LEHRBRIEFE FÜR DAS HOCHSCHULFERNSTUDIUM

HERAUSGEGEBEN
VON DER ZENTRALSTELLE FÜR DAS HOCHSCHULFERNSTUDIUM
DES MINISTERIUMS FÜR HOCH- UND FACHSCHULWESEN

Mathematische Methoden und Modelle

8. LEHRBRIEF

Anwendung des PS OPSI

02 4602 08 0

Mathematische Methoden und Modelle

8. Lehrbrief

Anwendung des PS OPSI

Verfaßt von

Dozent Dr. sc. Wolfgang L a s s m a n n
Martin-Luther-Universität Halle

Dipl.-Math. Heinz W e r n e r
VEB Robotron - Zentrum für Forschung und Technik, Dresden

Dipl.-Math. Peter S c h l o m s
VEB Robotron - Zentrum für Forschung und Technik, Dresden

Beilage (Seiten 1 bis 10)

02 4602 08 0

Das druckfertige Manuskript wurde an der Martin-Luther-Universität Halle hergestellt.

Bestell-Nr. 02 4602 08 0

Verfaßt für die Zentralstelle für das Hochschulfernstudium des Ministeriums für Hoch- und Fachschulwesen Dresden.

Herausgegeben im Auftrag des Ministeriums für Hoch- und Fachschulwesen der Deutschen Demokratischen Republik von der Zentralstelle für das Hochschulfernstudium Dresden.

Inhaltsverzeichnis

0. Vorbemerkung

Die lineare Optimierung hat in der betrieblichen Leitung und Planung nach wie vor eine große Bedeutung. Ihre Vorzüge sind:

1. Eine große Universalität und Flexibilität in der Anwendung.
2. Eine bemerkenswerte Unkompliziertheit wegen der einfachen Modellstruktur infolge linearer Beziehungen.
3. Eine effektive Berechenbarkeit umfangreicher Aufgaben mit Hilfe vorgefertigter Systemunterlagen.
4. Vielseitige Möglichkeiten der quantitativen Analyse und Verwertung der Ergebnisse.

Das "einheitliche System der elektronischen Rechentechnik" (ESER), an deren Entwicklung die sozialistischen Länder gemeinsam beteiligt sind, ist die gegenwärtige gerätetechnische Basis der elektronischen Datenverarbeitung in der DDR. Zur Lösung von vielfältigen Aufgabenstellungen der linearen Optimierung wird vom Kombinat Robotron das leistungsfähige Programmiersystem Optimierung mittels Simplextechnik (PS OPSI) bereitgestellt.

Nachdem im Abschnitt 1 auf die allgemeine Bedeutung der linearen Optimierung in der Ökonomie eingegangen wird, erfolgt im Abschnitt 2 eine Einführung in das PS OPSI. Sie ist beschränkt auf ausgewählte Teilgebiete, die für den Erstanwender unbedingt erforderlich sind. Dabei wird auf die Technologie der Nutzung des PS OPSI weitestgehend eingegangen.

Während im Abschnitt 3 ein durchgehendes Beispiel in einer einfachen und einer komplexen Anwendungsform mathematisch modelliert wird, demonstriert der Abschnitt 4 dessen rechentechnische Realisierung mit dem PS OPSI.

1. Ökonomische Problemstellung der linearen Optimierung

1.1. Bedeutung der linearen Optimierung in der Ökonomie

Die Lösung komplizierter Aufgaben der Volkswirtschaft, die optimale Gestaltung des Reproduktionsprozesses und die Rationalisierung der Produktion erfordern eine ständige Weiterentwicklung und Verbesserung der Leitung und Planung.

Die Objektivierung des Planungsprozesses setzt jedoch die Objektivierung des Entscheidungsprozesses voraus. Um die ständig anwachsende Informationsmenge zu bewältigen, die fast unüberschaubar gewordenen Verflechtungen innerhalb und außerhalb eines Wirtschaftsbereiches im wesentlichen zu berücksichtigen und auf die Dynamik der wirtschaftenden Einheiten entsprechend schnell zu reagieren, muß man das Ziel verfolgen, die wiederkehrenden Aufgaben mathematisch zu formulieren, ihre möglichen Lösungswege zu algorithmieren und die so entstandenen Entscheidungsmodelle rechentechnisch zu realisieren.

Dabei versteht man unter der mathematischen Entscheidungsvorbereitung eine Methode, mit der für eine gegebene Problemstellung aus der Menge der möglichen Lösungsvarianten entweder überhaupt eine zulässige Lösung oder eine verbesserte oder sogar die beste Lösung exakt oder näherungsweise berechnet werden kann. Die für das Auffinden der bestmöglichen Lösung speziell entwickelten mathematischen Verfahren werden unter der Bezeichnung "Optimierung" zusammengefaßt. Dementsprechend wurde von KREKO der Begriff der Optimierung als ein Verfahren zur Realisierung eines bestimmten Zieles bei rationellster Ausnutzung der gegebenen Möglichkeiten definiert.

Aus der Notwendigkeit heraus, eine optimale Gestaltung des Reproduktionsprozesses zu erreichen, schuf KANTOROWITSCH im Jahre 1939 mit seinem Werk "Mathematische Methoden der

Organisation und Planung der Produktion" wesentliche Grundlagen für eine neue mathematische Theorie - die lineare Optimierung. Das heute bekannteste Lösungsverfahren der linearen Optimierung, die Simplexmethode, wurde von DANTZIG 1947 entwickelt. Danach erfolgte eine rasche mathematische Entwicklung der Theorie und Anwendung der Optimierung. Unterstützt wurde diese Entwicklung durch die Verfügbarkeit der neuen elektronischen Rechentechnik.

In der DDR wird die Optimierung in den Wirtschaftswissenschaften seit etwa 20 Jahren angewendet. Bei ihrer Anwendung dominieren die Methoden der linearen Optimierung. Zur Zeit sind ca. 300 bedeutungsvolle Optimierungsmodelle bekannt, die für die ständige Entscheidungsvorbereitung in der Leitung und Planung genutzt werden.

Die lineare Optimierung führt zu beträchtlichen Effekten bei der Verbesserung des Planungsprozesses. Es können mit ihrer Hilfe optimale Lösungen einer Vielzahl volkswirtschaftlicher Problemstellungen erzielt werden. Zu den wichtigsten Anwendungsgebieten aus betriebswirtschaftlicher Sicht zählen Forschung und Entwicklung, Investitionsplanung, Materialplanung, Mischungsberechnungen, Arbeitskräfteeinsatzplanung, Materialzuschnittsplanung, Produktionsaufteilung sowie Absatz- und Exportplanung.

1.2. Mathematische Modellierung von linearen Optimierungsaufgaben

Aufgabe der linearen Optimierung ist die Maximierung (Minimierung) einer von n Variablen x_j , $j=1(1)n$, abhängigen linearen Funktion $Z(x_1,x_2,\ldots,x_n)$, wobei die Variablen x_j noch gewisse Nebenbedingungen, die in Form von linearen Ungleichungen und/oder Gleichungen gegeben sind, erfüllen müssen. Das mathematische Modell lautet:

$$Z(x_1,x_2,\dots,x_n) = c_1x_1+\dots+c_nx_n \longrightarrow \text{Max! (Min!)} \tag{1.1}$$

$$\begin{array}{l} a_{11}x_1+\dots+a_{1n}x_n \;\{\leq, \geq \text{ oder } =\}\; b_1 \\ a_{21}x_1+\dots+a_{2n}x_n \;\{\leq, \geq \text{ oder } =\}\; b_2 \\ \dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1+\dots+a_{mn}x_n \;\{\leq, \geq \text{ oder } =\}\; b_m \end{array} \tag{1.2}$$

$$x_1 \geq 0,\dots,x_n \geq 0 \;. \tag{1.3}$$

c_j , b_i und a_{ij} , $i=1(1)m$, $j=1(1)n$, sind gegebene reelle Zahlen. Die Funktion $Z(x_1,\dots,x_n)$ wird als Zielfunktion bezeichnet. (1.2) sind die Nebenbedingungen, (1.3) die Nichtnegativitätsbedingungen und beide Gruppen zusammen das Restriktionssystem. Eine allgemeinere Form ist im Abschnitt 2.1 dargestellt.

Eine Lösung $x_1,\dots,x_n$, die alle Restriktionen erfüllt, wird als zulässige Lösung und die zulässige Lösung, für die der Zielfunktionswert maximal (minimal) ist, als optimale Lösung bezeichnet.

Die mathematische Aufgabenstellung soll nun ökonomisch mit einem einfachen Modell zur Bestimmung eines optimalen Produktionsprogrammes untersetzt werden. Dieses ist Grundlage für die zu demonstrierende OPSI-Anwendung in diesem Lehrbrief.

Ein Betrieb stellt n verschiedene Produkte P_j , $j=1(1)n$, her. Unter dem Begriff Produkt sollen Enderzeugnisse, Baugruppen wie auch Einzelteile verstanden werden. Zur Herstellung dieser n Produkte kommen m Aufwandsarten A_i , $i=1(1)m$, zum Einsatz. Um eine Mengeneinheit des Produktes P_j herzustellen, sind von der Aufwandsart A_i a_{ij} Mengeneinheiten nötig. Für einen bestimmten Zeitraum T (Woche, Monat, Jahr) stehen dem Betrieb von der Aufwandsart A_i nur b_i Mengeneinheiten zur Verfügung. Der Gewinn, den der Betrieb für jede

Mengeneinheit von P_j erzielt, ist c_j .
Das Ziel des Betriebes besteht darin, bei gegebenen Werten c_j , b_i und a_{ij} die Produkte P_j im Zeitraum T in solchen Mengen zu produzieren, daß er einen maximalen Gewinn erzielt.

Sind x_j , $x_j \geqq 0$, die im Zeitraum T zu produzierenden Mengen der Produkte P_j , so lautet die Zielfunktion:

$$Z = c_1x_1 + \dots + c_nx_n \longrightarrow \text{Max!}$$

Die Größen x_j sind dabei so zu bestimmen, daß die von A_i benötigte Menge nicht größer als b_i ist, das bedeutet mathematisch, daß die x_j folgende Ungleichungen einhalten müssen:

$$a_{i1}x_1 + \dots + a_{in}x_n \leqq b_i \ , \ i=1(1)m \ .$$

Es ist also eine lineare Optimierungsaufgabe der Form

$$\begin{aligned} Z = \underline{c}^T\underline{x} &\longrightarrow \text{Max!} \\ \underline{A}\ \underline{x} &\leqq \underline{b} \\ \underline{x} &\geqq \underline{o} \end{aligned} \qquad (1.4)$$

zu lösen.

1.3. Die Simplexmethode

Die Simplexmethode ist das gebräuchlichste Lösungsverfahren der linearen Optimierung. Moderne Modifikationen der Simplexmethode sind speziell für die Lösung großer linearer Optimierungsaufgaben mit elektronischen Rechenanlagen entwickelt worden. Allen gemeinsam ist die Technik der sukzessiven Verbesserung zulässiger Basislösungen durch Basistransformationen oder Variablenaustauschschritte.
Dazu werden die Nebenbedingungen durch Hinzufügung jeweils einer Variablen (Schlupfvariable, künstlichen Variable, bei OPSI allgemein logische Variable) in Gleichungen umgeformt. Das so entstehende Gleichungssystem $\underline{\tilde{A}}\ \underline{\tilde{x}} = \underline{b}$ hat m+n Unbekannte, aber nur m Gleichungen. Es kann nach m Variablen aufgelöst werden, die dann von den restlichen n Variablen

abhängig sind. Erstere werden als Basisvariable und letztere als Nichtbasisvariable bezeichnet. Das Gleichungssystem lautet dann

$$\underline{\tilde{A}}_B \underline{x}_B + \underline{\tilde{A}}_N \underline{x}_N = 0 \; .$$

Daraus folgt nach Multiplikation mit $\underline{\tilde{A}}_B^{-1}$ von links

$$\underline{x}_B = \underline{\tilde{A}}_B^{-1} b - \underline{\tilde{A}}_B^{-1} \tilde{A}_N x_N$$

Die Matrix $\underline{\tilde{A}}_B$ wird als Basismatrix bezeichnet.
Die Nichtbasisvariablen $\underline{x}_N$ nehmen in der Darstellung ihren Schrankenwert an (bei OPSI zwei Schrankenwerte möglich, vgl. Abschnitt 2.1). Im hier beschriebenen Modell mit Nichtnegativitätsbedingungen ist $\underline{x}_N = \underline{o}$, also ist $\underline{x}_B = \underline{\tilde{A}}_B^{-1} \underline{b}$ ein Lösungsvektor.

Der Grundgedanke der Simplexmethode besteht nun darin, dass ausgehend von einer zulässigen Lösung unter Beachtung des Optimalitätskriteriums (Zielfunktion) schrittweise durch Austausch jeweils einer Basisvariablen mit einer Nichtbasisvariablen der Zielfunktionswert verbessert wird. Zuvor ist mit einem ähnlich gearteten Algorithmus eine zulässige Startlösung zu ermitteln. (Es gehört nicht zum Anliegen dieses Lehrbriefes, die Simplexmethode zu lehren. Das Literaturverzeichnis enthält Hinweise für geeignete Fachliteratur.)

Die Simplexmethode ist ein Lösungsverfahren, das im Falle der Existenz einer optimalen Lösung diese in endlich vielen Schritten ermittelt. Die optimale Lösung ist stets abhängig von der Zielfunktion und den Restriktionen, d.h. abhängig von den vorgegebenen konstanten Werten c_j , b_i und a_{ij} . Veränderungen irgendeiner Konstanten können zu einer anderen optimalen Lösung führen. Abgesehen von Veränderungen innerhalb des Stabilitätsbereiches ist dann die Lösung einer neuen Optimierungsaufgabe erforderlich.

1.4. Die Komplexmethode

Die komplexe Anwendung der linearen Optimierung (Komplexmethode) wird den Anforderungen, die sich aus der Methode der Planausarbeitung an die Methode der Planoptimierung ergeben, gerecht. Mit ihr ist für die betriebliche Planung ein systematischer Weg aufgezeigt, wie durch die wiederholte Anwendung des Optimierungsvorganges schrittweise sowohl die Planungsaufgabe als auch die Planziele selbst verbessert werden können. Dieser Weg gewährleistet die Verbindung des iterativen Planungsprozesses mit einem schrittweisen Optimierungsprozeß, wodurch die Leistungsfähigkeit von Modellen der linearen Optimierung im Prozeß der Planausarbeitung und Plandurchführung verbessert werden kann.

Die Problemstellung besteht darin, daß der für eine Optimierung als konstant aufgefaßte Ausgangszustand im allgemeinen gar nicht konstant, sondern variabel ist. Die mögliche Veränderung des Ausgangszustandes kann für eine optimale Verbesserung des Zielfunktionswertes genutzt werden.

Für den Fall, daß zum Zeitpunkt der ersten Optimierung noch ungenügende Informationen über die Richtung und die Höhe der Veränderung des Ausgangszustandes vorliegen, wird ein Vorgehen nach der im folgenden beschriebenen Komplexmethode vorgeschlagen.

Bei dieser komplexen Anwendung werden aus der 1. Optimierungsstufe Informationen für die Auswahl von veränderungswürdigen Ausgangsdaten gewonnen. Wesentliche Aussagen liefern u.a. die dualen Variablen zur optimalen Lösung. Es muß nun überprüft werden, ob diese Daten beeinflußbar sind und in welcher Höhe die eventuellen Veränderungen realisierbar wären. In der 2. Optimierungsstufe wird dann unter Berücksichtigung der möglichen Veränderungen eine optimale Veränderung des Ausgangszustandes errechnet.

Die Komplexmethode kann in allen Etappen der Planausarbeitung angewendet werden. Sie dient damit der Erarbeitung qualifizierterer Planentwürfe auf der Grundlage der staatlichen Planaufgaben und der Einbeziehung von Vorschlägen für deren

Überbietung. Somit trägt diese Methode zur ständigen Verbesserung des Planentwurfes und zur Erfüllung oder Übererfüllung der staatlichen Auflagen bei. Mit der Komplexmethode können gezielt optimale Veränderungen der Problemdaten linearer Optimierungsaufgaben berechnet werden.

Zur Veränderung zugelassen sind:

- Elemente der rechten Seite,
- Koeffizienten der Bedingungsmatrix und
- Koeffizienten der Zielfunktion.

Die Berechnung der optimalen Veränderungen von Problemdaten kann auch unter Berücksichtigung zusätzlicher Forderungen, wie der vollständigen oder teilweisen Beibehaltung der Optimalbasisstruktur der Grundaufgabe erfolgen.

1.4.1. Optimale Veränderung der rechten Seite

Der Einfachheit halber führen wir die Erläuterung der Komplexmethode anhand der linearen Aufgabenstellung (1.4) durch[1]. Sie wird im Rahmen der komplexen Anwendung der Optimierung als Grundaufgabe bezeichnet. Auf Verallgemeinerungen für die im Abschnitt 2.1 angegebene Aufgabenstellung wird hier nicht eingegangen.

Die Aufgabe (1.4) wird nun als gelöst vorausgesetzt. Eine Lösungsanalyse ist durchzuführen. Zu untersuchen sind die Nebenbedingungen, die als Gleichungen erfüllt sind (aktive Nebenbedingungen). Für sie wird der verfügbare Fonds voll ausgeschöpft, und sie stellen deshalb die Engpässe dar, die das weitere Anwachsen des Zielfunktionswertes nicht zulassen. Die zugehörigen dualen Variablen bewerten diese Engpässe. Ein hoher Wert läßt bei einer Erhöhung des Fonds einen hohen Effekt in der Zielfunktionsverbesserung erwarten. Welche Engpässe durch Erhöhung der Fonds beseitigt werden können,

---

[1] Weitergehende Ausführungen siehe [7], [8].

bedarf ökonomischer Untersuchungen. Es ist also zur Verbesserung der Produktion eine <u>extensive</u> Verbesserung der Produktionsbedingungen durchzuführen, und die optimale extensive Erweiterung ist mittels Optimierungsrechnung zu bestimmen. Es ist folgende verbesserte Aufgabe zu lösen:

$$
\begin{aligned}
Z + \Delta Z &= \underline{c}^T \underline{x} && \longrightarrow \text{Max!} \\
\underline{A}_1 \underline{x} & \leqq b_1 && \\
\underline{A}_2 \underline{x} - \underline{\Delta b}_2 &= b_2 && \\
\underline{x} & \geqq \underline{o} && \\
\underline{\Delta b}_2^u \leqq \underline{\Delta b}_2 & \leqq \underline{\Delta b}_2^o &&
\end{aligned}
\qquad (1.5)
$$

Dabei sind $\underline{\Delta b}_2$ als zusätzliche Variable aufzufassen, die die Fondserhöhung wiedergeben ($b_i^{(erhöht)} = b_i + \Delta b_i$). Sie werden nur zu den Nebenbedingungen hinzugefügt, deren Fonds als beeinflußbar erkannt wurden. Die letzte Ungleichung in (1.5) beschränkt die Fondserhöhungen.

Die Optimierung dieser erweiterten Aufgabe führt im allgemeinen auf einen verbesserten Zielfunktionswert und liefert neben dem optimalen Lösungsvektor $\underline{x}^{(opt)}$ auch den optimalen Fondsveränderungsvektor $\underline{\Delta b}_2^{(opt)}$.

Bemerkungen:

- Die Beschränkung der Fondserhöhung ist häufig nicht erforderlich, da bedingt durch andere Beschränkungen die Fondserhöhung in vertretbaren Grenzen bleibt. Dies ist jedoch modellabhängig.
- Das in Abschnitt 3 behandelte und für OPSI aufbereitete Beispiel enthält keine Beschränkungen für die Fondserhöhung. In diesem Falle ist es für die OPSI-Anwendung günstiger, die $\underline{\Delta b}_2$ indirekt durch Freisetzen der Nebenbedingungen zu berücksichtigen.

1.4.2. Weitere Aufgabenstellungen der Komplexmethode

Im vorangegangenen Abschnitt wurde die Veränderung der rechten Seite betrachtet, d.h., gewisse b_i-Werte wurden als veränderbar angesehen. Die optimale Veränderung ist mit Hilfe der Simplexmethode wieder bestimmbar (Komplexschritt). Entsprechende Aufgaben können auch für den Fall, daß die Koeffizienten a_{ij} oder c_j von (1.4) als beeinflußbar erkannt wurden, formuliert werden.
Die optimale <u>intensive</u> Erweiterung der Produktionsbedingungen (a_{ij} ist beeinflußbar) ergibt sich aus der Lösung folgender verbesserter Optimierungsaufgabe:

$$
\begin{aligned}
Z+\Delta Z &= \underline{c}^T \underline{x} \longrightarrow \text{Max!} \\
\underline{A}_1 \underline{x} &\leqq \underline{b}_1 \\
(\underline{A}_2 + \underline{\Delta A}_2)\underline{x} &= \underline{b}_2 \\
\underline{x} &\geqq \underline{o} \\
\underline{\Delta A}_2^u \leqq \underline{\Delta A}_2 &\leqq \underline{\Delta A}_2^o
\end{aligned}
\qquad (1.6)
$$

$\underline{\Delta A}_2 = (\Delta a_{ij})$ sind dabei die Veränderungen der a_{ij} .

Sollen einige Zielfunktionskoeffizienten c_j optimal verändert werden, so ist folgende verbesserte Aufgabe zu lösen:

$$
\begin{aligned}
Z+\Delta Z &= (\underline{c} + \underline{\Delta c})^T \underline{x} \longrightarrow \text{Max!} \\
\underline{A}\,\underline{x} &\leqq \underline{b} \\
\underline{x} &\geqq \underline{o} \\
\underline{\Delta c}^u \leqq \underline{\Delta c} &\leqq \underline{\Delta c}^o
\end{aligned}
\qquad (1.7)
$$

$\underline{\Delta c} = (\Delta c_j)$ sind dabei die Veränderungen der c_j .

Ausführliche Darlegungen sind in [7] und [8] zu finden.

1.4.3. Allgemeine Aussagen zur Komplexmethode

1. Mit einer einmaligen Optimierung können für eine gegebene Aufgabenstellung (Grundaufgabe) die optimalen Werte der Problemvariablen, mit der Komplexmethode durch eine mehrstufige Anwendung der Optimierung auch gezielte optimale Veränderungen der rechten Seite, der Koeffizienten der Bedingungsmatrix und der Koeffizienten der Zielfunktion bestimmt werden.
   Dabei wird (gegebenenfalls unter vollständiger oder teilweiser Beibehaltung der Struktur der Optimalbasis der ersten Optimierungsstufe) durch mindestens eine weitere Optimierungsstufe eine größtmögliche Verbesserung des Wertes der Zielfunktion erreicht.

2. Bei der Komplexmethode werden die gesamten Wechselbeziehungen zwischen der Lösung der Grundaufgabe und der verbesserten Aufgabe in die mehrstufige Berechnung des Optimums einbezogen. Die gezielte Gesamtheitsbetrachtung bezüglich der Wechselbeziehungen zwischen der Aufgabe und ihrer Lösung entspricht weitestgehend dem praktischen Planungsvorgang. Das wechselseitige Vorgehen zwischen Optimierung, Verbessern der Aufgabenstellung und wiederholtem Optimieren erhöht die Effektivität der Anwendung der linearen Optimierung.

3. Mit Hilfe der Komplexmethode ist man ferner in der Lage, für mehrere unterschiedliche Zielfunktionen mit ähnlicher Zielrichtung eine gemeinsame optimale (indifferente) Lösung zu berechnen, beziehungsweise sich einer solchen Lösung anzunähern. Der Lösungspunkt $\underline{x}^{(opt)}$ wird nach mehreren Optimierungsstufen gegenüber Schwankungen der Zielfunktionskoeffizienten immer unempfindlicher und genügt so einer breiten Klasse von Zielfunktionen.

Insgesamt kann festgestellt werden, daß die Komplexmethode vorrangig für solche Planungsaufgaben geeignet ist, wo Erst-oder Folgeoptimierungen unter unvollständiger Information begonnen beziehungsweise durchgeführt werden müssen und wo man über die Ergebnisinformationen zu neuen Ausgangsinformationen gelangen möchte. Dabei entstehen die neuen Ausgangsinformationen sowohl durch die Ergebnisanalyse als auch durch die Zeitdifferenz zwischen den Ausführungen der einzelnen Optimierungsschritte.

Eine Gesamtübersicht über die möglichen Komplexvarianten kann der Tabelle 1.1 entnommen werden.

Tabelle 1.1 Übersicht der möglichen Komplexvarianten

Optimierungs-Klasse und ihre ökonomische Zielstellung		0	I			II			III		IV			V	
		Grund-auf-gabe	exten-sive Verbes-serung			inten-sive Verbes-serung			ziel-extreme Verbes-serung		restrik-tive Verbes-serung			ziel-intensive Verbes-serung	
Komplex-Variante		0	1	2	3	4	5	6	7	8	9	10	11	12	13
rechte Seite <u>b</u>			x	x	x						x	x	x		
Bedingungs-Matrix <u>A</u>						x	x	x			x	x	x	x	x
Zielfunktions-koeffizienten <u>c</u>									x	x				x	x
Struktur der Optimal-basis	stabil		x			x			x		x			x	
	teil-weise stabil			x			x			x		x			x
	frei				x			x					x		

2. Vorstellung des PS OPSI

Im Rahmen der im Kombinat Robotron entwickelten problemorientierten Systemunterlagen (POS) für mathematische Verfahren ist das PS OPSI für solche Probleme der Optimierung gedacht, die sich mit Hilfe der Simplextechnik lösen lassen. Es erlaubt damit die Lösung vieler praktischer Aufgabenstellungen mit kontinuierlichen Variablen:

- lineare Optimierung einschließlich einparametrischer Erweiterungen
- lineare Quotientenoptimierung einschließlich einparametrischer Erweiterungen
- separable Optimierung
- Vektoroptimierung.

Das PS OPSI ist besonders geeignet für die Lösung großer Optimierungsaufgaben. Damit ist es prädistiniert für Problemstellungen, die in Verbindung mit dem Planungsprozeß stehen, wie die Optimierung von Produktionsprogrammen, Mischungen und Zuschnitten.

2.1. Lösbare Aufgabenstellungen

Bezeichnungen:	x_j,	$j = 1(1)n$	Problemvariable
	y_i,	$i = 1(1)m$	logische Variable
	c_j,	$j = 1(1)n$	Zielfunktionskoeffizienten
	c_o		additive Konstante in Zielfunktion
	a_{ij},	$\left.\begin{array}{l} i = 1(1)m \\ j = 1(1)n \end{array}\right\}$	Koeffizienten der Nebenbedingungen
	b_i^u, a_j^u,	$\left.\begin{array}{l} i = 1(1)m \\ j = 1(1)n \end{array}\right\}$	untere Schranken, für die formal auch $-\infty$ zugelassen sei
	b_i^o, a_j^o,	$\left.\begin{array}{l} i = 1(1)m \\ j = 1(1)n \end{array}\right\}$	obere Schranken, für die formal auch $+\infty$ zugelassen sei.

Das mit OPSI lösbare Modell der linearen Optimierung lautet:

Zielfunktion: $$\sum_{j=1}^{n} c_j x_j + c_o = \text{Extrem!} \qquad (2.1)$$

Nebenbedingungen: $$b_i^u \leqq \sum_{j=1}^{n} a_{ij} x_j \leqq b_i^o \;, \quad i=1(1)m \qquad (2.2)$$

Variablenbeschränkungen: $$a_j^u \leqq x_j \leqq a_j^o \;, \qquad j=1(1)n \qquad (2.3)$$

Durch das Restriktionssystem (2.2), (2.3) wird der zulässige Bereich K der Lösungen beschrieben, aus denen für die gewählte Zielfunktion (2.1) eine optimale Lösung zu ermitteln ist.

Die Variablenbeschränkungen (2.3) können im einzelnen folgende Intervalle für Problemvariable festlegen:

$-\infty < x_j \leqq a_j^o$	einseitig (nach oben) beschränkte Variable	(2.4)
$a_j^u \leqq x_j < \infty$	einseitig (nach unten) beschränkte Variable	(2.5)
$a_j^u \leqq x_j \leqq a_j^o$	zweiseitig beschränkte Variable	(2.6)
$x_j = a_j^x$	fixe oder feste Variable ($a_j^u = a_j^o = a_j^x$)	(2.7)
$-\infty < x_j < \infty$	freie Variable	(2.8)

Die a_j^u, a_j^o und a_j^x, $j \in \{1,2,\ldots,n\}$, sind reelle Konstanten, die keinerlei Vorzeichenbeschränkungen unterliegen.

In dieser Schrift werden nur die folgenden Arten von Nebenbedingungen behandelt:

$$\sum_{j=1}^{n} a_{ij} x_j \geqq b_i^u \qquad (2.9)$$

einseitig (nach unten) beschränkende Nebenbedingung .

$$\sum_{j=1}^{n} a_{ij} x_j \leqq b_i^o \qquad (2.10)$$

einseitig (nach oben) beschränkende Nebenbedingung

$$\sum_{j=1}^{n} a_{ij}x_j = b_i^g \quad \text{Gleichung} \tag{2.11}$$

$$-\infty < \sum_{j=1}^{n} a_{ij}x_j < \infty \quad \text{nichtbeschränkende Nebenbedingung} \tag{2.12}$$

Die a_{ij}, b_i^u, b_i^o und b_i^g, $i \in \{1,2,\ldots,m\}$, $j \in \{1,2,\ldots,n\}$, sind reelle Konstanten, die keinerlei Vorzeichenbeschränkungen unterliegen.

Weitere mit OPSI lösbare Aufgabenstellungen sind:

- Lineare Quotientenoptimierung (Zielfunktion ist der Quotient zweier linearer Funktionen)
- Separable Optimierung (Verwendung von nichtlinearen Funktionen der Form $f_{ij}(x_j)$ anstatt der Summanden $a_{ij}x_j$ in Zielfunktion und/oder Nebenbedingungen)
- Vektoroptimierung (gleichzeitige Berücksichtigung mehrerer Zielfunktionen)
- Einparametrische Optimierung (Parametrisierung der Zielfunktion, der rechten Seite oder von Koeffizienten der Problemmatrix)
- Blockangulare Optimalprobleme (spezielle Strukturierung der Koeffizientenmatrix $\underline{A} = (a_{ij})$)
- Lineare Gleichungssysteme

2.2. Datenbereitstellung

2.2.1. Schematische Darstellung der Problemdaten

Die folgenden Betrachtungen beziehen sich der Einfachheit halber nur auf Aufgabenstellungen der linearen Optimierung. Die Grundprinzipien sind auf die anderen Aufgabenstellungen übertragbar.

1. Grundbegriffe

- Zeilenaktivität: $\sum_{j=1}^{n} a_{ij}x_j$

- Rechte Seite oder RS-Vektor: Spaltenvektor $\underline{b} = (b_i)$
- Spaltenvariablenbereich oder SVB-Vektor:

  Zeilenvektor $\underline{a} = (a_j^u, a_j^o)$
- Problemdaten: Alle Daten einschließlich Namen für Vektoren, Zeilen und Spalten sowie Beschränkungstypangaben für Nebenbedingungen und Problemvariable einer Optimierungsaufgabe.

  Einer Problemmatrix $\underline{A} = (a_{ij})$ können mehrere Zielfunktionen, RS- und SVB-Vektoren zugeordnet sein.

2. Zielfunktionen müssen nichtbeschränkende Nebenbedingungen (2.12) sein. Eine eventuell vorhandene Konstante c_o wird als ein Element der rechten Seite in der Form $b = -c_o$ berücksichtigt.

3. Jede der Nebenbedingungen (2.9) bis (2.12) wird intern durch Einführung einer logischen Variablen (Zeilenvariablen) y_i als Gleichung behandelt:

$$y_i + \sum_{j=1}^{n} a_{ij}x_j = b_i \qquad (2.13)$$

Jeder Nebenbedingung wird ein Zeilentyp zugeordnet:

U nach unten beschränkende Nebenbedingung

O nach oben beschränkende Nebenbedingung

G Gleichung

N nichtbeschränkende Nebenbedingung

Tabelle 2.1: Abhängigkeiten zwischen Zeilentyp, rechter Seite und logischer Variable

Nebenbedingung	b_i	y_i	Zeilentyp
(2.9)	$b_i = b_i^u$	$y_i \leqq 0$	U
(2.10)	$b_i = b_i^o$	$y_i \geqq 0$	O
(2.11)	$b_i = b_i^g$	$y_i = 0$	G
(2.12)	$(b_i = -c_{io})$	$-\infty < y_i < \infty$	N

(Logische Variable werden von OPSI automatisch eingeführt und verlangen deshalb keine Berücksichtigung in den Eingabedaten. Die Kenntnis der Gleichung (2.13) erleichtert das Verständnis der Rechnerprotokolle und die Auswertung des Lösungsprotokolls.)

4. Namenszuordnung
Die Identifizierung aller Konstanten erfolgt nicht durch Indizes, sondern durch Namen: Spaltennamen und Zeilennamen. Dem Problem wird ebenfalls ein Name zugeordnet: der Problemname. Namen sind 8 Zeichen lange Ketten mit alphanumerischen und Sonderzeichen außer % und $\not{S}$.

Typ der Zeile	Zeilenname \ Spaltenname		Spalten der Matrix A SN1 SN2 ... SNn	RS-Vektoren RS1 ... RSk
T_1	Zeilen der Matrix A	ZN1	$a_{11}\ a_{12}\ \dots\ a_{1n}$	$b_1^1 \dots b_1^1$
T_2		ZN2	$a_{21}\ a_{22}\ \dots\ a_{2n}$	$b_2^1 \dots b_2^k$
...		...	....	....
T_m		ZNm	$a_{m1}\ a_{m2}\ \dots\ a_{mn}$	$b_m^1 \dots b_m^k$
	SVB-Vektoren	SVB1	$a_1^{1u}\ a_2^{1u}\ \dots\ a_n^{1u}$ $a_1^{1o}\ a_2^{1o}\ \dots\ a_n^{1o}$	
		....	....	
		SVBr	$a_1^{ru}\ a_2^{ru}\ \dots\ a_n^{ru}$ $a_1^{ro}\ a_2^{ro}\ \dots\ a_n^{ro}$	

Bild 2.1 Schematische Darstellung der Problemdaten

2.2.2. Dateneingabe

Die Datenbereitstellung erfolgt standardmäßig im Kartenformat, das heißt in einem festen Format mit 80 Zeichen Länge. Als Datenträger können Lochkarte, Magnetband oder Wechselplatte verwendet werden. Darüber hinaus ist die Datenbereitstellung in Form von EDO-Dateien möglich.
Das Datendeck zu einem Problem beginnt mit der DNAME-Karte (Datenname) und endet mit der DEND-Karte (Datenende).
Die Daten sind in vier Sektionen zu gliedern, die in enger Beziehung zur Darstellung in Bild 2.1 stehen. Jede Sektion wird durch eine Indikatorkarte mit der Sektionsbezeichnung eingeleitet.

Die Reihenfolge der Sektion ist vorgeschrieben:

ZEILEN		erforderlich
SPALTEN		erforderlich
RSN	(rechte Seiten)	erforderlich
SVBE	(Spaltenvariablenbereiche)	wahlweise

Die Tabelle 2.2 ist eine reduzierte Darstellung des OPSI-Datenformats.

Tabelle 2.2 Datenkarten für die Eingabe der Problemdaten

F1	Feld 2	Feld 3	Feld 4		Feld 5	Feld 6
DNAME		dname				
ZEILEN						
{U / O / G / M}	zname					
SPALTEN						
	sname	zname1	wert1	[	zname2	wert2]
RSN						
	rsname	zname1	wert1	[	zname2	wert2]
SVBE						
{US / OS / FX}	svbname	sname	wert			
FR	svbname	sname				
DEND						
* k o m m e n t a r						

Besondere Hinweise:

- [...] Inhalt in eckigen Klammern ist wahlfrei.
- {:::} Inhalt in geschweiften Klammern ist alternativ.
- Namen stehen im Feld stets linksbündig.
- Vektoren müssen zusammenhängend aufeinanderfolgen.

- Zur SVB-Sektion:

  . $a_j^u = 0$ und $a_j^o = \infty$ werden standardmäßig berücksichtigt und damit auch die Nichtnegativitätsbedingung $x_j \geq 0$. Deshalb ist eine Eingabekarte nicht erforderlich.

  . Ist zu einer Problemvariablen (sname) sowohl eine US- als auch eine OS-Wertangabe erforderlich, so erfolgt diese mit zwei unmittelbar aufeinanderfolgenden Datenkarten.

  . Die Reihenfolge der Spaltenvektoren (sname) der Spaltensektion ist bindend für die Reihenfolge der Schrankenkarten in der SVBE-Sektion.

- Kommentarkarten können im Kartendeck an beliebiger Stelle eingeschoben werden.

- Der Datenname ist nur dem Eingabedatendeck zugeordnet. Die OPSI-interne Abspeicherung erfolgt unter dem Problemnamen. Beide können gleich sein.

Mit OPSI eingelesene Problemdaten werden als <u>Problem</u> auf dem <u>Problemfile</u> (Problemdatei) in aufbereiteter Form gespeichert. Ein Problemfile kann mehrere Probleme enthalten. Sie werden durch den <u>Problemnamen</u> identifiziert. Problemdateien lassen sich mit OPSI leicht warten und ändern.

2.2.3. <u>Datenänderung</u>

OPSI ermöglicht folgende Änderungen von auf dem Problemfile befindlichen Problemen:

- Löschen von Vektoren (Zeilen, Spalten)
- Hinzufügen von Vektoren (Zeilen, Spalten)
- Ändern der Zeilen- und Schrankentypangaben sowie von Elementen in Vektoren

Das Bereitstellen der Änderungsdaten für ein bestehendes Problem hat wie bei der Dateneingabe gemäß Abschnitt 2.2.2 sektionsweise, jedoch ergänzt mit Steuerkarten zur Angabe der Änderungsart, zu erfolgen.

Tabelle 2.3 Karten für die Änderung von Problemdaten

F1	Feld2	Feld3	Feld4	Feld5	Feld6
LOESCHE			Löschen der nachfolgend angegebenen Vektoren: - Zeilenvekt. der Sekt. ZEILEN u. SVBE - Spaltenvekt. der Sekt. SPALTEN u. RSN		
	vname				
	:				
MACH		vname	Hinzufügen von Vektoren <u>nach</u> oder <u>vor</u> dem im Feld 3 angegebenen Vektor: - Datenkarten der Sektion ZEILEN; die zugehörigen Elemente in Spalten- u. RSN-Vektoren sind mit AENDERE in den entsprechenden Sektionen aufzunehmen. - Vollständige Vektoren bei den Sektionen SPALTEN, RSN und SVBE.		
VOR		vname			
datenkarten gem. tab 2.2					
	:	:			
AENDERE			Ändern des Zeilentyps oder von Elementen in Vektoren (auch Hinzufügen oder Löschen von einzelnen Elementen in Vektoren) einschließl. Schrankentypangaben in SVB-Vektoren.		
datenkarten gem. tab 2.2					
	:	:			

Besondere Hinweise:

- vname: zname, sname, rsname oder svbname (Vektorname)
- Jedes Änderungsdatendeck beginnt mit einer DNAME-Karte und endet mit einer DEND-Karte. Der DNAME ist frei wählbar.
- Die Änderungen werden sektionsweise angegeben. Die Sektion wird durch eine Indikator-Karte ZEILEN, SPALTEN, RSN oder SVBE eingeleitet. Innerhalb einer Sektion können beliebig viele der oben genannten Änderungen ausgeführt werden. Die Änderungen sind in der gleichen Reihenfolge, wie die Vektoren des zu ändernden Problems angeordnet sind, vorzunehmen. Nach einer Steuerkarte können mehrere Datenkarten folgen.
- Das Löschen einzelner Elemente in Vektoren hat mit der Funktion Ändern unter Verwendung von Datenkarten mit der Wertangabe Null im Feld 4 oder 6 zu erfolgen.
- Ist in der SVBE-Sektion für eine zweiseitig beschränkte Problemvariable nur eine Schranke zu ändern, so muß auch die zweite Schrankenkarte nochmals eingegeben werden.

F 2.2/1 Für die Optimierungsaufgabe

$$\begin{array}{lllll}
(Z) & x_1 & + 3x_2 & + 2x_3 & = \text{Max!} \\
(N1) & 0{,}27x_1 & & + 1{,}15x_3 & \leqq 112{,}5 \\
(N2) & 4{,}6x_1 & + 1{,}73x_2 & + 3{,}12x_3 & \geqq 240 \\
(N3) & x_1 & & - 2x_3 & = 0
\end{array}$$

$$x_1 \leqq 45, \quad 12 \leqq x_2 \leqq 40, x_3 \geqq 0$$

führen Sie folgende Vorbereitungsarbeiten aus:

a) Zuordnung von Namen zu allen Zeilen (logische Variablen) und Spalten (Problemvariablen)

b) Tabellarische Darstellung der Problemdaten

c) Erstellung des Lochbeleges für die Dateneingabe

F 2.2/2 Die Daten der Optimierungsaufgabe F 2.2/1 sind wie folgt zu ändern:

- Nebenbedingung (N3) ist zu löschen.
- Neu aufzunehmen ist die Nebenbedingung $2{,}45x_1 + 5{,}6x_2 + 1{,}3x_3 \leqq 161{,}5$
- Anstelle $12 \leqq x_2 \leqq 40$ ist $12 \leqq x_2 \leqq 46$ zu verwenden.

Es ist der Lochbeleg für die Änderung der Problemdaten zu erstellen.

2.3. Überblick zum Programmiersystem

Das Programmiersystem OPSI besteht aus einem Compilierprogramm, einem Ausführungsprogramm und einer Menge von Moduln (Prozeduren), die dem Ausführungsprogramm untergeordnet sind und der Realisierung OPSI-spezifischer Funktionen dienen.
Zur Lösung eines Optimierungsproblems ist mittels der OPSI-Sprache das Modell zu beschreiben und die Lösungsstrategie festzulegen (OPSI-Programm). Die OPSI-Prozeduren sind dafür wesentliche Sprachmittel.
Die Verarbeitung eines OPSI-Programms erfolgt in zwei Jobschritten. Im ersten, dem Compilierschritt, wird das OPSI-Programm auf syntaktische Fehler geprüft und für die Weiterverarbeitung aufbereitet. Im zweiten, dem Ausführungsschritt, erfolgt die Ausführung des OPSI-Programms.
Bild 2.2 veranschaulicht das Verarbeitungsprinzip.

Das Programmiersystem OPSI basiert auf

- der revidierten Simplexmethode unter Verwendung der Produktform der Inversen,
- der Upper-Bound-Technik und
- der Mehrfachbewertung (multiple pricing)

Der Optimierungsalgorithmus wird durch eine äußerst effektive Reinversionstechnik unterstützt. Datenbestände werden kompakt gespeichert. Eine dynamische Speicherplatzzuweisung sichert die volle Ausnutzung des verfügbaren Hauptspeichers.

Durch die Verwendung dieser Techniken ist es möglich,

- relativ umfangreiche Probleme zu lösen,
- eine hohe numerische Stabilität bei der Anwendung zu erzielen,
- Rechen- bzw. Ein- und Ausgabezeiten relativ niedrig zu halten.

Die Größe der mit OPSI lösbaren Probleme ist im wesentlichen nur von der Zeilenanzahl abhängig. Die Anzahl von Nebenbedingungen und Zielfunktionen darf zusammen maximal 4095 betragen.

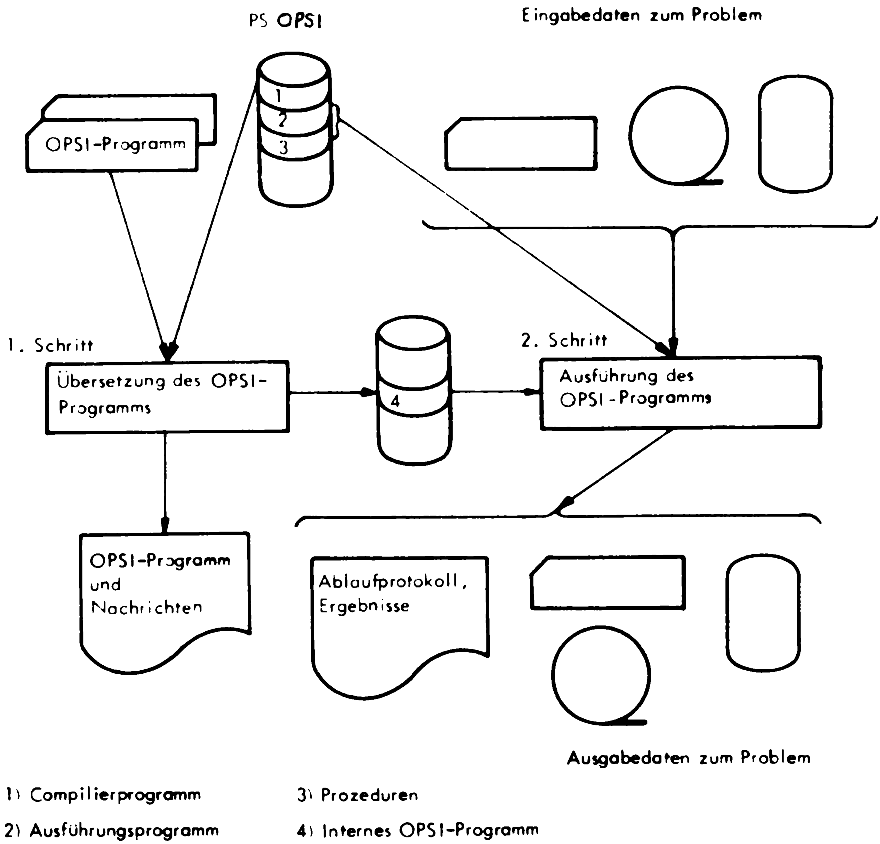


Bild 2.2 Jobschritte bei der Anwendung des PS OPSI

Bei Verwendung eines Programmbereichs von 50 K Byte können Probleme mit bis zu 250 Zeilen gelöst werden. Je weitere 10 K Byte erhöht sich die Zeilenzahl um etwa 250.

2.4. Die OPSI-Sprache

Die folgende Darstellung der Sprache ist eingeschränkt auf ausgewählte Elemente.

2.4.1. Elemente der Sprache

2.4.1.1. Syntaktische Regeln und Anweisungsaufbau

Mittels den OPSI-Sprachanweisungen wird festgelegt, was in welcher Reihenfolge auszuführen ist. Die OPSI-Sprachanweisungen sind einfach und speziell auf die Optimierung zugeschnitten. Sie ähneln Anweisungen bekannter höherer Programmiersprachen. Eine Anweisung enthält eine Operationsbezeichnung und kann als zusätzliche Elemente - meist wahlfrei - noch eine Marke und Argumente besitzen. In Anweisungen verwendete Marken und Konstanten müssen folgenden Konventionen genügen:

- Marke: Zeichenkette maximal 8 Zeichen lang. Erlaubt sind alle Buchstaben und Ziffern, wobei das erste Zeichen ein Buchstabe $\neq$ X sein muß.
  Beispiele: MARKE, L1, L1Ø, FREQ1

- Zeichenkonstante: Zeichenkette eingeschlossen in Hochkommas; maximale Länge 127 Zeichen.
  Beispiele: 'WERT', 'PLAN80-1', 'OPTIMIERUNG MIT ZF1'

Für häufig erforderliche Zeichenkonstanten enthält das Programmiersystem bereits definierte Speicherbereiche. Diese werden symbolisch über vorgegebene Namen, die sogenannten X-Namen, angesprochen. Da der Inhalt dieser Speicherbereiche häufig wie Parameter zur Steuerung des Programmablaufs benutzt wird, werden sie auch als X-Parameter bezeichnet.

Beispiele: XZF, XDNAME, XPBNAME

Anweisungsaufbau für festes Kartenformat:

```
[marke]   opname   [(arg1[,arg2[,...]])]
   |        |                          |
   |        └──── Spalte 10            |
   └───────────── Spalte 1             |
                  maximal Spalte 71 ───┘
```

Es bedeutet:

[...]	Inhalt eingeschlossen in eckige Klammern ist wahlfrei.
{...}	Inhalt eingeschlossen in geschweifte Klammern ist alternativ.
marke	Marke
opname	Operationsname, kennzeichnet inhaltlich die Sprachanweisung.
argi	Argumente, durch die der Operation zusätzliche Angaben zugeordnet werden.

2.4.1.2. Anweisungen der OPSI-Sprache

PROGRAM

Die PROGRAM-Anweisung muß die erste Anweisung eines OPSI-Programms sein.

PEND

Die PEND-Anweisung muß die letzte Anweisung eines OPSI-Programms sein.

[marke] TITLE(zeichenkonstante)

Die TITLE-Anweisung veranlaßt die Ausgabe eines anwendereigenen Titels (Kopfzeile) auf jeder folgenden Druckseite. Die Zeichenkonstante darf maximal 80 Zeichen lang sein.
Beispiel: TITLE('MAXIMIERUNG MIT: ZF=GEWINN, RS=KAP1')

```
[marke]  MOVE(xname1,{zeichenkonstante})
                      {xname2          }
```

Die MOVE-Anweisung überträgt eine Zeichenkonstante, die entweder direkt angegeben ist (zeichenkonstante) oder indirekt angegeben ist (xname2), in den mit xname1 angegebenen Speicherbereich.
Die direkt angegebene Zeichenkonstante darf maximal 8 Zeichen lang sein.

Beispiele: MOVE(XZF,'GEWINN')
MOVE(XAPBNAME,XPBNAME)

Beachte: Zeichenkonstante stets mit, symbolischer Name stets ohne Hochkommas!

```
[marke]  EXIT
```

Die EXIT-Anweisung bewirkt die Beendigung des OPSI-Ausführungsschrittes und die Rückgabe der Steuerung an das Betriebssystem.

```
[marke]  prozedurname [(par1[,par2[,...]])]
```

Die Prozeduraufrufanweisung ist die in einem OPSI-Programm am häufigsten auftretende Anweisung. Der Prozedurname ist durch den Namen der auszuführenden Prozedur zu ersetzen. par1,par2,... sind Prozedurparameter, die häufig wahlfrei sind. Es können Zeichen-, Integer- und Realkonstanten sowie X-Namen sein.
Die Beschreibung der (ausgewählten) Prozeduren erfolgt wegen ihrer speziellen Bedeutung und einer Reihe von Besonderheiten zusammengefaßt im nächsten Abschnitt.

2.4.2. OPSI-Prozeduren

2.4.2.1. Parameter

Als Prozedur wird ein unter der Steuerung des Ausführungsprogramms von OPSI ausführbares Programm verstanden, das eine genau definierte Funktion des Systems realisiert.

Parameter, die zur Ausführung einer Prozedur erforderlich sind oder die diese Ausführung nach Wunsch des Anwenders steuern sollen, müssen vor Ausführung derselben den dafür vorgesehenen Zellen des Verständigungsbereiches (z.B. mittels der MOVE-Anweisung) zugeordnet werden (X-Parameter) oder sie erscheinen als Argument in der Prozeduraufrufanweisung (Prozedurparameter).
X-Parameter tragen umfassenden Charakter, d.h. sie werden im allgemeinen von mehreren Prozeduren verwendet.
Prozedurparameter gelten im allgemeinen nur für die Prozedur, bei der sie als Argument erscheinen. Das schließt nicht aus, daß durch sie X-Parameter aktualisiert werden.

Im Folgenden werden ausgewählte X-Parameter zusammengestellt. Ihr Konstantentyp und evtl. durch OPSI zugeordnete Standardwerte werden mit angegeben.

Tabelle 2.4 Ausgewählte X-Parameter

X-Name	Konstantentyp	Inhalt
XAPBNAME	Zeichenk.	Name des Problems, das durch REVISE zu ändern ist
XDNAME	Zeichenk.	Name des Datendecks für Ein- und Ausgabe im Kartenformat
XPBNAME	Zeichenk.	Name des aktuellen Problems auf dem Problemfile
XRS	Zeichenk.	Name der rechten Seite
XZF	Zeichenk.	Name der Zielfunktion

2.4.2.2. Das Systemmakro INITIAL

Zur Vereinfachung der Programmierung mit der OPSI-Sprache und zur Erhöhung der Sicherheit des Systems enthält OPSI ein Systemmakro namens INITIAL. Dieses setzt eine Reihe von Standardwerten für Toleranzen und einige Parameter. Weiterhin enthält es Standardanweisungsfolgen, die vom System verwendet werden, wenn sich bei der Prozedurabarbeitung dazu die Notwendigkeit ergibt (z.B. bei logischen Fehlern im OPSI-Programm, Übertragungsfehlern, zu hoher numerischer Ungenauigkeit, Erkennen der Un- und Zulässigkeit eines Problems usw.). Damit wird in solchen Ausnahmefällen standardmäßig eine im allgemeinen vernünftige Weiterführung oder Beendigung der Rechnung durchgeführt. Im OPSI-Programm ist INITIAL wie eine Prozedur zu verwenden und sollte unmittelbar nach der PROGRAM-Anweisung stehen. Das Verzweigen zu Anweisungsfolgen von INITIAL ist im Ergebnisprotokoll zu erkennen.

2.4.2.3. Prozedurübersicht und Datenfluß

Bild 2.3 Prozedurübersicht und Datenfluß siehe nächste Seite.

2.4.2.4. Beschreibung der Prozeduren

[marke] CONVERT

X-Parameter: XDNAME, XPBNAME
CONVERT liest die in XDNAME spezifizierten Daten ein und speichert sie auf dem Problemfile unter dem XPBNAME zugeordneten Namen ab.

[marke] REVISE[('KBASIS')]

X-Parameter: XDNAME, XAPBNAME, XPBNAME
REVISE liest die in XDNAME spezifizierten Änderungsdaten ein, revidiert das alte Problem (Name in XAPBNAME) und speichert sie unter dem XPBNAME zugeordneten Namen wieder auf dem Problemfile ab. Sind die Namen in XAPBNAME und XPBNAME gleich, so wird das alte Problem überspeichert und nachfolgende zerstört. Ist der Parameter 'KBASIS' angegeben, so werden sichergestellte Basislösungen nicht wieder abgespeichert.

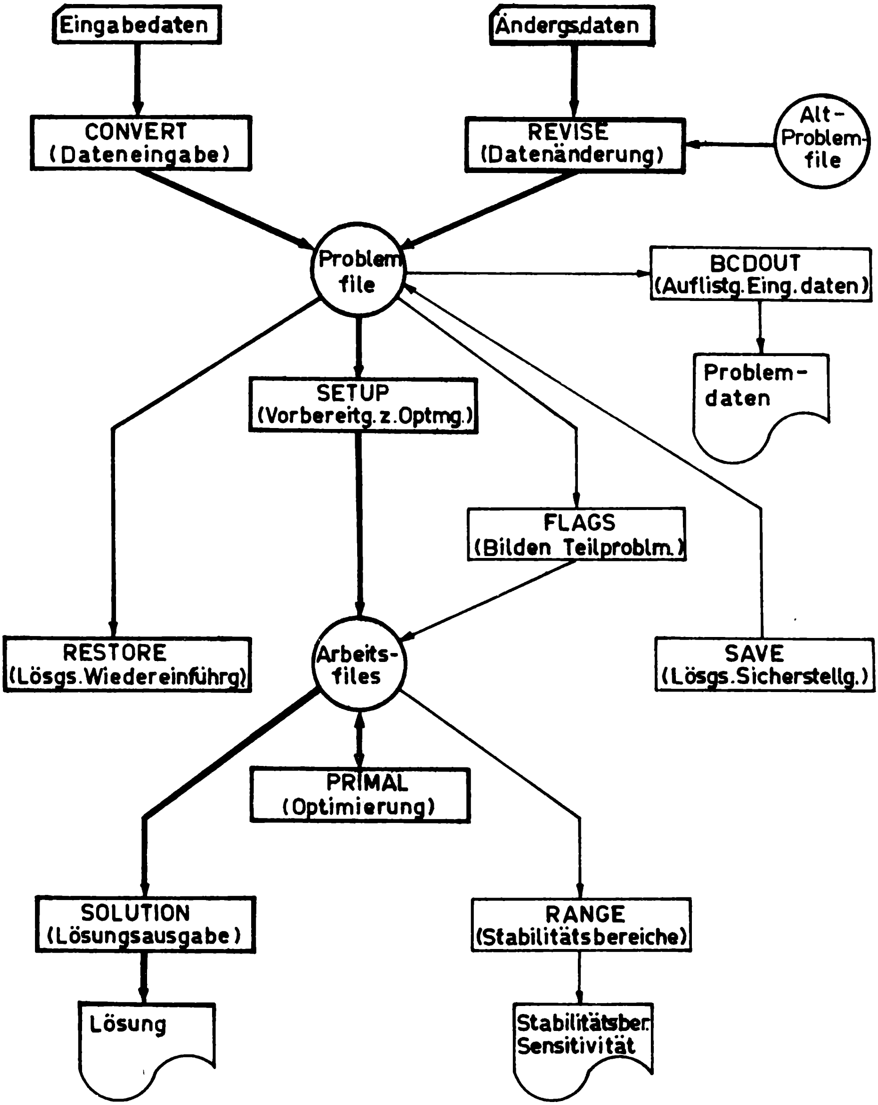


Bild 2.3 Prozedurübersicht und Datenfluß

```
[marke]  BCDOUT
```

X-Parameter: XPBNAME, XDNAME

BCDOUT gibt die Daten eines auf dem Problemfile abgespeicherten Problems (Name in XPBNAME) unter dem XDNAME zugeordneten Namen aus.

```
[marke] SETUP [('SVB',vektorname[,'MAX'])]
```

X-Parameter: XPBNAME

SETUP bereitet das Problem (Name in XPBNAME) für die Optimierung vor und erzeugt eine Startlösung. Falls Spaltenvariablenbeschränkungen verwendet werden, so ist der entsprechende Vektorname (in Hochkomma) anzugeben. Standardmäßig wird für die Minimierung vorbereitet. Soll maximiert werden, so ist 'MAX' anzugeben.

```
[marke]  FLAGS[(['ZMASKEN','zname1',...,'znamer',' ',]
                 ['SMASKEN','sname1',...,'snamep',' '])]
```

X-Parameter: keine

FLAGS kann verwendet werden, um nach SETUP aus einem Problem ein Teilproblem zu bilden. Es werden die mit der Zeilenauswahlliste 'ZMASKEN',...,' ' angegebenen Zeilen in N-Typ-Zeilen umgewandelt und die mit der Spaltenauswahlliste 'SMASKEN',...,' ' angegebenen Spalten aus dem Problem entfernt. Bei mehrfacher FLAGS-Anwendung nach SETUP wird stets erneut vom vollständigen Problem ausgegangen, insbesondere stellt ein FLAGS-Aufruf ohne Auswahllisten wieder das vollständige Problem her.

