



Optimierung von Fertigungsabläufen – Implementierung von Methoden der computergestützten Maschinenbelegungsplanung

Prof. Dr. Matthias Forster, Diana Malskies, Gabor Settegast, Frank Wardenski, Angelique Wortmann

1. Einleitung

PPS (Produktionsplanung und -steuerung) umfaßt die Planung von Produktionsprogrammen und das operative Produktionsmanagement. Zu den Aufgaben der PPS zählen unter anderem die Datenverwaltung, Produktionsprogrammplanung, Bedarfsplanung, Planung und Steuerung der Eigenfertigung und des Fremdbezugs. Man unterscheidet simultane und sukzessive Planungskonzepte. Simultan bedeutet, daß alle Teilaufgaben der Produktionsplanung gleichzeitig geplant werden. Da dieses mit einem extrem großen Rechenaufwand verbunden ist, haben sie sich in der Praxis nicht durchgesetzt. Bei der sukzessiven Planung wird die Gesamtaufgabe in Teilaufgaben zerlegt und diese dann nacheinander geplant (schrittweise, stufenweise). Die Ergebnisse der einen Teilaufgabe bilden den Input der nächsten Teilaufgabe. Mit dem Einsatz von PPS-Systemen sollen die Durchlaufzeiten reduziert und eine bessere Auslastung der Kapazitäten erreicht werden. Als weitere Ziele können die Bestandsreduzierung, die Erhöhung der Termintreue, die Steigerung der Flexibilität und die Erhöhung der Auskunftsbereitschaft genannt werden [2].

Wir stellen ein Projekt vor, das eine Aufgabe aus der Produktionsprozeßplanung zum Gegenstand hatte. Die Aufgabe bestand darin, Verfahren zur optimalen Belegung mehrerer Maschinen zu implementieren.

2. Darstellung des studentischen Projektes

2.1 Die Problemstellung

Dieses studentische Projekt wurde im Rahmen des Wahlpflichtfachs Produktionswirtschaft von Studenten/-innen der Wirtschaftsinformatik der Seminargruppe WI/93 bearbeitet. Ziel des Projektes war die Implementierung eines Planungssystems für ein sehr allgemeines Problem der Maschinenbelegung. Als Programmiersprache wurde Borland C++[®] verwendet.

In der Anlaufphase wurde von uns festgelegt, daß mehrere Fertigungsaufträge in vorgegebener Reihenfolge die verschiedenen Maschinen (feste Maschinenfolge) durchlaufen. Die Maschinen- und Auftragsfolgen sind unsere Eingabedaten. Sie werden als Matrizen dargestellt. Die Matrixzeilen entsprechen dabei den Aufträgen und die Matrixspalten den Maschinen. Die Ergebnisse werden von dem Programm in einer geeigneten, grafischen Darstellung präsentiert.

2.2 Die theoretischen Grundlagen

2.2.1 Allgemeines

Die Notwendigkeit der Festlegung von Auftragsfolgen ergibt sich daraus, daß mehrere Arbeitsvorgänge dieselben Maschinen in Anspruch nehmen können. Ablaufplanung bedeutet die spezifische Planung der Reihenfolge, in der die Aufträge auf den Produktiveinheiten bearbeitet werden sollen.

Man unterscheidet die Maschinenfolge und die Auftragsfolge. Hierbei bezieht sich die Maschinenfolge (technologische Folge) auf die technische Notwendigkeit, Arbeitsvorgänge eines Auftrages nacheinander bearbeiten zu müssen, z.B. auf Grund konstruktiver Gegebenheiten. Sie ist bei der Ablaufplanung als vorher bekannt zu betrachten. Bei der Auftragsfolge ist festzulegen, in welcher Reihenfolge die Aufträge auf einer Maschine bearbeitet werden. Das Maschinenbelegungsproblem besteht im wesentlichen darin, die Auftragsfolgen zu bestimmen. Somit stellt es sich wie folgt dar:

Es sollen n Aufträge auf M Maschinen bearbeitet werden. Es sind folgende Nebenbedingungen zu beachten:

- a) Die Maschinenfolge jedes Auftrages ist fest vorgegeben.
- b) Kein Auftrag kann gleichzeitig auf mehr als einer Maschine bearbeitet werden.
- c) Keine Maschine kann gleichzeitig mehrere Aufträge bearbeiten.
- d) Alle Aufträge sind zu Beginn des Planungszeitraumes bekannt.

