Mathematisches Modell und Algorithmen der Termin- und Reihenfolgeplanung

Dr.-Ing. (SU) Yuriy Zack, Europäisches Zentrum für Mechatronik Aachen

Dr. rer. nat. Sergey Rotin, Lehrstuhl für Produktionssystematik, RWTH Aachen

Überblick

- Termin- und Reihenfolgeplanung bestehen in zeitlicher Koordinierung einzelner Arbeitsvorgänge unter Berücksichtigung der technologischen Abhängigkeiten, des begrenzten Kapazitätsangebots verschiedener Ressourcen sowie der vorgegebenen Abschlusstermine einzelner Aufträge. Als Optimierungskriterium gilt hier die Minimierung der maximalen Auftragsdauer.
- Die formulierte Aufgabe entspricht dem konventionellen Job-Shop-Problem vom Typ $J||Cmax$ mit zusätzlichen Restriktionen auf Beginn und Abschluss einiger Jobs. Es wird möglich, schon in den ersten Schritten die eventuelle Unlösbarkeit des Problems festzustellen. Die bisher in der Produktionsplanung implementierten Näherungsverfahren sind schwer anzuwenden oder wenig effektiv.
- Hier werden exakte und approximative Lösungsverfahren entwickelt, welche auf sequenzieller Annäherung an die optimale Lösung „von unten“ beruhen. Die Genauigkeitsabschätzung der angenäherten Lösungen basiert auf den entwickelten Berechnungsformeln für die untere Grenze des optimalen Kriteriumswertes.
- Es werden die Eigenschaften der frühest- und spätestmöglichen Starttermine aller Arbeitsvorgänge untersucht. Darauf basierend wird die natürliche Ordnung in den Arbeitssystemen effektiv rekonstruiert oder die Engpässe in der Ressourcenkapazität sowie in den vorgegebenen Abschlussterminen einzelner Aufträge entdeckt.
- Die entwickelten Berechnungsalgorithmen sind in das interaktive Programmpaket zur Produktionsplanung eingebaut.

A. Einleitung

Die grundlegende Aufgabe der Termin- und Reihenfolgeplanung besteht in der zeitlichen Koordinierung unterschiedlicher Arbeitsvorgänge unter Berücksichtigung der technologischen Abhängigkeiten und der begrenzten Kapazität jeder produzierenden Einheit. Als Ausgangsdaten gelten dabei die Ablaufstrukturen, welche technologische Folgen der Arbeitsvorgänge in jedem Fertigungsauftrag definieren. Diese Reihenfolgebeziehungen sowie die Dauer jeder Tätigkeit lassen sich in den meisten praktischen Fällen im Voraus bestimmen. Als Ergebnis der Terminplanung gilt eine Tabelle, die alle Start- und Endtermine der durchzuführenden Arbeitsvorgänge enthält.

Die in unserem Beitrag vorgenommene Erweiterung der geschilderten Aufgabe auf den Fall, dass es zusätzliche Restriktionen auf Beginn und Abschluss einiger Fertigungsaufträge gibt, entspricht den Anforderungen praktischer Anwendungen. Besonders aktuell ist sie für die Werkstattfertigung. Dennoch auch beim Abschließen einer Reihe von Lieferungsverträgen, die gewisse Endtermine voraussetzen, hat man schon im Vorfeld die zur Verfügung stehende Produktionskapazität richtig abzuschätzen, um die Realisierbarkeit der abzuschießenden Verträge zu prüfen. Wenn die Endtermine für jeden komplexen Fertigungsauftrag vorliegen, so kann man mit Hilfe der Netzplantechnik die frühesten und die spätesten Zeitpunkte sowie die aus ihnen resultierenden Pufferzeiten für jede Tätigkeit bestimmen und dann die minimale Dauer des Gesamtprojekts vorerst ohne Berücksichtigung der Beschränkungen auf die Ressourcenkapazität ermitteln.

Verschiedene Erzeugnisse der Fertigungsindustrie werden oft auf denselben Anlagen produziert, wobei einige Arbeitsvorgänge die begrenzten Ressourcen (Anlagen, Vorrichtungen, Transport usw.) gleichzeitig benötigen. Es ist eher Regel als Ausnahme, dass Kapazitätsangebot und -nachfrage aufeinander nicht abgestimmt sind. Zwei oder mehr Arbeitsvorgänge können zu gleicher Zeit um die freien Kapazitäten werben und damit wären die vorgesehenen Starttermine dieser Operationen nicht durchführbar. Beanspruchen mehrere Fertigungsaufträge oder ihre einzelnen Arbeitsvorgänge gleichzeitig dieselben Werkzeugmaschinen oder Bearbeitungsplätze (im weiteren alle als *Arbeitssysteme* bezeichnet), so müssen bei der Produktionsplanung die Reihenfolgen der Fertigungsaufträge (die *Auftragfolgen*) festgelegt werden. Man muss außerdem die explizite Belegung der Arbeitssysteme

durch die Aufträge ermitteln, d. h. die zeitliche Reihenfolge festlegen, in der eine bestimmte Menge von Aufträgen in den Arbeitssystemen ausgeführt wird.
Somit stellt die Termin- und Reihenfolgeplanung in der Großserienfertigung immer eine komplexe Aufgabe dar.

Die präzisierte Aufgabenstellung der Termin- und Reihenfolgeplanung beruht auf den folgenden Annahmen[1]:

i. Jeder Fertigungsauftrag hat eine Reihe von bestimmten Arbeitssystemen durchzulaufen, deren Folge im voraus bekannt ist. Diese aus technologischen Gründen hervorgehende Reihenfolge kann für jeden Auftrag anders ausfallen.
ii. Kein Fertigungsauftrag kann gleichzeitig in mehr als einem Arbeitssystem ausgeführt werden. Kein Arbeitssystem kann gleichzeitig mehrere Aufträge ausführen.
iii. Der komplexe Fertigungsauftrag kann in einzelne aufeinanderfolgende Arbeitsvorgänge eindeutig zerlegt werden. Die Bearbeitungszeit jedes Arbeitsvorganges sei bekannt und von anderen Bearbeitungsfaktoren unabhängig. Keine Arbeitsvorgänge dürfen unterbrochen werden.
iv. Jedem Fertigungsauftrag kann ein Beginn- und ein Abschlusstermin eindeutig zugeordnet werden. Diese Zeittermine sollen als Ungleichungsrestriktionen streng eingehalten werden.

Die Aufgabe der Termin- und Reihenfolgeplanung besteht nun darin, einen zulässigen Ablaufplan für die durchzuführenden Arbeitsvorgänge zu erstellen, der bezüglich mindestens einer vordefinierten Zielgröße optimal ist.

Seien E_i der vorgesehene Abschlusstermin und T_i der Zeitpunkt der wirklichen Fertigstellung des i-ten Auftrages, so kann als Optimierungskriterium die maximale Auftragsdauer $F_{\max}$ oder die mittelgewichtete Summe der Terminverspätungen $F_{\Sigma \mathrm{wL}}$ gewählt werden, d. h.

(1) Minimiere $F_{\max} = \max_i T_i$

oder

(2) Minimiere $F_{\Sigma \mathrm{wL}} = \sum_i w_i \max\left\{T_i - E_i, 0\right\}$

wobei w_i positive Gewichtskoeffizienten bezeichnen.

Das Optimierungskriterium (2) ist für praktische Aufgaben der PPS aktuell; oft haben die Aufträge verschiedene Priorität, so dass die reguläre Ausführung der mehr wiegenden Aufträge eher gewährleistet werden muss. Der entsprechende Terminplan kann (im Gegensatz zu Balas u.a. (1998)) mit Hilfe der eingeführten verschiedenen

Gewichtskoeffizienten konstruiert werden. Des weiteren betrachten wir allerdings nur die Aufgabenstellungen mit dem Kriterium (1). Das zugehörige mehrdimensionale Problem der bilinear-booleschen Programmierung, in dem die Größe $F_{\max}$ die Rolle einer Zielfunktion spielt, wird in Abschnitt B.II formuliert und erläutert.

