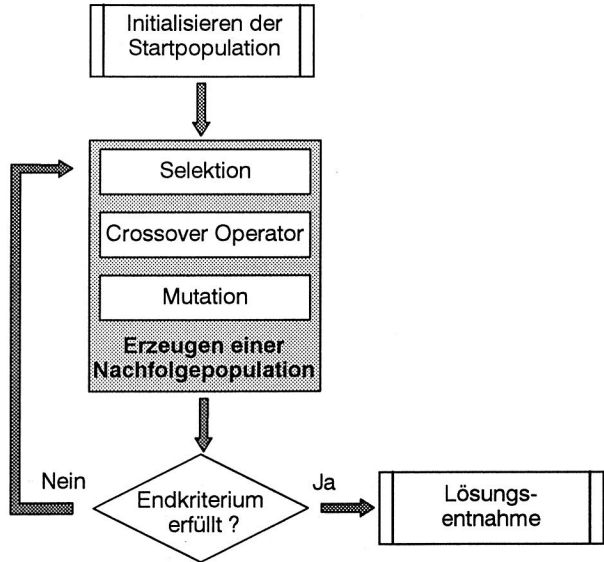


Bild 7-8: Grundschemata genetischer Algorithmen

Das Konzept der Schemata wird benutzt, um Ähnlichkeiten von Strings zu klassifizieren und den Einfluß von Operatoren zu beschreiben. Schemata mit überdurchschnittlicher Bewertung werden im Lauf der Generationen in immer größeren Zahlen in der Population vertreten sein, andere werden weniger oder verschwinden völlig. Unter Berücksichtigung der Operatoren Kreuzung, Auswahl und Mutation wurde von Holland folgende Gleichung, die auch als Fundamental Theorem der genetischen Algorithmen bezeichnet wird, entwickelt:

$$m(H, t+1) = m(H, t) \frac{f(H)}{\bar{f}} \left[ 1 - p_c \frac{\delta(H)}{L-1} - o(H) \right]$$

mit:

$m$  = Anzahl der Schemata

$H$  = Schema

$t$  = Zeitpunkt

$f(H)$  = durchschnittliche Fitneß der Strings, in denen das Schema  $H$  repräsentiert wird

$\bar{f} = \sum f_i/n$  = durchschnittliche Fitneß der Population

$p_c$  = Faktor für Wahrscheinlichkeit des Crossover-Prozesses

$L$  = Stringlänge

$\delta$  = Distanz

$o$  = Ordnung

Die Gleichung sagt aus, daß relativ kurze Schemata, die überdurchschnittlich fit sind und relativ wenig fixierte Stellen besitzen, im Laufe des Prozesses mit exponentiellen Wachstum bevorzugt werden, da die Wahrscheinlichkeit, daß sie durch einen Cross-over-Operator zerstört werden, geringer ist.

Die Konvergenzeigenschaft genetischer Algorithmen wird durch eine Reihe von Steuerparametern beeinflusst:

- die Populationsgröße
- die Crossover- und Mutationsrate
- die Wahl der Selektionsstrategie
- die Parameter für die Crossover Operatoren (z.B. die Anzahl der Cross-overpositionen)

Zur Einstellung der Parametergrößen wurden eine Reihe von Untersuchungen durchgeführt [93, 78, 18]. Allgemeingültige Vorgaben sind in der Theorie aber meistens nicht möglich. Vielmehr muß bei komplexen Problemstellungen eine geeignete Auswahl der Crossover-Operatoren und eine gut angepaßte Problemcodierung vorgenommen werden. Offensichtlich ist es notwendig, für spezielle Problemstellungen eine Codierung zu entwickeln, die die Erkenntnisse der Schematheorie berücksichtigt.

Die vorgestellten Optimierungsverfahren sind nur eine Auswahl an bekannten und getesteten Methoden. Für einige der vorliegenden Aufgabenstellungen aus der Rüst- und Umrüstoptimierung wurden diese Verfahren implementiert und getestet. In speziellen Fällen wurden zusätzlich maschinennahe Methoden konzipiert.

## **7.2 Entwicklung eines Schichtenmodells zur Optimierung**

Die zu lösenden Optimierungsaufgaben (s. Kap. 3.3.3) sind sehr verschiedenartig und derart komplex in ihrem Zusammenhang, so daß nur durch eine Aufteilung in mehrere Teilbereiche eine Lösung gefunden werden kann. Aufgrund dieser Komplexität und des großen Gesamtumfangs wurde ein mehrschichtiges Optimierungsmodell realisiert [26, 28] (s. Bild 7-9). Innerhalb des Bereiches Flachbaugruppenbestückung ergab sich eine hierarchische Einteilung bezogen auf mehrere Zielgruppen in:

- Linienübergreifende Optimierung (Schicht 1)
- Linienbezogene Optimierung (Schicht 2)
- Maschinenbezogene Optimierung (Schicht 3/4)

In Abhängigkeit von der vorliegenden Optimierungsaufgabe (Einstellungen aus der Steuerdatei) werden sequentiell die einzelnen Optimierungsstufen durchlaufen [90]. Jedes Optimierungslevel gibt hierbei seine Resultate an das in der Hierarchie Tiefer-

liegende weiter. Der modulare Aufbau innerhalb des Schichtenmodells ermöglicht, je nach Anwendungsfall, auch einen Quereinstieg, d.h. das Ausführen einzelner Teilaufgaben.

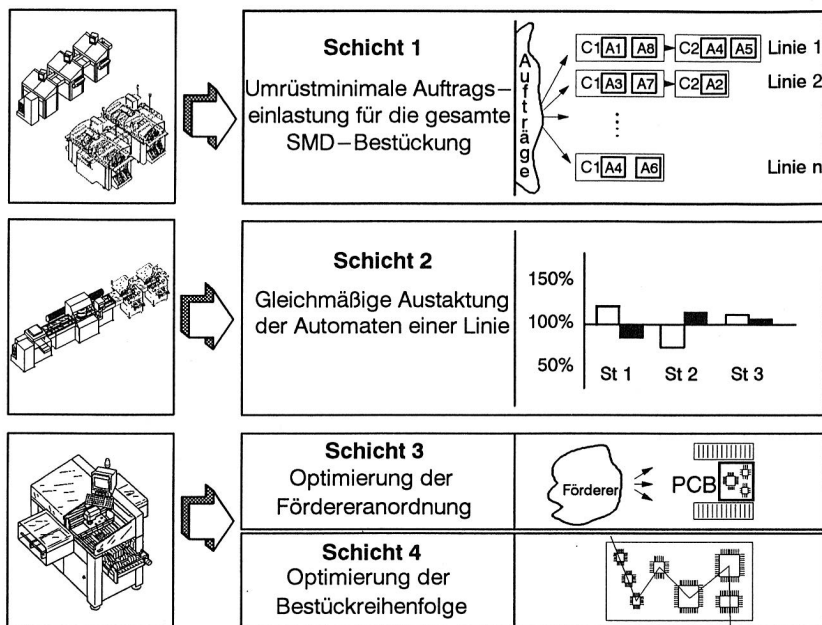


Bild 7-9: Mehrschichtiges Optimierungsmodell

Die Schicht 1 teilt die einzuplanenden Aufträge den verschiedenen Bestückungssystemen zu, mit dem Ziel, die Summe aller Nebenzeiten auf ein Minimum zu reduzieren. Sie berechnet für jedes Bestückungssystem eine umrüstopoptimale Abarbeitungsreihenfolge der Aufträge und eine minimale Anzahl an Chargen. Berücksichtigt werden müssen hierbei sämtliche technische Randbedingungen, Festrüstanteile und die zum Planungszeitpunkt festliegenden Ausgangsrüstungen.

Schicht 2, das sogenannte Line Balancing, berechnet für die in Stufe 1 ermittelten Chargen eine ausgetaktete Zuteilung der Bestückinhalte (Bauelement-Förderer-Kombinationen) auf die einzelnen Stationen einer Bestückungslinie. Die Bauelementtypen aller Leiterplatten dieser Charge werden auf die einzelnen Automaten aufgeteilt, so daß keinerlei Warte- und Blockierzeiten auftreten. Aufgrund der hinterlegten Umrüstzeiten und Auftragsdaten entscheidet diese Schicht auch, ob ein bereits gerüstetes Bauelement innerhalb der Linie umgerüstet wird.



Die maschinennahen Schichten 3 und 4 sind stark abhängig vom verwendeten Maschinentyp. Allgemeines Optimierungsziel dieser Schichten ist die Verringerung der Prozeßzeit. Erreicht wird dieses Ziel durch eine ideale Konfiguration der Feederanordnung und die Berechnung einer optimierten Bestücksequenz. Einige Automaten-typen erfordern aufgrund der hohen Wechselwirkung eine gleichzeitige Bearbeitung beider Schichten.

Die Ablaufsteuerung wertet die Steuerdateien aus und spezifiziert die Werte und Informationen in der Datenbank zum Füllen des Blackboards. Sie hat weiterhin dafür Sorge zu tragen, daß entsprechend den technischen und benutzerbedingten Vorgaben in die entsprechenden Schichten und in die relevanten Optimierungsalgorithmen verzweigt wird. Zudem müssen Zwischenergebnisse in das Blackboard-System eingetragen, kopiert und gelöscht werden.

Die unterschiedlichen Optimierungsproblematiken der einzelnen Schichten und die verwendeten Algorithmen werden in den folgenden Kapiteln im Detail erläutert.

### 7.3 Umrüstoptimierung und Chargenminimierung

Aufgabe dieser Optimierungsschicht ist die Ermittlung einer umrüstminimalen Auftragseinlastung, ausgehend von einem gegebenen Auftragspool und den vorliegenden Bestückungssystemen (s. Bild 7-10). Die Zuteilung der Aufträge und die Reihenfolgeermittlung innerhalb der Bestückungssysteme erfolgt in einem Schritt.

Die vorliegende Problemstellung läßt sich folgendermaßen formulieren:

- Zusammenfassung der Aufträge zu Chargen (**Kombination**)
- Festlegen der Auftrags- und Chargenreihenfolgen (**Permutation**)
- Minimierung der Gesamtrüstzeit (**Kostenfunktion**) unter Berücksichtigung aller Nebenbedingungen

Allein schon das Teilproblem der Auftrags- bzw. Chargenreihenfolge ( $n$  Aufträge  $\rightarrow n!$  Möglichkeiten) läßt erkennen, daß es sich hierbei um ein NP-Problem handelt.

In diesem Zusammenhang soll noch darauf hingewiesen werden, daß ohne Implementierung und direkten Vergleich in Frage kommender Algorithmen keine wissenschaftlich begründbare Auswahl möglich ist. So wurde, je nach vorliegender Aufgabe und Formulierung des Problems, über die Problemgröße und über den Anpassungsaufwand einer oder mehrere Algorithmen ausgewählt und implementiert.

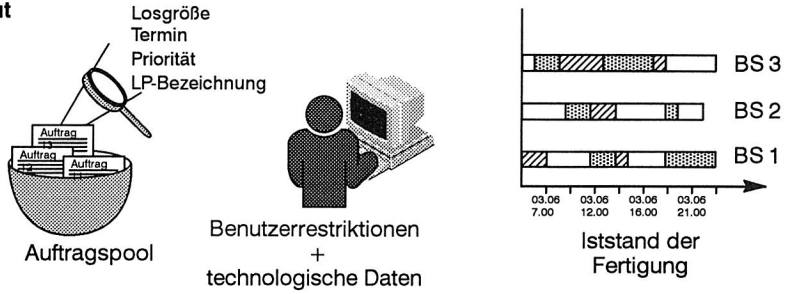
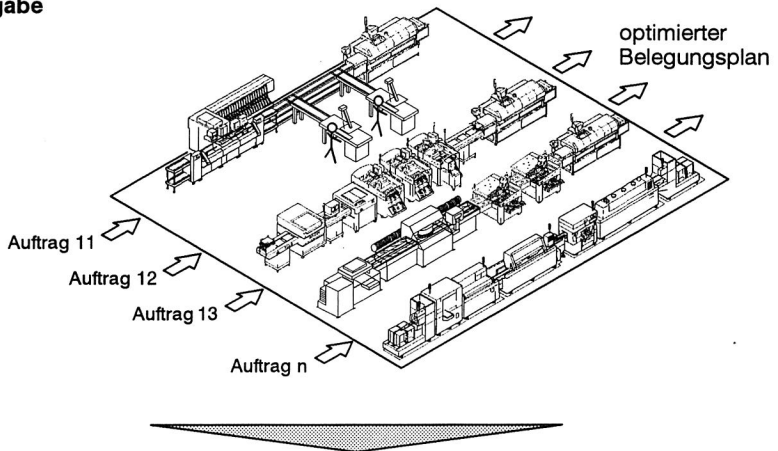
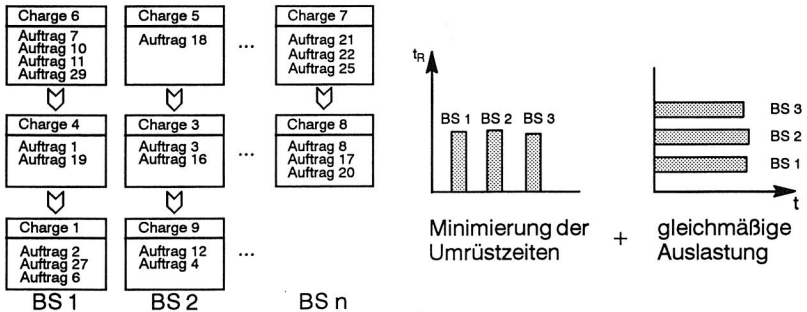
**Input****Aufgabe****Ziel**

Bild 7-10: Problemstellung Auftragsinlastung und Chargenminimierung

Aufgrund der hohen Flexibilität, der guten Adaptierbarkeit auf das vorliegende Problem und der guten Ergebnisse bei TSP-Problemen wurde ein Simulated Annealing Algorithmus für das Umrüstoptimierungsproblem gewählt.

Zur Lösung der gestellten Aufgabe der nebenzeitminimierten Auftragseinlastung entstanden sechs zeitlich aufeinanderfolgende Teilmodule. Ein Submodul beinhaltet den eigentlichen Optimierungsalgorithmus, alle anderen sind zur Verarbeitung der Randbedingungen und zur Verwaltung der Ein- bzw. Ausgabedaten.

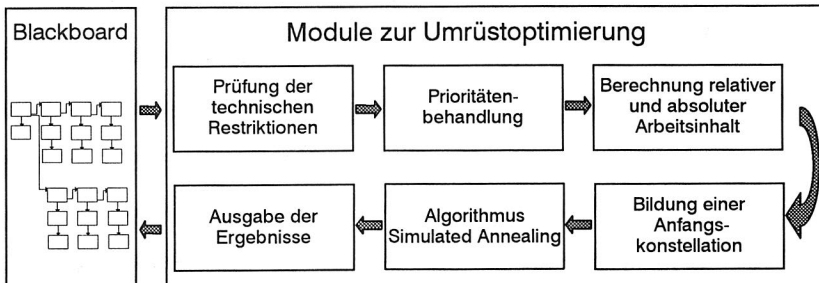


Bild 7-11: Submodule der Umrüstoptimierung

### Prüfung der technischen Restriktionen

Das erste Submodul prüft sämtliche technologischen Randbedingungen je Auftrag und Bestückungssystem. Diese Restriktionen stellen K.o.-Kriterien dar, d.h. sobald ein Bestückungssystem für eine Auftrag ausgeschlossen ist, wird dies in einer Ausschlußmatrix vermerkt. Die hierbei geprüften Kriterien sind:

- Benutzerbedingte Ausschlüsse von Leiterplatten auf bestimmten Bestückungssystemen
- Vergleich des Lötverfahrens für Leiterplatte und Bestückungssystem
- Abgleich der Leiterplattengröße mit den bestückbaren Bereichen der Bestückungssysteme
- Überprüfung der Leiterplattenfixierung (Leiterplattenfiducials ↔ Vision-System, Fangbohrung ↔ mechanische Fixierung)
- Überprüfung des zu bestückenden und bestückbaren Bauelementespektrums hinsichtlich:
  - benötigter ↔ rüstbarer Förderertyp
  - bestückbare Gehäuseform
  - Bauelementefiducial ↔ Vision-System
  - erforderliche ↔ vorhandene Koplanaritätsmessung
  - erforderliche ↔ vorhandene Zentrierstation
  - Benutzerausschlüsse von Linie, Stationstyp etc.

### **Prioritätenbehandlung**

Für jeden Auftrag sind entsprechend den Terminvorgaben verschiedene Prioritätsstufen hinterlegbar. Entsprechend der Dringlichkeit der Auftragsfertigstellung steuert die eingestellte Priorität die Reihenfolgeeinteilung dieses Auftrags.

### **Berechnung relativer und absoluter Arbeitsinhalte**

Dieses Submodul berechnet sowohl unabhängig als auch bezogen auf die Bestückungssysteme die Arbeitsinhalte der Aufträge. Der relative Arbeitsinhalt ist hierbei unabhängig vom Bestückungssequipment. Dieser wird aus der Anzahl der Bestückpositionen pro Leiterplatte und der Losgröße gebildet. Der absolute Arbeitsinhalt ergibt sich aus dem relativen Arbeitsinhalt unter Berücksichtigung der Leistungsziffer der Linie (Bestückungssystem), auf der die Aufträge gefertigt werden können. Die Leistungsziffer ist ein Maß für die Bestückleistung einer Linie. Diese wiederum ergibt sich durch Aufaddieren der Kapazitätsfaktoren der einzelnen Automaten. Als Basis für die Kapazitätsfaktoren dient die Bestückleistung pro Stunde.

### **Bildung einer Anfangskonstellation**

Der Simulated Annealing-Algorithmus setzt auf einer vorliegenden Anfangskonstellation auf. Die Methode zur Erzeugung der Konstellation stapelt die Aufträge sukzessive nach absteigender Prioritätsstufe auf die Linien. Innerhalb einer Prioritätsstufe teilt sie zunächst die Aufträge, die an eine Linie gebunden sind, zu. Es schließen sich die Aufträge mit zwei möglichen Linien an usw. Die Aufträge mit größeren relativen Arbeitsinhalt werden zuerst zugeteilt und zwar jeweils auf die Linie mit dem bisher geringsten Arbeitsinhalt, um eine möglichst gleichmäßige Auslastung der Linien zu erzielen.

Sind alle Aufträge zugeteilt, bildet die Methode Chargen. Dabei faßt sie die Aufträge vom "Stapelfuß" beginnend zusammen, so daß sie eine Linie bezüglich der Bauelementevielfalt maximal, jedoch nicht über 95% ausfüllen. Der Spielraum von 5% ist notwendig, damit ein häufig benötigtes Bauelement in der endgültigen Rüstung eventuell mehrfach gerüstet werden kann.

Die Gesamtumrüstzeit stellt die zu minimierende Zielfunktion dar. Zu deren Berechnung addiert man die Zeiten aller während einer Konstellation vorzunehmenden Ab- und Aufrüstzeiten, die zwischen Chargen entstehen. Abrüstzeiten sind die Zeiten, die benötigt werden, um einen Förderer von der Linie zu entfernen, Aufrüstzeiten entstehen, wenn Förderer für die nächste Charge gerüstet werden müssen. Die Umrüstzeiten je Förderer sind maschinenabhängig und anwenderspezifisch abgelegt (s. Kap. 4.3.2). Die Optimierungsschicht berücksichtigt hierbei festgerüstete Bauelemente und Bauelemente aus Ausgangsrüstungen. Für zugeteilte Bauelementetypen, die bereits in der Festrüstung oder Ausgangsrüstung auf dem Bestückungssystem enthalten sind, ergeben sich die Aufrüstzeiten zu Null. Die Abrüstzeiten der Bauelemente-Förderer-Kombinationen aus der Ausgangsrüstung, die in der neuen Charge nicht mehr vorkommen, kommen zur Gesamtumrüstzeit

hinzu. Einen weiteren Zeitanteil je Bestückungssystem bilden sämtliche Aufwendungen zu einer Programmumstellung.

### Simulated Annealing

Ausgehend von der übergebenen Rüstkonstellation beginnt der eigentliche Optimierungsalgorithmus. Zunächst entscheidet eine Zufallszahl darüber, ob das Submodul Vertauschung oder Reverse ausgeführt wird.

#### Tauschen

Tauschen von 1–n Aufträgen eines Bestückungssystems mit 1–m Aufträgen eines anderen bei gleicher Auftragspriorität (s. Bild 7-12). N und m sind hierbei gleichverteilte Zufallszahlen zwischen 1 und der maximalen Anzahl der Aufträge dieser Linienpriorität, wobei n und m noch dahingehend modifiziert werden, daß sich der Arbeitsinhalt nicht über ein bestimmtes Maß verschiebt. Aufträge, die aus technologischen oder benutzerbedingten Gründen nicht verschiebbar sind, werden ausgespart.

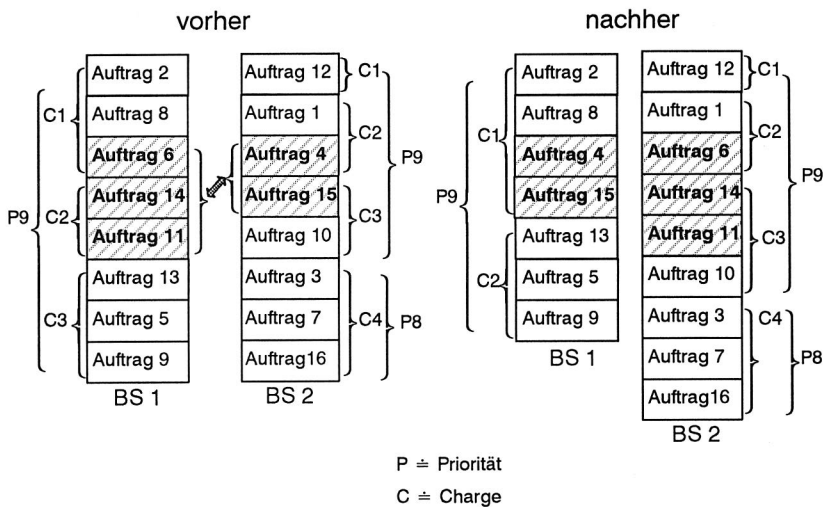


Bild 7-12: Submodul Tauschen des Simulated Annealing-Algorithmus

#### Reverse

Reihenfolgeänderung von 1–n Aufträgen gleicher Priorität innerhalb eines Bestückungssystems (s. Bild 7-13). Eine Zufallszahl bestimmt die betroffenen Aufträge. Ein Vertauschen wird nur durchgeführt, wenn eine Chargengrenze überschritten wird.

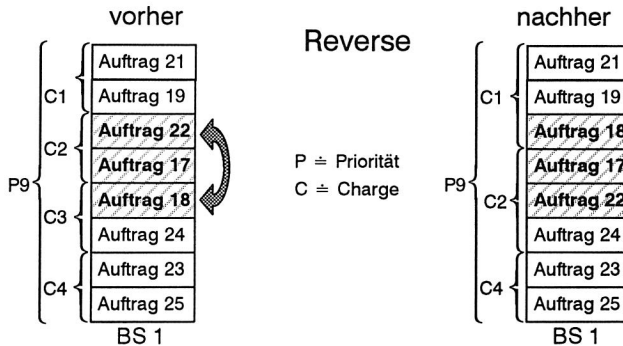


Bild 7-13: Submodul Reverse des Simulated Annealing-Algorithmus

Nach dem Rücksprung aus den Submodulen wird die Zielfunktion (Gesamturnrüstzeit) für die geänderte Konstellation neu berechnet. Ergibt sich eine kürzere Gesamturnrüstzeit, so wird die bisherige Konstellation durch die neue ersetzt. Ist eine Verschlechterung eingetreten, entscheidet die sogenannte Orakelbefragung mit der Wahrscheinlichkeit  $P = e^{\Delta E/T}$ , ob man diese zulässt. Der Kontrollparameter  $T$  verkleinert sich jeweils nach  $5N$  ( $N$  = Auftragszahl) um 10%. Der Algorithmus terminiert, wenn die letzten  $10 \times N$  Änderungen keine Verbesserungen mehr erbrachten.

### Ausgabe der Ergebnisse

Die ermittelten Ergebnisse:

- Auftragszuteilung je Bestückungssystem,
- Chargeneinteilung,
- Auftragsreihenfolgen,
- grobe Rüstungsvorschläge bzw. Bearbeitungszeiten
- und Umrüstzeiten der Chargen

werden in die Hauptspeicherstruktur eingetragen. Dort stehen sie den nachfolgenden Optimierungsschritten zur weiteren Verbesserung zur Verfügung.

## 7.4 Gleichmäßige Austaktung einzelner Bestückungslinien

Diese Optimierungsschicht wird durchlaufen, wenn ein Bestückungssystem aus mindestens zwei verketteten Bestückungsmodulen besteht. Das Balancing sorgt für eine optimale Aufteilung sämtlicher Bauelemente-Förderer-Kombinationen einer Charge auf die einzelnen Stationen einer Linie (s. Bild 7-14). Ziel ist es, eine Aufteilung zu finden, bei der ein Minimum an Warte- und Blockierzeiten auftritt. Als Wartezeit bezeichnet man die Zeit, die ein Automat der Linie auf die Leiterplatte der Vorgängerstation wartet. Blockierzeiten treten auf, wenn das Bestückungsmodul mit seinem Bestückungsanteil fertig ist, auf der Nachfolgestation aber noch kein Platz ist, um die Leiterplatte weiterzutransportieren.

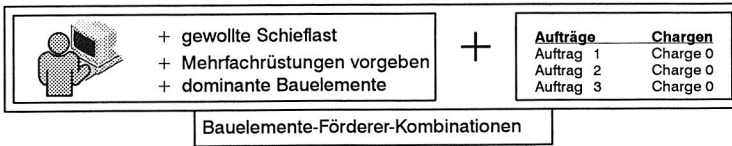
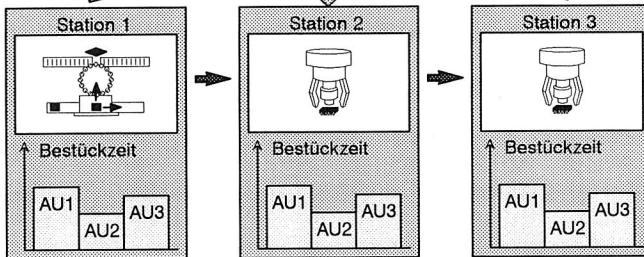
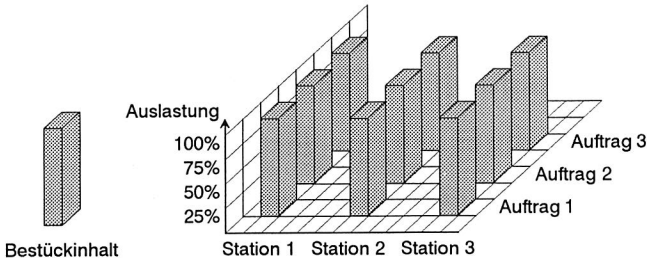
**Input****Aufgabe****Ziel**

Bild 7-14: Das Balancing-Problem

Ein ähnliches Problem aus der Montagetechnik ist das "Assembly Line Balancing"-Problem [48]. In dieser Problemstellung geht es vorrangig darum, die Arbeitselemente zur Herstellung eines Produktes auf die einzelnen Stationen eines Fließbandes zu verteilen. Die Schwierigkeit liegt vor allem darin, daß es Reihenfolgebeziehungen für die einzelnen Teilprodukte gibt. Diese werden meist durch einen Reihenfolgegraphen dargestellt. Beim Assembly Line Balancing-Problem wird versucht, für eine gegebene Taktzeit bzw. für eine vorgegebene Produktionsrate die Anzahl der notwendigen Stationen zu minimieren.

Bei genauerer Betrachtungsweise ergeben sich für das Austaktungsproblem verketteter Bestückungsautomaten jedoch einige Unterschiede, insbesondere hinsichtlich der Randbedingungen:

- Während beim Assembly Line Balancing-Problem die Reihenfolgebeziehungen vorgegeben sind, gibt es diese beim vorliegenden Problem nicht. Es existieren hingegen eine Reihe von Abhängigkeiten zwischen Bauelementen und Förderern.
- Die Zeiten für die einzelnen Arbeitselemente beim Assembly Line Balancing-Problem sind konstant, was für die Bestückzeiten der Bauelemente nicht gilt, da diese abhängig sind vom Bauelementtyp, von Sonderbehandlungen und vom Rüstort auf den unterschiedlichen Automatentypen.
- Weiterhin ist keine unbeschränkte Verteilung der Bauelemente auf die einzelnen Teilmodule möglich. Es muß stets darauf geachtet werden, daß eine (optimale) Lösung des Problems auch wirklich physikalisch rüstbar ist.