Zur Vereinfachung des Problems nehmen wir an, daß jedem Arbeitsvorgang die bearbeitende Maschine eindeutig zugeordnet ist: Jeder Arbeitsvorgang läßt sich daher durch das geordnete Paar (j,m) eindeutig kennzeichnen, wobei j die Auftragsnummer und m die Maschinennummer bezeichnet. Die theoretischen Grundlagen der Maschinenbelegungsplanung sind sehr ausführlich dargestellt in [1]. Einen Überblick der ablauforganisatorischen Aspekte der Produktionsplanung bietet [3].

2.2.2 Zielvorstellungen

Als mögliche Ziele für das Problem der Maschinenbelegungsplanung kommen vor allem zeitliche Ziele in Betracht. Kosten- bzw. Erfolgsgrößen lassen sich nicht genau erfassen. Die Möglichkeit der Lossplittung wird hier nicht in Betracht gezogen.

Betrachtet man nun die zeitlichen Ziele, so kristallisieren sich hauptsächlich folgende Zielbeziehungen heraus:



- a) Ziele bezüglich der Durchlaufzeiten
 - Minimierung der maximalen Durchlaufzeit
 - Minimierung der mittleren Durchlaufzeit
 - Minimierung der Wartezeiten
- b) Ziele bezüglich der Kapazitätsauslastung
 - Maximierung der Kapazitätsauslastung
 - Minimierung der Leerzeiten
- c) Ziele bezüglich der Terminabweichungen oder -einhaltung
 - Minimierung der maximalen Verspätung
 - Minimierung der Summe aller Verspätungen

Es sind noch eine Vielzahl weiterer Ziele denkbar (z.B. Minimierung der Umrüstzeiten). Zäpfel [7] verweist auf empirische Untersuchungen, aus denen klar ersichtlich wird, daß die Minimierung zeitbezogener Größen in der Praxis am wichtigsten ist.

2.2.3 Modellformulierung des Maschinenbelegungsproblems

Im nun folgenden Abschnitt und in 2.3 werden zwei in unserem Projekt angewandte Lösungsansätze näher analysiert. Hierbei wird zunächst ein gemischt-ganzzahliges Optimierungsmodell vorgestellt (siehe [4] und [6]). Das Modell zeigt auf eindrucksvolle Art und Weise die Struktur des Maschinenbelegungsproblems. In 2.3 wird ein Näherungsverfahren für das betrachtete Problem dargestellt. Hinsichtlich der Notation orientieren wir uns an Zäpfel [7].

Mit $S=(s_{jm})$ bezeichnen wir die Maschinenfolgmatrix. Die n Zeilen entsprechen den Aufträgen, die M Spalten den Maschinen. Ein Element s_{jm} dieser $n \times M$ -Matrix gibt für den Auftrag j die Stelle a_m , an der die Maschine m in der Maschinenfolge vorkommt.

Die Bearbeitungszeiten der einzelnen Arbeitsvorgänge jedes Auftrages werden in der Matrix (a_{jm}) festgelegt. Diese Matrix wird daher als Bearbeitungszeitenmatrix bezeichnet. Ein Element der Matrix (a_{jm}) gibt die benötigte Bearbeitungszeit des j -ten Auftrages auf der Maschine m an.

Variablen:

Für die Indizes soll jeweils gelten: $j=1, \dots, n$, $m=1, \dots, M$ und $k=1, \dots, n$ (mit $j \neq k$).

t_{jm} = Beginnzeit des Auftrages j auf der Maschine m

$$d_{jkm} = \begin{cases} 1, & \text{falls der Auftrag } j \text{ vor dem Auftrag } k \text{ auf} \\ & \text{der Maschine } m \text{ bearbeitet wird} \\ 0 & \text{sonst} \end{cases}$$

Konstanten:

a_{jm} = Produktionszeit des Auftrages j auf der Maschine m

C = hinreichend große Zahl (z. B. $C \geq \sum_j \sum_m a_{jm}$)

Nebenbedingungen:

(1) Auftragsfolgebedingungen

Zwei Aufträge können nicht gleichzeitig auf derselben Maschine bearbeitet werden:

$$\text{Entweder } t_{jm} \geq t_{km} + a_{km} \text{ oder } t_{km} \geq t_{jm} + a_{jm}$$

Diese Entweder-oder-Bedingungen lassen sich unter Zuhilfenahme der Binärvariablen $d_{jkm} \in \{0,1\}$ wie folgt ausdrücken:

$$C \cdot d_{jkm} + t_{jm} - t_{km} \geq a_{km}$$

$$C \cdot (1 - d_{jkm}) + t_{km} - t_{jm} \geq a_{jm}$$

Dabei ist C eine hinreichend große Zahl. Ist $d_{jkm} = 1$, dann wird die zweite Ungleichung $t_{km} - t_{jm} \geq a_{jm}$ aktiviert. Ist $d_{jkm} = 0$, dann wird die erste Ungleichung aktiviert.