Die hier eingeführte und zu Anfang nur verbal formulierte Aufgabe wurde seit Akers und Friedman (1955) sehr oft in der Literatur betrachtet[2]. Die Lösungsvorschläge im Bereich der Maschinenbelegung sind damit umfangreich. Wie Zäpfel (1996) zurecht bemerkt hat, existieren bei n Aufträgen und m Arbeitssystemen bis zu $(n!)^m$ theoretisch mögliche (nicht unbedingt zulässige) Belegungen. Dies deutet plausibel darauf hin, mit welchem Rechenaufwand eventuell konfrontiert wird, falls man es vorhat, sämtliche Kombinationen in der Aufgabe zu untersuchen. Daher wurden in der Fachliteratur verschiedene exakte und approximative Lösungsverfahren entwickelt: Optimierungsverfahren mit gemischten Variablen[3], Branch-and-Bound Methoden[4] u. a. Dennoch werden die meisten praxisbezogenen Aufgaben bis zum heute nur durch die approximativen Verfahren behandelt, welche phantasievolle Suchheuristiken und Prioritätsregel verwenden[5]. Besondere Rücksicht verdienen die heuristischen Verfahren von Balas (1998) sowie von Leon und Wu (1992). Diese Verfahren lassen die Job-Shop-Probleme mit Zusatzrestriktionen behandeln. Im Falle der Inkompatibilität des entstehenden Restriktionssystems schaltet sich das Verfahren von Balas automatisch auf die Suche nach anderer Lösung, welche dem Minimum der maximalen Terminverspätung (vgl. mit dem Optimierungskriterium (2)) entspricht.

Die bei den meisten Anwendungen benutzten approximativen Verfahren besitzen eine Reihe von wesentlichen Nachteilen. Es existieren zum Beispiel keine Auswertungen der Genauigkeit von ermittelten approximativen Lösungen. Fast alle bisher betrachteten Lösungsverfahren sind ungeeignet, wenn das oben formulierte Schedulingproblem die Zeitrestriktionen auf den Auftragsbeginn oder -schluss hat, und ermöglichen außerdem nicht, in den ersten Schritten des Algorithmus eine eventuell vorhandene Inkompatibilität des kompletten Restriktionssystems festzustellen.

In dieser Arbeit wird die Aufgabe der Termin- und Reihenfolgeplanung mit Annahme (iv) formuliert und gelöst. Wir untersuchen dabei die Eigenschaften der mit Hilfe der Netzplantechnik von Zimmermann (1971) ermittelten frühest- und spätest-möglichen Starttermine sämtlicher Arbeitsvorgänge. Auf den bereits in Zack (1978) erläuterten Eigenschaften von zulässigen und optimalen Lösungen basierend, lassen wir schon in den ersten Schritten des Algorithmus eine große Teilmenge von unzulässigen Plänen ausscheiden. Es werden dabei die explizit zugewiesenen Engpässe

in der Ressourcenkapazität oder in den auferlegten Zeitrestriktionen gefunden. Der Berechnungsprinzip beruht auf der Constraint-Programming Methode, die in moderner Literatur (vgl. Dorndorf u.a. (2000)) als *Constraint Propagation Technique* genannt wird. Der Hauptunterschied besteht hier aber in der Wahl des iterativ zu lösenden Ersatzproblems, von dessen Lösbarkeit a priori nichts bekannt ist. Mehr noch, der hier vorgeschlagene Algorithmus stützt sich auf keine obere Grenze, welche eventuell fehlt.

Es werden effektive Berechnungsformeln entwickelt, die eine untere Grenze der zu minimierenden Zielfunktion (1) für jede zu untersuchende Teilmenge der Terminpläne geben. Als Anfangswert F_0 der Gesamtprojektdauer wird jedes Mal die minimale untere Grenze gewählt. Nach der Überprüfung der Kompatibilität wird entweder eine Aufgabenlösung gefunden oder ein neuer Engpass in den Endzeitpunkten bestimmter Fertigungsaufträge festgestellt. Im Falle eines Engpasses ändert sich die minimale untere Grenze und die neue längere Gesamtprojektdauer wird vorausgesetzt, d. h. es gilt immer $F_s < F_{s+1}$. Sobald es keine zulässigen Lösungen bis zum neuen Bereich der Gesamtprojektdauer gibt, soll die in diesem Iterationsschritt zuerst ermittelte zulässige Lösung als optimal gelten.

Die Funktionalität der entwickelten Lösungsverfahren illustriert man anhand einiger numerischer Beispiele.

B. Mathematische Problemstellung

I. Grundbegriffe und Bezeichnungen

Es werden n Fertigungsaufträge in m unterschiedlichen Arbeitssystemen ausgeführt. Der i-te Auftrag enthält J_i aufeinanderfolgende Arbeitsvorgänge: O_{ij}, $j = 1, \ldots, J_i$. Somit ist jeder Arbeitsvorgang O_{ij} durch den Multiindex $\alpha = (i, j)$ eindeutig gekennzeichnet. Wir definieren:

$A = \{\alpha = (i, j) : \ i = 1, \ldots, n, \ j = 1, \ldots, J_i\}$ – die Menge sämtlicher Multiindizes,

$\mu(i, j)$ – die Nummer des Arbeitssystems, das der Arbeitsvorgang O_{ij} durchläuft,

$t(i, j)$ – die Bearbeitungsdauer des Arbeitsvorganges O_{ij},

$x(i, j)$ – den Starttermin des Arbeitsvorganges O_{ij},

$\theta(i, j)$ – den frühestmöglichen Starttermin des Arbeitsvorganges O_{ij},

$\tau(i, j)$ – den spätestmöglichen Starttermin des Arbeitsvorganges O_{ij},

$T_i = x(i, J_i) + t(i, J_i)$ – den wirklichen Endtermin des i-ten Fertigungsauftrags,

B_i – untere Schranke für den Beginn des i-ten Fertigungsauftrags,

E_i – obere Schranke für den Abschluss des i-ten Fertigungsauftrags,

$F_{\max} = \max\limits_{i=1,\ldots,n} T_i$ – den wirklichen Endtermin des Gesamtprojekts,

F_s – die im s-ten Iterationsschritt festgelegte obere Schranke für den Abschluss des Gesamtprojekts.

Die ganzwertigen Funktionen $\mu(\alpha)$ und $t(\alpha)$, $\alpha \in A$ sowie die Größen B_i und E_i, $i = 1, \ldots, n$ sind als bekannt vorausgesetzt. Gesucht werden die Realwerte $x(\alpha)$, $\alpha \in A$, welche dem Endtermin $F_{\max}$ ein globales Minimum liefern.

Offensichtlich müssen die Größen $E_{i,s} = \min\{E_i, F_s\}$, $i = 1, \ldots, n$ als effektive obere Schranken in jedem Iterationsschritt gelten. Falls der Wert B_i bzw. E_i nicht gegeben ist, so wird $B_i = 0$ bzw. $E_{i,s} = F_s$ angenommen. Die Anfangsdaten für die frühest- und spätestmöglichen Starttermine der Arbeitsvorgänge des i-ten Auftrags können nun aus folgenden Gleichungen ermittelt werden:

$$\begin{aligned} &\theta(i,1) = B_i, \\ &\theta(i, j+1) = \theta(i, j) + t(i, j), \qquad j = 1, \ldots, J_i - 1 \end{aligned} \tag{3}$$

$$\begin{aligned} &\tau(i, J_i) = E_{i,s} - t(i, J_i), \\ &\tau(i, j-1) = \tau(i, j) - t(i, j-1), \qquad j = 2, \ldots, J_i \end{aligned} \tag{4}$$

Der wirkliche Starttermin jedes Arbeitsvorgangs O_{ij} soll den folgenden Bedingungen genügen:

$$\theta(i, j) \le x(i, j) \le \tau(i, j) \tag{5}$$

Folglich hat man den Anfangswert F_0 im Iterationsverfahren so zu wählen, dass die Restriktionen (5) ursprünglich kompatibel sind. Danach erfolgen numerischen Umformungen, bei denen die Werte $\theta(i, j)$ steigen und die Werte $\tau(i, j)$ fallen, so dass die gewünschte Kompatibilität in (5) eventuell verletzt wird. In so einem Fall wird der Wert F_0 auf F_1 dermaßen erhöht, dass es im Bereich $F_{\max} \in [F_0, F_1)$ immer noch keine zulässigen Lösungen existieren. Der erste Wert F_s des Schlupfparameters, bei dem eine zulässige Lösung $x(i, j)$, $(i, j) \in A$ gefunden wird, gilt als exakte Lösung der Extremalaufgabe (1) bezüglich der Zielfunktion.