Das Optimum des Austaktungsproblems ist genau dann erreicht, wenn jeder einzelne Automat für jeden Leiterplattentyp der Charge 100% ausgelastet ist, wobei ein Abgleich zwischen Umrüstaufwand und Einsparung von Bestückzeit für bereits gerüstete Bauelemente erfolgen muß. Im Gegensatz zu anderen Optimierungsschichten ist in diesem Fall, (bei gleichen Automatentypen in der Linie) das zu erreichende Optimum für die Bestückzeit berechenbar:

$$t_{\text{ges}} = \sum_{i=1}^{\text{anzLP}} \frac{(t_B(i) \times \text{Lsgr.}(i))}{\text{anzStat}}$$

$t_{\text{ges}}$  = Gesamtzeit der Charge  
 $t_B(i)$  = Bestückzeit pro Leiterplatte  
 $\text{Lsgr.}(i)$  = Losgröße der Leiterplatte  
 $\text{anzLP}$  = Anzahl der Leiterplatten dieser Charge  
 $\text{anzStat}$  = Anzahl der Stationen

Bei der Berechnung der Zielfunktion gehen sowohl die Bestückzeiten der Bauelemente als auch die Rüstzeiten mit in die Bewertung ein. Entsprechend den anwenderspezifischen Umrüstzeiten entscheidet diese Optimierungsschicht, ob eine bereits gerüstete Bauelement-Förderer-Kombination (aus der Ausgangsrüstung) innerhalb der Linie umgesetzt wird, um eine bessere Austaktung zu erreichen, oder ob eine verminderte Umrüstzeit ein besseres Gesamtergebnis liefert. Die zu minimierende Zielfunktion des Balancing-Problems setzt sich folglich aus Bestück- und Umrüstzeiten zusammen:

$$z(t) = t_R + T_B$$

mit:

$$t_R = f(\text{Förderertyp}, \text{Ausgangsrüstung}, \text{Maschinentyp}, \text{Restriktionen})$$

$$t_B = f(\text{Bauelementtyp}, \text{Bestückposition}, \text{Losgröße}, \text{Maschinentyp}, \text{Restriktionen})$$

Die Bestückzeit für einen Leiterplattentyp wird hierbei durch die am längsten beschäftigte Station einer Linie bestimmt.

Zur Lösung der vorliegenden Problemstellung wurden drei verschiedene Algorithmen implementiert. Allen Methoden gemeinsam ist, daß sie eine mit einem Rüstalgo-



rithmus erzeugte Anfangsrüstung schrittweise verbessern. Hierzu verwenden die Algorithmen drei Grundoperationen (s. Bild 7-15):

- Verschieben von Bauelementtypen
- Vertauschen von Bauelementtypen
- Umverteilen von Bestückpositionen

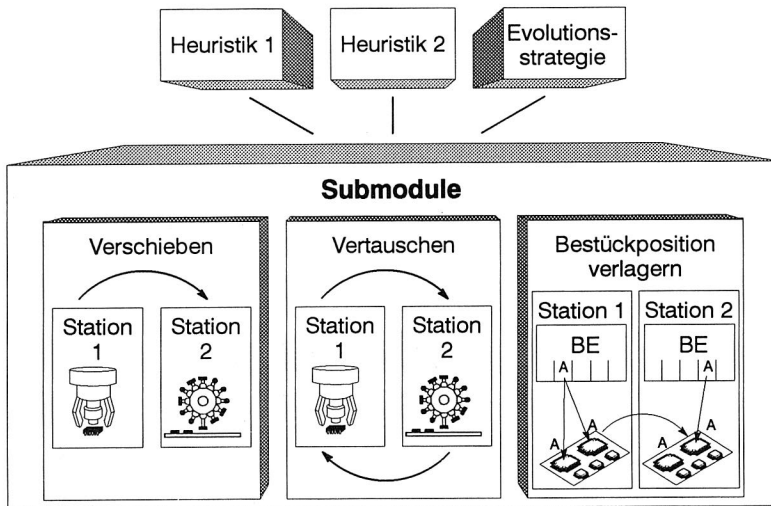


Bild 7-15: Submodule der Balancingalgorithmen

Beim Verschieben werden Bauelementtypen oder komplette Bauelemente-Förder-Kombinationen auf eine andere Station innerhalb der Linie verschoben. Verschieben von Bauelementen ist dann sinnvoll, wenn genügend Platz auf der Linie vorhanden ist. Mittels Verschiebungen (speziell von der maximal ausgelasteten zur am wenigsten ausgelasteten Station) lassen sich schnell Verbesserungen des Ausgangszustandes erzielen, da nur wenige Suchvorgänge nötig sind.

Das Vertauschen von Bauelementen erfolgt bevorzugt, wenn nur noch wenig Platz auf der Linie vorhanden ist. Aufgabe des Tauschens ist, es, zwei geeignete Tauschpartner zu finden, ohne dabei die technischen Randbedingungen zu verletzen.

Umverteilen von Bestückpositionen ist im Gegensatz zu den beiden anderen Mutationsarten nur bei Mehrfachrüstungen von Bauelementen möglich. Ist ein Bauelementtyp auf mindestens zwei Stationen innerhalb der Linie gerüstet, so können die Bestückpositionen dieser Komponente anteilig auf die Stationen verteilt werden.

Bei allen drei Methoden ist darauf zu achten, daß eine ganzheitliche Zeitermittlung erfolgen muß. Zur Optimierung der Zielfunktion werden in diesem Zusammenhang

alle Leiterplattentypen der Charge inklusive ihrer Losgrößen und fördererspezifischer Umrüstzeiten (bei vorhandener Ausgangsrüstung) betrachtet. Ein weiterer wichtiger Punkt ist die Berücksichtigung der technologischen und benutzerbedingten Randbedingungen. Einige Bauelementetypen können z.B. aufgrund benötigter Sonderbehandlungen (opt. Zentrierstation, Koplanaritätsmessung etc.) nur Automaten zugeteilt werden, auf denen die entsprechenden Zusatzmodule bereitgestellt sind.

Die implementierten Methoden sind bei Betrachtung des Optimierungsprinzipes unabhängig von den Automatentypen. Um nun unterschiedliche Automatentypen auszuwählen zu können, verwenden die Submodule (Tauschen, Verschieben, Umverteilen) intern dem Maschinentyp entsprechende Rüstmodi und Zeitmodelle. Die Differenzierung hinsichtlich der Rüstmodi ergibt sich aus den verschiedenen maschinenspezifischen Förderereigenschaften (Platzbedarf, Rüstzeiten etc.) und den unterschiedlichen Stellmöglichkeiten der Förderer. Bild 7-16 gibt einen groben Überblick über die Ablaufreihenfolge des Tauschvorgangs zwischen zwei Automaten einer Linie.

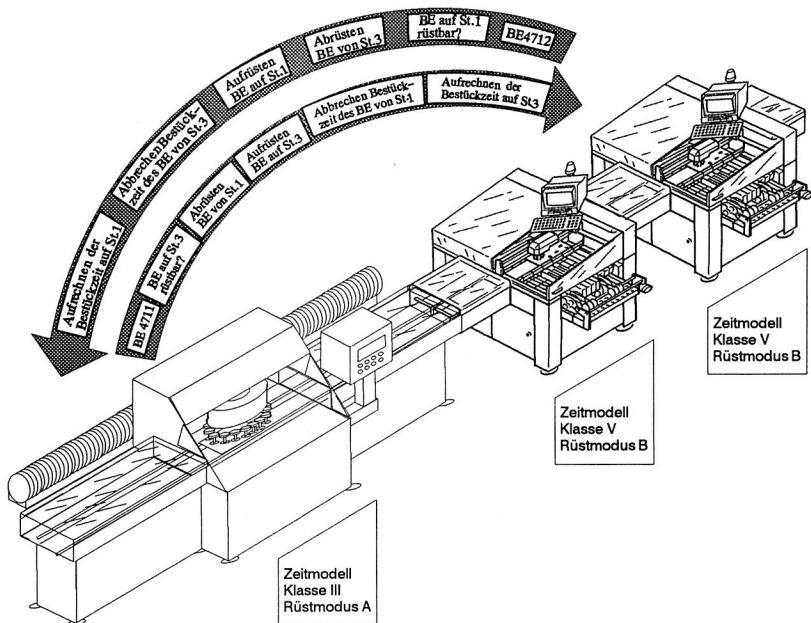


Bild 7-16: Tauschvorgang zwischen zwei Stationen einer Bestückungslinie

Die getesteten Balancingalgorithmen unterscheiden sich durch die Art und Weise, wie und vor allem wann getauscht, verschoben oder umverteilt wird.

Die implementierte Evolutionsstrategie führt ausgehend von einem Anfangszustand (Initialrüststand) zufällige Mutationen durch Veränderung verschiedener Parameter durch. Die günstigsten Mutationen werden selektiert und daraus wiederum Nachfahren erzeugt. Zur Überwindung von Suboptima wurde ein Monte Carlo-Verfahren verwendet, das durch wahlloses Vertauschen und Verschieben versucht, sich von diesem zu entfernen, um danach wiederum Verbesserungen zu erzielen.

Zusätzlich wurden zwei verschiedene Heuristiken implementiert. Die erste Heuristik bevorzugt Verschiebungen zwischen der maximal ausgelasteten Station und der mit dem geringsten Bestückvolumen. Scheitert dies, versucht sie den Bauelementetyp mit einem anderen zu tauschen. Dies gilt solange, bis die Heuristik auf ein mehrfachgerüstetes Bauelement stößt. In diesem Fall wird eine Umverteilung der Bestückpositionen vorgenommen.

Bei einer weiteren Heuristik wird das Vorgehen in zwei Phasen eingeteilt. In der 1. Phase bevorzugt sie Verschiebungen, wohingegen in der 2. Phase Vertauschungen dominieren. Bei mehrfachgerüsteten Bauelementen wird ebenso verfahren wie bei Heuristik 1.

In verschiedenen Versuchsreihen wurden die Algorithmen hinsichtlich Optimierungsergebnissen und Laufzeitverhalten auf Workstation von Hewlett Packard der Serie 9000 getestet. Die Optimierungsläufe wurden für verschiedene Steuerdateien mit verschiedenen Größen der Chargen und unterschiedlichen Anzahlen von Stationen in der Linie durchgeführt. Die Laufzeiten der Algorithmen wurden mittels der Systemzeit der UNIX Workstations ermittelt. Die zu minimierende Zielfunktion ist die Gesamtbestückzeit der kompletten Charge.

### Beispiel der durchgeführten Tests

Steuerdatei mit 3 Aufträgen:

LP 1 20 Bauelementtypen, 240 Bestückpositionen, Losgröße 150

LP 2 33 Bauelementtypen, 162 Bestückpositionen, Losgröße 100

LP 3 15 Bauelementtypen, 112 Bestückpositionen, Losgröße 50

3 Stationen (2 x Klasse V 1 x Klasse VI)

Laufzeit [s]	Zielfunktionswerte [s]		
	Heuristik 1	Evolution	Heuristik 2
0	30282	30282	30282
10	20638	19553	19226
20	20638	19434	19226
30	19957	19228	19226
40	19498	19486	19226
50	19498	19226	19226

Tabelle 7-2: Zielfunktionswerte und Laufzeiten für Testbeispiele

Geht man davon aus, daß die Stationen für alle Leiterplattentypen völlig gleichmäßig ausgelastet sind und keine Ausgangsrüstung vorliegt, demzufolge auch keine Umrüstzeiten anfallen, so kann man ein Optimum abschätzen (im Beispiel bei 17457 Sekunden).

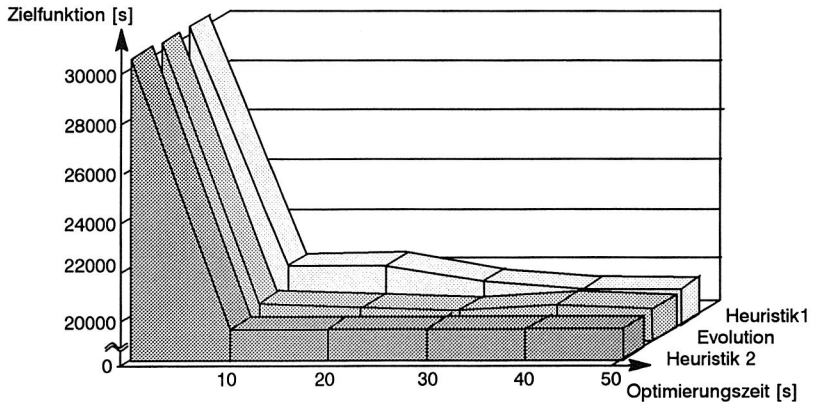


Bild 7-17: Abhängigkeit der Zielfunktion von der Rechenzeit

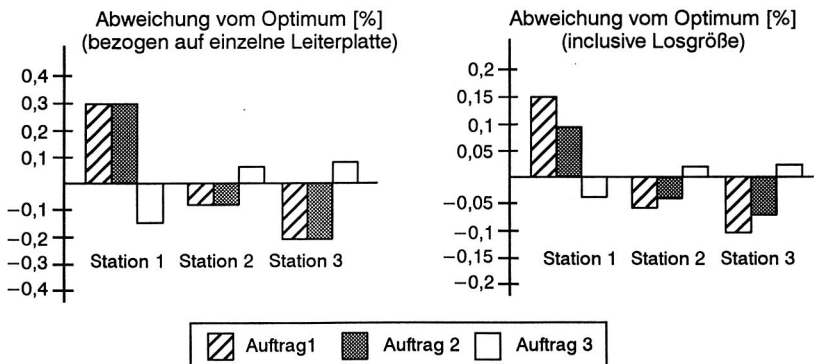


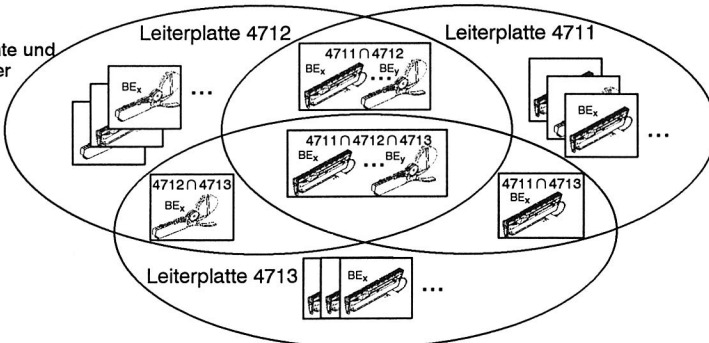
Bild 7-18: Auslastungsergebnisse der einzelnen Aufträge für Heuristik 2

## 7.5 Optimierung der Feederanordnungen

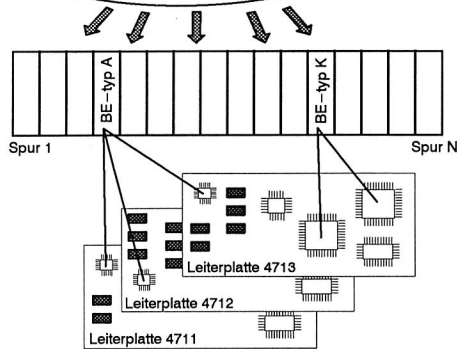
Das Optimierungsproblem der Feederkonfiguration (Spuroptimierungsproblem) besteht darin, durch günstige Anordnung der Bauelemente-Förderer-Kombinationen innerhalb der Automaten die Bestückzeit der einzelnen Leiterplatten zu reduzieren (s. Bild 7-19). Ebenso wie beim Balancing berücksichtigen die verwendeten Algorithmen alle Leiterplattentypen der Charge mit den Losgrößen und vorhandene Festrüstungen sowie Ausgangsrüstungen.

**Input:**

Bauelemente und  
Förderer der  
Charge N

**Aufgabe:**

Zuteilung aller Bauelemente  
mit Förderern der Charge  
auf die Zuführspuren

**Ziel:**

Reduzierung der Verfahrenswege  
(Bestückkopf, Schlitten, etc)

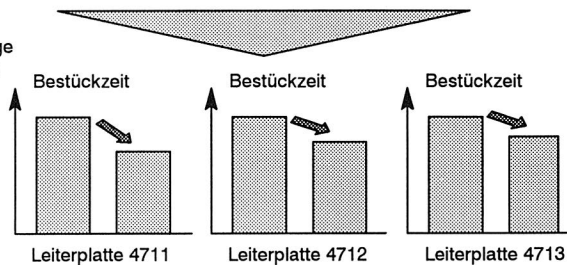


Bild 7-19: Input, Aufgabe und Ziel der Spuroptimierung

Die verwendeten Optimierungsalgorithmen dieser Schicht sind in erster Linie abhängig von der Kinematik der Maschinentypen. Teilweise ist bei einigen Maschinenklassen keine Trennung der Optimierungsschichten 3 und 4 möglich, so daß Spur- und Sequenzoptimierung innerhalb eines Optimierungsalgorithmus abgehandelt werden. Die verwendeten und getesteten Algorithmen für die unterschiedlichen Maschinenklassen sind in Tabelle 7-3 aufgelistet.

		Optimierungsschichten	
Maschinenklassen		Schicht 3	Schicht 4
I		<input type="radio"/> genetischer Algorithmus	
II		<input type="radio"/> maschinenspezifische Heuristik	<input type="radio"/> Nearest Neighbour
III		<input type="radio"/> genetischer Algorithmus	
IV		<input type="radio"/> genetischer Algorithmus	
V		<input type="radio"/> maschinenspezifische Heuristik <input type="radio"/> Transportalgorithmus	<input type="radio"/> Nearest Neighbour <input type="radio"/> Cheapest Insertion <input type="radio"/> Minimum Spanning Tree <input type="radio"/> Assignment and Patching <input type="radio"/> Positions-Sortierer <input type="radio"/> Städte-Permutierer <input type="radio"/> Städtefolgen-Permutierer
VI	Einfachportal	<input type="radio"/> Transportalgorithmus	<input type="radio"/> Nearest Neighbour <input type="radio"/> Cheapest Insertion
		<input type="radio"/> genetischer Algorithmus	
	Doppelportal	<input type="radio"/> genetischer Algorithmus	

*Tabelle 7-3: Übersicht zu den Spuroptimierungsalgorithmen der verschiedenen Maschinenklassen*

Im folgenden werden exemplarisch für einige Maschinenklassen Lösungen des Spuroptimierungsproblems näher erläutert.

### 7.5.1 Spuroptimierung für Maschinenklasse III

Die Spuroptimierung für einen Chip Shooter der Klasse III ist ein sehr komplexes Optimierungsproblem, da die Anordnung der Bauelemente-Förderer-Kombinationen auf dem beweglichen Schlitten und die resultierende Bestücksequenz eng miteinander gekoppelt sind (s. Bild 7-20). Es wurde hierzu eine kombinierte Lösungsstrategie implementiert, die beide Problemstellungen abdeckt. Die entworfene Optimierungsmethode besteht im Kern aus einem genetischen Algorithmus, der durch verschiedene Heuristiken unterstützt wird.

Charakteristika der sehr einfachen Heuristiken sind, daß sie in der Regel schon nach kurzer Zeit gute Anfangsergebnisse liefern, dann aber schnell an Grenzen stoßen. Genetische Algorithmen hingegen brauchen verhältnismäßig lange, bis sie aus einer zufällig initialisierten Startpopulation gute Lösungen erzielen, finden aber immer weitere Verbesserungen theoretisch bis zum Optimum. Es lag also nahe, die positiven Eigenschaften der beiden Methoden zu verknüpfen und einen kombinier-

ten Lösungsweg einzuschlagen. Die Stellen, an denen zur Unterstützung Heuristiken eingesetzt werden, sind:

- vor dem GA zur Erzeugung einer guten Startpopulation
- in einer Schleife mit dem GA, um bei vorzeitiger Konvergenz den GA neu zu starten
- nach dem GA zur Feinkorrektur der vom GA präferierten Lösung

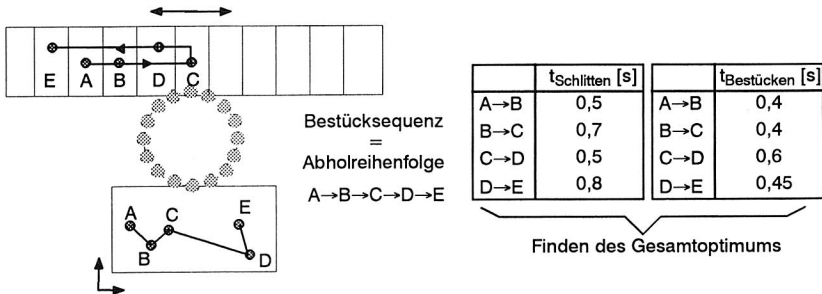


Bild 7-20: Optimierungsproblem für die Bestückungskategorie III

Zu Beginn des genetischen Algorithmus erzeugen drei verschiedene Heuristiken gute Startwerte aus einer zufällig initiierten Startpopulation. Dies bedeutet für den Bereich Verfahrensoptimierung kurze Kanten (Reihenfolge der Bestückpositionen) und für die Rüstung zusammenhängende Bereiche der Bauteiltypen eines Auftrags.

Zur Erhaltung guter Startwerte für die Optimierung der Verfahrenswegen wurden zwei Nearest Neighbour-Algorithmen implementiert. Die Unterschiede liegen darin, daß die eine Methode völlig frei die nächste Position, die andere nur Nachbarn des gleichen Bauelementtyps sucht, solange diese Typen noch vorhanden sind.

Die dritte Heuristik zur Erzeugung einer Startpopulation verbessert die Genstringbereiche, die für die Rüstung zuständig sind. Sie packt Bauelementtypen einer zufällig gewählten Leiterplatte der Charge zusammen, so daß Fahrwege des Bauelementeschlittens minimiert werden. Analog zu den Heuristiken zur Verbesserung der Verfahrenswegen berücksichtigt auch diese Methode noch keinerlei Zusammenhänge zwischen Bestückungssequenz und Rüstung. Ebenfalls die Anzahl an Bestückpositionen spielt hier noch keine Rolle.

Die angewendeten Algorithmen initialisieren Teile der Startpopulation. Dabei ist zu beachten, daß für einen Genstring der Population nur eine Methode angewendet wird und dies für einen zufällig gewählten Leiterplattentyp. Demzufolge entsteht ein Mix aus teiloptimierten Genstrings (entweder verfahrensweg- oder rüstungsoptimiert) für verschiedene Leiterplattentypen der gesamten Charge. Testergebnisse ergaben, daß eine Verteilung von je 30% für die Nearest Neighbour-Algorithmen und 40% für die Förderbereichsverteilung eine gute und breite Basis für die Startpopulation bilden.

Eine weitere Heuristik ist in einer Schleife mit dem genetischen Algorithmus verknüpft und sorgt dafür, daß der GA bei vorzeitiger Konvergenz neu gestartet wird. Da der verwendete edge recombination-Operator kaum neue Kanten erzeugt, sondern vielmehr Kanten neu kombiniert, konvergiert der GA bei den verwendeten kleinen Populationen auch in Punkten, die noch weit vom Optimum entfernt sind. Anstelle einer ziellosen Mutation, die zu schlechteren Fitneßwerten führen würde, läuft der GA in einer Schleife mit einer Heuristik, die zwar die bereits vorhandenen guten Werte nicht verbessert, aber neue nützliche Kanten (Gene) zur Verfügung stellt. Diese Vorgehensweise entspricht auch anderen Versuchen mit GAs, wobei es besser ist, kleinere Populationen zu verwenden und den GA mehrmals zu starten [36].

Ziel ist es, die Entropie in der Population zu erhöhen, ohne bereits gefundene Lösungen wesentlich zu verschlechtern. Deshalb arbeiten hier auf der gesamten Population Verfahren, die jeweils nur kleine Teile eines Genstrings gezielt verändern. Zur Erhöhung der Entropie sind tourerzeugende Algorithmen ungeeignet, da sie bereits gefundene Lösungen zerstören. Es wurden in diesem Fall Kantenaustauschverfahren [59] für die Verfahrensoptimierung eingesetzt. Insgesamt wurden fünf Verfahren verwendet: drei zur Manipulierung des Verfahrensweges und zwei zur Änderung der Rüstung:

- Kantenpermutationsverfahren
- Kantenpermutationsverfahren mit Berücksichtigung der Bauelementtypen
- Invertieren eines Abschnitts des Genstrings für die Rüstung. Erfordert die ermittelte Bestücksequenz eine Schlittenbewegung, die größer ist als eine vorgegebene Schrittweite, wird der Teil der Rüstung invertiert, so daß die beiden Rüstpositionen nebeneinander zu liegen kommen.
- Über den Genstring der Rüstung wird ein Fenster der Größe 3 (Rüstplätze) verschoben. Die Bestückhäufigkeit der Bauelemente wird abgezählt. Falls das Minimum in der Mitte liegt, wird getauscht, so daß eine Art Gaußkurve entsteht.
- Teile des Verfahrensweges werden entsprechend der Reihenfolge der Bauelementtypen auf dem Bauelementeschlitten umsortiert.

Den Kern der implementierten Lösung für die Spuroptimierung bildet ein genetischer Algorithmus, der nach der Generierung der Startposition und einer Tabelle für die Fitneßwerte erstmalig abläuft. Seine Vorgehensweise

- wähle aus Population zwei Eltern  $x_i$  und  $x_j$
- wähle aus Population einen Genstring  $x_k$
- kreuze  $x_i$  und  $x_j$ . Nachkomme ersetzt  $x_k$
- aktualisiere Fitneßtabelle

unterscheidet sich im Vergleich zum Standardmodell (Kap. 7.1.2) in zwei Punkten. Zum einen wird auf eine Mutation verzichtet, und zum anderen entsteht eine neue Generation nicht durch den Auswahloperator, sondern durch sukzessives Ersetzen einzelner Genstrings. Beides ist bedingt durch die Besonderheiten des verwendeten "edge recombination"-Kreuzungsoperators.



Die Codierung der Genstrings erfolgte auf der Basis natürlicher Zahlen. Es wurden sowohl die Bestückpositionen jeder Leiterplatte, als auch die Rüstpositionen der Charge numeriert. Die Codierung widerspricht zwar der Regel, mit möglichst kleinen Alphabeten zu arbeiten, für TSP-Probleme wurden aber hierzu schon einige gute Ergebnisse nachgewiesen [105]. Die Unterscheidung zum reinen TSP-Problem ist dadurch gegeben, daß

- außer dem Verfahrenweg noch die Rüstung optimiert wird und
- für mehrere Leiterplatten losgrößenabhängig die Optimierung erfolgt.

Die genannten Unterscheidungen führten zu zwei möglichen Wegen der Codierungsform:

- Codierung in einem Genstring der Länge  $n$  mit  $n = \sum_{i=1}^{\text{Anz. LP}} n_i + m$ , wobei  $n_i$  die Anzahl der Bestückpositionen auf der Leiterplatte  $i$  ist und  $m$  die Gesamtzahl der frei rüstbaren Bauelemente (s. Bild 7-21).
- Für jede Leiterplatte ein Genstring und einer für die Modulplatzeleiste (s. Bild 7-21). Ähnlich wie in der Genetik der Genotyp aus Chromosomen, besteht eine Struktur (ein Genotyp) aus mehreren Strings. Bei  $k$  Leiterplatten ergeben sich folglich  $k+1$  Strings.

eindimensionaler Genstring der Länge  $n_0 + n_1 + m$

Bestücksequenz Leiterplatte 0	Bestücksequenz Leiterplatte 1	Rüstung der Charge
$n_0$		$n_1$

zweidimensionale Anordnung für Sequenzen und Rüstung

Bestücksequenz Leiterplatte 0	$n_0$
Bestücksequenz Leiterplatte 1	$n_1$
Rüstung der Charge	
	$m$

Bild 7-21: Unterschiedliche Codierungsformen (eine Charge mit zwei Aufträgen)

Die Testergebnisse ergaben nur marginale Unterschiede der zwei Codierungsformen hinsichtlich der erzielten Optimierungsergebnisse. Bei der Rechenzeit zeigte die zweite Version Vorteile.

Der eingesetzte "edge recombination"-Kreuzungsoperator wurde eigens für TSP-Probleme entwickelt [102, 118]. Im hier verwendeten speziellen Fall wurde der Startpunkt nicht wie üblich zufällig gewählt. Er ist stattdessen immer Randpunkt oder ein Punkt, der keine gemeinsamen Kanten der beiden Eltern hat. Für Gene, die den Verfahrenweg beschreiben, waren demzufolge gute Ergebnisse zu erwarten. Wie die Testläufe zeigten, ergaben sich auch für die erweiterte, komplexere Problemstellung der vorliegenden Spur/Sequenzoptimierung sehr gute Ergebnisse.