(2) Maschinenfolgebedingungen

Die Maschinenfolgen für die Arbeitsvorgänge jedes Auftrages sind einzuhalten. Bezeichnet $t_{j[m]}$ bzw. $a_{j[m]}$ den Beginnzeitpunkt der Bearbeitung bzw. die Produktionszeit des Auftrages j auf der Maschine, die an m -ter Stelle der Maschinenfolge für den Auftrag j steht, so lauten die Bedingungen:

$$t_{j[m+1]} \geq t_{j[m]} + a_{j[m]}$$

Mit diesem Modell sind verschiedene Zielfunktionen darstellbar. Wir wählen die Minimierung der maximalen Durchlaufzeit. Sie läßt sich wie folgt ausdrücken: minimiere C_{\max} .

Wir führen also die zusätzliche Variable C_{\max} ein. Um den Zusammenhang zu den übrigen Variablen zu formulieren, sind zusätzliche Restriktion (Zykluszeitbedingungen) erforderlich. Die Zykluszeitbedingungen lauten:

$$t_{j[M]} + a_{j[M]} \leq C_{\max}$$

Die Nichtnegativitätsbedingungen und die 0-1-Bedingungen lauten:

$$t_{j[m]} \geq 0$$

$$d_{jkm} \in \{0,1\}$$

Die Problemgröße entwickelt sich gemäß folgender Tabelle:

| Variablenzahl | Auftragsfolgebedingungen | Maschinenfolgebedingungen | Zykluszeitbedingungen |
|--|---------------------------|---------------------------|-----------------------|
| $n \times M + 1 + n \times (n-1) \times M/2$ | $n \times (n-1) \times M$ | $n \times (M-1)$ | n |

Bei $n=6$ Aufträgen und $M=4$ Maschinen sind das bereits 85 Variable und 144 Nebenbedingungen.

2.3 Die Implementierung der Werkzeuge

2.3.1 Projektphasen

Der Umfang des Projektes erforderte die Definition mehrerer Implementierungsphasen:

- a) Modellformulierung und Matrixgenerator (Phase 1)
- b) Näherungsverfahren mit zwei Prioritätsregeln (Phase 2)
- c) Ergänzung des Näherungsverfahrens durch vier weitere Prioritätsregeln (Phase 3)
- d) Reportprogramm, grafische Ausgabe des Gantt-Diagramms (Phase 4).

In der *ersten Projektphase* wurden die Routinen konzipiert und geschrieben, die das exakte Lösen eines vorgegebenen Problems erlauben: Die Modellformulierung wurde in Abschnitt 2.2.3 dargestellt. Bei dem Matrixgenerator handelt es sich um ein Programm, welches eine



Problemdatei in dem Format erzeugt, das von einem Optimierungssystem verarbeitet werden kann. In der *zweiten Projektphase* wurde ein allgemeines Näherungsverfahren implementiert, das die Einbettung unterschiedlicher Prioritätsregeln zuläßt (siehe Abschnitt 2.3.2). Die Phasen 1 und 2 wurden im SS 1996 erfolgreich abgeschlossen. Das WS 96/97 diente dem Test und Ausbau der Software. In der *dritten Projektphase* wurden weitere, alternative Lösungsansätze in das Näherungsverfahren einbezogen. Aufgabe der *vierten Projektphase* war es, eine geeignete Darstellung der Rechenergebnisse zu implementieren. Der erfolgreiche Projektablauf wurde vor allem auch durch die geeignete Zerlegung in diese vier Arbeitsschritte ermöglicht.