Bemerkung: Die Lochkarte kann für Parameterangaben einschließlich abschließender Klammer nur bis zur Spalte 71 genutzt werden. Durch Lochen von * in Spalte 72 kann die Parameterfolge auf der nächsten Karte beginnend mit Spalte 1 fortgesetzt werden.

[marke] PRIMAL

X-Parameter: XZF, XRS
PRIMAL führt die Optimierung für das durch SETUP vorbereitete Problem aus. Zielfunktion und rechte Seite sind durch X-Parameter festzulegen.

[marke] RANGE[(['SKT1','/'],...,['SKT5','/'])]

X-Parameter: XZF, XRS
Mittels RANGE können für ein bereits optimiertes Problem die Stabilitätsbereiche und die zugehörige Sensitivität ermittelt werden. Dabei wird stets die Veränderung jeweils nur einer Schranke betrachtet. Die Berechnung wird sektionsweise ausgeführt:

Sektion 1:	Stabilitätsbereiche für	Zeilen an den Schranken
Sektion 2:	" "	Spalten " " "
Sektion 3:	" "	Zeilen in der Basis
Sektion 4:	" "	Spalten " " "
Sektion 5:	" "	Zielfunktions-koeffizienten

Die Angabe von Parameterpaaren 'SKT$_j$','/' ($j \in \{1,2,\ldots,5\}$) bewirkt, daß die Berechnung für die angegebenen Sektionen unterbleibt.

[marke] SOLUTION

X-Parameter: XPBNAME
SOLUTION gibt die augenblickliche Lösung des Optimierungsproblems aus.

[marke] SAVE [('NAME','name')]

X-Parameter: XPBNAME
SAVE stellt den momentanen Lösungsstatus eines Problems unter dem Namen "name" auf dem Problemfile für die spätere Wiederverwendung sicher.

```
[marke]  RESTORE [('NAME','name')]
```

X-Parameter: XPBNAME

RESTORE stellt den Lösungsstatus eines Problems her, wie er zum Zeitpunkt der SAVE-Anwendung vorlag.

2.4.3. Das Schreiben eines OPSI-Programms

Das Schreiben eines OPSI-Programms besteht in der von der für das Modell zu verfolgenden Lösungsstrategie abhängigen Aneinanderreihung von Anweisungen der OPSI-Sprache unter Verwendung der die Problemdaten bezeichnenden Namen (z.B. Datenname, Zielfunktion, rechte Seite usw.).
Ein OPSI-Programm wird stets durch die Anweisungen PROGRAM und PEND begrenzt.
Dem nachfolgenden Beispiel-Programm liegt folgende Aufgabenstellung zugrunde:

Beschreibung der Problemdaten:

Name der Datendecks:	DATEN
RS-Vektoren:	KAP1, KAP2
SVB-Vektor:	MENGE
Zielfunktion:	GEWINN

Lösungsstrategie:

- Abspeicherung der Problemdaten unter dem Problemnamen PLAN1
- Maximierung der Zielfunktion GEWINN unter Verwendung des RS-Vektors KAP1 und SVB-Vektors MENGE
- Ausgabe der letzten Lösung (sie kann auch unzulässig sein!)

OPSI-Programm:

```
PROGRAM                          Programmbeginn
INITIAL                          Systemmakro
MOVE(XDNAME,'DATEN')          }  Einlesen des Datendecks DATEN
MOVE(XPBNAME,'PLAN1')         }  und Abspeicherung auf den
CONVERT                       }  Problemfile als Problem PLAN1
BCDOUT                           Kontrolldruck der Problemdaten
```

SETUP('SVB','MENGE','MAX')	Vorbereitung der Optimierung unter Einbeziehung des SVB-Vektors MENGE
MOVE(XRS,'KAP1') MOVE(XZF,'GEWINN') PRIMAL	Maximierung der Zielfunktion GEWINN unter Verwendung des RS-Vektors KAP1
SOLUTION	Lösungsausgabe
EXIT	Rückgabe Steuerung an das Betriebssystem
PEND	Programmende

Das vom Anwender geschriebene OPSI-Programm ist nun für die rechentechnische Verarbeitung mit dem PS OPSI vorzubereiten. Das Verarbeitungsprinzip wurde bereits in Abschnitt 2.3 beschrieben. Insgesamt ist ein Job zusammenzustellen, der gemäß Bild 2.4 aufgebaut ist. Alle notwendigen Jobsteueranweisungen können den Rechnerausdrucken im Abschnitt 4 entnommen werden.

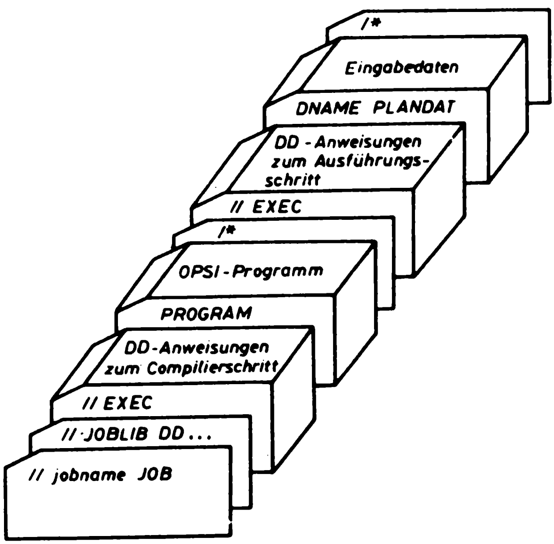


Bild 2.4 Kartendeck für eine Problembearbeitung

F 2.4/1 Für die Optimierungsaufgaben F 2.2/1 und F 2.2/2 sind die erforderlichen OPSI-Programme zu schreiben.

a) Lösungsstrategie für die Ausgangsaufgabe
   - Dateneingabe, Abspeicherung unter dem Namen BEISP1
   - Maximierung der Funktion GEWINN
   - Lösungsausgabe
   - Lösungssicherstellung unter dem Namen LOESG1

b) Lösungsstrategie für geänderte Aufgabe
   - Eingabe der Änderungsdaten, Abspeicherung unter dem Namen BEISP2
   - Maximierung unter Verwendung der sichergestellten Lösung
   - Lösungsausgabe
   - Neue Problemvorbereitung (SETUP) ohne Beschränkungen für die Problemvariablen
   - Maximierung
   - Lösungsausgabe

F 2.4/2 Was ist zu beachten, wenn dem geänderten Problem der gleiche Name wie dem Ausgangsproblem zugeordnet wird (hier BEISP1)?

2.5. Hinweise zur Anwendung des PS OPSI

2.5.1. Allgemeines Anwendungsprinzip

Voraussetzung für die Nutzung des PS OPSI ist, daß das System auf der EDVA bereits eingerichtet ist (einmaliger Vorgang). Die Durchführung einer Problemlösung erfordert im allgemeinen folgende Arbeitsweise:

1. Modellierung des ökonomischen, technologischen oder technischen Prozesses unter Beachtung der betrieblichen Besonderheiten und Klärung, inwieweit aus der Modellierung eine mit dem PS OPSI lösbare Aufgabenstellung der mathematischen Optimierung abgeleitet werden kann.
2. Beschreiben des Problems und Festlegen der modellabhängigen Lösungsstrategie in der optimierungs-spezifischen OPSI-Sprache sowie Vorbereiten der Problemdaten in einem Format, das vom PS OPSI verarbeitet werden kann.
3. Zusammenstellen des OPSI-Jobs (Abb. 2.4).
4. Ausführen der OPSI-Rechnung auf der EDVA. Die Optimierung, Ein- und Ausgabe sowie eventuelle Datensicherstellungen erfolgen nach der im OPSI-Programm festgelegten Lösungsstrategie.
5. Auswerten und Lösungsanalyse. Abhängig von der Auswertung der Ergebnisse können Modellveränderungen erforderlich sein, die eine nochmalige Optimierung mit veränderten Daten oder geänderter Lösungsstrategie erfordern.

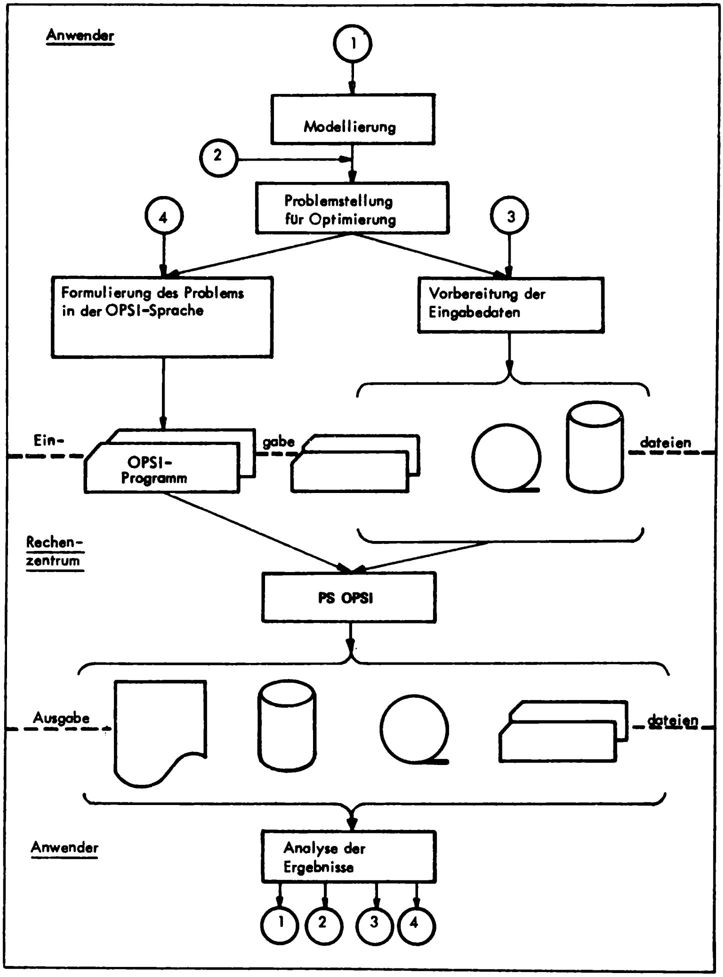


Bild 2.5 Allgemeines Anwendungsprinzip

2.5.2. Aufeinanderfolgende (mehrstufige) OPSI-Anwendungen

Ein Optimierungsproblem wird im allgemeinen nicht in einem Rechengang gelöst. Datenfehler, ein widersprüchliches Restriktionssystem (Abbruch wegen Feststellung der Unzulässigkeit), aber auch nachträgliche Modellkorrekturen, wie sie u.a. beim Anwenden der Komplexmethode auftreten, führen zu aufeinanderfolgenden OPSI-Anwendungen. Nachfolgend werden für einige häufig auftretende Problemkreise Hinweise für ein rationelles Arbeiten mit OPSI gegeben.

1. Behandlung von Datenfehlern

Eine Vielzahl von Datenfehlern wurden von CONVERT erkannt und entsprechend protokolliert. Nachfolgende Korrekturen erfordern meist keine erneute vollständige Dateneingabe, sondern nur eine Änderung des bestehenden Problems mittels REVISE. Als Hilfsmittel für die leichtere Fehlererkennung und -analyse kann bei Vorliegen relativ umfangreicher Probleme neben dem recht ausführlichen BCDOUT-Ausdruck folgendes empfohlen werden:

- Verwenden der Parameter 'STATISTK' und 'TEST' für CONVERT und REVISE (z.B. CONVERT('STATISTK','TEST')).
- Einschub von Kommentarkarten besonders bei umfangreicher SPALTEN-Sektion zur besseren Lokalisierung von Fehlern in Spaltenvektoren.
- Verwenden der Prozedur PICTURE (nach SETUP). Sie gibt einen Überblick über die Besetzung und Größenordnung der Elemente der Problemmatrix, der RS-Vektoren sowie der mit SETUP zugewiesenen SVB-Vektoren.

2. Analyse und Behandlung von Problemen mit unzulässiger Lösung

Bei widersprüchlichem Restriktionssystem wird von OPSI die Optimierung nach Erkennen der Unzulässigkeit abgebrochen. Die zuletzt erreichte unzulässige Lösung ergibt sich algorithmusbedingt und ist ohne modellierende Eingriffe des Anwenders nicht steuerbar. Das hat zur Folge, daß häufig

Verletzungen bezüglich solcher Restriktionen ausgewiesen werden, die nicht ohne weiteres abgeändert werden können.

Die Analyse unzulässiger Lösungen erfolgt vorrangig anhand der SOLUTION-Ausgabe. Zur Behebung der Unzulässigkeit können in der Lösung angenommene oder verletzte Schranken der Nebenbedingungen und Problemvariablen, aber auch Elemente der Aufwandsmatrix, die zu echt beschränkenden Nebenbedingungen oder Problemvariablen gehören, verändert werden. Verletzungen der Zulässigkeit sind im SOLUTION-Protokoll in der STATUS-Spalte (ST) mit ** gekennzeichnet. Die Größe der Verletzung ergibt sich aus der Differenz von verletzter Schranke und in Anspruch genommener Aktivität. Für eine verletzte Zeile besteht speziell folgender Zusammenhang mit der logischen Variablen:

$$b_i - \sum_{j=1}^{n} a_{ij}x_j = y_i \begin{cases} >0 \text{ für } b_i^u \\ <0 \text{ für } b_i^o \end{cases} .$$

3. Optimierungsrechnungen mit unterschiedlichen rechten Seiten, Zielfunktionen und SVB-Vektoren

Die nacheinanderfolgende Ausführung der Optimierungsrechnungen kann in einem OPSI-Programm allein durch nacheinanderfolgende Zuweisung der entsprechenden Namen der Vektoren mittels der MOVE-Anweisung erfolgen. Falls kein erneutes SETUP gefordert wird, ist die jeweils erhaltene optimale Lösung die Startlösung für die nächste Optimierung.

4. Zielgerichtete Verbesserung optimaler Lösungen (Komplexmethode)

Mit der Komplexmethode angewandt auf ein lineares Optimierungsmodell als Grundaufgabe wird die Absicht verfolgt, durch die gezielte Veränderung nach ökonomischen Gesichtspunkten ausgewählter Problemdaten eine Verbesserung des Optimums zu erreichen. Dies bedeutet eine wiederholte OPSI-Anwendung auf die modifizierte Grundaufgabe. Die Realisierung kann in unterschiedlichen Varianten erfolgen:

a) Keine Einschränkung bezüglich der Struktur der Optimalbasen
b) Teilweise Fixierung der Struktur
c) Vollständige Fixierung der Struktur entsprechend der ursprünglichen Optimallösungen

In den Modellen für die extensive und intensive Erweiterung der Produktion wird dabei die ursprüngliche Problemmatrix $\underline{A} = (a_{ij})$ weiter verwendet. Alle Erweiterungen bezüglich der Problemdaten können grundsätzlich durch Änderung der ursprünglichen Problemdaten mittels der OPSI-Prozedur REVISE ausgeführt werden.[1] Im Falle der Umwandlung von (beschränkenden) Nebenbedingungen in N-Typzeilen, d.h. in nichtbeschränkende Nebenbedingungen, kann vorteilhaft auch die OPSI-Prozedur FLAGS angewandt werden.

Die komplexe Behandlung eines Planungsproblems mittels OPSI ist durch folgende Schritte gekennzeichnet:

1. Schritt: Lösung der Grundaufgabe mittels OPSI mit Sicherstellung des Problems. Häufig geht diesem Schritt das Zulässigmachen des Problems voraus.

2. Schritt: Analyse der optimalen Lösung bezüglich Fondsauslastung und Absatzbeschränkungen unter besonderer Beachtung der dualen Variablen (SOLUTION und/oder RANGE-Protokoll). Ist die Lösung nicht befriedigend, so sind (vorwiegend) für die Engpaßrelationen Änderungsmöglichkeiten für die Daten zu untersuchen, ansonsten Beendigung der Berechnung.

3. Schritt: Erstellung der Änderungsdaten und Optimierung des geänderten Problems.
Erneut Ausführung von Schritt 2.

---

[1] Ausführliche Darlegungen zur gezielten Veränderung der Elemente der Aufwandsmatrix $\underline{A}$, der rechten Seite und der Zielfunktion einschließlich Ihrer rechentechnischen Realisierung mit dem PS OPSI siehe [7] und [8].

2.6. Weiterentwicklung des PS OPSI

Kennzeichnend für SUL komplexen Charakters ist es, daß sie ständig weiterentwickelt werden. Neben Anpassungen und Erweiterungen, die sich vorwiegend auf Veränderungen der Betriebssysteme und Gerätetechnik begründen, sind dies vor allem die Einbeziehung neuer Algorithmen und die Verbesserung der Anwendungsfreundlichkeit. Für das PS OPSI ist in dieser Richtung folgendes vorgesehen:

- Behandlung konvexer quadratischer Zielfunktionen
- Verbesserung der Kopplungsfähigkeit zu anderen SUL für mathematische Verfahren
- Möglichkeiten der anwenderspezifischen Ergebniszusammenstellung und -auswertung
- Erweiterung der Ein- und Ausgabemöglichkeiten von/auf EDO-Dateien

3. Ausgewählte Anwendungsbeispiele

Das Ziel dieses Abschnittes ist es, die ökonomische Leistungsfähigkeit der linearen Optimierung in Verbindung mit dem PS OPSI aufzuzeigen. Als allgemeinverständliches Anwendungsgebiet wurde die Produktionsplanung gewählt. Mit Absicht wurde als Anwendungsbeispiel kein fiktives, sondern ein konkretes Beispiel aus der Praxis in zwei Darstellungsformen verwendet. Es ist gewollt, daß der Studierende erfährt, wie eine Aufgabe aus dem täglichen Planungsgeschehen mit den ökonomisch-mathematischen Methoden und der ESER-Rechentechnik bewältigt werden kann.

Mit der Wahl eines Anwendungsbeispiels verhältnismäßig großem Problemumfanges wurde das Ziel verfolgt,

<u>erstens</u> den Qualitätsunterschied zwischen der einfachen (Abschnitt 3.1.) und der komplexen Produktionsplanoptimierung (Abschnitt 3.2.) zu verdeutlichen und

zweitens die Effektivität des PS OPSI im Abschnitt 4 überzeugend nachzuweisen.

3.1. Einfache Produktionsplanoptimierung

3.1.1. Problemanalyse

Der VEB Excelsiorwerk ist ein volkswirtschaftlich wichtiger Betrieb der bezirksgeleiteten Industrie Leipzig und stellt als Finalproduzent elektrische Meßinstrumente und Meßeinrichtungen her. Er wendet die einfache Produktionsplanoptimierung seit dem Jahre 1968 und die komplexe seit 1975 an.

3.1.1.1. Problemvariable

Das gesamte Produktionsprogramm umfaßt Erzeugnisse in etwa 2 500 verschiedenen Ausführungen. Da zum Zeitpunkt der Planung noch keine ausreichend spezifizierten Bedarfsanforderungen für die Einzelausführungen der Grundtypen vorliegen, wurde in den letzten Jahren diese Erzeugnisvielfalt für Planungszwecke auf 16 Typenvertreter verdichtet. Diese bilden die Grundlage für die gesamte Planvorbereitung, Planausarbeitung und Plandiskussion. Sie beinhalten Erzeugnisklassen, denen im mathematischen Modell die Problemvariablen entsprechen.
Die den einzelnen Typenvertretern zugeordneten Problemvariablen werden in ihrer Variationsbreite durch die Absatzbedingungen beschränkt. Die Absatznotwendigkeiten bilden die unteren und die Absatzmöglichkeiten die oberen Schranken der Problemvariablen. Dieser Variationsspielraum liegt im Durchschnitt bei 20 [%]. Es ist aber auch notwendig, einige Erzeugnisse mit fester Stückzahl vorzugeben. Die Erzeugnisklassen und die entsprechenden Schlüsselnummern der Problemvariablen, die gleichzeitig ihre Namen sind, enthält Tabelle 3.1.

Tabelle 3.1 Erzeugnisklassen und zugehörige Schlüsselnummern bzw. Namen der Problemvariablen

lfd. Nr.	Bezeichnung der Erzeugnisklassen	Schlüssel-Nr. im Modell
1	E 48 DS	010
2	D 52 DS	020
3	V 42 DS	030
4	Zellenprüfer	040
5	B 72 DS	050
6	E 72 DS	060
7	C 96 DS	070
8	P 144	080
9	P 150	090
10	E 48 DE	100
11	B 72 DE	110
12	E 72 DE	120
13	B 72 BE Spezial	130
14	Autostarter	140
15	Handvoltmeter	150
16	Kreuzzeiger	160

3.1.1.2. Nebenbedingungen

Als Nebenbedingungen wurden nur die für den Produktionsplan wesentlichen Restriktionen erfaßt. Das sind im vorliegenden Fall die Maschinenkapazitäten der Vorfertigung und die Handarbeitsplätze der Montagekollektive. Da die einzelnen Beschäftigtengruppen der Montage je nach der Qualifikation nur bestimmte Arbeiten ausführen können, bildet die Mehrzahl der Lohngruppen die entscheidenden Restriktionen.

Die technologischen Zeitnormative und Normative für Kosten, Gewinne und Preise wurden manuell verdichtet. Dabei war es auf Grund einer ausreichenden mittelfristigen Stabilität der Sortimentsstruktur möglich, die abgesetzten Erzeugnisstückzahlen des Basisjahres als Wichtungsfaktoren einzubeziehen.

In die Modellmatrix wurden folgende Restriktionen und Bilanzzeilen aufgenommen:

- Beschränkungen bezüglich des Maschinenzeitfonds der Vorfertigung und des Arbeitszeitfonds nach Montagelohngruppen.

- Bilanzzeilen für ausgewählte Materialarten, die Gesamtkapazität des Arbeitszeitfonds, den Grundlohn, die Warenproduktion und den Gewinn.

Im Rahmen der Problemanalyse und der Modellierung wurden die <u>Aufwandskoeffizienten</u> für die einzelnen Leistungsstellen mit Hilfe der Arbeitsplatzstammkarten und der Stücklisten ermittelt. Dabei mußten gewichtete Mittelwerte, entsprechend der Häufigkeit der bisher abgesetzten Einzelerzeugnisse, berechnet werden. Diese Daten wurden mit großer Sachkenntnis vom Bereich Technologie zugearbeitet. Die hohe Zuverlässigkeit dieser Problemdaten hatte für die Aussagefähigkeit des Planungsmodells eine entscheidende Bedeutung.

Die ökonomischen Abgaben wie Preise (IAP, BP), Gewinn, Grundlohn und Grundmaterial konnten den Kalkulationsunterlagen entnommen werden. Vom Bereich der Produktionsleitung wurden auf der Grundlage der nominellen Zeitfonds die effektiven Maschinen- und Arbeitszeitfonds ermittelt. Die Aufschlüsselung der geplanten Rationalisierungsmaßnahmen auf die Kapazitätsbilanz erfolgte durch den Bereich Technik.

Der Inhalt der Restriktions- und Bilanzzeilen sowie die entsprechenden Zeilennamen, deren drei Ziffern die Schlüsselnummern im Modell verkörpern, sind der Tabelle 3.2 zu entnehmen.

Tabelle 3.2 Restriktions- und Bilanzzeilen und zugehörige Zeilennamen

lfd. Nr.	Bezeichnung der Restriktions- und Bilanzzeilen	Zeilennamen
1	Vorfertigung - Exzenterpresse	V010
2	" - Drehmaschine	V020
3	" - Bohrmaschine	V030
4	" - Tischexzenterpresse	V040
5	" - Gewindeschneide-maschine	V050
6	" - Handarbeiten	V060
7	" - Gesamt	V065
8	F 10 - Lohngruppe 3	F070
9	F 10 - " 4	F080
10	F 10 - " 5	F090
11	F 10 - Gesamt	F095
12	F 21 - Lohngruppe 3	F100
13	F 21 - " 4	F110
14	F 21 - " 5	F120
15	F 21 - " 6	F130
16	F 21 - " 7	F140
17	F 21 - Gesamt	F145
18	F 22 - Lohngruppe 3	F150
19	F 22 - " 4	F160
20	F 22 - Gesamt	F165
21	F 31 - Lohngruppe 3	F170
22	F 31 - " 4	F180
23	F 31 - " 5	F190
24	F 31 - " 6	F200
25	F 31 - Gesamt	F205
26	F 33 - Lohngruppe 3	F210
27	F 33 - " 4	F220
28	F 33 - Gesamt	F225
29	F 40 - Lohngruppe 3	F230
30	F 40 - " 4	F240
31	F 40 - " 5	F250
32	F 40 - " 6	F260
33	F 40 - Gesamt	F265
34	F 51 - Lohngruppe 5	F270
35	F 51 - Gesamt	F275
36	F 52 - Lohngruppe 3	F280
37	F 52 - " 4	F290
38	F 52 - " 5	F300
39	F 52 - Gesamt	F305
40	F 65 - Lohngruppe 5	F310
41	F 65 - Gesamt	F315

Fortsetzung von Tabelle 3.2

lfd. Nr.	Bezeichnung der Restriktions- und Bilanzzeilen	Zeilennamen
42	Kapazität Gesamt	KAP.GES
43	Grundlohn	GRDLOHN
44	Grundmaterial	GRDMAT
45	Blech 0,8 mm AlMG-3 F 26	BLECHØ8
46	Warenproduktion zu BP	WP-BP
47	Warenproduktion zu IAP	WP-IAP
48	Gewinn	GEWINN

3.1.1.3. Optimalitätskriterien

Das vorrangige Optimalitätskriterium des Betriebes ist die maximale Warenproduktion zu Industrieabgabepreisen. Weiterhin interessiert sich die Betriebsleitung für die maximal erreichbare Warenproduktion zu Betriebspreisen und für das maximale Betriebsergebnis (Gewinn). Schließlich hat der Betrieb ein besonderes Interesse an solchen Informationen, mit denen er im Plan Wissenschaft und Technik auf zielgerichtete Rationalisierungsaufgaben orientieren kann, um seine Produktionsmöglichkeiten extensiv und intensiv erweitern zu können.

3.1.2. Modellierung

Die Ergebnisse der Problemanalyse führen auf das mathematische Modell der linearen Optimierung. Für jedes Optimalitätskriterium (Zielfunktion) ergibt sich je eine Aufgabenstellung (v = I,II,III):

$$z^{(v)} = \sum_{j=1}^{16} c_j^{(v)} x_j + c_o^{(v)} \qquad = \text{Max!}$$

$$\sum_{j=1}^{16} a_{ij} x_j \leqq b_i \qquad \text{für } i = 1(1)41 \qquad (3.1)$$

$$\sum_{j=1}^{16} a_{ij} x_j \qquad \text{unbeschränkt für } i = 42(1)48$$

$$a_j^u \leqq x_j \leqq a_j^o \qquad \text{für } j = 1(1)16$$

Erläuterungen:

$z^{(v)}$ Zielfunktionswert der Variante v

$\underline{c}^{(v)}$ Zielfunktionskoeffizienten der Variante v

$c_o^{(v)}$ additive Konstante zur Zielfunktion der Variante v = III

$\underline{x}$ Problemvariable

Die restlichen Koeffizienten und Konstanten sind wie im Abschnitt 2 verwendet.

Entsprechend diesem Modellaufbau wurden die Problemdaten OPSI-gerecht für die Dateneingabe erstellt. Sie sind im Abschnitt 4.2 als BCDOUT-Ausdruck vollständig wiedergegeben.

Die rechentechnische Lösung des mathematischen Modells der einfachen Produktionsplanoptimierung gemäß (3.1) als Grundaufgaben bezüglich der drei Zielfunktionen wurde mit Hilfe des PS OPSI durchgeführt.
Wichtige Druckausgaben zum Job, insbesondere der Lösungen, enthält Abschnitt 4.2 .

3.1.3. Auswertung der Ergebnisse der einfachen Produktionsplanoptimierung

Die Optimierungsrechnung für den Jahresplan des Betriebes wurde zunächst nacheinander nach den drei Zielfunktionen in einer ersten Optimierungsstufe durchgeführt.

In Auswertung dieser drei Einzelvarianten ergeben sich folgende Zielfunktionswerte, vgl. Tabelle 3.3:

Tabelle 3.3: Zielfunktionswerte für die drei Optimierungen (Angaben in Mark)

ZF / Zeilen	$Z_{max}^{I(o)}$ = WP-IAP [M]	$Z_{max}^{II(o)}$ = WP-BP [M]	$Z_{max}^{III(o)}$ = GEWINN [M]
WP - IAP	<u>15 583 644</u>	15 583 615	15 570 325
WP - BP	13 069 834	<u>13 069 897</u>	13 061 672
GEWINN[1)]	2 869 981	2 870 604	<u>2 873 767</u>

1)Es ist GEWINN = Zeilenaktivität - 3,34 Mio. Mark = -(log. Variable)

Obwohl die einzelnen Zielvarianten nur geringfügige Unterschiede in den Zielfunktionen liefern, besitzen ihre zugehörigen Lösungsvektoren wesentliche Abweichungen in den Problemvariablen. Im Hinblick auf die noch folgende Indifferenzbetrachtung ist diese Feststellung beachtenswert.

Aufbauend auf den ersten berechneten Ergebnissen der Grundaufgabe im Rahmen der ersten Planungsrunde (vgl. auch Tabelle 3.3) standen vor der Betriebsleitung in bezug auf nachfolgende Planungsrunden folgende Fragen im Vordergrund:

- Wie ist eine Steigerung der Produktion auf 16,2 Mio. Mark am effektivsten möglich?
- Wie verändert sich dabei das Kapazitäts- und Absatzprofil?
- Wo und in welcher Höhe müssen Rationalisierungsmaßnahmen wirksam werden?

Antworten auf diese wichtigen Leitungs- und Planungsfragen lassen sich aus der Anwendung der Komplexmethode ableiten.

3.2. Komplexe Produktionsplanoptimierung mit gezielter Veränderung der rechten Seite

3.2.1. Festlegung der beeinflußbaren Fonds

Für die im Abschnitt 3.1. beschriebene Grundaufgabe sind im Abschnitt 4.2. die wesentlichen Ergebnisprotokolle angegeben. Die Auswertung dieser Protokolle ist die Voraussetzung für das Aufstellen des ersten Komplexschrittes. Das Ziel der komplexen Veränderung der Fonds besteht in der ökonomischen Notwendigkeit, die Warenproduktion zu Industrieabgabepreisen (WP-IAP) um ca. 4 % auf 16,2 Mio. Mark zu erhöhen.

Die Analyse der Lösung der Grundaufgabe ist vom fachkundigen Planungskollektiv durchzuführen. Auszuwerten sind die SOLUTION- und RANGE-Protokolle für alle drei Lösungen. Zu analysieren sind vorrangig die kritisch bzw. aktiv gewordenen Restriktionen, besonders jene mit hoher Dualbewertung. Zusätzlich können solche Restriktionen betrachtet werden, die momentan noch nicht kritisch sind, mit hoher Wahrscheinlichkeit aber durch die Erweiterung der aktiven Fonds selbst kritisch werden.
In Tabelle 3.4 sind die Restriktionen zusammengestellt, die aus betrieblicher Sicht beeinflußbar sind. Für die drei optimalen Lösungen sind die Werte der logischen Variablen (Schlupfvariablen) ausgewiesen. Darüber hinaus sind für die Hauptzielfunktion die Zielfunktionsverbesserungen innerhalb des Stabilitätsbereiches bei isolierter Fondsveränderung einer Schranke in der Tabelle sichtbar gemacht.

Aus Tabelle 3.4 geht hervor, daß die Veränderungen innerhalb der einzelnen Stabilitätsbereiche in Bezug auf die ökonomische Forderung, die Warenproduktion um ca. 600 000 Mark zu erhöhen, völlig unzureichend sind.
Somit sind weitere Untersuchungen mittels der Anwendung der Komplexmethode durchzuführen. Zur Begrenzung der Gesamtverbesserung im Rahmen der betrieblichen Möglichkeiten ist im Optimierungsmodell die Warenproduktion zu IAP durch Einführung einer zusätzlichen Nebenbedingung auf den Vorgabewert 16,2 Mio. Mark beschränkt (Zeile OSWP-IAP).

Tabelle 3.4 Beeinflußbare Restriktionen für den ersten Komplexschritt

		ZF:WP-IAP				ZF:WP-BP	ZF:GEWINN
	RS-Werte	log. Variable	Zeilenbewertung	ob.Gültigkeitsbr.	ZF-Zuwachs	log. Variable	log. Variable
Zeilen	b_i^o[h]	y_i[h]	u_i[M/h]	b_i^o[h]	Z_{max}[M]	y_i[h]	y_i[h]
(1)	(2)	(3)	(4)	(5)	(4)•(5)	(7)	(8)
V065	32810	0	36,2	3,9	141,2	0	0
F095	45575	0	29,9	244,8	7319,5	0	0
F110	14718	0	164,0	0,1	16,4	0	0
F145	66175	0,6	0	-	-	0,6	5,0
F180	2659	0	445,2	0,8	356,2	0	18,6
F205	62250	0	73,2	8,7	636,8	0	6,1
F225	41410	2,3	0	-	-	2,3	45,7
F315	3975	0	77,8	0,4	31,1	0	3,7

3.2.2. Komplexe Veränderung bei Beibehaltung der Lösungsstruktur der Problemvariablen

Im Abschnitt 4.3. wird die rechentechnische Realisierung des ersten Komplexschrittes demonstriert. Die erforderlichen Änderungen der Problemdaten werden teils mit der Prozedur

REVISE (Fixierung der Problemvariablen, die Schrankenwerte annehmen, an der Schranke) und teils mit der Prozedur FLAGS (Umwandlung der in Tabelle 3.4 angegebenen Bedingungen, deren Fonds nicht mehr beschränkend sein sollen, in N-Typ-Zeilen) vorgenommen.

Tabelle 3.5 zeigt die erreichten Verbesserungen der einzelnen Zielfunktionswerte (oberer Wert: Gesamtwert, unterer Wert: Zuwachs).

Bezüglich der Hauptzielfunktion WP-IAP wird beim ersten Komplexschritt eine Verbesserung von 0,3 [%] erreicht. Dieser Zuwachs ist verglichen mit der ökonomischen Vorgabe noch völlig unzureichend. Deshalb führen wir einen weiteren Komplexschritt durch. Auf analoge Weise sind wieder zuerst die Fondsauslastungen zu prüfen, die Auswahl der für die Veränderung zugelassenen Aufwandsarten zu treffen und eine erneute Optimierungsrechnung durchzuführen.

Tabelle 3.5 Zielfunktionswerte für die drei Optimierungen bei Fixierung der Lösungsstruktur

ZF / Zeile	$Z_{max}^{I(1)}$ = WP-IAP [M]	$Z_{max}^{II(1)}$ = WP-BP [M]	$Z_{max}^{III(1)}$ = GEWINN [M]
WP-IAP	15 636 687 (+ 53 043)	15 636 687 (+ 53 072)	15 636 687 (+ 66 362
WP-BP	13 117 834 (+ 48 000)	13 117 834 (+ 47 937)	13 117 834 (+ 56 162)
GEWINN[1]	2 889 044 (+ 19 063)	2 889 044 (+ 18 440)	2 889 044 (+ 15 277)

[1] Siehe Fußnote von Tabelle 3.3

Für die nächsten Komplexschritte wurden jeweils zusätzlich zum vorangehenden Schritt folgende Aufwandsarten für die Veränderung zugelassen:

2. Komplexschritt: V040, F070, F275, F305

3. Komplexschritt: V010, V060, F190

4. Komplexschritt: F100

Mit dem letzten Schritt wird der Vorgabewert 16 200 000 [M] für WP-IAP erreicht. (Schranke für zusätzliche Nebenbedingung OSWP-IAP). Für die Nebenzielfunktionen WP-BP und GEWINN ergeben sich die Optimalwerte 13 586 303 [M] und 3 115 867 [M].

3.2.3. Komplexe Veränderung bei freier Struktur der Optimalbasis

Die modellmäßige Vorbereitung (Bestimmung der beeinflußbaren Restriktionen) entspricht voll der in den vorangegangenen Abschnitten 3.2.1. und 3.2.2. angegebenen Arbeitsweise. Es entfällt die Fixierung von Problemvariablen an Schrankenwerten. Die Tabelle 3.6 ist wie Tabelle 3.5 aufgebaut.

Tabelle 3.6 Zielfunktionswerte für die drei Optimierungen bei freier Basisstruktur

ZF / Zeile	$Z_{max}^{I(1)}$ = WP-IAP [M]	$Z_{max}^{II(1)}$ = WP-BP [M]	$Z_{max}^{III(1)}$ = GEWINN [M]
WP-IAP	15 644 890 (+ 61 246)	15 641 457 (+ 57 842)	15 630 320 (+ 59 995)
WP-BP	13 122 329 (+ 52 495)	13 123 540 (+ 53 643)	13 113 626 (+ 51 954)
GEWINN[1]	2 891 326 (+ 21 345)	2 891 372 (+ 20 768)	2 893 813 (+ 20 046)

[1] Siehe Fußnote von Tabelle 3.3

Die Zielfunktionswerte sind zwar besser als die entsprechenden in Tabelle 3.5, jedoch sind auch bei dieser Aufgabenstellung weitere Komplexschritte erforderlich. Zu dem jeweils vorangehenden wurden in diesem Falle folgende Aufwandsarten für die Veränderung zugelassen:

2. Komplexschritt: V040, F070, F190, F275, F305
3. Komplexschritt: V010, V060, F100

Mit dem dritten Schritt wird der Vorgabewert für WP-IAP erreicht. Die Gesamtheit der für die Veränderung zugelassenen Aufwandsarten ist die gleiche wie beim vierten Komplexschritt für feste Struktur.

3.2.4. Einschätzung der Ergebnisse

Nachdem das Anwendungsbeispiel sowohl in der komplexen Anwendungsform mit stabiler Struktur als auch in der mit freier Struktur berechnet worden ist, wollen wir die erhaltenen Ergebnisse für WP-IAP ⟶ Max! vergleichen und analysieren.

Tabelle 3.7 Vergleich der Zielfunktionswerte für stabile und freie Struktur

Name der Zielfunktion	Grundaufgabe	4. Komplexschritt-stabile Struktur		3. Komplexschritt-freie Struktur	
	[M]	[M]	[%]	[M]	[%]
WP-IAP	15 583 644	16 200 000	103,96	16 200 000	103,96
WP-BP	13 069 834	13 579 043	103,90	13 580 549	103,91
GEWINN	2 869 981	3 111 173	108,40	3 117 856	108,64

Tabelle 3.7 zeigt, daß sich die Werte der Nebenzielfunktionen WP-BP und GEWINN für stabile und freie Struktur nur geringfügig unterscheiden. Die Hauptzielfunktion WP-IAP nimmt den vorgegebenen Schrankenwert an. Die erforderlichen Fonds einschließlich ihrer prozentualen Erhöhung für die 16 zur Veränderung zugelassenen Aufwandsarten (alles Lohnstunden) sind in Tabelle 3.8 enthalten.

Tabelle 3.8 Vergleich von Restriktionen bei stabiler und freier Struktur

Name der Aufwandsart	Grundaufgabe	4. Komplexschritt-stabile Struktur		3. Komplexschritt-freie Struktur	
	[h]	[h]	[%]	[h]	[%]
V010	9 500	9 751	102,64	9 751	102,64
V040	7 600	7 815	102,83	7 815	102,83
V060	7 525	7 749	102,98	7 750	102,99
V065	32 810	33 946	103,46	33 968	103,53
F070	50 750	53 158	104,74	53 253	104,93
F095	45 575	48 027	105,38	48 109	105,56
F100	14 718	15 086	102,50	15 092	102,54
F110	14 718	15 788	107,27	15 786	107,26
F145	66 175	70 638	106,74	70 676	106,80
F180	2 659	2 824	106,21	2 824	106,21
F190	44 995	46 783	103,97	46 725	103,84
F205	62 250	65 639	105,44	65 577	105,34
F225	41 410	42 722	103,17	42 729	103,19
F275	3 875	4 021	103,77	4 023	103,82
F305	9 875	10 165	102,94	10 202	103,31
F315	3 975	4 037	101,56	4 066	102,29
Summe	418 410	438 149		438 346	
Ø	26 151	27 384	104,71	27 397	104,76

Tabelle 3.8 läßt erkennen, daß nur bei 40 % der Restriktionen eine durchschnittliche Fondserweiterung um 4,7 [%] erforderlich ist, um einen Gesamtoptimierungseffekt von 4 [%] zu erzielen. Die erforderlichen Fondserhöhungen sind in beiden Fällen etwa annähernd gleich. Bei freier Struktur werden 297 [h] mehr benötigt. Dem gegenüber steht eine Erhöhung von WP-BP um 1506 [M] und GEWINN um 6 683 [M].

Tabelle 3.9 zeigt die Entwicklung der Erzeugnismengen (Problemvariablen) bei der komplexen Anwendung der Optimierung für den Fall WP-IAP ⟶ Max! . Wesentliche Unterschiede bestehen nur bezüglich der Erzeugnisse 050 und 060, die sich jeweils um 20 Mengeneinheiten unterscheiden. Weiter ist zu erkennen, daß die Variablen 050 und 090 unterschiedliche Schrankenwerte annehmen (warum?).

Tabelle 3.9 Vergleich der optimalen Erzeugnisstückzahlen

Name der Problem-variablen	Absatzschranken		Lösung Grund-aufgabe	Lösung 4.K.schritt stab.Strukt.	Lösung 3.K.schritt freie Stru.
	untere	obere			
010	490	510	510	510	510
020	300	320	320	320	320
030	10	10	10	10	10
040	90	90	90	90	90
050	1000	1020	1020	1020	1000
060	200	500	250,06	344,31	364,19
070	18	20	20	20	20
080	34,4	35	34,43	35	35
090	5,2	5,5	5,2	5,2	5,5
100	120	140	120,54	120	120
110	1100	1235	1233,60	1235	1235
120	200	300	211,51	300	300
130	900	990	990	990	990
140	4,4	6	6	6	6
150	72	90	89,81	88,19	87,12
160	0,5	0,5	0,5	0,5	0,5

Die durchgeführten Berechnungen bestätigen die im Abschnitt 1.4.3 gegebenen drei Aussagen. Zusätzlich kann gesagt werden:

1. Bei freier Struktur der Optimalbasis tritt im allgemeinen bedingt durch die größere Optimierungsfreiheit ein schnelleres Anwachsen der Zielfunktion ein. Um einen gleichen Leistungsanstieg zu erreichen, können weniger Komplexschritte erforderlich sein.

2. Bei Festhalten der Struktur der Optimalbasis tritt im allgemeinen bedingt durch die Einschränkung der Variabilität eine schnellere Indifferenzentwicklung ein. Die Indifferenzentwicklung kann jedoch vorübergehend gestört werden, sobald eine inaktive Bedingung hervorgerufen

durch die Dehnung des zulässigen Bereiches zu einer aktiven, d.h. zum Engpaß wird.

3.3. Umsetzung der Ergebnisse im Betrieb

Der VEB Excelsiorwerk Leipzig wendet seit 1968 kontinuierlich die Produktionsplanoptimierung an. Von der Werkleitung wird eingeschätzt, daß auch heute immer noch die optimalen Ergebnisse der Grundaufgabe der Produktionsplanoptimierung um etwa 4 [%] besser sind als die eines manuell erstellten Planes. Dieser Prozentsatz ist beachtlich, da der Betrieb über Jahre eine relativ stabile Produktionsstruktur aufweist.

Durch die Anwendung der Komplexmethode wird ein zusätzlicher Effekt erreicht. Gegenüber der traditionellen Produktionsplanoptimierung (Lösung der Grundaufgabe) konnte der Optimierungseffekt etwa verdoppelt werden. Theoretisch wird folgender Gesamteffekt erzielt (Optimierungseffekt aus der Lösung der Grundaufgabe + Effekt aus der Anwendung der Komplexmethode):

- Steigerung der Warenproduktion zu IAP um 8 [%] = 1230 [TM]
- Steigerung der Warenproduktion zu BP um 8 [%] = 1020 [TM]
- Steigerung des Betriebsergebnisses um 12 [%] = 370 [TM]

Erfahrungsgemäß können von dem berechneten Optimierungseffekt infolge notwendiger Vereinfachungen im mathematischen Modell (Abstraktionsfehler) etwa nur 60 [%] realisiert werden. Die Effekte aus der komplexen Produktionsplanoptimierung betragen somit:

- Steigerung der Warenproduktion zu IAP um 4,8 [%] = 738 [TM]
- Steigerung der Warenproduktion zu BP um 4,8 [%] = 612 [TM]
- Steigerung des Betriebsergebnisses um 7,2 [%] = 222 [TM]

Dem gegenüber stehen für die erforderlichen Optimierungsrechnungen EDVA-Kosten unter 1000 [M]. Zusätzlich lassen sich aus der Anwendung der Optimierung wichtige Erkenntnisse für die Fondsentwicklung und Investitionspolitik der Folgejahre ableiten.

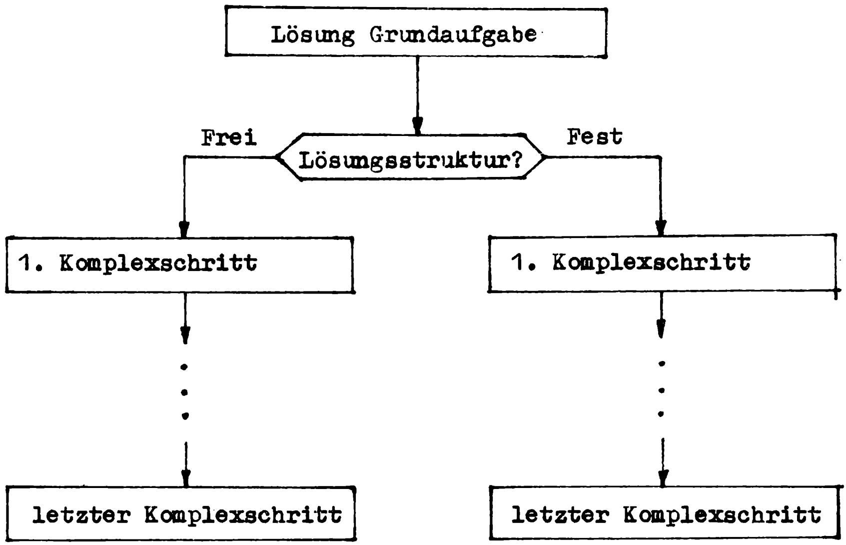
F3/1	Geben Sie zu den gewählten Optimalitätskriterien weitere Zielkriterien an, die bei einer Produktionsplanoptimierung sinnvoll verwendet werden können!

F3/2	Führen Sie eine Auswertung gemäß Tabellen 3,4, 3.7 und 3.8 für den ersten Komplexschritt mit fester Struktur durch! Welche Veränderungen würden Sie für den nächsten Komplexschritt vorschlagen?

4. Lösung der ausgewählten Anwendungsbeispiele mit dem PS OPSI

4.1. Erläuterungen zur Durchführung der Rechnungen

Die komplexe Anwendung der Optimierung erfordert eine wiederholte (oder mehrstufige) OPSI-Anwendung, da erst aus der manuellen Ergebnisauswertung sich die zu treffenden Modellveränderungen ableiten lassen. Der Ablauf kann wie folgt skizziert werden (jeder Schritt ein Job):



Der Ablauf ist zugunsten der Übersichtlichkeit vereinfacht dargestellt und entspricht dem ebenfalls vereinfachten Demonstrationsbeispiel. Der praktische Planungsablauf ist komplizierter. In der Regel wird zu Beginn der Planerarbeitung aufgrund der zu diesem Zeitpunkt größeren Freiheitsgrade mit der freien Basisstruktur begonnen. Mit dem Fortschreiten der Vorbereitungen zur Plandurchführung sind stärkere Fixierungen notwendig. Das hat zur Folge, daß man bei späteren Komplexschritten am Ende des Basisjahres und vor allem während des Planjahres von der freien über die gemischte zur festen Basisstruktur übergehen muß.

In den Abschnitten 4.2, 4.3 und 4.4 sind ausgewählte Teile der Rechnerprotokolle für drei Jobs wiedergegeben. Sie umfassen Jobsteueranweisungen, OPSI-Programme, Eingabe- und Änderungsdaten sowie Ergebnisausgaben. Die angegebene Jobsteuerung kann als Muster für ähnlich geartete OPSI-Anwendungen angesehen werden. Mit den angegebenen Jobsteueranweisungen können Probleme mit bis zu ca. 1000 Zeilen und 1000

Spalten bei höchstens 3 [%] Besetzung gerechnet werden. Es wird nur mit Plattendateien gearbeitet. Soll das ursprüngliche Problem, das mit dem Job GRDAUFG eingelesen und von den nachfolgenden Jobs wieder benötigt wird, auf Magnetband (Magnetbandspeicher 5010) sichergestellt werden, so sind folgende Jobsteuerkarten abzuändern:

Job GRDAUFG: Anweisung //PBFILE... ist zu ersetzen durch

```
    //PBFILE DD UNIT=5010,VOL=SER=datenträgerarchnr.,
               LABEL=(1,SL),DSN=dateiname,DISP=(NEW,KEEP)
```

Job KOMPLEX1: Anweisung //PBFILE ... ist zu ersetzen durch

```
    //PBFILE  DD UNIT=5010,VOL=SER=datenträgerarchnr.,
                LABEL=(1,SL),DSN=dateiname,DISP=(OLD,KEEP)
```

Job KOMPLEX2: Anweisung //PBFILE... ist zu ersetzen durch

```
    //PBFILE DD UNIT=5010,VOL=SER=datenträgerarchnr.,
               LABEL=(1,SL),DSN=dateiname,DISP=(OLD,KEEP)
```

datenträgerarchnr. ist dem Magnetband vom Rechenzentrum zugeordnet und somit für alle Jobs die gleiche Nummer. dateiname kann vom Anwender frei gewählt werden, muß aber für alle Jobs gleich sein (maximal 8 Zeichen lang).

Der im Abschnitt 4.2 wiedergegebene Job GRDAUFG beinhaltet das Einlesen der Problemdaten und die nacheinanderfolgende Optimierung nach den drei Zielfunktionen. Auf dem Problemfile werden die Daten unter dem Namen EXCELS abgespeichert, zusätzlich noch die optimale Lösung für WP-IAP ⟶ Max! . Diese Daten sind als Stammdaten zu betrachten und werden in den Jobs KOMPLEX1 als Altproblem und KOMPLEX2 als Problem benötigt.
Zum Job GRDAUFG sind neben der Jobsteuerung und dem OPSI-Programm nur folgende Ergebnisprotokolle wiedergegeben:

BCDOUT-Ausdruck	Seiten 67, 68, 69
SOLUTION-Ausdruck für WP-IAP ⟶ Max!	Seite 70
RANGE-Ausdruck für WP-IAP ⟶ Max!	Seite 71

Die SOLUTION-Ausgabe wurde für die Sektion 1 durch Auswahllisten auf die interessierenden Zeilen eingeschränkt.

Der Job KOMPLEX1 (Ausschnitte wiedergegeben im Abschnitt 4.3) beinhaltet den 1. Komplexschritt für stabile Basisstruktur. Die vom Job GRDAUFG erstellte Problemdatei EXCELS wird mittels REVISE geändert (Festsetzen bestimmter Problemvariablen an den Schranken - welche?). Mittels FLAGS werden die in der Auswahlliste aufgeführten Aufwandsarten (Zeilen) freigesetzt. Danach erfolgt die Optimierung für die drei Zielfunktionen. Es werden neben der Jobsteuerung und dem OPSI-Programm nur folgende Ergebnisprotokolle wiedergegeben:

REVISE-Ausdruck (enthält die Änderungsdaten)		Seite 73
SOLUTION-Ausdruck für WP-IAP $\longrightarrow$	Max!	Seiten 73, 74
RANGE-Ausdruck für WP-IAP $\longrightarrow$	Max!	Seite 74

Weitere Komplexschritte für stabile Struktur erfordern nur eine Erweiterung der Auswahlliste zur OPSI-Prozedur FLAGS.

Der Job KOMPLEX2 (Ausschnitte wiedergegeben im Abschnitt 4.4) beinhaltet den 1. Komplexschritt für freie Basisstruktur. Die vom Job GRDAUFG erstellte Problemdatei wird im Gegensatz zum Job KOMPLEX1 ohne Änderung mittels REVISE verarbeitet. Mittels FLAGS werden jedoch ebenfalls die in der Auswahlliste angegebenen Aufwandsarten (Zeilen) freigesetzt. Danach erfolgt wieder die Optimierung für die drei Zielfunktionen. Es wird nur die Jobsteuerung und das OPSI-Programm angegeben. Weitere Komplexschritte für freie Basisstruktur erfordern ebenfalls nur eine Erweiterung der Auswahlliste zur OPSI-Prozedur FLAGS.

Hinweis: Die Beschränkung des Anwachsens der Zielfunktion WP-IAP auf 16 200 000 [M] erfolgte bereits bei der Dateneingabe zur Grundaufgabe.

Ergänzende Erläuterungen zu einigen Tabellenköpfen der Ergebnisausgaben

SOLUTION

ST — Status der Restriktion für die ausgegebene Lösung

Bei ZEILEN:
- BS Zeilenaktivität an keiner Schranke (logische Variable ist Basisvariable)
- OS Zeilenaktivität an der Schranke OS (logische Variable ist Nichtbasisvariable)

Bei SPALTEN:
- BS Problemvariable ist Basisvariable
- OS Problemvariable ist Nichtbasisvariable mit Schrankenwert OS
- US Problemvariable ist Nichtbasisvariable mit Schrankenwert US
- FX Problemvariable ist Nichtbasisvariable mit Schrankenwert US=OS

ZEILENAKTIVITAET — $\sum_{j=1}^{n} a_{ij}x_j$

ZEILENBEWERTUNG — Duale Variable zur logischen; Werte nur vorhanden für Zeilen mit Status OS (Ausnahme: Zielfunktion)

SPALTENBEWERTUNG — Duale Variable zur Problemvariablen; Werte nur vorhanden für Spalten mit Status OS, US und FX

LOG.VARIABLE WERT — Berechnet sich aus Schrankenwert - Zeilenaktivität. Bei der Zeile GEWINN gilt analog

$$\text{Log.Var.} = -c_0 - \sum_{j=1}^{n} c_j x_j = 3\,340\,000 - \text{Zeilenaktivität}$$

Also ist GEWINN = -(log. Variable).

RANGE (nur SEKTION 1)

Zu jedem Zeilennamen erfolgt eine zweizeilige Ausgabe.