Die anschließende Auswahlfunktion hat die Aufgabe, gute Genstrings häufiger als schlechte in die nächste Generation zu übernehmen. Für den implementierten Algo-

rithmus wurde als Auswahlfunktion folgende Vorgehensweise gewählt. Nicht die Häufigkeit eines Genstrings in der nächsten Generation nimmt zu, sondern die Wahrscheinlichkeit, daß er gekreuzt wird. Im Gegenzug nimmt die Wahrscheinlichkeit, durch neu entstehende Genstrings aus der Population verdrängt zu werden, mit abnehmender Fitneß zu. Die Vorgehensweise des Auswahloperators gestaltet sich somit folgendermaßen:

- Wähle zwei Genstrings  $x_1$  und  $x_2$  als Eltern. Hohe Fitneß führt zu hoher Auswahlwahrscheinlichkeit.
- Wähle einen Genstring  $x_3$ . Niedrige Fitneß führt zu hoher Auswahlwahrscheinlichkeit.
- Kreuze  $x_1$  und  $x_2$ . Das Kind nimmt den Platz von  $x_3$  in der Population ein.
- Berechne neue Fitneß und trage sie in Fitneßtablelle ein.

Die Auswahlwahrscheinlichkeiten können durch unterschiedliche Randomfunktionen bestimmt werden, wodurch der genetische Algorithmus hinsichtlich schneller Konvergenz und breiter Suche im Lösungsraum beeinflusst wird.

"Let nature be your guide" ist Goldbergs erste Regel für die Arbeit mit genetische Algorithmen [35]. Rouletteauswahl und ähnliche Operatoren erzeugen in einem Schritt aus einer Population eine neue Population. In dem vorliegenden Fall sind, wie in der Natur, Individuen unterschiedlichen Alters in einer Population vorhanden. Mit jedem Kreuzungsvorgang entsteht ein neuer Genstring, der einen anderen verdrängt. Innerhalb einer Generation werden einige Genstrings mehrmals, andere dagegen überhaupt nicht gekreuzt. Die implementierte Version zählt aus Gründen der Zeitkontrolle die Generationen, um den GA rechtzeitig zu beenden.

Der genetische Algorithmus arbeitet nur mit Zahlenreihen (zyklischen Permutationen). Einzige Verbindung zum Optimierungsproblem sind die Fitneßwerte, anhand derer der Auswahloperator entscheidet, welche Genstrings sich vermehren und welche durch die Nachkommen verdrängt werden. Die Fitneßfunktion bildet also den Suchraum (Menge aller zyklischen Permutationen) auf den Lösungsraum (Menge aller Verfahrenswege und Rüstungen) ab. Aus den Chromosomen für die Leiterplatte wird der Verfahrensweg, aus den Chromosomen für die Rüstung die Anordnung der Bauelemente auf dem Schlitten bestimmt und dann die Bestückzeit berechnet.

Da mehrere Aufträge, die ohne Umrüsten bestückt werden, zu Chargen zusammengefaßt werden, ist die Fitneß eines Genstrings der Durchschnitt der Bestückzeiten aller Aufträge der Charge. Aufgrund der unterschiedlichen Stückzahlen der Leiterplatten innerhalb einer Charge werden die Bestückzeiten mit der Losgröße des Auftrags gewichtet. Dadurch wird erreicht, daß größere Aufträge mehr Anteil an der Fitneß

eines Genstrings haben als kleinere und damit bei der Optimierung stärker berücksichtigt werden.

$$F = \frac{1}{m} \sum_{j=1}^{\text{anz. auftr}} T_j m_j$$

$F$	= Gesamtfiteß
$T_j$	= Bestückzeit Auftrag $j$
$m_j$	= Losgröße Auftrag $j$
$m$	= Losgröße gesamt

Die Berechnung der Bestückzeiten erfolgt auf der Basis des bereits in Kap. 4.3.2 vorgestellten Zeitmodells für diese Bestückautomatenklasse. Der genetischen Algorithmus oder z.T. Zusatzfunktionen berücksichtigen die vorliegende Restriktionen. So legt der Algorithmus bei Mehrfachrüstungen für das mehrfach zu rüstende Bauelement entsprechend viele Stellen innerhalb des Genes an. Festrüstungen und das Vermeiden von Lücken auf dem Förderbereich werden durch eine separate Funktion zwischen der Übergabe des Gens und der Berechnung der Fitneß behandelt. Besondere Beachtung gilt auch Bauelementen, die nur mit reduzierter Geschwindigkeit bestückbar sind. Wird ein Bauelement mit dieser Randbedingung bestückt, kann ab diesem Zeitpunkt der restliche Teil nur noch mit reduzierter Beschleunigung gesetzt werden. Die entworfenen Lösungsmethode ordnet diese Bauelemente automatisch am Schluß der Sequenz an. Den kompletten Ablauf der Optimierungsmethode zeigt Bild 7-22.

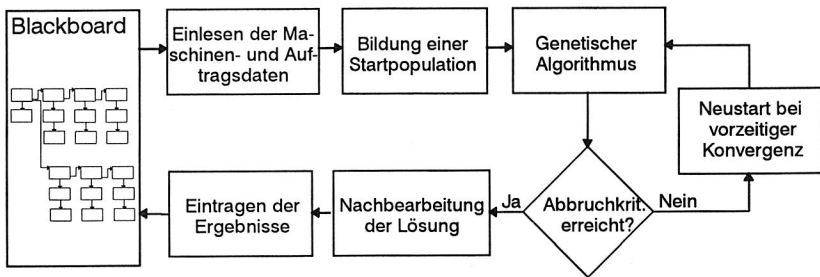


Bild 7-22: Ablaufstruktur des Optimierungsvorganges

Eine Reihe von Testergebnissen ließ erkennen, daß die Heuristiken, die wesentlich schneller arbeiten als der genetische Algorithmus, oft schon brauchbare Werte liefern, wenn nur wenig Zeit für die Optimierung zur Verfügung steht. Die Optimierungsergebnisse der Heuristiken sind abhängig von Zusammensetzung und Größe der Charge, dem Design der Leiterplatten und den Geschwindigkeiten von Positionier- und Transportschlitten. Je größer der Anteil der Schlittenbewegungen an der Gesamtbestückzeit, desto schlechter sind die Ergebnisse der Startheuristiken. Bei Chargen mit mehreren Aufträgen und mit einer relativ hohen Bauelementeüberdeckung, ist der genetische Algorithmus den Heuristiken weit überlegen. Zusammenfassend kann man sagen: Sind die Randbedingungen so gegeben, daß sich die Optimierung hauptsächlich auf die Fahrwegoptimierung konzentriert, werden von

den Heuristiken schon sehr gute Werte vorgegeben, die der GA nur noch marginal verbessern kann. Bei steigender Komplexität (d.h. mehrere Aufträge, große Bauelementevielfalt, die zu großen Sprüngen auf den Transportschlitten führt und bei steigender Bauelementeüberdeckung) kommen die Fähigkeiten und Vorzüge des genetischen Algorithmus voll zum Tragen.

Zur Validierung der implementierten Methode wurde u.a. auch mit einer speziellen Leiterplatte getestet, bei der das Optimum ersichtlich ist. Für die in Bild 7-23 dargestellte Platine mit 81 Bestückpositionen und 9 Bauelementetypen läßt sich eine optimale Bestücksequenz und Rüstung bestimmen. Die Bauelementetypen müssen hierbei in der Reihenfolge der Bestückung gerüstet werden.

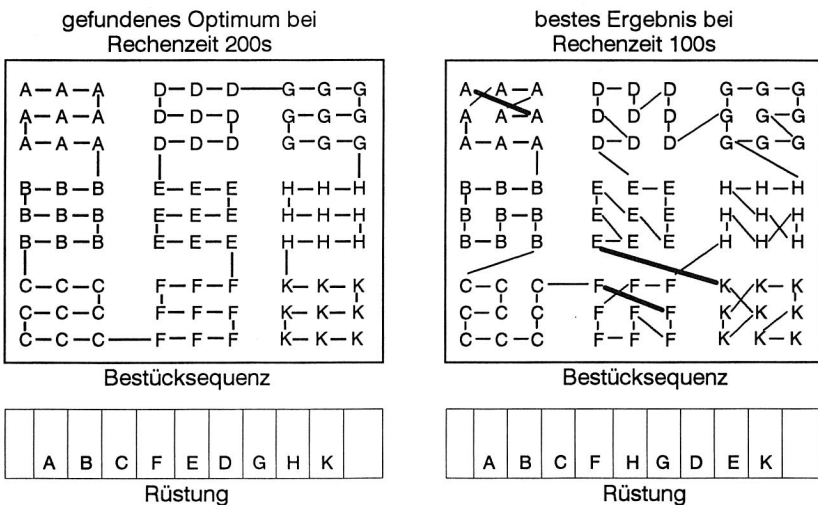


Bild 7-23: Validierung der Ergebnisse an einer fiktiven Leiterplatte

Nach 100 s Laufzeit und einer Populationsgröße von 100 waren alle fünf Werte weniger als 6% vom Optimum entfernt, drei davon weniger als 3%. Das beste Ergebnis mit 30.96 s ist im Bild 7-23 rechts dargestellt. Die Abweichungen vom Optimum sind fett hervorgehoben. Dadurch, daß die x und y-Achse nicht synchronisiert sind, sind diagonale Kanten nicht länger (kein euklidischer Abstand). Nach 200 s Laufzeit wurde bereits in drei von fünf Fällen ein Optimum (30.78 s), im Bild 7-23 links dargestellt, erreicht.

Ein weiteres Testbeispiel zeigt die Interdependenzen zwischen der Geschwindigkeit der Leiterplattenpositionierung und der Geschwindigkeit des Transportschlittens (Bild 7-24). Ist das Verhältnis  $v_{LP}/v_{FB}$  sehr groß, dann erreicht man optimale Bestückzeiten durch Bestücken gleicher Bauelementetypen hintereinander. Der Transportschlitten bewegt sich nur in eine Richtung und immer nur einen Förderer weiter. Völlig

andere Bestücksequenzen erhält man, wenn das Verhältnis  $v_{LP}/v_{FB}$  klein ist. Dann muß der Verfahrenweg möglichst kurz sein, und es ergibt sich ein echtes TSP-Problem.

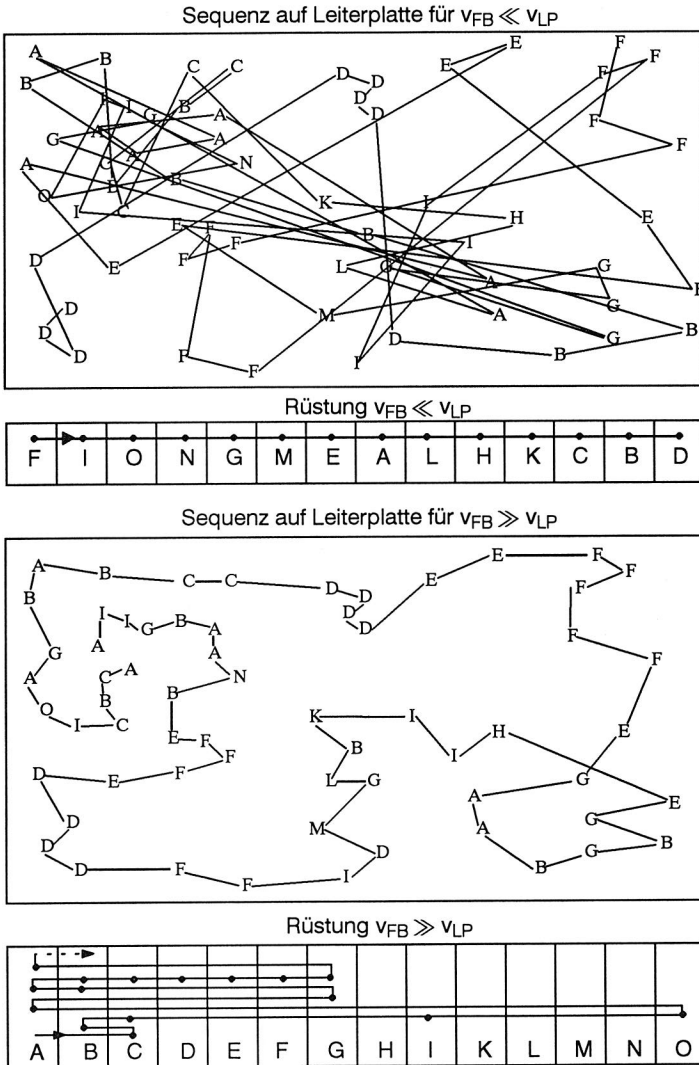


Bild 7-24: Testergebnisse bei unterschiedlicher Gewichtung des Verhältnisses  $v_{LP}/v_{FB}$

### 7.5.2 Spuroptimierung für Maschinenklasse V

Das Spuroptimierungsproblem für Pick and Place-Automaten (Klasse V) läßt eine unabhängige Optimierung der Schicht 3 und 4 zu. Das Rationalisierungspotential dieser Optimierungsebene liegt darin, die Feeder für eine vorgegebene Charge so auf den(die) Förderbereich(e) zu platzieren, daß die Wege des Bestückkopfes minimiert werden (Bild 7-25).

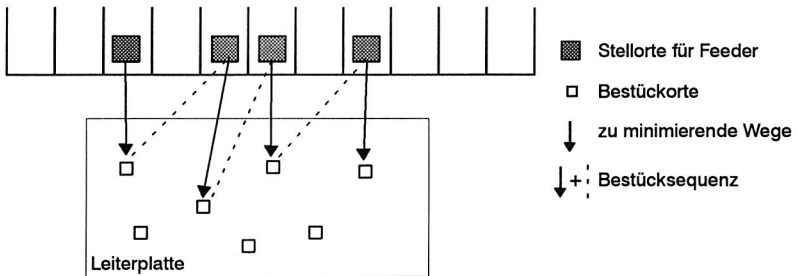


Bild 7-25: Feederkonfiguration zur Minimierung der Wege

Zu berücksichtigende Randbedingungen sind Fest-, Ausgangsrüstungen und Einschränkung der Stellmöglichkeiten bestimmter Feedertypen auf den Förderbereichen. Zur Lösung des Problems wurde eine maschinenspezifische Heuristik, ein Zuordnungsalgorithmus und eine Evolutionsstrategie implementiert. Zur Reduktion der Datenmengen bildet man zu jedem Bauelementtyp und zugehörigen Bestückpositionen Schwerpunkte auf der Leiterplatte (Bild 7-26). Die Algorithmen nehmen dann die Zuteilung der Feeder bezüglich der Schwerpunkte vor.

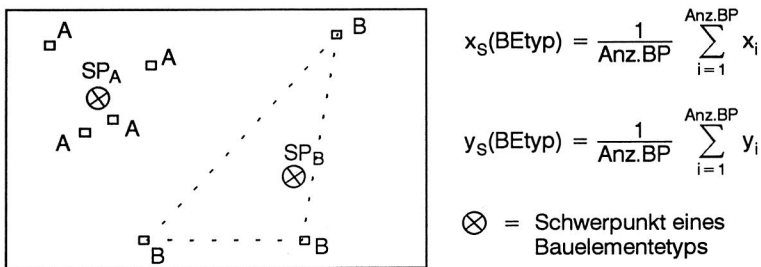


Bild 7-26: Schwerpunktbildung für Bauelementtypen auf der Leiterplatte

Sind für Bauelementtypen bestimmte Sonderbehandlungen (z.B. Zentrierung, Messung der Koplanarität etc.) erforderlich und sind diese Sondermodule innerhalb des Maschinentyps an fixierten Stellen angeordnet, wird die maschinenspezifische Heuristik eingesetzt. In diesem Fall ist eine Wegkomponente des Bestückkopfes, vom Sondermodul zum Bestückschwerpunkt, fix vorgegeben. Entsprechend der

Vorkommenshäufigkeit innerhalb der Charge werden diese Bauelementetypen auf die zum Sondermodul nächstgelegenen freien Stellplätze gesetzt (s. Bild 7-27).

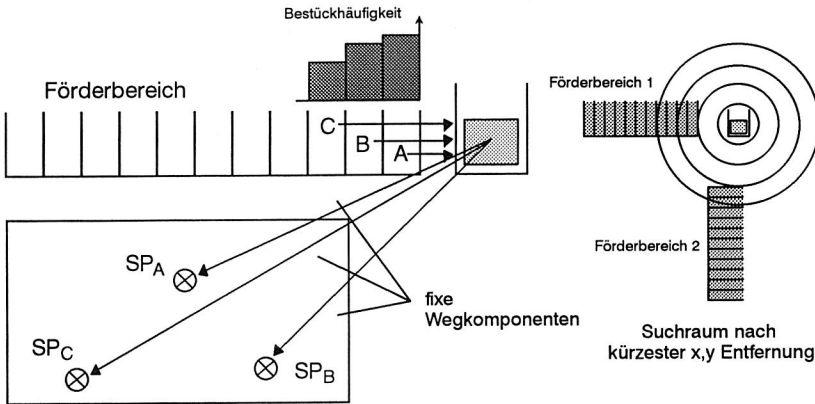


Bild 7-27: Vorgehensweise der maschinenspezifischen Heuristik

Für die Zuteilung der restlichen Bauelementetypen wurden zwei Verfahren implementiert und getestet, eine Evolutionsstrategie und ein Algorithmus für das klassische Zuordnungsproblem, basierend auf dem Satz von König [117]. Der implementierte Transportalgorithmus ist auf einen Spezialfall des Transportproblems zurückzuführen. Beim klassischen Transportproblem besteht die Aufgabe darin, die von Firmen  $F_1 \dots F_n$  produzierten Güter zu den Verbrauchern  $V_1 \dots V_m$  zu transportieren. Die von  $F_i$  produzierte Menge sei  $a_i > 0$ , der Verbraucher benötige  $b_j > 0$  Mengen des Gutes und die Kosten für den Transport einer Mengeneinheit von  $F_i$  nach  $V_j$  sollen  $c_{ij}$  betragen. Die Matrix  $m \times n$  mit den Elementen  $c_{ij}$  wird Kostenmatrix genannt. Ist ein Transport von  $F_i$  nach  $V_j$  nicht möglich, so kann man dies durch das Setzen von hohen Kostenkoeffizienten erzwingen.

Im vorliegenden Problemfall sind die Produzenten die möglichen Stellplätze der Bauelemente-Förderer-Kombinationen, die Verbraucher die Bestücksschwerpunkte auf der Leiterplatte und die Transportkosten die Zeiten, die der Bestückkopf von Stellplatz  $i$  zu Bestückort  $j$  benötigt. Die produzierte und vom Verbraucher benötigte Menge ist gleich 1. Das Zuordnungsproblem lässt sich nun folgendermaßen formulieren:

$$\text{minimiere: } \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

$$\text{unter den Nebenbedingungen: } \sum_{i=1}^n x_{ij} = a_i = 1 \quad (i = 1, \dots, n)$$

$$\sum_{j=1}^m x_{ij} = a_j = 1 \quad (j = 1, \dots, n)$$

Die zweite implementierte Alternative ist eine Evolutionsstrategie, die wie schon beim Balancing zufallsgesteuert Bauelemente-Förderer-Kombinationen herausgreift und bei Ergebnisverbesserung verstellt. Im Gegensatz zum Transportalgorithmus ist bei der Mutations-/Selektionsstrategie eine Startrüstung erforderlich.

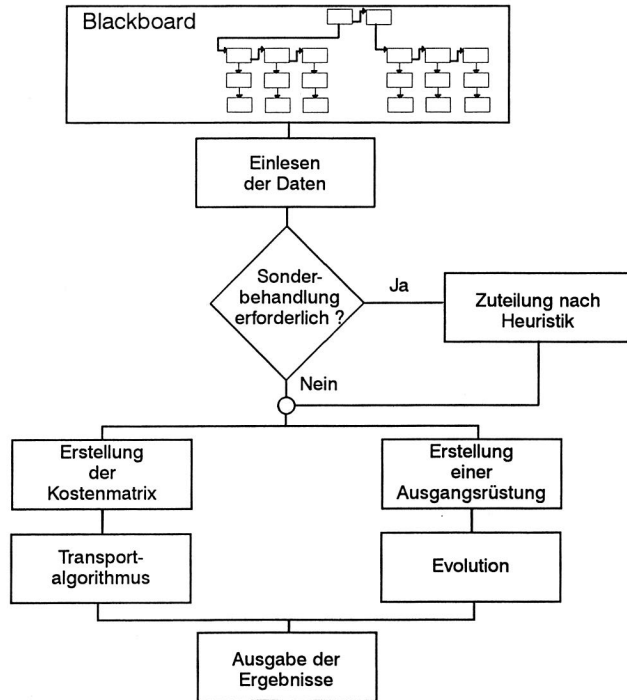


Bild 7-28: Alternative Abläufe der Spuroptimierung Klasse V

Da bei den durchgeführten Testläufen kein Optimum berechnet werden konnte, ist eine Aussage über die Annäherung ans Optimum nicht möglich. Im Durchschnitt wurde für die getesteten Chargen eine Reduzierung der Bestückzeiten von 9% erreicht. Der Transportalgorithmus ist der Evolution an Laufzeitverhalten weit überlegen, bei sehr hohen Laufzeiten erreicht die Evolution teilweise bessere Ergebnisse.

### 7.5.3 Spuroptimierung für Maschinenklasse VI

Die Funktionsweise der Collect and Place-Automaten (Beispiele sind SP120, SIPLACE 80S von Siemens) gliedert sich in verschiedenen Phasen (s. Bild 7-29). Die Collectphase zur Aufnahme der Bauelemente, die Placephase zum Plazieren der Bauelemente auf die Leiterplatte und die Transportwege zwischen Collect- und Placemodus. Während der Collectphase fährt der Bestückkopf sequentiell die einzelnen



Spuren auf einem Förderbereich ab und saugt jeweils mit der Pipette, die in der Abholposition des Kranzes steht, durch Unterdruck ein Bauelement an. Nach dem Aufnahmevorgang führt der Revolverkopf eine Rotationsbewegung (z.B.  $30^\circ$  bei 12 Pipetten) aus, die parallel zu einer Bewegung des Portals ausgeführt werden kann. Wurde der Kranz mit allen vorgesehenen Bauelementen gefüllt, schließt sich die Placephase an, bei der nach derselben Vorgehensweise die Bestückpositionen angefahren werden. Es findet eine Pufferung nach dem FIFO-Prinzip entsprechend der maximalen Anzahl von Pipetten am Revolverkopf statt. Der Vorteil im Vergleich zum rein sequentiellen Bestückvorgang besteht in der Vermeidung von maximaler Pipettenanzahl – 1 Verfahrenswegen zwischen einem Förderbereich und der Leiterplatte. Zusammengefaßt bezeichnet man die drei verschiedenen Phasen als Zyklus (s. Bild 7-29).

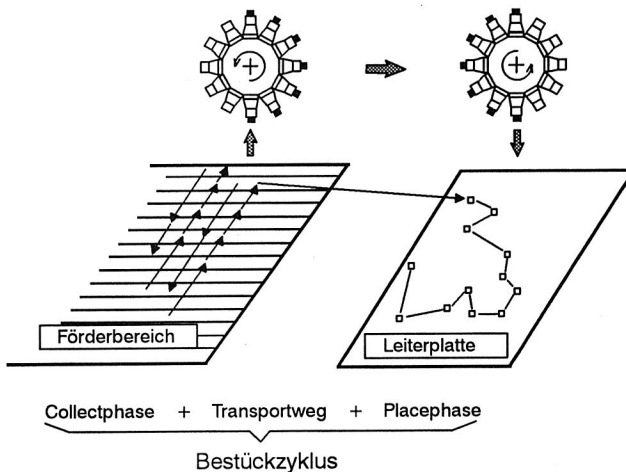


Bild 7-29: Bestückzyklus eines Collect and Place-Automaten

Weiterentwicklungen des Collect and Place-Automaten führten zu einer Verdoppelung der Bestückköpfe und somit zu einer Parallelbearbeitung. Für diesen Zweck wurde der Begriff Doppelzyklus eingeführt. Unter einem Doppelzyklus wird sinngemäß jeweils ein Zyklus der einzelnen Köpfe verstanden, der entsprechend dem Bestückprinzip verzahnt parallel abläuft (s. Bild 7-30). Die Collectphase des Zyklus des ersten Kopfes überschneidet sich zeitlich mit der Pickphase des zweiten Zyklus.

Wie schon bei der Problembeschreibung der Klasse III liegt auch bei diesem Maschinentyp kein linearer Zusammenhang zwischen Spuroptimierung und Sequenzoptimierung vor, so daß eine Trennung der beiden Optimierungsschichten nicht sinnvoll erscheint.

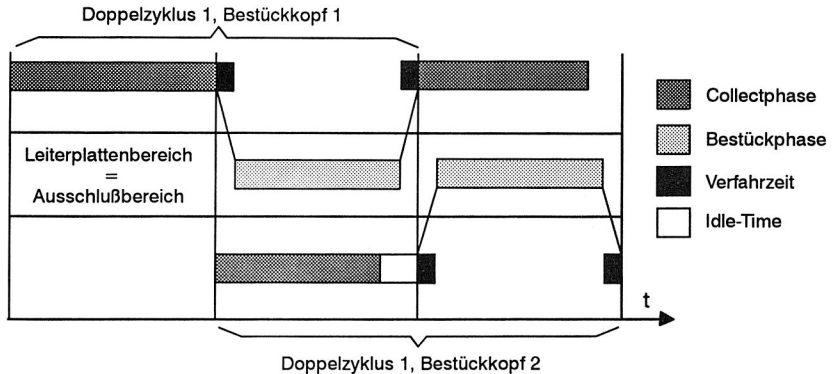


Bild 7-30: Zeitdiagramm zur Darstellung eines Doppelzyklus

Durch Analysen des Bestückprinzips und der einzelnen Kopffunktionen läßt sich ein relativ hoher Fixzeitsockel für die Zyklen erkennen, der in erster Linie aus der Umdrehung des Revolverkopfes resultiert. Diese Zeit wird benötigt, um nach Aufnahme oder Bestückung eines Bauelements ein Segment weiterzudrehen. Abgeleitet aus der Erkenntnis, daß vollbesetzte Zyklen kaum weniger Zeit benötigen, ergibt sich als Optimierungsziel mit dem höchsten Stellenwert die Minimierung der Gesamtzahl an Zyklen. Erreicht wird dies, wenn bei allen Zyklen die Bestückkränze vollbesetzt sind. Daraus ergeben sich zwei Anforderungen an die Optimierung: Die Berechnung einer optimalen Pipettenkonfiguration und eine ausgewogene Verteilung der Bauelemente auf die Förderbereiche. Bei einer Doppelportalmaschine gestaltet sich das Problem noch komplexer. Hier geht es darum abzuwägen, ob man eine über dem Minimum liegende Anzahl an Doppelzyklen akzeptiert und dafür möglicherweise eine bessere Synchronisation der Zyklen erreicht oder ob man in jedem Fall eine minimale Anzahl der Doppelzyklen anstrebt.

Weitere Probleme neben der Minimierung der Zyklen sind die Anordnung der Förderer innerhalb der Förderbereiche und die Zusammenstellung der Sequenzen je Zyklus, zur Minimierung der Fahrzeiten des Bestückkopfes. Eine wichtige Nebenbedingung diesbezüglich ist, daß die Menge der legalen Lösungen durch die Pipettenkonfiguration auf dem Bestückkopf eingeschränkt wird. Diese verhindert durch die Kompatibilitäten zwischen Bauelementtyp und Pipette(n), daß die Bauelemente auf beliebigen Positionen des Kranzes und damit in dem Zyklus vorkommen dürfen.

Zusammenfassend ergeben sich somit für den Optimierungsprozeß die folgenden Teilziele, die zu berücksichtigen sind:

- Festlegung einer legalen, optimierten Pipettenkonfiguration auf dem Bestückkopf(köpfen)

- Bestimmung einer minimalen Anzahl von (Doppel-)zyklen, die in ihrer Gesamtheit die komplette Bestückung abbilden
- Minimierung der Verfahrenswege der einzelnen Phasen der Doppelzyklen
- optimale Reihenfolgebildung der Doppelzyklen

Analog zu den geforderten Teilzielen wurde für das Gesamtproblem eine algorithmische Strukturierung in drei Teilebenen vorgenommen (s. Bild 7-31). Jede dieser Optimierungsebenen gibt an die tieferliegende Ergebnisse und Restriktionen weiter, die sich aus der zunehmenden Einengung des Lösungsraums ergeben.