II. Modell der bilinear-booleschen Programmierung

Anhand der in Abschnitt B.I eingeführten Bezeichnungen kann die Minimierungsaufgabe (1) als Problem der bilinear-booleschen Programmierung formuliert werden. Zunächst betrachten wir aber die disjunkten Untermengen der Multiindizes

$$A(k) = \{\alpha \in A:\ \mu(\alpha) = k\}, \quad k = 1, \ldots, m \tag{6}$$

aufgrund deren die Arbeitssysteme den einzelnen Arbeitsvorgängen zugewiesen werden. Um die Paare verschiedener Arbeitsvorgänge in demselben Arbeitssystem zu beschreiben, führen wir noch folgende Mengen ein:

$$\begin{aligned} &P(k) = \{(\alpha, \beta) \in A(k) \times A(k):\ \alpha \neq \beta\}, \quad k = 1, \ldots, m \\ &P = \bigcup_{k=1}^{m} P(k) \end{aligned} \tag{7}$$

Die Reihenfolge der dem k-ten Arbeitssystem zugewiesenen Arbeitsvorgänge kann durch die booleschen Komponenten $\gamma(\alpha_1, \alpha_2)$, $(\alpha_1, \alpha_2) \in P(k)$ der Ecken-Inzidenzmatrix[6] des Graphen partieller Ordnung angegeben werden. Jedem Arbeitssystem entspricht dabei sein eigener Graph partieller Ordnung. Diese Graphen können jeweils in einen vollständigen Graphen partieller Ordnung vereinigt werden, der dann zeitliche Reihenfolge sämtlicher Arbeitsvorgänge beschreibt[7]. Die boolesche Variable $\gamma(\alpha_1, \alpha_2)$ ist gleich Eins, wenn der Arbeitsvorgang mit dem Multiindex $\alpha_2 = (i_2, j_2)$ dem Arbeitsvorgang mit $\alpha_1 = (i_1, j_1)$ folgt, sonst ist $\gamma(\alpha_1, \alpha_2)$ gleich Null. Wenn alle Reihenfolgebeziehungen in jedem Arbeitssystem festgelegt sind[8], so sind dann auch sämtliche boolesche Variablen definiert. Die kontinuierlichen Variablen müssen dann zusammen mit den booleschen Variablen folgende bilineare Ungleichungsrestriktionen erfüllen:

$$\begin{aligned} &\gamma(\alpha_1, \alpha_2)\big(x(\alpha_1) + t(\alpha_1) - x(\alpha_2)\big) + \big(1 - \gamma(\alpha_1, \alpha_2)\big)\big(x(\alpha_2) + t(\alpha_2) - x(\alpha_1)\big) \leq 0, \\ &(\alpha_1, \alpha_2) \in P \end{aligned} \tag{8}$$

Außerdem müssen die ursprünglich festgelegten Reihenfolgen der Arbeitsvorgänge im Rahmen jedes Fertigungsauftrags eingehalten werden, d. h. es muss gelten:

$$x(i, j) + t(i, j) - x(i, j+1) \leq 0, \quad j = 1, \ldots, J_i - 1, \quad i = 1, \ldots, n \tag{9}$$

Wenn die Schranken B_i und E_i $(i = 1, \ldots, n)$ gegeben sind[9], so entstehen die zusätzlichen Restriktionen:

$$\begin{aligned} &B_i \leq x(i, 1), \quad i = 1, \ldots, n \\ &x(i, J_i) + t(i, J_i) \leq E_i, \quad i = 1, \ldots, n \end{aligned} \tag{10}$$

Um ein Problem der mathematischen Programmierung aufzustellen, führen wir die Tschebyschevsche Hilfsvariable η ein, die als gemeinsame obere Schranke für die Endtermine aller Fertigungsaufträge vorgestellt werden kann. Analog zu (9) muss dann gelten:

(11) $$x(i, J_i) + t(i, J_i) - \eta \leq 0, \quad i = 1, \ldots, n$$

Entsprechend dem Kriterium in (1) hat man die eingeführte Hilfsvariable zu minimieren. Somit lässt sich das folgende Modell der bilinear-booleschen Programmierung aufstellen:

(12) $$\left.\begin{array}{ll} \text{Minimiere} & \eta \\ \text{u. d. N.} & \text{Restriktionen (8 - 11)} \end{array}\right\} \quad \text{(BLBP)}$$

Die Minimierung in (12) erfolgt bezüglich η sowie aller x- und γ-Unbekannten.

Die grundsätzliche Komplexität des Problems (12) besteht im Vorliegen der booleschen Unbekannten, so dass unsere Anfangsaufgabe kombinatorischen Charakter erhält. Somit gilt sie als ein diskretes und nichtkonvexes Problem der mathematischen Programmierung. Wenn aber die booleschen Unbekannten ermittelt worden sind, so verwandelt sich (12) in das klassische Problem der linearen Programmierung (bezüglich kontinuierlicher Unbekannten x und η), welches aufgrund seiner einfachen Struktur äußerst effektiv behandelt werden kann.

Man untersucht ferner die Eigenschaften der zulässigen Terminpläne und lässt diejenigen Untermengen von Plänen ausscheiden, die bestimmt keine kompatiblen Lösungen liefern. Dank dieser Strategie erhalten einige boolesche Variablen einen fixierten Wert, so dass sich das ganze Problem für weitere Behandlung wesentlich vereinfacht. Sobald alle booleschen Variablen der optimalen Lösung ermittelt sind, werden die x-Unbekannten aus den einfachen Box-Restriktionen (5) errechnet, da die unteren und oberen Schranken für alle Starttermine im Rahmen unseres Algorithmus streng erhalten werden[10].

C. Vorbereitende Resultate

I. Eigenschaften der zulässigen und optimalen Terminpläne

Es sei bemerkt, dass bei weitem nicht alle Kombinationen von Werten der in (11) eingeführten booleschen Variablen erlaubt sind, doch nur diejenigen, die keine Schleifen im vollständigen Graph partieller Ordnung hervorbringen. Sonst ist die durch das Restriktionssystem (8-11) beschriebene zulässige Menge leer. Wenn man die booleschen Variablen als Steuervariablen erklärt, so sind die oben geschilderten

Bedingungen – aus der Sicht der Kontrolltheorie – einfache Steuerrestriktionen. Die kontinuierlichen Variablen x und η, kann man zugleich als Zustandsvariablen interpretieren. Sie bilden bei jeder fixierten Steuerung eine konvexe Menge in dem Zustandsraum der Terminpläne. Die Ungleichung $\eta \le F_s$ lässt sich bei so einer Auslegung als klassische Zustandsrestriktion betrachten. In diesem Abschnitt untersuchen wir die Eigenschaften der Erreichbarkeitsmengen bezüglich der Variablen x, wobei die Erfüllung der obenstehenden Zustandsrestriktion vorausgesetzt ist und ein Teil der Steuervariablen γ festliegt.

Behauptung 1. Die booleschen Variablen $\gamma^*(\alpha_1,\alpha_2)$, $(\alpha_1,\alpha_2) \in P$ erzeugen die Lösung des Problems (12) genau dann, wenn sie zusammen mit irgendwelchen Werten $x^{**}(\alpha)$, $\alpha \in A$ das Restriktionssystem (8-11) bei $\eta = F^*$ (F^* ist eine ganze Zahl) erfüllen, wohingegen dasselbe Restriktionssystem bei $\eta = F^* - 1$ bereits inkompatibel ist. Die Werte $x^*(\alpha)$, $\alpha \in A$ der entsprechenden optimalen Terminpläne bilden dann eine konvexe Lösungsmenge, welche durch die Restriktionen (8-11) beschrieben ist, in denen $\gamma(\alpha_1,\alpha_2) = \gamma^*(\alpha_1,\alpha_2)$, $(\alpha_1,\alpha_2) \in P$ und $\eta = F^*$ substituiert sind.

Behauptung 1 ergibt eine praktische Regel zur Prüfung, ob der Wert F_s (aus der eingeführten Zustandsrestriktion $\eta \le F_s$) mit minimaler Dauer des Gesamtprojekts F^* zusammenfällt, wenn bekannt ist, dass $F_s \le F^*$ gilt[11].