STABILITAETSBEREICH	Obere/untere Zeile: Die Lösung ist stabil für den STABILITAETSBEREICH (Wert ob. Zeile) = AKTIVITAET = (Wert unt. Zeile).
BESCHRKT DURCH	Wird die von der Lösung angenommene Schranke unter/über die angegebene Stabilitätsbereichsgrenze hinaus verkleinert/vergrößert, so wird die unter BESCHRKT DURCH angegebene Zeile oder Spalte Nichtbasisvariable an der
WIRD	unter WIRD angegebenen Schranke.
ZFWERTAEND/EINHEIT	Die ZielFunktionsWERTAENDerung pro EINHEIT der Schrankenänderung ist der angegebene Wert. Bei Veränderung der Schranke bis zur Stabilitätsbereichsgrenze oder darüber hinaus ergibt sich für die
ZFWERTAENDERUNG	ZielFunktionsWERTAENDERUNG der angegebene Gesamtwert.

F4/1 Welche Veränderungen sind im Jobstrom (JOB-Anweisung, Jobsteueranweisungen, OPSI-Programm, Änderungsdaten) für die Komplexschritte 2 bis 4 vorzunehmen?
(Studieren Sie bei der Beantwortung der Frage auch die Abschnitte 2.4.3 und 3.2)

Literaturverzeichnis

Anwenderdokumentation des VEB Robotron - Zentrum für Forschung und Technik, 8012 Dresden, PSF 330:

[1]	Anwendungsbeschreibung	H 4013-2002
[2]	Anleitung für den Systemverantwortlichen	H 4013-2003
[3]	Anleitung für den Programmierer	H 4013-2004
[4]	Sprachbeschreibung	H 4013-2007
[5]	Nachrichten	H 4013-2006

Literatur:

[6] DÜCK, W.; KÖHRT, H.; RUNGE, W.; WUNDERLICH, L.: Mathematik für Ökonomen, Band 2 Verlag Die Wirtschaft, Berlin 1980

[7] LASSMANN, W.; KUMMEROW, E.: Produktionsplanoptimierung Verlag Die Wirtschaft, Berlin 1977

[8] Autorenkollektiv: Komplexe Anwendung der linearen Optimierung rechentechnik/datenverarbeitung, 15(1978) Beiheft 1

[9] DREWEL, H.; LASSMANN, W.: Lehrbriefreihe Mathematische Methoden und Modelle 1. Lehrbrief - Produktionsplanoptimierung Dresden 1976

Antworten zu den Fragen

A 2.2/1 a) Namen (frei gewählt) siehe b)

b) Tabelle

		PROD1	PROD2	PROD3	KAP1
N	GEWINN	1	3	2	0
O	N1	0,27	0	1,15	112,5
U	N2	4,6	1,73	3,12	240
G	N3	1	0	-2	0
	MENGE	0 45	12 40	0	

c) Lochbeleg

```
NAME          DATEN
ZEILEN
 N  GEWINN
 O  N1
 U  N2
 G  N3
SPALTEN
    PROD1     GEWINN      1.Ø          N1            .27
    PROD1     N2          4.6          N3           1.Ø
    PROD2     GEWINN      3.Ø          N2           1.73
    PROD3     GEWINN      2.Ø          N1           1.15
    PROD3     N2          3.12         N3          -2.Ø
RSN
    KAP1      N1        112.5          N2         240.Ø
SVBE
 OS MENGE     PROD1      45.Ø
 US MENGE     PROD2      12.Ø
 OS MENGE     PROD2      4Ø.Ø
DEND
```

A 2.2/2 Lochbeleg

```
NAME          REVDAT
ZEILEN
  LOESCHE
    N3
  MACH        N3
 O  N4
SPALTEN
  AENDERE
    PROD1     N4          2.45
    PROD2     N4          5.6
    PROD3     N4          1.3
RSN
  AENDERE
    KAP1      N4        161.5
SVBE
  AENDERE
 US MENGE     PROD2      12.Ø
 OS MENGE     PROD2      46.Ø
DEND
```

A 2.4/1 a) OPSI-Programm

```
         PROGRAM        ;* A2.4/1 A)
         INITIAL
         MOVE(XDNAME,'DATEN')
         MOVE(XPBNAME,'BEISP1')
         CONVERT
         SETUP('SVB','MENGE','MAX')
         MOVE(XZF,'GEWINN')
         MOVE(XRS,'KAP1')
         PRIMAL
         SOLUTION
         SAVE('NAME','LOESG1')
         EXIT
         PEND
```

b) OPSI-Programm

```
PROGRAM        ;* A2.4/1 B)
INITIAL
MOVE(XDNAME,'REVDAT')
MOVE(XAPBNAME,'BEISP1')
MOVE(XPBNAME,'BEISP2')
REVISE
SETUP('SVB','MENGE','MAX')
MOVE(XZF,'GEWINN')
MOVE(XRS,'KAP1')
RESTORE('NAME','LOESG1')
PRIMAL
SOLUTION
SETUP('MAX')
RESTORE('NAME','LOESG1')
PRIMAL
SOLUTION
EXIT
PEND
```

A 2.4/2 Da das geänderte Problem das bestehende mit gleichem Namen überspeichert, steht das ursprüngliche nicht mehr zur Verfügung.
Beachte: Auch alle nachfolgenden Probleme sind dann nicht mehr verfügbar!

A 3/1

- Maximierung Fondsauslastung bzw. Minimierung der nicht genutzten Fonds
- Minimierung Materialeinsatz
- Minimierung zentrale Fertigung
- Minimierung der Selbstkosten bei vorgegebener Warenproduktion
- Maximierung Devisenerlöse
- Maximierung der betriebsbezogenen Exportrentabilität (Quotientenzielfunktion Devisenerlös/Warenproduktion)
- Maximierung des Quotienten Warenproduktion/Materialeinsatz

A 3/2 Die Auswertung der Lösungsvorschläge erfolgt im Seminar.

A 4/1 Nur Angabe der Änderungen für den 2. Komplexschritt:
Jobsteuerung: keine
OPSI-Programm: Neue FLAGS-Anweisung:

```
FLAGS('ZMASKEN','VØ4Ø','VØ65','FØ7Ø','FØ95',     *
'F11Ø','F145','F18Ø','F2Ø5','F225','F3Ø5','F315',' ')
```

Druck:

ZENTRALSTELLE FÜR LEHR- UND ORGANISATIONSMITTEL DES MINISTERIUMS FÜR HOCH- UND FACHSCHULWESEN, ZWICKAU

Ag 628/430/80/DDR/1100-ZLO 739/80

```
//GRDAUFG JOB   (01700494,F4254),SCHLOMS,MSGLEVEL=(2,0),
//              REGION=100K,MSGCLASS=S,CLASS=M
//JOBLIB    DD  DSN=OP.LAD,DISP=SHR
//OPCOMP EXEC   PGM=OP#COMP
//COMP1     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//COMP2     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//COMP3     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//COMP4     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//SYSPROG   DD  UNIT=SYSDA,SPACE=(400,(30,30)),DISP=(NEW,PASS)
//SYSPRINT  DD  SYSOUT=(S,N)
//SYSIN     DD  *

PS OPSI II  0.0         COMPILIERSCHRITT - OPSI-PROGRAMM.

           0001                    PROGRAM                      ;***
           0003                    INITIAL
           0070                    TITLE('DATENEINGABE')
           0073                    MOVE(XPBNAME,'EXCELS')
           0075                    MOVE(XDNAME,'GRDAUF')
           0077                    CONVERT
           0079                    BCDOUT
           0080                    TITLE('OPTIMIERUNG GRUNDAUFGA
           0081                    SETUP('SVB','VAR.1','MAX')
           0083                    MOVE(XRS,'75.1')
           0084                    MOVE(XZF,'WP-IAP')
           0086                    PRIMAL
           0088                    SAVE('NAME','GRDAUFG')
           0090                    SOLUTION('ZMASKEN','V010','V0
           0090             'F100','F110','F145','F180','F190','F2(
           0090             'WP-IAP','WP-BP','GEWINN',' ','SMASKEN
           0091                    RANGE('SKT2','/','SKT3','/','
           0092                    MOVE(XZF,'WP-BP')
           0094                    PRIMAL
           0096                    SOLUTION('ZMASKEN','V010','V0
           0096             'F100','F110','F145','F180','F190','F2(
           0096             'WP-IAP','WP-BP','GEWINN',' ','SMASKEN'
           0097                    RANGE('SKT2','/','SKT3','/','S
           0098                    MOVE(XZF,'GEWINN')
           0100                    PRIMAL
           0102                    SOLUTION('ZMASKEN','V010','V04
           0102             'F100','F110','F145','F180','F190','F20
           0102             'WP-IAP','WP-BP','GEWINN',' ','SMASKEN'
           0103                    RANGE('SKT2','/','SKT3','/','S
           0104                    EXIT
           0105                    PEND

//OPEXEC EXEC   PGM=OP#EXEC,COND=(0,NE,OPCOMP)
//SYSPROG   DD  DSN=*.OPCOMP.SYSPROG,DISP=(OLD,DELETE)
//PBFILE    DD  UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND),DIS
//FILE1     DD  UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//FILE2     DD  UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//MATRIX1   DD  UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//ETA1      DD  UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//SYSPRINT  DD  SYSOUT=(S,N)
//SYSIN     DD  *
```

B e i l a g e zum Lehrbrief Nr. 02 4602 08 0 - Mathematische Methoden und Modelle - (Seiten 1 - 10)

4.2. Rechnerprotokolle zur Grundaufgabe

```
                                   16.07.80   22.10    SEITE

 G R U N D A U F G A B E ***

                 ;*DATEN-
                 ;* EIN-
                 ;*  GABE

E')
                 ;*VORBEREITUNG OPT.

                 ;*OPTIMIERUNG
                 ;* FUER
                 ;*  ZF = WP-IAP
0','V060','V065','F070','F095', *
5','F225','F275','F305','F315', *
,'***',' ')
KT4','/','SKT5','/')
                 ;*OPTIMIERUNG
                 ;*FUER ZF = WP-RP
0','V060','V065','F070','F095', *
5','F225','F275','F305','F315', *
,'***',' ')
KT4','/','SKT5','/')
                 ;*OPTIMIERUNG
                 ;*FUER ZF = GEWINN
0','V060','V065','F070','F095', *
5','F225','F275','F305','F315', *
,'***',' ')
KT4','/','SKT5','/')



P=(NEW,KEEP)
```

2 PS OPSI II 0.0 - DATENEINGABE

SCDOUT AUSGABE DES PROBLEMS EXCELS MIT DNAME GRDAUF
ZEIT = 0.07 MIN.

DNAME GRDAUF

ZEILEN

O	V010	O	V020	O	V030	O	V040	O
O	F080	O	F090	O	F095	O	F100	O
O	F145	O	F150	O	F160	O	F165	O
O	F205	O	F210	O	F220	O	F225	O
O	F265	O	F270	O	F275	O	F280	O
O	F315	N	KAP.GES	N	GRDLOHN	N	GRDMAT	N
O	OSWP-IAP							

SPALTEN

010	V010	2.12000	V020	1.20000
010	V050	,93000	V060	1,28000
010	F095	7,40880	F265	62,78200
010	F260	10,55000	F270	,33000
010	F280	,89000	V065	6.18090
010	GRDMAT	885,00000	BLECHO,8	,36000
010	GEWINN	932,29000	OSWP-IAP	2303,00000
020	V010	1,87000	V020	1.20000
020	V050	,93000	V060	1,28000
020	F095	6,58650	F265	71,41040
020	F260	10,55000	F270	,38000
020	F280	1,57000	V065	6,11450
020	GRDMAT	806,00000	BLECHO,8	,43000
020	GEWINN	1113,42000	OSWP-IAP	2726,00000
030	V010	1,68000	V020	1,20000
030	V050	,93000	V060	1,28000
030	F095	1,83290	F265	63,99140
030	F260	10,55000	F275	,33520
030	F280	,77000	KAP.GES	74,48710
030	BLECHO,8	,20000	WP-IAP	2700,00000
030	F270	,22000	OSWP-IAP	2700,00000
040	V010	2,04000	V020	1,60000
040	V050	,93000	V060	1,28000
040	F095	6,52470	F265	69,28370
040	F260	10,55000	F270	,38000
040	F280	,77000	V065	6.58530
040	GRDMAT	1528,00000	BLECHO,8	,43000
040	GEWINN	1068,77000	OSWP-IAP	2756,00000
050	V010	1,99000	V020	1,20000
050	V060	1,32000	F070	11,40000
050	F100	10,11000	F110	11,35000
050	F170	1,85000	F190	13.82000
050	F205	17,88920	F225	6,18940
050	F275	,80730	F305	1,50950
050	KAP.GES	88,64160	GRDLOHN	288,07000
050	WP-IAP	3699,00000	WP-BP	3023,00000
060	V010	2,06000	V020	1,20000
060	V060	1,32000	F070	17,08000
060	F100	10,11000	F110	11,35000
060	F170	1,85000	F190	13,82000
060	F205	17,88920	F225	6,18940

V050	O	V060	O	V065	O	F070	
F110	O	F120	O	F130	O	F140	
F170	O	F180	O	F190	O	F200	
F230	O	F240	O	F250	O	F260	
F290	O	F300	O	F305	O	F310	
BLECH0.8	N	WP-IAP	N	WP-BP	N	GE-INN	

V030	.48000	V040	2.30000
F070	8.38000	F230	17.49000
F240	29.31000	F250	8.58000
F275	.50280	F305	1.28320
KAP.GES	78.15770	GRDLOHN	255.71000
WP-IAP	2303.00000	WP-BP	2073.00000
V030	.48000	V040	2.45000
F070	7.45000	F230	19.16000
F240	37.04000	F250	8.27000
F275	.57900	F305	2.26360
KAP.GES	86.95400	GRDLOHN	282.58000
WP-IAP	2726.00000	WP-BP	2202.00000
V030	2.13000	V040	2.64000
F070	2.08000	F230	18.37000
F240	29.93000	F250	8.27000
F305	1.11020	V065	7.21140
GRDLOHN	244.15000	GRDMAT	716.05000
WP-BP	2144.00000	GEWINN	1183.80000
V030	.48000	V040	2.45000
F070	7.38000	F230	17.73000
F240	36.38000	F250	8.27000
F275	.57900	F305	1.11020
KAP.GES	84.08290	GRDLOHN	275.23000
WP-IAP	2756.00000	WP-BP	2872.00000
V040	2.43000	V050	.60000
F095	10.07870	F145	46.48630
F120	14.37000	F130	15.38000
F200	.08000	F210	6.97000
F270	.53000	F280	1.03000
F290	.02000	V065	5.68100
GRDMAT	1333.00000	BLECH0.8	.95000
GEWINN	1201.93000	OSWP-IAP	3699.00000
V040	2.43000	V050	.60000
F095	15.10040	F145	46.48630
F120	14.37000	F130	15.38000
F200	.08000	F210	6.97000
F270	.53000	F280	1.03000

PS OPSI II 0.0 DATENEINGABE

060	F275	.80750	F305	1,50950
060	KAP.GES	93,70980	GRDLOHN	305,61000
060	WP-IAP	3830,00000	WP-BP	3210,00000
070	V010	2,11000	V020	1,20000
070	V060	1,31000	F070	14,26000
070	F100	11,71000	F110	11,03000
070	F170	2,38000	F190	17,40000
070	F210	6,97000	F270	,73000
070	F280	1,32000	V065	6,03030
070	GRDMAT	1485,00000	BLECHO,8	1,60000
070	GEWINN	1561,81000	OSWP-IAP	4974,00000
080	V010	4,33000	V020	34,57000
080	V050	13,13000	V060	41,64000
080	F095	30,37000	F145	144,74660
080	F130	78,87000	F140	55,58000
080	F200	25,80000	F220	56,72000
080	F270	1,68000	F280	19,59000
080	F290	16,15000	F300	,93000
080	F315	97,91600	KAP,GES	505,88960
080	WP-IAP	14617,00000	WP-BP	13969,00000
090	V010	3,05000	V020	61,02000
090	V050	10,30000	V060	10,06000
090	F090	2,13000	F100	24,56000
090	F120	5,43000	F130	78,87000
090	F190	17,65000	F200	13,37000
090	F210	8,35000	F220	20,80000
090	F275	9,33970	F305	129,94560
090	F310	104,88000	V065	91,07440
090	GRDLOHN	1748,05000	GRDMAT	5168,00000
090	GEWINN	6984,95000	OSWP-IAP	12847,00000
100	V010	1,66000	V020	,14000
100	V050	1,33000	V060	1,17000
100	F095	5,89690	F165	27,47430
100	F190	16,82000	F210	2,87000
100	F270	,33000	F280	1,01000
100	F290	,31000	V065	5,95130
100	GRDMAT	764,00000	BLECHO,8	,36000
100	GEWINN	912,46000	OSWP-IAP	2024,00000
110	V010	1,57000	V020	,96000
110	V050	1,95000	V060	1,16000
110	F095	6,73680	F165	23,86520
110	F180	1,84000	F190	13,20000
110	F210	8,81000	F270	,53000
110	F280	,97000	F290	,02000
110	F315	,06540	KAP,GES	66,86140
110	BLECHO,8	,95000	WP-IAP	2714,00000
110	OSWP-IAP	2714,00000		
120	V010	1,64000	V020	,09000
120	V050	1,95000	V060	1,16000
120	F095	11,76740	F165	23,86520
120	F180	1,84000	F190	13,20000
120	F270	,53000	F275	,80750
120	F290	,02000	F310	,07000
120	KAP,GES	71,93860	GRDLOHN	230,07000
120	WP-IAP	2806,00000	WP-BP	2326,00000
120	OSWP-IAP	2806,00000		

3

F290	,02000	V065	5,72750
GRDMAT	1373,00000	BLECH0.8	,95000
GEWINN	1531,39000	OSWP-IAP	3830,00000
V040	2,85000	V050	,60000
F095	12,60730	F145	50,28760
F120	17,40000	F130	15,35000
F205	22,48120	F225	6,18940
F275	1,11220	F305	1,90320
KAP.GES	100,61120	GRDLOHN	325,19000
WP-IAP	4974,00000	WP-BP	3572,00000
V030	21,03000	V040	4,26000
F070	17,36000	F080	14,08000
F100	24,54000	F120	5,43000
F170	3,82000	F190	1,28000
F205	31,34010	F225	50,29930
F275	2,55960	F305	49,25780
F310	104,88000	V065	99,41320
GRDLOHN	1664,29000	GRDMAT	5191,00000
GEWINN	7113,71000	OSWP-IAP	14417,00000
V030	23,71000	V040	3,27000
F070	17,36000	F080	8,33000
F095	26,51300	F145	144,76250
F140	55,58000	F170	1,73000
F205	35,26440	F225	25,86020
F270	6,13000	F280	53,95000
F290	41,67000	F300	,93000
F315	97,91600	KAP.GES	560,67580
WP-IAP	12847,00000	WP-BP	13901,00000
V030	2,60000	V040	1,48000
F070	6,67000	F150	8,40000
F160	23,08000	F170	2,80000
F205	22,29920	F225	2,54860
F275	,50280	F305	1,83480
KAP.GES	66,50790	GRDLOHN	215,54000
WP-IAP	2024,00000	WP-BP	1892,00000
V030	2,30000	V040	,72000
F070	7,62000	F150	9,19000
F160	18,40000	F170	2,60000
F205	20,05900	F225	7,82330
F275	,80750	F305	1,42290
F310	,07000	V065	6,08130
GRDLOHN	218,50000	GRDMAT	983,00000
WP-BP	2184,00000	GEWINN	1182,50000
V030	2,30000	V040	,72000
F070	13,31000	F150	9,19000
F160	18,40000	F170	2,60000
F205	20,05900	F210	8,81000
F305	1,42290	F280	,97000
V065	6,12790	F315	,06540
GRDMAT	1023,00000	BLECH0.8	,95000
GEWINN	1066,93000	F225	7,82330

4 PS OPSI II 0.0 DATENEINGABE

130	V010	2,08000	V030	2.30000
130	V060	1,16000	F070	11.30000
130	F150	9,19000	F160	18.40000
130	F210	10,88000	F220	11.78000
130	F275	,91420	F305	1,58600
130	GRDMAT	951,00000	BLECHO.8	.95000
130	GEWINN	1386,23000	OSWP-IAP	3130.00000
140	V010	3.96000	V020	8.17000
140	V050	3,73000	V060	1,31000
140	F095	20,49340	F145	46.66630
140	F130	15,35000	F170	5.75000
140	F205	29,98340	F225	6.18940
140	F275	,80750	F305	1.15340
140	KAP.GES	121,85960	GRDLOHN	401.90000
140	WP-IAP	9600,00000	WP-BP	10150.00000
150	V010	2,35000	V020	.23000
150	V050	,77000	V060	1,61000
150	F095	22,52890	F205	65.46820
150	F240	24,82000	F250	12.53000
150	F265	54,78680	F275	.57900
150	F305	1,58600	KAP.GES	150.04130
150	BLECHO.8	,65500	WP-IAP	5650.00000
150	OSWP-IAP	5650,00000		
160	V010	4,24000	V020	12.42000
160	V050	5,13000	V060	5,62000
160	F095	88,71920	F145	224.02000
160	F120	34,80000	F130	158,23000
160	F205	165,01880	F275	1,82830
160	F300	1,00000	V065	29,30860
160	GRDLOHN	1868,49000	GRDMAT	2388,00000
160	GEWINN	6168,51000	OSWP-IAP	13000,00000

RSN

75.1	V010	9500,00000	V020	6000,00000
75.1	V050	7400,00000	V060	7525.00000
75.1	F080	1000,00000	F090	1000.00000
75.1	F110	14718,00000	F120	20275,00000
75.1	F145	66175,00000	F150	27380,00000
75.1	F170	8975,00000	F180	2659,00000
75.1	F205	62250,00000	F210	35480,00000
75.1	F230	18750.00000	F240	36500.00000
75.1	F265	66682,00000	F270	3226.00000
75.1	F290	1000,00000	F300	1000,00000
75.1	F315	3975,00000	GEWINN	3340000.000

SVBE

VAR.1	010	US	490.00000	010	OS	510,00000
VAR.1	030	FX	10.00000	040	FX	90,00000
VAR.1	060	US	200.00000	060	OS	500,00000
VAR.1	080	US	34.40000	080	OS	35,00000
VAR.1	100	US	120.00000	100	OS	140,00000
VAR.1	120	US	200.00000	120	OS	300,00000
VAR.1	140	US	4.40000	140	OS	6,00000
VAR.1	160	FX	.50000			

DEND

V040	.66000	V050	1.95000
V065	5.76330	F095	9.99030
F105	23.86520	F225	20.10790
F270	.60000	F280	1.10000
KAP.GES	62.22690	GRDLOHN	217.77000
WP-IAP	3130.00000	WP-BP	2555.00000

V030	.17000	V040	4.17000
F070	23.18000	F100	10.12000
F110	11.57000	F120	14.37000
F190	13.95000	F200	7.67000
F210	6.97000	F270	.53000
F280	.80000	V065	16.56620
GRDMAT	5513.00000	BLECH0.8	.95000
GEWINN	4235.10000	OSWP-IAP	9600.00000
V030	.68000	V040	1.22000
F070	20.17000	F080	4.40000
F190	57.60000	F230	10.50000
F260	10.38000	F270	.38000
F280	1.10000	V065	5.09240
GRDLOHN	498.03000	GRDMAT	3242.24000
WP-BP	5650.00000	GEWINN	1709.73000

V030	1.67000	V040	8.20000
F070	28.90000	F090	59.18000
F100	22.24000	F110	25.70000
F170	3.92000	F200	162.17000
F270	1.20000	F280	78.00000
F305	113.82480	KAP.GES	622.71970
WP-IAP	13000.00000	WP-BP	10425.00000

V030	8000.00000	V040	7600.00000
V065	32810.00000	F070	50750.00000
F095	45575.00000	F100	14718.00000
F130	27480.00000	F140	2525.00000
F160	50475.00000	F165	63950.00000
F190	44995.00000	F200	2525.00000
F220	15050.00000	F225	41410.00000
F250	9975.00000	F260	15500.00000
F275	3875.00000	F280	7775.00000
F305	9875.00000	F310	4790.00000
OSWP-IAP	16200000.00		

020	US	300.00000	020	OS	320.00000
050	US	1000.00000	050	OS	1020.00000
070	US	18.00000	070	OS	20.00000
090	US	5.20000	090	OS	5.50000
110	US	1100.00000	110	OS	1235.00000
130	US	900.00000	130	OS	990.00000
150	US	72.00000	150	OS	90.00000

PS OPSI II 0.0 OPTIMIERUNG GRUNDAUFGABE

SOLUTION OPTIMALE LOESUNG ZU PROBLEM EXCELS
ZEIT = 0.14 MIN. ITERATIONSNUMMER = 16

BEZEICHNUNG.	.....WERT.....	DEFINIERT ALS
ZIELFUNKTION	15583643.6271	WP-IAP'
RECHTE SEITE		75.1
SV-BEREICH..		VAR.1

SEKTION 1 - ZEILEN

NUMMER	ZEILEN- ..NAME..	ST	ZEILEN- ..AKTIVITAET..	UNTERE ...SCHRANKE...	OBERE ...SCHRAN
1	V010	BS	9411,91821	KEINE	9500.0
4	V040	BS	7521.77842	KEINE	7600.0
6	V060	BS	7500,10751	KEINE	7525.0
7	V065	OS	32810,00000	KEINE	32810,0
8	F070	BS	50386,19405	KEINE	50750,0
11	F095	OS	45575,00000	KEINE	45575,0
12	F100	BS	14118,94512	KEINE	14718,0
13	F110	OS	14718,00000	KEINE	14718,0
17	F145	BS	66174,44804	KEINE	66175,0
22	F180	OS	2659,00000	KEINE	2659,0
23	F190	BS	44395,85314	KEINE	44995.0
25	F205	OS	62250.00000	KEINE	62250.0
28	F225	BS	41407,60169	KEINE	41410,0
35	F275	BS	3871,82548	KEINE	3875,0
39	F305	BS	9869,85303	KEINE	9875,0
41	F315	OS	3975.00000	KEINE	3975,0
46	WP-IAP	BS	15583643.6271	KEINE	K
47	WP-BP	BS	13069834.1134	KEINE	K
48	GEWINN	BS	6209980,54575	KEINE	K

SEKTION 2 - SPALTEN

NUMMER	SPALTEN- ..NAME..	ST	PROBL.VARIABLE .....WERT.....	UNTERE ...SCHRANKE...	OBERE ...SCHRAN
50	010	OS	510.00000	490,00000	510,(
51	020	OS	320.00000	300,00000	320,(
52	030	FX	10,00000	10,00000	10,(
53	040	FX	90,00000	90,00000	90,(
54	050	OS	1020,00000	1000,00000	1020,(
55	060	BS	250,05551	200,00000	500,(
56	070	OS	20,00000	18,00000	20,(
57	080	BS	34,43080	34,40000	35,(
58	090	US	5,20000	5,20000	5,5
59	100	BS	120,53577	120,00000	140,(
60	110	BS	1233,60177	1100,00000	1235,(
61	120	BS	211,50692	200,00000	300,(
62	130	OS	990,00000	900,00000	990,(
63	140	OS	0,00000	4,40000	6,(
64	150	BS	89,81403	72,00000	90,(
65	160	FX	,50000	,50000	,5

5

KE...	ZEILEN- ...BEWERTUNG..	LOG.VARIABLE .....WERT.....
0000	.	88.08179
0000	.	78.22158
0000	.	24.89249
0000	36.19692	.
0000	.	363.80595
0000	29.87978	.
0000	.	599.05488
0000	164.04674	.
0000	.	.55196
0000	445.16400	.
0000	.	599.14686
0000	73.20367	.
0000	.	2.33831
0000	.	3.17452
0000	.	5.14697
0000	77.79003	.
EINE	1.00000-	15583643.6271-
EINE	.	13069834.113--
EINE	.	2869980.94575-

KE...	SPALTEN- ...BEWERTUNG..	ZIEL-/ZAEHLER- .FUNKT.KOEFF..
0000	1857.89712	2303.00000
0000	2307.87074	2726.00000
0000	2384.02358	2700.00000
0000	2322.67580	2756.00000
0000	20.73045	3699.00000
0000	.	3830.00000
0000	923.87658	4974.00000
0000	.	14417.00000
0000	1440.18793-	12847.00000
0000	.	2024.00000
0000	.	2714.00000
0000	.	2866.00000
0000	2622.87830	3130.00000
0000	4295.10070	9600.00000
0000	.	5650.00000
0000	7007.77353-	13000.00000

PS OPSI II 0.0 OPTIMIERUNG GRUNDAUFGABE

```
RANGE     PROBLEM EXCELS
          ZEIT =    0.15 MIN.     ITERATIONSNUMMER =    10

          SKT2      = /
          SKT3      = /
          SKT4      = /
          SKT5      = /

          BEZEICHNUNG.    .....WERT....:   DEFINIERT ALS
          ZIELFUNKTION    15583600.        WP-IAP
          RECHTE SEITE                     75.1
          SV-BEREICH..                     VAR.1
```

SEKTION 1 - STABILITAETSBEREICHE FUER ZEILEN AN DEN SCHRANKEN

ZEILE NUMMER		AKTIVI-TAET STATUS	UNTERE SCHRANKE OBERE	STABILI-TAETS-BEREICH	BESCHRKT DURCH	WIRD	ZFWE AE EINH
V065		32810.0	KEINE	32807.7	100	US	36.
	7	OS	32810.0	32813.9	F225	OS	36.
F095		45575.0	KEINE	45567.9	110	US	29.
	11	OS	45575.0	45319.8	100	US	29.
F110		14718.0	KEINE	14498.1	100	OS	164.
	13	OS	14718.0	14718.1	F145	OS	164.
F180		2659.00	KEINE	2648.50	F145	OS	445.
	22	OS	2659.00	2659.84	F225	OS	445.
F205		62250.0	KEINE	62197.9	F225	OS	73.
	25	OS	62250.0	62258.7	150	OS	73.
F315		3975.00	KEINE	3971.98	080	US	77.
	41	OS	3975.00	3975.37	F145	OS	77.

RT- ND/ EIT	ZFWERT- AENDE- RUNG	LOG.VAR. WERT
1969-	82,0953-	.
1969	140,587	
8798-	210,991-	.
8798	7315,74	
047-	36071,3-	.
047	22,1078	
164-	4676,42-	.
164	375,122	
2037-	3811,41-	.
2037	639,840	
7900-	234,637-	.
7900	29,0454	

```
//KOMPLEX1 JOB (01700463,E4254),SCHLOMS,MSGLEVEL=(2,0),
//             REGION=100K,MSGCLASS=S,CLASS=M
//JOBLIB    DD  DSN=OP.LAD,DISP=SHR
//OPCOMP EXEC   PGM=OP#COMP
//COMP1     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//COMP2     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//COMP3     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//COMP4     DD  UNIT=SYSDA,SPACE=(400,(30,30))
//SYSPROG   DD  UNIT=SYSDA,SPACE=(400,(30,30)),DISP=(NEW,PASS)
//SYSPRINT  DD  SYSOUT=(S,N)
//SYSIN     DD  *
MS OPSI II  0.0          COMPILIERSCHRITT - OPSI-PROGRAMM.

            0001                     PROGRAM                  ]**** 1. K
            0003                     INITIAL
            0070                     TITLE('DATENAENDERUNG')
            0071                     MOVE(XAPBNAME,'EXCELS')
            0073                     MOVE(XPBNAME,'EXCELS1')
            0075                     MOVE(XDNAME,'STR-FIX')
            0077                     REVISE
            0079                     TITLE('OPTIMIERUNG 1, AENDERUN
            0080                     SETUP('SVB','VAR,1','MAX')
            0081                     FLAGS('ZMASKEN','V065','F095',
            0081            'F225','F315',' ')
            0083                     MOVE(XRS,'75,1')
            0084                     MOVE(XZF,'WP-IAP')
            0086                     RESTORE('NAME','GRDAUFG')
            0088                     PRIMAL
            0090                     SOLUTION('ZMASKEN','V010','V04
            0090            'F100','F110','F145','F180','F190','F20
            0090            'WP-IAP','WP-BP','GEWINN','OSWP-IAP','
            0091                     RANGE('SKT2','/','SKT3','/','S
            0092                     MOVE(XZF,'WP-BP')
            0094                     PRIMAL
            0096                     SOLUTION('ZMASKEN','V010','V04
            0096            'F100','F110','F145','F180','F190','F20
            0096            'WP-IAP','WP-BP','GEWINN','OSWP-IAP','
            0097                     RANGE('SKT2','/','SKT3','/','S
            0098                     MOVE(XZF,'GEWINN')
            0100                     PRIMAL
            0102                     SOLUTION('ZMASKEN','V010','V04
            0102            'F100','F110','F145','F180','F190','F20
            0102            'WP-IAP','WP-BP','GEWINN','OSWP-IAP','
            0103                     RANGE('SKT2','/','SKT3','/','S
            0104                     EXIT
            0105                     PEND
//OPEXEC EXEC   PGM=OP#EXEC,COND=(0,NE,OPCOMP)
//SYSPROG DD    DSN=*.OPCOMP.SYSPROG,DISP=(OLD,DELETE)
//PBFILE   DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND),DIS
//FILE1    DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//FILE2    DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//MATRIX1 DD    UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//ETA1     DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//SYSPRINT DD   SYSOUT=(S,N)
//SYSIN     DD  *
```

4.3. Rechnerprotokolle zum 1. Komplexschritt für stabile Struktur

```
                                        14.06.80   3.25   SEITE

OMPLEXSCHRITT ***


                   ;*FEST-
                   ;* SETZUNG
                   ;*  LOESUNGS-
                   ;*   STRUKTUR
; - FEST')

F110','F145','F180','F205',        *
                   ;*FREISETZ. ZFILEN

                   ;*OPTIMIERUNG
                   ;* FUER
                   ;*  ZF = WP-IAP
)','V060','V065','F070','F095', *
)','F225','F275','F305','F315', *
,'SMASKEN','***',' ')
:14','/','SKT5','/')
                   ;*OPTIMIERUNG
                   ;*FUER ZF = WP-BP
)','V060','V065','F070','F095', *
)','F225','F275','F305','F315', *
,'SMASKEN','***',' ')
:14','/','SKT5','/')
                   ;*OPTIMIERUNG
                   ;*FUER ZF = GEWINN
)','V060','V065','F070','F095', *
)','F225','F275','F305','F315', *
,'SMASKEN','***',' ')
:14','/','SKT5','/')



)=(OLD,KEEP)
```

∞ VS OPS1 II 0.0 DATENAENDERUNG

REVISE AENDERUNG ALTPROBLEM EXCELS DURCH DATENDECK SIR-FIX Z
ZEIT = 0.01 MIN.

S- SVBF-SEKTION

AENDERE

FX VAR.1	010	510
FX VAR.1	020	320
FX VAR.1	050	1020
FX VAR.1	070	20
FX VAR.1	090	5,2
FX VAR.1	130	990
FX VAR.1	140	6

0 KLEINERE(R) FEHLER - 0 GROESSERE(R) FEHLER

STATISTIK DER PROBLEMMATRIX: 49 ZEILEN, 16 SPALT

0 KLEINERE(R) FEHLER - 0 GROESSERE(R) FEHLER

VS OPS1 II 0.0 OPTIMIERUNG 1. AENDERUNG - FEST

SOLUTION OPTIMALE LOESUNG ZU PROBLEM EXCELS1
ZEIT = 0.09 MIN. ITERATIONSNUMMER = 8

ZMASKEN,V010,V040,V060,V065,F070,F095,F100,F110,F145,F1E
GEWINN,OSWP-IAP, SMASKEN,***,

BEZEICHNUNG.	.....WERT.....	DEFINIERT ALS
ZIELFUNKTION	15636687,1437	WP-IAP
RECHTE SEITE		75,1
SV-BEREICH..		VAR.1

SEKTION 1 - ZEILEN

NUMMER	ZEILEN- ..NAME..	ST	ZEILEN- ..AKTIVITAET..	UNTERE ...SCHRANKE...	OBERE ...SCHRANK
1	V010	BS	9435,75041	KEINE	9500,00
4	V040	BS	7597,08198	KEINE	7600,00
6	V060	BS	7508,36911	KEINE	7525,00
7	V065	BS	32804,17625	KEINE	KE
8	F070	CS	50750,00000	KEINE	50750,00
11	F095	BS	45897,05121	KEINE	KE
12	F100	BS	14556,77241	KEINE	14718,00
13	F110	BS	15210,37584	KEINE	KE
17	F145	BS	68186,61743	KEINE	KE
22	F180	BS	2583,82754	KEINE	KE
23	F190	BS	44457,76042	KEINE	44995,00
25	F205	BS	62205,81376	KEINE	KE
28	F225	BS	41353,03217	KEINE	KE
35	F275	BS	3873,62510	KEINE	3875,00

U PROBLEM EXCELS1

EN, 485 ELEMENTE, BESETZUNGSGRAD = 61.16 %

0,F190,F205,F225,F275,F305,F315,WP-IAP,WP-BW,

E...	ZEILEN- ...BEWERTUNG..	LOG.VARIABLE .....WERT.....
000	.	64.24959
000	.	2.91802
000	.	16.63089
INE	.	5.82375
000	105.69106	.
INE	.	322.05121-
000	.	161.22759
INE	.	492.37584-
INE	.	2011.61743-
INE	.	75.17246
000	.	537.23958
INE	.	44.18624
INE	.	56.36783
000	.	1.37490

39	F305	OS	9875,00000	KEINE	9875.
41	F315	BS	3969,31182	KEINE	
46	WP-IAP	BS	15636687,1437	KEINE	
47	WP-BP	BS	13117833,8566	KEINE	
48	GEWINN	BS	6229043,62213	KEINE	
49	OSWP-IAP	BS	15636687,1437	KEINE	16200000

MS OPS1 II 0.0 OPTIMIERUNG 1. AENDERUNG - FEST

SECTION 2 - SPALTEN

NUMMER	SPALTEN- ..NAME..	ST	PROBL, VARIABLE .....WERT.....	UNTERE ...SCHRANKE...	OBER ...SCHRA
50	010	FX	510,00000	510,00000	510.
51	020	FX	320,00000	320,00000	320.
52	030	FX	10,00000	10,00000	10.
53	040	FX	90,00000	90,00000	90.
54	050	FX	1020,00000	1020,00000	1020.
55	060	BS	293,43664	200,00000	500.
56	070	FX	20,00000	20,00000	20.
57	080	US	34,40000	34,40000	35.
58	090	FX	5,20000	5,20000	5.
59	100	US	120,00000	120,00000	140.
60	110	BS	1204,25410	1100,00000	1235.
61	120	US	200,00000	200,00000	400.
62	130	FX	990,00000	990,00000	990.
63	140	FX	6,00000	6,00000	6.
64	150	OS	90,00000	72,00000	90.
65	160	FX	,50000	,50000	.

MS OPS1 II 0.0 OPTIMIERUNG 1. AENDERUNG - FEST

RANGE PROBLEM EXCELS1
ZEIT = 0,10 MIN. ITERATIONSNUMMER = 8

SKT2 = /
SKT3 = /
SKT4 = /
SKT5 = /

BEZEICHNUNG.	.....WERT.....I	DEFINIERT ALS
ZIELFUNKTION	15636700.	WP-IAP
RECHTE SEITE		75.1
SV-BEREICH..		VAR.1

SECTION 1 - STABILITAETSBEREICHE FUER ZEILEN AN DEN SCHRANKEN

ZEILE NUMMER	AKTIVI-TAET STATUS	UNTERE SCHRANKE OBERE	STABILI-TAETS-BEREICH	BESCHRKT DURCH	ZEI WIRD A EIN
F070	50750,0	KEINE	50498,3	F275	US 105
8	US	50750,0	50765,8	V040	US 105
F305	9875,00	KEINE	9868,99	V040	US 134
39	US	9875,00	9877,30	F275	US 134

00000	1341.36912	.
KEINE	.	5.68818
KEINE	1.00000-	15636687.1437-
KEINE	.	13117833.8566-
KEINE	.	2889045.62213-
.0000	.	503312.85628

E NKE...	SPALTEN- ...BEWERTUNG..	ZIEL-/ZAEHLER- .FUNKT.KOEFF..
00000	303.93594-	2303.00000
00000	1097.72154-	2726.00000
00000	990.97460	2700.00000
00000	486.81198	2756.00000
00000	469.32522	3699.00000
00000	.	3830.00000
00000	913.95177	4974.00000
00000	53463.86115-	14417.00000
20000	163292.81161-	12847.00000
00000	1142.10343-	2024.00000
00000	.	2714.00000
00000	449.38214-	2866.00000
00000	191.72041-	3130.00000
00000	5602.94607	9600.00000
00000	1390.79988	5650.00000
50000	142735.54318-	13000.00000

ERT- END/ HEIT	ZFWERT- AENDE- RUNG	LOG.VAR. WERT
.691-	26600.2-	
.691	1665.20	
1.37-	8056.58-	
1.37	3090.44	

```
//KOMPLEX2 JOB (04700463,E4254),SCHLOMS,MSGLEVEL=(2,0),
//              REGION=100K,MSGCLASS=S,CLASS=M
//JOBLIB   DD   DSN=OP,LAD,DISP=SHR
//OPCOMP EXEC   PGM=OP#COMP
//COMP1    DD   UNIT=SYSDA,SPACE=(400,(30,30))
//COMP2    DD   UNIT=SYSDA,SPACE=(400,(30,30))
//COMP3    DD   UNIT=SYSDA,SPACE=(400,(30,30))
//COMP4    DD   UNIT=SYSDA,SPACE=(400,(30,30))
//SYSPROG  DD   UNIT=SYSDA,SPACE=(400,(30,30)),DISP=(NEW,PASS
//SYSPRINT DD   SYSOUT=(S,N)
//SYSIN    DD   *

PS OPSI II  C.0          COMPILIERSCHRITT - OPSI-PROGRAMM.

           0001                  PROGRAM                    ;**** 1.
           0003                  INITIAL
           0070                  TITLE('OPTIMIERUNG 1. AENDER
           0071                  MOVE(XPBNAME,'EXCELS')
           0072                  SETUP('SVB','VAR.1','MAX')
           0073                  FLAGS('ZMASKEN','V065','F095
           0073        'F225','F315',' ')
           0075                  MOVE(XRS,'75.1')
           0076                  MOVE(XZF,'WP-IAP')
           0078                  RESTORE('NAME','GRDAUFG')
           0080                  PRIMAL
           0082                  SOLUTION('ZMASKEN','V010','V
           0082        'F100','F110','F145','F180','F190','F
           0082        'WP-IAP','WP-BP','GEWINN','OSWP-IAP',
           0083                  RANGE('SKT2','/','SKT3','/',
           0084                  MOVE(XZF,'WP-BP')
           0086                  PRIMAL
           0088                  SOLUTION('ZMASKEN','V010','V
           0088        'F100','F110','F145','F180','F190','F
           0088        'WP-IAP','WP-BP','GEWINN','OSWP-IAP',
           0089                  MOVE(XZF,'GEWINN')
           0091                  PRIMAL
           0093                  SOLUTION('ZMASKEN','V010','V
           0093        'F100','F110','F145','F180','F190','F
           0093        'WP-IAP','WP-BP','GEWINN','OSWP-IAP',
           0094                  RANGE('SKT2','/','SKT3','/',
           0095                  EXIT
           0096                  PEND

//OPEXEC EXEC  PGM=OP#EXEC,COND=(0,NE,OPCOMP)
//SYSPROG DD   DSN=*.OPCOMP.SYSPROG,DISP=(OLD,DELETE)
//PBFILE  DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND),D
//FILE1   DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//FILE2   DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//MATRIX1 DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//ETA1    DD   UNIT=SYSDA,SPACE=(3500,(100),,CONTIG,ROUND)
//SYSPRINT DD  SYSOUT=(S,N)
//SYSIN   DD   *
```

4.4. Rechnerprotokolle zum 1. Komplexschritt für freie Struktur

```
                                        14.06 80    3.26    SEITE     1

K O M P L E X S C H R I T T ***

NG - FREI')


,'F110','F145','F180','F205',        *
                ;*FREISETZ. ZEILEN

                ;*OPTIMIERUNG
                ;*  FUER
                ;*   ZF = WP-IAP
40','V060','V065','F070','F095', *
05','F225','F275','F305','F315', *
 ','SMASKEN','***',' ')
SK14','/','SKT5','/')
                ;*OPTIMIERUNG
                ;*FUER ZF = WP-BP
40','V060','V065','F070','F095', *
05','F225','F275','F305','F315', *
 ','SMASKEN','***',' ')
                ;*OPTIMIERUNG
                ;*FUER ZF = GEWINN
40','V060','V065','F070','F095', *
05','F225','F275','F305','F315', *
 ','SMASKEN','***',' ')
SK14','/','SKT5','/')



SP=(OLD,KEEP)
```

LEHRBRIEFE FÜR DAS HOCHSCHULFERNSTUDIUM

HERAUSGEGEBEN
VON DER ZENTRALSTELLE FÜR DAS HOCHSCHULFERNSTUDIUM
DES MINISTERIUMS FÜR HOCH- UND FACHSCHULWESEN

Mathematische Methoden und Modelle

9. LEHRBRIEF

Grundlagen und Anwendung des PP DISKO

02 4602 09 0

Mathematische Methoden und Modelle

9. Lehrbrief

Grundlagen und Anwendung des PP DISKO

Autoren:

Dr. rer. oec. Heike B o h n e n k a m p
Martin-Luther-Universität Halle
Sektion Wirtschaftswissenschaften

Dipl.-math. Frank F i c k e r
VEB Robotron Dresden

Dr. rer. oec. Axel S t o l z e
Martin-Luther-Universität Halle
Sektion Wirtschaftswissenschaften

Dr. rer. nat. Gottfried W i l d e n h a i n
VEB Robotron Dresden

02 4602 09 0

Das druckfertige Manuskript wurde an der
Martin-Luther-Universität Halle hergestellt.

Bestell-Nr. 02 4602 09 0
Verfaßt für die Zentralstelle für das Hochschulfernstudium des Ministeriums für Hoch- und Fachschulwesen Dresden.
Herausgegeben im Auftrag des Ministeriums für Hoch- und Fachschulwesen der Deutschen Demokratischen Republik von der Zentralstelle für das Hochschulfernstudium Dresden.

Inhaltsverzeichnis Seite

Seite

0. Vorbemerkungen

Die lineare diskrete Optimierung besitzt für viele Spezialprobleme der Leitung und Planung ökonomischer Prozesse eine große Bedeutung. Zu den wichtigsten Anwendungsgebieten gehören:

1. Die gemischt-ganzzahlige lineare Optimierung für ökonomische Probleme mit Ganzzahligkeitsforderungen.
2. Die lineare 0-1-Optimierung für ökonomische Probleme mit Alternativcharakter.
3. Spezialverfahren der linearen Transportoptimierung.

Das "einheitliche System der elektronischen Rechentechnik" (ESER), an dessen Entwicklung die sozialistischen Länder gemeinsam beteiligt sind, ist die gegenwärtige gerätetechnische Basis der elektronischen Datenverarbeitung in der DDR. Zur Lösung von vielfältigen Aufgabenstellungen der diskreten Optimierung wird vom VEB Robotron das leistungsfähige Programmpaket diskrete Optimierung (PP DISKO) bereitgestellt.

Nachdem im Abschnitt eins auf die allgemeine Bedeutung der linearen diskreten Optimierung in der Ökonomie eingegangen wird, erfolgt im Abschnitt zwei eine Einführung in die Grundlagen des PP DISKO. Sie ist beschränkt auf ausgewählte Teilgebiete, die für den Erstanwender erforderlich sind.

Während im Abschnitt drei zwei repräsentative ökonomische Beispiele mit Hilfe der Methoden der diskreten Optimierung mathematisch modelliert werden, demonstriert der Abschnitt vier deren rechentechnische Realisierung mit dem PP DISKO.

Halle, im Herbst 1980

Doz. Dr. sc. W. Lassmann
Martin-Luther-Universität Halle
Sektion Wirtschaftswissenschaften
Wissenschaftsbereich Informationsverarbeitung
Lehrgruppe Mathematik
(Leitung der Lehrbriefe 8-10)

1. Ökonomische Problemstellung der diskreten Optimierung

Die innerhalb der diskreten Optimierung auftretenden ökonomischen Probleme linearen Charakters lassen sich wie folgt gruppieren:

- gemischt-ganzzahlige lineare Optimierungsprobleme.
- lineare 0-1-Optimierungsprobleme,
- lineare Transport-Optimierungsprobleme.

Zur ersten Gruppe, den gemischt-ganzzahligen Problemen, zählen solche Aufgabenstellungen, bei denen es vom ökonomischen Sachverhalt her notwendig und sinnvoll ist, daß einige oder auch alle Variablen in der Problemlösung ganzzahlig auftreten.
Das wird immer dann der Fall sein, wenn Anzahlen, wie Arbeitskräfte, Erzeugnisstückzahlen usw. im Lösungsausdruck erscheinen.
Probleme der zweiten Gruppe sind durch Ja-Nein-Entscheidungen gekennzeichnet. Mathematisch wird dieser Sachverhalt durch Variable ausgedrückt, die nur die Werte 0 (Nein) oder 1 (Ja) annehmen dürfen.
Aufgabenstellungen der 3. Gruppe nehmen eine gewisse Sonderstellung ein, da einige modellmäßig der ersten und andere der zweiten Gruppe zuzuordnen sind.
Ihre Gemeinsamkeit besteht u.a. in der Existenz spezieller, nur für die Lösung dieser Probleme geeigneter Algorithmen.
Im folgenden werden einige ökonomische Aufgabenstellungen vorgestellt, die sich in eine der drei vorgestellten Gruppen einordnen lassen.

1.1. Bestimmung eines optimalen Produktionssortiments

Für ein Kombinat oder einen Betrieb wird das zukünftige Produktionsprofil bestimmt. Hierbei sind bezüglich der Frage - wird das Erzeugnis produziert oder nicht? - zwei Gruppen von Produkten zu unterscheiden:

a) fest vorgegebene Produkte

b) frei wählbare Produkte.

Für die unter a) erfaßten Produkte sind die zu produzierenden Mengen zu ermitteln. Für die frei wählbaren Produkte wird entschieden, ob sie in das Produktionsprogramm aufgenommen werden. Bei Aufnahme der Produkte ins Produktionsprogramm ist eine vorgegebene Mindestmenge zu produzieren.
Im Modell werden deshalb stetige Variable x_j und ganzzahlige Variable y_j mit folgender Bedeutung verwendet:

x_j	zu produzierende Menge des Produktes P_j, $j=1(1)n$
$y_j = 1$	Produkt P_j wird in das Produktionsprogramm aufgenommen, $j \in \{1,2, \ldots,n\}$
$y_j = 0$	Produkt P_j wird nicht in das Produktionsprogramm aufgenommen, $j \in \{1,2, \ldots,n\}$.

Als Restriktionen kommen z.B. in Frage:

- Material,
- Investitionsmittel,
- Arbeitskräfte,
- Maschinenzeit,
- Bedarf.

Weiterhin können zusätzliche Forderungen und Möglichkeiten für bestimmte Produkte im Modell berücksichtigt werden:

- Es können verschiedene Fertigungstechnologien angewendet werden; je Produkt darf aber nur genau eine gewählt werden.
- Bei der Aufnahme eines Produktes in das Programm entstehen von der Produktionshöhe unabhängige (konstante) Kosten.
- Verschiedene Produkte sind gegenseitig substituierbar.

Es gilt, das Produktionssortiment in Struktur und Menge zu bestimmen sowie die Technologien sinnvoll zuzuordnen. Als Zielfunktion können dabei z.B.

- maximaler Gewinn,
- maximales Betriebsergebnis oder

- minimale Kosten

auftreten.

Unter der Voraussetzung, daß als Zielkriterium das Betriebsergebnis oder der Gewinn gewählt wurde, lautet das mathematische Modell wie folgt:

Zielfunktion

$$(1.) \qquad Z = \sum_{j=1}^{n} c_j \, x_j - \sum_{j=1}^{n} y_j \, k_j \Longrightarrow \text{Max!}$$

Restriktionen

$$(2.1.) \qquad \sum_{j=1}^{n} a_{ij} \, x_j \gtreqless b_i \qquad \text{für } i=1(1)m$$

$$(2.2.) \qquad x_k \geqq u_k \, y_k \qquad k \in \{1,2, \ldots, n\}, \quad u_k > 0$$

$$(2.3.) \qquad x_j \leqq o_j \, y_j \qquad j=1(1)n$$

$$(2.4.) \qquad x_j \geqq 0 \qquad j=1(1)n$$

$$(2.5.) \qquad y_j \in \{0,1\} \qquad j=1(1)n$$

Erläuterungen:

c_j	Zielfunktions-Koeffizient, z.B. Gewinn, Betriebsergebnis
x_j	Produktionsmengen
y_j	Variable, die die Aufnahme des Erzeugnisses ins Produktionsprogramm angibt
k_j	mengenunabhängige (fixe) Kosten
a_{ij}	Aufwand des Erzeugnisses j an Restriktion i
b_i	rechte Seiten der Restriktion i (z.B. Materialfonds)

u_k Mindestmenge an Produkt P_k, die bei Aufnahme in das Programm zu produzieren ist

o_j Höchstmenge des Produktes P_j

n Gesamtanzahl der Erzeugnisse

m Anzahl der Restriktionen.

Im Modell wurden als zusätzliche Forderungen berücksichtigt:

1. Produktion einer geforderten Mindestmenge [vgl. Bedingung (2.2.)]
2. Berücksichtigung der mengenunabhängigen Kosten [vgl. (1.) und (2.3.)].

Vorliegendes Problem gehört zur Gruppe der <u>gemischt-ganzzahligen</u> linearen Optimierungsprobleme.

1.2. Probleme der Grundfondsreproduktion

In einem Kombinat oder Betrieb ist für einen bestimmten Zeitraum der Plan der Reproduktion der Grundmittel aufzustellen. Es sind mehrere Vorhaben (Objekte) vorhanden, deren Reproduktion nach verschiedenen Varianten erfolgen kann. Auf Grund beschränkter Ressourcen macht sich eine <u>Auswahl</u> der zu realisierenden Objekte <u>erforderlich.</u> Grundsätzlich darf dabei <u>nur eine Variante je Vorhaben</u> gewählt werden.
<u>Restriktionen</u> des Modells sind ökonomische Größen, die für Investitionen wesentlich sind. Das sind z.B.:

- finanzielles Investitionsvolumen,
- Bauvolumen,
- Arbeitskräftefonds oder -änderung,
- Materialbedarf,
- Bedarf an Ausrüstungen,
- Bedarfskennziffern,
- staatliche Plankennziffern.

Dabei können die Restriktionen sowohl für die Realisierung

der Investition als auch für das spätere Betreiben der Anlage gelten.
Bei Problemen der Grundfondsreproduktion sind in der Regel Kennziffern, die in verschiedenen Formen das Verhältnis von Investitionsaufwand und -ergebnis erfassen, Zielkriterium. Hier bieten sich u.a. folgende Zielfunktionen an

- Maximierung der Investitionsrentabilität,
- Maximierung der Grundfondsquote,
- Maximierung der Arbeitsrentabilität.

Aus den oben getroffenen Aussagen ergibt sich das mathematische Modell in allgemeiner Form:

Zielfunktion

(1.) $$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} g_{ij}\, x_{ij} \Longrightarrow \text{Max!}$$

Restriktionen

(2.1.) $$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^{(k)}\, x_{ij} \gtreqless b_k \qquad \text{für } k=1(1)l$$

(2.2.) $$\sum_{j=1}^{n} x_{ij} = 1 \qquad i=1(1)m$$

(2.3.) $$x_{ij} \in \{0,1\} \qquad \begin{matrix} i=1(1)m \\ j=1(1)n \end{matrix}$$

Die Variable x_{ij} gibt an, ob das Vorhaben i nach Variante j realisiert wird ($x_{ij} = 1$) oder nicht ($x_{ij} = 0$).

Erläuterungen:

g_{ij}	Zielfunktionskoeffizient (z.B. Grundfondsquote) des Vorhabens i in der Variante j,
$a_{ij}^{(k)}$	Aufwand des Vorhabens i in der Variante j bezüglich Restriktion k,
b_k	rechte Seite der Restriktion k (u.a. vorgegebene Fonds),
m	Anzahl der Vorhaben,
n	Anzahl der Varianten,
l	Anzahl der Restriktionen.

Das Problem der Grundfondsreproduktion ist ein lineares 0-1-Optimierungsproblem.

1.3. Problem der Maschinenbelegung

In einem Betrieb sind in einem vorgegebenen Zeitraum bestimmte Mindestmengen an Produkten in einem einstufigen Produktionsprozeß zu produzieren.
Einschränkend wirkt die Kapazität der Maschinen.
Außerdem ist zu beachten:

- Die Bearbeitung eines Produktes erfolgt jeweils nur auf einer Maschine, d.h. sie beinhaltet nur einen Arbeitsgang.
- Die Maschinen sind nicht generell substituierbar.
- Der Bearbeitungsaufwand je Produkt und je Maschine ist unterschiedlich.
- Bei der Bearbeitung eines Produktes entstehen mengenabhängige Produktions- und mengenunabhängige Umstellungskosten.