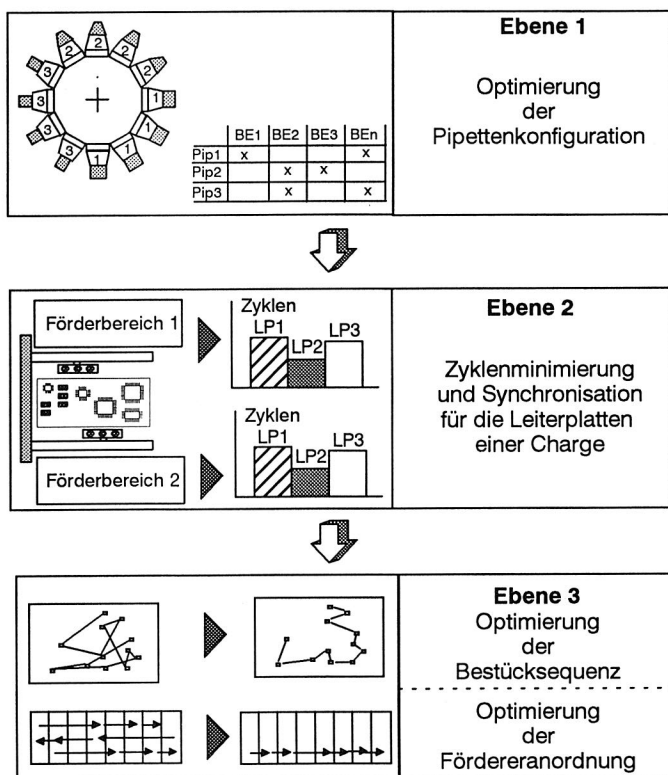
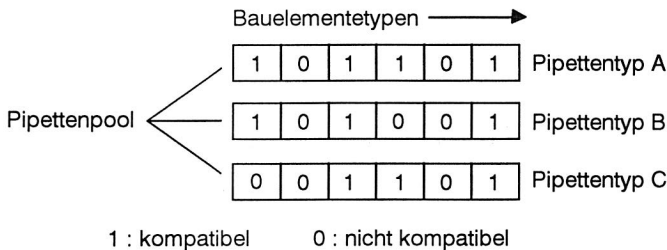


Bild 7-31: Strukturelle Aufgliederung des Spuroptimierers in Ebenen

### Ebene 1

Die Optimierung der Pipettenkonfiguration ist optional, da diese am Bestückkopf z. T. vorgegeben ist. Ziel dieses Optimierungslevels ist die Minimierung der Menge der zu benutzenden Pipettentypen. Dazu wird durch das Eingabemodul des Spuroptimierers für jeden Bauelementtyp, im Rahmen der typspezifischen Aufbereitung der

Daten aus der Hauptspeicherstruktur, eine Tabelle mit den kompatiblen Pipetten erzeugt. Um die Kompatibilität aller Bauelementtypen zu möglichst vielen Pipettentypen auf den Bestückköpfen zu maximieren und dadurch die Auslastung zu verbessern, wird die minimale Menge der notwendigen Pipettentypen ermittelt. Dies erfolgt durch eine einfache Betrachtung der Kompatibilitäten aller Pipetten zu den Bauelementtypen. Ein Pipettentyp kann genau dann aus der Menge der notwendigen Pipettentypen gestrichen werden, wenn alle Bauelementtypen, die durch diese Pipette bestückbar sind, auch von anderen Pipettentypen aus der Gesamtmenge bestückbar sind. Im Beispiel von Bild 7-32 kann der Pipettentyp C gestrichen werden, da er durch den Typen A und B substituiert werden kann.



*Bild 7-32: Abbildung des Pipettenpools*

## Ebene 2

Für die Ebene 2 und 3 wurden aufgrund der Komplexität und großen Zahl an Wechselwirkungen genetische Ansätze zur Lösungsfindung gewählt. Die Wahl des Codierungsansatzes ist hierbei der bestimmende Faktor für die Strukturierung der Optimierungsebene. Um die Bauelemente-Förderer-Kombinationen auf den Förderbereichen anzuordnen, wurden verschiedene Ansätze analysiert und verglichen:

- Codierung der Bauelemente und der zugehörigen Förderer in einem String. Optimierung als Einheit.
- Optimierung der Bauelementeverteilung und der Förderer getrennt
  - Codierung in einem hybriden Genstring
  - durch zwei völlig getrennte genetische Algorithmen
  - Genetischer Algorithmus nur zur Bauelementeverteilung, kombiniert mit einer Rüstheuristik

In Anbetracht der Tatsache, daß der Großteil des Optimierungspotentials auf der Ebene der Zyklenminimierung liegt und diese primär von der Aufteilung der Bauelementtypen abhängt, wurde die Strategie eines genetischen Algorithmus für die Bauelementeverteilung mit anschließender Fördererheuristik implementiert. Weitere entscheidende Vorteile dieser Methode sind die zu erwartenden besseren Laufzeiten und die Verwendung eines Codierungsalphabets mit nur 2 Elementen. Strings mit illegalen Allelekombinationen können ausgeschlossen werden, so daß keine Repa-

raturfunktionen innerhalb des genetischen Algorithmus nötig sind und das ganze Spektrum an bekannten Operatoren verwendbar ist. Nachteilig ist der höhere Implementierungsaufwand für die Rüstheuristik.

Neben den genetischen Operatoren ist der entscheidende Faktor im Ablauf des genetischen Algorithmus die Fitneßfunktion zur Bewertung einzelner Strings der Population. Diese Funktion muß als Ergebnis ein Maß für die Gesamtbestückzeit aller Leiterplatten der betrachteten Charge liefern. Auf den Ebenen der Zyklen- bzw. Doppelzyklenminimierung besteht dieses Maß in der Anzahl der notwendigen Zyklen. Die hierfür notwendigen Parameter sind die Pipettenkonfiguration, die Beschreibungsdaten der Leiterplattentypen und die Verteilung der Bauelementtypen auf die Förderbereiche, die in dem betrachteten String codiert sind. Die Vorgehensweise der Fitneßfunktion besteht darin, für jede Position auf dem(n) Bestückkopf(köpfen) jeweils ein Bauelement aus dem zugeordneten Förderbereich zu wählen. Jeder Position auf dem(n) Bestückkopf(köpfen) wird eine Bestückposition auf der Leiterplatte zugeordnet. Es muß dabei lediglich sichergestellt sein, daß der Bauelementtyp in dem aktuellen Förderbereich vorhanden ist, wobei Abholpositionen zu diesem Zeitpunkt noch nicht relevant sind. Die Zuordnung der Bestückpositionen geschieht leiterplattenweise, d.h. jede Bestückposition eines Leiterplattentyps muß einem Zyklus zugeordnet sein. Der bestimmende Faktor für die Bestückmöglichkeiten der aktuell betrachteten Kopfposition bei der Zyklusbestimmung ist die Pipettenkompatibilität der gerüsteten Pipette. Zur Minimierung der Zyklenzahlen muß sichergestellt sein, daß die Auslastung des Bestückkopfes oder der -köpfe maximal wird.

Die Vorgehensweise der Fitneßfunktion gestaltet sich folgendermaßen: Es werden ein Startbestückkopf (bei Doppelportalmaschine) und eine aktuelle Kopfposition gewählt. Für die Leiterplattendaten wird die erste Leiterplatte der Charge selektiert. Auf der aktuellen Kopfposition ist jeweils die Pipette für das nächste zu bestückende Bauelement gerüstet. Die Funktion muß demzufolge den Pipettentyp und die zu diesem Typ kompatiblen Bauelemente ermitteln. Anschließend sucht sie eine noch nicht abgearbeitete Bestückposition auf der Leiterplatte, deren Bauelementtyp in dem momentan ausgewählten Förderbereich vorkommt und kompatibel zu der Pipette ist. Die entsprechende Bestückposition auf der Leiterplatte wird als verwendet markiert und im weiteren Verlauf nicht mehr betrachtet. Konnte sie keine passende Bestückposition finden, muß die Pipettenposition des Kopfes leer bleiben. Dieser Vorgang wiederholt sich, bis der gesamte Kopf abgearbeitet ist. Danach findet bei Doppelportalmaschinen ein Bestückkopfwechsel und damit einhergehend ein Förderbereichswechsel statt. Sind alle Bestückpositionen eines Leiterplattentyps abgedeckt, wird ein neuer Leiterplattentyp verwendet, bis die gesamte Charge abgearbeitet ist. Auf diese Weise kann die minimale Anzahl an (Doppel)zyklen ermittelt werden.

Das Kernstück der Ebene 2 bildet der genetische Algorithmus zur Bauelementeverteilung auf die Förderbereiche und die nachgeschaltete Rüstheuristik, die aus der generierten Bauelementeverteilung hinsichtlich Fördererrestriktionen und anderer technischer Randbedingungen, entsprechend dem Maschinenmodell, eine zuläs-

sige Rüstung erzeugt. In einer Initialisierungsphase wird aus der übergebenen Startrüstung ein Pool aller nicht festgerüsteten Bauelemente gewonnen, in dem die Zugehörigkeit zu einem Förderbereich gekennzeichnet ist. Diese Bauelementetypen sind Bestandteil der Optimierung auf dieser Ebene und ihre Verteilung auf den Förderbereichen wird in den Strings codiert. Die Anzahl der Bauelementetypen ist entsprechend mit der verwendeten Stringlänge der Gene gleichzusetzen. Ein weiterer Parameter des genetischen Algorithmus ist die verwendete Populationsgröße, die über eine Funktion in Abhängigkeit von der Poolgröße errechnet wird.

Ein weiterer Initialisierungsschritt ist der Aufbau einer Startpopulation, die den Startpunkt für den genetischen Prozeß darstellt. Im Anschluß daran läuft in einer Schleife der genetische Algorithmus ab. Als Abbruchkriterium dient eine Funktion, die überprüft, ob es in der Ermittlung einer neuen Population zu einer Verbesserung der bestehenden besten Lösung gekommen ist. Mittels eines Zählers kann die maximale Zahl festgelegt werden, in der keine Verbesserungen stattfinden.

Die Aufgabe der Rüstheuristik, die sich dem genetischen Algorithmus auf der Ebene der Zyklusminimierung anschließt, ist es, den ermittelten Pool der Bauelemente mit möglichst wenigen Veränderungen unter Berücksichtigung aller gegebenen technischen Restriktionen auf eine korrekte Rüstung abzubilden. Sie muß sicherstellen, daß die gefundene Bauelementeverteilung auf die Förderbereiche möglichst optimal auf die Rüstung übertragen wird. Da in den meisten Fällen nicht alle Bauelementetypen des Pools in demselben Maß an der Zyklusbildung des jeweiligen Förderbereichs beteiligt sind, muß eine Abwägung in Form einer Priorisierung stattfinden. Die Heuristik behandelt bevorzugt Bauelemente mit einer höheren Priorität, um die Wahrscheinlichkeit zu maximieren, daß sie auf dem vorgesehenen Förderbereich gerüstet werden.

Die Rüstheuristik geht nach der Initialisierung und der Priorisierung der Bauelemente in drei Ebenen vor. In der ersten Ebene wird die maximale Anzahl an Bauelementen auf die bestehende Fördererrüstung der Förderbereiche plaziert, ohne Förderer zu bewegen. Fördererbewegungen finden erst in der zweiten Ebene statt. Unterschieden werden die Elementaroperationen Fördererverschiebung und Förderervertauschung, mit denen sich alle Rüstungsveränderungen darstellen lassen.

Die dritte Ebene dient zur Restrüstung der Bauelemente des Pools, die zu diesem Zeitpunkt noch nicht durch die Heuristik abgearbeitet werden konnten. Da in jedem Fall eine Rüstung dieser Bauelemente sichergestellt werden muß, wird die Priorisierung nicht in der Form wie in den vorangegangenen Ebenen berücksichtigt. Der komplette Ablauf der Ebene 2 ist in Bild 7-33 nochmals im Überblick dargestellt.

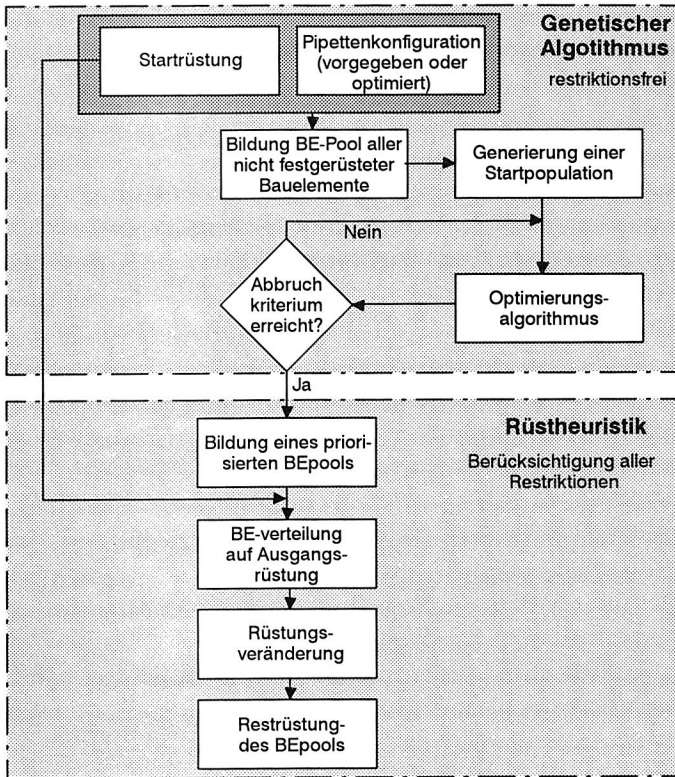


Bild 7-33: Ablaufstruktur der Ebene 2

### Ebene 3

Die Struktur der Ebene 3 ähnelt grundsätzlich dem Aufbau der Ebene für die Zyklenoptimierung. Für den kompletten Ablauf werden jedoch zwei genetische Algorithmen verwendet, für die Berechnung der Bestückpositionsreihenfolge innerhalb eines Zyklus und für die eigentliche Spuroptimierung. Die genetischen Algorithmen benötigen zwei Eingabeparameter, die vollständige Rüstung der zweiten Optimierungsebene sowie eine komplette Darstellung der gesamten Verfahrenwege in Form von (Doppel)zyklen.

Da bei Eintritt in die dritte Ebene nur eine aktualisierte Rüstung vorliegt, besteht der erste Schritt in der Generierung des Gesamtverfahrenweges. Die Optimierung des Zyklenverfahrenweges verbessert im Anschluß die Einzelzyklen dieses Gesamtverfahrenweges des Bestückvorgangs. Diese Berechnung bezieht sich dabei nur auf die Bestückpositionen, da die optimierten Bauelementepositionen auf dem Förderbereich erst durch die Spur- bzw. Pickphasenberechnung festgelegt werden. Die Bestückreihen-

folge, der einem Zyklus zugeordneten Bestückpositionen, legt auch die entsprechende Abholreihenfolge auf den Förderbereichen fest. Die Spuroptimierung ist also funktional von der Berechnung des Zyklusverfahrweges abhängig, wobei der größere Einfluß auf die Spuroptimierung jedoch sicherlich in der Festlegung der Zuordnung von Bestückpositionen zu den Zyklen liegt und weniger in der Bestückreihenfolge.

Eine Initialisierungsphase legt zu Beginn eine Verfahrenwegrepräsentation fest. Aufgabe ist es, die gesamten Verfahrenwege des Automaten während der Bestückung einer kompletten Charge festzuhalten. Der Verfahrenweg wird sowohl in der Reihenfolgeoptimierung eines Zyklus als auch für die Spuroptimierung benutzt. Die Verfahrenwegsinitialisierung erfolgt mittels eines Nearest Neighbour-Algorithmus, der eine zyklusorientierte bestückpositionsbezogene Clusterung der Leiterplatte vornimmt.

Die Optimierung der Bestückreihenfolge beruht auf der Möglichkeit, die Bestückpositionen der Placephase permutieren zu können, ohne die Bestückinhalte zu verändern. Das Optimierungskriterium ist dabei die Weglänge zwischen den Positionen der jeweiligen Bestückreihenfolge. Der verwendete genetische Algorithmus gleicht im Aufbau grundsätzlich dem aus der Optimierungsebene 2, wobei aber ein zusätzlicher Mechanismus die Berücksichtigung der Pipettenrestriktion sicherstellen muß.

Die Spur- bzw. Pickoptimierung ist das letzte Modul in der Optimierungsebene der Verfahrenwegoptimierung. Da zu diesem Zeitpunkt sowohl die Menge der Bestückpositionen der einzelnen Zyklen zugeordnet ist als auch deren Reihenfolge bekannt ist, handelt es sich um eine parallele Verfahrenwegoptimierung der Pickphase aller Zyklen. Die Spuroptimierung bezieht sich auf jeweils einen Förderbereich. Für den genetischen Algorithmus wurde eine hybride Codierung von Bauelementen verwendet. Eine Aufspaltung analog zur Ebene 2 hätte dazu geführt, daß die Überprüfung einer legalen Rüstung für jeden String der Population hätte ausgeführt werden müssen, was einen großen Zusatzaufwand bedeutet.

Die Aufgabe der Fitneßfunktion der Spuroptimierung besteht in einer quantitativen Bewertung verschiedener Rüstvorschläge für einzelne Förderbereiche, die in codierter Form in den Strings der Population vorliegen. Dabei muß die Fitneßfunktion in der Lage sein, auch zwischen minimal unterschiedlichen Rüstungen zu unterscheiden, wie sie sich beispielsweise durch Verschiebungen eines Bauelementes um einen Modulplatz ergeben können. Mit Hilfe der bereits vorgestellten Zeitmodelle wird eine exakte Simulation des kompletten Bestückvorgangs für die manipulierten Rüstungen vorgenommen und als Fitneßwerte übernommen.

Ein Kriterium für die Güte eines Ergebnisses anzugeben, ist aufgrund der Komplexität der Aufgabenstellung nicht möglich. Um eine Aussage über die Tests der Optimie-



rungsläufe zu machen, wurde für den Bereich der Zyklusminimierung ein theoretisch erreichbarer Auslastungsgrad eingeführt:

$$\Theta = \frac{A_{BP}/A_{PP}}{Z}$$

$\Theta$  = Auslastungsgrad  
 $A_{BP}$  = Anzahl Bestückpositionen  
 $A_{PP}$  = Anzahl Pipettenpositionen  
 $Z$  = optimierte Zyklenzahl

Die Variable  $\Theta$  kann im Idealfall den Wert 1 erreichen. Da auftretende Restriktionen das Erreichen des Idealwertes verhindern, wird bei steigender Anzahl von Restriktionen eine Aussage über die Güte eines Optimierungsergebnisses zunehmend schwieriger. In mehreren Testläufen für eine SIPLACE 80S-Maschine wurden Leiterplatten mit unterschiedlicher Anzahl an Bauelementen, erforderlichen Pipettentypen und Anzahl benötigter Förderer getestet. Bei großen Leiterplatten und wenig Restriktionen, bedingt durch die Pipettentypen, wurden im Durchschnitt sehr gute Ergebnisse für die Zyklenauslastung erreicht. Optimierungsprobleme mit wenig Bauelementtypen und im Verhältnis dazu vielen Pipettentypen führten zu einem deutlichen Absinken des  $\Theta$ -Wertes. In diesen Fällen ist eine Aussage über die Güte des Optimierungsalgorithmus schwer möglich, denn das wirklich zu erreichende theoretische Optimum könnte durchaus in der Nähe des erreichten Wertes liegen.

Leiterplatte	Pipetten-typen	Bauelemente-typen	Anzahl Bestückpos.	Doppel-zyklen	$\Theta$	Optimierungsdauer [s]
A	4	9	64	5	0,6	6
B	3	8	108	8	0,625	11
C	2	7	99	6	0,833	12
D	2	12	96	4	1	12
E	2	32	856	38	0,94	312
F	2	28	1117	48	0,97	408

Tabelle 7-4: Optimierungsläufe zu einigen Testleiterplatten

Um die Methode des kompletten Optimierungsvorganges zu validieren, wurde zusätzlich eine ganz einfache Testplatte verwendet, bei der das beste Ergebnis von Spur- und Verfahrensoptimierung deutlich zu erkennen ist (s. Bild 7-34). Getestet wurde mit unterschiedlichen Laufzeiten. Die optimierte Rüstung wurde bei allen Testläufen erreicht. Bei Laufzeiten von 20s und 30s erfolgte für den Verfahrensweg in drei von fünf bzw. zwei von fünf Fällen ein Sprung über zwei Bestückpositionen und zurück (im Beispiel: von A zum nächsten A in x-Richtung, zurück nach B in x-Richtung und weiter zum nächsten B in x-Richtung). Ab 30s Laufzeit wurde auch hier das Optimum erreicht.

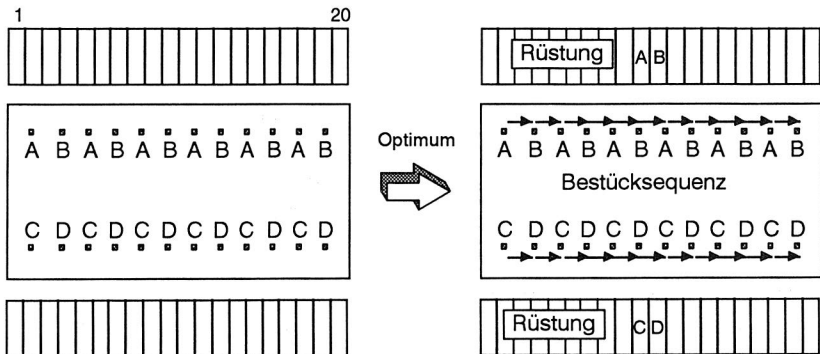


Bild 7-34: Testleiterplatte mit bestimmbarem Optimum

## 7.6 Optimierung der Bestücksequenzen

Die Optimierung der Bestücksequenz ist die unterste Schicht in dem entwickelten Optimierungsmodell. Wie schon im letzten Kapitel erläutert, werden bei einigen Maschinenmodellen Schicht 3 und Schicht 4 aufgrund der starken Wechselwirkungen in einem Algorithmus behandelt.

Für die verbleibenden Maschinenklassen läßt sich die Bestücksequenzoptimierung auf ein TSP-Problem zurückführen. Die Modellbildung am Beispiel der Pick and Place-Maschine gestaltet sich wie folgt:

Die Pick- und Placephase sowie der Transportweg gehen als fixe Werte in die Berechnung des Fahrweges ein. Da die Abholpositionen und die Bestückpositionen der Bauelemente festliegen und somit ein Teil des Fahrweges vorgegeben sind, lassen sich lediglich die Wege von den Bestückposition zurück zu einer neuen Abholposition minimieren (s. Bild 7-35).

Bezeichnet man die Übergänge von Bauelement<sub>i</sub> zu Bauelement<sub>i+1</sub> als Kanten und die Summe der konstanten Werte aus Transportweg, Pick- und Placephase als Knoten, so erhält man einen Graphen G. Sind auf der Platine n Bestückpositionen vorhanden, gehen n-1 Kanten zu allen Knoten. Gewichtet man die Kanten (i,j) mit dem Aufwand  $c_{ij}$ , gelangt man zum erwähnten TSP-Problem. Die Kosten  $c_{ij}$  entsprechen der Zeit, die der Bestückkopf von der Abholposition i zur Bestückposition j benötigt. Ziel ist es, eine Sequenz zu finden, deren Summe an Gewichten minimal ist.

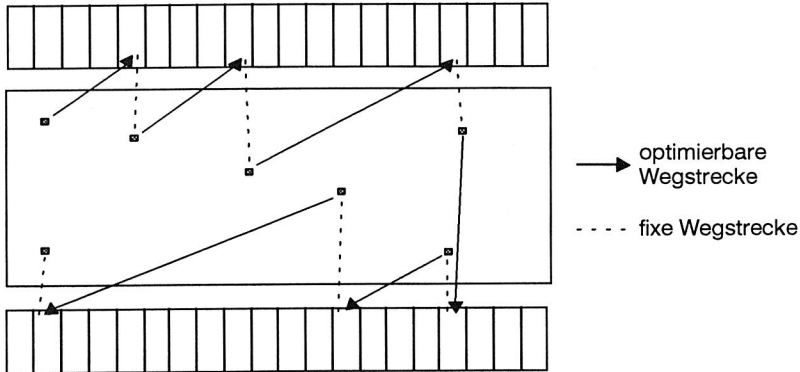


Bild 7-35: Bestücksequenz und fixe Wegstrecken

Die Komplexität des TSP-Problems erhöht sich im speziellen Fall dadurch, daß eine Asymmetrie vorliegt und die sogenannte Dreiecksungleichung nicht erfüllt ist. Eine Asymmetrie, als spezielle Eigenart des TSP-Problems liegt vor, wenn das Gewicht der Kanten von  $i$  nach  $j$  und von  $j$  nach  $i$  unterschiedlich ist. Um dieses Problem zu umgehen, wurde der Graph um zwei gegenseitig gerichtete Kanten mit unterschiedlicher Gewichtung erweitert. Auch die Gültigkeit der Dreiecksungleichung, die besagt, daß das Gewicht der Kanten  $c_{ij}$  kleiner sein muß als die Summe der Gewichte  $c_{ik}$  und  $c_{kj}$ , ist in diesem Fall nicht gewährleistet. Aus diesem Grund addiert man zum Gewicht der Kante  $(i,j)$  die Zeiten von Transportweg, Pick- und Placephase hinzu.

Zur Lösung des Problems wurden folgende Algorithmen implementiert und getestet:

- Assignment and Patching
- Nearest Neighbour
- Nearest Insertion
- Minimum Spanning Tree
- Maschinenspezifischer Positionssortierer
- Cheapest Insertion

und zwei tourverbessernde Algorithmen

- Städte-Permutationsverfahren
- Städtefolgen-Permutations-Verfahren

Nach Untersuchungen des Laufzeitverhaltens (gemessen an einer Workstation HP9000/345) wurden deutliche Unterschiede der einzelnen Verfahren festgestellt (s. Bild 7-36). Für Leiterplatten mit mehr als 500 Bestückpositionen bieten lediglich die Algorithmen Nearest Neighbour und der maschinenspezifische Algorithmus akzeptable Ergebnisse. Die Methode Minimum Spanning Tree scheidet außer zu Testzwecken wegen der hohen Laufzeit schon für Leiterplatten  $\geq 100$  Bestückpositionen aus.

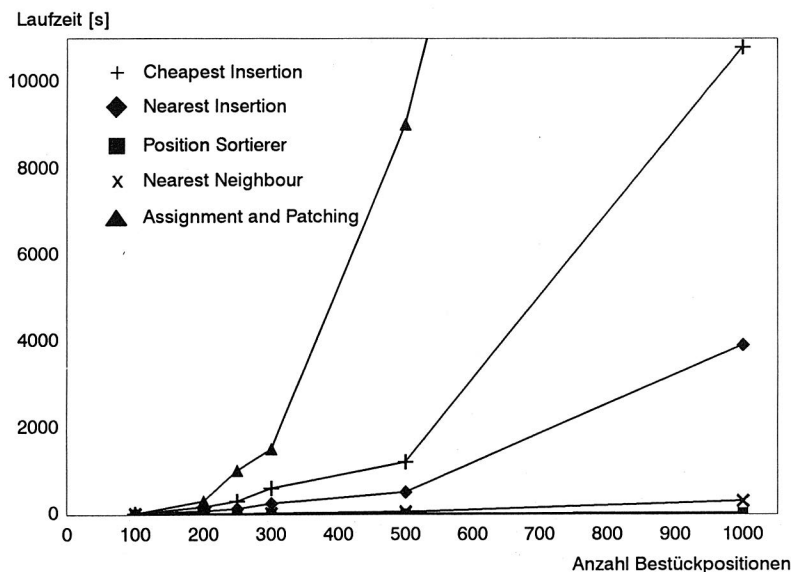


Bild 7-36: Rechenzeiten der Algorithmen für steigende Anzahl an Bestückpositionen

Für die zu erreichenden Optimierungspotentiale wurden Leiterplatten unterschiedlicher Abmessungen und variierender Anzahl an Bestückpositionen getestet (s. Bild 7-37). Bei steigender Zahl der Bauelemente erkennt man einen größeren Abstand zwischen den Lösungen. Den Einfluß der Leiterplattenmaße erkennt man aus dem Vergleich des jeweils besten und schlechtesten Ergebnisses innerhalb einer Grafik (der zugehörige Wert ist jeweils über dem größten Balken angegeben). Wie zu erwarten war, ergibt sich mit zunehmender Leiterplattengröße eine steigende Differenz zwischen bestem und schlechtestem Ergebnis, was besagt, daß mit zunehmender Leiterplattengröße das Optimierungspotential pro Leiterplatte steigt. Aus der Testreihe erhält man zudem eine Aussage über die zu erzielenden Zeiteinsparungen durch die Optimierung auf dieser letzten Ebene. Für die Leiterplatte mit 500 Bestückpositionen ergeben sich je nach Leiterplattengröße Bestückzeiten von 500–700 s. Für die größte getestete Leiterplatte mit den meisten Bestückpositionen beträgt die maximale Differenz  $|\max - \min|$  51 Sekunden. Dies entspricht 10% der Bestückzeit. Bei den kleinen Leiterplatten wurden hingegen Einsparungen von maximal 2% erreicht.