2.3.2 Ein Näherungsverfahren für das Problem der Maschinenbelegung

Die von uns verwendete Heuristik entspricht dem Vorgehen an einer Plantafel. Es geht auf Giffler und Thompson (1960, nach [7]) zurück. Bei der Vorwärtsterminierung werden drei Arten von Arbeitsvorgängen unterschieden:

- eingeplante Arbeitsvorgänge
- einplanbare Arbeitsvorgänge
- noch nicht einplanbare Arbeitsvorgänge.

Eingeplanten Arbeitsvorgängen wurde bereits ein fester Platz in der Auftragsfolge zugewiesen. Arbeitsvorgänge, deren Vorgänger in der Maschinenfolge bereits eingeplant sind, nennt man einplanbare Arbeitsvorgänge. Ihnen kann in der Auftragsfolge ein Platz zugewiesen werden. Steht ein Arbeitsvorgang an Platz 1 der Auftragsfolge, ist er von Beginn an einplanbar.

Noch nicht einplanbare Arbeitsvorgänge haben keinen Vorgänger in der Maschinenfolge, der schon vollständig eingeplant ist, so daß deren Abarbeitung noch nicht möglich ist. Wollen gleichzeitig mindestens zwei Arbeitsvorgänge dieselbe Maschine belegen, entsteht ein Konflikt. In diesem Fall gibt es Arbeitsvorgänge (j,m) und (k,m) , die sich überlappen, d.h. eine der Bedingungen $t_{jm} \in [t_{km}, t_{km} + a_{km})$ oder $t_{km} \in [t_{jm}, t_{jm} + a_{jm})$ ist dann erfüllt. In diesem Falle ist eine Entscheidung über die Bearbeitungsreihenfolge notwendig. Dies bedeutet, daß die übrigen am Konflikt beteiligten Arbeitsvorgänge sich zeitlich verzögern.

Hierbei stellt sich die Frage nach der Methode der Konfliktlösung. Die Lösung des Konflikts basiert auf der geeigneten Auswahl einer Prioritätsregel, d. h. einer Regel über die Auswahl eines Arbeitsvorganges auf Grund eines meßbaren Kriteriums. Von uns wurden folgende Prioritätsregeln implementiert (siehe auch [7]):

- Kürzeste Operationszeit (KOZ-Regel)
- Längste Operationszeit (LOZ-Regel)
- Größte Restbearbeitungszeit (GRB-Regel)
- Kürzeste Restbearbeitungszeit (KRB-Regel)
- Größte Gesamtbearbeitungszeit (GGB-Regel)
- Kleinste Gesamtbearbeitungszeit (KGB-Regel).

Beispielsweise wird im Konfliktfall bei Anwendung der KOZ-Regel der Arbeitsvorgang mit der kürzesten Bearbeitungszeit ausgewählt.

Somit stellt sich das Näherungsverfahren wie folgt dar:

- 1. *Schritt*: Der erste Arbeitsvorgang jedes Auftrages wird bestimmt. Der Beginnzeitpunkt wird gleich Null gesetzt. Die Menge der eingeplanten Arbeitsvorgänge ist leer. Somit ist der Fertigstellungszeitpunkt des ersten Arbeitsvorganges gleich der Produktionszeit.
- 2. *Schritt*: Gesucht wird hier der einplanbare Arbeitsvorgang, der zuerst mit der Bearbeitung fertig ist. Bei mehreren möglichen Arbeitsvorgängen wird ein beliebiger ausgesucht. Der Arbeitsvorgang gehört zum Auftrag k und wird auf Maschine m bearbeitet. Jetzt wird die Menge der einplanbaren Arbeitsvorgänge auf dieser Maschine bestimmt (Konfliktmenge).
- 3. *Schritt*: Ist die Konfliktmenge singulär, wird nach Schritt 5 verzweigt.
- 4. *Schritt*: Aus der Konfliktmenge wird ein Arbeitsvorgang nach einer der oben genannten Prioritätsregeln ausgewählt und eingeplant. Die Endzeitpunkte der restlichen Arbeitsvorgänge aus dieser Menge ergeben sich aus dem Endzeitpunkt des gewählten Arbeitsvorganges plus dessen Produktionszeit.
- 5. *Schritt*: Die Fertigstellungszeitpunkte der nachfolgenden Operationen des Vorgangs werden aktualisiert. Für die anderen einplanbaren Arbeitsvorgänge ändert sich dabei nichts.
- 6. *Schritt*: Der gewählte Arbeitsvorgang wird aus der Menge der einplanbaren Vorgänge in die Menge der eingeplanten übernommen. Die Menge der einplanbaren Vorgänge wird durch die Menge der direkten Nachfolgeoperationen des Vorgangs ergänzt.
- 7. *Schritt*: Wenn alle Aufträge abgearbeitet sind, ist die Lösung gefunden. Ansonsten verzweigen wir nach Schritt 2.