Ziel ist die Minimierung der Gesamtkosten. Das mathematische Modell hat folgende allgemeine Form:

Zielfunktion

(1.) $$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}\, x_{ij} + \sum_{i=1}^{m} \sum_{j=1}^{n} u_{ij}\, y_{ij} \implies \text{Min!}$$

Restriktionen

(2.1.) $$\sum_{j=1}^{n} a_{ij}\, x_{ij} \leqq k_i \qquad i=1(1)m$$

(2.2.) $$\sum_{i=1}^{m} x_{ij} \geqq b_j \qquad j=1(1)n$$

(2.3.) $$x_{ij} \leqq o_j\, y_{ij} \qquad i=1(1)m,\; j=1(1)n$$

(2.4.) $$x_{ij} \geqq 0 \qquad i=1(1)m,\; j=1(1)n$$

(2.5.) $$y_{ij} \in \{0,1\} \qquad i=1(1)m,\; j=1(1)n$$

Erläuterungen:

p_{ij}	Produktionskosten für eine Einheit des Erzeugnisses j auf Maschine i
u_{ij}	Umstellungskosten für Erzeugnis j auf Maschine i
x_{ij}	Produktionsmenge von Erzeugnis j auf Maschine i
$y_{ij}=1$	Erzeugnis j wird auf Maschine i produziert
$y_{ij}=0$	Erzeugnis j wird nicht auf Maschine i produziert
a_{ij}	Bearbeitungsaufwand für eine Einheit des Produktes j auf Maschine i

k_i	Kapazität der Maschine i
b_j	Bedarf an Erzeugnis j
o_j	Höchstmenge des Erzeugnisses j
m	Anzahl der Maschinen
n	Anzahl der Erzeugnisse

Das dargestellte Problem ist der Gruppe der gemischt-ganzzahligen linearen Optimierungsprobleme zuzuordnen.

1.4. Optimierung des Materialzuschnittes

Bestimmte Rohmaterialien müssen in Einzelteile unterschiedlichster Form und Größe zerschnitten werden. Dabei existieren verschiedene Zuschnittvarianten, in denen mehrere Einzelteile nach technologischen, materialökonomischen und anderen Gesichtspunkten günstig angeordnet sind.
Als Restriktionen dienen hierbei die für die Einzelteile geforderten Mengen.
Als Optimalitätskriterium können z.B.

- minimaler Materialverbrauch,
- minimaler Abfall,
- minimale Gesamtkosten oder
- minimale Anzahl von Zuschnittvarianten

gewählt werden.
Es ergibt sich folgendes mathematische Modell:

Zielfunktion

(1.) $$Z = \sum_{j=1}^{n} y_j \Longrightarrow \text{Min!}$$

Restriktionen

(2.1.) $$\sum_{j=1}^{n} a_{ij}\, x_j \geqq b_i \qquad \text{für } i=1(1)m$$

(2.2.)	$x_j \leqq o_j\, y_j$	$j=1(1)n$
(2.3.)	$x_j \geqq 0$, ganzz.	$j=1(1)n$
(2.4.)	$y_j \in \{0,1\}$	$j=1(1)n$

Erläuterungen:

x_j	Menge an Rohmaterial, die nach Variante j zerschnitten wird
$y_j=1$	Variante j wird benötigt
$y_j=0$	Variante j wird nicht benötigt
o_j	Höchstmenge an Rohmaterial, die nach Variante j zerschnitten werden kann
a_{ij}	Anzahl der Einzelteile i in Variante j
b_i	benötigte Menge an Einzelteilen i
m	Anzahl der Einzelteile
n	Anzahl der Zuschnittvarianten.

Hierbei handelt es sich um ein ganzzahliges lineares Optimierungsproblem.
Bei Materialzuschnittproblemen ist zu beachten, daß zu ihrer Lösung auch PS OPSI verwendet werden kann. Das ist z.B. dann der Fall, wenn die Lösung nichtganzzahlig sein darf.

1.5. Lieferproblem

Ein Betrieb hat mit einer Anzahl von Kunden Lieferverträge abgeschlossen. Es werden mehrere Produkte geliefert.
Der Betrieb muß in Abhängigkeit von seinen Lagerbeständen entscheiden, welche Kunden im zu planenden Zeitraum zu beliefern sind.

Wird ein Kunde beliefert, erfolgt die Belieferung mit allen Produkten.
Als Restriktionen können neben Lagerbeständen auch begrenzte Transportkapazitäten verwendet werden.
Weiterhin ist es möglich, Prioritäten für einzelne Kunden zu berücksichtigen. Ebenso kann die Belieferung einer Mindestanzahl Kunden gefordert werden.
Ziel ist eine Minimierung der Lagerbestände oder -kosten.
Um die Pufferfunktion des Lagers zu sichern, ist hierbei gegebenenfalls ein Mindestbestand festzulegen.
Es ergibt sich folgendes mathematische Modell:

Zielfunktion

(1.) $$Z = \sum_{j=1}^{n} c_j \, x_j \Longrightarrow \text{Max!}$$

Restriktionen

(2.1.) $$\sum_{j=1}^{n} a_{ij} \, x_j \leqq b_i \qquad \text{für } i=1(1)m$$

(2.2.) $$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} \, x_j \leqq K$$

(2.3.) $$x_j \in \{0,1\} \qquad j=1(1)n$$

Erläuterungen:

c_j Lagerkapazität, die die Bestellung des Kunden j beansprucht mit
$$c_j = \sum_{i=1}^{m} a_{ij}$$

$x_j = 1$ Kunde j wird beliefert

$x_j=0$ Kunde j wird nicht beliefert

a_{ij} bestellte Menge an Produkt i durch Kunden j

b_i Lagerbestand an Produkt i

K Fahrzeugkapazität

m Anzahl der Produkte

n Anzahl der Kunden.

Sollen zusätzlich Prioritäten berücksichtigt werden, so kann das in folgender Weise geschehen:

$$x_{j_1} \geqq x_{j_2} ,$$

d.h. x_{j_2} kann nur dann gleich 1 sein, wenn $x_{j_1} = 1$ ist.

Der Kunde j_2 wird also nur beliefert, wenn der Kunde j_1 beliefert wird.

Das Lieferproblem gehört zu den linearen 0-1-Optimierungsproblemen.

1.6. Problem der optimalen Auswahl von Technologien

In einem Betrieb sind für die Produktion einzelner Produkte jeweils mehrere Fertigungstechnologien anwendbar, die sich hinsichtlich ihres Aufwandes an Zeit und anderen Fonds unterscheiden.

Es soll aber nur eine Technologie je Produkt gewählt werden.

Ziel der Optimierung sind minimale Kosten.

Daraus entsteht folgendes mathematische Modell:

Zielfunktion

(1.) $$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \Longrightarrow \text{Min!}$$

Restriktionen

(2.1.) $$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^{(k)} x_{ij} \leqq b_k \qquad k=1(1)l$$

(2.2.) $$\sum_{j=1}^{n} x_{ij} = 1 \qquad i=1(1)m$$

(2.3.) $$x_{ij} \in \{0,1\} \qquad i=1(1)m, \; j=1(1)n$$

Erläuterungen:

c_{ij}	Kosten für Technologie j und Produkt i
$x_{ij}=1$	Produkt i wird mit Technologie j hergestellt
$x_{ij}=0$	Produkt i wird nicht mit Technologie j hergestellt
$a_{ij}^{(k)}$	Gesamtaufwand an Fonds k für Technologie j und Produkt i
b_k	zur Verfügung stehende Fonds
m	Anzahl der Produkte i
n	Anzahl der Technologien j
l	Anzahl der Fonds k.

Bei dem dargestellten Problem handelt es sich ebenfalls um eine Aufgabenstellung der 0-1-Optimierung.

1.7. Transportverteilungsproblem

Transportverteilungsprobleme treten, bedingt durch die umfassende Arbeitsteilung, zahlreich auf.
Inhaltlich geht es dabei um die Ermittlung optimaler Lieferbeziehungen zwischen einer Anzahl m von Aufkommensorten A_i,

$i=1(1)m$, und einer Anzahl n von Bedarfsorten B_j, $j=1(1)n$. Dabei verfügen die A_i, $i=1(1)m$, über ein Aufkommen a_i, $i=1(1)m$, die B_j, $j=1(1)n$, über einen Bedarf b_j, $j=1(1)n$, an einer <u>austauschbaren Gutart.</u>

Zunächst wird vorausgesetzt, daß die Bedingung

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

erfüllt ist.

Dann lautet das allgemeine mathematische Modell für das <u>klassische Transportverteilungsproblem:</u>

<u>Zielfunktion:</u>

(1.) $$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \Longrightarrow \text{Min!}$$

<u>Restriktionen:</u>

<u>aufkommensseitig:</u>

(2.1.) $$\sum_{j=1}^{n} x_{ij} = a_i \qquad i=1(1)m$$

<u>bedarfsseitig:</u>

(2.2.) $$\sum_{i=1}^{m} x_{ij} = b_j \qquad j=1(1)n$$

<u>Gleichgewichtsbedingung:</u>

(2.3.) $$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

<u>Nichtnegativitätsbedingung:</u>

(2.4.) $$x_{ij} \geqq 0 \qquad i=1(1)m,\ j=1(1)n$$

Erläuterungen:

Über das schon Gesagte hinausgehend sind mit c_{ij}; $i=1(1)m$, $j=1(1)n$, die Aufwendungen für die Realisierung der Transporte pro Produkteinheit zwischen allen Aufkommens- und Bedarfsorten gegeben.
Mit x_{ij}; $i=1(1)m$, $j=1(1)n$, werden die von den Aufkommens- zu den Bedarfsorten zu transportierenden Mengen bezeichnet.
Es ist ein solcher Transportplan zu finden, bei dem die insgesamt anfallenden Transportaufwendungen minimal werden.

Desweiteren können die für den klassischen Fall angegebenen Restriktionen allgemeiner gefaßt werden:

$$(2.5.) \qquad \sum_{j=1}^{n} x_{ij} \gtreqless a_i \qquad i=1(1)m,$$

$$(2.6.) \qquad \sum_{i=1}^{m} x_{ij} \gtreqless b_j \qquad j=1(1)n.$$

Dabei muß in jeder der m unter (2.5.) festgelegten Beziehungen entweder das Zeichen "$\geqq$" oder "=" oder "$\leqq$" stehen. Gleiches gilt für die Beziehung (2.6.).

Unter Berücksichtigung dieser Aussagen können damit beispielsweise folgende Problemstellungen, die über das klassische Transportverteilungsproblem hinausgehen, gelöst werden:

Überschußproblem:

Hierbei sind die Aufkommensmengen der Aufkommensorte größer als die Bedarfsmengen der Bedarfsorte. Das heißt, daß die Bedingungen (2.1.) und (2.3.) ersetzt werden durch:

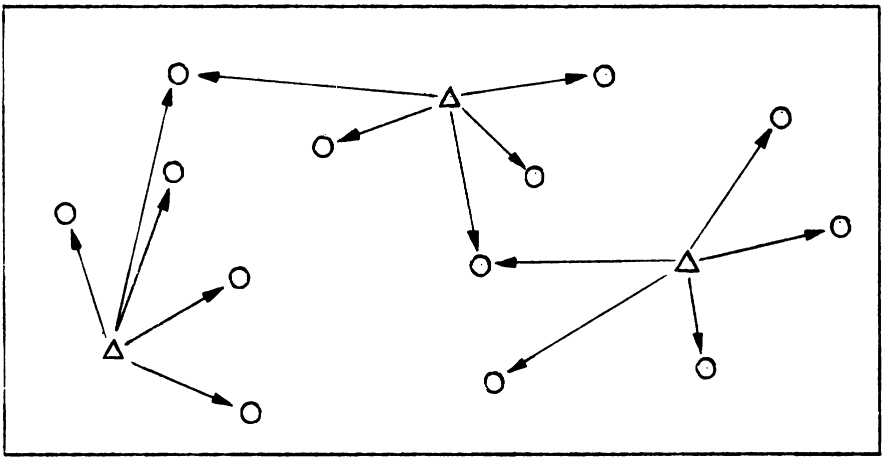
$$(2.7.) \qquad \sum_{j=1}^{n} x_{ij} \leqq a_i \qquad i=1(1)m,$$

$$(2.8.) \qquad \sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \ .$$

Gleiches gilt, wenn nicht der gesamte Bedarf gedeckt werden kann (Mangelproblem).

Damit wird auch die Gleichgewichtsbedingung (2.3.) entschärft; es kann also auch das unausgeglichene Transportverteilungsproblem gelöst werden. Mit dieser Vorgangsweise wird gleichzeitig die Einführung fiktiver Aufkommens- bzw. Bedarfsorte umgangen, deren Verwendung Probleme in der Realisierung der gefundenen Lösungen mit sich bringen kann. Schließlich gestattet das Programm auch die Berücksichtigung der Forderungen, daß

- einige Aufkommensorte ihr Aufkommen in einer Mindesthöhe absetzen müssen
- einige der Bedarfsorte ihren Bedarf in einer Mindesthöhe decken müssen.



△ : Aufkommensorte
○ : Bedarfsorte

Bild 1.1.: Transportbeziehungen beim Transportverteilungsproblem

1.8. Transportrundfahrtproblem

Rundfahrtprobleme treten in vielen Bereichen der Volkswirtschaft in großem Umfang auf.
Innerhalb eines vorgegebenen Netzes von Verbindungsstrecken befinden sich die n anzufahrenden Punkte $P_1, P_2, \ldots, P_n$, die jeweils einen bekannten Transportbedarf $p_1, p_2, \ldots, p_n$ besitzen. Ein Fahrzeug mit der Kapazität K soll von einem Ausgangspunkt P_j, $j \in \{1(1)n\}$, dem Startpunkt, nacheinander alle anderen Punkte anfahren, wobei jeder anzufahrende Punkt nur einmal berührt werden darf und muß dann zum Ausgangspunkt, nunmehr dem Zielpunkt, zurückkehren. Start- und Zielpunkt sind beim Rundfahrtproblem identisch. Ist das nicht der Fall, handelt es sich um ein Durchfahrtproblem. Der Gesamttransportaufwand (Gesamttransportentfernung oder Gesamttransportkosten oder Gesamttransportzeit) soll ein Minimum werden. Es wird vorausgesetzt, daß der Transportbedarf aller anzufahrenden Punkte kleiner oder gleich der Kapazität K des Fahrzeuges ist. Bei Verteilungsfahrten ist p_i, $i=1(1)n$, eine nach P_i, $i=1(1)n$, zu transportierende Menge. Bei Sammelfahrten ist p_i, $i=1(1)n$, eine von P_i, $i=1(1)n$, abzufahrende Menge.

Das mathematische Modell lautet:

Zielfunktion:

(1.) $$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \Longrightarrow \text{Min!}$$

Restriktionen:

(2.1.) $$\sum_{j=1}^{n} x_{ij} = 1 \qquad \text{für } i=1(1)n,$$

(2.2.) $$\sum_{i=1}^{n} x_{ij} = 1 \qquad \text{für } j=1(1)n,$$

(2.3.) $x_{ij} = 0$ für $i = j,$
$i,j = 1(1)n.$

<u>Nichtnegativitäts- und Ganzzahligkeitsbedingung:</u>

(2.4.) $0 \leqq x_{ij} \leqq 1$; ganzzahlig
$i,j = 1(1)n.$

Zudem muß gewährleistet werden, daß die Punkte nacheinander innerhalb einer <u>geschlossenen Rundfahrt</u> angefahren werden, also keine <u>Kurzzyklen</u> auftreten.
Über die aufgeführten Nebenbedingungen hinausgehend kann berücksichtigt werden, daß <u>ausgesuchte Verbindungen</u> in der Problemlösung <u>enthalten sein müssen</u> oder <u>nicht enthalten sein dürfen</u>.

<u>Erläuterungen:</u>

Ausgangspunkt der Berechnung sind die <u>Aufwands- oder Erlöskoeffizienten</u> c_{ij}, $i;j = 1(1)n$.

Das Rundfahrtproblem kann als Minimierungs- oder Maximierungsproblem auftreten. Beim Minimierungsproblem ist c_{ij}, $i;j = 1(1)n$, ein <u>Aufwandskoeffizient</u>, der den Aufwand bei der Realisierung der Verbindung $P_i \longrightarrow P_j$, $i;j = 1(1)n$, kennzeichnet. Beim Maximierungsproblem ist c_{ij}, $i;j = 1(1)n$, ein <u>Erlöskoeffizient</u>, der den Erlös bei der Realisierung der Verbindung $P_i \longrightarrow P_j$, $i;j = 1(1)n$, darstellt. Die Aufwands- oder Erlöskoeffizienten c_{ij}, $i;j = 1(1)n$, werden zu einer quadratischen Matrix $\underline{C}_{(n)}$ zusammengefaßt.

Die Matrix $\underline{C}_{(n)}$ kann <u>symmetrisch oder unsymmetrisch</u> sein.
$\underline{C}_{(n)}$ ist symmetrisch, wenn für die einzelnen Elemente der Matrix gilt:

$$c_{ij} = c_{ji} \qquad \text{für } i;j = 1(1)n.$$

$\underline{C}_{(n)}$ ist unsymmetrisch, wenn in mindestens einem Fall gilt:

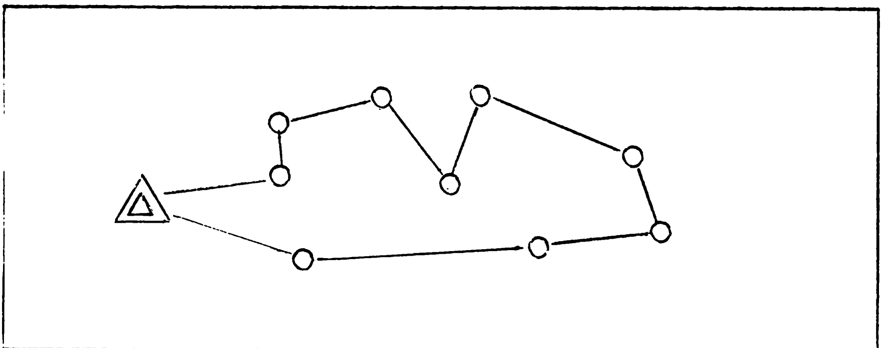
$$c_{ij} \neq c_{ji} \qquad \text{für } i;j = 1(1)n.$$

Die Matrix $\underline{C}_{(n)}$ wird im Fernverkehr zumeist symmetrisch, im Stadtverkehr dagegen fast immer unsymmetrisch sein.
Gesucht sind die Elemente x_{ij}, $i;j = 1(1)n$, die in Matrix $\underline{X}_{(n)}$ zusammengefaßt werden und den Wert <u>Null oder Eins</u> annehmen können.

$x_{ij} = 1$, wenn die Verbindung $P_i \longrightarrow P_j$, $i;j \in \{1(1)n\}$ realisiert wird und

$x_{ij} = 0$, wenn die Verbindung $P_i \longrightarrow P_j$, $i;j \in \{1(1)n\}$ nicht realisiert wird.

In jeder Zeile und in jeder Spalte der Lösungsmatrix $\underline{X}_{(n)}$ darf nur ein Element gleich Eins sein. Dadurch wird gesichert, daß jeder Punkt nur einmal angefahren wird.



△ : Start- und Zielpunkt
○ : Bedarfsträger

<u>Bild 1.2.:</u> Transportbeziehungen beim Transportrundfahrtproblem

1.9. <u>Transporttourenproblem</u>

Im Unterschied zum Transportrundfahrtproblem, bei dem $\sum_{i=1}^{n} p_i \leq K$ war, gilt beim Tourenproblem $\sum_{i=1}^{n} p_i > K$,

das bedeutet, daß der Transportbedarf p_i, i=1(1)n, aller anzufahrenden Punkte größer als die Kapazität K eines Fahrzeugs ist.
Das Transporttourenproblem tritt bei der Ermittlung von Lieferbeziehungen zwischen einem "Depot" P_o (Produktionsbetrieb, Lager) und einer Anzahl von Bedarfsträgern P_i, i=1(1)n, auf. Die Transporte werden mit Fahrzeugen eines vorhandenen Fuhrparks realisiert. Die Bedarfsträger haben einen feststehenden Bedarf p_i, i=1(1)n. Dieser darf, bezogen auf eine Tour, die Kapazität des Kraftfahrzeuges nicht übersteigen. [1)]
Die Fahrzeuge können mehrfach eingesetzt werden.
Es besteht die Möglichkeit, mehrere benachbarte Bedarfsträger zu einer Zone zusammenzufassen. Dadurch kann die Anzahl der im Problem auftretenden Punkte unter Umständen erheblich reduziert werden.
Schließlich soll innerhalb des vorgegebenen Netzes von Verbindungsstrecken, vom Depot ausgehend, eine entsprechende Anzahl von Touren in Form einzelner Rundfahrten gefahren werden. Innerhalb der Touren sollen die Bedarfsträger so wenig wie möglich und so oft wie notwendig angefahren werden. Am Ende muß das Fahrzeug jeweils zum Depot zurückkehren.
Es ergibt sich folgendes mathematische Modell:

Zielfunktion:

(1.) $$Z = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{t=1}^{q} x_{ijt} \, c_{ij} \Longrightarrow \text{Min!}$$

Restriktionen:

(2.1.) $$\sum_{i=0}^{n} x_{ijt} = 0 \text{ oder } = 1, \quad j=0(1)n, \quad t=1(1)q$$

1) Diese Bedingung muß Bestandteil des mathematischen Modells sein, wird jedoch der Einfachheit halber hier nicht weiter ausgeführt.

(2.2.) $$\sum_{j=0}^{n} x_{ijt} = 0 \text{ oder } = 1, \quad i=0(1)n, \quad t=1(1)q$$

$$x_{ijt} = \begin{cases} 1, & \text{wenn die Verbindung } P_i \longrightarrow P_j \text{ in Tour } t \text{ realisiert wird } (i \neq j) \\ 0, & \text{ansonsten.} \end{cases}$$

Nichtnegativitäts- und Ganzzahligkeitsbedingung:

(2.3.) $$0 \leqq x_{ijt} \leqq 1, \text{ ganzzahlig}$$
$$i = 0(1)n$$
$$j = 0(1)n$$
$$t = 1(1)q$$

Erläuterungen:

q läßt die im Zusammenhang mit der Gesamtlösung des Problems zu bestimmende Anzahl der Touren erkennen.
c_{ij}, $i;j = 0(1)n$, kennzeichnet den Aufwand für die Realisierung der Verbindung $P_i \longrightarrow P_j$, $i;j = 0(1)n$.
Gesucht sind die Werte x_{ijt}, $i;j = 0(1)n$, wobei sich $t=1(1)q$ auf die entsprechende Tour bezieht.
Die Lösungsvariablen können wiederum nur die Werte Null oder Eins annehmen.
Es kann sowohl mit symmetrischen als auch mit unsymmetrischen Entfernungsmatrizen gearbeitet werden.
Eine andere Möglichkeit des Herangehens an die Lösung des Transporttourenproblems besteht in der Aufteilung des Gesamtproblems in zwei Teilprobleme:

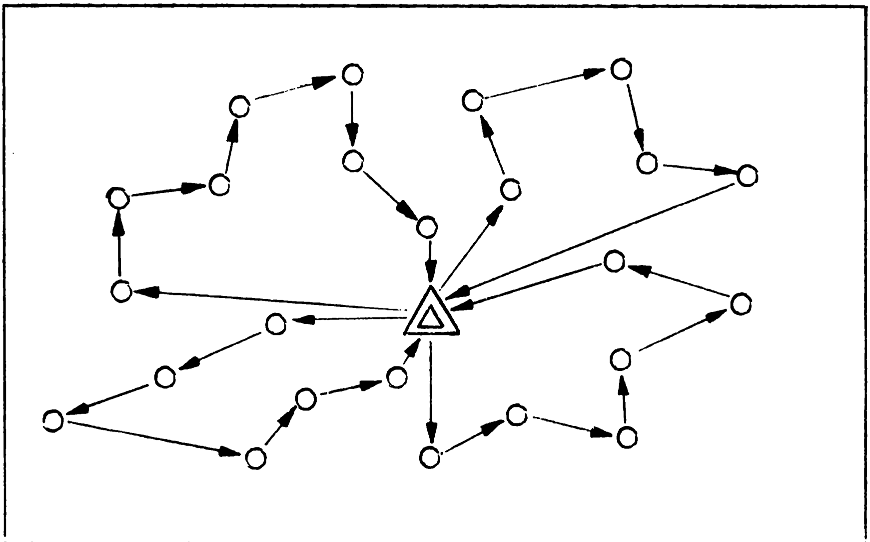
1. Aufteilung der insgesamt n anzufahrenden Punkte auf m Rundfahrten

2. Optimierung der m Rundfahrten.

Dabei ergibt sich m als kleinste ganze Zahl größer oder gleich

$$Q = \frac{\sum_{i=1}^{n} p_i}{K} ,$$

wobei K eine Fahrzeugdurchschnittskapazität ist.
Das erste Teilproblem kann über das <u>Transportverteilungsproblem</u>, das zweite als <u>Transportrundfahrtproblem</u> gelöst werden.
Näher kann darauf im Seminar eingegangen werden.



△ : Depot
○ : Bedarfsträger
→ : gerichtete Verbindung

<u>Bild 1.3.:</u> Transportbeziehungen beim Transporttourenproblem

Kontrollfragen:

1. Überlegen Sie, welche ähnlichen Problemstellungen Ihnen aus Ihrem Arbeitsgebiet bekannt sind, und ordnen Sie diese einer der drei im Punkt 1 genannten Gruppen zu.
   Prüfen Sie, welche Restriktionen jeweils zu beachten sind!

2. Stellen Sie für das Materialzuschnittproblem die Zielfunktion auf, die minimale Gesamtkosten fordert!

3. Bei der Lösung des Transporttourenproblems durch Zerlegung in zwei Teilprobleme kommt es in jedem Fall zu Mehrfachzuordnungen einzelner Bedarfsträger auf verschiedene Rundfahrten.
   Welche Bedeutung messen Sie dieser Tatsache bei?
   Welche Möglichkeiten der Beseitigung der Mehrfachzuordnungen sehen Sie?

Lösungen:

Zu 2: $$Z = \sum_{j=1}^{n} g_j x_j + \sum_{j=1}^{n} u_j y_j \Longrightarrow \text{Min!}$$

mit

$y_j=1$ Variante j wird verwendet

$y_j=0$ Variante j wird nicht verwendet

g_j mengenproportionale Kosten für Variante j

x_j Menge des nach Variante j zerschnittenen Rohmaterials

u_j Umstellungskosten für Variante j

Zu 3: Mehrfachzuordnungen sollten in der Realisierungsphase verhindert werden.
Dazu gibt es mehrere Vorgehensweisen:

- Reduzierung der Fahrzeugkapazitäten um einen festen Betrag von vornherein,
- Ausgleich über eingeführte, fiktive Bedarfsmengen.

2. Vorstellung des PP DISKO

2.1. Allgemeine Bemerkungen

Das PP DISKO gehört zur Palette der vom VEB Robotron ZFT für ESER-Anlagen (insbesondere ES 1040 und ES 1055) mit dem Betriebssystem OS/ES bereitgestellten Systemunterlagen für mathematische Verfahren. Mit ihm können Probleme der diskreten Optimierung gelöst werden.
In diesem Kapitel werden in kurzer Form die Konzeption, die Ein- und Ausgabe des PP DISKO sowie die möglichen Problemlösungen beschrieben. Die konkrete Anwendung des PP wird (in Verbindung mit Kapitel 4) für das Transportproblem und das 0-1-Problem dargestellt. Die Darlegungen gewährleisten die Lösung von Transportproblemen oder 0-1-Problemen in einfacher Form, ohne Dokumentationen des PP zu verwenden.
Bezüglich einer ausführlichen Beschreibung des PP DISKO, insbesondere hinsichtlich seiner vollen Variabilität, wird auf die Dokumentationen des PP verwiesen:

- Anwendungsbeschreibung (Übersichtsinformationen zum PP) — H 4113-2002 [1]
- Anleitung für den Programmierer, Teil 1 (Beschreibung der Anwendungsprogramme des PP) — H 4113-2004
- Anleitung für den Programmierer, Teil 2 (spezielle Informationen zum PP sowie Informationen zu einigen Moduln des PP) — H 4113-2008
- Nachrichten (Beschreibung der Nachrichten des PP) — H 4113-2006
- Anleitung für den Systemverantwortlichen (Informationen für die Übernahme des PP in Programmbibliotheken einer EDVA) — H 4113-2003

2.2. Konzeption und Inhalt des PP DISKO

Mit dem PP DISKO können die in Kapitel 1 beschriebenen Probleme der diskreten Optimierung gelöst werden.

---

1) Bestell- bzw. Klassifikationsnummer

Dem Anwender werden zur Lösung seiner Probleme insbesondere sogenannte Anwendungsprogramme zur Verfügung gestellt. Bei Lösung eines Problems mittels einem Anwendungsprogramm ist seitens des Anwenders keine Programmierung durchzuführen. Von ihm sind nur

- das Problem beschreibende Eingangsinformationen (Problemdaten) sowie
- Jobsteueranweisungen für das Betriebssystem der EDVA

bereitzustellen.
Mit einem Anwendungsprogramm kann die Lösung eines mathematischen Problems (z.B. Rundfahrtoptimierung) oder eines rechentechnischen Problems (z.B. Einlesen von Daten und Abspeicherung der Daten in einer Plattendatei) erfolgen. Für die vollständige Lösung einer Aufgabe (Dateneingabe, Optimierung, Ergebnisausgabe) mit dem PP DISKO müssen im allgemeinen mehrere Anwendungsprogramme abgearbeitet werden. Die Übergabe von Daten zwischen zwei nacheinander abzuarbeitenden Anwendungsprogrammen erfolgt über EDO-Dateien.
EDO-Dateien sind Dateien auf Magnetplatte, die für die bereitgestellten Systemunterlagen für mathematische Optimierung, Simulation und mathematische Statistik nach einheitlichen Prinzipien aufgebaut sind.
Für die Optimierung werden im PP DISKO die in Tabelle 2.1. dargestellten Anwendungsprogramme bereitgestellt.

Programm-name	zu lösende Probleme	Bemerkungen zur Lösung
VD#GG	Gemischt-ganzzahlige lineare Optimierungsprobleme	Verzweigungsverfahren nach LAND/DOIG, DAKIN und DRIEBEEK; wahlweise exakte oder näherungsweise Lösung
VD#NVA1	Lineare 0-1-Optimierungsprobleme	Verzweigungsverfahren nach LEMKE/SPIELBERG und BALAS; wahlweise exakte oder näherungsweise Lösung
VD#RLT	Rundfahrtprobleme	Verzweigungsverfahren nach LITTLE; wahlweise exakte oder näherungsweise Lösung
VD#TD1	Transportprobleme und verwandte Probleme	Algorithmus von DANTZIG; exakte Lösung
VD#CSA1 VD#CSA2 VD#CUA1 VD#CUA2	Tourenprobleme	Heuristischer Algorithmus nach CLARKE/WRIGHT; näherungsweise Lösung

Tabelle 2.1.: Anwendungsprogramme des PP DISKO für Optimierungsprobleme

Die für die in Tabelle 2.1. angegebenen Anwendungsprogramme notwendigen Problemdaten zur Beschreibung des zu lösenden Problems gruppieren sich im allgemeinen in

- Parameter und
- in Matrizen oder Vektoren zusammengefaßte numerische Daten, einschließlich ggf. dazugehöriger Zeilen- und Spaltenbezeichnungen.

Neben Parametern zur Beschreibung des Problems sind in den Anwendungsprogrammen des PP u.a. auch Parameter zur Steuerung des betreffenden Algorithmus vorhanden.
Alle Anwendungsprogramme des PP DISKO bedingen die Vorgabe von Parametern. Ihre Eingabe erfolgt in Form von Schlüsselwortparametern im Lochkartenformat. Die Eingabe ist dabei nur

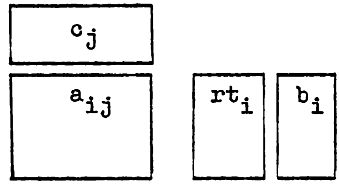
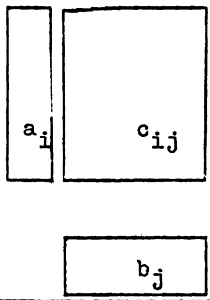
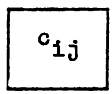
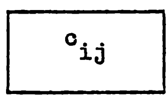
erforderlich, wenn eine Abweichung von einem programmintern festgelegten Standardwert sinnvoll bzw. erforderlich ist. Die Eingabe erfolgt in der Form

parametername = wert   oder
parametername = 'zeichenkette'.

Die Trennung von Parametern auf einer Lochkarte erfolgt durch Kommata. Parameterkarten für ein Anwendungsprogramm werden in eine &PARAM-Karte und eine &END-Karte eingeschlossen. Die Lochung der Karten erfolgt jeweils ab Spalte 2.

Die in Matrizen oder Vektoren zusammengefaßten Problemdaten werden von den in Tabelle 2.1. angegebenen Anwendungsprogrammen in EDO-Dateien vorausgesetzt, wobei eine Matrix oder ein Vektor jeweils in einer Dateneinheit einer EDO-Datei abgespeichert ist.
Bild 2.1. enthält eine schematische Darstellung der notwendigen Problemdaten in Matrix- oder Vektorform für die mittels dem PP DISKO zu lösenden Optimierungsprobleme.

Mathematisches Modell	Notwendige Problemdaten in Matrix- oder Vektorform
Ganzzahlige lineare Optimierungsprobleme:	
$\sum_j c_j x_j = \text{Extr.!}$	c_j
$\sum_j a_{ij} x_j \lesseqqgtr b_i$	a_{ij} rt_i b_i
$x_j \geqq d_j^u$	d_j^u
$x_j \leqq d_j^o$	d_j^o
$x_j \geqq 0$	vt_j
x_k ganz für $k \in N_G$	

Lineare 0-1-Optimierungsprobleme: $\sum_j c_j x_j$ = Extr.! $\sum_j a_{ij} x_j \lesseqqgtr b_i$ $x_j = 0,1$	
Transportprobleme: $\sum_i \sum_j c_{ij} x_{ij}$ = Extr.! $\sum_j x_{ij} \lesseqqgtr a_i$ $x_{ij} \lesseqqgtr b_j$ $x_{ij} \geqq 0$	
Rundfahrtprobleme	
Tourenprobleme	

<u>Bild 2.1.:</u> Notwendige Problemdaten in Matrix- oder Vektorform

Mit dem Vektor $rt=(rt_i)$ wird der Typ der Nebenbedingungen des Optimierungsproblems beschrieben. Es gilt

rt_i = 1 Nebenbedingung i ist eine '$\geqq$'-Bedingung,
= −2 Nebenbedingung i ist eine '$\leqq$'-Bedingung,
= 0 Nebenbedingung i ist eine Gleichung,

= 3 Nebenbedingung i wird nicht in die Lösung des Problems einbezogen.

Vektor $vt=(vt_j)$ beschreibt den Typ der Variablen. Dabei gilt:

vt_j = 1 Variable x_j ist ganzzahlig,

= 0 Variable x_j ist kontinuierlich.

Für das Einlesen von Daten in Matrix- oder Vektorform und ihre Speicherung in EDO-Dateien sowie ihre Änderung steht im PP DISKO das Anwendungsprogramm VD#DEM zur Verfügung. Die Bereitstellung dieser Daten in EDO-Dateien kann aber z.B. auch durch anwendereigene Programme erfolgen.

Die Ausgabe des PP DISKO erfolgt wahlweise über Drucker und in EDO-Dateien. Auf Drucklisten werden insbesondere ausgegeben

- Lösungsinformationen (Ergebnisse) und
- Nachrichten, die ggf. über Fehler oder über den Rechenablauf informieren.

Der Umfang der Druckausgabe kann mittels einem Parameter gesteuert werden. In EDO-Dateien werden Lösungsinformationen ausgegeben, die durch den Anwender maschinell weiterverarbeitet werden können.

Bild 2.2. enthält die graphische Darstellung für ein häufiges Vorgehen bei Lösung eines Optimierungsproblems mit dem PP DISKO

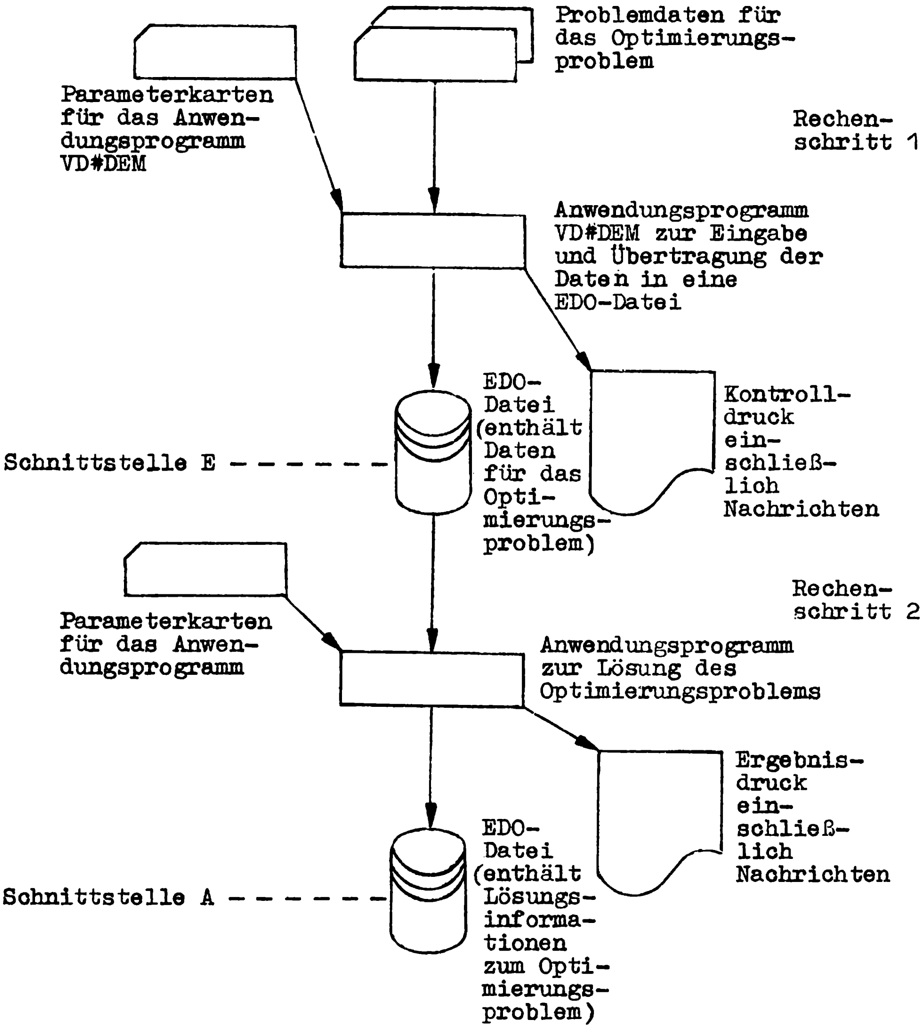


Bild 2.2.: Beispiel für Anwendung des PP DISKO

Im Rechenschritt 1 werden mittels dem Anwendungsprogramm VD#DEM die notwendigen Problemdaten in Matrix- und Vektorform, die das zu lösende Optimierungsproblem beschreiben, eingelesen und in eine EDO-Datei gespeichert. Im Rechenschritt 2 werden mit einem der ersten vier in Tabelle 2.1. genannten Anwendungsprogramme das Optimierungsproblem gelöst und entsprechende Lösungsinformationen ausgegeben. (Die Lösung des Tourenproblems bedingt die aufeinanderfolgende Abarbeitung von mehr als zwei Anwendungsprogrammen.)
Zwischen Rechenschritt 1 und 2 wird bei der Lösung praktischer Probleme in der Regel eine zeitliche Unterbrechung gegeben sein, da die Richtigkeit der in der EDO-Datei gespeicherten Daten zu überprüfen ist bzw. entsprechende Korrekturen (mittels VD#DEM) durchzuführen sind.

Durch E und A sind im Bild 2.2. die Schnittstellen für eine <u>datenseitige Kopplung des PP DISKO</u> mit anwendereigenen Programmen oder anderen Systemunterlagen gekennzeichnet. Inhaltlich sind insbesondere folgende drei Kopplungsmöglichkeiten für das PP DISKO gegeben:
Schnittstelle E:

- Eingabe von notwendigen Problemdaten für ein Optimierungsproblem in eine EDO-Datei mittels einem anwendereigenen Programm oder einer anderen Systemunterlage (Rechenschritt 1) und Lösung des Optimierungsproblems mittels PP DISKO (Rechenschritt 2).

- Eingabe von notwendigen Problemdaten für ein Optimierungsproblem (mit kontinuierlichen Variablen) in eine EDO-Datei mittels dem Anwendungsprogramm VD#DEM des PP DISKO (Rechenschritt 1) und Lösung des Optimierungsproblems mit dem PS OPSI (Rechenschritt 2).

Schnittstelle A:

- Abspeicherung von Lösungsinformationen in eine EDO-Datei durch ein Anwendungsprogramm des PP DISKO und Weiterverarbeitung dieser Informationen durch ein anwendereigenes Programm.

In Tabelle 2.2. sind die Anwendungsprogramme des PP DISKO für Eingabe, Ausgabe und Datenaufbereitung angegeben.

Programm-name	Problem
VD#DEM	Eingabe von Matrizen und Vektoren in eine EDO-Datei und Änderung
VD#DDR	Ausgabe (Drucken, Stanzen) von Matrizen und Vektoren aus einer EDO-Datei
VD#DET	Eingabe und Änderung von Textdaten in eine EDO-Datei und Ausgabe (Drucken, Stanzen) dieser Daten
VD#KMA1 VD#KMA2 VD#KBA1 VD#KBA2	Berechnung der Entfernungsmatrix auf der Basis vorgegebener direkter Entfernungen und Abspeicherung in einer EDO-Datei
VD#KKM	Berechnung der Entfernungsmatrix auf der Basis vorgegebener Koordinaten und Abspeicherung in einer EDO-Datei
VD#CFK	Eingabe von Kunden- und Fuhrparkdaten für ein zu lösendes Tourenproblem
VD#CDR	Ergebnisdruck für ein Tourenproblem

Tabelle 2.2.: Anwendungsprogramme des PP DISKO für Eingabe, Ausgabe und Datenaufbereitung

2.3. Anwendungsprogramm VD#DEM - Eingabe und Änderung von Matrizen und Vektoren

Das Anwendungsprogramm VD#DEM dient zur Eingabe und Änderung von Matrizen (ein Vektor wird im weiteren als Matrix mit einer Zeile oder Spalte aufgefaßt) in EDO-Dateien. Die notwendigen Daten werden im Lochkartenformat eingelesen. Die Speicherung einer Matrix erfolgt zeilen- oder spaltenweise in einer Dateneinheit einer EDO-Datei.
Wichtige Parameter sind u.a.:

KORR= $\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$ (Standard: 0)

Es muß KORR=0 sein, wenn eine Matrix eingelesen werden soll. KORR ist gleich 1 anzugeben, wenn eine in einer EDO-Datei gespeicherte Matrix geändert werden soll.

Q= 'dename'

dename ist der Name der Dateneinheit, in die die Matrix gespeichert werden soll bzw. in der die zu ändernde Matrix gespeichert ist. Der Name darf maximal 8 Zeichen lang sein.

M=zeilenanzahl (Standard: 1)

zeilenanzahl ist die Zeilenanzahl der Matrix.

N=spaltenanzahl (Standard: 1)

spaltenanzahl ist die Spaltenanzahl der Matrix.

SPCH= $\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$ (Standard: 0)

Es muß SPCH=0 sein, wenn die Matrix zeilenweise gespeichert werden soll oder wenn die zu ändernde Matrix zeilenweise gespeichert ist. Andernfalls ist SPCH=1 anzugeben.

INIT=initwert (Standard: 0)

Allen nicht eingelesenen Matrixelementen wird (falls KORR=0) der Wert initwert zugeordnet.

Für den Fall KORR=1 ist die Angabe der Parameter M, N und INIT nicht erforderlich.

Den Parameterkarten folgen die für die Matrix notwendigen Datenkarten. Dies können sein:

- Bezeichnungskarten (mit Zeilen- und Spaltenbezeichnungen),
- Matrixdatenkarten (mit Matrixelementen) und
- Prüfsummenkarten (mit Prüfsummen für Zeilen bzw. Spalten).

Die Eingabe der Daten kann in verschiedenen Formen erfolgen. Im folgenden wird nur die 'indexgesteuerte' Eingabe von Matrixdaten beschrieben.

Den Matrixdatenkarten geht eine Vorlaufkarte mit folgendem Inhalt voraus:

Spalte 1: Z oder S (Z für SPCH=0, S für SPCH=1)
Spalte 2: I

Eine Matrixdatenkarte hat folgenden Inhalt:

Spalte 1:	Z oder S (entsprechend Vorlaufkarte)
Spalte 2:	Anzahl der Matrixelemente in dieser Karte (1,2,3,4 oder 5)
Spalte 3-10:	Zeilenindex i
Spalte 13-20:	Spaltenindex j
Spalte 21-30:	Wert für das Matrixelement (i,j)
Spalte 31-40:	Wert für das Matrixelement (i,j+1) bzw. (i+1,j)
Spalte 41-50:	Wert für das Matrixelement (i,j+2) bzw. (i+2,j)
Spalte 51-60:	Wert für das Matrixelement (i,j+3) bzw. (i+3,j)
Spalte 61-70:	Wert für das Matrixelement (i,j+4) bzw. (i+4,j)

Indizes und ganzzahlige Werte sind rechtsbündig abzulochen. Reelle Werte sind mit einem Punkt anstelle des Dezimalkommas abzulochen. Es werden jeweils so viele Werte von einer Karte von links nach rechts gelesen, wie in Spalte 2 angegeben. Der in Spalte 31-40 abgelochte Wert entspricht dem Matrixelement (i,j+1), falls in Spalte 1 Z abgelocht ist, andernfalls entspricht der Wert dem Element (i+1,j). Analoges gilt für die folgenden Werte. Eine zeilenweise Abspeicherung einer Matrix in einer EDO-Datei (SPCH=0) bedingt also eine zeilenweise Ablochung der Matrix, eine spaltenweise Abspeicherung (SPCH=1) eine spaltenweise Ablochung.

Der Kartenstrom für die Eingabe bzw. Änderung einer Matrix mit VD#DEM hat folgenden Aufbau:

```
     :
     :      Jobsteueranweisungen
&PARAM
     :
     :      Parameterkarten
&END
     :      Vorlaufkarte und
     :      Matrixdatenkarten
/*
```

Mittels VD#DEM können in einem Rechenlauf auch mehrere Matrizen in EDO-Dateien eingegeben bzw. geändert werden.
In diesem Falle ist die Abschlußkarte /* durch eine &-Karte (& in Spalte 2) zu ersetzen. Anschließend folgen wieder Parameterkarten sowie Vorlauf- und Matrixdatenkarten für eine Matrix.
Die &-Karte zeigt jeweils an, daß Parameter- und Datenkarten für eine weitere einzugebende bzw. zu ändernde Matrix folgen.

Kontrollfragen:

1. Es soll ein Zeilenvektor mit der Dimension 2ØØ, der ausschließlich Werte gleich -2 enthält, mittels VD#DEM in die Dateneinheit HALLE einer EDO-Datei eingegeben werden. Wie lauten die notwendigen Parameter- und Datenkarten unter der Zielstellung, daß möglichst wenige Karten eingelesen werden sollen?

2. Eine in einer Dateneinheit BERLIN spaltenweise abgespeicherte Matrix soll mittels VD#DEM korrigiert werden. Für das Element (121,3) ist der richtige Wert -997 und für das Element (5,71) der richtige Wert 88 abzuspeichern. Wie lauten die für VD#DEM erforderlichen Parameter- und Datenkarten?

Lösungen:

Zu 1: Die notwendigen Parameterkarten sind:

```
&PARAM
Q='HALLE', N=2ØØ, INIT=-2
&END
```

Datenkarten sind nicht einzulesen, da durch Vorgabe von INIT=-2 erreicht wird, daß alle (nicht eingelesenen) Elemente des Vektors in der Dateneinheit gleich -2 gesetzt werden.

Zu 2: &PARAM

```
KORR=1
Q='BERLIN', SPCH=1
&END
SI
S1     121     3     -997
S1       5    71       88
```

2.4. Anwendungsprogramm VD#TD1 - Lösung von Transportproblemen

Das Anwendungsprogramm VD#TD1 dient zur Lösung von Transportproblemen. In einer EDO-Datei müssen folgende Daten als Matrizen oder Vektoren zur Verfügung stehen (s. Bild 2.1.):

1. Entfernungsmatrix $\underline{C}=(c_{ij})$ zeilenweise gespeichert,
2. Aufkommen (Angebot) $\underline{a}=(a_i)$ als Spaltenvektor,
3. Bedarf $\underline{b}=(b_j)$ als Zeilenvektor.

Diese Daten werden intern als ganzzahlige Werte verarbeitet. Sind sie reellwertig gespeichert, wird nur mit dem ganzzahligen Anteil gerechnet, d.h. die Dezimalstellen entfallen. Die zulässigen Wertebereiche sind:

$$-100\,000 \leq c_{ij} \leq 100\,000,$$
$$0 \leq a_i \leq 1000\,000,$$
$$0 \leq b_j \leq 1000\,000.$$

Wichtige Parameter sind u.a.:

ZTYP= $\begin{Bmatrix} -2 \\ 1 \end{Bmatrix}$ (Standard: -2)

Soll die Zielfunktion minimiert werden, muß ZTYP=-2 sein, bei einer Maximierung ist ZTYP=1 anzugeben.

RTA= $\begin{Bmatrix} \emptyset \\ 1 \\ -2 \end{Bmatrix}$ (Standard: 0)

Dieser Parameter legt den Typ der Nebenbedingungen bezüglich dem Angebot wie folgt fest:

RTA= Ø bedeutet $\sum_j x_{ij} = a_i$,

RTA= 1 bedeutet $\sum_j x_{ij} \geqq a_i$,

RTA= -2 bedeutet $\sum_j x_{ij} \leqq a_i$.

RTB= $\begin{Bmatrix} Ø \\ 1 \\ -2 \end{Bmatrix}$ (Standard: 0)

Dieser Parameter legt den Typ der Nebenbedingungen bezüglich dem Bedarf wie folgt fest:

RTB= Ø bedeutet $\sum_i x_{ij} = b_j$,

RTB= 1 bedeutet $\sum_i x_{ij} \geqq b_j$,

RTB= -2 bedeutet $\sum_i x_{ij} \leqq b_j$.

Sind RTA=Ø, RTB=Ø und $\sum_i a_i \neq \sum_j b_j$, so wird programmintern folgende Parameterkorrektur durchgeführt:

RTA= -2 für $\sum_i a_i > \sum_j b_j$ bzw. RTB= -2 für $\sum_i a_i < \sum_j b_j$.

A='adename'

adename ist der Name der Dateneinheit, in der das Angebot gespeichert ist.

B='bdename'

bdename ist der Name der Dateneinheit, in der der Bedarf gespeichert ist.

C='cdename'

cdename ist der Name der Dateneinheit, in der die Entfernungsmatrix gespeichert ist.

Für die berechnete optimale Lösung werden geordnet nach Versendern für jede Verbindung zwischen Versender und Empfänger

- die transportierte Menge,
- die Entfernung und
- die Kosten

über Drucker ausgegeben.
Der notwendige Kartenstrom für VD#TD1 hat folgenden Aufbau:

```
  :          Jobsteueranweisungen
&PARAM
  :          Parameterkarten
&END
/*
```

2.5. Anwendungsprogramm VD#NVA1 - Lösung von linearen 0-1-Optimierungsproblemen

Das Anwendungsprogramm VD#NVA1 dient zur wahlweise exakten oder näherungsweisen Lösung von linearen 0-1-Optimierungsproblemen.
Folgende Daten müssen als Matrizen oder Vektoren in einer EDO-Datei zur Verfügung stehen (s. Bild 2.1.):

- Zielfunktion $\underline{c}=(c_j)$ als Zeilenvektor,
- Rechte Seite $\underline{b}=(b_i)$ als Spaltenvektor,
- Problemmatrix $\underline{A}=(a_{ij})$ spaltenweise gespeichert,
- Nebenbedingungstyp $\underline{rt}=(rt_i)$ als Spaltenvektor.

Ist $rt_i=3$, wird die Nebenbedingung i nicht in die Lösung des Problems einbezogen. Beim Ergebnisdruck werden aber bezüglich dieser Nebenbedingung Informationen (z.B. $\sum_j a_{ij} x_j$) mit ausgegeben.
Die Daten für Zielfunktion, rechte Seite und Problemmatrix werden von VD#NVA1 ganzzahlig verarbeitet. Sofern sie in der EDO-Datei reellwertig abgespeichert sind, wird nur mit dem

ganzzahligen Anteil gerechnet, d.h. die Dezimalstellen werden nicht berücksichtigt. Die zulässigen Wertbereiche sind: -2147483647 ... 2147483647.

Wichtige Parameter sind u.a.:

$ZTYP=\begin{Bmatrix} 1 \\ -2 \end{Bmatrix}$ (Standard: -2)

Soll die Zielfunktion minimiert werden, muß ZTYP=-2 sein, bei einer Maximierung ist ZTYP=1 anzugeben.

Z2=gzf (Standard: 2.10^9 für Maximumproblem, -2.10^9 für Minimumproblem)

Der Parameter Z2 gibt die geforderte Güte für die zu berechnende beste zulässige Lösung an. Die Rechnung wird beendet, wenn eine zulässige Lösung mit einem Zielfunktionswert, der nicht größer (Minimumproblem) bzw. nicht kleiner (Maximumproblem) als gzf ist, gefunden worden ist.

Zulässiger Wertbereich:

$-2147483647 \leqq gzf \leqq 2147483647$ und ganz.

A='adename'

adename ist der Name der Dateneinheit, in der die Problemmatrix gespeichert ist.

B='bdename'

bdename ist der Name der Dateneinheit, in der die rechte Seite gespeichert ist.

C='cdename'

cdename ist der Name der Dateneinheit, in der die Zielfunktion gespeichert ist.

CZIX=cnr

Die Angabe des Parameters CZIX ist nur erforderlich, wenn in der Dateneinheit cdename eine Matrix mit mehr als einer Zeile gespeichert ist. Die Zeile cnr der Matrix wird als Zielfunktion verwendet.

RT='rtdename'

rtdename ist der Name der Dateneinheit, in der der Vektor mit den Informationen zum Nebenbedingungstyp gespeichert ist.

Bis zur Berechnung einer optimalen Lösung oder einer zulässigen Lösung geforderter Güte werden in der Regel mehrere zulässige Lösungen berechnet. Für jede zulässige Lösung werden folgende Informationen über Drucker ausgegeben:

- Zielfunktionswert,
- Abweichung des Zielfunktionswertes von Z2,
- Indizes der gleich 1 gesetzten Variablen,
- Aktivitäten $\sum_j a_{ij} x_j$ für $i=1(1)m$,
- Differenzen $\left| b_i - \sum_j a_{ij} x_j \right|$ für $i=1(1)m$.

Der notwendige Kartenstrom für VD#NVA1 hat folgenden Aufbau:

```
    :
    :        Jobsteueranweisungen
&PARAM
    :
    :        Parameterkarten
&END
/*
```

Kontrollfrage:

Für ein mittels VD#NVA1 zu lösendes lineares 0-1-Optimierungsproblem mit 50 Nebenbedingungen sind 3 Zielfunktionen gegeben. Welches Vorgehen gewährleistet, daß bei der Optimierung nach einer dieser Zielfunktionen bei der Ausgabe von zulässigen Lösungen auch die jeweiligen Zielfunktionswerte der anderen beiden Zielfunktionen mit ausgegeben werden?

Lösung:

Die Zielfunktionen werden in das Nebenbedingungssystem des Problems einbezogen, z.B.

Zielfunktion 1 als Nebenbedingung 51,
Zielfunktion 2 als Nebenbedingung 52 und

Zielfunktion 3 als Nebenbedingung 53.

Die Elemente b_{51}, b_{52} und b_{53} der rechten Seite werden gleich Ø gesetzt, rt_{51}, rt_{52} und rt_{53} zur Kennzeichnung des Typs der Nebenbedingungen 51, 52 und 53 gleich 3. Die für eine zulässige Lösung für die Nebenbedingungen 51, 52 und 53 ausgegebenen Aktivitäten ($\sum_j a_{ij} x_j = \sum_j c_j x_j$) sind die Werte der entsprechenden Zielfunktionen.