Zusammenfassend konnte festgestellt werden, daß für die getesteten Varianten die Kombination des Positionssortierers mit der Städtefolgen-Permutation hinsichtlich Laufzeit und Optimierungsergebnis die besten Ergebnisse erbrachte.

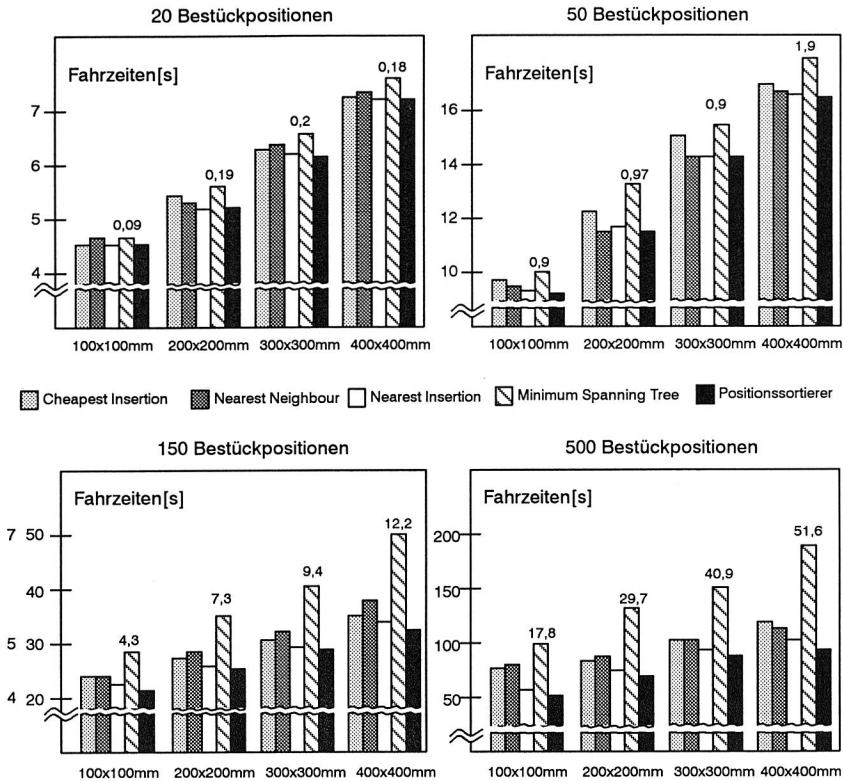


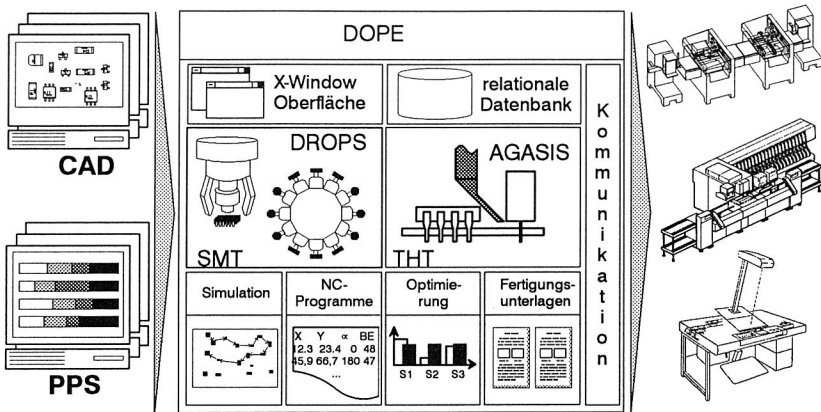
Bild 7-37: Optimierungsergebnisse bei unterschiedlichen Leiterplattengrößen und Bestückpositionen

## 7.7 Einbindung der Optimierung für Automaten der Durchsteckmontage

Ein weiterer funktionaler Bestandteil des Optimierungsmoduls ist die Bestücksequenzoptimierung und Simulation für Leiterplatten der Through Hole Technology. Durch die Integration dieser weiteren Komponente besteht die Möglichkeit, mit Hilfe des Optimierungsmoduls, das auch als eigenständiges Planungssystem verwendet werden kann (s. Bild 7-38), den kompletten Bestückbereich zu optimieren.

Im Bereich der Durchsteckmontage erfolgt im Gegensatz zur SMT eine Aufteilung der automatischen Bestückung hinsichtlich der Bauelementtypen. Neben einigen Sonderautomaten, die den gesamten bedrahteten Bereich abdecken, existieren in

der Regel die Kategorien Axial, Radial- und DIP-Automat. Neben dem gleichen Problem der Bestücksequenzoptimierung ist im bedrahteten Bereich die Kollisionsfreiheit des Bestückvorgangs ein Hauptgesichtspunkt.



*Bild 7-38: Datenbankgestütztes Optimierungssystem für die Elektronik (DOPE)*

Durch die verwendeten Ober- und Unterwerkzeuge dieser Maschinentypen ist ein Check zur Vermeidung von Kollisionen der einzelnen Werkzeuge dieser Automatentypen durchzuführen (s. Bild 7-39). Optimierungsziel ist demzufolge die zeit(kosten)günstigste Bestückreihenfolge ohne, oder mit den wenigsten Kollisionen.

Dedektierte Bauelemente/Greifer Überschneidungen, die bedingt durch eine feste Reihenfolge der Automaten vom Algorithmus nicht vermieden werden können, werden ausgegeben, wobei sich zwei Alternativen ergeben:

- Die Bauelemente, die eine Kollision hervorrufen und die wenigsten Bestückkosten verursachen, werden als Handbestückung ausgewiesen.
- Veränderungen des Layouts der Leiterplatte, um eine komplette automatisierte Bestückung zu gewährleisten.

Wurde vom Algorithmus eine komplett kollisionsfreie, und verfahrwegoptimierte Bestücksequenz generiert, wird diese in die Datenbank geschrieben.

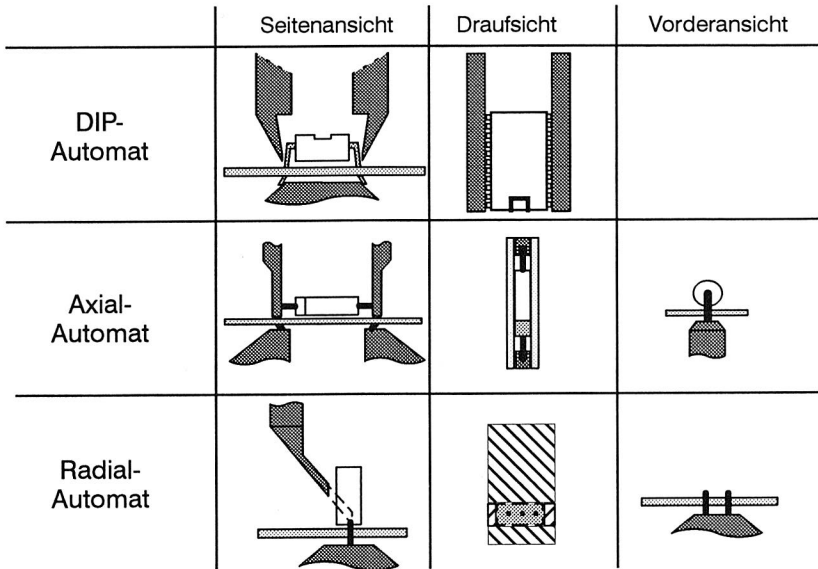


Bild 7-39: Greiferausprägungen der bedrahteten Bestückungsautomaten

## 7.8 Zusätzliche Optimierungsmöglichkeiten

### 7.8.1 Nutzenoptimierung

Eine weit verbreitete Methode in der Herstellung von Flachbaugruppen ist die sogenannte Nutzentechnik. Bei der Verwendung dieses Verfahrens wird darauf verzichtet, jede entwickelte Schaltung als einen Bestandteil auf einer Leiterplatte zu realisieren, stattdessen werden mehrere gleiche oder auch verschiedene Schaltungsentwürfe zu einem sogenannten Nutzen zusammengefaßt. Bevorzugt werden Standardleiterplattenabmaße wie das Europa- oder Doppeleuropaformat.

Die Methodik der Nutzentechnik bietet sowohl aus technischer als auch wirtschaftlicher Sicht eine Reihe von Vorteilen:

- Durch einheitliche Leiterplattenformate ergibt sich eine bessere und einfachere Handhabung der Leiterplatten sowohl innerhalb der Maschinen (einheitliche Breitereinstellung) als auch für den Transport (Magazine, Behälter etc.).
- Die Erhöhung der Bestückpositionen pro Leiterplatte bewirken eine Reduzierung der Nebenzeiten, da sich die Leiterplattenwechselzeiten pro Maschine minimieren.
- Komplexe Einzelschaltungsgeometrien können zu einer einfach handhabbaren Standardleiterplattenkontur zusammengefaßt werden. In einigen Fällen kann dies zusätzlich mit einer besseren Materialausnutzung gekoppelt sein.

- Verschiedene Einzelschaltungen können zu einem Leiterplattenlayout zusammengefaßt werden, so daß die Erstellung einer Platine nur einmalig anfällt.

Als Nachteil ist anzuführen, daß durch die Nutzentchnik ein zusätzlicher Prozeßschritt, die Trennung der gesamten Leiterplatte in Einzelschaltungen, anfällt.

Ein Optimierungsproblem, das sich in diesem Zusammenhang ergibt, ist die Erstellung eines flächen- und stückzahloptimierten Leiterplattenlayouts. D.h für unterschiedliche Einzelschaltungen, die in eventuell verschiedenen Stückzahlen zu produzieren sind, ist für sämtliche Einzelschaltungen ein einziges Leiterplattenlayout, in einer minimalen Gesamtbestückzahl zu erstellen.

Die sich ergebende Optimierungsaufgabe kann der Klasse der Verschnitt- bzw. Anordnungsproblemen zugeordnet werden. Im speziellen handelt es sich hierbei um ein zweidimensionales Verschnittproblem. Da es in der Praxis meist um Nutzenanordnungen kleiner 10–15 geht und dies als kleines Problem dieser Klasse gewertet wird, wurde als eine Lösungsvariante ein Algorithmus zum sequentiellen, deterministischen Anordnen gewählt [106]. Die zu positionierenden Figuren (Einzelschaltungen) werden hierbei in einer festgelegten Reihenfolge, in dem zur Verfügung stehenden Gebiet, (Leiterplattenfläche) angeordnet (s. Bild 7-40). Das Gebiet G, das noch zur weiteren Anordnung von Einzelschaltungen zur Verfügung steht und das belegte Gebiet  $G^*$  werden als Figuren angesehen. Die Anordnung einer neuen Figur (Einzelschaltung) wird somit auf gegenseitige Lage zweier Figuren  $G^*$  und T zurückgeführt. Die Kurve, die zur dichten Anordnung von T an  $G^*$  benötigt wird, bezeichnet man als Hodograph.

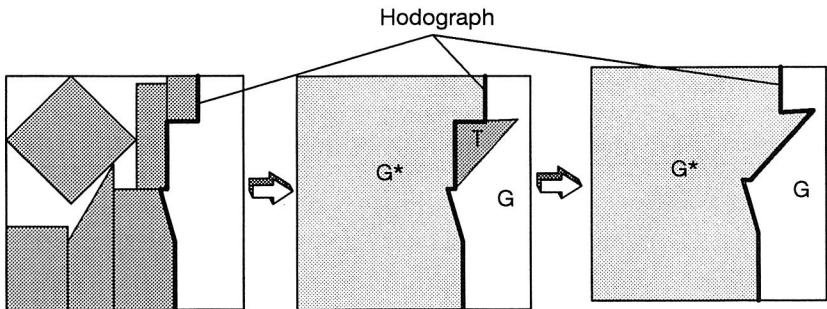


Bild 7-40: Sequentielles Anordnen

Aus der Aufgabenstellung und für die Lösungsmöglichkeit des sequentiellen Anordnens ergibt sich folgender Gesamttablauf:

1. Auswahl der Einzelschaltung und Bestimmung der erforderlichen Einzelstückzahlen
2. Einlesen der geometrischen Daten aus der Datenbank



3. Erstellung von Stückzahlalternativen mit dem Ziel, eine minimale Anzahl an Leiterplatten zu erhalten. Zur Bestimmung einer Obergrenze für die Einzelschaltungen, auf der zur Verfügung stehenden Leiterplattenfläche, wird ein einfaches Flächennäherungsverfahren angewandt:

$$\sum_{i=1}^{\text{Anz. ES}} \text{Fläche(ES}_i) \leq \text{Fläche(LP)}$$

4. Auswahl der zunächst günstigsten Stückzahlalternative und Generierung von Reihenfolgenalternativen.
5. Für die festgelegte Stückzahlalternative und die vorgegebene Reihenfolgealternative wird die Positionierung der Einzelschaltungen vorgenommen. Das Schema der Positionierung richtet sich nach folgender Vorgehensweise:
- Anordnung der Einzelschaltungen von links nach rechts,
  - nach jeder positionierten Einzelschaltungszeile wird mit einem Hodographen ein definierter Abschluß der Figur gezogen.

Durch die Positionierung der Einzelschaltungen in 0° und 90° Lage ergeben sich für die jeweils nächste Einzelschaltung vier Alternativen (s. Bild 7-41), die in einer systematischen Reihenfolge abgearbeitet werden. Der Abstand des Hodographen von den Einzelschaltungen ergibt sich aus den Mindestabständen der Einzelschaltungen untereinander. Konnte die Positionierung nicht komplett vorgenommen werden, wird die nächste Reihenfolgealternative herangezogen. Wurde auch hier kein realisierbares Ergebnis gefunden, gelangt man zur nächsten Stückzahlalternative.

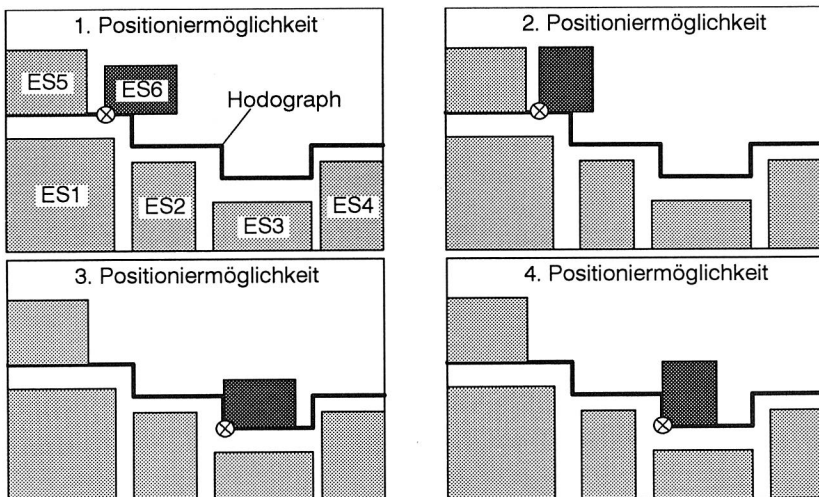


Bild 7-41: Positionierungsmöglichkeiten der Einzelschaltungen

In Tabelle 7-5 ist ein Beispiel einer Testkonstellation dargestellt. Die geometrischen Eingangsdaten und die zu produzierenden Stückzahlen können der Tabelle entnommen werden. Die grafische Anordnung des Testlaufes zeigt Bild 7-42.

Eingabedaten:

Objekte	Länge [mm]	Breite [mm]	Randabstände [mm]			
			oben	unten	links	rechts
Leiterplatte	180	140	0	0	15	15
Einzelerschaltung A Stückzahl: 37	40	20	10	10	10	10
Einzelerschaltung B Stückzahl: 27	70	50	10	10	10	10
Einzelerschaltung C Stückzahl: 30	30	20	10	10	10	10

Ergebnisdaten:

Anzahl der zu fertigenden Platinen			10 Stück
Flächenausnutzung			73,8 %
Rechenzeit			12 s
Einzelerschaltungen	A	Anzahl pro Leiterplatte	4
		überproduzierte Anzahl	3
	B	Anzahl pro Leiterplatte	3
		überproduzierte Anzahl	3
	C	Anzahl pro Leiterplatte	3
		überproduzierte Anzahl	0

Tabelle 7-5: Eingabedaten und Ergebnisdaten aus Testlauf

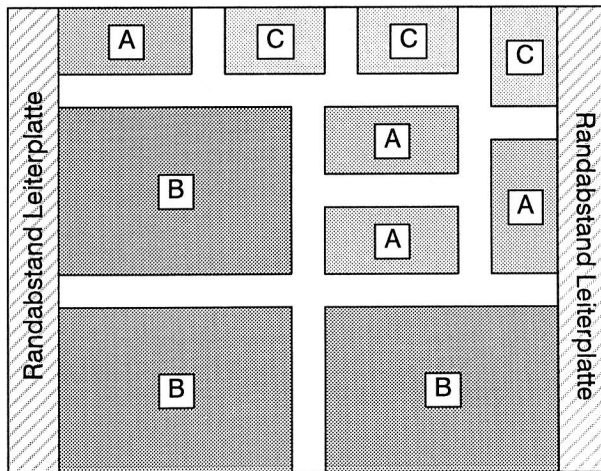


Bild 7-42: Leiterplattenlayout der Testkonfiguration

## 7.8.2 Ermittlung einer Basisrüstung

Eine weitere Optimierungsaufgabe ist die Ermittlung einer sogenannten Basis- bzw. Festrüstung für eine Bestückungsline oder für den gesamten Bestücksbereich. In vielen Unternehmen der Flachbaugruppenbestückung existieren eine bestimmte Menge an Flachbaugruppen, die in unterschiedlichen Stückzahlen und mit nicht vorhersehbaren Terminen immer wieder in die Fertigung eingelastet werden. Optimierungsaufgabe ist es, für dieses Spektrum eine Festrüstung an den Bestückungssystemen vorzusehen, um für alle Flachbaugruppentypen die Summe an Umrüstaufwendungen zu minimieren. Dabei bestimmt die Vorkommenshäufigkeit jeder einzelnen Flachbaugruppe in Form eines Gewichtungsfaktors, wie groß ihr Anteil an der Festrüstung ist. Bei der Festlegung von Festrüstanteilen ist vor allem darauf zu achten, daß genügend Stellplatzkapazität für die Förderer vorhanden bleibt, um sämtliche Bauelementetypen des Leiterplattenspektrums, auch für die, die nicht in der Festrüstung berücksichtigt werden, rüsten zu können.

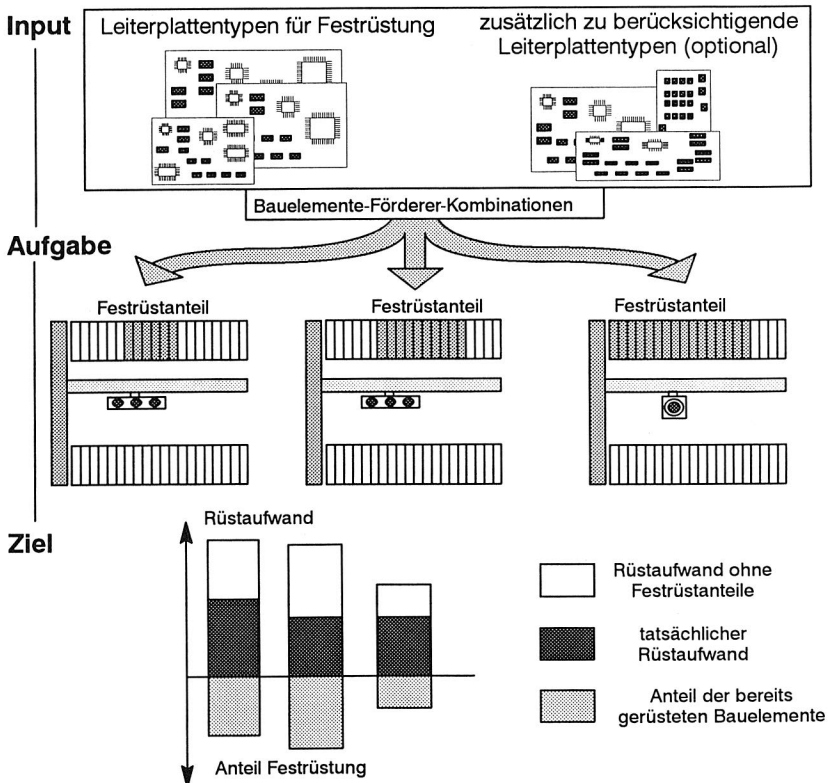


Bild 7-43: Optimierungsaufgabe: Berechnung von Festrüstanteilen

Um ein großes Maß an Flexibilität zu gewährleisten, wurden für die Optimierungsstrategie folgende Eingabemöglichkeiten zugelassen:

- Der Optimierungsalgorithmus übernimmt die Zuteilung der Festrüstanteile auf alle Bestückungssysteme für alle eingegebenen Leiterplattentypen.
- Der Benutzer nimmt eine Bindung von Leiterplattentypen an eine oder mehrere Bestückungssysteme vor.
- Jedem Leiterplattentyp kann zu jedem Bestückungssystem ein Gewichtungsfaktor zugeteilt werden.
- Es können Leiterplattentypen bestimmt werden, die nicht in der Festrüstung zu berücksichtigen sind, für die aber Stellkapazitäten vorzusehen sind.
- Ein Teil der Basisrüstung soll manuell fixierbar sein.

Die implementierte Lösungsstrategie beruht auf dem Prinzip der linearen Optimierung, kombiniert mit einer Evolutionsstrategie. Den prinzipiellen Ablauf der implementierten Lösungsstrategie beschreibt Bild 7-44. Die Bauelementegewichtung ist das Maß für die mittlere aufzuwendende Umrüstzeit pro Bauelement und Linie (inklusive der Förderereigenschaften). Die Bauelementedichte ist ein Maß für die Bauelementegewichtung, bezogen auf die beanspruchten Stellplätze.

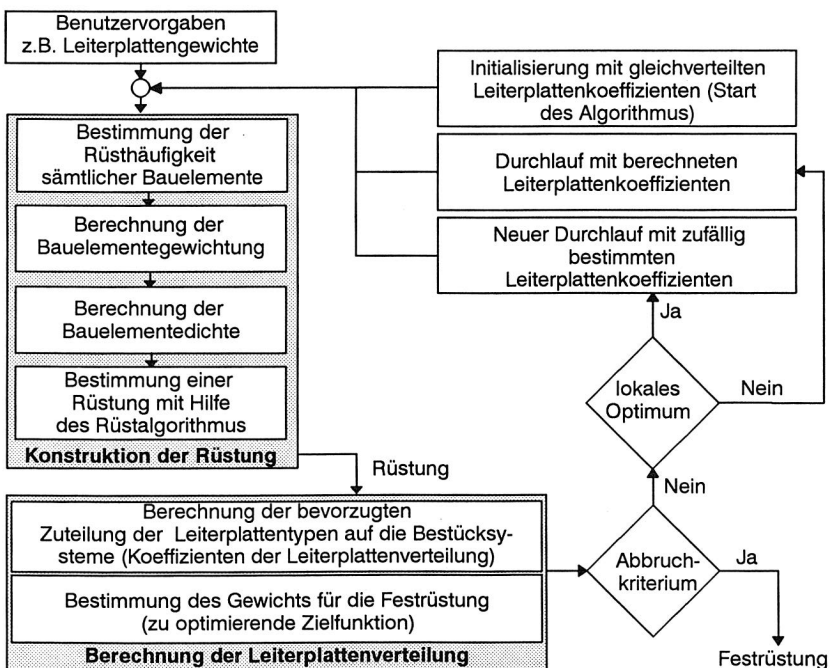


Bild 7-44: Methode zur Bestimmung von Festrüstungen

## 7.9 Visualisierung von Optimierungsergebnissen

Die darstellbaren Visualisierungsergebnisse sind die Resultate der einzelnen Optimierungsschichten und die vom Planungssystem vorgeschlagenen optimierten Rüstungen sowie Auftragszuteilungen. Einige Beispiele der Visualisierungsmöglichkeiten des Systems für die einzelnen Optimierungshierarchien zeigt Bild 7-45.

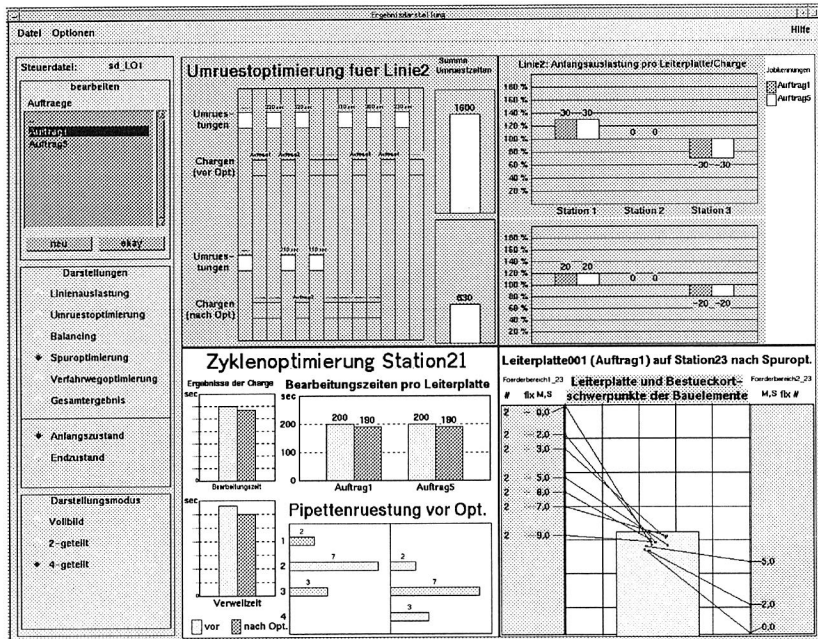


Bild 7-45: Visualisierung verschiedener Optimierungsergebnisse

Die darstellbaren Möglichkeiten im einzelnen sind:

- Auftragsreihenfolge pro Bestückungssystem
- Zusammenfassung der Aufträge zu Chargen
- Gesamtbearbeitungszeiten der Chargen und Zeitdauer der Umrüstungen
- Austaktung der Linie
- Pipettenkonfiguration der Bestückköpfe
- Simulation der Bestückwege
- Bearbeitungszeiten von Leiterplatten stationsweise und linienbezogen

Bei einigen Detaildarstellungen wird sowohl der Ausgangs- als auch der optimierte Endzustand visualisiert, so daß man Vergleiche ziehen kann und auch Ergebnisverbesserungen ablesbar sind. Außer der grafischen Ausgabe ist ein Ausdruck der Ergebnisse über einen realisierten Datentransfer in ein angekoppeltes Textverarbeitungsprogramm möglich.

tungssystem möglich. So können in tabellarischer Form Auftragszuteilungen, Auftragsreihenfolgen, Umrüstanweisungen, Rüstungen der Bauelemente auf den Bestückungssystemen und eine summarische Aufteilung der Bauelementtypen sowie der Bestückpositionen mit prozentualen Auslastungen ausgegeben werden.

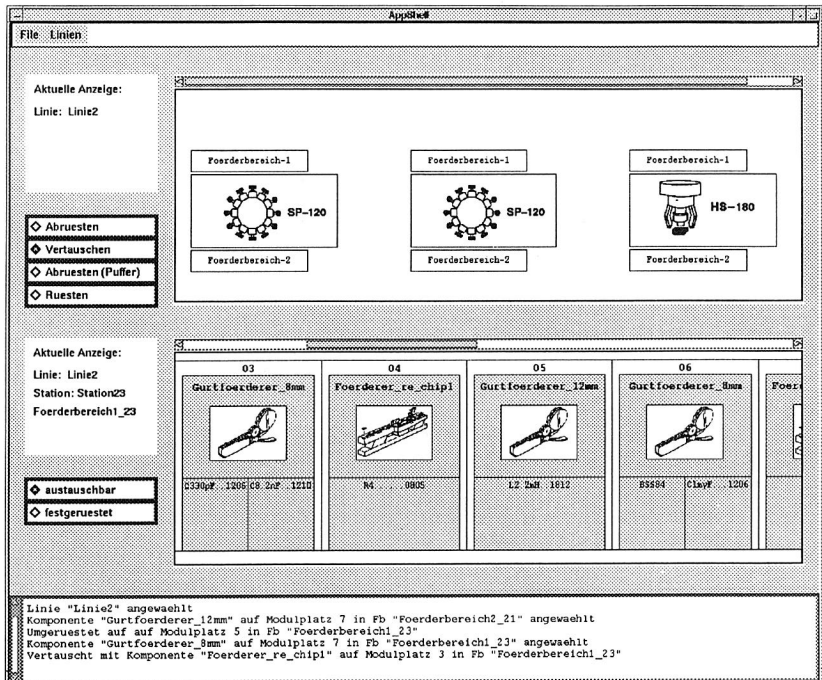


Bild 7-46: Ergebnisvorschlag zur Rüstung der Bestückungssysteme

Bevor die optimierte Rüstung mittels des entwickelten Postprozessors in das gewünschte Maschinenformat umgesetzt wird, hat der Anwender die Möglichkeit, mit einem Editor (s. Bild 7-46) die optimierte Rüstung grafisch zu visualisieren und gegebenenfalls noch zu modifizieren. Der Editor basiert auf einem grafischen, sensitiven Modell, in dem Linien, einzelne Stationen, Förderbereiche und die Zuführspuren durch Mausklick anwählbar sind. Mit Hilfe der zwei Darstellungsfenster sind Verschiebungen und Vertauschungen zwischen Förderbereichen und Stationen grafisch interaktiv durchführbar.