Behauptung 2. Es sei bekannt, dass der Schlupfparameter F_s nicht größer als die minimale Dauer des Gesamtprojekts ist. Es liege außerdem ein Teil der transitiv abgeschlossenen booleschen Variablen $\gamma_s(\alpha_1,\alpha_2)$, $(\alpha_1,\alpha_2) \in P_0$ ($P_0 \subset P$) vor, deren Werte ermittelt sind und nicht geändert werden können, ohne die Kompatibilität des Restriktionssystems (8-11) bei $\eta = F_s$ zu verletzen. So gilt $F_s = F^*$ genau dann, wenn es solche Werte $\gamma^*(\alpha_1,\alpha_2)$, $(\alpha_1,\alpha_2) \in P \setminus P_0$ gibt, die zusammen mit den bereits ermittelten booleschen Variablen und zusammen mit irgendwelchen Werten $x^{**}(\alpha)$, $\alpha \in A$ das Restriktionssystem (8-11) bei $\eta = F_s$ erfüllen.

Behauptung 2 lässt die Untersuchung eines Steuersystems mit $|P|$ Steuervariablen auf die Untersuchung des Steuersystems mit $|P \setminus P_0|$ Steuervariablen

zurückführen. Die Verringerung der Dimension des Steuerraumes führt offenbar zur wesentlichen Vereinfachung des zu behandelnden Problems[12].

Ein Teil der festgelegten transitiv abgeschlossenen booleschen Steuervariablen lässt sich also durch die Indexmenge $P_0 \subset P$ eindeutig angeben. Wenn die restlichen booleschen Variablen unbestimmt bleiben, so entsteht die im allgemeinen nicht-konvexe *Erreichbarkeitsmenge* $G_s(P_0)$ bezüglich der Variablen x. Diese Menge gilt als Vereinigung der durch die Restriktionen (8-11) bei $\eta = F_s$ beschriebenen konvexen Mengen, wobei die booleschen Variablen $\gamma(\alpha_1, \alpha_2)$, $(\alpha_1, \alpha_2) \in P \setminus P_0$ alle möglichen Werte (0 und 1) durchlaufen. Es ist ein NP-schweres Problem, einen Punkt der Erreichbarkeitsmenge zu finden beziehungsweise die Tatsache festzustellen, das diese Menge leer ist. Im zweiten Fall gibt es jedoch hinreichende Kriterien, die durch die Eigenschaften der konvexen Einhüllenden von $G_s(P_0)$ formuliert werden können. Die Menge

$$(13) \qquad Q_s(P_0) = \{ x(\sigma \in A) : \ \theta(\alpha) \le x(\alpha) \le \tau(\alpha), \ \alpha \in A \}$$

mit willkürlichen ganzzahligen positiven Parametern $\theta(\alpha) \le \tau(\alpha)$, $\alpha \in A$ heißt *konvexe Einhüllende* von $G_s(P_0)$, wenn $G_s(P_0) \subseteq Q_s(P_0)$ gilt. Die konvexen Einhüllenden mit unteren und oberen Schranken für die Starttermine lassen sich öfters leicht konstruieren oder liegen bereits vor.

Behauptung 3. Es wird vorausgesetzt, dass die Menge $Q_s(P_0)$ für den Teil P_0 der festgelegten booleschen Variablen aufgebaut ist. Wenn für zwei Arbeitsvorgänge O_α und O_β mit $(\alpha, \beta) \in P(k)$ folgendes Ungleichungssystem

$$(14) \qquad \begin{cases} \theta(\alpha) + t(\alpha) > \tau(\beta) \\ \theta(\beta) + t(\beta) > \tau(\alpha) \end{cases}$$

mit Parametern aus $Q_s(P_0)$ erfüllt ist, so gilt $G_s(P_0) = \varnothing$.

Behauptung 3 ergibt eine praktische Regel zur Ausscheidung der unzulässigen Steuermengen. Nun untersuchen wir weitere Eigenschaften der Erreichbarkeitsmenge.

Behauptung 4. Es wird vorausgesetzt, dass die Menge $Q_s(P_0)$ für den Teil P_0 der festgelegten booleschen Variablen aufgebaut ist. Wenn für zwei Arbeitsvorgänge O_α und O_β mit $(\alpha, \beta) \in P(k)$ folgendes Ungleichungssystem

$$(15)\qquad \begin{cases} \theta(\alpha)+t(\alpha)\le\tau(\beta) \\ \theta(\beta)+t(\beta)>\tau(\alpha) \end{cases}$$

mit Parametern aus $Q_s(P_0)$ erfüllt ist, so gilt $x(\alpha)\le x(\beta)$ für jeden Punkt der Menge $G_s(P_0)$, falls sie nicht leer ist.

Behauptung 4 eröffnet den Weg, die neuen booleschen Variablen aufgrund der bereits festgelegten booleschen Variablen eindeutig zu bestimmen. Falls (15) erfüllt ist, so muss $\gamma(\alpha,\beta)=1$ und $\gamma(\beta,\alpha)=0$ gesetzt werden, was eine neue Information mit sich bringt, wenn $(\alpha,\beta)\notin P_0$. In diesem Fall bezeichnen wir mit P_0' die transitive Abschließung von $P_0\cup(\alpha,\beta)\cup(\beta,\alpha)$. Derartige Selbsterweiterung von P_0 kann auch aufgrund einer Gruppe von den demselben Arbeitssystem angehörenden Arbeitsvorgängen erfolgen. Die entsprechenden Prüfungskriterien auf die Parameter der konvexen Einhüllenden stimmen mit den Ansätzen in Carlier und Pinson (1989) sowie in Dorndorf u.a. (2000) überein. Ein Analogon zu Behauptung 3 kann dann ebenfalls konstruiert werden. Offensichtlich gilt $G_s(P_0')=G_s(P_0)$. Die Menge $Q_s(P_0')$ kann aber im Vergleich zu $Q_s(P_0)$ verkleinert werden.

Behauptung 5. Es wird vorausgesetzt, dass die Menge $Q_s(P_0)$ für den Teil P_0 der festgelegten booleschen Variablen aufgebaut ist. Die unteren und oberen Schranken der Menge $Q_s(P_0')$ fallen mit denen von $Q_s(P_0)$ zusammen außer vielleicht denjenigen, die aus folgenden Gleichungen neu zu berechen sind:

$$(16)\qquad \begin{aligned} &\tau(i,j')=\min\{\tau(i,j'),\tau(i,j'+1)-t(i,j')\},\qquad j'=1,\ldots,j-1 \\ &\tau(i,j)=\min\{\tau(i,j),\tau(p,r)-t(i,j)\} \end{aligned}$$

$$(17)\qquad \begin{aligned} &\theta(p,r)=\max\{\theta(p,r),\theta(i,j)+t(i,j)\}, \\ &\theta(p,r')=\max\{\theta(p,r'),\theta(p,r'-1)+t(p,r'-1)\},\qquad r'=r+1,\ldots,J_p \end{aligned}$$

wobei $(i,j)=\alpha$ und $(p,r)=\beta$.

Es sei bemerkt, dass die nach Formeln (16), (17) zusammengesetzte Menge $Q_s(P_0')$ selbst niemals leer wird, wenn die Ungleichungen (19), (21-23) von nachfolgenden Behauptungen 6, 7 ursprünglich erfüllt sind. Wenn die Menge jedoch den Bedingungen aus Behauptung 3 (eventuell mit einem anderen k) genügt, dann gilt sofort $G_s(P_0')=G_s(P_0)=\varnothing$.