2.6. Jobsteueranweisungen für ein Anwendungsprogramm

Für die Anwendungsprogramme des PP DISKO ist im allgemeinen der folgende Satz von Steueranweisungen erforderlich:

```
Spalte 1                Spalte 16
//          EXEC        PGM=progname
//STEPLIB   DD          DSN=VD1SOPS.LAD,VOL=SER=VD1GEN,
//                      UNIT=SYSDA,DISP=SHR
//VDISKO    DD          DSN=abcdat,VOL=SER=platte,UNIT=SYSDA,
//                      DISP=SHR
//FTØ6FØØ1  DD          SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,
                        BLKSIZE=6Ø5)
//FTØ5FØØ1  DD          *
```

Durch die EXEC-Anweisung wird das Anwendungsprogramm mit dem Namen progname aufgerufen. Durch die STEPLIB-Anweisung wird die Bibliothek vereinbart, in der das gerufene Programm enthalten ist. Die in dieser Anweisung enthaltenen Angaben setzen voraus, daß die Übernahme des PP DISKO auf die betreffende EDVA entsprechend der Dokumentation 'Anleitung für den Systemverantwortlichen' erfolgt ist.
Mit der VDISKO-Anweisung wird die EDO-Datei definiert, die die Problemdaten in Matrix- oder Vektorform enthält oder in die diese gespeichert werden sollen. abcdat ist der Name der EDO-Datei und platte die Datenträgerarchivnummer der Magnetplatte, auf der sich die Datei befindet bzw. befinden soll. Ist die Datei noch nicht vorhanden, d.h. muß sie erst noch

gegründet werden, so ist die Angabe DISP=SHR durch DISP=(NEW, KEEP) zu ersetzen und zusätzlich der Parameter SPACE zur Festlegung der Dateigröße anzugeben.
Damit lautet die zweite Karte der VDISKO-Anweisung

```
//          DISP=(NEW,KEEP),SPACE=(...)
```

Die Angabe von SPACE=(TRK,(2Ø,,1)) gewährleistet in der Regel, daß in der betreffenden EDO-Datei z.B. eine (100x100)-Matrix und vier Vektoren mit jeweils 100 Werten gespeichert werden können. Durch die FTØ6FØØ1-Anweisung wird die Druckausgabe definiert. Die FTØ5FØØ1-Anweisung zeigt an, daß unmittelbar hinter dieser Karte Parameterkarten und ggf. Datenkarten folgen.
Zusätzlich zu den hier angegebenen Steueranweisungen sind bei einigen Anwendungsprogrammen noch eine oder mehrere DD-Anweisungen notwendig, um erforderliche externe Dateien zu definieren. So ist z.B. bei Anwendung von VD#TD1 (nach Beschreibung unter 2.4.) zusätzlich die Anweisung

```
//FTØ8FØØ1  DD   UNIT=SYSDA,SPACE=(3624,13Ø)
```

erforderlich.

3. Ausgewählte Anwendungsbeispiele

Ausgehend von den in Punkt 1. des Lehrbriefes vorgestellten ökonomischen Aufgabenstellungen sollen zwei

- das Transportverteilungsproblem und
- das Problem der Grundfondsreproduktion

ausführlich betrachtet werden.
Diese Problemstellungen wurden u.a. nach folgenden Gesichtspunkten gewählt:

- hohe volkswirtschaftliche Bedeutung
- Verwendung unterschiedlicher mathematischer Lösungsalgorithmen
- Vorliegen zahlreicher praktischer Anwendungsbeispiele auf dem Gebiet der Transportoptimierung.

3.1. Transportoptimierung (Verteilungsproblem)

3.1.1. Ökonomische Aufgabenstellung

Ausgehend von der Forderung, die Transportbeziehungen auf das notwendige Maß zu reduzieren, wird ein Transportverteilungsproblem untersucht, das einem Bereich der Volkswirtschaft entstammt, in dem die Anwendung mathematischer Methoden, speziell der Methoden der Transportoptimierung, recht umfassend vorgenommen werden kann. Es handelt sich hierbei um ein Problem aus dem Bereich des Bauwesens.
Eine der Hauptanforderungen an das zu transportierende Gut, Homogenität im Sinne von Austauschbarkeit der einzelnen Gutarten, ist bei einer großen Zahl von Gütern, die zwischen den Betrieben des Bauwesens transportiert werden müssen, erfüllt. Deshalb zählen Baustoffe neben einigen anderen Gütern auch zu den Gutarten, deren Transportrealisierung sich für eine Optimierungsrechnung förmlich anbietet.
Im Beispiel wird die optimale Verteilung von Betonkies einer bestimmten Körnung behandelt.
Die Untersuchung erstreckt sich sowohl hersteller- als auch

verbraucherseitig auf einen ausgewählten Bezirk der DDR.
Im Zusammenhang mit den auftretenden Aufkommens- und Bedarfsorten soll erwähnt werden, daß, abhängig vom speziellen Problem, die Möglichkeit besteht, mehrere Aufkommens- bzw. Bedarfsorte zu jeweils einem "Punkt" zusammenzufassen. Dabei sollte man sich von der Größe des Gesamtproblems, von geografischen Gesichtspunkten, von Produktionshöhen der einzelnen Untersuchungsobjekte und weiteren Merkmalen leiten lassen.
Auf jeden Fall kann so die Problemgröße insgesamt reduziert werden, wenn dadurch auch Ungenauigkeiten in die Lösung Eingang finden. Das Vorgehen wird von Fall zu Fall unterschiedlich sein. Zur Charakterisierung des Transportaufwandes für den Transport einer Guteinheit von den Aufkommensorten zu den Bedarfsorten können unterschiedliche Größen Verwendung finden. Am häufigsten werden die Transportaufwendungen als

- Transportentfernung,
- Transportkosten,
- Transportzeit

formuliert.

3.1.2. Ermittlung der Problemdaten

In dieser Phase der Untersuchung sind alle Daten zu analysieren, die für eine möglichst wirklichkeitsnahe Darstellung des Problems in der Modellierungsetappe relevant sind.
Diese Arbeiten stehen in enger Wechselwirkung mit der mathematischen Modellierung, die im nachfolgenden Punkt behandelt wird.
Es müssen Aussagen getroffen werden zu den

- Aufkommensorten A_i ; $i=1(1)m$ und deren
  Aufkommen a_i ; $i=1(1)m$,
- Bedarfsorten B_j ; $j=1(1)n$ und deren
  Bedarf b_j ; $j=1(1)n$,
- Transportaufwendungen pro zu transportierende Guteinheit
  c_{ij} ; $i=1(1)m$, $j=1(1)n$.

In unserem Beispiel treten als Aufkommensorte sechs Kiesgruben A_i ; $i=1(1)6$ auf, die unter anderem jeweils Betonkies einer ausgewählten TGL-gerechten Körnung produzieren. Die Kiesgruben sind über das Territorium eines ausgewählten Bezirkes verteilt und haben im Untersuchungszeitraum ein Aufkommen a_i ; $i=1(1)6$, an Betonkies in Höhe von:

Aufkommen	Menge [kkg]
a_1	51,00
a_2	35,00
a_3	4,00
a_4	1,60
a_5	5,00
a_6	6,00

Tabelle 3.1.: Aufkommensmengen

Als Bedarfsorte haben wir im Beispiel 15 Verarbeitungsbetriebe B_j ; $j=1(1)15$, zu berücksichtigen, die einen Bedarf b_j ; $j=1(1)15$ an dem genannten Betonkies in folgender Höhe haben:

Bedarf	Menge [kkg]
b_1	0,10
b_2	3,00
b_3	1,00
b_4	0,10
b_5	0,70
b_6	3,00
b_7	0,10
b_8	6,00
b_9	2,50
b_{10}	5,67
b_{11}	8,83
b_{12}	18,60
b_{13}	11,00
b_{14}	20,00
b_{15}	22,00

Tabelle 3.2.: Bedarfsmengen

Aus hier nicht näher zu analysierenden Gründen verwenden wir zur Angabe des Transportaufwandes pro transportierte Guteinheit die Transportentfernung, wobei wir von proportionaler Entwicklung der Transportaufwendungen zur transportierten Menge ausgehen.
Dieses Vorgehen führt uns beim betrachteten Problem zur Formulierung des Optimalitätskriteriums

- Minimum der Gesamttransportentfernung -.

Es wird also nicht die günstigste Einzelverbindung zwischen zwei Punkten gesucht. Es soll vielmehr der zur Realisierung aller notwendigen Verbindungen auftretende Gesamttransportaufwand minimal gestaltet werden.
Die komplette Entfernungsmatrix $\underline{C}_{(6;15)}$ hat folgendes Aussehen:

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}
A_1	137	129	134	130	133	105	71	68	42	25	7	57	60	105	140
A_2	102	97	100	102	98	70	39	38	47	43	45	12	6	51	88
A_3	116	100	47	128	28	19	64	90	102	88	95	61	62	56	115
A_4	58	52	60	67	80	50	36	50	82	71	73	38	36	12	63
A_5	65	81	150	50	159	130	77	63	104	114	128	89	91	105	58
A_6	17	17	118	31	132	111	78	79	114	128	140	108	102	68	28

[Angaben in km]

Tabelle 3.3.: Entfernungsmatrix

3.1.3. Mathematische Modellierung

Unter Zugrundelegung der in den Abschnitten 1. und 3.1.2. gemachten Aussagen kommt man zur Formulierung des mathematischen Modells für das vorgestellte Transportverteilungsproblem:

Zielfunktion:

(1.) $$Z = \sum_{i=1}^{6} \sum_{j=1}^{15} c_{ij} \, x_{ij} \Longrightarrow \text{Min!}$$

Restriktionen:

aufkommensseitig:

(2.1.) $$\sum_{j=1}^{15} x_{ij} = a_i \qquad \text{für } i=1(1)6$$

bedarfsseitig:

(2.2.) $$\sum_{i=1}^{6} x_{ij} = b_j \qquad \text{für } j=1(1)15$$

Gleichgewichtsbedingung:

(2.3.) $$\sum_{i=1}^{6} a_i = \sum_{j=1}^{15} b_j$$

Nichtnegativitätsbedingung:

(2.4.) $$x_{ij} \geqq 0 \quad i=1(1)6,\ j=1(1)15$$

In ausführlicher Schreibweise würde man erhalten:

Zielfunktion:

(1.) $$Z = 137x_{11} + 129x_{12} + 134x_{13} + \ldots + 28x_{6;15} \Longrightarrow \text{Min!}$$

Restriktionen:

$$
\begin{array}{ll}
(2.1.) & x_{11} + x_{12} + x_{13} + \dots + x_{1;15} = 51{,}00 \\
 & x_{21} + x_{22} + x_{23} + \dots + x_{2;15} = 35{,}00 \\
 & x_{31} + x_{32} + x_{33} + \dots + x_{3;15} = 4{,}00 \\
 & \vdots \\
 & x_{61} + x_{62} + x_{63} + \dots + x_{6;15} = 6{,}00
\end{array}
$$

$$
\begin{array}{ll}
(2.2.) & x_{11} + x_{21} + x_{31} + \dots + x_{61} = 0{,}10 \\
 & x_{12} + x_{22} + x_{32} + \dots + x_{62} = 3{,}00 \\
 & x_{13} + x_{23} + x_{33} + \dots + x_{63} = 1{,}00 \\
 & \vdots \\
 & x_{1;15} + x_{2;15} + x_{3;15} + \dots + x_{6;15} = 22{,}00
\end{array}
$$

$$
\begin{array}{ll}
(2.3.) & 51{,}00 + 35{,}00 + 4{,}00 + \dots + 6{,}00 = \\
 & 0{,}10 + 3{,}00 + 1{,}00 + \dots + 22{,}00 \\
 & 102{,}60 = 102{,}60
\end{array}
$$

(2.4.) $x_{ij} \geqq 0 \quad i=1(1)6,\ j=1(1)15.$

Das mathematische Modell zeigt, daß es sich um ein ausgeglichenes Problem handelt, da Bedingung (2.3.) erfüllt ist. Das bedeutet, daß das Gesamtaufkommen an Betonkies in den Kiesgruben dem Gesamtbedarf an Betonkies in den Verarbeitungsbetrieben entspricht.
Auf Möglichkeiten der Lösung praktischer Verteilungsprobleme bei Nichterfülltsein der Bedingung (2.3.) wurde bereits im Punkt 1.3.1. eingegangen.

Kontrollfragen:

1. Welche ökonomischen Auswirkungen hat das Nichterfülltsein der Gleichgewichtsbedingung auf die Lösung praktischer Probleme und deren Realisierung?
2. Wie kann man das Problem bei Berücksichtigung des Vorhandenseins unterschiedlicher Körnungen des Betonkieses in den Aufkommensorten und bei Bedarf an unterschiedlichen Körnungen bei den Bedarfsträgern lösen?
3. Welche Vor- und Nachteile bringt die Verwendung der unterschiedlichen Optimalitätskriterien mit sich?
   Wann sind welche Kriterien am vorteilhaftesten anzuwenden?

Lösungen:

Zu 1: Generell läßt sich ein Transportverteilungsproblem bei Nichterfülltsein der Gleichgewichtsbedingung immer lösen. Sinnvoll scheint jedoch nur der Fall

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \qquad \text{(Überschußproblem)}$$

zu sein, da bei Vorliegen der Bedingung

$$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j \qquad \text{(Mangelproblem)}$$

unter Umständen eine Verteilung vorgenommen werden muß, die nicht unbedingt der optimalen Lösung entspricht. Allgemein muß festgestellt werden, daß das nichtabsetzbare Aufkommen bzw. der nicht zu befriedigende Bedarf dort auftreten, wo, bedingt durch entsprechend ungünstige Aufwandskoeffizienten, keine Zuordnung erfolgt.

Zu 2: Entweder müßte jedes Sortenproblem für sich allein gelöst werden oder das Gesamtproblem über die Lösung eines Mehrsortenproblems bearbeitet werden.
Auf das Mehrsortenproblem sollte im Seminar eingegangen werden.

Zu 3: Die Aufwandsgröße Transportentfernung, die uns zum Optimalitätskriterium Minimum an Gesamttransportentfernung führt, läßt sich aus entsprechendem Kartenmaterial entnehmen, bzw. mit dem Kurvimeter von der Karte direkt ermitteln. Hingegen müssen bei Verwendung der Größe Transportzeit mehrere Messungen pro Verbindung vorgenommen werden, um aussagekräftig zu sein.
Die Verwendung der Aufwandsgröße Transportkosten erscheint nur dann sinnvoll, wenn dabei die nichtlineare Kostenentwicklung bei zunehmender Transportentfernung berücksichtigt wird. Das dürfte jedoch in der Regel schwerfallen. Der Aufwand für solche Untersuchungen wäre sicher zu groß.
Vom Ermittlungsaufwand her bietet sich gewiß die Aufwandsgröße Transportentfernung an.
Eine nicht unbedeutende Rolle spielt bei der Festlegung des Optimalitätskriteriums auch die zu transportierende Gutart. Zum Beispiel könnten bei leichtverderblichen Gütern Transportzeiten verwendet werden.
Neben den genannten Aufwandsgrößen gibt es weitere, die unter Umständen dem geforderten Ziel besser entsprechen. Darauf sollte im Seminar eingegangen werden.

3.2. 0-1-Optimierung (Komplexe Grundfondsplanung)

3.2.1. Ökonomische Aufgabenstellung

Die zur Erfüllung der Hauptaufgabe erforderliche Steigerung der Arbeitsproduktivität wird u.a. in entscheidendem Maß von der Ausstattung der Arbeit mit materiellen Fonds sowie deren rationeller Nutzung bestimmt. Das erfordert von den Kombinaten und Betrieben die Erarbeitung eines Plans der komplexen Grundfondsreproduktion. Hierbei handelt es sich um eine langfristige Aufgabe.
Im ausgewählten Beispiel plant ein Kombinat der chemischen Industrie seine Grundfondsreproduktion.
Gemäß dem Erkenntnisstand zum Zeitpunkt der Planerarbeitung

stehen 9 größere Vorhaben mit insgesamt 24 Reproduktionsvarianten zur Diskussion. Dazu kommen noch verschiedene Kleininvestitionen, die aber in einer Sammelposition zusammengefaßt und in der Untersuchung vernachlässigt werden.
Durch die Optimierung sollen gleichzeitig die zu realisierenden Vorhaben sowie deren Variante ausgewählt werden. Voraussetzung ist dabei, daß höchstens eine Variante je Vorhaben in der Lösung enthalten ist.
Der Planzeitraum betrage mehrere Jahre.
Zunächst ist eine Variantenmatrix aufzustellen. Diese soll für das Beispiel folgende Gestalt haben:

$$\underline{V} = \begin{pmatrix} v_{11} & v_{12} & v_{13} & 0 \\ v_{21} & v_{22} & 0 & 0 \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & 0 & 0 & 0 \\ v_{51} & v_{52} & 0 & 0 \\ v_{61} & v_{62} & v_{63} & 0 \\ v_{71} & v_{72} & v_{73} & 0 \\ v_{81} & v_{82} & 0 & 0 \\ v_{91} & v_{92} & v_{93} & v_{94} \end{pmatrix}$$

Naturgemäß bestehen zwischen den einzelnen Vorhaben mit ihren Varianten Querverbindungen, die bei der Modellierung zu berücksichtigen sind. So sind folgende Forderungen zu beachten:

1. Vorhaben 1 muß realisiert werden.
2. Vorhaben 3 muß realisiert werden.
3. Dann und nur dann, wenn für Vorhaben 2 Variante 2 gewählt wird, muß für Vorhaben 5 Variante 2 gewählt werden.
4. Dann und nur dann, wenn für Vorhaben 1 Variante 3 gewählt wird, muß für Vorhaben 3 Variante 2 realisiert werden.

Als nächstes muß entschieden werden, welche ökonomischen Größen als Restriktionen ins Modell aufgenommen werden. Möglich und sinnvoll sind z.B. folgende

- Investitionsvolumen (finanziell)
- Bauvolumen
- Arbeitskräfte oder Arbeitskräftefondsänderung
- Materialfonds
- obere Absatzschranke
- untere Absatzschranke
- Betriebsflächen
- Devisenfonds
- Außenhandelsrentabilität.

Für das Beispiel wurden ausgewählt:

Investitionsvolumen (finanziell)	b_1
Bauvolumen	b_2
Materialfonds	b_3
Arbeitskräftefondsänderung	b_4
obere Absatzschranke	b_5
untere Absatzschranke	b_6

b_1 stellt das im gesamten Berechnungszeitraum zur Verfügung stehende Investitionsvolumen des Kombinates dar.

b_2 umfaßt die insgesamt zur Verfügung stehende Menge an Bauproduktion (wertmäßig).

b_3 bezeichnet eine Gruppe von Restriktionen. Hier werden Engpaßmaterialien erfaßt, im Beispiel zwei.

b_4 erfaßt die Arbeitskräftefondsänderung aus $\Delta Ak = Ak_n - Ak_o$ mit

Ak_n - Arbeitskräftefonds im Jahr n des Berechnungszeitraumes

Ak_o - Arbeitskräftefonds im Basisjahr.

b_5 stellt die voraussichtliche Absatzmenge im letzten Jahr des Berechnungszeitraumes dar. Hier können Absatzschranken für mehrere Erzeugnisse vorgegeben werden.

Im Beispiel wird nur ein Erzeugnis - das profilbestimmende - einbezogen. Gleiches gilt sinngemäß für Restriktion b_6. Außerdem müssen ein oder mehrere Optimalitätskriterien festgelegt werden. Mögliche Zielfunktionen für Investitionsoptimierungen sind u.a.:

- Maximierung der Investitionsrentabilität
- Maximierung der Arbeitsrentabilität
- Maximierung der Investitionsquote.

Unter der Investitionsrentabilität wird das Verhältnis von Zuwachs an einheitlichem Betriebsergebnis zum Investitionsaufwand verstanden:

$$RI = \frac{\Delta BE}{I} \qquad \left[\frac{M}{M} \cdot a\right]$$

mit

RI	-	Investitionsrentabilität
ΔBE	-	Zuwachs an einheitlichem Betriebsergebnis
I	-	Investitionsvolumen
a	-	Jahr.

Arbeitsrentabilität ist das Verhältnis von einheitlichem Betriebsergebnis zur Anzahl der benötigten Arbeitskräfte:

$$RAK = \frac{BE}{AK} \qquad \left[\frac{M}{VbE \cdot a}\right]$$

mit

RAK	-	Arbeitsrentabilität
BE	-	einheitliches Betriebsergebnis
AK	-	Arbeitskräfte.

Die Investitionsquote erfaßt das Verhältnis zwischen dem Investitionsaufwand und dem Warenproduktionszuwachs

$$QI = \frac{I}{\Delta WP} \qquad \left[\frac{M \cdot a}{M}\right]$$

Im Beispiel werden zwei Zielfunktionen verwendet:

Z_1 = Maximierung der Investitionsrentabilität und

Z_2 = Maximierung der Arbeitsrentabilität.

3.2.2. Ermittlung der Problemdaten

Für das zu erstellende Modell sind die entsprechenden Daten bereitzustellen.
Die rechten Seiten der Restriktionen b_1 bis b_6 sind Tabelle 3.4. zu entnehmen.

	Menge	Dimension
b_1	50	[Mio M]
b_2	22	[Mio M]
b_{31}	15	[k ME]
b_{32}	25	[k ME]
b_4	60	[VbE]
b_5	80	$\left[k \frac{ME}{a}\right]$
b_6	25	$\left[k \frac{ME}{a}\right]$

Tabelle 3.4.: Restriktionen

Die Koeffizienten $a_{ij}^{(k)}$ und die Zielfunktionskoeffizienten $c_{ij}^{(1)}$ bzw. $c_{ij}^{(2)}$ sind in Tabelle 3.5. zusammengefaßt.
Die Bereitstellung dieser Daten erfordert in der Praxis besondere Sorgfalt. Hier ist zu beachten, daß - gemäß den gesetzlichen Bestimmungen - verstärkt Varianten für Investitionsvorhaben auszuarbeiten und zu bewerten sind.
Der erhöhte Aufwand bei der Datenbereitstellung wird durch bessere Lösungen kompensiert.

Vorh. Vw / Restr.	V_{11}	V_{12}	V_{13}	V_{21}	V_{22}	V_{31}	V_{32}	V_{33}	V_{34}	V_{41}	V_{51}	V_{52}
b_1	6	4	5,5	7	6	9,5	10	8,5	11	6	7	7,5
b_2	2,5	1,5	3	4	4	5,5	5	4	5	4	4	3,5
b_{31}	0	0	0	1	0,6	7	6	6	7	0	1	3
b_{32}	0,8	0,5	0,7	0	0	3	2	4	3,5	0	4,2	2,5
b_4	8	8	8	0	0	25	20	30	15	-2	5	6
b_5	0	0	0	8	10	14	15	12	20	0	8	7
b_6	0	0	0	8	10	14	15	12	20	0	8	7
z_1	0,8	0,7	0,9	0,6	0,9	1,0	1,1	0,9	1,0	1,3	0,8	0,7
z_2	2	1,5	2,1	1,9	1,8	2,7	2,6	2,4	1,9	3,0	2,2	2,3

Tabelle 3.5.: Zielfunktions- und Aufwandskoeffizienten

V_{61}	V_{62}	V_{63}	V_{71}	V_{72}	V_{73}	V_{81}	V_{82}	V_{91}	V_{92}	V_{93}	V_{94}	Dim.
8,8	9,5	7,6	5,7	6	3,8	8,5	6	8,2	6,9	9,5	5,5	[Mio M]
3	2	1,2	2,2	3	1,1	2,5	3	3,4	2,0	3,1	1,7	[Mio M]
2,8	3	2,6	1,4	0,9	0,5	0	0	0,5	0,8	1,0	1,1	[k ME]
0	0	0	2,4	3,1	2,8	8,5	7	6,1	5,7	2,1	0,2	[k ME]
10	12	15	9	8	7	-4	0	3	0	2	3	[VbE]
10	8	8,5	7,5	6	4,5	0	0	16	14	22	12	$\left[\frac{\text{k ME}}{\text{a}}\right]$
10	8	8,5	7,5	6	4,5	0	0	16	14	22	12	$\left[\frac{\text{k ME}}{\text{a}}\right]$
1,0	1,3	1,6	1,2	1,4	1,2	0,7	0,6	0,8	0,8	1,4	0,6	$\left[\frac{\text{M}}{\text{VbE a}}\right]$
2,4	2,7	2,9	3,1	2,8	2,5	1,8	1,4	1,7	1,9	2,3	2,1	$\left[\frac{\text{M}}{\text{M a}}\right]$

3.2.3. Mathematische Modellierung

Aus den in 3.2.1. und 3.2.2. gemachten Aussagen ergibt sich das mathematische Modell für das untersuchte Beispiel in allgemeiner Form.

Zielfunktion

Maximierung der Investitionsrentabilität

(1.1.) $$Z_1 = c_{11}^{(1)}x_{11} + c_{12}^{(1)}x_{12} + c_{13}^{(1)}x_{13} + c_{21}^{(1)}x_{21} + \dots + c_{94}^{(1)}x_{94} \Longrightarrow \text{Max!}$$

Maximierung der Arbeitskräfterentabilität

(1.2.) $$Z_2 = c_{11}^{(2)}x_{11} + c_{12}^{(2)}x_{12} + c_{13}^{(2)}x_{13} + c_{21}^{(2)}x_{21} + \dots + c_{94}^{(2)}x_{94} \Longrightarrow \text{Max!}$$

Restriktionen

Investitionsvolumen

(2.1.) $$a_{11}^{(1)}x_{11} + a_{12}^{(1)}x_{12} + \dots + a_{94}^{(1)}x_{94} \leqq b_1$$

Bauvolumen

(2.2.) $$a_{11}^{(2)}x_{11} + a_{12}^{(2)}x_{12} + \dots + a_{94}^{(2)}x_{94} \leqq b_2$$

Material

(2.3.) $$a_{11}^{(3,1)}x_{11} + a_{12}^{(3,1)}x_{12} + \dots + a_{94}^{(3,1)}x_{94} \leqq b_{3,1}$$

(2.4.) $$a_{11}^{(3,2)}x_{11} + a_{12}^{(3,2)}x_{12} + \dots + a_{94}^{(3,2)}x_{94} \leqq b_{3,2}$$

Arbeitskräftefondsänderung

(2.5.) $$a_{11}^{(4)}x_{11} + a_{12}^{(4)}x_{12} + \dots + a_{94}^{(4)}x_{94} \leqq b_4$$

obere Absatzschranke

(2.6.) $a_{11}^{(5)}x_{11} + a_{12}^{(5)}x_{12} + \dots + a_{94}^{(5)}x_{94} \leqq b_5$

untere Absatzschranke

(2.7.) $a_{11}^{(6)}x_{11} + a_{12}^{(6)}x_{12} + \dots + a_{94}^{(6)}x_{94} \geqq b_6$

logische Bedingungen

(2.8.)
$$\begin{aligned}
x_{11} + x_{12} + x_{13} &= 1\\
x_{21} + x_{22} &\leqq 1\\
x_{31} + x_{32} + x_{33} + x_{34} &= 1\\
x_{51} + x_{52} &\leqq 1\\
x_{61} + x_{62} + x_{63} &\leqq 1\\
x_{71} + x_{72} + x_{73} &\leqq 1\\
x_{81} + x_{82} &\leqq 1\\
x_{91} + x_{92} + x_{93} + x_{94} &\leqq 1
\end{aligned}$$

$$x_{22} = x_{52}$$
$$x_{13} = x_{33}$$

Nichtnegativitäts- und Ganzzahligkeitsbedingung

(2.9.) $x_{ij} \in \{0,1\}$

Es bedeuten

$x_{ij} = 1$ Vorhaben i wird mit Variante j realisiert

$x_{ij} = 0$ Vorhaben i wird nicht mit Variante j realisiert

$c_{ij}^{(1)}$ - Investitionsrentabilität des Vorhabens i, falls Variante j realisiert wird

$c_{ij}^{(2)}$ - Arbeitsrentabilität des Vorhabens i, falls Variante j realisiert wird

$a_{ij}^{(k)}$ - Aufwand an Restriktion k für Vorhaben i und Variante j für k=1,2,3.1;3.2,4

$a_{ij}^{(k)}$ - Produktionshöhe des Vorhabens i und der Variante j für k=5,6.

Für die in 3.2.1. gegebene Aufgabenstellung ergibt sich unter Verwendung der in Tabelle 3.4. und 3.5. erfaßten Daten folgendes konkrete Modell:

<u>Zielfunktion</u>

(1.1.) $Z_1 = 0{,}8x_{11} + 0{,}7x_{12} + 0{,}9x_{13} + \ldots + 1{,}4x_{93} + 0{,}6x_{94} \Longrightarrow \text{Max!}$

(1.2.) $Z_2 = 2x_{11} + 1{,}5x_{12} + 2{,}1x_{13} + \ldots + 2{,}3x_{93} + 2{,}1x_{94} \Longrightarrow \text{Max!}$

<u>Restriktionen</u>

a) <u>Ökonomische Restriktionen</u>

(2.1.) $6x_{11} + 4x_{12} + 5{,}5x_{13} + \ldots + 9{,}5x_{93} + 5{,}5x_{94} \leqq 50$

(2.2.) $2{,}5x_{11} + 1{,}5x_{12} + 3x_{13} + \ldots + 3{,}1x_{93} + 1{,}7x_{94} \leqq 22$

(2.3.) $0x_{11} + 0x_{12} + 0x_{13} + \ldots + 1{,}0x_{93} + 1{,}1x_{94} \leqq 15$

(2.4.) $0{,}8x_{11} + 0{,}5x_{12} + 0{,}7x_{13} + \ldots + 2{,}1x_{93} + 0{,}2x_{94} \leqq 25$

(2.5.) $8x_{11} + 8x_{12} + 8x_{13} + \ldots + 2x_{93} + 3x_{94} \leqq 60$

(2.6.) $0x_{11} + 0x_{12} + 0x_{13} + \ldots + 22x_{93} + 12x_{94} \leqq 80$

(2.7.) $0x_{11} + 0x_{12} + 0x_{13} + \ldots + 22x_{93} + 12x_{94} \geqq 25$

b) <u>Logische Restriktionen</u>

(2.8.)
$$
\begin{aligned}
x_{11} + x_{12} + x_{13} &= 1\\
x_{21} + x_{22} &\leqq 1\\
x_{31} + x_{32} + x_{33} + x_{34} &= 1\\
x_{51} + x_{52} &\leqq 1\\
x_{61} + x_{62} + x_{63} &\leqq 1\\
x_{71} + x_{72} + x_{73} &\leqq 1\\
x_{81} + x_{82} &\leqq 1\\
x_{91} + x_{92} + x_{93} + x_{94} &\leqq 1
\end{aligned}
$$

$$x_{22} - x_{52} = 0$$
$$x_{13} - x_{33} = 0$$

c) Nichtnegativitäts- und Ganzzahligkeitsbedingung

(2.9.) $$x_{ij} \in \{0,1\} \qquad \begin{matrix} i=1(1)9 \\ j=1(1)4 \end{matrix}$$

Da zwei Zielfunktionen ausgewählt wurden, erhält man zwei Investitionsprogramme. Diese können unter Beachtung weiterer Kriterien bewertet und verglichen werden.

Kontrollfragen:

1. Folgende Forderungen sind bei der Modellierung zu beachten:
   a) Vorhaben 4 muß realisiert werden.
   b) Bei Realisierung der Variante 1 für Vorhaben 7 muß Variante 3 von Vorhaben 9 gewählt werden.

   Stellen Sie die logischen Bedingungen auf!

2. Als weitere Restriktion soll ein Zwischenprodukt aufgenommen werden, dessen Produktion durch die Investitionsvorhaben gesichert werden soll. Es besteht ein Mindestbedarf an diesem Produkt.
   Stellen Sie die Restriktion in allgemeiner Form ausführlich dar!

3. Es wird gefordert, daß Vorhaben 4 unbedingt zu realisieren ist. Welche Auswirkungen hat diese Forderung auf die Problemgröße? Welche Änderungen treten im Modell auf?

Lösungen:

Zu 1: $x_{41} = 1$

$$x_{71} \leqq x_{93} \longrightarrow x_{71} - x_{93} \leqq 0$$

Zu 2:

$$a_{11}^{(7)}x_{11} + a_{12}^{(7)}x_{12} + a_{13}^{(7)}x_{13} + a_{21}^{(7)}x_{21} + a_{22}^{(7)}x_{22} + a_{31}^{(7)}x_{31}$$

$$+ a_{32}^{(7)}x_{32} + a_{33}^{(7)}x_{33} + a_{34}^{(7)}x_{34} + a_{41}^{(7)}x_{41} + a_{51}^{(7)}x_{51} + a_{52}^{(7)}x_{52}$$

$$+ a_{61}^{(7)}x_{61} + a_{62}^{(7)}x_{62} + a_{63}^{(7)}x_{63} + a_{71}^{(7)}x_{71} + a_{72}^{(7)}x_{72} + a_{73}^{(7)}x_{73}$$

$$+ a_{81}^{(7)}x_{81} + a_{82}^{(7)}x_{82} + a_{91}^{(7)}x_{91} + a_{92}^{(7)}x_{92} + a_{93}^{(7)}x_{93} + a_{94}^{(7)}x_{94}$$

$$\geqq b_7$$

$a_{ij}^{(7)}$ Menge des Zwischenproduktes, die Vorhaben i produzieren kann, falls Variante j realisiert wird

b_7 Mindestbedarf am Zwischenprodukt

Zu 3: $x_{41} = 1$ ist gefordert. Deshalb verringert sich die Anzahl der Variablen von 24 auf 23.

In den Zielfunktionen tauchen statt des Produkts $c_{41}^{(1)}x_{41}$ bzw. $c_{41}^{(2)}x_{41}$ die Konstanten $c_{41}^{(1)}$ bzw. $c_{41}^{(2)}$ auf.

Die rechten Seiten der Restriktionen werden folgendermaßen korrigiert:

$$b_k' = b_k - a_{41}^{(k)}.$$

4. Rechentechnische Realisierung der ausgewählten Beispiele

4.1. Lösung des Transportverteilungsproblems

4.1.1. Programmauswahl

Die Lösung des in Abschnitt 3.1. dargestellten Transportverteilungsproblems erfordert die Abarbeitung der beiden folgenden Anwendungsprogramme in der angegebenen Reihenfolge:

1. VD#DEM Einlesen der Problemdaten (Entfernungsmatrix, Aufkommensvektor und Bedarfsvektor) und deren Speicherung in einer EDO-Datei
2. VD#TD1 Lösung des Transportverteilungsproblems und Ergebnisausgabe

4.1.2. Problemdaten in Matrix- oder Vektorform - EDO-Datei

Das Programm VD#TD1 arbeitet nur mit ganzzahligen Werten für das Aufkommen und den Bedarf, deshalb ist eine geeignete Festlegung der Maßeinheit notwendig. Ausgehend von den vorliegenden Daten in Abschnitt 3.1.2. wird als Maßeinheit 10 kg verwendet. Damit verschwinden in den Aufkommens- und Bedarfsangaben alle Stellen hinter dem Komma.
Der Aufkommensvektor soll als Dateneinheit mit dem Namen AVEK in die EDO-Datei TRANS gespeichert werden. In die gleiche EDO-Datei sollen auch der Bedarfsvektor, als Dateneinheit BVEK, und die Entfernungsmatrix, als Dateneinheit CMAT, gespeichert werden.
Die EDO-Datei soll auf einer Magnetplatte mit der Datenträgerarchivnummer BF1111 angelegt werden.

4.1.3. Kartenstrom für VD#DEM

Die Eingabe der Problemdaten soll in einem Rechenlauf in der Reihenfolge Aufkommensvektor, Bedarfsvektor und Entfernungsmatrix erfolgen. Der notwendige Kartenstrom lautet:

```
//              EXEC     PGM=VD#DEM
//STEPLIB       DD       DSN=VD1SOPS.LAD,VOL=SER=VD1GEN,UNIT=SYSDA,
//                       DISP=SHR
//VDISKO        DD       DSN=TRANS,VOL=SER=BF1111,UNIT=SYSDA,
//                       SPACE=(TRK,(5,,1)),DISP=(NEW,KEEP)
//FTØ6FØØ1      DD       SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6Ø5
//FTØ5FØØ1      DD       *
&PARAM
 Q='AVEC',M=6,N=1,SPCH=1,TYP=Ø
&END
```

Parameter- und Matrixdatenkarten für Aufkommensvektor

```
SI
S    1    1   51ØØ   35ØØ    4ØØ    16Ø    5ØØ
S    6    1    6ØØ
&
&PARAM
 Q='BVEC',M=1,N=15,TYP=Ø
&END
ZI
Z    1    1     1Ø    3ØØ    1ØØ     1Ø     7Ø
Z    1    6    3ØØ     1Ø    6ØØ    25Ø    567
Z    1   11    883   186Ø   11ØØ   2ØØØ   22ØØ
&
```

Parameter- und Matrixdatenkarten für Bedarfsvektor

```
&PARAM
 Q='CMAT',M=6,N=15,TYP=Ø
&END
ZI
Z    1    1    137    129    134    13Ø    133
Z    1    6    1Ø5     71     68     42     25
Z    1   11      7     57     6Ø    1Ø5    14Ø
 .
 .
 .
/*
```

Parameter- und Matrixdatenkarten für Entfernungsmatrix

Die Steueranweisungen entsprechen dem in Abschnitt 2.6. angegebenen Satz, wobei die EXEC-Anweisung den Programmnamen VD#DEM enthält.
In der VDISKO-DD-Anweisung sind der Name der EDO-Datei und die Datenträgerarchivnummer BF1111 eingetragen. Der SPACE-

Parameter ist bei neu zu gründenden Dateien anzugeben, die Größe ist ausreichend für die Aufnahme der Problemdaten.
Die für die Eingabe erforderlichen Parameterkarten werden eingeschlossen durch eine &PARAM- und eine &END-Karte.
Über den Parameter Q wird der jeweilige, frei wählbare Dateneinheitsname (siehe Abschnitt 4.1.2.) angegeben. M und N werden für die Angabe der Zeilen- bzw. Spaltenanzahl verwendet. Der Aufkommensvektor ist ein Spaltenvektor, deswegen ist bei dessen Eingabe SPCH=1 anzugeben.

4.1.4. Kartenstrom für VD#TD1

Der notwendige Kartenstrom für die Abarbeitung des Programmes VD#TD1 lautet:

```
//          EXEC   PGM=VD#TD1
//STEPLIB   DD     DSN=VD1SOPS.LAD,VOL=SER=VD1GEN,DISP=SHR
//VDISKO    DD     DSN=TRANS,VOL=SER=BF1111,UNIT=SYSDA,DISP=SHR
//FT08F001  DD     UNIT=SYSDA,SPACE=(3624,130)
//FT06F001  DD     SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605)
//FT05F001  DD     *
&PARAM
 C='CMAT',A='AVEC',B='BVEK'
&END
/*
```

Die Steueranweisungen entsprechen dem in Abschnitt 2.6. angegebenen Satz, wobei in der EXEC-Anweisung der Programmname VD#TD1 steht. In der VDISKO-DD-Anweisung sind die Angaben enthalten, die die Verarbeitung der Daten in der EDO-Datei TRANS ermöglichen.
Über die Parameter C, A und B werden die Namen der Dateneinheiten für die Entfernungsmatrix, den Aufkommens- und den Bedarfsvektor angegeben.

4.1.5. Ergebnisdiskussion

Die Lösung der in den vorigen Abschnitten beschriebenen Aufgabe erfolgte auf einer EDVA ES 1055. Die Rechenzeit betrug 3 Minuten Verweilzeit und 17 Sekunden ZE-Zeit.
Der Zielfunktionswert beträgt 511096; das bedeutet, es entsteht eine mittlere Transportentfernung von 49.81 Kilometern bei 10 260 Mengeneinheiten, das sind 102,60 kkg.
Folgender Transportplan ist optimal:

Aufkommensort	Bedarfsort	transportierte Mengen in kkg
1	2	3,00
	4	0,10
	6	0,70
	7	0,10
	8	6,00
	9	2,50
	10	5,67
	11	8,83
	12	18,60
	15	5,50
2	13	11,00
	14	18,40
	15	5,60
3	3	1,00
	5	0,70
	6	2,30
4	14	1,60
5	15	5,00
6	1	0,10
	15	5,90

Tabelle 4.1.: Ergebnis des Transportverteilungsproblems

4.2. Lösung des Problems der komplexen Grundfondsplanung

4.2.1. Programmauswahl

Die Lösung des unter 3.2. dargestellten Problems der komplexen Grundfondsplanung als lineares 0-1-Optimierungsproblem erfolgt durch die Abarbeitung der folgenden Anwendungsprogramme in der angegebenen Reihenfolge:

1. VD#DEM Einlesen der für das 0-1-Problem benötigten Daten für Problemmatrix, rechte Seite, Zielfunktion und Nebenbedingungstyp sowie Speichern dieser Daten in einer EDO-Datei.

2. VD#NVA1 Lösung des 0-1-Problems und Ergebnisausgabe.

4.2.2. Problemdaten in Matrix- oder Vektorform - EDO-Datei

Das dargestellte Problem hat 24 0-1-Variable und 17 Nebenbedingungen (7 ökonomische und 10 logische Nebenbedingungen). Die Daten für Problemmatrix, rechte Seite, Zielfunktion und Nebenbedingungstyp sollen in Dateneinheiten mit den Namen MATRIX, SEITE, ZIEL und TYP der EDO-Datei mit dem Namen GRUFO gespeichert werden. Die EDO-Datei soll auf der Magnetplatte mit der Datenträgerarchivnummer MLUHW angelegt werden.
Das Programm VD#NVA1 arbeitet nur mit einfachindizierten Variablen. Die Zuordnung von Programmvariablen x_k zu den zweifachindizierten Problemvariablen x_{ij} sei, unter Berücksichtigung der in Punkt 3.2. angegebenen Reihenfolge der Variablen x_{ij}, wie folgt:

k	i,j	k	i,j	k	i,j	k	i,j	k	i,j
1	1,1	6	3,1	11	5,1	16	7,1	21	9,1
2	1,2	7	3,2	12	5,2	17	7,2	22	9,2
3	1,3	8	3,3	13	6,1	18	7,3	23	9,3
4	2,1	9	3,4	14	6,2	19	8,1	24	9,4
5	2,2	10	4,1	15	6,3	20	8,2		

Der Problemvariablen x_{34} entspricht damit z.B. die Programmvariable x_9.
Es sind zwei Zielfunktionen Z_1 und Z_2 gegeben. Da die Lösung des Problems mittels beider Zielfunktionen erfolgen soll, werden beide in der Dateneinheit ZIEL (Z1 als Zeile 1, Z2 als Zeile 2) abgespeichert.
Der Vektor für Nebenbedingungstyp lautet bei Berücksichtigung der unter 3.2. angegebenen Reihenfolge der Nebenbedingungen (erst 7 ökonomische, dann 10 logische Nebenbedingungen jeweils in der angegebenen Reihenfolge):
(-2, -2, -2, -2, -2, -2, 1, Ø, -2, Ø, -2, -2, -2, -2, -2, Ø, Ø).
Da das Programm VD#NVA1 nur ganzzahlige Werte verarbeitet, ist es erforderlich, die Daten für die Zielfunktionen sowie für die Nebenbedingungen 1, 2, 3, 4, 6 und 7 (durch Multiplikation mit 10) in ganzzahlige Daten überzuführen. Damit ist z.B. die erste Nebenbedingung:

$$60x_{1,1} + 40x_{1,2} + 55x_{1,3} + \dots + 95x_{9,3} + 55x_{9,4} \leq 500$$

oder für den Fall einfachindizierter Variablen

$$60x_1 + 40x_2 + 55x_3 + \dots + 95x_{23} + 55x_{24} \leq 500.$$

Die rechte Seite ist wie folgt gegeben:
(500, 220, 150, 250, 60, 800, 250, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0).

4.2.3. Kartenstrom für VD#DEM

Unter der Voraussetzung, daß die Eingabe der Daten für die Problemmatrix, die rechte Seite, die Zielfunktionen und den Nebenbedingungstyp in die EDO-Datei GRUFO in dieser Reihenfolge in einem Rechenlauf mittels VD#DEM erfolgen soll, lautet der notwendige Kartenstrom für VD#DEM:

```
//             EXEC    PGM=VD#DEM
//STEPLIB      DD      DSN=VD1SOPS.LAD,VOL=SER=VD1GEN,
//                     UNIT=SYSDA,DISP=SHR
//VDISKO       DD      DSN=GRUFO,VOL=SER=MLUHW,UNIT=SYSDA,
//                     DISP=(NEW,KEEP),SPACE=(TRK,(5,,1))
//FT06F001     DD      SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605)
//FT05F001     DD      *
&PARAM
 Q='MATRIX'                   Parameterkarten für
 M=17,N=24,SPCH=1             Problemmatrix
&END

SI
S5   1   1   60   25   0   8   8          Matrixdaten-
S1   8   1    1                           karten für
                                          Spalte 1-3
S5   1   2   40   15   0   5   8          der Problem-
S1   8   2    1                           matrix
S5   1   3   55   30   0   7   8
S1   8   3    1
S1  17   3    1
 :
 &
 &PARAM
 Q='SEITE',M=17,INIT=1,SPCH=1   Parameterkarte für rechte Seite
 &END

SI
S5   1   1  500  220  150  250   60       Matrixdaten-
S2   6   1  800  250                      karten für
                                          rechte Seite
S2  16   1    0    0
 &
 :  Parameter- und Datenkarten für die Zielfunktionen
 &
 :  Parameter- und Datenkarten für Nebenbedingungstyp
/*
```

Die Steueranweisungen entsprechen dem unter 2.6. angegebenen Satz, wobei in der EXEC-Anweisung der Name VD#DEM angegeben ist. In der VDISKO-Anweisung sind der Name der EDO-Datei (GRUFO) sowie die Datenträgerarchivnummer (MLUHW) eingetragen.

Da die EDO-Datei neu gegründet werden muß, ist der SPACE-Parameter anzugeben. Die festgelegte Größe der Datei ist ausreichend für die Abspeicherung der für die Lösung des 0-1-Problems notwendigen Daten.
Zuerst erfolgt die Eingabe der Problemmatrix. Über den Parameter Q ist der Name der Dateneinheit angegeben, in welcher die Matrix gespeichert werden soll, über M und N die Dimensionen der Matrix.
Da die Matrix spaltenweise abzuspeichern ist, ist SPCH=1 erforderlich.
Der Parameter INIT ist nicht angegeben. Damit wird für alle im folgenden nicht eingelesenen Matrixelemente der Wert 0 (Standardwert von INIT) abgespeichert. Im folgenden sind die Matrixdatenkarten für die ersten drei Spalten dargestellt. Für die Spalten 1 und 2 werden für die Zeilen 1, 2, 3, 4, 5 und 8, für Spalte 3 für die Zeilen 1, 2, 3, 4, 5, 8 und 17 Werte eingelesen und abgespeichert.
In den Parameterkarten für die rechte Seite ist über den Parameter Q der Name der Dateneinheit angegeben, in die die rechte Seite gespeichert werden soll, über den Parameter M die Anzahl der Zeilen (Elemente) der rechten Seite. Da die rechte Seite als Spaltenvektor (1 Spalte) gespeichert wird, entfällt die Angabe des Parameters N, dessen Standardwert 1 ist. SPCH=1 bewirkt die spaltenweise Eingabe. Da mehrere Elemente der rechten Seite gleich 1 sind, ist INIT=1 angegeben. Damit sind für die betreffenden Elemente über die Matrixdatenkarten keine Werte gleich 1 einzulesen.

Kontrollfragen:

1. Wie lauten die Parameter- und Datenkarten für die Zielfunktionen?
2. Wie lauten die Parameter- und Datenkarten für Nebenbedingungstyp?

Lösungen:

Zu 1:

```
 &PARAM
 Q='ZIEL',M=2,N=24
 &END
ZI
Z5   1   1   8   7   9   6   9
Z5   1   6  1Ø  11   9  1Ø  13
Z5   1  11   8   7  1Ø  13  16
Z5   1  16  12  14  12   7   6
Z4   1  21   8   8  14   6
Z5   2   1  2Ø  15  21  19  18
Z5   2   6  27  26  24  19  3Ø
Z5   2  11  22  23  24  27  29
Z5   2  16  31  28  25  18  14
Z4   2  21  17  19  23  21
```

Zu 2: Folgende Parameter- und Datenkarten sind möglich:

```
 &PARAM
 Q='TYP',M=17,SPCH=1,INIT=-2
 &END
SI
S2   7   1   1   Ø
S1  1Ø   1   Ø
S2  16   1   Ø   Ø
```

Auf Grund der Angabe von INIT=-2 ist die Angabe der Werte -2 in den Datenkarten nicht erforderlich.

4.2.4. Kartenstrom für VD#NVA1

Unter der Voraussetzung, daß unter Verwendung der Zielfunktion Z2 eine optimale Lösung des Problems berechnet werden soll, lautet der notwendige Kartenstrom für VD#NVA1:

```
//            EXEC  PGM=VD#NVA1
//STEPLIB     DD    DSN=VD1SOPS.LAD,VOL=SER=VD1GEN,
//                  UNIT=SYSDA,DISP=SHR
//VDISKO      DD    DSN=GRUFO,VOL=SER=MLUHW,UNIT=SYSDA,DISP=SHR
//FT06F001    DD    SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605)
//FT05F001    DD    *
 &PARAM
 A='MATRIX',B='SEITE',RT='TYP'
 C='ZIEL',CZIX=2,ZTYP=1
 &END
/*
```

Die Steueranweisungen entsprechen dem unter 2.6. angegebenen Satz. Mittels der EXEC-Anweisung wird das notwendige Programm VD#NVA1 aufgerufen. Da die für das 0-1-Problem notwendigen Matrix- und Vektordaten mittels VD#DEM in die EDO-Datei GRUFO auf der Magnetplatte mit der Datenträgerarchivnummer MLUHW abgespeichert worden sind, sind diese Angaben (GRUFO, MLUHW) in der VDISKO-Anweisung einzutragen.

Über die Parameter A, B, C und RT sind die Namen der Dateneinheiten der EDO-Datei GRUFO, in der die Daten für die Problemmatrix, rechte Seite, Zielfunktion und den Nebenbedingungstyp (mittels VD#DEM) abgespeichert sind, angegeben. Da in der Dateneinheit ZIEL eine Matrix mit 2 Zeilen gespeichert ist, muß der Parameter CZIX angegeben werden. CZIX=2 bewirkt, daß die Zeile 2 (Z_2) als Zielfunktion verwendet wird. Da eine Maximierung erfolgen soll, ist ZTYP=1 erforderlich.

4.2.5. Ergebnisdiskussion

Die Lösung des 0-1-Problems mit dem unter 4.2.4. beschriebenen Kartenstrom für VD#NVA1 erfolgte auf einer EDVA ES 1055. Es wurden insgesamt 8 zulässige Lösungen berechnet. Die Berechnung der achten zulässigen Lösung erfolgte nach 1157 Iterationen, die Bestätigung ihrer Optimalität nach 1269 Iterationen. Die Rechenzeit betrug 8 Minuten Verweilzeit bzw. 66 Sekunden ZE-Zeit.

Folgende Programmvariablen x_k bzw. Problemvariablen x_{ij} sind in der optimalen Lösung gleich 1:

k	2	7	10	11	15	18	2Ø	24
i,j	1,2	3,2	4,1	5,1	6,3	7,3	8,2	9,4

Von den vorhandenen Vorhaben werden also folgende realisiert:

Vorhaben 1 mittels Variante V_{12}
Vorhaben 3 mittels Variante V_{32}
Vorhaben 4 mittels Variante V_{41}
Vorhaben 5 mittels Variante V_{51}
Vorhaben 6 mittels Variante V_{63}
Vorhaben 7 mittels Variante V_{73}
Vorhaben 8 mittels Variante V_{82}
Vorhaben 9 mittels Variante V_{94}

Die dabei erreichte Arbeitsrentabilität beträgt 18,2.
Bezüglich der Nebenbedingungen 1-7 wurden für die optimale Lösung die folgenden Informationen ausgegeben.

Nebenbedingung	Aktivität $\sum a_{ij} x_j$	Differenz $b_i - \sum a_{ij} x_j$ Prozent (zu b_i)	Wert
1 (Investitionsvolumen)	49,9	0,20	0,1
2 (Bauvolumen)	21,5	2,27	0,5
3 (Materialfonds 1)	11,2	25,33	3,8
4 (Materialfonds 2)	16,7	33,20	8,3
5 (Arbeitskräftefonds)	56	6,67	4
6 (obere Absatzschranke)	48,0	40,00	32,0
7 (untere Absatzschranke)	48,0	92,00	23,0

Damit werden z.B. vom zur Verfügung stehenden Investitionsvolumen von 50 (Mio M) 49,9 benötigt, d.h. 0,1 bzw. 0,20% werden nicht verbraucht.
Die Lösung des Problems mit Zielfunktion Z_1 ergab 5 zulässige Lösungen. Die zulässige Lösung 5 wurde nach 199 Iterationen berechnet, die Bestätigung ihrer Optimalität erfolgte nach 979 Iterationen. Die Gesamtrechenzeit betrug 8 Minuten Verweilzeit

bzw. 59 Sekunden ZE-Zeit. Die Programmvariablen x_k, k= 1, 7, 10, 11, 15, 18, 23, sind in der optimalen Lösung gleich 1. Der Zielfunktionswert, d.h. die erreichte Investitionsrentabilität ist 8,2.

Bezüglich der Nebenbedingungen 1-7 wurden für die optimale Lösung folgende Informationen ausgegeben:

Nebenbedingung	Aktivität $\sum a_{ij} x_j$	Differenz $\lvert b_i - \sum a_{ij} x_j \rvert$ Prozent (zu b_i)	Wert
1	49,9	0,20	0,1
2	20,9	5,00	1,1
3	11,1	26,00	3,9
4	11,9	52,40	13,1
5	55	8,33	50
6	58,0	27,50	22,0
7	58,0	132,00	33,0

Kontrollfragen:

1. Welche Vorhaben werden bei Berücksichtigung der Zielfunktion Z_1 realisiert und mittels welcher Varianten? Welche Unterschiede ergeben sich diesbezüglich zur Rechnung mit Zielfunktion Z_2?
2. Ergibt sich aus der Tatsache, daß für die berechneten optimalen Lösungen die vorgegebenen Fonds (Nebenbedingung 1-5) nicht ausgeschöpft werden, daß durch Vergrößerung der Fondsvorgaben keine Verbesserung der optimalen Lösung mehr möglich ist?

Lösungen:

Zu 1:

Entsprechend der unter 4.2.2. angegebenen Variablenzuordnung ergibt sich auf Grund der berechneten optimalen Lösung die Realisierung folgender Vorhaben:

Vorhaben 1 mittels Variante V_{11}
Vorhaben 3 mittels Variante V_{32}
Vorhaben 4 mittels Variante V_{41}
Vorhaben 5 mittels Variante V_{51}
Vorhaben 6 mittels Variante V_{63}
Vorhaben 7 mittels Variante V_{73}
Vorhaben 9 mittels Variante V_{93}

Damit wird.für beide Zielfunktionen das Vorhaben 2 nicht realisiert. Bei Optimierung nach Z1 wird des weiteren Vorhaben 8 nicht realisiert. Die Realisierung der Vorhaben 1 und 9 erfolgt jeweils mittels verschiedener Varianten.

Zu 2:

Nein. Aus der Ganzzahligkeit der Variablen folgt, daß durch Vorgabe größerer Fonds ggf. bessere optimale Lösungen berechnet werden können.

5. Literaturverzeichnis

/ 1 / Mathematik für Ökonomen
Hochschullehrbuch, Teil 2
Verlag Die Wirtschaft,
Berlin 1980

/ 2 / Kadlec, V.; Vodáček, L.
Lineare Optimierung im
Transportwesen
transpress, VEB Verlag
für Verkehrswesen,
Berlin 1967

/ 3 / Dück, W.
Diskrete Optimierung
Akademie-Verlag,
Berlin 1977

/ 4 / Korbut, A.A.; Finkelstein, J.J.
Diskrete Optimierung
Akademie-Verlag,
Berlin 1971

/ 5 / Programmpaket Diskrete Optimierung
Problembeschreibung
VEB Robotron, ZFT,
Dresden

Druck:

ZENTRALSTELLE FÜR LEHR- UND ORGANISATIONSMITTEL DES MINISTERIUMS FÜR HOCH- UND FACHSCHULWESEN, ZWICKAU

Ag 628/485/80/DDR/1100-ZLO 719/80

LEHRBRIEFE FÜR DAS HOCHSCHULFERNSTUDIUM

HERAUSGEGEBEN
VON DER ZENTRALSTELLE FÜR DAS HOCHSCHULFERNSTUDIUM
DES MINISTERIUMS FÜR HOCH- UND FACHSCHULWESEN

Mathematische Methoden und Modelle

10. LEHRBRIEF

Grundlagen und Anwendung des PS SIMDIS

02 4602 10 0

Mathematische Methoden und Modelle

10. Lehrbrief

Grundlagen und Anwendung des PS SIMDIS

Autoren:

Prof. Dr. sc. oec. Dr. rer. nat. Walter R u n g e
Sektion Sozialistische Betriebswirtschaft
Wilhelm-Pieck-Universität Rostock

Dr. oec. Günter P e i ß k e r
Sektion Wirtschaftswissenschaften
Martin-Luther-Universität Halle

Dipl.-Math. Jürgen S t r o t t m a n n
VEB Robotron Dresden

02 4602 10 0

Das druckfertige Manuskript wurde an der
Martin-Luther-Universität Halle hergestellt.