## 8 Entwicklung eines Moduls zur Auftragseinplanung

Termintreue, als eine der wichtigsten marktstrategischen Elemente, wird oft höher bewertet als der Marktpreis [2]. Insbesondere gilt dies für Firmen, die mit ihren Lieferanten eine Just-in-Time Logistikanbindung aufgebaut haben. Das vorliegende Kapitel beschreibt hierzu entwickelte Lösungen zur termingerechten Auftragseinplanung, zur Simulation von Umplanungen und zur Abschätzung von Fertigungskosten. Als Grundlage der Auftragseinplanung dienen die realisierten Zeit- und Rüstmodelle inklusive der Ergebnisse aus dem Optimierungsmodul. Weiterhin werden in diesem Kapitel die Erstellung von benötigten Fertigungsunterlagen sowie die Informationsgewinnung aus Planungsdaten und dem laufenden Prozeß beschrieben.

### 8.1 Entwicklung eines Werkskalenders

Wichtige Voraussetzung zur terminlichen Umsetzung der Zeitergebnisse und somit Grundlage aller Terminierungsmöglichkeiten ist die Entwicklung eines Werkskalenders. Dieser hat die Aufgabe, in einem Zeitmodell die Produktionsstunden eines Unternehmens abzubilden und eine Umsetzung von Nettozeiten in Bruttozeiten vorzunehmen, d.h. die Umrechnung einer Zeitdauer (Sekunden, Minuten etc.) in eine definierte Zeitaussage (22.09.1994, 8<sup>00</sup> Uhr bis 23.09.1994, 10<sup>30</sup> Uhr).

Das entwickelte Modell zur rechnerinternen Darstellung werkspezifischer Arbeitstage und Produktionszeiten beinhaltet mehrere zusammenhängende Zeittabellen (s. Bild 8-1)

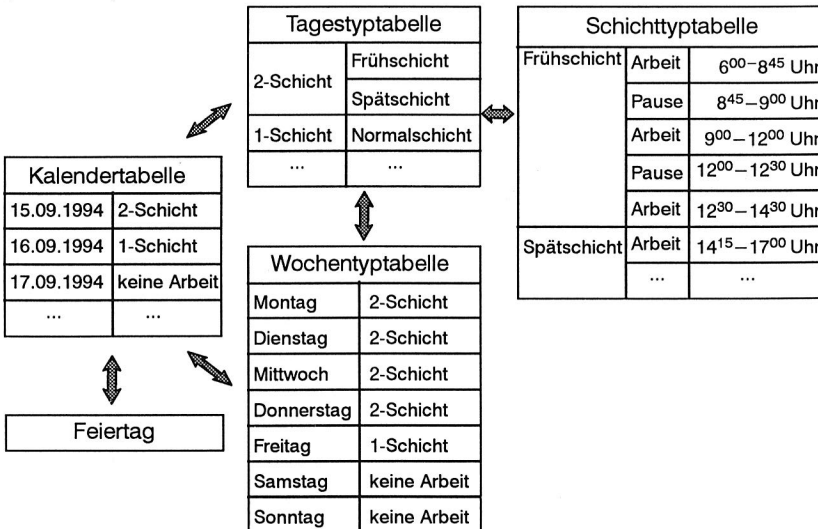


Bild 8-1: Zeitmodelle und ihre Zusammenhänge

Das entwickelte Werkzeug ermöglicht eine einfache und schnelle Erstellung eines produktionsspezifischen Kalenders mit Hilfe einer grafischen X11-Oberfläche (s. Bild 8-2). Durch den hierarchischen Aufbau Woche, Tag, Schichttyp, Zeittyp ist eine detaillierte Spezifikation eines Werkskalenders möglich. An der Schnittstelle zur Außenwelt bietet der entwickelte Werkskalender folgende Berechnungsmodi an:

- **Vorwärtsterminierung:**  
Vorgabe des Anfangstermins, Zeitdauer, Berechnung des Endtermins
- **Rückwärtssterminierung:**  
Vorgabe des Endtermins, Zeitdauer, Berechnung des spätest möglichen Anfangstermins
- **Bestimmung von Produktionszeiten:**  
Vorgabe Anfangs- und Endtermin, Berechnung der zur Verfügung stehenden Produktionszeit

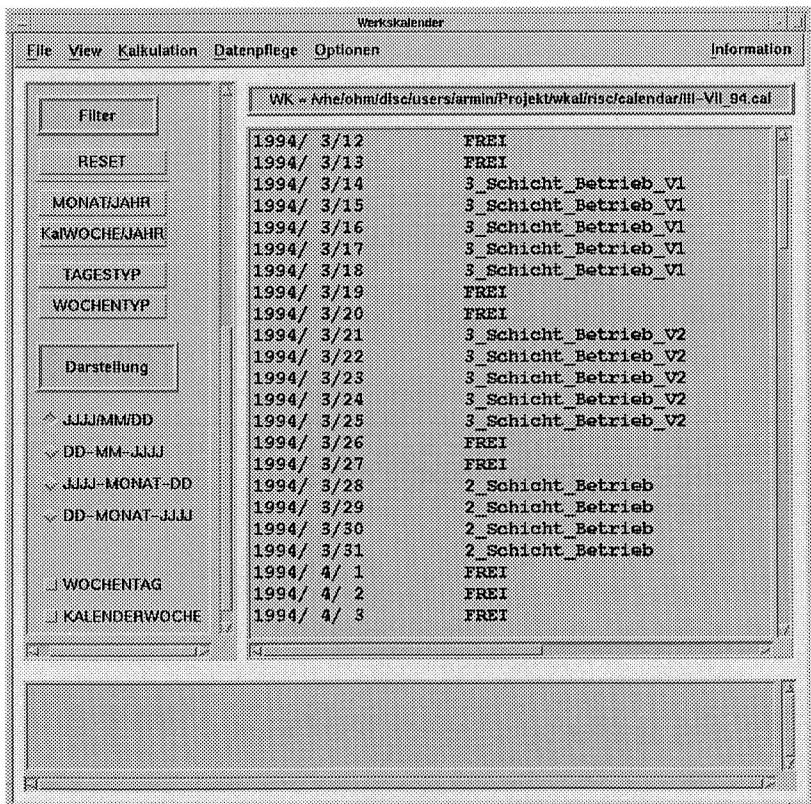


Bild 8-2: Editiermöglichkeiten des Werkskalenders



## 8.2 Elektronische Plantafel zur Auftragsterminierung

Die Visualisierung der Auftragsterminierung erfolgt mit einer elektronischen Plantafel, die die Abbildung, z.B. einer Auftragsliste, auf die Zeitachse des Produktionsunternehmens vornimmt. Grafisch dargestellt werden die zeitlichen Abläufe mit einem Gantt-ähnlichen Diagramm, das einen Überblick über die Auftragsituation der verschiedenen Produktionslinien oder der einzelnen Maschinen vermittelt. Über eine sensitive Oberfläche können aus dem Gantt-ähnlichen Diagramm auch detailliertere Informationen zu den Aufträgen abgerufen werden (z.B. Kunde, Losgröße etc.).

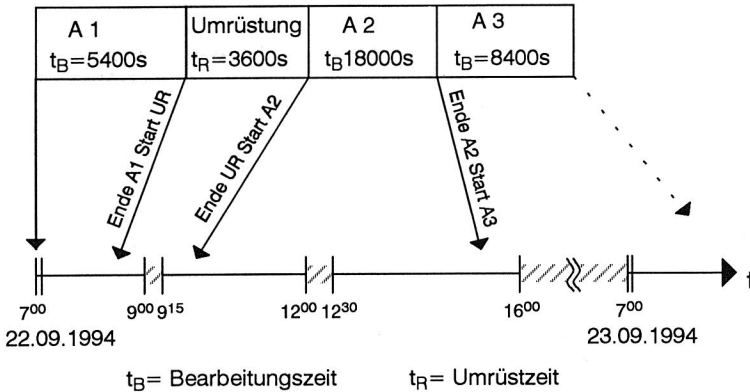


Bild 8-3: Abbildung einer Auftragsliste auf die Zeitachse

### 8.2.1 Feinterminierung von Auftragslisten

Eine Möglichkeit der Terminierung von Auftragslisten ist die Verbindung der Ergebnisse des Optimierungstools und des Werkkalenders. Die Zuteilung der Aufträge auf Bestückungssysteme, die Auftragsreihenfolgen und die erforderlichen Umrüstungen inklusive der berechneten Bearbeitungs- und Rüstzeiten stellen die Eingabeparameter aus der Optimierung dar. Zusammen mit der Vorgabe eines Anfangs- oder Endtermins wird eine detaillierte Auftragseinplanung für den spezifizierten Optimierungslauf durchgeführt. Der Benutzer erhält somit definierte Terminaussagen über die erforderlichen Startzeitpunkte oder über frühest mögliche Fertigstellungstermine.

Eine identische Visualisierung kann durchgeführt werden, wenn die Auftragsliste vom Benutzer erstellt wurde oder von einem anderen Planungssystem übernommen wurde.

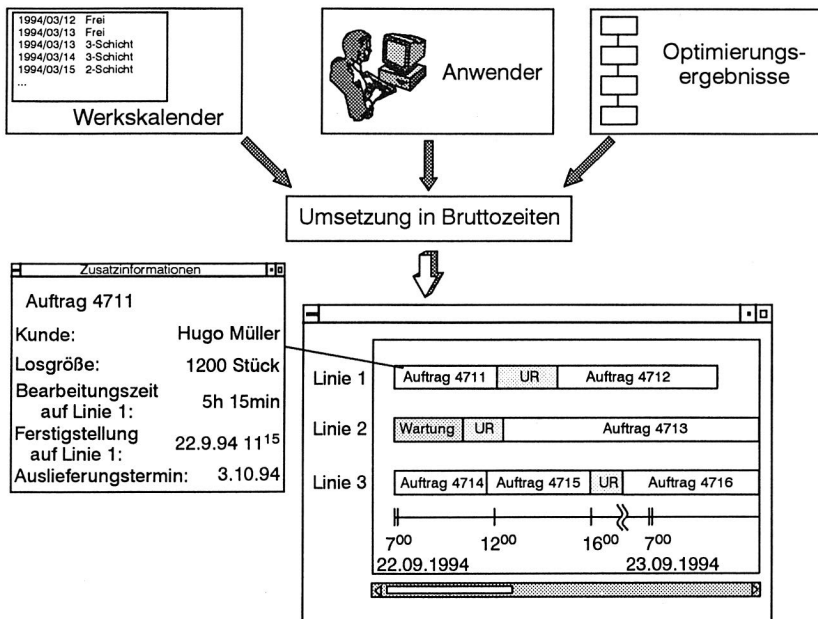


Bild 8-4: Visualisierung einer optimierten Auftragseinplanung

## 8.2.2 Simulation von Umanplanungen

Eine weitere wichtige Funktionalität ist die Simulation von Umanplanungen, die sich durch Auftragsverschiebungen ergibt. Ursache dafür sind nicht vorhersehbare Ereignisse wie Maschinenausfälle, Störungen, kein Material verfügbar etc. oder die Einplanung von Eilaufträgen (s. Bild 8-5). Durch Interaktionen des Benutzers können Aufträge zwischen bzw. innerhalb eines Bestückungssystems verschoben oder ein neuer Auftrag an einer spezifizierten Stelle eingefügt werden. Der neue Sachverhalt wird berechnet und zeitliche Auswirkungen sowie erforderliche Umrüstungen an der Plantafel angezeigt.

Der Benutzer hat hierbei die Wahl, die Reihenfolge vorzugeben oder einen neuen Optimierungslauf zu starten. Gibt der Anwender z.B. einen Eilauftrag an einer bestimmten Stelle ein, wird die Reihenfolge der nachfolgenden Aufträge beibehalten und mit den Zeit- und Rüstmodellen die sich ergebenden Auswirkungen berechnet. Als weitere Möglichkeit kann für den Eilauftrag und die verbleibenden Aufträge in der Auftragsliste ein neuer Optimierungslauf gestartet werden. Um zu gewährleisten, daß der Eilauftrag an die erste Stelle der neuen Auftragsliste platziert wird, erhält er als einziger die Priorität 1. Für den restlichen Auftragspool inklusive der Auswirkungen, die

sich durch den Eilauftrag ergeben, werden eine neu optimierte Reihenfolge und neue Rüststände errechnet.

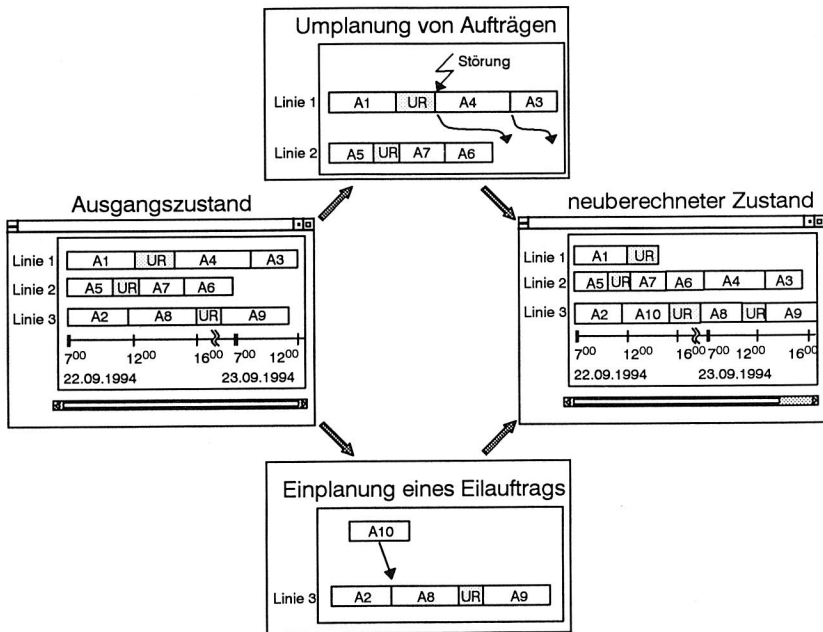


Bild 8-5: Möglichkeiten der Umplanung

### 8.3 Kalkulation der Fertigungskosten

Die Kalkulation der Fertigungskosten (am Beispiel des Bestückens) wurde auf Grundlage des Maschinenstundensatzes durchgeführt. Der Maschinenstundensatz kann sich dabei auf einen einzelnen Automaten (oder Arbeitsplatz) beziehen, es kann aber auch eine linienweise oder bereichsweise (z.B. komplette SMD-Bestückung) Betrachtung durchgeführt werden. Nimmt man als Berechnungsbasis den Bruttomaschinenstundensatz sind auch Löhne und lohnabhängige Kosten für den Maschinenbediener und Restfertigungsgemeinkosten mit eingeschlossen [114]. Beim einfachen oder auch Nettomaschinenstundensatz betrachtet man hingegen nur die reinen Maschinenkosten.

Um die Kostenentstehung besser verfolgen zu können, wird der Bestückvorgang ähnlich der Prozeßkostenrechnung in mehrere Einzelschritte bzw. Prozesse aufgeteilt [63]. Für die erstellten Teilschritte Bestücken, Umrüsten und Nachrüsten der Systeme werden in den Berechnungsmodulen der Optimierung detaillierte Zeiten

ermittelt, so daß sich eine Aufsummierung der Prozeßkosten und eine Kostenkalkulation für den gesamten Bestückvorgang durchführen läßt. Mit Hilfe der Einzelzeiten und der Beachtung der Maschinenspezifika lassen sich somit die Auftragskosten bestimmen. Zudem können auch in Abhängigkeit von Losgröße, Anzahl der Bestückpositionen und Anzahl der Bauelementetypen pro Auftrag Kostensimulationen und Kostenvergleiche zwischen den Bestückungssystemen angestellt und grafisch dargestellt werden.

Die verschiedenen Kostenverläufe bezüglich der Bestückungssysteme ergeben sich durch die unterschiedlichen Bestückleistungen, den unterschiedliche Umrüstaufwendungen pro Förderertyp und dadurch daß Nachrüstungen von Bauelementen ohne oder mit Maschinenstillstand möglich sind. Bild 8-6 zeigt ein Beispiel eines Kostenverlaufs bei unterschiedlichen Losgrößen für einen Auftrag mit 200 Bestückpositionen und 30 verschiedenen Bauelementetypen für drei verschiedenartige Bestückungssysteme:

- **Handbestückplatz**

Umrüstaufwand: 30s pro Bauelementebereitstellung  
Bestückleistung: 600 Bauelemente in der Stunde  
Maschinenstundensatz: 60 DM/Stunde  
keinen Stillstand bei Nachrüstung

- **Pick and Place Automat**

Umrüstaufwand: 60s – 120s in Abhängigkeit des Förderertyps  
Bestückleistung: 3000 Bauelemente in der Stunde  
Maschinenstundensatz: 200 DM/Stunde  
keinen Stillstand bei Nachrüstung

- **Chipshooter**

Umrüstaufwand: 120s – 160s in Abhängigkeit des Förderertyps  
Bestückleistung: 14000 Bauelemente in der Stunde  
Maschinenstundensatz: 350 DM/Stunde  
keine Möglichkeit, zeitgleich zum Bestücken nachzurüsten

Zu erkennen sind in Bild 8-6 die unterschiedlichen Startpunkte der Kurven (fixer Rüstkostenanteil), die sich durch die verschiedenen Rüstaufwendungen ergeben. Mit steigender Stückzahl ist der Chipshooter, bedingt durch seine hohe Bestückleistung die kostengünstigste Alternative. Die kleinen Sprünge in dessen Kostenverlauf stammen von den Zeiten, in der die Maschine zum Nachrüsten von Bauelementen kurz angehalten werden muß. Das vorliegende Beispiel soll rein die theoretischen Zusammenhänge der Einzelkosten erläutern. Technologische Randbedingungen, daß z.B. bestimmte Bauelemente nur auf bestimmten Maschinen bestückt werden können, wurden für dieses Beispiel außer acht gelassen.

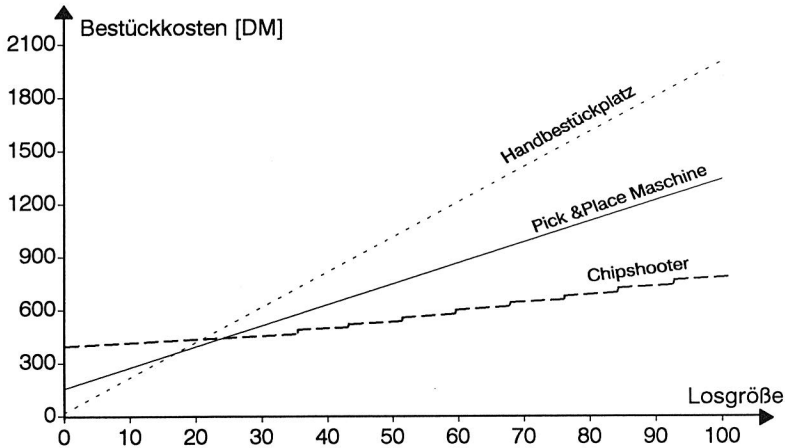


Bild 8-6: Kostenvergleich für verschiedene Bestückungssysteme

#### 8.4 Adaptive Erstellung von Fertigungsunterlagen

Die Vielzahl an Informationen in der zentralen Datenbank kann über verschiedene Mechanismen weiterverwendet werden. Über spezifische Formateinstellungen lassen sich die erforderlichen Informationen in der gewünschten Form ausdrucken. Beispiele hierzu sind Rüstlisten oder Umrüstanweisungen für das Rüstpersonal, die genau spezifizieren, an welchen Zuführspuren der Bestückungssysteme Förderer ab- und aufzurüsten sind. Die Daten können aber auch direkt oder über ein Zwischenformat in andere Softwaresysteme überspielt werden, z.B. in Textverarbeitungssysteme zur Unterstützung der Arbeitsplanung oder in Tabellenkalkulationssysteme zu statistischen Auswertungen (s. Bild 8-7).

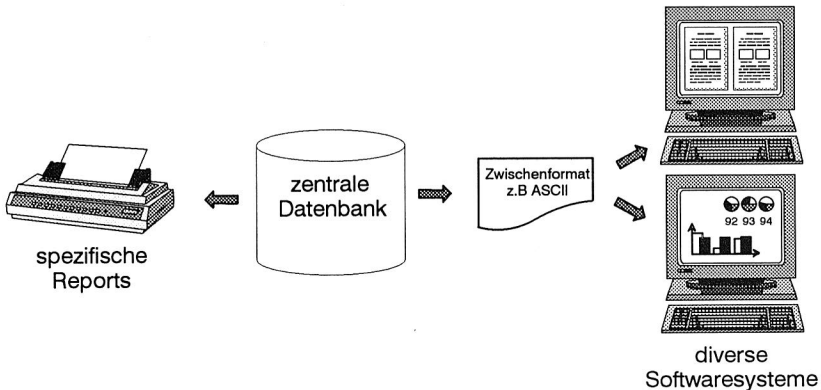


Bild 8-7: Weiterführende Nutzung der Prozeß-, Produkt- und Ergebnisdaten

### 8.5 Aufbau einer BDE/MDE-Kopplung

Durch eine Betriebsdatenerfassung und eine Rückführung der relevanten Daten in die Datenbank des Optimierungssystems (s. Bild 8-8) ist eine automatische Validierung der Plandaten möglich. Zur Aufnahme der Daten werden die Maschinendaten aus dem Steuerungsrechner der Bestückungssysteme mittels eingebauter Interface-karten direkt abgegriffen [29]. Die gefilterten Steuerungsdaten enthalten Informationen über Zustandsmeldungen einzelner Anlagenelemente, Verfügbarkeitskontrollen von Material und Ansteuerungszeitpunkte der einzelnen Achsen sowie der SPS-Systeme z.B. zum Wechsel der Leiterplatten. Mit Hilfe der erfaßten Zeiten hat der Anwender die Möglichkeit, entsprechend seiner Anforderungen (z.B. Festlegen der taktgebenden Maschine) die Bestückungssysteme zu kalibrieren. Alternativ dazu kann vom System automatisch durch eine statistische Auswertung der Bestück- und Rüstzeiten eine Feinjustierung der Zeitmodelle vorgenommen werden. Nach einem Einschwingvorgang nähern sich somit die Planungsdaten den Praxiswerten der Bestückungssysteme an. Durch diese Vorgehensweise erfolgt eine detaillierte, sich stetig verbessernde fertigungsspezifische Adaption des Planungssystems an die unterschiedlichen Fertigungslandschaften und -spezifika.

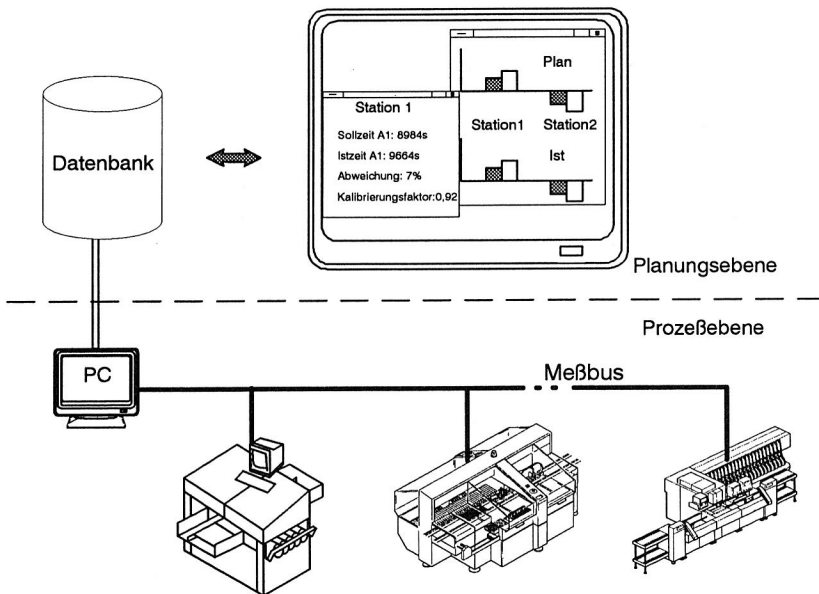


Bild 8-8: Kalibrierung der Planungsdaten mit Hilfe von BDE

## 9 Testergebnisse des realisierten Optimierungssystems DOPE

Zusätzlich zu den durchgeführten Versuchen mit Testleiterplatten am Rechner wurde das Optimierungsmodul, Datenbankgestütztes **OPT**imierungssystem für die **Elektro**nik, auch im Praxiseinsatz getestet. Das folgende Kapitel beschreibt und erörtert die Ergebnisse aus den Testläufen. Aufbauend auf diesen Daten und durch den Vergleich mit gemessenen Praxiswerten lassen sich Einsparungspotentiale bestimmen und einige Wirtschaftlichkeitsbetrachtungen durchführen.

### 9.1 Beschreibung der Testparameter

Zur Erfassung realistischer Parameter für das Zeitmodell sowie zur Überprüfung von Optimierungsergebnissen wurden in einem Unternehmen zur Herstellung von Steuerungsbaugruppen Untersuchungen an einer SMD-Bestückungslinie durchgeführt. Die Linie bestand aus zwei verketteten Bestückungsmodulen SP120 der Firma Siemens (Bestückungsklasse VI).

Die Optimierungsaufgabe bestand darin, für 11 vorgegebene Leiterplattentypen mit unterschiedlichem monatlichen Bedarf eine Festrüstung zu generieren, die eine optimierte Austaktung der Linie und eine optimierte Feederanordnung gewährleistet. Die Menge der Bauelementtypen war hierbei niedriger als die Stellkapazität der Linie, so daß nur Optimierungsschicht 2 (Line Balancing) und Schicht 3/4 (Spur und Sequenzoptimierung) relevant waren. Der Monatsbedarf entspricht im Optimierungslauf der Losgröße und somit der Gewichtung der Leiterplatte für die Austaktung.

In Tabelle 9-1 sind die Leiterplatten beschrieben, deren Daten für den Optimierungslauf eingegeben wurden und für die auszugsweise Zeitmessungen durchgeführt wurden.

Bau- gruppe	Größe [mm]			Einzel- schaltungen	Bestückposi- tionen	Losgröße
	Länge	Breite	Dicke			
T1	390,00	351,00	2,00	1	1116	47
T2	390,00	351,00	2,00	1	807	80
T3	148,00	144,00	1,60	1	211	180
T4	148,00	144,00	1,60	1	189	55
T5	233,40	220,00	1,60	1	451	150
T6	180,00	146,00	1,60	1	215	340
T7	234,50	220,00	1,60	1	390	268
T8	233,50	220,00	1,60	1	323	135
T9	233,43	220,01	1,60	1	464	150
T10	234,00	160,00	1,60	6	330	59
T11	234,00	160,00	1,60	6	342	79

Tabelle 9-1: Leiterplattendaten zu den durchgeführten Testläufen

## 9.2 Validierung der Optimierungs- und Zeitmodelle mittels Praxisergebnissen

Zur Validierung der Zeit- und Optimierungsmodelle wurden Bestückzeiten verschiedener Baugruppen und die Verfahrzeiten einzelner Achsen gemessen. Ein Ziel der Untersuchungen war, das Zeitmodell für die einzelnen Arbeitsschritte anhand von praxisnahen Messungen zu überprüfen und gegebenenfalls die entsprechenden Parameter zu korrigieren. Von besonderem Interesse waren dabei die Zeiten für die Zyklen und die Bearbeitungszeiten der kompletten Leiterplatte sowie pro Bauelement.

Eine weitere Zielsetzung war es, mit den durchgeführten Tests die berechneten Ergebnisse mit dem Zustand vor der Optimierung zu vergleichen. So können Optimierungspotentiale für die Austaktung und der Bestückzeitminimierung ermittelt werden.

Für die Bestimmung der Bestückzeiten wurden bei einer vorliegenden Rüstung in der laufenden Produktion einige Messungen für die verschiedenen komplexen Leiterplatten durchgeführt. Die Baugruppen, die während des Versuchszeitraums gefertigt wurden und die ermittelten Einzelzeiten zeigt Tabelle 9-2. Diese Messungen dienen als Grundlage für die Validierung der Zeitmodelle.