Das Näherungsverfahren benötigt zwei Matrizen, die Fertigungszeitenmatrix und Maschinenfolgematrix. Die Matrizen sind in einer Eingabedatei zusammengefaßt, die zusätzlich die Anzahl der Maschinen und Aufträge angibt.

2.3.3 Ein exaktes Verfahren für das Problem der Maschinenbelegung

Es gibt eine Reihe von Optimierern für gemischt-ganzzahlige Probleme, die kommerziell angeboten werden. In einem der Labore der Technischen Fachhochschule Wildau ist das Programm MOPS (Mathematisches Optimierungssystem) installiert [5]. MOPS benötigt die Problemdaten im sogenannten MPS Format. Das MPS Format ist ein Industriestandard zur Repräsentation mathematischer Optimierungsprobleme. Das MPS-Format ermöglicht, ein Lineares Programm mit seinen Nebenbedingungen in einer Datei abzubilden. Die Eingabedatei wird mit Hilfe unseres Programmes in das MPS-Format umgewandelt. Mit MOPS wird das Problem dann optimiert und eine Ausgabedatei erstellt. Zur Auswertung dieser Optimierung wird diese Ausgabedatei in ein lesbares Format konvertiert und in einem Gantt-Diagramm dargestellt.

2.3.4 Die grafische Darstellung der Ergebnisse

Von großer Bedeutung für den Anwender ist die geeignete Darstellung der Rechenergebnisse. In der Fertigung



ist es üblich, zur Repräsentation einer konkreten Maschinenbelegung sog. Maschinenbelegungsdiagramme zu verwenden. Betrachten wir etwa einen Anwendungsfall mit zwei Maschinen, drei Fertigungsaufträgen und den Maschinenfolgen: Auftrag 1 zuerst auf Maschine 1 und dann auf Maschine 2, Auftrag 2 zuerst auf Maschine 2 und dann auf Maschine 1 und Auftrag 3 nur auf Maschine 2. Dann stellt Abbildung 1 eine zulässige Maschinenbelegung im Maschinenbelegungsdiagramm dar. Die Abszisse entspricht der Zeitachse. Sie dient dazu, die aktuelle Maschinenbelegung abzulesen. Die Ordinate repräsentiert die beiden Maschinen. Die Länge der horizontalen Balken gibt die Bearbeitungszeit des entsprechenden Arbeitsvorganges an.

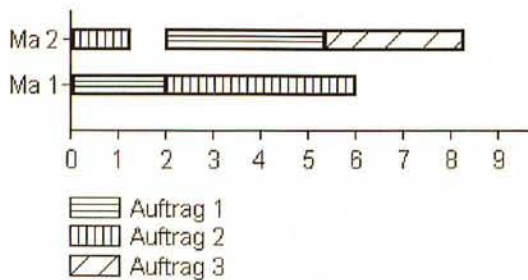


Abb. 1 Maschinenbelegungsdiagramm

Eine alternative Darstellungsform ist das Auftragsfortschrittsdiagramm. Die Ordinate repräsentiert hier die drei Fertigungsaufträge. In dem Auftragsfortschrittsdiagramm der Abbildung 2 lesen wir den Arbeitsfortschritt jedes Auftrages ab. Maschinenbelegungs- und Auftragsfortschrittsdiagramme sind verschiedene Formen von Gantt-Diagrammen.

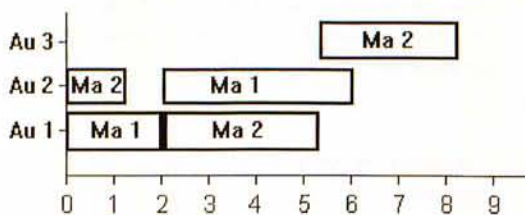


Abb. 2 Auftragsfortschrittsdiagramm

2.4 Die Grenzen der Verfahren

Falls das Programm in der betrieblichen Praxis eingesetzt werden sollte, müßte das Modell noch realitätsnäher formuliert werden. So wären unter anderem folgende Gesichtspunkte zu berücksichtigen: Möglichkeit der Lossplittung und/oder Beachtung reihenfolgeabhängiger Rüstzeiten. Aufgabe unseres Projektes ist es lediglich, dem Benutzer einen ersten Überblick über relativ kleine Aufgaben der Maschinenbelegung zu ermöglichen. Dafür aber mit einer zu vertretenden Laufzeit, die sich in Grenzen hält.