Bestell-Nr. 02 4602 10 0

Verfaßt für die Zentralstelle für das Hochschulfernstudium des Ministeriums für Hoch- und Fachschulwesen Dresden.

Herausgegeben im Auftrag des Ministeriums für Hoch- und Fachschulwesen der Deutschen Demokratischen Republik von der Zentralstelle für das Hochschulfernstudium Dresden.

Inhaltsverzeichnis

0. Vorbemerkungen

Die Simulation diskreter Prozesse besitzt für die Entscheidungsvorbereitung komplizierter Problemstellungen der Leitung und Planung in der Wirtschaft eine zunehmende Bedeutung. Zu den wesentlichen Vorzügen der Simulation gehören:

1. Algorithmische Modellierbarkeit vielparametriger Prozesse mit hoher Variabilität.
2. Algorithmische Modellierbarkeit von Systemen mit komplizierten mathematischen Strukturen.
3. Algorithmische Modellierbarkeit von ökonomischen Prozessen mit starken stochastischen Einflüssen.
4. Algorithmische Modellierbarkeit von zeitabhängigen dynamischen Vorgängen.

Das "einheitliche System der elektronischen Rechentechnik" (ESER), an deren Entwicklung die sozialistischen Länder gemeinsam beteiligt sind, ist die gegenwärtige gerätetechnische Basis der elektronischen Datenverarbeitung in der DDR. Zur Lösung vielseitiger Aufgabenstellungen der diskreten Simulation wird vom Kombinat Robotron das leistungsfähige Programmiersystem Simulation diskreter Systeme (PS SIMDIS) bereitgestellt.

Nachdem im Abschnitt 1. auf die allgemeine Bedeutung der diskreten Simulation in der Ökonomie eingegangen wird, erfolgt im Abschnitt 2. eine Einführung in die Grundlagen des PS SIMDIS.

Während im Abschnitt 3. zwei repräsentative Beispiele aus der Betriebswirtschaft mit Hilfe der Simulation mathematisch modelliert und algorithmiert werden, demonstriert der Abschnitt 4. deren rechentechnische Realisierung mit dem PS SIMDIS.

Halle, im Herbst 1980

Doz. Dr. sc. W. Lassmann
Martin-Luther-Universität Halle
Sektion Wirtschaftswissenschaften
Wissenschaftsbereich Informationsverarbeitung
Lehrgruppe Mathematik
(Leitung der Lehrbriefe Nr. 8 - 10)

1. Ökonomische Problemstellung der Simulation

1.1. Wesen und Methoden der Simulation

Seit der ersten Veröffentlichung zur Simulation im Jahre 1954 sind international weit über 10 000 Publikationen zur Entwicklung und Anwendung der Simulation erschienen.
Im Jahre 1956 erfolgte die erste Anwendung der Simulation bei der Modellierung diskontinuierlicher Prozesse ohne Berücksichtigung stochastischer Einflüsse.
Seit dieser Zeit hat sich die Simulation aus bescheidenen Anfängen heraus zu einem der am häufigsten genutzten und leistungsfähigsten Verfahren neben den Methoden der mathematischen Optimierung (vor allem der linearen Optimierung) und den statistischen Methoden (vor allem mathematisch-statischen und ökonomisch-statistischen Methoden) in der modernen Entscheidungsfindung entwickelt. Die Simulation wird in erster Linie für den Aufgabenbereich der Operationsforschung angewendet.
Daher ordnen viele Autoren die Simulation den Methoden der Operationsforschung zu, andere fassen die Simulation als selbständige Disziplin auf.
Eine allgemein anerkannte, einheitliche Begriffsdefinition für die Simulation gibt es bisher nicht.
Es existiert vielmehr eine Reihe zweckbestimmter Auffassungen und Definitionen, die eine Vielzahl von Methoden und Technologien der Simulation bezeichnen.
Der Gebrauch des Begriffs Simulation ist mit spezifischen Aspekten und Anwendungsbereichen der Modellmethode, also der Entwicklung und Anwendung von mathematischen Verfahren zur Erkenntnisgewinnung und -aufbereitung verbunden.
Vereinfacht kann deshalb gesagt werden, daß unter Simulation

a) Verfahren zur Ermittlung von Lösungen für ein mathematisch-analytisches Optimierungsmodell (z. B. ein Modell der nichtlinearen Optimierung)
oder

b) ein die Modellierung und Algorithmierung in einem einheitlichen Prozeß einschließendes sogenanntes algorithmisches Modell zur Abbildung von Systemen und ihrem Verhalten, also den in ihren ablaufenden Prozessen,

verstanden wird.
Im ersten Fall wird die Simulation anstelle entsprechender Lösungsverfahren verwendet und hat damit eine Ersatzfunktion. Dieser Anwendungsbereich der Simulation ist von untergeordneter Bedeutung.
Im zweiten Fall sind für den Anwendungsbereich der Simulation typische Aufgaben zu lösen, hier werden die eigentlichen Möglichkeiten der Simulation im Sinne ihrer Grundfunktion genutzt.

Die Anwendung der Simulation erfolgt, wenn mindestens eine der folgenden Situationen eintritt:

1. Modelle des 4. Abstraktionsgrades - also mathematisch-analytische Modelle - gestatten es nicht, das Problem ausreichend genau zu erfassen (z. B. vielparametrige Prozesse) oder die relevanten Parameter und Variablen in ihrem Zusammenhang analytisch darzustellen (z. B. Bedienungssysteme mit komplizierter Struktur, Auftreten von Zufallsgrößen mit verschiedenartigen Verteilungsgesetzen).

2. Für analytische Modelle, die vorhanden sind oder entwickelt werden können, ist kein oder kein effektiv nutzbarer Algorithmus zur Berechnung verfügbar (z. B. Reihenfolgeaufgaben der Ablaufplanung für diskrete Produktionsprozesse).

3. Es liegen komplexe Aufgabenstellungen vor, deren Teilprobleme eine unterschiedliche mathematische Struktur aufweisen und damit die Anwendung unterschiedlicher mathematischer Disziplinen, wie z. B. der Bedienungstheorie, der Automatentheorie und der Graphentheorie, erforderlich machen.
   Diese unterschiedlichen Modelltypen als Grundlage für ein zu entwickelndes Simulationsmodell bedingen unterschiedliche elementare Bestandteile des algorithmischen Simulationsmodells.

4. Es sind Systeme und darin ablaufende Prozesse zu untersuchen, die sehr stark zufälligen Einflüssen unterliegen, wobei es nicht ausreicht, diese durch Mittelwerte oder andere statistische Größen zu berücksichtigen, sondern für die

eine Behandlung als Zufallsprozesse erforderlich ist.

5. Wenn neben komplizierten, insbesondere stochastischen Zusammenhängen eine Abhängigkeit von der Zeit existiert, die zur zusätzlichen, in der Regel expliziten Berücksichtigung der Zeit und damit zu einem mathematischen Modell, das "abläuft", nicht aber gelöst wird, führt.

Der wesentlichste Vorzug der Simulation besteht darin, daß Simulationsmodelle die Abbildung von Systemen mit einem Grad an Anpassungsfähigkeit, Komplexität und Wirklichkeitsnähe gestatten, wie er mit analytischen Modellen nicht erreichbar ist. Andererseits setzt die Anwendung der Simulation wie keine andere Methode statistische Beobachtungen über längere Zeiträume voraus.
Langfristige statistische Erfassungen dienen der Erkenntnis der vorliegenden Gesetzmäßigkeiten und der Schaffung der Voraussetzungen für die Anwendung der Simulation.

Für die Klassifizierung von Simulationsmodellen gibt es unterschiedliche Gesichtspunkte.
Einige wesentliche davon sind:

1. Unterteilung nach den im Modell abgebildeten Charakteristika der Abläufe in realen oder künftigen Systemen, wobei diese Abläufe Folgen von zeitabhängigen Zustandsänderungen der Systeme sind

   a) <u>stetige</u> und <u>diskrete Simulation</u>
   Die <u>stetige Simulation</u> (auch kontinuierliche Simulation genannt) <u>geht von</u> einer "kontinuierlichen" Folge von Zeitpunkten, also <u>einer stetigen Zeitskala im Ablauf des Modells und von stetigen Zustandsänderungen aus</u>.
   Damit werden differentiell kleine Abstände zwischen aufeinanderfolgenden Ereigniszeitpunkten und zwischen den einzelnen Zuständen angenommen. Die dazugehörigen Simulationsmodelle haben in der Regel die Form von Differentialgleichungen und weisen in der Form keinen Unterschied zu mathematisch-analytischen Modellen auf. Im Rahmen der Anwendung mathematischer Methoden und Modelle in der Ökonomie hat die <u>diskrete Simulation</u> eine große

Bedeutung.
Das Systemverhalten wird als Folge von "Ereignissen" dargestellt, wobei unter einem Ereignis die (unstetige) Veränderung mindestens eines Merkmals oder Parameters verstanden wird.
Bei der diskreten Simulation treten damit Veränderungen der Zeit- und Zustandsgrößen im wesentlichen nur in diskreten Werten auf.

b) deterministische und stochastische Simulation

Die deterministische Simulation ist dadurch gekennzeichnet, daß alle Daten, Einflußfaktoren und Parameter fest vorgegeben sind. Das Simulationsmodell liefert bei einmaligem Durchspielen eine eindeutig bestimmte Lösung.
Die stochastische Simulation stellt eine Erweiterung der deterministischen Simulation durch die Berücksichtigung von zufälligen Parametern dar.
Bei stochastischen Simulationsmodellen findet eine Berücksichtigung der Zufallsabhängigkeit (stochastische Einflußfaktoren und Parameter) des zu simulierenden Prozesses durch die mehrfache Erzeugung unabhängiger Realisierungen der relevanten Zufallsgrößen durch Simulationsexperimente auf der Grundlage von Zufallszahlen bestimmter ermittelter theoretischer oder empirischer Wahrscheinlichkeitsverteilungen statt.
Um statistisch gesicherte Resultate zu erhalten, wird die Reaktion des Modells gemessen und die Ermittlung neuer Zufallszahlen so oft wiederholt, bis eine ausreichend repräsentative Stichprobe realisiert ist.
Die Simulation von Zufallsprozessen ist vor allem wichtig, wenn sehr starke zufällige Einflüsse auftreten und es nicht ausreicht, diese durch Mittelwerte oder andere statistische Größen zu beschreiben.

c) statische und dynamische Simulation

Bei der statischen Simulation findet die Variable "Zeit" keine primäre Beachtung im Simulationsmodell (z. B. Lösung der klassischen Reihenfolgeaufgabe durch Erzeugung

zufälliger Permutationen).
Bei der dynamischen Simulation soll das Verhalten eines Systems in Abhängigkeit von der Zeit untersucht werden.

2. Unterteilung nach der Methodik des Aufbaus und der Realisierung des diskreten Simulationsmodells

   a) Ereignisfolgesimulation und Zeitfolgesimulation

   In beiden Fällen geht es um die Ermittlung von Zeitpunkten, zu denen Zustandsänderungen, also Ereignisse, eintreten.

   Die Ereignisfolgesimulation bildet die Ereignisse in der Reihenfolge ihres geplanten oder tatsächlichen zeitlichen Eintretens nach. Die Zeitfortschreibung im Simulationsmodell ist variabel.

   Die Zeitfolgesimulation bezeichnet Simulationsmodelle mit einer Zeitfortschreibung in konstanten Zeitschritten nach dem sogenannten Δt-Prinzip.
   Der Vorteil der Zeitfolgesimulation besteht im Wegfall aller Suchoperationen für die Bestimmung des folgenden Ereigniszeitpunktes. Dabei werden jedoch auch Zeitpunkte berücksichtigt, in denen kein Ereignis eintritt.
   Die Simulationsuhr (simulierte Zeit) beginnt bei der Ereignis- und Zeitfolgesimulation im allgemeinen bei "0" und rückt dann unter Verwendung einer "absoluten" bzw. einer "relativen" Zeit in variablen Zeitschritten bei der Ereignisfolgesimulation oder in konstanten Zeitschritten bei der Zeitfolgesimulation weiter. Nach Beendigung der Simulation kann die Simulationszeit durch Kalenderroutinen unter Verwendung von Kalendertabellen in erforderliche betriebliche Arbeitstagekalender transformiert werden.

   b) Monte-Carlo-Simulation und gezielte Simulation

   Die Monte-Carlo-Simulation, auch Methode der statistischen Versuche genannt, ist ein Verfahren zur Erzeugung von Zufallszahlen einer bestimmten theoretischen oder empirischen Wahrscheinlichkeitsverteilung und der Zuord-

nung dieser Werte zu den Zufallsgrößen im Simulationsmodell.
Die gezielte Simulation, der eine Monte-Carlo-Simulation vorausgehen kann, zeichnet sich durch Verwendung einer Suchstrategie zur schnelleren Ermittlung der effektivsten Lösung aus.

Simulationsmodelle zeichnen sich dadurch aus, daß die Modellierung und Algorithmierung in einem Schritt erfolgen.
Die möglichen Darstellungsformen des entsprechenden Simulationsmodells sind

- die Operatorenschreibweise nach A. A. Ljapunow,
- algorithmische Graphen,
- Programmablaufpläne.

Dabei erfolgt mit Hilfe von Programmablaufplänen eine blockorientierte Darstellung des Simulationsmodells.

Kontrollfragen

1. Nennen Sie die Unterschiede zwischen deterministischer und stochastischer Simulation sowohl hinsichtlich der ökonomischen Aufgabenstellung als auch der Realisierung des Simulationsmodells!
2. Wie kann mit Hilfe der Simulation eine Suche nach der effektivsten Variante eines Prozeßablaufes erfolgen?

1.2. Rationalisierungsmittel für die Simulation

Ein entscheidender Vorteil der Simulation ist ihre Anpassungsfähigkeit an die unterschiedlichsten realen Bedingungen. Die Untersuchung eines jeden Prozesses erfordert im allgemeinen die Entwicklung eines speziellen Modells. Daraus resultiert, daß es kaum auch nur für zwei reale Prozesse vollständig identische Modelle gibt. Um die Anwendung von Simulationsmodellen dennoch zu rationalisieren, sind allgemeine Merkmale und Eigenschaften der Simulation zu beachten und zu nutzen.
Dabei werden zwei Wege beschritten:

a) Durch einen bausteinförmigen Aufbau von Simulationsmodellen wird eine vereinfachte Entwicklung und Anpassung von Simulationsmodellen an unterschiedliche konkrete Systeme erreicht.
   Für in Simulationsmodellen häufig auftretende organisatorische Probleme, wie z. B. die Steuerung des zeitlichen Ablaufes im Modell, die Bereitstellung von Zufallszahlen und speziellen Verteilungen, die Verarbeitung von Funktionen oder die Erfassung und Aufbereitung von statistischen Größen, werden multivalent einsetzbare Standardalgorithmen erarbeitet.

b) Die rechentechnische Realisierung von Simulationsmodellen ist im allgemeinen nur über ihre Programmierung vor allem in einer problemorientierten Sprache möglich. Dazu wurden ALGOL, FORTRAN, COBOL, PL 1 u. a. genutzt.
   Mit der Verbreitung der ESER-Rechentechnik und der Entwicklung dazugehöriger Systemunterlagen erlangten mit dem Ziel der Erhöhung der Effektivität der Programmierung, rechentechnischen Realisierung und praktischen Anwendung Simulationssprachen eine zunehmende Bedeutung.

Eine vorrangige Bedeutung für die Simulation diskreter Systeme, Prozesse und Problemstrukturen gewannen die blockorientierte Simulationssprache SIMDIS (Simulation diskreter Systeme) in der DDR und verschiedene GPSS-Versionen vor allem in der Sowjetunion.
Im Vergleich dazu sind in den kapitalistischen Ländern die Simulationssprachen SIMSCRIPT, GPSS und zunehmend auch SIMULA weit verbreitet.
Die Simulationssprache GPSS (General Purpose Simulation System) nimmt dabei die Position einer Basissprache für viele heute gebräuchliche Versionen von Simulationssprachen ein.
Erfahrungen zur Implementierung und Anwendung von SIMULA und zu ihrer Einbeziehung in die Ausbildung gibt es auch in der UdSSR, ČSSR und VR Polen. Insgesamt sind als wichtige Simulationssprachen für Rechenanlagen der sozialistischen Länder SDS für BESM 6, VOPS SIMDIS DOS/ES, PS SIMDIS OS/ES, GPSS für ESER-Rechenanlagen (von ES 1020 aufwärts) und GPS 22 für MINSK 22

in Anwendung. In der DDR flossen die Erfahrungen vieler Anwendungen der digitalen Simulation diskreter Modelle in die Entwicklung von VOPS SIMDIS DOS/ES als Programmiersystem für die diskrete Systemsimulation ein, das inzwischen als ESER-bestätigte Software in der Version PS SIMDIS OS/ES seit Ende 1977 vorliegt.

Die Simulationssprache SIMDIS ist die auf ESER-Rechenanlagen am häufigsten genutzte blockorientierte Sprache. Die Anwendung von SIMDIS nimmt daher sowohl gegenwärtig und infolge ihrer Leistungsfähigkeit auch künftig die Schlüsselstellung für die Simulation diskreter Modelle auf ESER-Rechenanlagen in der DDR ein. Das PS SIMDIS besitzt eine blockorientierte Sprache. Damit ist das Verständnis für die Entwicklung eines SIMDIS-Programms dann am leichtesten, wenn von einem Ablaufplan für das Simulationsmodell ausgegangen wird.

Die OS/ES-Version des PS SIMDIS beinhaltet die Möglichkeit, vom Anwender geschriebene ASSEMBLER-, FORTRAN- und PL 1-Programme aufzurufen und zu verarbeiten.

Weiterhin bietet das PS SIMDIS die Möglichkeit des Zugriffs auf externe Dateien mittels EDO (Einheitliche Dateiorganisation) und erlaubt damit die Kopplung mit anderen PP/PS (z. B. die Kopplung mit dem PP STATISTIK zwecks Datenauswertung).

1.3. Anwendungsmöglichkeiten

Die insgesamt möglichen Anwendungsmöglichkeiten der Simulation ergeben sich aus den in Bild 1 ersichtlichen, über die digitale Systemsimulation hinausreichenden Kombinationsmöglichkeiten.

Diese führen zu vier anwendungsbezogenen Klassen von Simulationsverfahren:

1. Kombination der Merkmale von I, II, IV;
2. Kombination der Merkmale von I, III, IV;
3. Kombination der Merkmale von I, II, III, IV;
4. Kombination der Merkmale von I und II mit einem mathematisch-analytischen Modell.

Merkmale/Bestandteile des Simulationsmodells	ausgewählte Kombinationen									Anwendungsbeispiele
	1	2	3a	3b	4a	4b	5	6	7	
I - Analoge Nachbildung	x	x								Variante 1: - Flugsimulatoren, - Fahrtrainer
- Digitale Nachbildung			x	x	x	x	x	x	x	
II Nachbildung eines realen Prozesses . Nachbildung deterministischer Merkmale										Variante 2: - Netzwerksimulatoren
. Nachbildung stochastischer Merkmale	x						x	x	x	Variante 3: - Variantenrechnungen für volkswirtschaftliche und betriebliche Probleme
III Nachbildung eines Lösungsverfahrens . deterministische Merkmale		x	x	x						Variante 4: - Reihenfolgeprobleme bei Fertigungsprozessen (eine von vielen Varianten)
. stochastische Experimente					x	x	x			
für ein . deterministisches (Optimierungs-)Problem					x		x			Variante 5: - Berechnung nach Integralen - Berechnung von Differentialgleichungen
. stochastisches (Optimierungs-)Problem						x				
IV - Prozeßverhalten wird optimiert		x	x		x			x		Variante 6: - Systemsimulation mit Optimierung
- Prozeßverhalten wird nicht optimiert	x			x		x	x		x	Variante 7: - Systemsimulation ohne Optimierung

<u>Bild 1</u>: Ausgewählte Kombinationen von Merkmalen von Simulationsmodellen und dazugehörige Anwendungsbeispiele

Vorrangiges Anwendungsgebiet der Simulation ist die Entwicklung und Nutzung von Simulationsmodellen für die Analyse und Synthese sowie den Entwurf oder Ersatz real existierender oder hypothetischer dynamischer Systeme. Dabei ist die digitale Systemsimulation unter Berücksichtigung stochastischer Elemente besonders interessant und wichtig.
Die Anwendung der Simulation beinhaltet die Arbeitsetappen System- und Problemanalyse, Entwicklung des Simulationsmodells (hier werden Modellierung und Algorithmierung in einem einheitlichen "Modellierungsalgorithmus" verbunden), zielgerichtetes Experimentieren zur Verbesserung des Modells und zur Lösung der Aufgaben. Dabei wird mit einfachen Modellkonzeptionen begonnen, die sukzessive verfeinert werden können.
Als Zielstellung der Anwendung der Simulation kann u. a. die Gewinnung von Erkenntnissen über den tatsächlich ablaufenden Prozeß, über das Systemverhalten bei gegebenen oder variierenden Bedingungen, über Struktur, Funktion und Verhalten projektierter Systeme u. a. betrachtet werden.
Über vielfältige Anwendungsmöglichkeiten verfügt dazu das PS SIMDIS.
Dabei ist mit Hilfe des PS SIMDIS sowohl die Analyse und Synthese als auch der Entwurf und Ersatz von einfachen stationären bis hin zu komplizierten dynamischen Prozessen möglich.

Die volkswirtschaftlichen Anwendungsmöglichkeiten des PS SIMDIS reichen von den Haupt- und Hilfsprozessen im Produktionsbereich vor allem der metallverarbeitenden Industrie und Leichtindustrie über Informationsprozesse und Entscheidungsprozesse der Leitung und Organisation in den einzelnen Industriezweigen bis hin zu Aufgabenstellungen aus dem Bergbau, dem Hüttenwesen, dem Bauwesen, der Landwirtschaft, dem Verkehrswesen, dem Nachrichten- und Fernmeldewesen, dem Dienstleistungsbereich usw.

Im Bereich industrieller Fertigungsprozesse spielen vor allem die Gestaltung sowie die Planung und die Steuerung

- von Produktionsprozessen (technologische und organisatorische Gestaltung von Prozeßabläufen, Mehrmaschinenbedienung, Pufferung bei Fließstraßen usw.),

- von TUL-Prozessen (Instandhaltungs- und Transportprozesse),
- von Informationsprozessen (Erarbeitung von Steueralgorithmen für Informationsprozesse, Untersuchung und Rationalisierung von Echtzeit- und Stapelverarbeitungsinformationsprozessen, Rationalisierung von Leitungssystemen usw.)

eine große Rolle.
Besonders bekannt sind die Aufgaben der optimalen Abfertigungsorganisation, z. B. die Abfertigung von Kunden in Kaufhallen und Einrichtungen des Dienstleistungsbereiches, die Abfertigung an Schaltern und Kontrollstellen (Zollkontrolle u. a.), das Beladen und Löschen von Schiffen, die Abfertigung von Flugzeugen auf Flughäfen oder Autos an Tankstellen.

Die charakteristische Gemeinsamkeit vieler dieser Prozesse besteht darin, daß zu diskreten Zeitpunkten sich bewegende Elemente, wie z. B. Maschinen, Werkstücke, Baugruppen, Informationen und Waren, in Beziehung zu den im System bzw. zu den in den darin ablaufenden Prozessen feststehenden Elementen wie Maschinen, Bedienungseinrichtungen, Abfertigungsstellen usw. treten.

<u>Kontrollfragen</u>

1. Nennen Sie die relevanten Faktoren, die bei der Simulation des Kaufgeschehens in einer Kaufhalle unter dem Aspekt des Eintretens eines Kunden, seiner Bedienung und der Abkassierung zu beachten sind.
2. Welche Kriterien zur Bewertung der Effektivität eines Produktionsprozesses sind bei der Simulation des Ablaufs von Produktionsprozessen von Bedeutung?

2. Allgemeine Vorstellung des PS SIMDIS

2.1. Konzeption des PS SIMDIS

Das PS SIMDIS ist ein Programmiersystem für die diskrete Systemsimulation. Unter diskreter Systemsimulation verstehen wir die Simulation eines Systems - natürlich stets im Hinblick auf die Lösung einer vorgegebenen Aufgabe, das wesentliche Zustandsänderungen in diskreten Zeitabständen erfährt. Jede Zustandsänderung (zu irgendeinem Zeitpunkt) wird ein Ereignis genannt.

Dem Programmiersystem SIMDIS liegt eine ereignisorientierte Steuerung des zeitlichen Ablaufes zugrunde. Das bedeutet, daß das Modell nur zu den Ereigniszeitpunkten Abläufe im System vorsieht; zwischen aufeinanderfolgenden Ereignissen bleibt das System in Ruhe, d. h. verändert sich sein Zustand nicht. Die Länge der einzelnen Intervalle ist abhängig von der Aufeinanderfolge der Ereignisse. Sie wird in jedem Falle ein ganzzahliges Vielfaches der vom Anwender einheitlich für das gesamte Modell festgelegten Zeiteinheit sein.

Das PS SIMDIS besitzt eine Reihe von Algorithmen, wie sie bei der Entwicklung von Simulationsmodellen und ihrer Umsetzung in eine Programmiersprache immer wieder benötigt werden. Dies sind u. a. Algorithmen zur

- Steuerung des zeitlichen Ablaufes,
- Realisierung parallel verlaufender Vorgänge,
- Erzeugung von Zufallszahlen gemäß beliebiger und/oder theoretischer Verteilungen,
- Erfassung und Aufbereitung statistischer Größen,
- Datenein- und -ausgabe.

Besonders geeignet ist das PS SIMDIS zur Lösung von Bedienungsproblemen im weitesten Sinne. Neben den oben genannten Algorithmen, die für eine Vielzahl von Simulationsmodellen von Bedeutung sind, existieren im PS SIMDIS spezielle Algorithmen, die die Simulation von Bedienungsprozessen unterstützen. Dazu gehören Algorithmen zur

- Realisierung unterschiedlicher Eingangsströme,

- Steuerung von Eingangs- und Ausgangsströmen nach unterschiedlichen Prioritätsregeln,
- ein- bzw. mehrkanaligen Bedienung,
- Erfassung von Warteschlangen und deren statistischen Auswertung,
- Abfertigung nach unterschiedlichen Warteschlangendisziplinen.

Durch Nutzung der genannten SIMDIS-Algorithmen kann sich der Anwender von PS SIMDIS voll auf die Modellierung seines Systems konzentrieren. Die Modellformulierung kann direkt unter Verwendung von SIMDIS-Sprachelementen vorgenommen werden. Mit diesen Sprachelementen können die Fortschreibung der Simulationszeit, die einkanalige und mehrkanalige Bedienung sowie Warteschlangen dargestellt werden. Des weiteren können umfangreiche Dateneingaben und -ausgaben, arithmetische und logische Operationen und anderes beschrieben werden. Das PS SIMDIS sammelt während eines Simulationsexperiments laufend statistische Daten, die am Ende automatisch in Tabellenform ausgegeben werden.

Dem Leser wird empfohlen, sofern er mit Begriffen und Inhalt der Bedienungstheorie noch nicht vertraut ist, sich in [1] und [2] darüber zu informieren. Kenntnisse auf diesem Gebiet erleichtern das Verständnis der weiteren Abhandlungen.

Zur Formulierung eines Problems dienen Sprachanweisungen. Die damit verbundenen internen Algorithmen entsprechen in einem Programmablaufplan einem mehr oder weniger großen Teil, einem Block. In Anlehnung daran heißen die ausführbaren Anweisungen Blöcke. Für die Erzeugung von Aktivatoren genügt die Angabe eines Blockes, das Betreten und Verlassen eines Speichers wird ebenfalls durch Angabe von jeweils einem Block dargestellt usw. Über Definitionsanweisungen können Funktionen beschrieben, Rechenvorschriften erklärt und Anfangswerte zugewiesen werden. Eine von der standardisierten Ergebnisausgabe abweichende Form erlaubt das Zusammenstellen von eigenen Drucklisten, Kurven und Histogrammen. Durch die Einfachheit der Sprache kann der Problemanalytiker zum großen Teil selbst die Arbeit des Programmierers übernehmen. Das Konzept der Sprache garantiert auch

eine gute Übersichtlichkeit komplizierter Systeme.
Ein SIMDIS-Programm wird in Lochkartenform eingelesen. Eine oder mehrere Lochkarten bilden eine Anweisung.
Entsprechend ihrer Wirkungsweise werden fünf verschiedene Anweisungstypen unterschieden:

- Blockanweisungen,
- Definitionsanweisungen,
- Steueranweisungen,
- Anweisungen für benutzereigene Ausgabe,
- Anweisungen für Ein- und Ausgabe mittels EDO.

Die Blockanweisungen, kurz Blöcke genannt, dienen entsprechend ihrer Anordnung der Darstellung der Logik des zu simulierenden Systems. Die Eintragungen im Operandenfeld eines Blockes sind als Positionsparameter festgelegt. Es wird zwischen notwendigen, korrigierenden und ergänzenden Eintragungen unterschieden.
Die Definitionsanweisungen dienen der Festlegung der Rechenvorschriften in Berechnungselementen, der Zuweisung von Anfangswerten einiger Systemelemente und der Definition von Tabellen, Funktionen und Matrizen.
Die Steueranweisungen legen den Ablauf der Simulation fest. Sie veranlassen die Prüfung der Eingabeanweisungen, die Übersetzung, die Ausführung und den Abschluß eines Simulations- bzw. SIMDIS-Programmes einschließlich Ausgabe von Ergebnissen und Zwischenergebnissen, das Löschen bzw. Aufbewahren verschiedener Werte nach einer Simulationsphase u. a.
Die Anweisungen für die benutzereigene Ausgabe ermöglichen die Ausgabe einer eigenen Zusammenstellung der Ausgabewerte anstelle der Standardausgabe bzw. zu deren Ergänzung.
Die EDO-Anweisungen erlauben die Ein- und Ausgabe von Daten auf externe(n) Dateien. Ausgabedaten von PS SIMDIS können von anderen PP/PS, z. B. von PP STATISTIK, weiterverarbeitet werden. Ebenso können EDO-Dateien eingelesen und weiterverarbeitet werden, die ein Ergebnis früherer Rechnungen waren. So können z. B. beliebige Daten von Häufigkeitsmessungen mittels der Anweisung EDFUNCTION eingelesen und als Verteilungsfunktion bereitgestellt werden.

Für die Nachbildung der den Prozeßablauf bestimmenden Ereignisse kann man unterscheiden

- die aktiven Elemente (die Ereignisse auslösenden Elemente) und
- die passiven Elemente (die an Ereignissen beteiligten Elemente).

Hinsichtlich des fortschreitenden Prozesses sind also dynamische und statische Elemente zu unterscheiden. Zu beiden Typen kommen Elemente, die die Ereignisoperation beschreiben, sowie Hilfsgrößen zu Berechnungen. Unterschieden wird in

- Operationselemente,
- dynamische Elemente,
- statische Elemente,
- Berechnungselemente.

Die Klasse der Operationselemente umfaßt 54 Blöcke, von denen hier nur eine Auswahl vorgestellt werden soll. Die meisten Blöcke dienen der Beschreibung von Ereignisoperationen im Simulationsmodell, z. B. Belegen einer Bedieneinrichtung, Verlassen eines Speichers u. a. Sie entsprechen in einem Programmablaufplan einem Vorgang. Die für einen Block spezifischen Operationen werden dann ausgelöst, wenn ein dynamisches Element (Aktivator) diesen Block betritt.
Die Klasse der dynamischen Elemente umfaßt die Aktivatoren. Sie sind die Handlungsträger bzw. die ereignisauslösenden Elemente im Modell des zu simulierenden Prozesses. Der Lauf der Aktivatoren entspricht den Flußlinien in einem Programmablaufplan. Rechentechnisch gesehen sind die Aktivatoren Organisationstabellen. Mit den gespeicherten Daten und Informationen können die erforderlichen Unterprogramme des PS SIMDIS ermittelt und ausgeführt werden.

Aktivatoren können nach Bedarf erzeugt und vernichtet werden. Jedem Aktivator kann eine Menge Parameter zugeordnet werden, in die durch Zahlen verschlüsselte Merkmale oder Eigenschaften des Aktivators eingetragen werden können. Ein Standardmerkmal eines Aktivators ist seine Zugehörigkeit zu einer Prioritätsklasse. Es gibt 128 Prioritätsklassen (0 - 127). Über diesen

Parameter (Kurzzeichen PR) kann auf die Reihenfolge der Abarbeitung der Aktivatoren durch die SIMDIS-Routinen Einfluß ausgeübt werden. Ferner ist jeder Aktivator Mitglied einer "Kette" (augenblickliche, zukünftige, Unterbrechungs- oder Benutzerkette).
Die statischen Elemente erfahren durch die dynamischen Elemente Zustands- und/oder Werteänderungen. Eine Einrichtung (einkanalige Bedieneinheit) kann durch einen Aktivator "BELEGT" oder "NICHTBELEGT" (also "FREI") sein, kann "UNTERBROCHEN" oder "NICHTUNTERBROCHEN" (also kann belegt werden) sein. Ebenso kann ein logischer Schalter "EIN" oder "AUS" sein, ein Speicher kann "VOLL"/"NICHT VOLL", "LEER"/"NICHT LEER" u. a. sein. Zu den statischen Elementen gehören folgende Elementetypen:

- Einrichtungen
- Speicher
- Warteschlagen
- logische Schalter
- Benutzerketten
- Gruppen
- Skalare
- Matrizen
- Tabellen
- Zufallszahlengeneratoren
- Systemuhr

Die Berechnungselemente sind Hilfselemente zur Berechnung des Wertes einer Größe nach einer vorgegebenen Rechenvorschrift während des Programmablaufes. Hierzu gehören

- Funktionen,
- arithmetische Variable,
- Boolesche Variable.

Einige statische Elemente und sämtliche Berechnungselemente müssen definiert werden. Dafür existieren Definitionsanweisungen.
Jedes in SIMDIS vorkommende Element einer Klasse bzw. eines Typs wird durch wesentliche Merkmale beschrieben, die während eines Programmdurchlaufes verschiedene Werte bzw. Zustände annehmen können. Viele dieser charakteristischen Merkmale können während der Simulation abgefragt werden. Es gibt

- numerische Standardsymbole (NSS, Symbole für numerische Merkmale) und

- <u>logische Standardsymbole</u> (LSS, Symbole für logische Merkmale).

Beispiele für solche Symbole werden unten angegeben. Im Anhang ist eine vollständige Übersicht dieser Symbole zu finden.
Der Zugriff zu solchen Merkmalen erfolgt durch die Angabe des entsprechenden Symbols in der Regel mit nachstehender Zahlenangabe. Die Zahlenangabe entspricht der Nummer eines Elements innerhalb der Elementeklasse bzw. des Elementetyps. Die nachstehende Zahlenangabe kann ebenso in Form eines symbolischen Namens oder mittels indirekter Indizierung erfolgen.
Symbolische Namen sind maximal fünf Zeichen lang, von denen die ersten 3 Buchstaben sein müssen; im Unterschied dazu bestehen die NSS aus zwei Buchstaben. Diesen Namen wird von SIMDIS eine Nummer zugeordnet, die als Adresse des Elementes eines Typs aufgefaßt wird. Der symbolische Name wird vom NSS/LSS mit dem Zeichen "#" getrennt.
Die indirekte Indizierung wird mittels des Zeichens "*" kenntlich gemacht. Hier muß während des Simulationsablaufes erst die Adresse (Nummer) des Elements eines Typs bestimmt werden.

Auf der Lochkarte ist eine SIMDIS-Anweisung wie folgt abzulochen:

Spalte	2 - 6	8 - 17	19 - 71
Bedeutung	Nummer/ Name	Operation	Operanden

Das Operandenfeld erstreckt sich von der 19. bis zur 71. Spalte, in wenigen Ausnahmen sind Folgekarten erlaubt. Die Operanden sind als Positionsoperanden aufzufassen, d. h. jede Position hat ihre Bedeutung entsprechend der Operation.
Operandeneintragungen können sein:

K	Konstante (max. 6 Zeichen für Festkommakonstante, max. 10 Zeichen für Gleitkommakonstante)
NSS bzw. NSSj	NSS mit oder ohne direktem Index j
NSS * NSSj	NSS mit indirektem Index NSSj
LSS bzw. LSSj	LSS mit oder ohne direktem Index j
LSS * NSSj	LSS mit indirektem Index NSSj

In der Übersicht zu den SIMDIS-Sprachelementen wird für diese Eintragungen kurz "st" geschrieben. Die Angabe von "st*" deutet auf eine Einschränkung der erlaubten NSB bzw. deren Adressierung hin. Genaueres darüber ist in [11] zu finden.

Beispiele für Operandeneintragungen:

SN3	augenblickliche Länge des Speichers 3
XB#ZAEHL	Wert des Byte-Skalares ZAEHL
SN * XB#ZAEHL	augenblickliche Länge des Speichers, dessen Nummer durch den augenblicklichen Wert des Byte-Skalares ZAEHL festgelegt wird.

Beispiele für numerische Konstanten:

101	
.12345E-4	bedeutet: $0.12345 * 10^{-4}$
10.279E+10	bedeutet: $0.10279 * 10^{12}$
-.9876	

Innerhalb eines SIMDIS-Programmes werden alle Blöcke automatisch in aufsteigender Folge numeriert. Jedem Block kann wie jedem Elementetyp (außer Aktivatoren) ein symbolischer Name zugeordnet werden. Über diese Namen kann der Programmierer Sprungbefehle und andere Bezugnahmen auf anzusprechende Blöcke realisieren.

Beispiel:

Die ersten 100 Zufallszahlen zwischen 0 und 1 sollen in einer Matrix (Gleitkomma) gespeichert werden.

```
       SIMULATE
RAND   MATRIX      ML,10,10                MATRIX  (10,10)
       GENERATE    ,,,1,,2PH               ERZEUGEN AKTIVATOR
       ASSIGN      1,1,,PH                 HALBWORT-PARAMETER 1 = 1
ZEILE  ASSIGN      2,1,,PH                 HALBWORT-PARAMETER 2 = 1
       MSAVEVALUE  RAND,PH1,PH2,RL1,ML     RL1 IST ZUFALLSZAHL
       LOOP        2PH,10,,ZEILE+1         ERHOEHEN SPALTENINDEX BIS 10
       LOOP        1PH,10,,ZEILE           ERHOEHEN ZEILENINDEX BIS 10
       TERMINATE   1
       START       1
```

In diesem Beispiel steht RAND für die Gleitkommamatrix und ZEILE für den 3. Block, d. h. ZEILE hat den Wert 3. Die Adresse ZEILE+1 verweist demzufolge auf den 4. Block (MSAVEVALUE). Die Anweisung MATRIX ist eine Definitionsanweisung, mit der der Typ (hier Gleitkomma) und die Anzahl der Zeilen und Spalten der Matrix RAND festgelegt werden. Die beiden LOOP-Blöcke erhöhen jeweils die Parameterwerte um 1 und der den Block betretende Aktivator setzt die Bearbeitung bei dem im 4. Operanden angegebenen Block solange fort, bis die Parameterwerte größer als 10 sind. Mittels MSAVEVALUE werden die Zufallszahlen zeilenweise in der Matrix RAND gespeichert.

Das PS SIMDIS arbeitet nach dem Prinzip der ereignisorientierten Steuerung. Die zeitliche Aufeinanderfolge und die Reihenfolge der Bewegung von Aktivatoren durch das Modell wird mittels einer fiktiven Uhr, der sogenannten Systemuhr, geregelt. Die Bewegung eines Aktivators zu einem bestimmten Zeitpunkt durch das Modell entspricht einem Ereignis. Ereignisse sind z. B.:

- Erzeugung und Vernichtung von Aktivatoren,
- Änderungen von Werten oder Merkmalen der Elemente,
- Änderung des Zustandes eines Aktivators,
- Änderung der Zugehörigkeit von Aktivatoren zu Ketten, Gruppen und Verbänden.

Mehrere Ereignisse zu ein und demselben Zeitpunkt bedeuten vielfach die Bewegung von mehreren Aktivatoren. Bei gleicher Stellung der Systemuhr (Uhrzeit) werden die Aktivatoren nacheinander durch das mit dem Modell dargestellte System bewegt.

Dem internen Steuersystem des PS SIMDIS stehen für die Steuerung des zeitlichen Ablaufes zwei wesentliche Systemeinrichtungen zur Verfügung:

- eine Liste der miteinander verketteten zukünftigen Ereignisse, genannt zukünftige Ereigniskette (ZEK);
- eine Liste der miteinander verketteten augenblicklichen Ereignisse, genannt augenblickliche Ereigniskette (AEK).

Die Steuerung des Ablaufes wird ferner durch die dem jeweiligen Aktivator zugeordnete Prioritätsklasse beeinflußt.

Zu Beginn der Simulation wird die ZEK aufgebaut. In ihr befinden sich die nach den Ereigniszeitpunkten T1, T2, ..., Tn geordneten Aktivatoren. Dies ist jeweils nur der erste Aktivator eines jeden GENERATE-Blockes.
Der erste Ereigniszeitpunkt T1 wird auf der Systemuhr eingestellt. Alle diesem Zeitpunkt zugeordneten Aktivatoren werden aus der ZEK entfernt und in die AEK eingeordnet entsprechend ihrer Priorität. Dort werden die Aktivatoren beginnend bei der höchsten Prioritätsklasse nacheinander abgearbeitet. Der an zweiter Stelle stehende Aktivator wird erst dann bewegt, wenn der erste in der AEK befindliche Aktivator in seiner Bewegung aufgehalten wird oder aus dem System ausgeschieden ist. Dieses Prinzip gilt auch für die folgenden Aktivatoren der AEK. Dabei kann vorgesehen werden, daß einzelne Aktivatoren erneut in die ZEK aufgenommen und später liegenden Ereigniszeitpunkten zugeordnet werden; Bedienungs- und Verweilvorgänge werden auf diese Weise realisiert. Ist in der AEK eine Bewegung von Aktivatoren nicht mehr möglich, wird die Systemuhr auf den Ereigniszeitpunkt T2 gestellt. Die T2 zugeordneten Aktivatoren der ZEK werden aus ihr entfernt und in die AEK übernommen. Anschließend erfolgt die Abarbeitung. Diese Vorgänge wiederholen sich solange, bis eine Ende-Bedingung erreicht wird.

2.2. Vorstellung ausgewählter SIMDIS-Sprachelemente

Im folgenden werden einige Anweisungen kurz inhaltlich beschrieben, die im Kapitel 4 eine praktische Verwendung finden. Diese Beschreibung ist jedoch nicht vollständig. Im Kapitel 4 wird auf einige Anweisungen genauer eingegangen. Außerdem informiert der Anhang, in dem ausgewählte Anweisungen zusammengestellt sind, über die Bedeutung der Operanden und die Möglichkeiten der Eintragungen in die Operanden.

Blockanweisungen:

GENERATE erzeugt Aktivatoren, deren Zwischenankunftszeiten nach empirischen oder theoretischen Verteilungen oder nach anderen Gesetzmäßigkeiten berechnet werden können. Jeder dieser Aktivatoren hat eine bestimmte Anzahl von Parametern (Vollwort, Halbwort, Byte oder Gleitkomma) und kann verschiedenen Prioritätsklassen (0 - 127) zugeordnet werden.

ADVANCE erlaubt das Verzögern von Aktivatoren entsprechend einer gewünschten Verteilung oder anderen Gesetzmäßigkeit. Betritt ein Aktivator einen ADVANCE-Block, so verbleibt er bis zu dem berechneten Ereignispunkt in diesem Block, ohne daß er Zustandsänderungen im System vornehmen kann. Erst wenn die Systemuhr auf diesen Zeitpunkt vorgerückt ist, kann dieser Aktivator den ADVANCE-Block verlassen.

TERMINATE vernichtet jeden den Block betretenden Aktivator und verringert den Startzähler (vgl. START) und den im Operanden A angegebenen Wert.

SPLIT erzeugt von dem den Block betretenden Aktivator eine gewünschte Menge von Kopien, die Eigenschaften des Originals (vollständig oder teilweise) übernehmen. Ferner können das Original und die Kopien fortlaufend numeriert werden.

ASSEMBLE sammelt erzeugte Kopien. Der erste den Block betretende Aktivator bleibt vollständig erhalten, während alle weiteren, die bis zum Erreichen des Wertes im Operanden A ankommen, vernichtet werden.

PRIORITY gestattet die Änderung der Zugehörigkeit eines Aktivators zu einer Prioritätsklasse. Damit können Vorrangsituationen simuliert, zufällige Reihenfolge in der Abarbeitung der Aktivatoren u. dgl. m. erzeugt werden.

ASSIGN ordnet den Parametern des den Block betretenden Aktivators numerische Werte zu.

MSELEC erlaubt im Zusammenhang mit Vergleichsoperatoren die Auswahl von Werten aus einer Matrix bzw. Untermatrix.

QUEUE	registriert den Eintritt eines Aktivators in eine Warteschlange.
DEPART	registriert den Austritt eines Aktivators aus einer Warteschlange.
SEIZE	erlaubt einem Aktivator das Betreten einer einkanaligen Bedienstelle (Einrichtung), sofern diese nicht schon belegt ist. Vor solch einer Bedienstelle kann eine Warteschlange von Aktivatoren entstehen.
RELEASE	registriert das Verlassen einer einkanaligen Bedienstelle durch den belegenden Aktivator. Die Bedienstelle kann durch wartende Aktivatoren wieder belegt werden.
ENTER	simuliert das Betreten einer mehrkanaligen Bedienstelle (Speicher), das bis zu ihrer Kapazitätsgrenze möglich ist. Bei vollem Speicher oder bei nicht mehr unterzubringenden Mengen bildet sich vor diesem Block eine Warteschlange von Aktivatoren.
LEAVE	simuliert die Entnahme von Mengen aus einem Speicher bzw. das Verlassen einer mehrkanaligen Bedienstelle. Hier kann nur soviel aus dem Speicher entnommen werden, wie der augenblickliche Inhalt zuläßt.
LINK	erlaubt das Entfernen von Aktivatoren aus der AEK und die Einordnung in eine Benutzerkette nach vorgegebenen Regeln (z. B. FIFO, LIFO).
UNLINK	erlaubt das Entfernen von Aktivatoren aus einer Benutzerkette unter Beachtung gewisser Auswahlkriterien und die Einordnung in die AEK als jeweils letzte ihrer Prioritätsklasse.
LOGIC	bringt bei Betreten eines Aktivators einen logischen Schalter in eine der Stellungen EIN oder AUS.

GATE	erlaubt den Test von Zuständen bestimmter Systemelemente und Verzweigungen in Abhängigkeit vom Testergebnis.
TEST	erlaubt den Vergleich von numerischen Werten untereinander und Verzweigungen in Abhängigkeit vom Testergebnis.
TRANSFER	erlaubt durch bedingte und unbedingte Verzweigungen Abweichungen vom sequentiellen Programmablauf.
LOOP	erlaubt das mehrmalige Abarbeiten eines Programmteiles.
TABULATE	erfaßt spezielle, statistisch interessierende Daten in einer definierten Tabelle oder einer EDO-Datei.
SAVEVALUE	ordnet Skalaren numerische Werte zu.
MSAVEVALUE	ordnet Matrixelementen numerische Werte zu.

Definitionsanweisungen:

FUNCTION	definiert diskrete, stetige oder Listenfunktionen durch Angabe ihrer Stützstellen.
VARIABLE	definiert numerische Ausdrücke, die bei Aufruf berechnet werden.
BVARIABLE	definiert logische Ausdrücke, die bei Aufruf berechnet werden.
INITIAL	stellt Anfangswerte für Skalare, Matrixelemente, Speicherkapazitäten, Zufallsgeneratoren und Schalterstellungen bereit.

Steueranweisungen:

SIMULATE	veranlaßt nach der Programmübersetzung die Ausführung des SIMDIS-Programmes.
START	veranlaßt den Beginn einer Simulationsphase und stellt den sogenannten Startzähler. Betreten Aktivatoren einen TERMINATE-Block, dessen 1. Operand eine positive Zahl enthält, wird diese Zahl vom

	Startzähler subtrahiert. Sobald der Startzähler kleiner oder gleich Null ist, endet der Simulationslauf, und sofern nicht unterdrückt, folgt die Ausgabe aller bisher gesammelten Daten.
CLEAR	löscht den Zustand des Simulationsmodells. Im Operandenfeld angegebene Skalare oder Matrizen bleiben erhalten.
RESET	löscht statistische Werte, erhält jedoch den Systemzustand. Für im Operandenfeld eingetragene Elemente bleiben die statistischen Werte erhalten.
END	ist die letzte Anweisung in einem SIMDIS-Programm.

2.3. Anwenderunterstützungen einschließlich Dialogbetrieb

Das PS SIMDIS unterstützt mit seiner Sprache den Anwender nicht nur bei der Programmierung seines Modelles, sondern auch bei der Testung des Modelles und beim Experimentieren mit dem Modell. Die wichtigsten Programmierhilfen sind Anweisungen zum Ändern bereits bestehender Programme (CREATE, UPDATE) und zur Definition und zum Aufruf von häufig wiederkehrenden Programmteilen (STARTMACRO, ENDMACRO, MACRO).
Während der Simulation erleichtern gezielte Druckausgaben die Testung des Modells (TRACE, PRINT). Die standardmäßige Ergebnisausgabe umfaßt eine Vielzahl von Daten und Statistiken, die während der Simulation erfaßt wurden. Diese Ergebnisausgabe kann durch benutzereigene Ausgaben vervollständigt werden, z. B. durch graphische Darstellungen von Histogrammen, Verteilungsfunktionen u. a. m. Eine weitere große Hilfe für den Anwender ist die ausführliche Fehlerdiagnostik sowohl während der Übersetzung als auch während der Simulation.
Für mit dem PS SIMDIS nicht zu lösende oder nur umständlich zu lösende Programmteile können vom Anwender selbst geschriebene ASSEMBLER-, FORTRAN- und PL 1-Programme integriert und aufgerufen werden. Eine Vielzahl von Anweisungen erlaubt die Definition, das Lesen und Schreiben von bzw. auf EDO-Dateien (einheitliche Dateiorganisation). Mittels EDO ist auch eine daten-

seitige Kopplung eines SIMDIS-Programmes mit Programmen anderer PP/PS möglich. So können z. B. vom PS SIMDIS ermittelte Verteilungen über eine EDO-Datei an das PP STATISTIK zwecks statistischer Auswertung übergeben werden.
Der Dialogbetrieb als Mittel zur direkten Kommunikation zwischen Rechner und Programmierer wird vom PS SIMDIS gegenwärtig nicht unterstützt.

Kontrollfragen

1. a) Welche Bedeutung haben Zufallszahlen für die Simulation? Untersuchen Sie die Anwendbarkeit von Zufallszahlengeneratoren, die reproduzierbare und nichtreproduzierbare Zufallszahlenfolgen liefern, in Simulationsmodellen.

   b) Geben Sie den Algorithmus für die Transformation von im Intervall $[0,1)$ gleichverteilten Zufallszahlen in Zufallszahlen an, die im Intervall $[1,7)$ gleichverteilt sind. Die Transformation soll sowohl stetig als auch diskret (nur ganze Zahlen) durchgeführt werden. Fertigen Sie dazu jeweils einen Programmablaufplan an.

2. Stellen Sie aus Ihrem Arbeits- oder Erlebnisbereich ein Bedienungsmodell in Form eines Programmablaufplanes dar. Verwenden Sie dabei die Elemente Warteschlange, Einrichtung (für einkanalige Bedienstelle) und Speicher (für mehrkanalige Bedienstelle). Welche Daten bzw. Statistiken sind in Ihrem Modell für die verwendeten Elemente von Interesse? Vergleichen Sie dazu die Menge der vom PS SIMDIS automatisch erfaßten Daten bzw. Statistiken.

3. Ausgewählte Anwendungsbeispiele

Schwerpunkt der Anwendung des PS SIMDIS ist gegenwärtig und künftig die weitere Vervollkommnung von Produktionsprozessen. Dazu soll im weiteren die Simulation des Ablaufs eines Fertigungsprozesses, wie er beispielsweise für Vorfertigungsbereiche in Betrieben der metallverarbeitenden Industrie typisch ist, dargestellt werden.

Der kontinuierliche Ablauf von Produktionshauptprozessen wird maßgeblich durch die Organisation der Hilfsprozesse beeinflußt. Als zweites Beispiel soll deshalb die Simulation eines Instandhaltungsprozesses vorgestellt werden.

3.1. Simulation eines diskreten Fertigungsprozesses

Diese Aufgabenstellung kann infolge der möglichen Vielfalt der technologischen, organisatorischen und ökonomischen Voraussetzungen und Bedingungen des Prozeßablaufes in vielen Modifikationen auftreten. Als Grundaufgabe sollen deshalb folgende Bedingungen und Zielstellungen betrachtet werden.

1. Aufgabenstellung

a) Bedingungen:

In einem Fertigungssystem mit I Fertigungsstufen (S_1, S_2, ..., S_i, ..., S_I) sollen J verschiedene Werkstücke (A_1, A_2, ..., A_j, ..., A_J) bearbeitet werden.
Für die Verfügbarkeit der Werkstücke zur Bearbeitung im Fertigungssystem werden bekannte Zeitpunkte angenommen:

$$\underline{a} = (a_1, a_2, \ldots, a_j, \ldots, a_J)$$

Für die Werkstücke sind Fertigstellungstermine (Termine für den Abschluß der letzten Bearbeitung) vorgegeben und durch den tatsächlichen Ablauf einzuhalten:

$$\underline{f} = (f_1, f_2, \ldots, f_j, \ldots, f_J)$$

Bei Überschreitung der Fertigstellungstermine ist in Abhängigkeit vom Grad des Verzuges eine Strafe in Höhe von x_i Kosteneinheiten pro Zeiteinheit Terminüberschreitung zu zahlen:

$$\underline{x} = (x_1, x_2, \ldots, x_j, \ldots, x_J)$$

Die Werkstücke durchlaufen im Fertigungssystem mehrere Stufen, wobei in jeder dieser Stufe eine Bearbeitung erfolgt. Jede der Fertigungsstufen kann in der Praxis aus einer oder aus mehreren Maschinen vergleichbarer Art (Maschinengruppe) bestehen. Das Fertigungssystem wird für einen bestimmten Zeitraum, den Planzeitraum, betrachtet.

Die Einsatzbereitschaft der Fertigungsstufen kann zu Beginn des Planzeitraumes z. B. durch eine noch bestehende Belegung aus dem vorhergehenden Planzeitraum oder durch Instandhaltungsmaßnahmen eingeschränkt sein.
Deshalb wird als frühester Termin der Einsatzbereitschaft der i-ten Fertigungsstufe v_i definiert:

$$\underline{v}^T = (v_1, v_2, \ldots, v_i, \ldots, v_r)$$

Die Werkstücke durchlaufen die Fertigungsstufen zur Bearbeitung. Dabei besitzt jedes Werkstück eine sogenannte <u>technologische Reihenfolge</u>.
Sie gibt an, in welcher Reihenfolge das jeweilige Werkstück die Fertigungsstufen durchläuft.
Die technologischen Reihenfolgen können in einer Matrix, wir bezeichnen sie mit $\underline{Q}$, zusammengefaßt werden.

$$\underline{Q} = \left\| q_{ij} \right\|_{IJ}$$

Jede Spalte dieser Matrix stellt eine technologische Reihenfolge dar, z. B. die j-te Spalte die technologische Reihenfolge des Werkstückes A_j.
Demzufolge bedeutet zum Beispiel $q_{ij} = 3$, daß das Werkstück A_j seine dritte Bearbeitungsoperation in der Fertigungsstufe S_i erfährt.
Die Bearbeitungszeiten werden in der Matrix $\underline{T}$ angegeben:

$$\underline{T} = \left\| t_{ij} \right\|_{IJ}$$

Jedes Werkstück wird nach Abschluß seiner Bearbeitung in einer Fertigungsstufe zu der Fertigungsstufe weitergegeben, in der die nächste Bearbeitungsoperation zu erfolgen hat.
Ist dabei eine Fertigungsstufe durch ein anderes Werkstück belegt, dann muß das in der Fertigungsstufe eintreffende Werkstück warten.

b) <u>Zielstellung</u>:

Die ökonomischen Zielstellungen können im jeweiligen konkreten Fall sehr unterschiedlich sein.