Baugruppe	Anzahl Bestückpositionen			Visionzeit je LP [s]	Bestückzeit [s] ohne LP-Vision		mittl. Bestückzeit je Bauelement [s]			Abweichung vom Mittelwert	
	ges.	St. 1	St. 2		St. 1	St. 2	St. 1	St. 2	Mittel	[s]	[%]
T1	1116	558	558	7	346,1	388,0	0,6203	0,6953	0,6578	0,0375	5,70
T2	807	404	403	6,5	253,9	272,0	0,6285	0,6749	0,6517	0,0232	3,56
T6	215	107	107	7	70,5	76,7	0,6589	0,7168	0,6879	0,0290	4,22
T7	390	195	195	6,5	116,8	128,1	0,5990	0,6569	0,6280	0,0290	4,62
T11	342	222	120	14,5	144,0	70,0	0,6486	0,5833	0,6120	0,0287	4,69
Durchschnittliche Bestückzeit je Bauelement [s]							0,6249	0,6590	0,6420		

Tabelle 9-2: Gemessene Bestückzeiten einiger Testleiterplatten

Für den durchgeführten Optimierungslauf wurden die benötigten Leiterplattendaten (s. Kap. 9.1), die in dem Siemens-spezifischen Maschinenformaten zur Verfügung standen, mit Hilfe des Importmoduls in die Datenbank des Planungssystems eingelesen. In umgekehrter Reihenfolge setzte das Exportmodul nach dem Optimierungslauf die optimierte Rüstung aus der Datenbank wieder in das Maschinenformat um, so daß die Daten wieder am Linienrechner eingespielt werden konnten. Die Ergebnisse des Optimierungslaufes (Austaktung und Bestückzeiten) zeigt Tabelle 9-3.



Bau- gruppe	Aufteilung Bestückpositionen			berechnete Bestückzeiten[s]		Abweichung vom Optimum (bez. auf Leiterplatte)		Abweichung vom Optimum (inkl. Losgröße) [%]
	ges.	St. 1	St. 2	St.1	St. 2	sec.	%	
T1	1116	554	562	350,8	354,0	1,6	0,5	0,08
T2	807	404	403	252,3	249,8	1,25	0,5	0,1
T3	211	105	106	67,2	67,8	0,3	0,4	0,05
T4	189	94	95	59,9	60,3	0,2	0,3	0,01
T5	451	225	226	144,0	144,6	0,3	0,2	0,04
T6	215	109	105	65,9	65,2	0,35	0,5	0,13
T7	390	195	195	122,2	120,9	0,65	0,5	0,19
T8	323	161	162	101,4	102,0	0,3	0,3	0,04
T9	464	230	228	146,0	144,8	0,6	0,5	0,09
T10	330	162	168	102,0	105,8	1,9	1,8	0,11
T11	342	166	176	104,6	110,9	3,15	3	0,26

Tabelle 9-3: Optimierungsergebnisse zu den Testleiterplatten

Bedingt durch das bestimmbare Optimum für die Austaktung der Linie, lassen sich für den Balancinglauf Ergebnisbewertungen vornehmen. In Tabelle 9-3 sind diesbezüglich die errechneten Abweichungen vom theoretischen Optimum dargestellt. Betrachtet man die Leiterplatten einzeln, ergaben sich berechnete Bestückzeitunterschiede zwischen den beiden Stationen von 0,2 bis 3 Sekunden, was einer Abweichung von 0,2 bis 3% entspricht. Das aussagekräftigere Resultat ist die Abweichung vom Optimum mit Einbeziehung der Losgrößen, da nicht leiterplattenbezogen sondern chargenbezogen optimiert wurde. Das sehr gute Ergebnis zeigt lediglich Schwankungen im Bereich von 0,01 bis 0,26%.

Da man das theoretische Optimum für die Bestückzeit pro Leiterplatte nicht bestimmen und dies auch nicht abschätzen kann, ist eine Aussage über die prozentuale Abweichung der Feeder- und Sequenzoptimierung vom theoretischen Optimum unmöglich. Die Ergebnisse dieses Optimierungsvorgangs lassen sich demzufolge nur bewerten, wenn man Vergleiche zwischen Ausgangslage und optimierten Zustand zieht oder die Resultate verschiedener Optimierungsläufe untereinander vergleicht.

Zur Validierung wurden für die optimierte Rüstung Messungen direkt an der Linie vorgenommen. Unter Rücksichtnahme auf die laufende Fertigung erfolgten die Zeitmessungen exemplarisch für die Baugruppe mit dem größten Monatsbedarf. Die für den direkten Vergleich zwischen ursprünglicher und optimierter Rüstung relevanten Zeiten für T6 sind in Tabelle 9-4 dargestellt.

Es zeigt sich, daß aufgrund der veränderten Rüstkonfiguration deutliche Verbesserungen eintraten. Die Gesamtzeit für die Baugruppe T6 konnte mit der neuen Rüstung von 147,2 s um 11,2 s auf 136 s verbessert werden. Dies bedeutet, daß allein mit der Spur- und Sequenzoptimierung eine Minimierung der Bestückzeiten um 7,6% erreicht wurde. Selbst bei gleicher Anzahl von 107 Bestückpositionen je Station (in

der Ausgangslage und für die optimierte Rüstung) ist auch eine Verbesserung der Austaktung erkennbar. Mit der alten Rüstung lag die Differenz zwischen Station 1 und 2 bei 6,2 s, bei der neuen Rüstung liegt sie bei 2 s, was einer Reduzierung von 4,2% auf 1,4% für die Abweichung vom Optimum entspricht.

Bau- gruppe	Rüstung	Anzahl Bestückpositio- nen (Aufteilung lt. Nut- zentest)			Bestückzeit [s]			
					incl. LP-Vision		ohne LP-Vision	
		gesamt	St. 1	St. 2	St. 1	St. 2	St. 1	St. 2
T6	ALT	214	107	107	77,5	83,7	70,5	76,7
	OPTIMIERT	214	107	107	76,0	74,0	69,0	67,0

*Tabelle 9-4: Vergleich gemessener Bestückzeiten zwischen alter und optimierter Rüstung*

Vergleicht man zusätzlich die gemessenen Bestückzeiten der Baugruppe T6 aus Tabelle 9-4 mit den berechneten Werten aus Tabelle 9-3, die mit den Zeitmodellen simuliert wurden, ergeben sich Abweichungen von 3,1 s und 1,8 s pro Leiterplatte und Station. Mittelt man diese Werte entspricht dies einer Abweichung von 3,2 %.

Zur weiteren Überprüfung der Zeitmodelle wurde ein Optimierungslauf mit einer Festrüstung durchgeführt und die ermittelten Zeiten mit denen in der Praxis gemessenen verglichen (s. Tabelle 9-2). Diese Festrüstung ist hierbei identisch mit der Rüstung, die bei den Bestückzeitmessungen in Tabelle 9-2 zugrunde lag.

Baugruppe	BP je Auftrag	Gesamtbestückzeit[s]	Bestückzeit je Leiterpl. [s]	errechnete Bestückzeit je BE [s]	gemessene Bestückzeit je BE [s]	Abweichung [s]	Abweichung [%]
T1	52452	33219,30	706,79	0,633	0,657	0,025	3,8
T2	64560	40318,07	503,97	0,624	0,651	0,027	4,1
T6	73100	48319,1	142,1	0,661	0,687	0,026	3,8
T7	104520	62837,9	234,47	0,601	0,628	0,027	4,2
T11	27018	16037,67	203,0	0,593	0,612	0,019	3,1

*Tabelle 9-5: Berechnete Bestückzeiten durch vorgegebene Festrüstung*

Im Vergleich liegen die berechneten Zeiten alle etwas unter den gemessenen, was auf die idealisierten, theoretischen Grundlagen der Zeitmodelle zurückzuführen ist. Eine Analyse der Ergebnisse zeigt jedoch, daß unabhängig von der Komplexität der Leiterplatten und Anzahl der Bestückpositionen eine konstante Abweichung von 3 bis 4% gemessen wurde. Dieser Wert deckt sich auch mit dem berechneten und gemessenen Wert für die Baugruppe T6 des Optimierungslaufs. Damit wurde verifiziert, daß die zugrunde gelegten Zeitmodelle den Bestückvorgang an den Stationen nahezu ideal nachbilden. Die stationsbezogenen Kalibrierungsfaktoren ermöglichen die Kompensation der durchschnittlichen Abweichungen und somit eine ideale Anpassung an die maschinenspezifischen Eigenschaften der einzelnen Stationen.

### 9.3 *Diskussion der Optimierungsläufe*

In diesem Kapitel wird das Verhalten der Optimierungsläufe unter verschiedenen Randbedingungen behandelt. Um Aufschlüsse über die Verhältnisse Laufzeit zu erzielten Ergebnisse zu erhalten, wurden mit verschiedenen Parametereinstellungen Testläufe vorgenommen. Als Testobjekte dienten die in Kapitel 9.2 beschriebenen Leiterplatten. Innerhalb der Optimierungsläufe variierten die Einstellungen maximale Laufzeit, Optimierungsgüte, der Auftragsumfang und die Linienkonfigurationen. Für den Balancing-Prozeß standen zwei verschiedene Optimierungsstrategien zur Verfügung. Ein Beispiel der durchgeführten Tests (alle 11 Leiterplatten, 2 Module in der Linie) zeigt Bild 9-1.

Um eine Aussage über die erreichte Güte der Optimierung zu treffen, wurde für den Balancinglauf (nur für diese Schicht ist es möglich) ein theoretisches Optimum berechnet. Hierzu wurden sämtliche Bestückpositionen der 11 Leiterplattentypen mit den jeweiligen Losgrößen multipliziert und die Gesamtzeit auf die zwei Module der Linie aufgeteilt. Die somit erhaltene, minimal erreichbare Gesamtzeit der Charge wurde in den Optimierungsläufen mit der aktuellen Gesamtbestückzeit verglichen.

Am Verhalten der Bestückzeit-Kurven für den Optimierungsvorgang der Schicht 2 kann man die wesentliche Eigenschaften z.B. einer Heuristik und Evolution ablesen. Generell zeigten alle Optimierungsläufe den in Bild 9-1 repräsentativ dargestellten Verlauf. Die Evolution wählt die vorzunehmende Mutationsoperation (Verschieben, Vertauschen) nach dem Zufallsprinzip aus. Ebenfalls zufällig entscheidet sie dann, an welchem Bauteil diese Operation durchzuführen ist. Die jeweilige Entscheidung zur Rüstung auf der anderen Station wird durch eine Neuberechnung der Bestückzeiten gefällt. Tritt durch die Operation eine Verbesserung der Gesamtbestückzeit ein, so verlegt die Evolution die entsprechenden Bauelemente, falls nicht, startet sie einen neuen Schleifendurchlauf. Dies kann dazu führen, daß die Bestückzeiten über eine längere Zeit hinweg unverändert bleiben, da für die gewählte Mutation keine passenden Bauelemente gefunden werden. Dieses Charakteristikum der Evolution ist im Bild 9-1 an den längeren geraden Teilstücken zu erkennen, bei denen sich keine Verbesserung der Gesamtbestückzeit ergab.

Die Heuristik hingegen läßt ein zielstrebigeres Verhalten erkennen, so daß sie durchschnittlich in kürzerer Laufzeit bessere Ergebnisse erreicht. In Bild Bild 9-1 ist das deterministische Vorgehen der Heuristik gut an dem nahezu linearen Abfall der Kurve, zumindest am Anfang, zu erkennen. Bei längeren Laufzeiten erbrachte die Evolution in 60 Prozent der Testläufe bessere Resultate als die Heuristik.

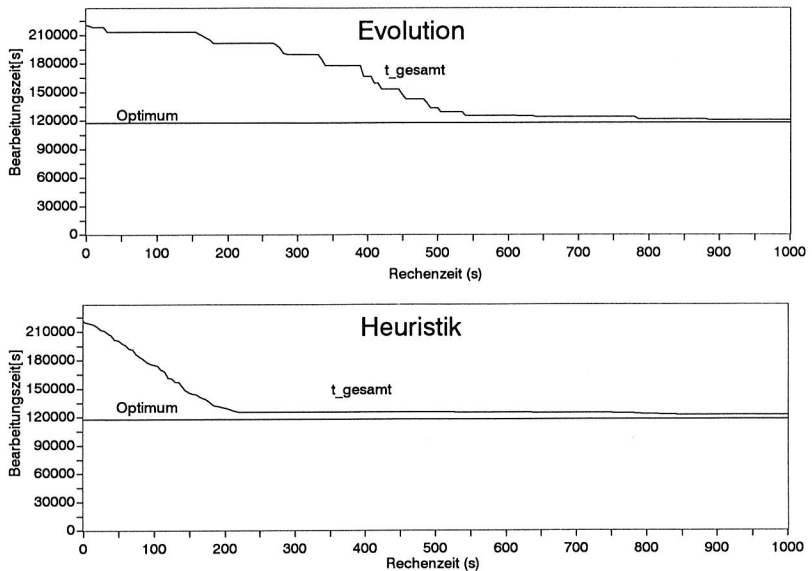


Bild 9-1: Beispiele zu den Auswertungen der Testläufe

Als Ergebnis aus diesen Untersuchungen läßt sich ableiten, daß bei zeitkritischen Planungen die Heuristik vorzuziehen ist. Bereits nach 220 Sekunden wurde für das Testbeispiel ein Ergebnis erreicht, das um 6,5% vom theoretisch erreichbarem Optimum (117000 s) abweicht. Dieses Resultat wurde von der Evolution erst nach 540 Sekunden ausgegeben. Ab einer Rechenzeit größer 620 Sekunden war die Evolution der Heuristik überlegen. Nach 900 Sekunden ergaben beide Methoden keine weitere Ergebnisverbesserung. Zu diesem Zeitpunkt war die Evolution 2,2% die Heuristik 3,4% vom theoretisch erreichbaren Optimum entfernt. Die Rechenzeit zur Optimierung der Rüstung und der Bestücksequenzen für die 11 Leiterplatten betrug 320 Sekunden. Als eine weitere wesentliche Erkenntnis kann man aus den Testläufen erkennen, daß für ein Optimierungsproblem dieser Größenordnung (11 Leiterplat-tentypen, 32 Bauelementetypen, einer Gesamtanzahl von 4838 Bestückpositionen und einer Linie mit zwei Modulen) eine Rechenzeit von ca. 600 Sekunden durchaus ausreicht, um ein sehr gutes Ergebnis zu erzielen.

#### 9.4 Wirtschaftlichkeitsbetrachtungen und Einsparungspotentiale

Zur Untersuchung der Wirtschaftlichkeit und der erreichbaren Rationalisierungspotentiale wurden bei verschiedenen Unternehmen (sowohl bei Massenfertigern als auch Auftragsfertigern) eine Kostenverfolgung mehrerer SMD-Leiterplatten durchgeführt. Das Verhältnis Material- zu Fertigungskosten variierte in Abhängigkeit von der Komplexität der Leiterplatte, der verwendeten Bauelemente, dem Bestückungs-

system und der Organisationsstruktur des Unternehmens. Der Anteil der reduzierbaren Fertigungskosten lag dabei zwischen 20 und 40%. Im Bild 9-2 sind exemplarisch die Aufteilung der Fertigungszeiten und die erzielbaren Kosteneinsparungen für zwei Fälle dargestellt. Beispiel 1 zeigt die Situation bei einem Massenfertiger, Beispiel 2 die eines Auftragsfertigers.

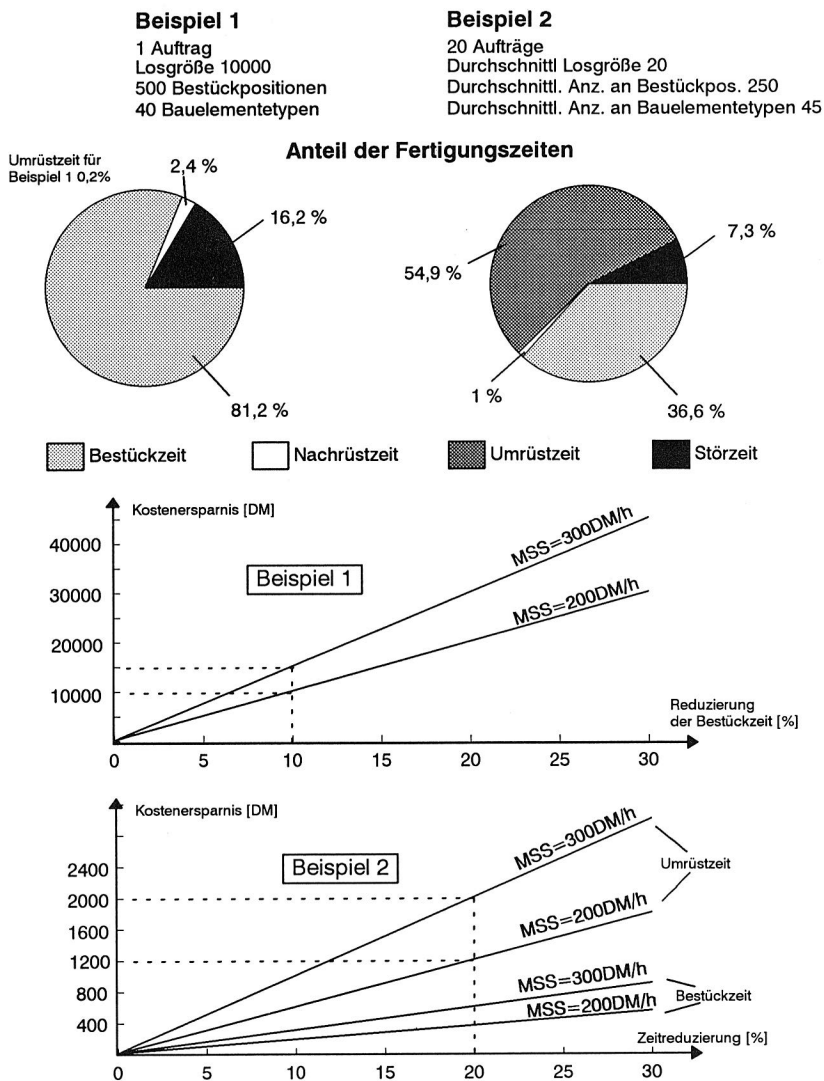


Bild 9-2: Verteilung der Fertigungszeiten und zu erreichende Kostenersparnisse

Als allgemeine Berechnungsgrundlage gelten folgende Randbedingungen:

- Die Bestückleistung der betrachteten Fertigungslinie liegt bei 10000 BE/h.
- Die Umrüstung pro Bauelementtyp beträgt zwei Minuten.

Die Ergebnisse der beiden ausgewählten Beispiele bezüglich der Kosteneinsparung sind sehr unterschiedlich. Bei Beispiel 1 ist nur die Einsparung der Prozeßzeit relevant, da die Umrüstzeit bei großen Aufträgen nicht ins Gewicht fällt. Geht man hierbei von einer 10-prozentigen Einsparung der Bestückzeit aus, z.B durch optimale Aus-taktung und Feederoptimierung, reduzieren sich die Fertigungskosten bei einem Maschinenstundensatz von 300.- DM bereits um 15.000.- DM.

Im 2. Beispiel ist für die Reduzierung der Gesamtzeit und somit der Fertigungskosten die Umrüstzeit entscheidender. Bei einer Verringerung der Umrüstzeit um 20% kann der Fertigungskostenanteil um 2000.- DM gesenkt werden. Eine Verringerung der Bestückzeiten aller Leiterplatten um 20% ergibt in diesem Fall eine Kostensenkung von 600.- DM. Zum Vergleich sind in dieser Grafik die Einsparungspotentiale für eine Senkung der Bestück- und der Umrüstzeit dargestellt.

Generell sind die erzielbaren Reduzierungen, die durch die einzelnen Optimierungsmethoden erreicht werden können, schwer zu quantifizieren. Sie sind stark abhängig vom vorliegendem Ausgangszustand, der Menge der zu optimierenden Aufträge, den Leiterplattentypen und von den vorgegebenen Restriktionen. Da für gleiche Problemstellungen Tests mit unterschiedlichen Optimierungsstrategien durchgeführt wurden, lassen sich durch relative Vergleiche der Ergebnisse durchschnittliche Einsparungspotentiale bestimmen. Die Resultate hieraus decken sich auch weitgehend mit Ergebniswerten aus Praxisberichten [19, 84, 40].

So ergaben sich bei Auftragsmengen größer zehn eine Reduzierung der Umrüstzeit von durchschnittlich 20 %. Die mittlere Einsparung für den Bereich Line-Balancing liegt bei zwei oder drei Modulen in der Linie und drei Aufträgen in der Charge bei 10 %. Für die maschinenorientierten Optimierungsmethoden (Feederkonfiguration und Sequenzoptimierung) schwanken die Resultate in Abhängigkeit des Maschinentyps und der Leiterplattenspezifika zwischen 5 und 15 %. In diesem Bereich liegt auch das Testergebnis aus dem Praxisversuch (s. Kap. 9.2). Außer den geschätzten Einsparungsmöglichkeiten gibt es noch eine Reihe nicht quantifizierbarer Rationalisierungspotentiale des Optimierungssystems wie:

- Erleichterung der Dateneingabe durch grafisch orientierte Editoren
- Einheitliche Dateneingabe für alle Maschinentypen
- Ausführliche Informations- und Ergebnisdarstellung
- Online Datenübertragung an die Maschinen
- Vermeidung von Eingabefehlern durch automatisierte Programmerzeugung etc.

Die gemachten Untersuchungen zeigten, daß durch die vorgestellten Optimierungsmethoden die Durchlaufzeiten in der Flachbaugruppenbestückung im 2-stelligen Prozentbereich reduziert werden. Auch ohne die Einbeziehung der nicht quantifizierbaren Vorteile, ergibt sich für das Optimierungssystem ein beachtliches Rationalisierungspotential zur Senkung der Fertigungskosten.

## **10 Zusammenfassung**

Die zunehmende Integration elektronischer Komponenten und die Substitution mechanischer durch elektronische Aggregate in vielen Produkten sowie die Übernahme größerer Funktionalität innerhalb der Produkte durch die Elektronik, zeigen den großen und weiterhin wachsenden Einfluß der elektronischen Systeme auch auf andere Produktbereiche. Dies sind nur einige Gründe, die der Elektronik als Schlüsseltechnologie eine immer größer werdende Bedeutung zukommen lassen.

Das Kernstück dieser Technologie, die Fertigung der Flachbaugruppen ist ein äußerst innovativer Fertigungsbereich, der ständig von Neuentwicklungen und Veränderungen beeinflusst wird. Neue Montagetechnologien (COB, Flip Chip etc.), neuartige Schaltungsträger (3D-MID) oder neue Bauelementeformen (BGA etc.) sind nur einige Beispiele hierzu.

Die Beherrschung der Prozesse und ein sinnvolles Einbinden neuartiger Technologien sind Anforderungen, denen ein wettbewerbsfähiges Unternehmen gewachsen sein muß. Die Vielzahl und Komplexität der Daten erfordern für die Firmen eine effiziente Planung, um kostengünstig zu produzieren. Durch die hohe Rechnerdurchdringung im Entwicklungsbereich der Leiterplatten liegen die meisten produktbeschreibenden Daten schon im Rechner vor. Dieser Vorteil ist auszunutzen, um von der Entwicklung über die Planungsebene bis zur Prozeßebene einen durchgängigen Datenfluß zu realisieren.

Ziel dieser Arbeit war die Entwicklung eines Softwaresystems zur Unterstützung der Planungsaktivitäten in der Flachbaugruppenfertigung. Hierzu wurden im Vorfeld am Markt verfügbare Systeme, die Datenformate in der Elektronikfertigung und die Ansprüche an ein Planungssystem von verschiedenen Unternehmen analysiert. Ausgehend von den Defiziten wurde ein Anforderungsprofil und darauf aufbauend ein flexibles, erweiterbares Konzept erarbeitet.

Das realisierte modulare Planungssystem besitzt als Kern eine Datenbank, in der sämtliche Informationen über die Produkte, Prozesse und die anforderungsspezifischen Fertigungsstrukturen abgelegt sind. Die eigenständigen Module beziehen ihre Informationen aus der Datenbank, legen dort die Ergebnisse ab und können über diese kommunizieren. Schwerpunktmäßig wurde in dieser Arbeit bei der Entwicklung der Module, insbesondere des Optimierungsmoduls der Bestückprozeß betrachtet.

Das Design-for-Manufacturing-System zur Analyse und Bewertung von Leiterplattendesigns, das Import-/Exportmodul und die Komponente zur Auftragseinplanung sind prozeßübergreifend anwendbar. Der entwickelte Prototyp des DFM-Systems besitzt eine allgemeingültige Basisstruktur, die mit fachspezifischem Wissen gefüllt wird. Anpaßbar an die jeweiligen Fertigungsspezifika kann der Anwender neue Regeln zu allen Prozeßschritten der Flachbaugruppenfertigung eingeben. Mit dem im

Laufe der Zeit abgelegten Know-how wird eine Layoutdiagnose hinsichtlich bestück-, löt-, test-, und montagegerechtem Design durchgeführt. Die eventuell erforderliche Modifikation des Layouts hinsichtlich der Design-Rules führt zu einer problemloseren Fertigung der Flachbaugruppe und somit zu einer Reduzierung der Fertigungskosten.

Das komplexeste Tool ist das entwickelte Optimierungsmodul. Unabhängig von der Bestücktechnologie und den Bestückungsmethoden können mit diesem Werkzeug vom Anwender spezifizierte Optimierungsvorgänge durchgeführt werden. Bedingt durch die Komplexität des gesamten Optimierungsproblems, ergab sich ein hierarchisches Modell. Dieses ist in den maschinennahen Optimierungsebenen nochmals unterteilt nach den verschiedenen, in dieser Arbeit festgelegten Bestückungsklassen. Es wurden hierzu eine Menge diverser Optimierungsalgorithmen implementiert und die Anwendungsmöglichkeiten auf die vorliegenden Problemstellungen untersucht. Das Ziel dieser Komponente, die Reduzierung von Umrüstzeiten und die Minimierung der Bestückungszeiten für den kompletten Bestückungsbereich, wurde in einer ganzen Reihe von Testläufen und auch in Praxiseinsätzen validiert. So ergibt sich für das entwickelte Planungssystem durch die erzielbare Zeiteinsparung und Senkung der Fertigungskosten ein beachtliches Rationalisierungspotential. Durch zusätzliche Möglichkeiten in der Datenaufbereitung, des Datentransfers und der Informationsdarstellung kommen einige weitere nicht meßbare Arbeitserleichterungen hinzu.

Der Ausbau zu einem kompletten Leitstandsystem wurde mit dem Modul der Auftragseinplanung realisiert. Die entwickelten Funktionen ermöglichen eine detaillierte Einplanung, Verfolgung und Terminierung der Aufträge sowie die Simulation von Umplanungen. Die Informationen bezieht diese Komponente über die gemeinsame Datenbank, aus den eingegebenen Ergebnissen des Optimierungstools oder aus dem Input anderer Datenquellen wie z.B. der BDE-Kopplung. Weitere Funktionalitäten des Moduls Auftragseinplanung ist die Kostenkalkulation und Kostensimulation für unterschiedliche Bestückungssysteme oder bei veränderten Parametern. Zudem kann sich der Benutzer selbst spezifizierte Fertigungsdokumente ausgeben lassen.