In Abschnitt 2.2.3 hatten wir bereits auf die in Abhängigkeit der Auftragsanzahl und der Maschinenanzahl schnell wachsenden Problemgrößen hingewiesen. Zudem sind gemischt-ganzzahlige Probleme extrem

schwer zu lösen. Die Auswirkung der Stärken von Mehrprozessorrechnern, die in größeren Firmen durchaus üblich sind, auf die Rechenbarkeit solcher Modelle ist noch nicht eindeutig geklärt. Da wir aber das Programm nicht für Einsatz in solchen Firmen konzipiert haben, sondern eher als Darstellungsobjekt mit dem Einsatzgebiet in Fachhochschulen, mußten wir einen Konsens finden zwischen den technischen Möglichkeiten unseres Produkts und seinem Laufzeitverhalten. Und gerade mit Hinsicht auf das Einsatzgebiet unseres Produkts war es uns wichtig, die Problemgrößen zu beschränken. Im Sinne eines Kompromisses haben wir uns dazu entschlossen, die Anzahl der am Prozeß beteiligten Maschinen auf 20 und die möglichen Fertigungsaufträge auf den Maschinen ebenfalls auf 20 zu reduzieren.

2.5 Ein Anwendungsbeispiel

Als Beispiel haben wir eine Aufgabe mit vier Aufträgen und vier Maschinen gewählt. Die Eingabedatei ist dann so aufgebaut (<Zeilennummer> in der Eingabedatei):

<01> nFA (Anzahl der Aufträge) = 4
 <02> nMA (Anzahl der Maschinen) = 4

Die Maschinenfolgematrix dieser Problemstellung wurde von uns so festgelegt (MFM):

| | | | | |
|------|---|---|---|---|
| <03> | 1 | 3 | 2 | 0 |
| <04> | 1 | 2 | 4 | 3 |
| <05> | 3 | 0 | 2 | 1 |
| <06> | 1 | 2 | 3 | 4 |

Die Zeilen bedeuten die Aufträge und die Spalten sind den Maschinen zugeordnet. Wie oben erläutert, geben die Zahlen 1 bis 4 die Abarbeitungsreihenfolge der einzelnen Arbeitsvorgänge auf den entsprechenden Maschinen an: Der Auftrag 1 ist zuerst auf der Maschine 1, dann auf der Maschine 3 und schließlich auf der Maschine 2 zu bearbeiten. Die Null in der ersten Zeile/vierten Spalte der Maschinenfolgematrix bedeutet, daß der erste Auftrag die Maschine 4 nicht durchläuft.

Drei der Aufträge beginnen auf der ersten Maschine (die Aufträge 1, 2 und 4). Die Bearbeitungszeiten spielen eine wichtige Rolle bei der Minimierung der gesamten Durchlaufzeit aller Fertigungsaufträge. Deshalb benötigen wir die Bearbeitungszeitenmatrix (Fertigungszeitenmatrix, Arbeitszeitmatrix, AZM). Sie beinhaltet in unserem Beispiel folgende Daten:

| | | | | |
|------|----|----|----|----|
| <07> | 60 | 20 | 40 | 0 |
| <08> | 50 | 10 | 30 | 20 |
| <09> | 40 | 0 | 20 | 10 |
| <10> | 70 | 30 | 20 | 60 |

Auch hier entsprechen die Zeilen den Aufträgen und die Spalten den Maschinen. Der erste Auftrag benötigt 60 Zeiteinheiten (ZE) auf Maschine 1, 20 ZE auf Maschine 2 und 40 ZE auf Maschine 3. Da der erste Fertigungsauftrag die vierte Maschine nicht belastet, ist das entsprechende Element der AZM gleich Null. Mit Hilfe von



MOPS wurde folgendes Ergebnis berechnet (als Maschinenbelegungsdiagramm):