Sie leiten sich aus den gesamtbetrieblichen Aufgaben und Effektivitätskriterien ab.
Im vorliegenden Fall wird orientiert auf

- die Minimierung von Fertigungsterminüberschreitungen und der damit entstehenden Strafe (F_1);
- die Maximierung der Auslastung des Fertigungssystems bei der Bearbeitung des Werkstücksortiments (entspricht der Minimierung der Wartezeiten der Arbeitskräfte und der Stillstandszeiten der Maschinen) (F_2);
- die Minimierung der Gesamtaufenthaltsdauer der Werkstücke im System (entspricht der Minimierung der Umlaufmittelbindung) (F_3).

Diese Zielstellungen sind für die Berechnungen mathematisch zu formulieren.
Für F_1 werden alle Werkstücke bestimmt, für die bei Vorliegen einer Lösung der Fertigstellungstermin überschritten wird. Der reale Fertigstellungstermin des j-ten Werkstückes lautet:

$$\max_{\forall i} \left(\sum_{k=1}^{K} x_{ijk} \cdot k \right)$$

Dazu wird eine diskrete Zeitskala zu den Zeitpunkten $k = 1\ (1)\ K$ untersucht, wobei gilt:

$$x_{ijk} = \begin{cases} 1 & \text{wenn zum Zeitpunkt } k \text{ in der i-ten Fertigungsstufe die Bearbeitung des j-ten Werkstücks beendet wird} \\ 0 & \text{in allen anderen Fällen} \end{cases}$$

$$K = \sum_{i=1}^{I} \sum_{j=1}^{J} t_{ij} \qquad \text{Gesamtdauer des Werkstückdurchlaufes im Fall, daß sich jeweils nur ein Werkstück im Fertigungssystem befindet}$$

Liegt eine Fertigungsterminüberschreitung vor, dann gilt:

$$\max \left(0, \max_{\forall i} \left(\sum_{k=1}^{K} x_{ijk} \cdot k \right) - f_j \right) > 0$$

Diesen Ausdruck multipliziert mit der Strafe bei Terminverzug, erhalten wir die Zielfunktion:

$$F_1 = \sum_{j=1}^{J} \left[x_j \cdot \max \left(0, \max_{\forall i} \left(\sum_{k=1}^{K} x_{ijk} \cdot k\right) - f_j\right) \right] \rightarrow \min$$

Für F_2 und F_3 können folgende Formeln entwickelt werden:

$$F^2 = \sum_{i=1}^{I} \left[\max_{\forall j} \left(\sum_{k=1}^{K} x_{ijk} \cdot k\right) - v_i \right] \longrightarrow \min$$

$$F_3 = \sum_{j=1}^{J} \left[\max_{\forall i} \left(\sum_{k=1}^{K} x_{ijk} \cdot k\right) - a_j \right] \longrightarrow \min$$

Die vorliegende Aufgabe gehört zur Klasse der kombinatorischen Optimierungsaufgaben, die nur mit Schwierigkeiten in einem analytischen Modell abgebildet werden können.
Bei der Berechnung der Optimierungsaufgaben ist eine solche Lösung zu ermitteln, die einen möglichst kleinen Wert der jeweils zugrunde gelegten Zielfunktion ergibt.
Die Lösung hat bei diesem Aufgabentyp die Form einer sogenannten organisatorischen Reihenfolge je Fertigungsstufe.
Unter organisatorischer Reihenfolge wird die Reihenfolge der Bearbeitung der einzelnen Werkstücke in der jeweiligen Fertigungsstufe verstanden.
Die Lösung wird in der Lösungsmatrix $\underline{P}$ angegeben.

$$\underline{P} = \left\| p_{ij} \right\|_{IJ}$$

Jede Zeile dieser Matrix gibt eine organisatorische Reihenfolge an, wobei der Zeilenindex i die Fertigungsstufe definiert, für die die Reihenfolge Gültigkeit besitzt.
Der Wert $p_{ij} = 2$ gibt damit z. B. an, daß das Werkstück A_j in der Fertigungsstufe S_i als zweites bearbeitet wird.
Die Lösung kann neben der Angabe mittels der Matrix $\underline{P}$ auch graphisch mit Hilfe des sogenannten Gantt-Balkendiagramms dargestellt werden.

Zum Verständnis der allgemeingültig formulierten Aufgabenstellung wollen wir ein einfaches Beispiel betrachten.

Beispiel:

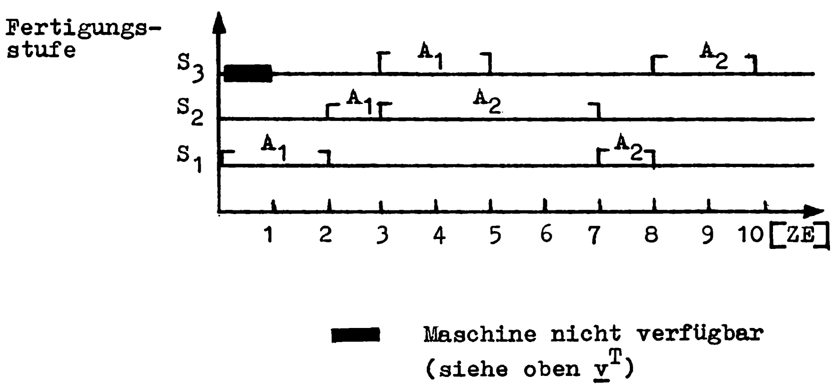
In einem Fertigungssystem, bestehend aus drei Fertigungsstufen mit jeweils einer Maschine, sind zwei Werkstücke zu bearbeiten. Dafür sind folgende Eingangsdaten gegeben:

$$\underline{a} = (0,2) \quad \underline{f} = (7,8) \quad \underline{x} = (2,4) \quad \underline{v}^T = (0,0,1)$$

$$\underline{Q} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 3 \end{pmatrix} \qquad \underline{T} = \begin{pmatrix} 2 & 1 \\ 1 & 4 \\ 2 & 2 \end{pmatrix}$$

In den Vektoren $\underline{a}$, $\underline{f}$, $\underline{v}^T$ sind Zeitpunkte angegeben, im Vektor $\underline{x}$ erfolgt die Angabe in Kosteneinheiten pro Zeiteinheit, die Matrix $\underline{T}$ enthält schließlich Angaben in Zeiteinheiten.
Im Beispiel liegen unterschiedliche technologische Reihenfolgen (voneinander abweichende Spalten in $\underline{Q}$) vor.
Demzufolge wird es praktisch nicht möglich sein, gleiche organisatorische Reihenfolgen zu ermitteln.
In einem solchen Fall gibt es insgesamt $(I\,!)^J$ verschiedene Lösungen. Für das Beispiel existieren daher $(3\,!)^2 = 36$ unterschiedliche Lösungsvarianten $P_1, P_2, \ldots, P_{36}$.

Bestimmen wir eine davon:
Wir nehmen an, daß in der jeweiligen Fertigungsstufe das Werkstück mit dem kleinsten Index j zuerst bearbeitet wird.
Die Lösung kann z. B. graphisch anhand eines Gantt-Balkendiagramms ermittelt werden, indem zunächst der erste und dann der zweite Auftrag unter Beachtung von $\underline{a}$ und $\underline{v}^T$ eingeplant wird.



Die Lösungsmatrix $\underline{P}$ lautet in diesem Fall:

$$\underline{P}_1 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{pmatrix}$$

Die Berechnung der Zielfunktionen ergibt:

$$F_1 = 2 \cdot 0 + 4 \cdot 2 = 8 \qquad [\text{Kosteneinheiten}]$$
$$F_2 = 8 + 7 + 9 = 24 \qquad [\text{Zeiteinheiten}]$$
$$F_3 = 5 + 8 = 13 \qquad [\text{Zeiteinheiten}]$$

<u>Zur Übung</u>

Stellen Sie für die Lösungsvarianten

$$\underline{P}_2 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 2 \end{pmatrix} \qquad \underline{P}_3 = \begin{pmatrix} 2 & 1 \\ 2 & 1 \\ 1 & 2 \end{pmatrix}$$

die Gantt-Balkendiagramme auf und berechnen Sie für $\underline{P}_2$ und $\underline{P}_3$ die Werte von F_1, F_2, F_3.
Welche Lösung von $\underline{P}_1$, $\underline{P}_2$, $\underline{P}_3$ ist nach der Zielfunktion F_2 die effektivste?

2. Modellierung und Algorithmierung

Da zwar eine mathematisch-analytische Darstellung der Zielfunktionen und mit einigen Schwierigkeiten auch der Restriktionen möglich ist, dafür aber kein Lösungsverfahren vorliegt, wird ein Simulationsmodell entwickelt.
Spezifisch für Simulationsmodelle ist die Zusammenfassung der Modellierung und Algorithmierung in einem Schritt. Sie erfolgt in einem sogenannten algorithmischen Simulationsmodell zur Lösung des Problems der Fertigungssteuerung.
Da das PS SIMDIS eine blockorientierte Sprache besitzt, ist es zweckmäßig, das jeweilige Simulationsmodell in einem <u>Ablaufplan</u> darzustellen.
Für das Simulationsmodell zur Planung und Steuerung eines Fertigungsablaufes kann das im Bild 2 dargestellte allgemeine Schema (Ablaufplan) entwickelt werden.

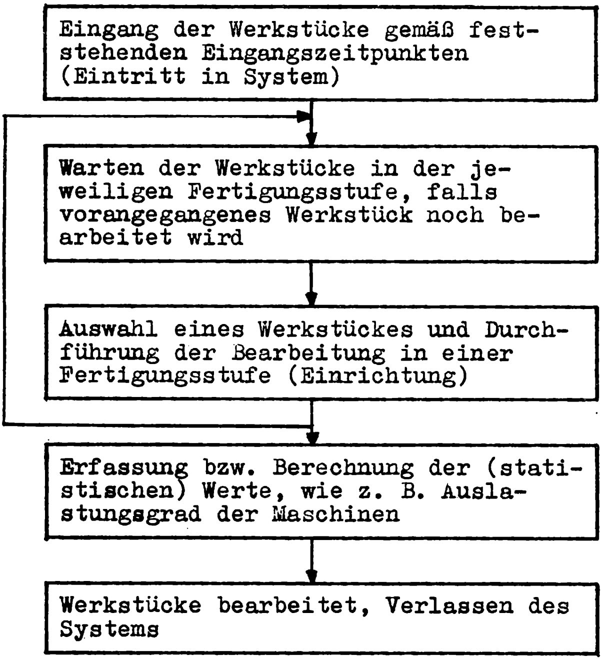


Bild 2: Ablaufplan für Fertigungsprozeßsteuerung

Das Simulationsmodell beinhaltet zwei Komponenten:

1. die Darstellung des Prozeßablaufes,
2. die Lösung von Entscheidungssituationen.

Die 2. Komponente findet an den Stellen des Simulationsmodells Anwendung, wo der weitere Ablauf des Fertigungsprozesses nicht eindeutig aus den Modellbedingungen abgeleitet werden kann. Das trifft hier zu, wenn eine Fertigungsstufe frei wird und mehrere Werkstücke auf die Bearbeitung warten (ihre gleichzeitige Bearbeitung ist nicht möglich).
Als Entscheidungskriterium (Verfahren zur Auswahl eines Werkstückes aus der existierenden Warteschlange) werden Prioritätsregeln verwendet.
Prioritätsregeln sind als Strategien zu verstehen, bestimmte Verhaltensweisen eines Fertigungsprozesses durch die mathematische Formulierung von logischen Zusammenhängen, Vermutungen,

Analogien, Hypothesen usw. auf der Grundlage bestimmter Parameter des Prozesses zu realisieren.
Das geschieht, indem jedem in eine Warteschlange vor einer Fertigungsstufe eingeordneten Werkstück in Abhängigkeit von den in der Prioritätsregel berücksichtigten Werkstück - und/ oder arbeitsgangbezogenen Kenngrößen (Bearbeitungszeit, Vertragsstrafe, Fertigungsrestzeit, Anzahl der noch ausstehenden Arbeitsgänge usw.) eine Dringlichkeitsziffer zugeordnet wird. Diese entscheidet über die Rangfolge jedes Werkstückes bei seiner Auswahl zur Bearbeitung aus der Warteschlange.
In der Literatur sind dazu die unterschiedlichsten Prioritätsregeln entwickelt und untersucht worden.
Hier sollen drei Varianten von Prioritätsregeln Anwendung finden:

Variante I: Prioritätsregel FIFO (first in first out)
Das erste eingetroffene Werkstück wird zur Bearbeitung eingesteuert. Ein gleichzeitiger Eintritt mehrerer Werkstücke geschehe dabei mit "aufsteigendem Index".

Variante II: Prioritätsreihe aus den Prioritätsregeln KOZ (kürzeste Bearbeitungszeitregel) - FIFO
Aus der Warteschlange wird das Werkstück ausgewählt, dessen Bearbeitungszeit auf der entsprechenden Maschine am geringsten ist.
Handelt es sich hier um mehrere Werkstücke, dann erfolgt die Anwendung von FIFO.

Variante III: Prioritätsregel SLACK (Verspätungsregel)

$$SLACK_j^{t_{li}} = f_j - (t_{li} + \sum_{\forall \underline{i}} t_{\underline{i}j})$$

= Fertigstellungstermin - (Ist-Zeitpunkt + Zeit der noch auszuführenden Arbeitsgänge am Werkstück)

$\underline{i}$ - Index für die noch nicht ausgeführten Arbeitsgänge. Ausgewählt wird das Werkstück mit dem kleinsten SLACK-Wert.

Die Aufgabenstellung besteht nun darin, mittels eines SIMDIS-Programms

1. mit Hilfe der verschiedenen Prioritätsregeln eine Lösung zu finden und für diese Lösung die einzelnen Zielfunktionswerte zu bestimmen,
2. die wirksamste Prioritätsregel anhand der besten Lösung zu ermitteln,
3. für die beste (optimale) Lösung die Auswertung der ökonomischen Kennziffern vorzunehmen (Verweilzeit der Werkstücke im System, Stillstandszeit der Maschinen im Intervall vom Beginn des ersten bis zum Abschluß des letzten Arbeitsganges auf jeder von ihnen, Terminüberschreitungen bei der Fertigstellung und daraus entstehende zusätzliche Kosten).

3. Numerisches Anwendungsbeispiel

In einem Fertigungssystem mit 4 Fertigungsstufen, in denen jeweils eine Maschine zur Verfügung steht, sind 6 Werkstücke zu bearbeiten. Die Ankunft der Werkstücke erfolgt zu den Zeitpunkten

$$\underline{a} = (0, 0, 0, 0, 0, 0),$$

d. h. alle Werkstücke liegen zu Beginn des Planzeitraumes (Stand der Simulationsuhr = 0) im System vor.

Als Fertigstellungstermine sind angegeben:

$$\underline{f} = (18, 15, 15, 10, 10, 10)$$

Dabei ist bereits abzusehen, daß diese Termine so eng gestellt sind, daß eine Einhaltung kaum möglich erscheint. Die Zahlung von Vertragsstrafen ist nicht vorgesehen, falls ein Fertigstellungstermin nicht zu halten ist.
Daraus ist bei der Aufbereitung der Daten abzuleiten:

$$\underline{x} = (1, 1, 1, 1, 1, 1)$$

Die Verfügbarkeit der Maschinen zu Beginn des Planzeitraumes soll bei allen Maschinen außer S_2 gegeben sein. Die 2. Maschine (S_2) ist erst eine Zeiteinheit nach Beginn des Planzeitraumes verfügbar:

$\underline{v}^T = (0, 1, 0, 0)$

Die technologische Reihenfolge der Werkstücke ist aus der Matrix $\underline{Q}$ zu entnehmen:

$$\underline{Q} = \begin{array}{c|cccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\ \hline S_1 & 1 & 2 & 3 & 1 & 4 & 2 \\ S_2 & 2 & 4 & 1 & 2 & 1 & 1 \\ S_3 & 4 & 1 & 2 & 3 & 3 & 4 \\ S_4 & 3 & 3 & 4 & 4 & 2 & 3 \end{array}$$

Hieraus ergibt sich, daß beispielsweise das erste Werkstück (siehe 1. Spalte) die Fertigungsstufen in der Reihenfolge $S_1 - S_2 - S_4 - S_3$ durchläuft.

Für die Bearbeitungszeiten ist die Matrix $\underline{T}$ definiert:

$$\underline{T} = \begin{array}{c|cccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\ \hline S_1 & 2 & 5 & 3 & 4 & 5 & 1 \\ S_2 & 5 & 1 & 1 & 2 & 2 & 0 \\ S_3 & 7 & 5 & 1 & 3 & 7 & 3 \\ S_4 & 1 & 2 & 6 & 1 & 0 & 1 \end{array}$$

Für die vorliegende Aufgabenstellung ist nun das entsprechende SIMDIS-Programm zu entwickeln (s. 4.2.).
Die im numerischen Beispiel angeführten Daten werden als Eingangsdaten für eine konkrete Berechnung mit diesem SIMDIS-programm genutzt.

4. Erweiterungen in der Aufgabenstellung

Entsprechend den praktischen Bedingungen und den Anforderungen an das Simulationsmodell sind Erweiterungen

- der Modellbedingungen und Zielstellungen sowie
- der Methoden zur Ermittlung der Reihenfolge der Werkstücke in einer Warteschlange - im einfachsten Fall bedeutet dies Veränderung der Prioritätsregeln - möglich.

Es können zum Beispiel folgende Erweiterungen angenommen werden:

- Alle oder ausgewählte Zeitangaben erhalten durch Angabe von Verteilungsfunktionen zufälligen Charakter.
  Im vorliegendem Beispiel sollen die Bearbeitungszeiten gleichverteilt im Intervall $[1,7]$ liegen und ganzzahlig sein.

- Beim Einsatz von Maschinen sind Ausfälle und Sperrzeiten für PVI-Maßnahmen zu beachten.
  Im vorliegenden Beispiel sollen Maschinenausfälle auftreten, deren Wahrscheinlichkeitsverteilungen für die Dauer zwischen zwei Maschinenausfällen und für die Dauer der Stillstandszeiten nach eingetretener Störung durch die Exponentialverteilung determiniert werden.

- Der Werkstückankunftsprozeß kann stochastisch z. B. als Poissonstrom mit der Ankunftsrate λ erfolgen.

Darüber hinaus können fertigungstechnologische, -organisatorische und ökonomische Veränderungen eintreten:

- Jede Fertigungsstufe besteht aus mehreren Maschinen mit unterschiedlicher Produktivität.

- Die Werkstücke werden nicht einzeln, sondern als Lose in den Fertigungsprozeß eingesteuert.

- Die Weitergabe erfolgt nach dem Prinzip des Parallelverlaufes, d. h. die ersten Werkstücke eines Loses werden bereits weitergeleitet, wenn noch andere Teile auf der jeweiligen Maschine bearbeitet werden.

In der Praxis treten darüber hinaus vielfältige andere Erweiterungen der Bedingungen auf.

<u>Kontrollfragen</u>

1. Nennen Sie weitere sinnvolle Prioritätsregeln!

2. In welchen Fällen ist der Übergang von der deterministischen zur stochastischen Simulation erforderlich?

3. Warum sind bei der stochastischen Simulation wesentlich mehr Simulationsversuche erforderlich als bei der deterministischen Simulation mit variierbaren Parametern?

4. Stellen Sie die Lösung $\underline{P}$ der nachfolgenden Aufgabe graphisch dar! Werten Sie nach F_1, F_2 und F_3 aus!

$$\underline{P} = \begin{pmatrix} 1 & 5 & 4 & 2 & 3 \\ 2 & - & 4 & 3 & 1 \\ 2 & 1 & - & - & 3 \\ 2 & 4 & 3 & 1 & - \\ 3 & 5 & 1 & 4 & 2 \end{pmatrix} \qquad \begin{aligned} \underline{a} &= (0,\ 5,\ 1,\ 0,\ 2) \\ \underline{f} &= (30,\ 20,\ 25,\ 19,\ 40) \\ \underline{x} &= (10,\ 11,\ 8,\ 5,\ 8) \\ \underline{v}^T &= (0,\ 2,\ 0,\ 1,\ 0) \end{aligned}$$

$$\underline{Q} = \begin{pmatrix} 1 & 2 & 3 & 1 & 2 \\ 2 & 4 & 5 & 4 & 1 \\ 3 & 1 & 2 & 2 & 3 \\ 4 & 3 & 4 & 3 & 5 \\ 5 & 5 & 1 & 5 & 4 \end{pmatrix} \qquad \underline{T} = \begin{pmatrix} 5 & 1 & 5 & 2 & 2 \\ 1 & 0 & 5 & 3 & 1 \\ 3 & 1 & 0 & 0 & 2 \\ 2 & 6 & 4 & 4 & 0 \\ 7 & 1 & 3 & 1 & 3 \end{pmatrix}$$

3.2. Simulation eines Instandhaltungsprozesses

Für ein Fertigungssystem sind Fragen der Instandhaltung von Grundmitteln (Maschinen) zu untersuchen.
Von Maßnahmen vorbeugender Instandhaltung soll abstrahiert werden. Es sind Instandhaltungsmaßnahmen, die erst nach Ausfall mindestens einer Maschine einzuleiten sind (sogenannte Havarieinstandsetzung oder Schadensfallinstandsetzung), vorzubereiten.
Dazu kann folgende Aufgabenstellung formuliert werden.

Aufgabenstellung

Bedingungen:

Das Fertigungssystem besteht aus 62 Maschinen fünf unterschiedlicher Typen.
Statistische Untersuchungen der Ausfallhäufigkeit in jeder Maschinengruppe über einen ausreichend großen Zeitraum haben eine Ausfallintensität von λ_1, λ_2, λ_3, λ_4, λ_5 Maschinen der jeweiligen Gruppe/Zeiteinheit (10 Tage) ergeben.

Die Zahl der pro ZE (10 Tage) anfallenden Reparaturanforderungen besitze eine Poisson-Verteilung mit dem Parameter λ_i.

Die Zwischenankunftszeit zweier aufeinanderfolgender Forderungen aus dem Bereich eines jeweiligen Maschinentyps an die bestehende Reparaturbrigade (Zeit zwischen zwei aufeinanderfolgenden Ausfällen von Maschinen gleichen Typs) genügt einer Exponentialverteilung mit dem Mittelwert

$$\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}, \frac{1}{\lambda_4}, \frac{1}{\lambda_5}.$$

Die Analyse der erforderlichen Instandsetzungszeiten für die Maschinen der einzelnen Maschinentypen führt zur Schlußfolgerung, daß diese in bestimmten Intervallen gleichverteilt sind. Die Instandhaltungskapazität hängt von der veränderlichen Anzahl der Mechaniker einer Reparaturbrigade ab.

Als revelante Eingangsdaten werden angenommen:

Maschinentyp	Anzahl der Maschinen (Anzahl)	Ausfallquote (Maschinen/ 10 Tage)	Dauer der Reparatur (Tage/Reparatur)
1	16	3	1 - 2
2	4	1	3 - 4
3	10	1	1 - 5
4	17	2	2 - 5
5	15	1	1 - 8

<u>Zielstellung</u>:

Mit Hilfe der Simulation ist eine optimale Zahl von Mechanikern zu ermitteln, die eine effektive Auslastung der Maschinen im Produktionsprozeß gewährleistet.

Für die einzelnen Varianten und die optimale Lösung sind auszuwerten:

- die Wartestillstandszeit der Maschinen je Typ,
- die Reparaturstillstandszeit der Maschinen je Typ,
- die Laufzeit der Maschinen,
- die Anzahl der Reparaturen im Simulationszeitraum,

- die maximale Anzahl gleichzeitig wartender Maschinen,
- die durchschnittliche Anzahl wartender Maschinen,
- die durchschnittliche Anzahl der beschäftigten Mechaniker.

Aus dem Verhältnis der Kennziffern, der durchschnittlichen Anzahl der beschäftigten Mechaniker und der durchschnittlichen prozentualen Laufzeit der Maschinen kann die kosten- und zeitgünstige Lösung abgeleitet werden.

4. Rechentechnische Realisierung der ausgewählten Beispiele mit dem PS SIMDIS

4.1. Rechentechnische Bedingungen

Die Lademoduln des PS SIMDIS werden dem Anwender auf einem Magnetband zur Verfügung gestellt. Nach dem Einrichten des PS SIMDIS sind diese Moduln in der Bibliothek VM1SOPS.LAD auf einem Magnetplattenstapel verfügbar. Die Übersetzung und Ausführung eines SIMDIS-Programmes geschieht in einem Jobschritt. Übersetzte SIMDIS-Programme stellen keine Lademoduln dar, sie sind nur unter Verwendung der Lademoduln des PS SIMDIS ausführbar.
Zur Übersetzung und Ausführung eines SIMDIS-Programmes sind folgende Jobsteueranweisungen erforderlich:

```
//...        JOB     ...
//SIMDIS     EXEC    PGM=VM#SIMØ,PARM=liste
//STEPLIB    DD      DSN=VM1SOPS.LAD,DISP=SHR
//SIMUT1     DD      UNIT=SYSDA,SPACE=(188,(500,100))
//SIMUT2     DD      UNIT=(SYSDA,SEP=SIMUT1),SPACE=(360,(500,100))
//SIMUT3     DD      UNIT=(SYSDA,SEP=SIMUT2),SPACE=(1016,(30,10))
//SIMUT4     DD      UNIT=(SYSDA,SEP=(SIMUT2,SIMUT3)),
//                   SPACE=(80,(100,50))
//SYSIN      DD      *

             SIMDIS-Programm
/*
```

Für liste steht die Parameterliste, in der folgende Parameter angegeben werden können:

A,B oder C	Auswahl einer Standardzuordnung für die Anzahl von Elementen eines jeden Typs. Damit verbunden ist direkt der benötigte Hauptspeicherplatz (90, 120 oder 180 K Bytes)
LIST	Auflistung des SIMDIS-Programmes
NOLIST	Unterdrückung der Auflistung des SIMDIS-Programmes
XREF	Ausgabe der Referenztabelle verwendeter symbolischer Namen
NOXREF	Unterdrückung der Auflistung der Referenztabelle
SIM	Ausführung der Simulation nach fehlerfreier Übersetzung. Voraussetzung ist, daß die SIMULATE-Anweisung im eingelesenen Kartendeck vorhanden ist.
NOSIM	Keine Ausführung der Simulation trotz fehlerfreier Übersetzung und vorhandener SIMULATE-Anweisung.

Die unterstrichenen Parameter werden bei Fehlen standardmäßig angenommen.

Beispiele für EXEC-Anweisungen:

```
// EXEC     PGM=VM#SIMØ,PARM=(A,NOXREF)
// EXEC     PGM=VM#SIMØ,PARM=(LIST,B,NOSIM)
// EXEC     PGM=VM#SIMØ
```

Genaueres über die Jobsteuerung ist in [11] (Anleitung für den Programmierer) zu finden.

4.2. Ein SIMDIS-Programm zur Nachbildung eines Fertigungsprozesses

Nachstehend wird ein Simulationsmodell für den Fertigungsprozeß dargestellt, auf den sich die Aufgabenstellung des Abschnittes 3.1. bezieht. Dieses Modell ermöglicht ein systematisches Experimentieren mit diesem Prozeß unter den a.a.O. formulierten Bedingungen, die zunächst kurz wiederholt werden.

Auf i Maschinen sollen j Werkstücke mit jeweils einer werkstückabhängigen technologischen Reihenfolge bearbeitet werden. Die Bearbeitungszeiten auf den einzelnen Maschinen je Werkstück sind vorgegeben. Die Verfügbarkeit einer oder mehrerer Maschinen kann zu Beginn der Simulation eingeschränkt sein. Ferner erfordern vertragliche Regelungen die Einhaltung von Lieferzeiten bzw. erlauben nur eine beschränkte Lieferzeitüberschreitung. Das Simulationsmodell muß in der Lage sein, die hier nochmals kurz dargestellten Zusammenhänge nachzubilden. Es muß ferner in der Lage sein, die Steuerung der Werkstücke nach vorgegebenen organisatorischen Gesichtspunkten zu simulieren. In diesem Falle soll das Modell die Bearbeitungsreihenfolge auf den Maschinen nach den Regeln FIFO, KOZ und SLACK berücksichtigen.

Ähnlich wie in 3.1. werden Matrizen definiert, die das jeweilige Problem mit entsprechenden Daten beschreiben. Mittels INITIAL-Anweisungen werden diese Matrizen mit den Problemdaten gefüllt. In diesem Beispiel handelt es sich um vier Maschinen, auf denen sechs Werkstückgruppen gemäß vorgegebener Reihenfolge bearbeitet werden sollen. Die Matrix VERF gibt die Zeitpunkte der Verfügbarkeit einer jeden Maschine an, und die Matrix FZEIT beinhaltet diejenigen Bearbeitungszeiten, die möglichst nicht überschritten werden sollen. Der Einfachheit halber soll zunächst die Arbeitsweise des Modells nach der Regel FIFO beschrieben werden. Diese Regel besagt, daß aus einer Gesamtheit wartender Aufträge (Werkstücke) stets derjenige einer freiwerdenden Maschine zugeführt wird, der am längsten auf Bearbeitung wartet bzw. der als erster in die Warteschlange eintrat.

Der GENERATE-Block erzeugt einen Aktivator mit 4 Byte und 2 Vollwortparametern. Entsprechend der Anzahl Werkstückarten werden mit dem SPLIT-Block weitere Aktivatoren (Kopien) erzeugt; Kopien und Original ergeben die Gesamtzahl. Im Byte-Parameter 4 werden diese Aktivatoren numeriert, so daß jede Werkstückart einer Nummer zugeordnet ist. Bei der Regel FIFO gelangt jeder Aktivator zum Block mit Namen WAHL. Für jeden Aktivator bzw. jede Werkstückart wird diejenige Maschine ausgewählt, auf der nach der vorgegebenen technologischen Reihen-

folge die Bearbeitung erfolgen soll, d. h. aus jeder Spalte der Matrix REIFO (Spaltennr. = Werkstückart) wird das jeweils kleinste Element ermittelt. Der Wert steht im Byte-Parameter 1, der Zeilenindex bzw. die Maschinennummer im Byte-Parameter 2 und der Spaltenindex im Byte-Parameter 3. Die Werte von PB3 und PB4 sind demnach gleich. Wichtig für das Modell ist PB2 mit der Maschinennummer. Der anschließende TEST-Block beendet die Bearbeitung einer Werkstückart, sobald keine weitere Maschine angelaufen werden muß. Der nachfolgende MSAVEVALUE-Block addiert zu dem niedrigsten Wert in der Spalte PB4 die Zeilenanzahl der Matrix REIFO dazu, damit bei der nächsten Maschinenauswahl auch die nächste Maschine in der technologischen Folge gefunden wird.
Mit dem ASSIGN-Block wird dem Vollwortparameter 1 die Bearbeitungszeit auf der ausgewählten Maschine zugewiessen. Ist diese Bearbeitungszeit Null, d. h. wird das Werkstück nicht auf dieser Maschine bearbeitet, so wird die nächste Maschine ausgewählt. Fehlt dieser TEST-Block, so ist ein Überspringen dieser Maschine nicht möglich, d. h. die Bearbeitung auf der nächsten Maschine erfolgt erst dann, wenn alle vor diesem Werkstück wartenden Werkstücke bearbeitet wurden.
Der Block mit dem Namen REGEL und der Operation LINK hat in diesem Modell eine große Bedeutung. Dieser Block reiht ankommende Aktivatoren in Benutzerketten ein. Hier sind die Benutzerketten als Warteschlangen vor den Maschinen zu verstehen. Der zweite Operand FIFO gibt die Regel der Einordnung in diese Warteschlange an. Der dritte Operand gibt die Adresse an, zu der der eingetretene Aktivator weitergehen soll, sofern der eingetretene Aktivator der erste ist.
Die Maschinen werden im Modell durch Einrichtungen dargestellt, d. h. mit den SIMDIS-Operationen SEIZE und RELEASE. Der Parameter PB2 gibt dabei die Maschinennr. an. Im GATE-Block wird der Zustand der zu belegenden Maschine getestet; falls die Maschine nicht belegt ist, kann sie vom Werkstück (Aktivator) belegt werden, im anderen Falle verbleibt es in der Warteschlange. Nach der Bearbeitungszeit, dargestellt durch den ADVANCE-Block, wird die Maschine frei und das näch-

ste Werkstück aus der Warteschlange kann die Maschine belegen (UNLINK entfernt den ersten Aktivator aus der Benutzerkette und schickt ihn zu der Maschine).
In den Matrizen BEGIN und ENDE wird für jede Werkstückart der Bearbeitungsbeginn und das Bearbeitungsende auf jeder Maschine registriert. Die letzte Zeile der Matrix ENDE enthält das Bearbeitungsende eines jeden Werkstückes.
Nachdem eine Werkstückart auf einer Maschine bearbeitet wurde, wird die nächste Maschine ausgewählt (TRANSFER ,WAHL).
Sind alle Werkstückarten bearbeitet worden, werden die Aktivatoren im ASSEMBLE-Block zu einem Aktivator vereinigt, der als Abschluß des Simulationsexperimentes in der Matrix FOLGE die Bearbeitungsreihenfolge der Werkstücke auf den einzelnen Maschinen ermittelt.
Die Verfügbarkeit von Maschinen ab vorgegebenen Zeitpunkten wird durch Aktivatoren realisiert, die infolge ihrer relativ hohen Priorität die Maschinen vor den Werkstücken für eine bestimmte Zeit belegen.
GENERATE ,,,1,10,4PB erzeugt einen Aktivator mit Priorität 10 und 4 Byte-Parametern. Dieser wird mittels SPLIT weiter aufgeteilt, so daß 4 Aktivatoren, sprich Maschinen, zur Verfügung stehen. Mittels SEIZE und RELEASE erfolgt die Nachbildung der Maschinen, deren Nummer im Byte-Parameter 2 steht. Im ADVANCE-Block erfolgt die Verzögerung gemäß Vorgabe in der Matrix VERF. Erst nach Ablauf dieser Verzögerung steht die Maschine zur Verfügung, und aus der Warteschlange kann eine Werkstückart die Maschine belegen.
Das Programm zur Nachbildung der FIFO-Regel kann einfacher gestaltet werden, denn der Aufbau von Warteschlangen nach der FIFO-Regel geschieht innerhalb von PS SIMDIS immer dann, wenn ein Aktivator einen Folgeblock nicht betreten kann. In diesem Falle veranlaßt SEIZE den Aufbau einer Warteschlange, falls die adressierte Einrichtung belegt ist. RELEASE veranlaßt bei Freiwerden einer belegten Einrichtung, so daß der erste Aktivator aus einer eventuell bestehenden Warteschlange diese Einrichtung wieder belegen kann. Das Programm erhält somit folgende vereinfachte Form:

```
        GENERATE    ,,,1,,4PB,2PF
        SPLIT       5,,4PB
        ADVANCE     MH#ANKU(1,PB4)
WAHL    MSELEC MIN  1PB,REIFO,1-ZB#REIFO,PB4,,,MB
        TEST  LE    PB1,ZB#REIFO,ENDE
        MSAVEVALUE  ...
        ASSIGN      ...
        TEST  G     ...
        SEIZE       PB2
        MSAVEVALUE  BEGIN,PB2,PB4,RC,MF
        ADVANCE     PF1
        RELEASE     PB2
        MSAVEVALUE  ENDE,...
        MSAVEVALUE  ENDE,...
        TRANSFER    ,WAHL
        :
        GENERATE    ,,,1,10,4PB
        SPLIT       3,,2PB
        SEIZE       PB2
        ADVANCE     MH#VERF(1,PB2)
        RELEASE     PB2
        TERMINATE
```

Die etwas ausführlichere Form des Modells zeigt sehr anschaulich, wie flexibel PS SIMDIS bei der Anwendung ist. Durch kleine Änderungen kann anstelle der FIFO-Regel eine Simulation nach den Regeln KOZ oder SLACK erfolgen. Zu diesem Zwecke werden bei den Regeln KOZ und SLACK ausgewählte Blöcke redefiniert bzw. überschrieben. Dadurch ändert sich die Blockfolge für jeden Aktivator.

Bei der Regel KOZ wird als erstes der Block mit Namen REGEL ersetzt durch

```
        LINK        PB2,1PF .
```

Hier fehlt der 3. Operand, d. h. in die Warteschlange werden alle Werkstückarten aufgenommen, die zu diesem Zeitpunkt die Maschine belegen wollen. Erst danach kann eine Auswahl nach der kürzesten Bearbeitungszeit erfolgen. Die Ordnung in der

Warteschlange wird durch die Bearbeitungszeiten (PF1) gebildet, die Werkstückart mit kürzester Bearbeitungszeit befindet sich am Anfang der Warteschlange. Im Modell kommt folgender Teil zur Wirkung:

```
        :
        TEST    G   PF1,0,WAHL
SPR3    CONTINUE
        ASSIGN      2,...
        TEST    E   CN*PB2,0,REGEL
        SPLIT       1,UNLI
REGEL   LINK        PB2,1PF
        :
```

Der CONTINUE-Block ist ein Block ohne Wirkung, er ersetzt den bisherigen TRANSFER-Block mit seinem unbedingten Sprung.
Der folgende ASSIGN-Block ist z. Z. noch uninteressant, beinhaltet jedoch die Zuweisung 'SLACK'-Zeit zu Vollwortparameter 2 und ist für die SLACK-Regel von Bedeutung. Falls sich in der Warteschlange vor der Maschine noch keine Werkstücke befinden (CN*PB2 gleich 0), wird eine Kopie von diesem Aktivator erzeugt. Durch SPLIT erzeugte Aktivatoren gelangen an das Ende der AEK. Die AEK beinhaltet alle Aktivatoren, die zu einem Zeitpunkt bewegt werden können, hier also alle Werkstücke, die von einer Maschine auf eine andere überwechseln. Dieser Wechsel muß erst vollzogen sein, ehe eines dieser Werkstücke eine Maschine belegt, denn sonst ist eine Auswahl nach KOZ nicht möglich. Sind also alle Werkstücke an den betreffenden Maschinen angekommen, so veranlaßt die für jede Maschine erzeugte Kopie über

```
        UNLI        UNLINK   PB2,BELEG,1
```

die richtige Auswahl eines Werkstückes; die Kopie wird anschließend vernichtet.

Die dritte Variante ist die Simulation nach der SLACK-Regel. Hier erfolgen weitere Modelländerungen und Ergänzungen. Die Einordnung in die Warteschlange erfolgt jetzt nach

```
        LINK    PB2,2PF
```

also nach der 'SLACK'-Zeit, nach derjenigen Zeit, die bis zum vorgegebenen Fertigungsende noch zur Verfügung steht. Die notwendige Bearbeitungszeit wird in der einen Zeile der Matrix SLACK zu Beginn berechnet. Die Auswahl des günstigsten Werkstückes geschieht erst, nachdem alle Werkstücke ihren anstehenden Wechsel vollzogen haben und sich in die Warteschlange eingeordnet haben.
Anders liegt hier der Fall, wenn ein Werkstück eine Maschine verläßt und das nächste Werkstück die Maschine belegen kann. Während bei der KOZ-Regel Änderungen der Reihenfolge in der Warteschlange während der Bearbeitung eines Werkstückes jeweils eine aktuelle und richtige Reihenfolge widerspiegeln, ist dies bei der SLACK-Regel nicht immer garantiert. Die SLACK-Zeit eines Werkstückes ist abhängig von dem Zeitpunkt, zu dem es sich in die Warteschlange einreiht, sie ist umso größer, je eher es sich einreihte. Wartezeiten bleiben dabei zunächst unberücksichtigt. Dieser Mangel wird dadurch korrigiert, daß bei Verlassen einer Maschine alle Aktivatoren (Werkstücke) aus der Warteschlange entfernt werden und für sie eine Korrektur bzw. Neuberechnung der SLACK-Zeit vorgenommen wird. Erst danach gelangt das nächste Werkstück zur Maschine. Realisiert wird dies durch die Blockfolge:

```
       UNLINK      PB2,KORR1,ALL
       PRIORITY    PR,BUFFER
       UNLINK      PB2,BELEG,1
         :
KORR1  ASSIGN      2,AV#SLACK,,PF
       LINK        PB2,2PF
```

Der PRIORITY-Block ordnet den augenblicklichen Aktivator ans Ende der AEK in der Prioritätsklasse Null ein und setzt die Bearbeitung mit den aus der Warteschlange zwecks Korrektur entfernten Aktivatoren fort. Erst nachdem wieder alle Aktivatoren sich in der Warteschlange befinden (jetzt eventuell mit einer anderen Ordnung) kann ein Werkstück aus ihr entnommen werden.

Wird jedem Werkstück eine Priorität in der Bearbeitungsfolge zugeordnet, dann erfolgt die Auswahl eines Werkstückes in den Fällen, in denen mehrere Werkstücke gleichzeitig zur Auswahl stehen, nicht nach FIFO, sondern nach der entsprechenden Priorität. Nach dem SPLIT-Block, der die Werkstücke erzeugt, müßte nur ein PRIORITY-Block eingefügt werden, z. B.

```
PRIORITY   MB#PRIOR(1,6)
```

und in der Matrix PRIOR stehen die gewünschten Prioritäten.
Zu beachten ist dabei, daß der die Maschinen erzeugende GENERATE-Block immer Aktivatoren mit noch größerer Priorität erzeugt, damit die Maschinen zu Beginn der Simulation auch gesperrt werden können. Soll das Modell einen Fertigungsprozeß mit stochastischen Bearbeitungszeiten nachbilden, so müssen in Abhängigkeit von den gewünschten Aussagen einige Änderungen vorgenommen werden. Im deterministischen Falle reicht eine (einmalige) Durchrechnung des Auftragsdurchlaufes aus. Im stochastischen Fall jedoch muß der Prozeß der Maschinenbelegung mehrmals durchgerechnet werden, um zu statistischen Aussagen über diesen Prozeß zu gelangen, da jeder einzelne Durchlauf der Auftragsgesamtheit nur eine Realisierung der in Frage kommenden Prozeßgrößen (wie Durchlaufzeit der Aufträge, Wartezeit der Maschinen usw.) liefert. Hierbei entsteht die Frage nach den unter diesen Umständen aussagefähigen Daten. Welche Daten müssen gesammelt werden? Wieviele Rechnungen sind notwendig? Ebenso müssen die Regeln KOZ und SLACK modifiziert werden; bei stochastischen Bedienzeiten ist es erst nach der Bedienung möglich, die benötigte Zeit festzustellen, während die Regeln KOZ und SLACK diese vor der Bedienung zwecks richtiger Auswahl benötigen.
Die Ergebnisinterpretation bei Verwendung von deterministischen oder stochastischen Bedienzeiten unterscheidet sich auch beträchtlich. Das Modell mit deterministischen Bedienzeiten gerechnet, liefert zu jeder Regel einen exakten Maschinenbelegungsplan, der ebenso wie die Bedienzeiten determiniert ist. Bei Verwendung stochastischer Bedienzeiten haben die Ergebnisse jedoch statistischen Charakter. Mehrmaliges Durchrechnen einer Variante ist nach folgenden Änderungen möglich:

```
1       TABLE       MC,20,1,20
:                   :
6       TABLE
        GENERATE    ,,,,,4PB,2PF    } für
        GATE    LR  1               } GENERATE    ,,,1,,4PB,2PF
        LOGIC   S   1               }
        :
        ASSIGN      1,MH#BZEIT(PB2,PB4),,PF
        ASSIGN      1,DX1(PF1),,PF
        :
ENDE    TABULATE    PB4
        ASSEMBLE    6
        :
        LOGIC   R   1
        TERMINATE   1
        :
        GENERATE    ,,,1,10,4PB
        SPLIT       3,,2PB
VERF    SEIZE       PB2
        :
SPR1    UNLINK
        GATE    FU  PB2         } für
        GATE    LR  1           } TERMINATE
        TRANSFER    ,NVERF      }
        :
        START       20
```

In diesem Beispiel wird eine Variante zwanzigmal durchgerechnet. Die Bedienzeiten sind exponential verteilt, die Mittelwerte entsprechen den bisherigen deterministischen Bedienzeiten. In den Tabellen 1 bis 6 werden die Durchlaufzeiten pro Werkstück erfaßt.

Kontrollfragen

1. Untersuchen Sie die Auswirkungen unterschiedlicher Terminvorgaben auf die Ergebnisse bei Anwendung der SLACK-Regel. Verwenden Sie als neue Terminvorgabe die bisher errechneten Fertigstellungstermine nach den Regeln FIFO, KOZ und SLACK sowie die Vektoren $\underline{a}$ (Ankunftszeitpunkte der Werkstücke) und $\underline{b}$ mit $b_j = \sum_{1}^{I} t_{ij} + a_j$. Welche Bedeutung haben die Vektoren $\underline{a}$ und $\underline{b}$ als Terminvorgaben bei Anwendung der SLACK-Regel bzw. nach welchen Kriterien erfolgt in diesen Fällen die Werkstückauswahl?

2. Welche der angegebenen Regeln sind in der Praxis einfach anwendbar? Unterscheiden Sie dabei zwischen deterministischen und stochastischen Bedienzeiten.

4.3. Ein SIMDIS-Programm zur Nachbildung eines Instandhaltungsprozesses

An dieser Stelle soll für die im Abschnitt 3.2. gegebene Aufgabenstellung ein Simulationsmodell dargestellt werden. Ebenso wie das vorangehende Modell erlaubt auch dieses Modell ein systematisches Experimentieren mit diesem Prozeß.
Die im Abschnitt 3.2. festgelegten Regeln seinen hier nochmals kurz wiederholt.

1. Die Arbeiter gehören einer Reparaturbrigade an, wobei jeder Arbeiter gleiche Fähigkeiten und Fertigkeiten besitzt.

2. Wegen der Verschiedenartigkeit der Maschinen sind diese in Gruppen eingeteilt. Die Laufzeiten und Reparaturzeiten der Maschinen einer Gruppe sind jeweils identisch verteilt.

3. Zur Reparatur einer Maschine ist nur eine Arbeitskraft notwendig, die diese ohne Unterbrechung ausführt.

4. Auftretende Wegezeiten eines Arbeiters zu den schadhaften Maschinen werden nicht besonders berücksichtigt.

5. Prioritäten in der Reihenfolge der Instandsetzung bei

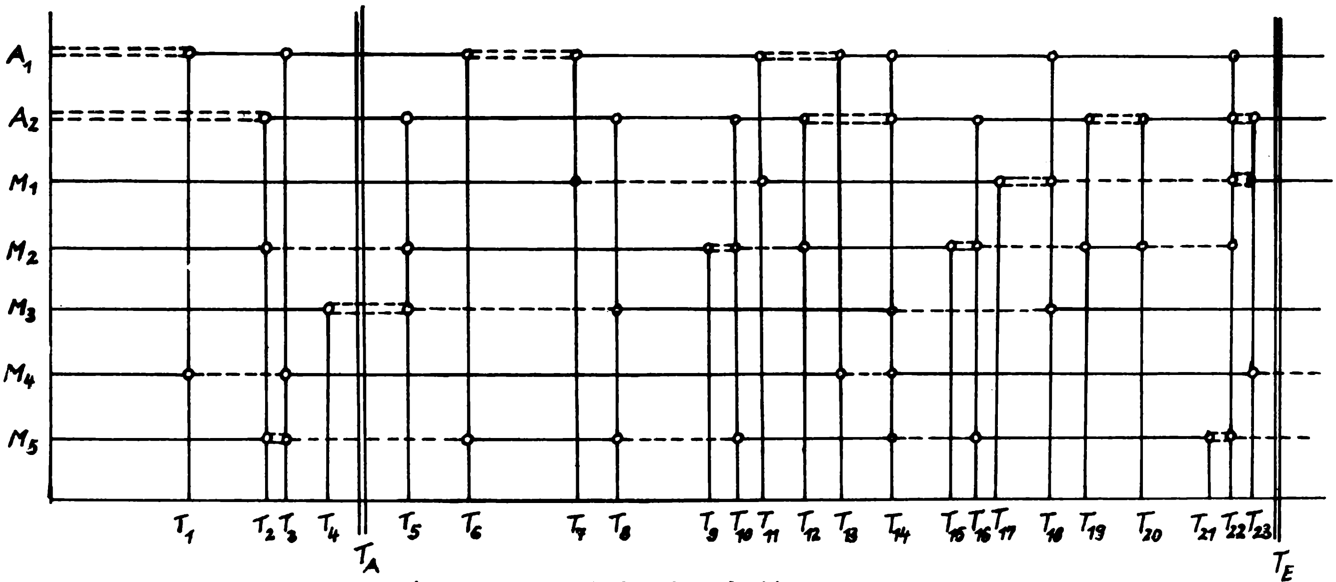
gleichzeitigem Stillstand mehrerer Maschinen werden nicht berücksichtigt.

Bevor auf die Modellierung des Prozesses mit SIMDIS-Sprachelementen eingegangen wird, soll anhand einer Grafik (Bild 3) die Verknüpfung der Teilprozesse verdeutlicht werden, die dem beschriebenen Instandhaltungsprozeß immanent sind. Zu diesem Zweck sei angenommen, daß 2 Arbeiter 5 Maschinen einer Gruppe betreuen. Die Maschinen sollen anfangs alle in Betrieb sein.

Betrachtet man die Zustände, die eine Maschine annehmen kann, so stellt man fest, daß eine Maschine in Betrieb sein kann, auf eine Reparatur warten kann oder sich im Zustand der Reparatur befindet. Die Prozeßfolge "Laufzeit" - "Wartezeit auf Reparatur" - "Reparatur" wird als Elementarzyklus einer Maschine bezeichnet. Das entsprechende Simulationsmodell muß also in der Lage sein, diese Elementarzyklen und parallel verlaufende Vorgänge nachzubilden.
Der im Bild dargestellte Sachverhalt wird mittels SIMDIS-Sprachelementen auf folgende Art dargestellt:

```
LAUF   INITIAL     SK#ARB,2        ANZAHL ARBEITER
       GENERATE    ,,,1,,1PB
       SPLIT       4,,1PB          ERZEUGEN 5 MASCHINEN
       SEIZE       PB1             ERFASSEN LAUFZEIT/MASCHINE
       ADVANCE     DX1(...)        MASCHINENLAUFZEIT
       RELEASE     PB1
       QUEUE       WARTE           ERFASSEN WARTEZEIT
       ENTER       ARB             ARBEITER REPARIERT
       DEPART      WARTE
       ADVANCE     DU1(...,...)    REPARATURZEIT
       LEAVE       ARB             ARBEITER IST FREI
       TRANSFER    ,LAUF           ENDE ELEMENTARZYKLUS
       GENERATE    100             NACH JEWEILS 100 ZEITEINHEITEN
       TERMINATE   1               SIMULATIONSLAUF BEENDEN
       START       5               ZAHL DER SIMULATIONSLAEUFE
```

In diesem Modell werden die Maschinen durch die Aktivatoren dargestellt. In den beiden ADVANCE-Blöcken werden die Lauf- und Reparaturzeiten nachgebildet, die Maschinenlaufzeiten



Maschinenlaufzeit/Reparaturzeit durch Arbeiter

Maschinenstillstandszeit wegen Reparatur

Maschinenwartezeit auf Reparatur / Arbeiter wartet auf Reparatur

Bild 3: Lauf-, Stillstands- und Wartezeitverhalten der Maschinen seien hierbei <u>exponentiell-verteilt</u> und die Reparaturzeiten seinen gleichverteilt. Durch die Blöcke SEIZE und RELEASE wird eine Statistik über die Maschinenlaufzeit pro Maschine geführt. Mit ENTER und LEAVE wird das vorhandene Arbeitskräftepotential nachgebildet. Die Anzahl verfügbarer Arbeiter wird mittels INITIAL festgelegt. Jeder Eintritt eines Aktivators in den ENTER-Block verringert die verfügbare freie Kapazität des Speichers.

Ein Aktivator kann dabei nur dann den ENTER-Block betreten, wenn der Speicher ARB noch nicht voll ist. Das heißt, solange noch Arbeiter zur Verfügung stehen, kann die Maschine repariert werden. Im anderen Falle tritt eine Wartezeit auf. Die Statistik der auftretenden Warteschlange wird durch die Blöcke QUEUE und DEPART geführt.

Dieses Modell erfüllt alle zu Beginn festgelegten Bedingungen. Der folgerichtige Ablauf für die parallel verlaufenden Vorgänge ist garantiert, da sich in den ADVANCE-Blöcken mehrere Aktivitäten zur gleichen Zeit aufhalten können. Dieses Modell ist die Grundlage für das im Anhang gezeigte erweiterte Modell, das um einige Definitionsanweisungen und Blöcke zur statistischen Datenerfassung und -auswertung ergänzt wurde. Durch Austausch der EQU- und INITIAL-Anweisungen kann die Anzahl der Gruppen und die der Maschinen verändert werden. Bei gleicher oder geänderter Gruppeneinteilung kann die Anzahl der Arbeitskräfte verändert werden, um so eine den Erfordernissen gerecht werdende Variante finden zu können.
Der Abbau der Warteschlange erfolgt nach der Regel FIFO, d. h. alle Maschinen werden als gleichbedeutend für den Produktionsprozeß angesehen. Durch Vergabe von Prioritäten entsprechend der Bedeutung der Maschine (hohe Priorität entspricht großer Bedeutung) erfolgt die Reparaturausführung nach der Prioritätsregel. Dazu muß nur ein Block eingefügt werden, der einer jeden Maschinengruppe eine Priorität zuordnet, z. B.

```
        GENERATE    ,,,1,,3PB,1PH
STRUK   MACRO
        PRIORITY    FN#PRIOR
PRIOR   FUNCTION    PB1,L5
,20/,0/,0/,10/,0
```

Durch diese Maßnahme wird man jedoch keine Änderungen in der Auslastung der Arbeitskräfte erreichen. Lediglich eine Änderung der Verhältnisse von Maschinenlaufzeit, Wartezeit und eigentlicher Reparaturzeit innerhalb einer jeden Gruppe wird erreicht, aber gerade dies kann von Bedeutung sein bei der Wahl einer optimalen Strategie zur bestmöglichen Grundfondsauslastung.
Zu Beginn der Simulation mit diesem Modell existiert ein Zustand, der sich nachteilig auf die erhaltenen Daten auswirken kann, zu Beginn laufen alle Maschinen und alle Arbeiter warten auf eine Reparatur. Bei einem sehr kurzen Simulationszeitraum könnte es passieren, daß keine Reparatur anfällt. Also muß der Simulationszeitraum für ein Experiment ausreichend groß sein. Um aber den Einfluß des Anfangszustandes auf die Simulationsergebnisse auszuschließen bzw. zu verringern, muß angestrebt werden, vor Beginn der eigentlichen Simulation einen zufälligen Zustand zu erreichen. Dieser Zustand ist dadurch charakterisiert, daß eine Anzahl der Maschinen läuft, wartet oder repariert wird. Ein solcher Zustand wird in einer sogenannten Einschwingphase mittels eines Pseudosimulationslaufes erreicht. Im Modell wird dies durch die Anweisungsfolge

```
        START       1,NR
        RESET
```

verwirklicht. Die RESET-Anweisung hat zur Folge, daß alle bisherigen Statistiken gelöscht werden und nur der Modellzustand erhalten bleibt. Durch Erhöhen der verfügbaren Arbeitskräfte können auf diese Art und Weise mehrere Varianten hintereinander gerechnet werden. Mit wieviel Arbeitskräften beginnt man aber zu experimentieren? Diese Frage läßt sich leicht durch eine grobe Schätzung beantworten. Bildet man den Mittelwert über alle Laufzeiten und Reparaturzeiten aller Maschinen, so ergibt sich in diesem Falle, daß eine Maschine durchschnittlich 6,9 Tage läuft und 3,1 Tage repariert wird, dann wieder 6,9 Tage läuft usw. Unter Vernachlässigung der Wartezeit auf Reparatur ergibt sich eine Zykluszeit von 10 Tagen. In diesen 10 Tagen kann eine Arbeitskraft rund 3,2 Maschinen reparieren, für 62 Maschinen sind also ungefähr 19 Arbeitskräfte notwendig. Die Simulationsergebnisse bestätigen diese Überschlagsrechnung. Als Modellzeiteinheit wurde ein Zehntel Tag gewählt. Bei Verwendung der Zeiteinheit Tag wird mit dem vorliegenden Datenmaterial die Exponentialverteilung zu ungenau nachgebildet. Jedes Experiment wurde über 100 Tage durchgeführt. Die Ergebnisse sind in folgender Tabelle zusammengestellt.