Behauptung 6. Es wird vorausgesetzt, dass die Menge $Q_s(P_0)$ für den Teil P_0 der festgelegten booleschen Variablen aufgebaut ist, wobei $G_s(P_0)$ nicht leer sei. Wenn der Arbeitsvorgang mit dem Multiindex $\sigma \in A(k)$ der ganzen Gruppe von Arbeitsvorgängen mit $\alpha_j^\sigma \in A(k)$, $j = 1, \ldots, J_\alpha^\sigma$ zuvorkommt, d. h.

$$\text{(18)} \qquad \begin{aligned} &\bigcup_{j=1}^{J_\alpha^\sigma} (\sigma, \alpha_j^\sigma) \in P_0 \quad \text{und} \\ &\gamma(\sigma, \alpha_j^\sigma) = 1, \quad \gamma(\alpha_j^\sigma, \sigma) = 0, \quad j = 1, \ldots, J_\alpha^\sigma \end{aligned}$$

so darf der Wert $\tau(\sigma)$ im folgenden Bereich liegen:

$$\text{(19)} \qquad \tau(\sigma) \le \min\left\{ \max_{j=1,\ldots,J_\alpha^\sigma} \left\{ \tau(\alpha_j^\sigma) + t(\alpha_j^\sigma) \right\} - \sum_{j=1}^{J_\alpha^\sigma} t(\alpha_j^\sigma), \min_{j=1,\ldots,J_\alpha^\sigma} \tau(\alpha_j^\sigma) \right\} - t(\sigma)$$

Die entsprechenden Schranken der benachbarten Arbeitsvorgänge desselben Auftrags können gemäß (16) (mit $(i, j) = \sigma$) berichtigt werden.

Wenn der Arbeitsvorgang mit dem Multiindex $\sigma \in A(k)$ der ganzen Gruppe von Arbeitsvorgängen mit $\beta_j^\sigma \in A(k)$, $j = 1, \ldots, J_\beta^\sigma$ folgt, d. h.

$$\text{(20)} \qquad \begin{aligned} &\bigcup_{j=1}^{J_\beta^\sigma} (\sigma, \beta_j^\sigma) \in P_0 \quad \text{und} \\ &\gamma(\sigma, \beta_j^\sigma) = 0, \quad \gamma(\beta_j^\sigma, \sigma) = 1, \quad j = 1, \ldots, J_\beta^\sigma \end{aligned}$$

so darf der Wert $\theta(\sigma)$ im folgenden Bereich liegen:

$$\text{(21)} \qquad \theta(\sigma) \ge \max\left\{ \min_{j=1,\ldots,J_\beta^\sigma} \theta(\beta_j^\sigma) + \sum_{j=1}^{J_\beta^\sigma} t(\beta_j^\sigma), \max_{j=1,\ldots,J_\beta^\sigma} \left\{ \theta(\beta_j^\sigma) + t(\beta_j^\sigma) \right\} \right\}$$

Die entsprechenden Schranken der benachbarten Arbeitsvorgänge desselben Auftrags können gemäß (17) (mit $(p, r) = \sigma$) berichtigt werden.

Behauptung 7. Es wird vorausgesetzt, dass die Menge $Q_s(P_0)$ für den Teil P_0 der festgelegten booleschen Variablen aufgebaut ist, wobei $G_s(P_0)$ nicht leer sei. Dann dürfen folgende Ungleichungen erfüllt sein:

$$\text{(22)} \qquad \tau(i, j) \le \min_{r=j+1,\ldots,J_i} \left\{ \tau(i, r) - \sum_{r'=j}^{r-1} t(i, r') \right\}, \quad (i, j) \in A$$

$$\text{(23)} \qquad \theta(i, j) \ge \max_{r=1,\ldots,j-1} \left\{ \theta(i, r) + \sum_{r'=r}^{j-1} t(i, r') \right\}, \quad (i, j) \in A$$

Formeln (19), (21-22) ermöglichen, die Einhüllende $Q_s(P_0)$ mit maximal engen Schranken zu konstruieren.

II. Untere Grenzen für die Dauer des Gesamtprojekts

Behauptungen 6, 7 kann man als gewisse Vorbereitungen zur Untersuchung der unteren Grenze für die minimale Dauer des Gesamtprojekts F^* ansehen. Wir nehmen an, dass die frühestmöglichen Starttermine nach Formeln (3) (d. h. unmittelbar aus den Eingangsdaten) berechnet worden sind.

Behauptung 8. Es gilt $F_{low} \le F^*$, wobei

$$F_{low} = \max\left\{ \max_{i=1,\ldots,n} \left\{ \theta(i,1) + \sum_{j=1}^{J_i} t(i,j) \right\}, \max_{k=1,\ldots,m} \left\{ \min_{(i,j)\in A(k)} \theta(i,j) + \sum_{(i,j)\in A(k)} t(i,j) \right\} \right\} \tag{24}$$

Formel (24) widerspiegelt die offene Tatsache, dass die Dauer des Gesamtprojekts nicht kürzer sein kann, als die schnellste Ausführung der Arbeitsvorgänge im Rahmen nur eines Fertigungsauftrags oder eines Arbeitssystems. Die Abschätzung der Ausführungsdauer im Rahmen des k-ten Arbeitssystems kann jedoch verbessert werden. Dazu führen wir aufgrund der bereits festgelegten frühestmöglichen Startterminen folgende Mengen der Multiindizes ein:

$$W^{\xi}(k) = \{(i,j) \in A(k) : \theta(i,j) \ge \xi\}, \quad k = 1,\ldots,m \tag{25}$$

Für den effektiven Wertebereich der ganzzahligen Hilfsvariable ξ jeweils gilt: $\xi \in \{\theta(i,j), (i,j) \in A(k)\}$. Die Indexmenge (25) lässt sich somit durch die Änderung von k und ξ variieren. Sie gibt jedes Mal die Arbeitsvorgänge an, die das k-te Arbeitssystems durchlaufen und nicht früher als ξ gestartet werden können.

Nun wird jedem Arbeitssystem eine charakteristische Zahl zugewiesen, welche zeitliche Spannung in ihm indiziert:

$$\rho(k) = \max_{\xi} \left\{ \min_{(i,j)\in W^{\xi}(k)} \theta(i,j) + \sum_{(i,j)\in W^{\xi}(k)} t(i,j) + \min_{(i,j)\in W^{\xi}(k)} \sum_{j'=j+1}^{J_i} t(i,j') \right\}, \tag{26}$$

$$k = 1,\ldots,m$$

Das Maximum in (26) berechne man nur über diejenigen Werte ξ, die mit den frühestmöglichen Startterminen der Arbeitsvorgänge aus dem k-ten Arbeitssystem zusammenfallen.

Behauptung 9. Es gilt $\overline{F}_{low} \le F^*$, wobei

$$(27) \qquad \overline{F}_{low} = \max\left\{ \max_{i=1,\ldots,n} \left\{ \theta(i,1) + \sum_{j=1}^{J_i} t(i,j) \right\}, \max_{k=1,\ldots,m} \rho(k) \right\}$$

Die in (27) definierte untere Grenze $\overline{F}_{low}$ ist immer besser als diejenige in (25). Die Größe F_{low} lässt sich aber etwas leichter berechnen. Noch größere Werte der zeitlichen Spannungen (und damit auch bessere untere Grenze) können durch die anschließende Maximierung über alle möglichen Teilmengen von $A(k)$ erreicht werden. Wie Carlier (1982) gezeigt hat, erfordern derartige Berechnungen keinen exponentialen Rechenaufwand.

III. Untersuchung der Restriktionen auf Beginn und Abschluss jedes Auftrags

Die unteren Grenzen für die Dauer des Gesamtprojekts können hilfreich sein, insbesondere wenn die oberen Schranken E_i, $i = 1, \ldots, n$ für den Abschluss einzelner Fertigungsaufträge gegeben sind. Es kann eine präventive Analyse dieser Schranken durchgeführt werden, bevor komplexe Suchalgorithmen über die Datenmenge gestartet sind. Falls $\max\limits_{i=1,\ldots,n} E_i < \overline{F}_{low}$, so hat die Aufgabe sowieso keine Lösung. Mindestens eine der oberen Schranken muss dann auf den Wert $\overline{F}_{low}$ erhöht werden.

Zur präventiven Analyse der vorliegenden oberen Schranken können auch die spätestmöglichen Starttermine ausgenutzt werden, deren Werte sich nach Formeln (4) mit $E_{i,0}$ und $F_0 = \overline{F}_{low}$ errechnen lassen. Analog zu (25) führen wir nun folgende Mengen der Multiindizes ein:

$$(28) \qquad V^{\xi}(k) = \left\{ (i,j) \in A(k) : \ \tau(i,j) + t(i,j) \le \xi \right\}, \quad k = 1, \ldots, m$$

Jede Menge gibt die Arbeitsvorgänge an, die das k-te Arbeitssystems durchlaufen und nicht später als ξ beendet werden können.