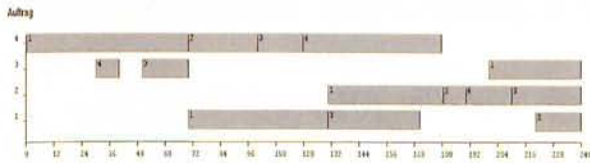


Abb. 3 Maschinenbelegungsdiagramm

Die Optimallösung hat eine Durchlaufzeit von 240 Zeiteinheiten. Mit Hilfe der von uns implementierten Varianten des Näherungsverfahrens wurden folgende Lösungen bestimmt:

| Lösungsverfahren | Durchlaufzeit [ZE] |
|------------------|--------------------|
| KOZ | 290 |
| LOZ | 240 |
| GRB | 240 |
| KRB | 240 |
| GGB | 240 |
| KGB | 290. |

In diesem kleinen Beispiel lieferten 4 der 6 implementierten Lösungsansätze die Optimallösung. Das Beispiel ist natürlich nicht repräsentativ.

3. Ausblick

Die Implementierung von Methoden der computergestützten Maschinenbelegungsplanung im Rahmen eines zweisemestrigen, studentischen Projektes zeigte, daß sich anwendungsorientierte, moderne Optimierungsmethoden in Lehrveranstaltungen umsetzen lassen. Auf Grund des Umfangs von insgesamt 4 Semesterwochenstunden waren einige Modellannahmen notwendig, die den Anwendungsbereich des Programms einschränken. Eine Weiterentwicklung des Systems wird folgende Gesichtspunkte berücksichtigen: Zunächst sind die Zielfunktion und die Nebenbedingungen so zu erweitern, daß die Möglichkeit der Lossplittung berücksichtigt wird. Auf diese Weise wird eine gleichmäßige Kapazitätsauslastung bei kurzer Durchlaufzeit erreicht. Bei den Näherungsverfahren sollen weitere Maschinenbelegungsheuristiken Eingang finden. Der Benutzer kann dann in Testläufen die für seine Anwendungen günstigste Alternative auswählen, welche in den Produktionsläufen zum Einsatz kommen soll. Die Erfahrung zeigt nämlich, daß die Wahl einer günstigen Heuristik sehr von dem jeweiligen Anwendungsgebiet abhängen kann. Zusammenfassend halten wir fest, daß mit dem Optimierungssystem ein ausbaufähiges, modulares Experimentierfeld für die Durchführungsplanung in Fertigungsbetrieben geschaffen wurde.

An diesem Projekt wurde im Vertiefungsfach Produktionswirtschaft 2 (Studiengang Wirtschaftsinformatik) gearbeitet. An dem Projekt waren die Studenten und Studentinnen Diana Malskies, Katja Schneider, Gabor Settegast, Frank Wardenski, Angelique Wortmann und Carsten Wunderlich beteiligt.

Literatur

- [1] Blazewicz, J., Ecker, K., Schmidt, G., Weglarz, J.: Scheduling in Computer and Manufacturing Systems, 1st ed., Berlin/Heidelberg 1993.
- [2] Corsten, H.: Lexikon der Betriebswirtschaftslehre. Stw. Produktionsplanung und -steuerung, 2., unwesentl. veränd. Aufl., München/Wien 1993.
- [3] Domschke, W., Scholl, A., Voß, S.: Produktionsplanung. Ablauforganisatorische Aspekte, 1. Aufl., Berlin/Heidelberg 1993.
- [4] Dürr, W., Kleibohm, K.: Operations Research. Lineare Modelle und ihre Anwendungen, 2., vollst. durchges. u. verb. Aufl., München/Wien 1988.
- [5] Suhl, U.: MOPS Mathematical Optimization System. Benutzerhandbuch Version 2.5, Berlin im Oktober 1995, Copyright © 1989–1995.
- [6] Williams, H.P.: Model Building in Mathematical Programming, 3rd ed., Chichester 1990.
- [7] Zäpfel, G.: Produktionswirtschaft. Operatives Produktions-Management, 1. Aufl., Berlin/New York 1982.

Verfasser

Prof. Dr. rer. pol. Matthias Forster
Technische Fachhochschule Wildau
Fachbereich Wirtschaft, Verwaltung und Recht
Tel. (03375) 508-950

Diana Malskies
Gabor Settegast
Frank Wardenski
Angelique Wortmann
Technische Fachhochschule Wildau
Fachbereich Wirtschaft, Verwaltung und Recht
Studentinnen und Studenten im Studiengang Wirtschaftsinformatik