Anzahl Arbeitskräfte	durchschn. sind beschäftigt	wartende Maschinen		Laufleistung in %	ausgeführte Reparaturen
		durchschn.	maximal		
17	16,89	7,84	20	60,1	644
18	17,64	5,02	20	63,4	669
19	18,41	4,31	18	63,3	676
20	18,75	3,34	19	64,3	689
21	20,22	2,77	16	62,9	765
22	19,70	0,79	8	66,9	773
23	19,69	0,54	7	67,3	746

In diesem Modell ist der anfangs beschriebene Elementarzyklus nur einmal enthalten. Durch Zuordnung von Werten zu den Parametern wurde die Zuordnung zu den statischen Elementen wie Einrichtungen, Warteschlangen usw. getroffen. Ebenso wäre es möglich gewesen, ein Programm zu schreiben, in dem für jede Maschine oder Gruppe der Elementarzyklus aufgeschrieben wird. In diesem Falle wären es 62 Elementarzyklen, falls für jede Maschine einer verwendet wird, bzw. 5 Elementarzyklen, falls die Maschinen zu Gruppen zusammengefaßt werden. Die Adressen bzw. Nummern der Elemente eines jeden Elementetyps müssen aber in jedem Falle eindeutig vergeben werden. So ergibt sich bei vielen Simulationsmodellen, die mit SIMDIS-Sprachelementen aufgestellt werden, neben der Modellierung des Prozesses das Problem der Zuordnung bzw. Adressierung der Elemente. Im vorliegenden Falle kann durch Änderung weniger INITIAL-Anweisungen ein ähnliches Problem gelöst werden, bei dem jedoch eine andere Maschinenzahl und Gruppierung vorliegt.

Kontrollfragen

1. Entwickeln Sie für den Instandhaltungsprozeß eine Kostenfunktion. Bestimmen Sie anhand dieser Kostenfunktion und mit den Daten aus den Simulationsexperimenten die kostengünstigste Variante sowohl in bezug auf eine Maschinengruppe als auch in bezug auf den Gesamtprozeß. Setzen Sie dabei voraus, daß die anfallenden Kosten innerhalb einer Maschinengruppe gleich, jedoch von Maschinengruppe zu Maschinengruppe unterschiedlich sind.
2. Verändern Sie bei vorgegebenen Kostenfaktoren die Prioritäten bei der Reparaturausführung mit dem Ziel, eine noch kostengünstigere Variante zu finden. Dies kann geschehen durch Vergabe einer höheren Priorität für Maschinen mit geringen Laufzeiten, hohen Reparaturzeiten usw.

4.4. Durchführung von Experimenten mit SIMDIS-Modellen

Ein Simulationsmodell besteht aus der logischen Beschreibung des Prozesses und den zugehörigen Daten. Das Experimentieren mit dem Modell kann sich auf das Durchrechnen mit jeweils anderen Daten beschränken (vgl. Instandhaltungsprozeß), kann Änderungen im logischen Ablauf vorsehen (vgl. Fertigungsprozeß) und kann aus einer Mischung von beiden bestehen. Ein Experiment wird in jedem Falle mit der SIMDIS-Anweisung START eingeleitet. Alle bis zu dieser Anweisung getroffenen Zuordnungen, seien es Daten oder Block- und Definitionsanweisungen, haben für dieses Experiment ihre Gültigkeit. Erst nachdem der Startzähler (Operand A in START-Anweisung) zu Null geworden ist, ist diese START-Anweisung abgearbeitet und eventuell folgende Anweisungen können ein neues Experiment vorbereiten und mittels weiterer START-Anweisung zur Ausführung bringen. Beim Modell des Instandhaltungsprozesses sieht das folgendermaßen aus:

```
        :             SIMDIS-Modell mit Daten
        START         1,NR   Einschwingphase, kein Ergebnisdruck
        INITIAL       XB#ANF,2/XB#EXPMT,7   Datenänderung
        RESET            bisherige Statistik löschen, Zustand
                         bleibt erhalten
PRINT   PRINT         100,200,REP   Redefinition des Blockes mit
                                    Adr. PRINT
        START         5,NR   1. Experiment mit 17 Arbeitern
        INITIAL       SK#ARB,18     Anzahl Arbeitskräfte erhöhen
                                    auf 18
        RESET            bisherige Statistik löschen, Zustand
                         bleibt erhalten
        START         5,NR   2. Experiment mit 18 Arbeitern
        :
        END
```

Im Modell des Fertigungsprozesses wurden nach jedem Experiment alle bisher gesammelten Statistiken und Daten mittels CLEAR-Anweisung gelöscht. Nachfolgende INITIAL-Anweisungen setzen wieder die Eingangsdaten. Mit den angegebenen Blockredefinitionen wurde die Programmlogik der jeweils nachzubildenden Bearbeitungsregel angepaßt. Die Redefinition von Blockanweisungen ist über symbolische Namen durchzuführen. Dabei wird der bisher gültige Block vollständig mit den neuen Angaben überschrieben. Ein GENERATE-Block kann dabei nur durch einen anderen GENERATE-Block überschrieben werden, die Redefinition muß bei Verwendung von CLEAR vor dieser Anweisung durchgeführt werden. Ebenso müssen alle in der GENERATE-Anweisung benötigten Werte gesetzt sein.

Beispiel:

```
        INITIAL     XH10,200
GEN     GENERATE    DX1(XH10)
        CLEAR       XH10      löschen aller Statistiken, löschen
                              Zustand, erhalten XH10
        START       100
```

Literaturverzeichnis

[1] Krampe, H., Kubat, J., Runge, W. — Bedienungsmodelle, Ein Leitfaden für die praktische Anwendung, Verlag Die Wirtschaft, Berlin 1973

[2] Tietböhl, G., Runge, W. — Bedienungsmodelle, 2. Lehrbrief der Lehrbriefreihe für das Hochschulfernstudium MATHEMATISCHE METHODEN UND MODELLE, Verlag Technik, Berlin 1976

[3] Göttner, R., Kurzhals, S., Olbrich, M. — Lagerhaltungsmodelle, 4. Lehrbrief der Lehrbriefreihe für das Hochschulfernstudium MATHEMATISCHE METHODEN UND MODELLE, Dresden 1976

[4] Autorenkollektiv — Simulation diskreter Prozesse - Grundlagen, Systemunterlagen, Anwendungen, Zeitschrift Rechentechnik/Datenverarbeitung, Beiheft 3/1978

[5] Autorenkollektiv — Simulationsmodelle für ökonomisch-organisatorische Probleme, Schriftenreihe Datenverarbeitung, Verlag Die Wirtschaft, Berlin 1974

[6] Autorenkollektiv — Simulationsmodelle in der EDV-Projektierung, Schriftenreihe Informationsverarbeitung, Verlag Die Wirtschaft, Berlin 1974

[7] Buslenko, N. P. — Simulation von Produktionsprozessen, B. G. Teubner-Verlagsgesellschaft, Leipzig 1971

[8] Frank, M., Lorenz, P. — Simulation diskreter Prozesse - Eine Einführung für den Anwender, Fachbuchverlag, Leipzig 1979

[9] Piehler, J., Zschiesche, H.-U. — Simulationsmethoden, Reihe MINÖL, Bd. 20, B. G. Teubner-Verlagsgesellschaft, Leipzig 1976

[10] Schreiter, D. — Simulationsmodelle für ökonomisch-organisatorische Probleme, Schriftenreihe Datenverarbeitung, Verlag Die Wirtschaft, Berlin 1968

[11] Anwenderdokumentation des VEB Robotron zum PS SIMDIS

- Anwendungsbeschreibung — H 4313 - 2002
- Anleitung für den Systemverantwortlichen — H 4313 - 2003
- Anleitung für den Programmierer — H 4313 - 2004
- Sprachbeschreibung — H 4313 - 2007
- Nachrichten — H 4313 - 2006

Anhang

Teil 1 Übersicht der Blockanweisungen

Die Eintragungen in der dritten Spalte bedeuten:
X: Block kann einem Aktivator den Eintritt verweigern
Z: Block kann einen Aktivator bis zu einem späteren Zeitpunkt aufhalten

Operation	Wirkungsweise		mnemon. Operator	Operanden (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung) A	B	C	D	E	F	G
ADVANCE	Festlegen eines neuen Ereigniszeitpunktes	Z		st Mittelwert oder st Faktor (0)	st maximale Abweichung oder FN_j Funktionswert (0)	st obere Zeitgrenze für Blockabgangszeit (keine Begrenzung)				

Operation	Wirkungsweise		mnemon. Operator	Operanden (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung) A	B	C	D	E	F	G
ASSIGN	Änderung von Parameterwerten			st[-st] [+/-] Nummer eines Parameters bzw. untere und obere Grenze einer Parameterfolge: +: addieren -: subtrah. ohne Zeich.: ersetzen (Fehler)	st,name Wert für die Änderung (Fehler)	st,name Nummer einer Funktion zur Modifikation des Wertes im Operanden B (ohne Wirkung)	Px(x=F,H,B,L) Parametertyp (erster definierter Typ in der Reihenfolge H,F,B,L)			

Fortsetzung zu Teil 1 Übersicht der Blockanweisungen

Operation	Wirkungsweise		mnemon. Operator	Operanden (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung) A	B	C	D	E
GENERATE	Erzeugung eines Aktivators			st* Mittelwert oder st* Faktor (0)	st* maximale Abweichung oder FN_j Funktionswert (0)	st* Blockabgangszeit für ersten Aktivator (Angaben in A und B sind bestimmend)	st* maximale Anzahl der zu erzeugenden Aktivatoren (keine Beschränkung)	st* Priorität (0)

Operanden ... F	G
bis zu 4 Operanden st*Px (x = F, H, B, L) Anzahl und Typ der Parameter (8PH)	

Operation	Wirkungsweise		mnemon. Operator	Operanden (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung) A	B	C	D	E	F	G
LINK	Einordnen in eine Benutzerkette	Z		st,name Nummer oder Name der Benutzerkette (Fehler)	FIFO,LIFO, PR,stPx(x=F,H,B,L) Vorschrift für das Einordnen der Aktivatoren (Fehler)	st,name Blockadresse für Verzweigung, wenn LINK-Bit AUS (unbedingte Einordnung)				

Fortsetzung zu Teil 1 Übersicht der Blockanweisungen

Operation	Wirkungsweise		mnemon. Operator	Operanden (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung) A	B	C	D
MSAVEVALUE	Änderung des Wertes von Matrix- elementen			st[-st][±], name [-name][±] Nummer oder Name der Matrix bzw. untere und obere Grenze einer Folge von Matrizen +: addieren -: subtra- hieren ohne Zeichen: ersetzen (Fehler)	st[-st] untere [und obere] Grenze des Zeilenbe- reichs (Fehler)	st[-st] untere [und obere] Grenze des Spalten- bereichs (Fehler)	st Wert für Änderung (Fehler)

Operanden ... E	F	G
Mx (x=F,H,B,L) Typ der Matrix (MF)		

Fortsetzung Teil 1 Übersicht der Blockanweisungen

Operation	Wirkungsweise		mnemon. Operator	Operanden (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung) A	B	C	D	E
MSELEC	Bestimmen des ersten Elementes einer Gesamt(Unter)-Matrix, das einer vorgegebenen Bedingung genügt bzw. einen Extremwert annimmt		G,GE,L, LE,E,NE MAX,MIN	stPx(x=F,H,B) Parameternummer mit Typangabe (Fehler)	st,name Nummer oder Name der Matrix (Fehler)	st[-st] untere [und obere] Grenze des Zeilenbereichs (gesamter Zeilenbereich)	st[-st] untere [und obere] Grenze des Spaltenbereichs (gesamter Spaltenbereich)	st Vergleichswert (Fehler)

Operanden ... F	G
st,name Blockadresse für Verzweigung, falls kein Element Bedingung erfüllt (Folgeblock)	Mx (x=F,H,B,L) Typ der Matrix (MF)

Fortsetzung Teil 1 Übersicht der Blockanweisungen

Operation	Wirkungsweise		mnemon. Operator	Operanden (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung) A	B	C	D	E	F	G
SPLIT	Erzeugung von Kopien eines Aktivators			st Anzahl der Kopien (Fehler)	st,name Blockadresse für die Kopien (Kopien und Original zum Folgeblock)	stPx(x=F,H,B) Parameternummer für Numerierung (keine Numerierung)	bis zu 4 Operanden stPx (x=F,H,B,L) Anzahl und Typ der Parameter für die Kopien (wie Original)			
TEST	Test einer arithmetischen Bedingung	X	G,GE, L,LE, E,NE	st erster Wert für den Vergleich (Fehler)	st zweiter Wert für den Vergleich (Fehler)	st,name Blockadresse für Verzweigung, falls Bedingung nicht erfüllt (Aktivator wartet)				

Teil 2 Übersicht der Definitionsanweisungen

Operationsfeld	Bedeutung	Namensfeld	Operandenfeld			Bemerkungen
FUNCTION	Festlegung des Typs und des Arguments einer Funktion	k,name Nummer oder Name der Funktion	st* Argument der Funktion	Ck,Dk,Lk,Ek,Mk,Sk Typ der Funktion und Anzahl k der Wertepaare	liste Liste von Symbolen der Elementetypen für Funktionen vom Typ S	Funktionsfolgekarten notwendig
INITIAL	Zuweisung des Zustandes EIN bzw. AUS für Schalter	-	LSi[-LSj] bzw. LRi[-LRj] z. B. INITIAL LR5/LS7-LS10		i,j Nummer oder Name des Schalters	Zuweisung erfolgt während der 2. Übersetzungsstufe
	Zuweisung von Werten für Skalare	-	Xxi[-Xxj],st* z. B. INITIAL X1-X3,5/XH7,10 INITIAL XL3,5.23E-3		i,j Nummer oder Name des Skalars st* einzutragender Wert x Typ des Skalars (x=F,H,B,L)	Zuweisung für verschiedene Elementetypen mit einer INITIAL-Anweisung ist möglich

Fortsetzung Teil 2 Übersicht der Definitionsanweisungen

Operations-feld	Bedeutung	Namensfeld	Operandenfeld		Bemerkungen
INITIAL	Zuweisung von Werten für Matrixelemente bzw. für sämtl. Elemente von Gesamt (Unter)-Matrizen		Mxi[-Mxj] [(zu-zo,su-so)],st* oder Mxj[(zu-zo,su-so)],liste z. B. INITIAL MH3,7 INITIAL MF7(1-4,3-4),((3)(5,6),*,2)	x Typ der Matrix (x=F,H,B,L) i,j Nummer oder Name der Matrix zu,zo,su,so Begrenzung einer Untermatrix	
	Zuweisung von Anfangsmultiplikatoren für Zufallszahlengeneratoren		RNi [-RNj],st* z. B. INITIAL RN1-RN3,37/RN5,49	i,j Nummer des Zufallszahlengenerators st* Anfangsmultiplikator	
	Festlegen der Kapazität eines Speichers		SKi [-SKj],st*	i,j Nummer oder Name des Speichers st* Kapazität	Standardkapazität beträgt $2^{31}-1$ Einheiten

Teil 3 Übersicht der Steueranweisungen

Operationsfeld	Wirkung	Operandenfeld (Klammerangabe: Maßnahme des PS SIMDIS bei fehlender Eintragung)	Bemerkungen
CLEAR	Löschen des Zustandes des Simulationsmodells. Im Operandenfeld eingetragene Skalare oder Matrizen bleiben erhalten.	Xxi [-Xxj] i,j Nummer des Skalars bzw. der Matrix Mxi [-Mxj] [(zu-zo,su-so)] zu,zo,su,so Begrenzung der Untermatrix z. B. CLEAR XB1-XB7,ML3(3-5,4-7),MF5 x = F,H,B,L	ZEK wird neu aufgebaut
RESET	Löschen statistischer Werte. Bei den im Operandenfeld eingetragenen Elementen bleiben die statistischen Werte erhalten.	symboli[-symbolj] symbol Symbol eines Elementetyps i,j Nummer des Elementes z. B. RESET F1,T1-T4,Q7-Q8,Q11	Zu beachten sind die für diese Anweisung zulässigen Symbole

Übersicht der numerischen Standardsymbole (NSS)

Elementeklasse	Elemententyp	Standardsymbol	Bedeutung	Wertebereich
1	2	3	4	5
Operationselement	Block	BNj	Anzahl der bisher in den Block j eingetretenen Aktivatoren	$[0,2^{24}-1]$
		BWj	Anzahl der sich augenblicklich im Block j befindenden Aktivatoren	$[0,2^{15}-1]$
Dynamisches Element	Aktivator	PFj	Wert des Parameters des augenblicklich bewegten Aktivators vom Typ Vollwort	$[-2^{31}+1,2^{31}-1]$
		PHj	Halbwort	$[-2^{15},2^{15}-1]$
		PBj	Byte	$[-2^{7},2^{7}-1]$
		PLj	Gleitkomma	$\pm(5{,}4*10^{-79}\ldots7{,}2*10^{75})$
		PR	Prioritätsklasse, der der augenblicklich bewegte Aktivator angehört	$[0,127]$
		MC	Laufzeit des augenblicklich bewegten Aktivators	$[0,2^{31}-1]$
		MPjPF	Laufzeit des augenblicklich bewegten Aktivators bez. einer Eintragung im Parameter j v.Typ Vollwort	$[0,2^{31}-1]$
		MPjPH	Halbwort	$[0,2^{15}-1]$
		BG	Nummer des Blockes, an dem der Aktivator erzeugt wurde	$[0,2^{15}-1]$
Statisches Element	Einrichtung	FSj	Zustand der Einrichtung j	0 oder 1
		FLj	mittlere Auslastung der Einrichtung j	$[0,1]$
		FCj	Anzahl der bisher die Einrichtung j benutzenden Aktivatoren	$[0,2^{31}-1]$

Fortsetzung Übersicht der numerischen Standardsymbole (NSS)

1	2	3	4	5
		FTj	mittlere Verweilzeit eines Aktivators in der Einrichtung j	$[0,2^{31}-1]$
	Speicher	SKj	Kapazität des Speichers j	$[0,2^{31}-1]$
		SNj	augenblicklicher Inhalt des Speichers j (Anzahl der augenblicklich im Speicher befindenden Einheiten) 1)	$[0,2^{31}-1]$
		SAj	mittlerer Inhalt des Speichers j (Anzahl der sich im Speicher befindenden Einheiten je Zeiteinheit) 1)	$[0,2^{31}-1]$
		SMj	maximaler Inhalt des Speichers j (maximale Anzahl der sich bisher gleichzeitig im Speicher befindenden Einheiten) 1)	$[0,2^{31}-1]$
		SPj	augenblicklich freie Kapazität im Speicher j 1)	$[0,2^{31}-1]$
		SLj SWj SXj	mittlere Auslastung des Speichers j während der { Gesamtzeit, verfügbaren Zeit, nichtverfügbaren Zeit }	$[0,1]$
		SCj	Anzahl der in den Speicher j eingetretenen Einheiten 1)	$[0,2^{31}-1]$
		STj	mittlere Verweilzeit einer Einheit im Speicher j 1)	$[0,2^{31}-1]$
	Warteschlange	QNj	augenblickliche Länge der Wartschlange j (Anzahl der sich augenblicklich in der Warteschlange j befindenden Einheiten) 1)	$[0,2^{31}-1]$
		QAj	mittlere Länge der Warteschlange j (Anzahl d. sich in der Warteschlange j befindenden Einheiten je Zeiteinheit) 1)	$[0,2^{31}-1]$
		QMj	maximale Länge der Warteschlange j (maximale Anzahl der sich bisher gleichzeitig in der Wartschlange j befindenden Einheiten) 1)	$[0,2^{31}-1]$

Fortsetzung Obersicht der numerischen Standardsymbole (NSS)

1	2	3	4	5
		QCj	Anzahl der insgesamt in die Warteschlange eingetretenen Einheiten 1)	$[0,2^{31}-1]$
		QZj	Anzahl der in die Warteschlange j eingetretenen Einheiten mit Wartezeit gleich Null 1)	$[0,2^{31}-1]$
		QTj	mittlere Wartezeit einer Einheit in der Warteschlange j 1)	$[0,2^{31}-1]$
		QXj	mittlere Wartezeit einer Einheit in der Warteschlange j (ausschließlich Einheiten mit Wartezeit gleich Null) 1)	$[0,2^{31}-1]$
	logischer Schalter	LSj	Zustand des logischen Schalters j	0 oder 1
	Benutzerkette	CNj	augenblicklicher Inhalt der Benutzerkette j (Anzahl der sich augenblicklich in der Benutzerkette j befindenden Aktivatoren)	$[0,2^{15}-1]$
		CAj	mittlerer Inhalt der Benutzerkette j (Anzahl der sich in der Benutzerkette j befindenden Aktivatoren je Zeiteinheit)	$[0,2^{15}-1]$
		CMj	maximaler Inhalt der Benutzerkette j (maximale Anzahl der sich bisher gleichzeitig in der Benutzerkette j befindenden Aktivatoren)	$[0,2^{15}-1]$
		CCj	Anzahl der in die Benutzerkette j eingetretenen Aktivatoren	$[0,2^{31}-1]$
		CTj	mittlere Verweilzeit eines Aktivators in der Benutzerkette j	$[0,2^{31}-1]$

Fortsetzung Obersicht der numerischen Standardsymbole (NSS)

1	2	3	4	5
	Gruppe	GNj	Anzahl der sich augenblicklich in der Gruppe befindenden Mitglieder	$[0,2^{15}-1]$
	Skalar	XFj	Wert des Skalars j vom Typ Vollwort	$[-2^{31}+1,2^{31}-1]$
		XHj	Wert des Skalars j vom Typ Halbwort	$[-2^{15},2^{15}-1]$
		XBj	Wert des Skalars j vom Typ Byte	$[-2^{7},2^{7}-1]$
		XLj	Wert des Skalars j vom Typ Gleitkomma	$\pm(5{,}4*10^{-79}...7{,}2*10^{75})$
		SQj(x)	Wurzelwert von x	$[0,2^{24}-1]$
		TG	Stand des Startzählers	$[-2^{31},2^{31}-1]$
	Matrix	MFj(k,l)	Element (k,l) der Matrix j vom Typ Vollwort	$[-2^{31}+1,2^{31}-1]$
		MHj(k,l)	Element (k,l) der Matrix j vom Typ Halbwort	$[-2^{15},2^{15}-1]$
		MBj(k,l)	Element (k,l) der Matrix j vom Typ Byte	$[-2^{7},2^{7}-1]$
		MLj(k,l)	Element (k,l) der Matrix j vom Typ Gleitkomma	$\pm(5{,}4*10^{-79}...7{,}2*10^{75})$
		YFj	Spaltenanzahl der Matrix j vom Typ Vollwort	$[0,2^{15}-1]$
		YHj	Spaltenanzahl der Matrix j vom Typ Halbwort	$[0,2^{15}-1]$
		YBj	Spaltenanzahl der Matrix j vom Typ Byte	$[0,2^{15}-1]$
		YLj	Spaltenanzahl der Matrix j vom Typ Gleitkomma	$[0,2^{15}-1]$
		ZFj	Zeilenanzahl der Matrix j vom Typ Vollwort	$[0,2^{15}-1]$
		ZHj	Zeilenanzahl der Matrix j vom Typ Halbwort	$[0,2^{15}-1]$
		ZBj	Zeilenanzahl der Matrix j vom Typ Byte	$[0,2^{15}-1]$
		ZLj	Zeilenanzahl der Matrix j vom Typ Gleitkomma	$[0,2^{15}-1]$

Fortsetzung Übersicht der numerischen Standardsymbole (NSS)

1	2	3	4	5
	Tabelle	TBj	Mittelwert der in der Tabelle j eingetragenen Argumentwerte 2)	$[-2^{31}+1, 2^{31}-1]$
		TCj	Anzahl der Eintragungen in Tabelle j 2)	$[0, 2^{31}-1]$
		TDj	Standardabweichung der in der Tabelle j eingetragenen Argumentwerte 2)	$[-2^{31}+1, 2^{31}-1]$
	Zufallszahlengenerator	RNj	Zufallszahl des Zufallszahlengenerators j	$[0, 999]$ als Funktionsargument: $[0, 0.999999]$
		RLj	Zufallszahl des Zufallszahlengenerators j	$[0, 0.999999]$
	Systemuhr	RC	relative Zeit	$[0, 2^{31}-1]$
		AC	absolute Zeit	$[0, 2^{31}-1]$
Berechnungselement	Funktion	FNj	Wert der Funktion j	$[-2^{31}+1, 2^{31}-1]$
		DEj(a,b)	Funktionswert der Erlangverteilung a Erwartungswert b Anzahl Phasen	$[0, 10^9]$
		DLj(a,b)	Funktionswert der Log.-Normalverteilung a Erwartungswert b geometrische Dispersion	$[0, 10^9]$
		DNj(a,b)	Funktionswert der Normalverteilung a Erwartungswert b Standardabweichung	$[-10^9, +10^9]$
		DUj(a,b)	Funktionswert der Gleichverteilung a untere Grenze b obere Grenze	$[-10^9, +10^9]$

Fortsetzung Übersicht der numerischen Standardsymbole (NSS)

1	2	3	4	5
		DWj(a,b)	Funktionswert der Weibullverteilung a Parameter 1 Skalenparameter b Parameter 2 Formparameter	$[0,10^9]$
		DXj(a)	Funktionswert der Exponentialverteilung a Erwartungswert	$[0,10^9]$
			zu dem vom Zufallsgenerator j aufgerufenen Funktionswert	
	arithmetische Variable	AVj	Wert der arithmetischen Variablen j	$[-2^{31}+1,2^{31}-1]$
	Boolesche Variable	BVj	Wert der Booleschen Variablen j	0 oder 1

(nachgestelltes j ist ganze Zahl)

1) Im Operanden B des Eingangsblockes wird angegeben, wieviel Einheiten ein Aktivator belegt.

2) Je nach Angabe im TABULATE-Block gewichtet bzw. ungewichtet

```
PS SIMDIS   5,0

BLOCK-
NUMMER  *ADR   OPERATION  OPERANDEN                    KOMMENTAR
        *               SIMULATION EINES FERTIGUNGSPROZESSES
               SIMULATE
         REIFO MATRIX     MB,4,6         TECHNOLOGISCHE REIHENFOLGE
         BZEIT MATRIX     MH,4,6         BEARBEITUNGSZEITEN
         VERF  MATRIX     MH,1,4         VERFUEGBARKEIT DER MASCHINEN
         ANKU  MATRIX     MH,1,6         ANKUNFTSVERSCHIEBUNG PRO WERKSTUECK
         FZEIT MATRIX     MH,1,6         FERTIGSTELLUNGSTERMINE
         SLACK MATRIX     MH,1,6         RESTLICHE BEARBEITUNGSZEITEN
         BEGIN MATRIX     MF,4,6         BEARBEITUNGSBEGINN
         ENDE  MATRIX     MF,5,6   BEARBEITUNGSENDE PRO MASCHINE UND WERKSTUECK
         FOLGE MATRIX     MB,4,6         ORGANISATORISCHE REIHENFOLGE
         SLACK VARIABLE   MH#FZEIT(1,PB4)-RC-MH#SLACK(1,PB4)

        *         SIMULATION DER BELEGUNG GEMAESS TECHNOLOGISCHER FOLGE

1              GENERATE   ,,,1,,4PB,2PF
2              SPLIT      5,,4PB         ERZEUGEN WERKSTUECKE
3              ADVANCE    MH#ANKU(1,PB4)  ANKUNFTSTERMIN PRO WERKSTUECK
4        SPR2  TRANSFER   ,WAHL          REGEL: FIFO,KOZ
5              MSAVEVALUE SLACK,1,PB4,0,MH
6              ASSIGN     2,1,,PB
7              MSAVEVALUE SLACK+,1,PB4,MH#BZEIT(PB2,PB4),MH
8              LOOP       2PB,ZB#REIFO,,*-1
9        WAHL  MSELEC MIN 1PB,REIFO,1-ZB#REIFO,PB4,,,MB  AUSWAHL GEM. FOLGE
10             TEST LE    PB1,ZB#REIFO,ENDE
11             MSAVEVALUE REIFO+,PB2,PB4,ZB#REIFO,MB
12             ASSIGN     1,MH#BZEIT(PB2,PB4),,PF  BEARBEITUNGSZEIT GEM. STUFE
13             TEST G     PF1,0,WAHL     UEBERSPRINGEN MASCHINE
14       SPR3  TRANSFER   ,REGEL         REGEL: FIFO
15             ASSIGN     2,AV#SLACK,,PF   SLACK-ZEIT
16             TEST E     CN*PB2,0,REGEL
17             SPLIT      1,UNLI
18       REGEL LINK       PB2,FIFO,BELEG FIFO-REGEL
```

```
19  BELEG GATE FNU    PB2,REGEL       WARTEN,FALLS MASCHINE BELEGT IST
20        SEIZE       PB2             BELEGEN MASCHINE
21        MSAVEVALUE  BEGIN,PB2,PB4,RC,MF  BEARBEITUNGSBEGINN REGISTRIEREN
22        ADVANCE     PF1             BEARBEITUNGSZEIT
23        RELEASE     PB2             FREIGEBEN MASCHINE
24        MSAVEVALUE  ENDE,PB2,PB4,RC,MF  BEARBEITUNGSENDE REGISTRIEREN
25        MSAVEVALUE  ENDE,ZF#ENDE,PB4,RC,MF
26  SPR4  TRANSFER    ,SLK1           REGEL: FIFO,KOZ
27        UNLINK      PB2,KORR1,ALL
28        PRIORITY    0,BUFFER        WARTEN,BIS 'SLACK'-ZEIT KORRIGIERT IST
29        MSAVEVALUE  SLACK-,1,PB4,PF1,MH  NOCH NOTWENDIGE BEARBEITUNGSZEIT
30  SLK1  UNLINK      PB2,BELEG,1
31        TRANSFER    ,WAHL
32  ENDE  ASSEMBLE    YB#REIFO

    *       AUFSTELLUNG DES PLANES DER BEARBEITUNGSFOLGE PRO MASCHINE
33        MSAVEVALUE  REIFO-,1-ZB#REIFO,1-YB#REIFO,ZB#REIFO,MB
34        ASSIGN      2,1,,PB
35  ZEILE ASSIGN      4,1,,PB
36  SPALT MSELEC MIN  1PB,BEGIN,PB2,1-YF#BEGIN,,,MF
37        MSAVEVALUE  BEGIN+,PB2,PB3,RC,MF
38        SAVEVALUE   FOLGE+,1,XB
39        TEST E      MH#BZEIT(PB2,PB3),0,FOLGE
40        SAVEVALUE   FOLGE,0,XB
41  FOLGE MSAVEVALUE  FOLGE,PB2,PB3,XB#FOLGE,MB
42        LOOP        4PB,YF#BEGIN,,SPALT
43        MSAVEVALUE  BEGIN-,PB2,1-YF#BEGIN,RC,MF
44        SAVEVALUE   FOLGE,0,XB
45        LOOP        2PB,ZF#BEGIN,,ZEILE
46        TERMINATE   1
47  KORR  ASSIGN      2,AV#SLACK,,PF   'SLACK'-ZEIT
48        TRANSFER    ,BELEG
49  KORR1 ASSIGN      2,AV#SLACK,.PF   'SLACK'-ZEIT
50        LINK        PB2,2PF
```

```
51      UNLI  UNLINK      PB2,BELEG,1    WERKSTUECKAUSWAHL GEMAESS REGEL
52            TERMINATE

        *       SIMULATION DER VERFUEGBARKEIT DER MASCHINEN

53            GENERATE    ,,,1,10,4PB
54            SPLIT       3,,2PB         ERZEUGEN FERTIGUNGSSTUFEN
55            SEIZE       PB2
56            ADVANCE     MH#VERF(1,PB2)  BELEGEN DER FERTIGUNGSSTUFE
57            RELEASF     PB2            MASCHINE VERFUEGBAR
58      SPR1  UNLINK      PB2,BELEG,1    REGEL: FIFO,KOZ
59            TERMINATE

        *       VARIANTE NACH FIFO-REGEL
              INITIAL     MB#REIFO(1-4,1-6),(1,2,3,1,4,2,2,4,1,2,1,1,
        4,1,2,3,3,4,3,3,4,4,2,3)
              INITIAL     MH#BZEIT(1-4,1-6),(2,5,3,4,5,1,5,1,1,2,2,0,
        7,5,1,3,7,3,1,2,6,1,0,1)
              INITIAL     MH#VERF(1,1-4),(0,1,0,0)
              INITIAL     MH#FZEIT(1,1-6),(18,(3)15,(2)10)
              START       1

        *       VARIANTE NACH KOZ-REGEL
18      REGEL LINK        PB2,1PF        KOZ-REGEL
VMC0001  MULTIPLF DEFINITION
14      SPR3  CONTINUE    REGEL: SLACK,KOZ
VMC0001  MULTIPLF DEFINITION
              CLEAR
              INITIAL     MB#REIFO(1-4,1-6),(1,2,3,1,4,2,2,4,1,2,1,1,
        4,1,2,3,3,4,3,3,4,4,2,3)
              INITIAL     MH#BZEIT(1-4,1-6),(2,5,3,4,5,1,5,1,1,2,2,0,
        7,5,1,3,7,3,1,2,6,1,0,1)
              INITIAL     MH#FZEIT(1,1-6),(18,(3)15,(2)10)
              INITIAL     MH#VERF(1,1-4),(0,1,0,0)
              START       1
```

Fortsetzung P S S I M D I S 5,0

```
        *         VARIANTE NACH SLACK-REGEL
58        SPR1  UNLINK      PB2,KORR,ALL  REGEL; SLACK
VMC0001    MULTIPLE DEFINITION
4         SPR2  CONTINUE    REGEL; SLACK
VMC0001    MULTIPLE DEFINITION
18        REGEL LINK        PB2,2PF       SLACK-REGEL
VMC0001    MULTIPLE DEFINITION
26        SPR4  CONTINUE    REGEL: SLACK
VMC0001    MULTIPLE DEFINITION
                CLEAR
                INITIAL     MB#REIFO(1-4,1-6),(1,2,3,1,4,2,2,4,1,2,1,1,
        4,1,2,3,3,4,3,3,4,4,2,3)
                INITIAL     MH#BZEIT(1-4,1-6),(2,5,3,4,5,1,5,1,1,2,2,0,
        7,5,1,3,7,3,1,2,6,1,0,1)
                INITIAL     MH#VERF(1,1-4),(0,1,0,0)
                INITIAL     MH#FZEIT(1,1-6),(18,(3)15,(2)10)
                START       1
                END
```

SYMBOL CROSS-REFERENCE

ELEMENT TYPE	SYMBOL	NUMBER	DEFINITIONS		
BLOCK	BELEG	19	34		
	ENDE	32	47		
	FOLGE	41	58		
	KORR	47	65		
	KORR1	49	67		
	REGEL	18	33	92	106
	SLK1	30	45		
	SPALT	36	53		
	SPR1	58	79	104	
	SPR2	4	19	105	
	SPR3	14	29	93	
	SPR4	26	41	107	
	UNLI	51	69		
	WAHL	9	24		
	ZEILE	35	52		
SAVEVALUE (BYTE)	FOLGE	1			
MATRIX (FULLWORD)	BEGIN	2	9		
	ENDE	3	10		
MATRIX (HALFWORD)	ANKU	3	6		
	BZEIT	1	4		
	FZEIT	4	7		
	SLACK	5	8		
	VERF	2	5		
MATRIX (BYTE)	FOLGE	3	11		
	REIFO	2	3		
VARIABLE	SLACK	1	12		

REFERENCES

33	45	66	69	79					
25									
56									
104									
42									
29	31	34							
41									
59									
32									
19	28	46							
62									
55	57	58	61						
36	53	53	54	59	60	60	62		
39	40	40							
18									
22	27	56	85	97	111				
12	88	99	114						
12	20	22	44						
77	87	100	113						
58									
23	24	24	25	26	26	47	50	50	50
50	83	95	109						
30	65	67							

```
BLOCK-
NUMMER  *ADR   OPERATION   OPERANDEN                  KOMMENTAR
               REALLOCATE  FAC,100/BLO,100/XAC,100

        *        SIMULATION EINES INSTANDHALTUNGSPROZESSES

               SIMULATE
        REP    EQU         1(5),Q,XF,XH
        LAUF   EQU         6(5),Q,XF,XH
        WARTE  EQU         11(5),Q,XF
        REPO   EQU         11(5),XH
        GRAPH  EQU         1(40),XB
        PRIOR  MATRIX      MB,1,5

        ANZ    FUNCTION    PB1,L5          MASCHINEN/GRUPPE
       ,16/,4/,10/,17/,15

               INITIAL     MB#PRIOR(1,1-5),(20,0,0,10,0)
               INITIAL     SK#ARB,17       ANZAHL ARBEITSKRAEFTE
               INITIAL     XB#MAGRP,5      ANZAHL DER MASCHINENGRUPPEN
               INITIAL     XB#ANF,2        BEGINN DER EXPERIMENTE MIT 2 ARBEITERN
               INITIAL     XB#EXPMT,7      ANZAHL EXPERIMENTE
               INITIAL     XH#PFPKT,200
               INITIAL     XH#REP-XH5,10/XH2,30/XH4,20  REP-ZEITEN (UNTERE GRENZ
               INITIAL     XH#REPO,20/XH12,40/XH13-XH14,50/XH15,80 (OBERE GRENZE
               INITIAL     XH#LAUF,33/XH7-XH10,100/XH9,50 MITTLERE LAUFZEIT
        MAGRP  VARIABLE    XB#MAGRP-1      ANZAHL KOPIEN BEI GRUPPENERZEUGUNG
        MANUM  VARIABLE    FN#ANZ-1        ANZAHL KOPIEN BEI MASCHINENERZEUGUNG
        REP    FVARIABLE   QA*PB1*10000/FN#ANZ
        LAUF   FVARIABLE   QA*PB2*10000/FN#ANZ
        WARTE  FVARIABLE   QA*PB3*10000/FN#ANZ
        DMLZT  FVARIABLE   XL#DMLZT*1000/XH#ZAEHL
        DMLZ   VARIABLE    XF#DMLZT/10
        MASCH  FVARIABLE   QA#TOTAL*100/XH#ZAEHL
        ALEI   FVARIABLE   SL#ARB*100      AUSLASTUNG DER AK IN PROZENT
```

```
           STRUK STARTMACRO
          *    FESTLEGEN EINER STRUKTUR INNERHALB VON ELEMENTETYPEN
          *       PB1:  NUMMER FUER REPARATUR/NUMMER DER MASCHINENGRUPPE (1-N)
          *       PB2:  NUMMER FUER LAUFEN  (N+1 - 2N)
          *       PB3:  NUMMER FUER WARTEN  (2N+1 - 3N)
                SPLIT       V#MAGRP,,3PB  ERZEUGEN MASCHINENGRUPPEN
                ASSIGN      1-2+,PB3,PB
                ASSIGN      2-3+,XB#MAGRP,PB
                ASSIGN      3+,XB#MAGRP,PB
                ENDMACRO

1               GENERATE    ,,,1,,3PB,1PH
           STRUK MACRO
         ++    FESTLEGEN EINER STRUKTUR INNERHALB VON ELEMENTETYPEN
         ++       PB1:  NUMMER FUER REPARATUR/NUMMER DER MASCHINENGRUPPE (1-N)
         ++       PB2:  NUMMER FUER LAUFEN  (N+1 - 2N)
         ++       PB3:  NUMMER FUER WARTEN  (2N+1 - 3N)
2        +      SPLIT       V#MAGRP,,3PB  ERZEUGEN MASCHINENGRUPPEN
3        +      ASSIGN      1-2+,PB3,PB
4        +      ASSIGN      2-3+,XB#MAGRP,PB
5        +      ASSIGN      3+,XB#MAGRP,PB
6               PRIORITY    MB#PRIOR(1,PB1)  PRIORITAET DER BEARBEITUNG
7               ASSIGN      1,XH#ZAEHL,PH
          *       PH1:  FORTLAUFENDE MASCHINENNR, UEBER ALLE GRUPPEN
8               SPLIT       V#MANUM,LAUF,1PH ERZEUGEN MASCHINEN JE GRUPPE
9               SAVEVALUE   ZAEHL+,FN#ANZ,XH LETZTE MA-NR, DER GRUPPE
          *                    ABLAUF ELEMENTARZYKLUS: LAUFEN,WARTEN,REPARIEREN
```

```
10       LAUF   QUEUE      PB2               ERFASSEN DER LAUFZEITEN JE GRUPPE
11              SEIZE      PH1               ERFASSEN DER LAUFZEITEN JE MASCHINE
12              ADVANCE    DX1(XH*PB2)       MASCHINENLAUFZEIT ENTSPR. GRUPPE
13              RELEASE    PH1               AUSFALL DER MASCHINE
14              DEPART     PB2
15              QUEUE      PB3               ERFASSEN DER WARTEZEITEN JE GRUPPE
16              QUEUE      TOTAL             ERFASSEN DER WARTEZEITEN (TOTAL)
17              ENTER      ARB               WARTEN AUF FREIE ARBEITSKRAFT
18              DEPART     TOTAL
19              DEPART     PB3
20              QUEUE      PB1               ERFASSEN DER RERARATURZEITEN JE GRUPPE
21              ADVANCE    DU2(XH*PB1,XH*PB3)  REPARATURZEIT JE GRUPPE
22              LEAVE      ARB               ARBEITSKRAFT
23              DEPART     PB1                  FREIGEBEN
24              TRANSFER   ,LAUF             ENDE ELEMENTARZYKLUS

        *                  AUSWERTUNG DER SIMULATIONSERGEBNISSE

25              GENERATE   XH#PFPKT,,,,,3PB   STEUERUNG DER PRUEFZEITPUNKTE
26              TEST E     TG1,1,ENDE
27              SAVEVALUE  DMLZT,0,XL
         STRUK  MACRO
       **    FESTLEGEN EINER STRUKTUR INNERHALB VON ELEMENTETYPEN
       **       PB1:  NUMMER FUER REPARATUR/NUMMER DER MASCHINENGRUPPE (1-N)
       **       PB2:  NUMMER FUER LAUFEN   (N+1 - 2N)
       **       PB3:  NUMMER FUER WARTEN   (2N+1 - 3N)
28     +        SPLIT      V#MAGRP,,3PB  ERZEUGEN MASCHINENGRUPPEN
29     +        ASSIGN     1-2+,PB3,PB
30     +        ASSIGN     2-3+,XB#MAGRP,PB
31     +        ASSIGN     3+,XB#MAGRP,PB
32              SAVEVALUE  PB1,V#REP         BERECHNUNG DER FUER
33              SAVEVALUE  PB2,V#LAUF           DIE TABELLE
34              SAVEVALUE  PB3,V#WARTE             NOTWENDIGEN PROZENTUALEN
35              SAVEVALUE  DMLZT+,QA*PB2,XL           WERTW
36              ASSEMBLE   XB#MAGRP
```

Fortsetzung   P S   S I M D I S   5,0

```
37                SAVEVALUE  DMLZT,V#DMLZT,XF
38                SAVEVALUE  ZEIT,RC,XF
39                SAVEVALUE  XB#ANF,V#ALEI,XB
40                SAVEVALUE  NUM,XB#ANF,XB
41                SAVEVALUE  NUM+,XB#EXPMT,XB
42                SAVEVALUE  XB#NUM,V#DMLZ,XB
43                SAVEVALUE  NUM+,XB#EXPMT,XB
44                SAVEVALUE  XB#NUM,V#MASCH,XB
45                SAVEVALUE  ANF+,1,XB
46          PRINT CONTINUE
47          ENDE  TERMINATE  1

                  START      1,NR
                  INITIAL    XB#ANF,2        BEGINN DER EXPERIMENTE
                  INITIAL    XB#EXPMT,7      ANZAHL EXPERIMENTE
46          PRINT PRINT      100,200,REP
 VMCOOCI      WIEDERHOLTE DEFINITION EINES SYMBOLS
                  RESET
                  START      5,NP
                  RESET
                  INITIAL    SK#ARB,18       ANZAHL ARBEITSKRAEFTE
                  START      5,NP
                  RESET
                  INITIAL    SK#ARB,19       ANZAHL ARBEITSKRAEFTE
                  START      5,NP
                  RESET
                  INITIAL    SK#ARB,20       ANZAHL DER ARBEITSKRAEFTE
                  START      5,NP
                  RESET
                  INITIAL    SK#ARB,21       ANZAHL DER ARBEITSKRAEFTE
                  START      5,NP
                  RESET
                  INITIAL    SK#ARB,22       ANZAHL DER ARBEITSKRAEFTE
                  START      5,NP
                  RESET
                  INITIAL    SK#ARB,23       ANZAHL DER ARBEITSKRAEFTE
```

```
46         PRINT CONTINUE
 VMCOOCI    WIEDERHOLTE DEFINITION EINES SYMBOLS
               START       5;NS

               REPORT
               OUTPUT
        M 100  LABEL
               SPACE       3
         35    TEXT        SIMULATION EINES INSTANDHALTUNGSPROZESSES
               SPACE
         10    TEXT        SIMULATIONSZEITRAUM: "XF#ZEIT,2/1LXXX" TAGE
               SPACE
         30    TEXT        WARTE-  REP,-  LAUF-  ANZAHL  WARTEZEIT  MAX, ANZAHL
         12    TEXT        TYP                 ZEIT    ZEIT    ZEIT   D.REP.  VOR E*
        INER  GLEICHZEITIG
         60    TEXT        REPARATUR  WART. MASCH.
               SPACE       1
         10    TEXT        GRUPPE 1               "XF#WARTE,2/2LXX,X"   "XF#REP,2/2*
        LXX,X"  "XF#LAUF,2/2LXX,X"   "Q#REP,4/XXXX"    "Q#WARTE,8/1LXX,XX"   *
              "Q#WARTE,2/XXX"
         10    TEXT        GRUPPE 2               "XF#WARTE+1,2/2LXX,X"   "XF#REP+1*
        ,2/2LXX,X"  "XF#LAUF+1,2/2LXX,X"   "Q#REP+1,4/XXXX"     "Q#WARTE+1,8/1*
        LXX,XX"        "Q#WARTE+1,2/XXX"
         10    TEXT        GRUPPE 3               "XF#WARTE+2,2/2LXX,X"   "XF#REP+2*
        ,2/2LXX,X"  "XF#LAUF+2,2/2LXX,X"   "Q#REP+2,4/XXXX"     "Q#WARTE+2,8/1*
        LXX,XX"        "Q#WARTE+2,2/XXX"
         10    TEXT        GRUPPE 4               "XF#WARTE+3,2/2LXX,X"   "XF#REP+3*
        ,2/2LXX,X"  "XF#LAUF+3,2/2LXX,X"   "Q#REP+3,4/XXXX"     "Q#WARTE+3,8/1*
        LXX,XX"        "Q#WARTE+3,2/XXX"
         10    TEXT        GRUPPE 5               "XF#WARTE+4,2/2LXX,X"   "XF#REP+4*
        ,2/2LXX,X"  "XF#LAUF+4,2/2LXX,X"   "Q#REP+4,4/XXXX"     "Q#WARTE+4,8/1*
        LXX,XX"        "Q#WARTE+4,2/XXX"
               SPACE       1
         15    TEXT        MAXIMAL WARTEN           "Q#TOTAL,2/XXX"   MASCHINEN  *
               DURCHGEFUEHRTE REPARATUREN                "S#ARB,9/XXXXX"
```

```
  15     TEXT         DURCHSCHNITTLICH WARTEN "Q#TOTAL,3/XX.XX" MASCHINEN  *
         DURCHSCHN.   LAUFLEISTUNG ALLER MASCHINEN   "XF#DMLZT,2/1LXX.X" %
         SPACE        1
  15     TEXT         ANZAHL DER ARBEITER     "S#ARB,2/XX"
  15     TEXT         DAVON SIND DURCHSCHNITTLICH "S#ARB,3/XX.XX" BESCHAEFT*
 IGT
M 200    LABEL
         GRAPH        XB,2,8

         GRAPH        XB,9,15,-
         ORIGIN       32,10,.
         X            NO,4,7
         Y            38,2,31,1
  1      STATEMENT    1,4,%
  1      STATEMENT    12,90,--- AUSLASTUNG DER MASCHINEN
  1      STATEMENT    14,90,*** AUSLASTUNG DER ARBEITER
  1      STATEMENT    33,20,17          18          19          20          21 *
          22           23    ARBEITER
         ENDGRAPH
         GRAPH        XB,16,22
         ORIGIN       23,10,.
         X            NO,4,7
         Y            0,1,22,1
  1      STATEMENT    1,5,%
  1      STATEMENT    10,35,PROZ. ANTEIL WARTENDER MASCHINEN
  1      STATEMENT    24,20,17          18          19          20          21 *
          22           23    ARBEITER
         ENDGRAPH
         END
```

SYMBOLNACHWEIS

ELEMENTTYP	SYMBOL	NUMMER	DEFINITIONEN		
BLOCK	ENDE	47	91		
	LAUF	10	54		
	PRINT	46	90	96	116
SPEICHER	ARB	1			
WARTESCHLANGE	LAUF	6	7		
	REP	1	6		
	TOTAL	16			
	WARTE	11	8		
SKALAR (VOLLWORT)	DMLZT	16			
	LAUF	6	7		
	REP	1	6		
	WARTE	11	8		
	ZEIT	17			
SKALAR (HALBWORT)	LAUF	6	7		
	PFPKT	16			
	REP	1	6		
	REPO	11	9		
	ZAEHL	17			
SKALAR (BYTE)	ANF	42			
	EXPMT	43			
	GRAPH	1	10		
	MAGRP	41			
	NUM	44			
SKALAR (GLEITK.)	DMLZT	1			
MATRIX (BYTE)	PRIOR	1	11		
FUNKTION	ANZ	1	13		
VARIABLE	ALEI	9	33		
	DMLZ	7	31		
	DMLZT	6	30		
	LAUF	4	28		
	MAGRP	1	25		
	MANUM	2	26		
	MASCH	8	32		
	REP	3	27		
	WARTE	5	29		

BEZUGNAHMEN

73	
51	68

17	33	61	66	100	103	106	109	112	115
151	155	157							

136	139	142	145	148					
32	60	62	151	153					
136	136	139	139	142	142	145	145	148	148

31	81	153		
136	139	142	145	148
136	139	142	145	148
136	139	142	145	148
82	127			

24			
21	72		
22			
23			
30	32	49	52

19	83	84	89	94
20	85	87	95	

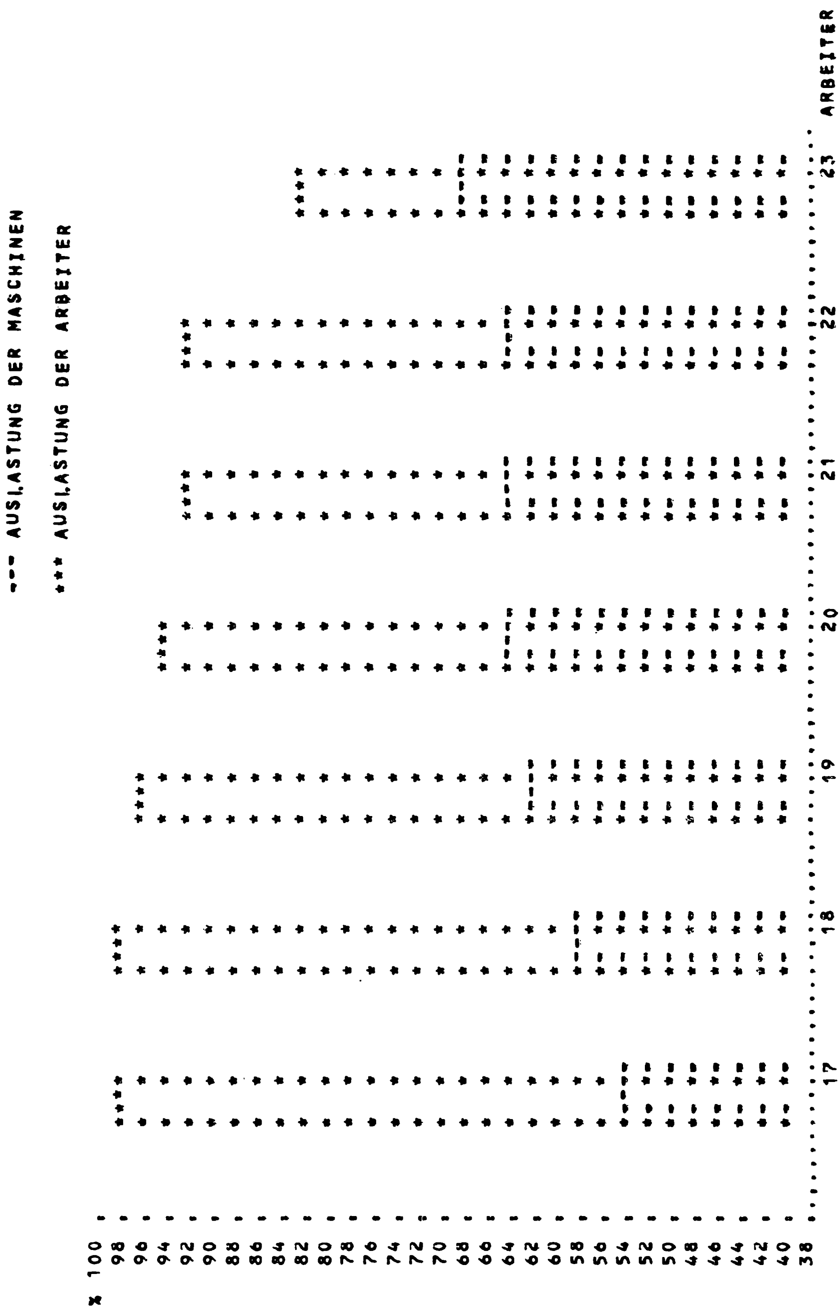
18	25	47	47	75	75	80
84	85	86	87	88		

30	74	79

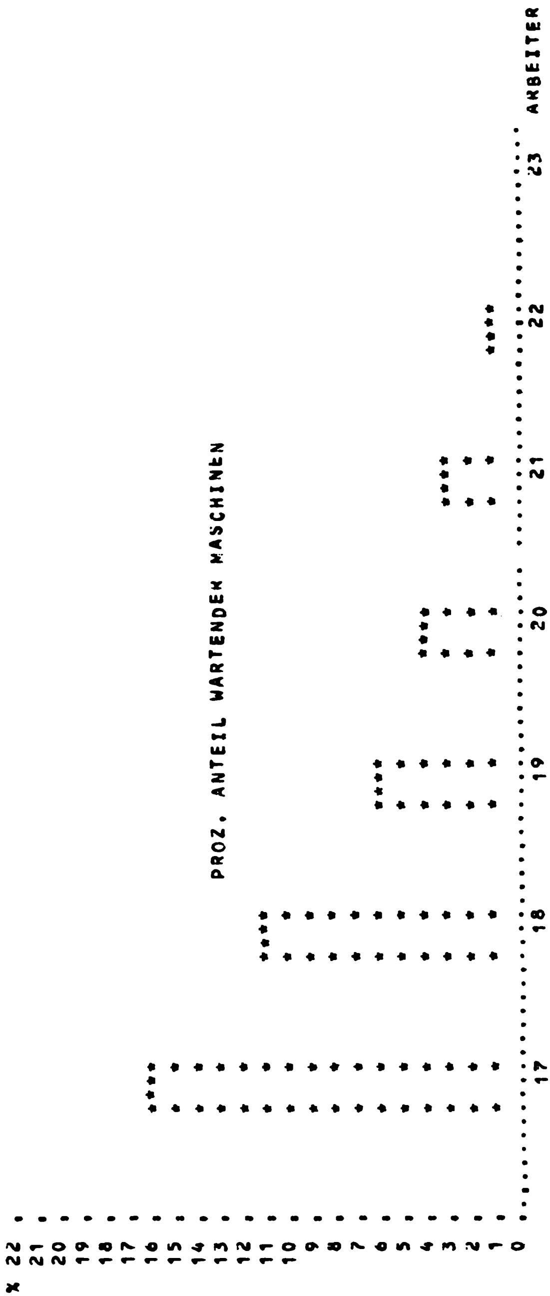
16	48

26	27	28	29	52

83	
86	
81	
77	
47	75
51	
88	
76	
78	



-·- AUSLASTUNG DER MASCHINEN
*** AUSLASTUNG DER ARBEITER
x 100
98
96
94
92
90
88
86
84
82
80
78
76
74
72
70
68
66
64
62
60
58
56
54
52
50
48
46
44
42
40
38
17
18
19
20
21
22
23
ARBEITER


PROZ. ANTEIL WARTENDER MASCHINEN
%
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
17
18
19
20
21
22
23
ARBEITER

Druck:

ZENTRALSTELLE FÜR LEHR- UND ORGANISATIONSMITTEL DES MINISTERIUMS FÜR HOCH- UND FACHSCHULWESEN, ZWICKAU

Ag 628/499/80/DDR/1100-ZLO 722/80