Behauptung 10. Wenn der Wert $\xi \in \left\{ \tau(i,j) + t(i,j), (i,j) \in A(k) \right\}$ für das k-te Arbeitssystem folgende Ungleichung erfüllt:

$$(29) \qquad \Delta(\xi) = \min_{(i,j) \in V^{\xi}(k)} \theta(i,j) + \sum_{(i,j) \in V^{\xi}(k)} t(i,j) \ - \xi > 0 \,,$$

so muss mindestens eine der Größen $E_{i,0}$ um $\Delta(\xi)$ erhöht werden, wobei die Nummern i den Bedingungen: $\tau(i,j) + t(i,j) = \xi$ und $(i,j) \in A(k)$ genügen. Sonst gilt $G_0 = \varnothing$.

Die Größe $E_{i,0}$ kann entweder durch die Erhöhung der unteren Grenze $F_0 = \overline{F}_{low}$ oder aufgrund der Lockerung der entsprechenden oberen Schranke E_i geändert werden. Der Fall $E_{i,0} = E_i$ für alle Nummern i aus Behauptung 10 bedeutet, dass die gestellte Aufgabe keine Lösung hat. Mindestens eine der aufgedeckten oberen Schranken muss gelockert werden.

Die unteren Schranken B_i, $i = 1, \ldots, n$ können auf ähnliche Art und Weise untersucht werden, d. h. man formuliert Behauptung 10 bezüglich der Mengen $W^{\xi}(k)$.

D. Basisalgorithmen

I. Ermittlung einiger booleschen Variablen im Rahmen eines Arbeitssystems

Wir setzen voraus, dass alle Größen $\rho(k)$, $k = 1, \ldots, m$ frühzeitig berechnet sind. Dann ordnen wir die Arbeitssysteme entsprechend der absteigenden Reihenfolge dieser Größen an, d. h. es gilt: $\rho(\kappa_k) \geq \rho(\kappa_{k+1})$, $k = 1, \ldots, m-1$. Für beliebiges k betrachten wir den folgenden Algorithmus:

Algorithmus 1(k). Der Kenner z_{A1} wird auf Null gesetzt.

Schritt 1. **Für alle Paare der Arbeitsvorgänge (O_α, O_β), $(\alpha, \beta) \in P(\kappa_k)$ werden die Bedingungen (14-15) geprüft.**

Wenn die Ungleichungen (14) gültig sind, so wird $z_{A1} := 2$, $l := l + 1$ und $\Delta_{s,l} := \min\{\theta(\beta) + t(\beta) - \tau(\alpha), \theta(\alpha) + t(\alpha) - \tau(\beta)\}$ gesetzt. Dabei ist Algorithmus 1 zu Ende.

Wenn $\alpha = (i, j)$ und $\beta = (p, r)$ die Ungleichungen (15) erfüllen, mit $(\alpha, \beta) \notin P_0$, so wird $\gamma(\alpha, \beta) := 1$, $\gamma(\beta, \alpha) := 0$ festgelegt und die Menge P_0 von Paaren der Multiindizes wird durch die transitive Abschließung von $P_0 \cup (\alpha, \beta) \cup (\beta, \alpha)$ ersetzt. In diesem Fall werden auch $l := l + 1$ und $\Delta_{s,l} := \theta(p, r) + t(p, r) - \tau(i, j)$ gespeichert.

Es wird $z_{A1} := 1$ gesetzt, falls entweder $\tau(i, j) + t(i, j) > \tau(p, r)$ oder $\theta(i, j) + t(i, j) > \theta(p, r)$. Dann werden die Werte $\tau(i, j)$ und $\theta(p, r)$ sowie die entsprechenden Schranken der benachbarten Arbeitsvorgänge desselben Auftrags nach Formeln (16-17) korrigiert.

Schritt 2. **Für alle Arbeitsvorgänge O_σ, $\sigma \in A(\kappa_k)$ werden bereits festgelegte Nachfolger und Vorgänger aufgrund der vorliegenden Menge P_0 gefunden.**

Wenn der Arbeitsvorgang mit dem Multiindex σ der ganzen Gruppe von Arbeitsvorgängen mit α_j^σ, $j = 1, \ldots, J_\alpha^\sigma$ zuvorkommt, so wird der Wert $\tau(\sigma)$ entsprechend Formel (19) eventuell verkleinert. Die entsprechenden Schranken der benachbarten Arbeitsvorgänge desselben Auftrags werden gemäß (16) berichtigt.

Wenn der Arbeitsvorgang mit dem Multiindex σ der ganzen Gruppe von Arbeitsvorgängen mit β_j^σ, $j = 1, \ldots, J_\beta^\sigma$ folgt, so wird der Wert $\theta(\sigma)$ entsprechend Formel (21) eventuell vergrößert. Die entsprechenden Schranken der benachbarten Arbeitsvorgänge desselben Auftrags werden gemäß (17) berichtigt.

Wenn tatsächliche Schrankenänderungen in Schritt 2 stattgefunden haben, so wird $z_{Al} := 1$ gesetzt. Damit endet Algorithmus 1.

Nach Ablauf des Algorithmus 1 wird entweder nichts geändert (dann endet er mit $z_{Al} = 0$), oder die neue Menge Q_s aufgebaut, d. h. die neuen unteren und oberen Schranken für die Starttermine berechnet ($z_{Al} = 1$), oder die Tatsache $G_s = \varnothing$ festgestellt ($z_{Al} = 2$). Im Laufe von Algorithmus 1 werden der Teilmenge $P_0 \subset P$, die bereits festgelegte boolesche Variablen angibt, zusätzliche neue Paare der Multiindizes hinzugefügt. Dadurch erhöht sich automatisch die Ordnung der Arbeitsvorgänge im κ_k-ten Arbeitssystem[13]. Parallel dazu werden die Größen $\Delta_{s,l}$, $l = 1, 2, \ldots$ gesammelt, mit denen später die untere Grenze von $F_{\max}$ für entgegengesetzten Wert der vorerst festgelegten booleschen Variable berechnet werden kann.

II. Zerlegung der Menge zulässiger Terminpläne in die Untermengen

Man kann Algorithmus 1(k) mehrmals bei fixiertem k laufen lassen, bis er den Kenner $z_{Al} = 0$ ausgibt, dann kann eine Schleife über $k = 1, \ldots, m$ ausgeführt werden, wobei die Ermittlung der lokalen Ordnung von den Arbeitssystemen mit größter Spannung angefangen wird. Die große zeitliche Spannung soll die Entwicklung der Menge P_0 eher begünstigen. Trotz der geschilderten Berechnungsstrategie kann man niemals garantieren, dass die Teilmenge P_0 bis die ganze Menge P ausgebaut wird, das heißt alle booleschen Variablen festgelegt werden. Es werden eventuell einige Gruppen der partiell geordneten Arbeitsvorgänge bleiben, deren exakte Reihenfolge sich allein durch Algorithmus 1 nicht ermitteln lässt.

Für die zwei aneinander nicht geordneten Arbeitsvorgänge O_α und O_β mit $(\alpha, \beta) \in P(k)$ ist das folgende Ungleichungssystem

$$(30)\qquad \begin{cases} \theta(\alpha)+t(\alpha)\le\tau(\beta) \\ \theta(\beta)+t(\beta)\le\tau(\alpha) \end{cases}$$

mit Parametern aus $Q_s(P_0)$ erfüllt. Wenn Algorithmus 1 erschöpft ist, so bleibt der Wert $\gamma(\alpha,\beta)$ unbestimmt und man muss die konvexe Einhüllende Q_s auf künstliche Art und Weise verkleinern. Dies kann entweder durch unmittelbare Teilung der Menge im Raum der Zustandsvariablen erfolgen, oder mittelbar, durch die Analyse der Verzweigung der bisher unbestimmten booleschen Variablen, d. h. im Steuerraum. Hiermit wird nur die zweite Berechnungsstrategie erläutert.