## **Literatur**

1. Aarts, E.;  
Korst, J.: Simulated Annealing and Boltzman machines  
John Wiley and Sons, Chichester 1989
2. Abeln, O.: Die CA-Techniken in der industriellen Praxis: Handbuch  
der computerunterstützten Ingenieurmethoden  
Carl Hanser Verlag, München 1990
3. Ablay, P.: Optimieren mit Evolutionsstrategien  
In: Spektrum der Wissenschaft 7 (1987) S. 104-115
4. AWF/REFA: Handbuch der Arbeitsvorbereitung Teil1: Arbeitspla-  
nung  
Beuth Verlag, Berlin
5. Balzert, H.: Einführung in die Software-Ergonomie  
Walter de Gruyter Verlag, Berlin 1988
6. Barker, J. A.;  
Handerson, D.: What is "liquid"? Understanding the states of matter  
In: Reviews of Modern Physics 48 (1976) S. 587-671
7. Bao, H.P.: An Expert System for SMT Printed Circuit Board  
Assembly  
In: Manufacturing Review 1 (1988) S. 275-280
8. Becker, B.: Künstliche Intelligenz: Konzepte, Systeme,  
Verheißungen  
Campus Verlag, Frankfurt/Main 1992
9. Bergman, D.  
(Ed.): Surface Mount Council - Status of the Technology  
Industry Activities and Action Plan  
Institute for Interconnecting and Packaging Electronic  
Circuits, Lincolnwood 1994
10. Berkmann, T.: Von CAD zu CAM  
In: Elektronik Praxis 12 (1993) S. 108-110
11. Berlage, Z.: OSF/Motif und das X-Window System  
Addison-Wesley Verlag, Bonn, München 1991
12. Booch, G.: Object oriented design  
Redwood City, California 1991
13. Cala, M.;  
Srihari, K.: Knowledge Based Process Planning in Surface Mount  
Technology  
Proceedings of the IEEE/CHMT IEMT Symposium 1991,  
New York
14. Cerny, V.: Thermodynamical approach to the traveling salesman  
problem: an efficient simulation algorithm  
In: Journal of Optimization Theory and Applications 45  
(1985) S.41-45

15. Chen, P. P.: Entity-Relationship Approach: The Use of ER Concept in Knowledge Representation  
Fourth Int. Conference on Entity-Relationship Approach, Chicago 1985, IEEE Computer Society Press, Washington 1985
16. Corbett, J.: Design for Manufacture: Strategies, principles and techniques  
Addison-Wesley Verlag, Wokingham 1991
17. Däumler, K.-D.; Grabe, J.: Kostenrechnungsllexikon, ABC der Kostenrechnung  
Verlag Neue Wirtschafts-Briefe, Herne Berlin 1992
18. Davis, L.: Adapting Operator Probabilities in Genetic Algorithms  
Proceedings 3rd. Int. Conference of Genetic Algorithms, San Mateo 1989
19. Deutschländer, A.: Integrierender Ansatz zur Reduzierung der Durchlaufzeit in der Leiterplattenmontage  
In: Zwf CIM 86 (1991) S. 160-165
20. Dräger: SMD Gehäusetrends und deren Konsequenz für die Verarbeitungstechnologie  
Dokumentation Tutorial XIII, SMT/ASIC/Hybrid '92, Messe & Kongreß GmbH, Stuttgart
21. Erman, L.D.: The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty  
In: Computational Surveys 12 (1980)
22. Eversheim, W.: Organisation in der Produktionstechnik. Band 1: Grundlagen  
VDI-Verlag GmbH, Düsseldorf 1981
23. Eversheim, W.; Schneewind, J.: Integrierte Arbeitsplanung und Fertigungsfeinsteuerung  
In: Zwf CIM 87 (1992) 7 S. 411-414
24. Feldmann, K.: Wettbewerbsfähig durch Integration neuer Technologien und optimierte Prozeßketten  
Fachtagung "Flexible Produktionssysteme für innovative Elektronik", Productronica 1993
25. Feldmann, K.; Franke, J.: MID-Technologie und Marktentwicklung in Europa  
Tagungsband zum 1. internationalen MID-Kongreß, Erlangen 1994
26. Feldmann, K.; Roth, N.: Optimization of Set-up Strategies for Operating Automated SMT Assembly Lines.  
Annals of the CIRP, Vol. 40/1/1991
27. Feldmann, K.; Rothhaupt, A.: An Intelligent Optimization Tool for the Electronics Production  
Proceedings of the Surface Mount International, 1994 S. 704-710

28. Feldmann, K.; Grampp, K.; Koch, M.; Rothhaupt, A.: Optimale Rüst- und Umrüststrategien steigern die Produktivität  
In: Leiterplattentechnik (Supplement zu F&M, MO, QZ) Mai 1992, Carl Hanser Verlag, LP46-LP50
29. Feldmann, K.; Zöllner, B.: Verfügbarkeit durch Diagnose  
In: Produktionsautomatisierung pa 2 (1993), Oldenbourg-Verlag, München 1993 S. 29-32
30. Franke, J.: Standardisierung der CAD/CAM-Verfahrensketten für die Bestückung von Leiterplatten (IPC-D-355)  
Vortragsband zur FED-Konferenz 1993, Hrsg. Fachverband Elektronik Design, Berlin 1993
31. Franke, J.; Feldmann, K.: CAD/CAM Verfahrensketten in der Elektronikproduktion  
Tagungsband zur AME, Essen 1991
32. Fritzke, B.; Wilke, P.: FLEXMAP- A neural network with linear time and space complexity for the traveling salesman problem  
Proceedings of IJCNN-91, Singapore (1991) S. 929-934
33. Frühwald, C.: Analyse und Planung produktionstechnischer Rüstabläufe  
Dissertation Hannover, VDI-Verlag GmbH, Düsseldorf 1990
34. Görz, G.: Einführung in die künstliche Intelligenz  
Addison-Wesley Verlag, 1993
35. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning  
Addison-Wesley Verlag, 1989
36. Goldberg, D.: Sizing Population for Serial and Parallel Genetic Algorithms  
Proceedings of the Third International Conference on Genetic Algorithms, San Mateo 1989
37. Grosch, G.; Ziegler, V.: Taschenbuch der Mathematik, Ergänzende Kapitel zu Bronstein, Semendjajew  
Teubner Verlagsgesellschaft, 1979
38. Günther, H.-O.; Gronalt, M.; Piller, F.: Computergestützte Simulation eines hochautomatisierten Montagesystems  
In: Management & Computer 2 (1994) S. 25-32
39. Herrmann, G.; Egerer, K.: Handbuch der Leiterplattentechnik (Band 2)  
Eugen G. Leutze Verlag, Saulgau/Würthenberg 1991
40. Hemmeter, G.: Rüstoptimierung für variantenreiche SMD-Fertigung  
In: productronic 3 (1993) S. 53-55
41. Holden, H.: The challenge: to find a framework and vision to launch DFM/A  
Proceedings of the Surface Mount International, 1994 S. 713-718

42. Holden, H.: DFM in PWB Fab: A Review of predictive Engineering benefits  
Proceedings of the Surface Mount International, 1993 S. 877-881
43. Holland, J. H.: Adaption in natural and artificial systems  
The MIT Press, Cambridge, Massachusetts 1992
44. Hume, H.; Komm, R.; Garrison, T.: Design Report Card: A Method for measuring design for manufacturability  
Proceedings of the Surface Mount International, San Jose 1992 S. 986-991
45. Hummel, S.; Männel, W.: Kostenrechnung I, Grundlagen, Aufbau und Anwendung  
Betriebswirtschaftlicher Verlag Dr. Th. Gabler, Wiesbaden 1986
46. Irizarry-Gaskins, V.; Chang, T.-C.: Knowledge-Based Process Planning for Electronic Assembly  
In: Journal of Intelligent and Robotics Systems 3 (1990) S. 3-16
47. ISO 10303 STEP (Standard of the Exchange of Product data)
48. Kilbridge, M.; Wester, L.: The Balancing Delay Problem  
In: Management Science 8 (1961) S. 69-84
49. Kirkpatrick, S.; Gelatt, J. D.; Vecchi, M. P.: Optimization by simulated annealing  
In: Science 220 (1983) S. 671-680
50. Kirkpatrick, S.: Optimization by simulated annealing: quantitative studies  
In: Journal of Statistical Physics 34 (1984) S. 975-986
51. Kohonen, T.: Self-organized formation of topologically correct feature maps  
In: Biological Cybernetics 43 (1982) S. 59-69
52. Kohonen, T.: Self-organization and associative memory  
Springer Verlag, Heidelberg 1984
53. Kreutzfeld, J.; Schmidt, B.: Integrierte Arbeitsplanung und Fertigungssteuerung  
In: CIM Management 3 (1992) S. 53-60
54. Krups, R.: SMT-Handbuch  
Vogel Verlag, Würzburg 1991
55. V. Laarhoven, P.; Aarts, E. H. L.: Simulated Annealing  
Reidel, 1988
56. Lawler, E. L.: The Traveling salesman problem  
Wiley Verlag, 1990
57. Lea, C.: A Scientific Guide to Surface Mount Technology  
Electrochemical Publications Limited, Ayr, Scotland 1988

- 
58. Liao, J.-S.; Young, R.: A Process Planning and Concurrent Engineering System for PCBs.  
In: Manufacturing Review 6 (1993) S. 25-39.
59. Lin, S.: Computer Solutions for the Traveling salesman problem  
In: Bell System Technical Journal 44 (1965) S. 2245-2269
60. Lin, S.; Kernighan, B.W.: An effective heuristic algorithm for the traveling salesman problem  
In: Operations Research 11 (1973)
61. Maria, A.; Srihari, K.: A Review of Knowledge-based Systems in Printed Circuit Board Assembly  
In: The International Journal of Advanced Manufacturing Technology 7 (1992) S.368-377
62. Martin, J.: Principles of object-oriented analysis and design  
PTR Prentice-Hall Inc. Englewood, New Jersey 1993
63. Mayer, R.: Prozeßkostenrechnung und Prozeßkostenmanagement  
In: (Hrsg.) Horvath & Partner: Prozeßkostenmanagement, Vahlen Verlag, München 1991
64. Ming, C. L.; Zhiming, J.: Computer Aided Process Planning for Printed Circuit Board Assembly  
Proceedings of the national electronic packaging and production conference, Les Plaines 1991
65. Miller, G. S.; Colton, J. S.: The Complementary Roles of Expert Systems and Database Management Systems in a Design for Manufacturing Environment  
In: Engineering with Computers 8 (1992) S. 139-149
66. Miska, F. M.: CIM  
Verlag moderne Industrie 1988
67. Molloy, E.; Browne, J.: A knowledge-based approach to design for manufacturing using features  
In: (Editor) Parsai, H.; Sullivan, G.: Concurrent Engineering, Verlag Chapman and Hall, London 1993
68. N.N.: Entity-Relationship-Modell: Das Objekt der Datenmodellierer  
In: Datenbankfokus 2 (1992) S.35-38
69. N.N.: Die offene CAM-Welt von heute  
In: CADs 8 (1992)
70. N.N.: Produktionsautomatisierung in der Baugruppenfertigung  
In: EPP 4 (1994) S. 22-24
71. N.N.: Ingres/SQL Reference Manual  
Ingres Corporation, 1991

72. N.N.: PROKAPPA Reference Manual Volume I and II  
IntelliCorp Inc. 1991
73. N.N.: Einheit mit vier Ebenen  
In: Productronic 5 (1991) S. 101-102
74. EDIF Technical Committee: EDIF-Electronic Design Interchange Format  
Version 3.00  
Electronic Industries Association, 1993
75. Neumann, K.; Morlock, M.: Operations Research  
Carl Hanser Verlag, München 1993
76. Ntuen, C. A.: An expert system for integrated circuit board inspection  
In: Computers in Industry 19 (1992) S. 281-289
77. Manderik, B.; De Weger, M.; Spiessens, P.: The Genetic Algorithm and the Structure of the Fitness Landscape  
Proceedings 4th. Int. Conference of Genetic Algorithms,  
University of California, San Diego 1991
78. Metropolis, N.: Equatation of state calculations by fast computing machines  
In: Journal of Chemical Physics 21 (1953) S. 1087-1092
79. Papageorgiou, M.: Optimierung  
Oldenbourg Verlag, München 1991
80. Parker, G. R.; Rardin, L. R.: Discrete Optimization  
Academic Press Inc. 1988
81. Pham, D.T.; Taccin, E.: Techiques for Intelligent Computer-Aided Design  
In: (Editor) Pham, D.T.: Artificial Intelligence in Design,  
Springer Verlag, London 1991
82. Plapp, C.: Detaillierte Kapazitätsplanung und Reihenfolgeoptimierung unterstützen die Fertigungssteuerung  
In: Zwf CIM 86 (191) 11 S. 542-545
83. Pöhls, A.: Rüstooptimierung in der SMT-Baugruppenbestückung  
In: Feinwerttechnik & Meßtechnik 98 (1990) S. 313-315
84. Puppe, F.: Einführung in Expertensysteme  
Springer Verlag, 1991
85. Rechenberg, I.: Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologischen Evolution  
Friedrich Fromann Verlag, Stuttgart 1973
86. Reed, K.A.; Kelly, J. C.; Harrod, D.; Conroy, W.: The Initial Graphics Exchange Specification (IGES)  
Version 5.1,  
National Computer Graphics Association, Fairfax, VA, 1991
87. Reitinger, M.: SMT-Montage in der Flachbaugruppenfertigung - heute und morgen  
Tagungsband Leiterplatte 1994, VDI-Verlag GmbH, Düsseldorf 1994

88. REFA: Methodenlehre des Arbeitsstudiums, Teil 2  
Datenerfassung  
München 1971
89. Rothhaupt, A.: Durchlaufzeitverkürzung in der SMD-Bestückung durch  
Rüst- und Umrüstoptimierung.  
Tagungsband zur SMT, ASIC, Hybrid, Nürnberg Juni  
1991
90. Saleveston, M.E.: The Assembly Line Balancing-Problem  
In: Journal of Industrial Engineering 6 (1962) S. 167-192
91. Sauer, W;  
Weigert, G: Erhöhung der Auslastung flexibler Bestückungssysteme  
durch Simulation und Optimierung.  
Tagungsband zur AME, Essen 1991
92. Schaffer, J.D.: A study of control parameters affecting online perfor-  
mance of Genetic Algorithms for function optimization.  
Proceedings 3rd. Int. Conference of Genetic Algorithms,  
San Mateo 1989
93. Schöneburg, E.;  
Heinzmann, F.;  
Fedderson, S.: Genetische Algorithmen und Evolutionsstrategien  
Addison-Wesley Verlag, Bonn 1994
94. Schindler, H.;  
Müller, G.: Rüstop Optimierung bei der SMD-Bestückung  
Tagungsband zur AME, Essen 1991
95. Scholz-Reiter, B.;  
von Issendor, C.: CAD-Schnittstellen in der Praxis  
In: CIM Management 2 (1994) S. 23-30
96. Schraft, R. D.;  
Wolf, E.;  
Leicht, T. R.: Bestückungsautomaten  
Hüthig Verlag, Heidelberg 1989
97. Sedgewick, R.: Algorithmen  
Addison-Wesley Verlag, Bonn 1992
98. Sisir, K. P;  
Dwivedi, S. N.: An object-orientated knowledge-based system for  
manufacturability of printed wiring boards  
In: Computers in Industry 18 (1992) S. 77-89
99. Smith, W. A.;  
Sutton, J. C.: A Multiple Perspective Printed Circuit Board Design  
Guide: Expert System Prototyp  
In: Association for Computing Machinery (1990)  
S. 760-765
100. Spur, G.;  
Mertins, K. J.: Planung und Steuerung von Fertigungszellen für die  
Montage elektronischer Baugruppen  
In: ZwF 84 (1989) 4, S. 169-175
101. Starkweather, T.: A Comparison of Genetic Sequencing Operators  
Proceedings 4th. Int. Conference of Genetic Algorithms,  
University of California, San Diego 1991
102. Staubach, G.: UNIX-Werkzeuge zur Textmusterverarbeitung  
Springer Verlag, Berlin 1989

103. Stoyan, H.: Programmiermethoden der künstlichen Intelligenz, Band 2 Springer Verlag, Berlin 1991
104. Syswerda, H.: Schedule Optimization Using Genetic Algorithms  
In: Davis, L. (Hrsg.) Handbook of Genetic Algorithms, 1991
105. Terno, J.;  
Lindemann, R.;  
Scheithauer, G.: Zuschnittprobleme und ihre praktischen Lösungen  
Harry Deutsch Verlag, Frankfurt 1987
106. Thim, C.: Rechnerunterstützte Optimierung von Materialflußstrukturen in der Elektronikmontage durch Simulation  
Hrsg.: K. Feldmann. Diss. Univ. Erlangen-Nürnberg, Carl Hanser Verlag, München 1992
107. Trautloft, R.: Datenbanken: Entwurf und Anwendung  
Verlag Technik GmbH, Berlin 1991
108. Tönshoff, H. K.;  
Menzel, E.;  
Park, H. S.: A Knowledge-Based System for Automated Assembly Planning  
Annals of the CIRP, Vol. 41/1/1992
109. Tönshoff, H. K.;  
Hamelmann, S.;  
Schmidt, M.T.: Arbeitspläne für Variantenteile wissensbasiert erstellen  
In: Zwf CIM 7 (1992) S. 406-410
110. Verguld, M. M.;  
Klein, R. J.: Hochleistungs-SMD-Bestückung Voraussetzungen und Grenzen,  
In: Feinwerk- & Meßtechnik 3 (1992) S. 77ff
111. Wagner, R.: Ist das Standard-Gerber-Datenformat für die Leiterplattenfertigung noch aktuell?  
In: CADs & SMT 5 (1993) S. 75-77
112. Warnecke, H.J.;  
Kühnle, H.: Integration of CAPP and PPC-The PPC Part  
Annals of the CIRP Vol. 41/1/1992
113. Warnecke, H.J.;  
Bullinger, H.-J.;  
Hichert, R.: Kostenrechnung für Ingenieure (3. Auflage)  
Carl Hanser Verlag, München 1990
114. Wang, Y.;  
Hsieh, L.-H.;  
Seliger, G.: Knowledge-Based Integration of Design and Assembly Process Planning  
In: Manufacturing Systems, Vol. 22 (1993) No.2
115. Waguih, H.: Integration Assembly Planning and Scheduling - CAPP Related Issues  
Annals of the CIRP Vol. 41/1/1992
116. Walther, H.;  
Nägler, G.: Graphen Algorithmen-Programme  
Springer Verlag, Wien, New York 1987
117. Whitley, D.: Scheduling Problems and Traveling Salesman - The Genetic Edge Recombination Operator  
Proceedings of the third international Conference on Genetic Algorithms, San Mateo 1989



- 
118. Wiendahl, H.-P.: Betriebsorganisation für Ingenieure  
Carl Hanser Verlag, München 1983
  119. Wiendahl, H.-P.: Rüstzeitverkürzung-ein Beitrag zur flexiblen Produktion  
AWF Fachtagung "Rüstzeitverkürzung". Bad Soden/Ts.  
20.-21.9. 1988
  120. Wiendahl, H.-P.: Simulationsmodelle in Produktionsplanung und  
-steuerung  
ZwF CIM 85 (1990) 3 S. 137 ff.
  121. Willy, A.;  
Schmaus, T.: Umrüststillstand wird kürzer  
In: Maschinenmarkt 97 (1991) S. 74-77
  122. Witte, H.-H.;  
Ali, E. : Werkzeuge und Werkzeugsysteme der Metallbearbei-  
tung  
VDI-Z 134 (1992) Nr. 6 S. 68-75
  123. Zhang, H.-C.: IPPM-A Prototyp to Integrate Process Planning and Job  
Shop Scheduling Functions  
Annals of the CIRP Vol. 42/1/1993

## Lebenslauf

Armin Rothhaupt

geboren am 22.12.1962 in Nürnberg

verheiratet, 2 Kinder

- |               |  |
|---------------|--|
| 1969 - 1973   | Grundschule in Nürnberg  |
| 1973 - 1982   | Hans-Sachs Gymnasium in Nürnberg,<br>math.-naturwiss. Zweig  |
| 1982 - 1983   | Grundwehrdienst  |
| 1983 - 1989   | Studium der Fertigungstechnik<br>an der Universität Erlangen-Nürnberg<br>Abschluß 06/1989: Dipl.-Ing.  |
| bis Ende 1989 | Mitarbeiter bei BiiN Informationssysteme<br>im Technischen Support   |
| 1990-1995     | Wissenschaftlicher Mitarbeiter am Lehrstuhl für<br>Fertigungsautomatisierung und Produktionssystematik<br>an der Universität Erlangen-Nürnberg.<br>Leiter: Prof.- Dr.-Ing. K. Feldmann |

# Reihe Fertigungstechnik Erlangen

## Band 1

Andreas Hemberger

### **Innovationspotentiale in der rechnerintegrierten Produktion durch wissensbasierte Systeme**

208 Seiten, 107 Bilder. 1988. Kartonierte.

## Band 2

Detlef Classe

### **Beitrag zur Steigerung der Flexibilität automatisierter Montagesysteme durch Sensorintegration und erweiterte Steuerungskonzepte**

194 Seiten, 70 Bilder. 1988. Kartonierte.

## Band 3

Friedrich-Wilhelm Nolting

### **Projektiertung von Montagesystemen**

201 Seiten, 107 Bilder, 1 Tabelle. 1989.

Kartonierte.

## Band 4

Karsten Schlüter

### **Nutzungsgradsteigerung von Montagesystemen durch den Einsatz der Simulationstechnik**

177 Seiten, 97 Bilder. 1989. Kartonierte.

## Band 5

Shir-Kuan Lin

### **Aufbau von Modellen zur Lageregelung von Industrierobotern**

168 Seiten, 46 Bilder. 1989. Kartonierte.

## Band 6

Rudolf Nuss

### **Untersuchungen zur Bearbeitungsqualität im Fertigungssystem Laserstrahlschneiden**

206 Seiten, 115 Bilder, 6 Tabellen. 1989. Kartonierte.

## Band 7

Wolfgang Scholz

### **Modell zur datenbankgestützten Planung automatisierter Montageanlagen**

194 Seiten, 89 Bilder. 1989. Kartonierte.

## Band 8

Hans-Jürgen Wißmeier

### **Beitrag zur Beurteilung des Bruchverhaltens von Hartmetall-Fließpreßmatrizen**

179 Seiten, 99 Bilder, 9 Tabellen. 1989. Kartonierte.

## Band 9

Rainer Eisele

### **Konzeption und Wirtschaftlichkeit von Planungssystemen in der Produktion**

183 Seiten, 86 Bilder. 1990. Kartonierte.

Band 10  
Rolf Pfeiffer  
**Technologisch orientierte Montageplanung am Beispiel der Schraubtechnik**  
216 Seiten, 102 Bilder, 16 Tabellen. 1990. Kartoniert.

Band 11  
Herbert Fischer  
**Verteilte Planungssysteme zur Flexibilitätssteigerung der rechnerintegrierten Teilefertigung**  
201 Seiten, 82 Bilder. 1990. Kartoniert.

Band 12  
Gerhard Kleineidam  
**CAD/CAP : Rechnergestützte Montagefeinplanung**  
203 Seiten, 107 Bilder. 1990. Kartoniert.

Band 13  
Frank Vollertsen  
**Pulvermetallurgische Verarbeitung eines übereutektoiden verschleißfesten Stahls**  
XIII + 217 Seiten, 67 Bilder, 34 Tabellen. 1990. Kartoniert.

Band 14  
Stephan Biermann  
**Untersuchungen zur Anlagen- und Prozeßdiagnostik für das Schneiden mit CO<sub>2</sub>- Hochleistungslasern**  
VIII + 170 Seiten, 93 Bilder, 4 Tabellen. 1991. Kartoniert.

Band 15  
Uwe Geißler  
**Material- und Datenfluß in einer flexiblen Blechbearbeitungszelle**  
124 Seiten, 41 Bilder, 7 Tabellen. 1991. Kartoniert.

Band 16  
Frank Oswald Hake  
**Entwicklung eines rechnergestützten Diagnosesystems für automatisierte Montagezellen**  
XIV + 166 Seiten, 77 Bilder. 1991. Kartoniert.

Band 17  
Herbert Reichel  
**Optimierung der Werkzeugbereitstellung durch rechnergestützte Arbeitsfolgenbestimmung**  
196 Seiten, 73 Bilder, 2 Tabellen. 1991. Kartoniert.

Band 18  
Josef Scheller  
**Modellierung und Einsatz von Softwaresystemen für rechnergeführte Montagezellen**  
196 Seiten, 65 Bilder. 1991. Kartoniert.

Band 19  
Arnold vom Ende  
**Untersuchungen zum Biegeumformen mit elastischer Matrize**  
166 Seiten, 55 Bilder, 13 Tabellen. 1991. Kartoniert.

Band 20  
Joachim Schmid  
**Beitrag zum automatisierten Bearbeiten von Keramikguß mit Industrierobotern**  
XIV + 176 Seiten, 111 Bilder, 6 Tabellen. 1991. Kartoniert.

Band 21

Egon Sommer

**Multiprozessorsteuerung für kooperierende  
Industrieroboter in Montagezellen**

188 Seiten, 102 Bilder. 1991. Kartonierte.

Band 22

Georg Geyer

**Entwicklung problemspezifischer Verfahrensketten  
in der Montage**

192 Seiten, 112 Bilder. 1991. Kartonierte.

Band 23

Rainer Flohr

**Beitrag zur optimalen Verbindungstechnik in der  
Oberflächenmontage (SMT)**

186 Seiten, 79 Bilder. 1991. Kartonierte.

Band 24

Alfons Rief

**Untersuchungen zur Verfahrensfolge Laserstrahlschneiden  
und -schweißen in der Rohkarosserieherstellung**

VI + 145 Seiten, 58 Bilder, 5 Tabellen. 1991. Kartonierte.

Band 25

Christoph Thim

**Rechnerunterstützte Optimierung von Materialflußstrukturen  
in der Elektronikmontage durch Simulation**

188 Seiten, 74 Bilder. 1992. Kartonierte.

Band 26

Roland Müller

**CO<sub>2</sub> – Laserstrahlschneiden von kurzglasverstärkten Verbundwerkstoffen**

141 Seiten, 107 Bilder, 4 Tabellen. 1992. Kartonierte.

Band 27

Günther Schäfer

**Integrierte Informationsverarbeitung bei der Montageplanung**

195 Seiten, 76 Bilder. 1992. Kartonierte.

Band 28

Martin Hoffmann

**Entwicklung einer CAD/CAM-Prozesskette für die Herstellung  
von Blechbiegeteilen**

149 Seiten, 89 Bilder. 1992. Kartonierte.

Band 29

Peter Hoffmann

**Verfahrensfolge Laserstrahlschneiden und -schweißen :  
Prozessführung und Systemtechnik in der 3D-Laserstrahlbearbeitung von Blech-  
formteilen**

186 Seiten, 92 Bilder, 10 Tabellen. 1992. Kartonierte.

Band 30

Olaf Schrödel

**Flexible Werkstattsteuerung mit objektorientierten Softwarestrukturen**

180 Seiten, 84 Bilder. 1992. Kartonierte.

Band 31

Hubert Reinisch

**Planungs- und Steuerungswerkzeuge zur impliziten  
Geräteprogrammierung in Roboterzellen**

XI + 212 Seiten, 112 Bilder. 1992. Kartonierte.

Band 32  
Brigitte Bärnreuther  
**Ein Beitrag zur Bewertung des Kommunikationsverhaltens  
von Automatisierungsgeräten in flexiblen Produktionszellen**  
XI + 179 Seiten, 71 Bilder. 1992. Kartoniert.

Band 33  
Joachim Hutfless  
**Laserstrahlregelung und Optikdiagnostik in der Strahlführung  
einer CO<sub>2</sub>-Hochleistungslaseranlage**  
175 Seiten, 70 Bilder, 17 Tabellen. 1993. Kartoniert.

Band 34  
Uwe Günzel  
**Entwicklung und Einsatz eines Simulationsverfahrens für operative  
und strategische Probleme der Produktionsplanung und -steuerung**  
XIV + 170 Seiten, 66 Bilder, 5 Tabellen. 1993. Kartoniert.

Band 35  
Bertram Ehmann  
**Operatives Fertigungscontrolling durch Optimierung auftragsbezogener  
Bearbeitungsabläufe in der Elektronikfertigung**  
XV + 167 Seiten, 114 Bilder. 1993. Kartoniert.

Band 36  
Harald Kolléra  
**Entwicklung eines benutzerorientierten Werkstattprogrammersystems  
für das Laserstrahlschneiden**  
129 Seiten, 66 Bilder, 1 Tabelle. 1993. Kartoniert.

Band 37  
Stephanie Abels  
**Modellierung und Optimierung von Montageanlagen  
in einem integrierten Simulationssystem**  
188 Seiten, 88 Bilder. 1993. Kartoniert.

Band 38  
Robert Schmidt-Hebbel  
**Laserstrahlbohren durchflußbestimmender  
Durchgangslöcher**  
145 Seiten, 63 Bilder, 11 Tabellen. 1993. Kartoniert.

Band 39  
Norbert Lutz  
**Oberflächenfeinbearbeitung keramischer Werkstoffe mit  
XeCl-Excimerlaserstrahlung**  
187 Seiten, 98 Bilder, 29 Tabellen. 1994. Kartoniert.

Band 40  
Konrad Grampp  
**Rechnerunterstützung bei Test und Schulung an  
Steuerungssoftware von SMD-Bestücklinien**  
178 Seiten, 88 Bilder. 1995. Kartoniert.

Band 41  
Martin Koch  
**Wissensbasierte Unterstützung der Angebotsbearbeitung  
in der Investitionsgüterindustrie**  
169 Seiten, 68 Bilder. 1995. Kartoniert.

Band 42  
Armin Gropp  
**Anlagen- und Prozeßdiagnostik beim Schneiden mit einem  
gepulsten Nd:YAG-Laser**  
160 Seiten, 88 Bilder, 7 Tabellen. 1995. Kartoniert.

Band 43

Werner Heckel

**Optische 3D-Konturerfassung und on-line Biegewinkelmessung  
mit dem Lichtschnittverfahren**

149 Seiten, 43 Bilder, 11 Tabellen. 1995. Kartoniert.

Band 44

Armin Rothhaupt

**Modulares Planungssystem zur Optimierung  
der Elektronikfertigung**

180 Seiten, 101 Bilder. 1995. Kartoniert.