Es gilt die noch nicht festgelegten booleschen Variablen nach ihrer aufsteigenden Priorität anzuordnen: $\gamma\big(\alpha(h),\beta(h)\big)$ mit $\big(\alpha(h),\beta(h)\big)\in P\setminus P_0$, $h=1,\dots,H_s$, wobei $H_s\ge 2$ die Gesamtzahl solcher Variablen im s-ten Schritt des Hauptalgorithmus ist. Dabei lassen sich verschiedene heuristische Anordnungsregeln vorschlagen. Jedes Mal werden beide Zuweisungen $\gamma\big(\alpha(H_s),\beta(H_s)\big):=0$ und $\gamma\big(\alpha(H_s),\beta(H_s)\big):=1$ betrachtet, aufgrund deren eine Verzweigung erfolgt. An Stelle der Menge Q_s werden zwei neue eventuell kleinere Mengen hervorgebracht:

$$(31)\qquad \begin{aligned} Q_s^{\gamma=0} = \big\{ & x(\sigma\in A)\in Q: \\ & \max\{\theta(\alpha(h)),\theta(\beta(h))+t(\beta(h))\}\le x(\alpha(h))\le\tau(\alpha(h)), \\ & \theta(\beta(h))\le x(\beta(h))\le\min\{\tau(\beta(h)),\tau(\alpha(h))-t(\beta(h))\}\big\}, \\ Q_s^{\gamma=1} = \big\{ & x(\sigma\in A)\in Q: \\ & \theta(\alpha(h))\le x(\alpha(h))\le\min\{\tau(\alpha(h)),\tau(\beta(h))-t(\alpha(h))\}, \\ & \max\{\theta(\beta(h)),\theta(\alpha(h))+t(\alpha(h))\}\le x(\beta(h))\le\tau(\beta(h))\big\} \end{aligned}$$

mit Parametern aus $Q=Q_s$ bei $h=H_s$. Nach diesem Schritt kann wieder mit Algorithmus 1 verfahren werden, der weitere boolesche Variablen für $Q_s^{\gamma=0}$ und für $Q_s^{\gamma=1}$ automatisch einstellt. Im allgemeinen entsteht zum Schluss ein Binärbaum. Wir nehmen an, dass die konvexen Einhüllenden $Q_s(\nu)$, $\nu=1,\dots,\mathrm{N}_s$ den Endknoten dieses Baumes entsprechen. Dann gilt offensichtlich

$$(32)\qquad G_s\subset\bigcup_{\nu=1}^{\mathrm{N}_s}Q_s(\nu)$$

Wenn nach der Behandlung mit Algorithmus 1 festgestellt wurde, dass jede dieser Einhüllenden nur den leeren Anteil der Erreichbarkeitsmenge enthält, so gilt $G_s=\varnothing$.

Dann muss der Parameter F_s auf $F_{s+1} > F_s$ vergrößert werden, um die Kompatibilität des Restriktionssystems (8-11) bei $\eta = F_{s+1}$ zu erzielen.

Während der Ausführung von Algorithmus 1 und der Festlegung neuer boolescher Variablen sammelt man die Größen $\Delta_{s,l} \geq 1$, $l = 1, 2, \ldots$ aller eventuellen Engpässe.

Behauptung 11. Es gilt $F_{s+1} \leq F^*$, wobei

$$F_{s+1} = F_s + \min_{l=1,\,2,\,\ldots} \Delta_{s,l} \tag{33}$$

Wenn nach der Behandlung mit Algorithmus 1 alle booleschen Variablen für mindestens einen Knoten des Binärbaumes ermittelt worden sind, so entsprechen diese Werte nach Behauptung 2 einer optimalen Steuerung der Starttermine. Es könnte dann kein Terminplan erstellt werden, so dass das Gesamtprojekt kürzer als F_s dauerte.

III. Exaktes Lösungsverfahren zur Ermittlung des globalen Optimums

In vorherigen Abschnitten haben wir sämtliche Vorbereitungen getroffen, um nun den Hauptalgorithmus zusammenstellen zu können. Er basiert auf gradueller Vergrößerung des Parameters F_s in der Zustandsrestriktion $\eta \leq F_s$. Bei jedem Parameterwert wird die Untersuchung der Erreichbarkeitsmenge G_s durchgeführt, solange sich die vorherige Erreichbarkeitsmenge G_{s-1} als leer erweist. Am Anfang des Algorithmus gilt es noch, die einfachsten Kriterien der Lösbarkeit zu prüfen.

Algorithmus 2. Der Kenner z_{A2} wird auf Null gesetzt.

Schritt 1. Es werden die Bedingungen

$$B_i + \sum_{j=1}^{J_i} t(i, j) \leq E_i, \qquad i = 1, \ldots, n$$

geprüft. Falls einige von ihnen verletzt sind, so wird Algorithmus 2 mit der Meldung angehalten, dass die entsprechenden oberen oder unteren Schranken gelockert werden müssen. Sonst hat die Aufgabe keine Lösung.

Schritt 2. Es werden die Werte $\theta(\alpha)$, $\alpha \in A$ nach Formeln (3) berechnet. Auf Grund dieser Werte werden die untere Grenze $\overline{F}_{low}$ und die zeitlichen Spannungen einzelner Arbeitssysteme $\rho(k)$, $k = 1, \ldots, m$ mit Hilfe von (25-27) bestimmt. Die untere Grenze wird als Startwert $F_0 := \overline{F}_{low}$ angenommen. Die Reihenfolge κ_k, $k = 1, \ldots, m$ der absteigenden Spannungen $\rho(\kappa_k)$ wird ermittelt.

Der Zähler äußerer Schleife s wird auf Null gesetzt.

Schritt 3. Es werden die Werte $E_{i,s} = \min\{E_i, F_s\}$, $i = 1, \ldots, n$ initialisiert. Falls die Bedingung $\max\limits_{i=1,\ldots,n} E_{i,s} \geq F_s$ nicht erfüllt ist, so wird Algorithmus 2 mit der Meldung angehalten, dass einige obere oder untere Schranken gelockert werden müssen. Sonst hat die Aufgabe keine Lösung.

Schritt 4. Es werden die Werte $\tau(\alpha)$, $\alpha \in A$ nach Formeln (4) berechnet. Damit wird die erste Einhüllende Q_s konstruiert.

Bei $s = 0$ werden die Bedingungen aus Behauptung 10 geprüft. Wenn nach dieser Prüfung der Startwert vergrößert werden muss, so setze $F_0 := F_0 + \Delta(\xi')$ und gehe zum Schritt 3 zurück. Wenn sich die Schranke B_i oder E_i als zu eng erweist, so wird Algorithmus 2 mit der Meldung angehalten, dass die entsprechende Schranke gelockert werden muss. Sonst hat die Aufgabe keine Lösung.

Die Zähler l und d werden auf Null gesetzt, $Q := Q_s$.

Schritt 5(Q). Es wird die Schleife in $k = 1, \ldots, m$ organisiert. Bei jedem laufenden k wird Algorithmus 1(k) zur Untersuchung der konvexen Einhüllende Q ausgeführt. Wenn dieser Algorithmus mit Kenner $z_{Al} = 2$ endet und $d = 0$ gilt, so geh zu Schritt 8 über. Bei $d > 0$ geh zu Schritt 7 über.

Wenn Algorithmus 1(k) mit Kenner $z_{Al} = 1$ endet und dabei alle booleschen Variablen schon festgelegt sind (d. h. $P_0 = P$), so ist die optimale Lösung gefunden. Dann geh zu Schritt 9 über.

Wenn Algorithmus 1 bei jedem k mit Kenner $z_{Al} = 0$ endet, so geh zu Schritt 6 über, sonst wiederhol Schritt 5, um die Einhüllende Q eventuell weiter zu verkleinern.

Schritt 6. Setze $z_{A2} := 2$, $d := d + 1$. Die noch unbestimmten booleschen Variablen werden gemäß aufsteigender Priorität angeordnet: $\gamma\big(\alpha(h), \beta(h)\big)$, $h = 1, \ldots, H(P_0)$. Setze ferner $h := H(P_0)$. An Stelle der vorliegenden Einhüllende Q werden zwei neue Mengen $Q^{\gamma=0}$ und $Q^{\gamma=1}$ (entsprechend (31) mit Parametern aus Q) konstruiert.

Setze $\gamma\big(\alpha(h), \beta(h)\big) := 1$ und $\gamma\big(\beta(h), \alpha(h)\big) := 0$. Die entsprechende transitive Abschließung von $P_0 \cup \big(\alpha(h), \beta(h)\big) \cup \big(\beta(h), \alpha(h)\big)$ wird als Menge P_d gespeichert.

Setze ferner $\gamma\big(\alpha(h), \beta(h)\big) := 0$ und $\gamma\big(\beta(h), \alpha(h)\big) := 1$. Die andere transitive Abschließung von $P_0 \cup \big(\alpha(h), \beta(h)\big) \cup \big(\beta(h), \alpha(h)\big)$ wird als Menge P_0 gespeichert.

Setze ferner $Q := Q^{\gamma=0}$, $\overline{Q}_d := Q^{\gamma=1}$ und geh zu Schritt 5(Q) zurück.

Schritt 7. **Wenn $z_{A2} = 2$, dann substituiere $z_{A2} := 1$, $P_0 := P_d$, $Q := \overline{Q}_d$, $d := d - 1$ und geh zu Schritt 5(Q) zurück.**

Wenn $z_{A2} = 1$, so substituiere $P_0 := P_d$, $Q := \overline{Q}_d$, $d := d - 1$ und geh zu Schritt 5(Q) zurück.

Schritt 8. **Die Aufgabe mit $\eta \le F_s$ hat keine Lösung. Der Wert wird entsprechend (33) vergrößert. Der Zähler wird geändert: $s := s + 1$. Dann geh zu Schritt 3 zurück.**

Schritt 9. **Die transitiven Graphen voller Ordnung sind nun für jedes Arbeitssystem aufgebaut. Sie entsprechen der optimalen Steuerung der Reihenfolgen der Arbeitsvorgänge und werden durch die entsprechenden Kettengraphen repräsentiert. Die optimalen Starttermine werden durch die vorhandenen frühest- oder spätestmöglichen Starttermine festgelegt: Setze z. B. $x^{**}(\alpha) := \theta(\alpha)$, $\alpha \in A$. Die Ergebnisse werden in Form eines Gantt-Diagramms grafisch dargestellt.**

Der vorgeschlagene Algorithmus liefert immer Antwort, wenn alle oberen Schranken E_i, $i = 1, \ldots, n$ gegeben sind. Wenn nur ein Teil dieser Größen vorliegt und die Möglichkeit besteht, dass die Aufgabe aufgrund der zu engen Beschränkungen unlösbar ist, so sollte man Algorithmus 2 modifizieren. Die Fertigungsaufträge sind zunächst gemäß der aufsteigenden Abschlusstermine anzuordnen, wobei die Aufträge ohne Abschlussbeschränkung als letzte gestellt werden sollen. Dann konstruiere man eine Folge von Hilfsaufgaben kleinerer Dimension. Die erste Hilfsaufgabe muss nur zwei erste Fertigungsaufträge mit ihren oberen Schranken enthalten. Jeder weiteren Hilfsaufgabe wird ein nächster Fertigungsauftrag aus der Reihe hinzugefügt, wobei die Belegung aller m Arbeitssysteme ständig untersucht wird. Algorithmus 2 wird auf jede Hilfsaufgabe der Reihe nach angewendet, das erlaubt die eventuell vorliegende Unlösbarkeit frühzeitig zu entdecken. In den meisten Fällen kann auch die untere Grenze $\overline{F}_{low}$ der nachfolgenden Hilfsaufgabe durch die Lösung der vorherigen deutlich verbessert werden, denn es kann nun ein Maximum zwischen $\overline{F}_{low}$ und dem zuletzt gefundenen Optimalwert für den Start genommen werden.

Die geschilderte Folge der Hilfsaufgaben endet, sobald nur Fertigungsaufträge ohne Abschlussbeschränkung in der Reihe bleiben. Wenn keine Unlösbarkeit bis dahin entdeckt worden ist, so gilt auch die Ausgangsaufgabe als lösbar. Danach muss Algorithmus 2 für diese ganze Aufgabe (mit eventuell verbesserter unterer Grenze) zum letzten Mal gestartet werden. Das globale Optimum wird nun mit Sicherheit gefunden.

Wenn das zu behandelnde Schedulingproblem sehr groß ist, so sollte am Schluss der Berechnungen die Möglichkeit vorbehalten werden, die äußere Schleife frühzeitig mit bestmöglicher Lösung zu verlassen, denn es geht um die Behandlung eines NP-schweren Problems, dessen wirkliche Berechnungsdauer sich im Vorfeld kaum abschätzen lässt. Man kann unter Umständen zur Heuristik wechseln, indem man etwas größere Zuwächse des Parameters F_s erlaubt, als sie in Schritt 8 von Algorithmus 2 vorgesehen sind. Damit verfehlt man möglicherweise das globale Optimum, erzielt aber die angenäherte Lösung mit geringerem Rechenaufwand. Die einzige Voraussetzung dabei ist die Lösbarkeit des Problems, die auf Grund der beschriebenen Folge der Hilfsprobleme vorerst festzustellen ist.

Die Folge der zeitlich beschränkten Fertigungsaufträge und der mit ihnen verbundenen Hilfsaufgaben kann auch anders konstruiert werden. Die Anordnung darf dabei den aufsteigenden Abschlussterminen nicht unbedingt entsprechen. Bei jeder Anordnungsstrategie wird nur der Fakt der Unlösbarkeit möglichst frühzeitig ermittelt sowie die Teilmenge der beschränkten Fertigungsaufträge, deren Schranken gelockert werden müssen. Um wie viel jede Schranke geändert werden muss, bleibt aber offen.

Bei praktischer Behandlung der umfangreichen Schedulingprobleme mit zusätzlichen Restriktionen auf Beginn und Abschluss einiger Aufträge ist die möglichst große Zahl von Varianten der notwendigen Schrankenänderungen im Falle der Unlösbarkeit von Bedeutung. Jede Art der notwendigen Restriktionsverletzung kann nachträglich mit Hilfe des alternativen Optimierungskriteriums (2) bewertet werden. Somit erkennt der Benutzer alle vorhandenen Engpässe, modifiziert seine Eingangsdaten gezielt und steuert sie sukzessiv in Richtung der Lösbarkeit.

Zum Zweck der Ermittlung sämtlicher Engpässe im Falle der Unlösbarkeit kann ein Sonderalgorithmus (ohne Hilfsrestriktion $\eta \le F_s$) vorgeschlagen werden, der Algorithmus 2 jedoch ähnlich ist und allein auf die Arbeitsvorgänge aus den zeitlich beschränkten Fertigungsaufträgen angewendet wird. Die vollständigen Varianten werden teils aufgrund Algorithmus 1, teils aufgrund des eventuell auftretenden Binärbaumes zusammengestellt. Wenn nur die im Sinne (2) kleinsten Engpässe zu ermitteln sind, so gilt es ein neues Branch-and-Bound Verfahren für die Probleme vom Typ $J||\Sigma w_i L_i$ zu entwickeln.

E. Illustrative und numerische Beispiele

Das Job-Shop-Problem mit zusätzlichen Restriktionen auf Beginn und Abschluss einiger Aufträge kann im Einzelfall recht unterschiedliche Eigenschaften erweisen.

Das in dieser Arbeit beschriebene Verfahren reagiert darauf genauso eigenartig. Zunächst werden zwei illustrative Beispiele, mit und ohne Lösbarkeit des Problems, angeführt.

Beispiel 1. Es werden 3 Fertigungsaufträge betrachtet, welche in 4 verschiedenen Arbeitssystemen ausgeführt werden. Die Anzahl der Arbeitsvorgänge in jedem Fertigungsauftrag entspricht genau der Anzahl der Arbeitssysteme[14]. Sämtliche Eingangsdaten sind in Tabelle 1 dargestellt, wobei die Zuweisung einzelner Arbeitsvorgänge den Arbeitssystemen mit Hilfe der Buchstaben (anstatt Zahlen) festgelegt wird. Die Datentabelle enthält in ihrem rechten Teil die Dauer aller Arbeitsvorgänge und die Schranken für Beginn und Abschluss der entsprechenden Fertigungsaufträge, falls diese Schranken vorhanden sind.

Tab. 1: Eingangsdaten für Beispiel 1

$i=$	$j=1$	$j=2$	$j=3$	$j=4$	$j=1$	$j=2$	$j=3$	$j=4$	B_i	E_i
1	A	B	C	D	30	6	5	12	#	#
2	A	C	B	A	35	28	40	3	#	120
3	D	C	B	D	5	50	4	27	#	130

Bevor Algorithmus 2 auf das vorliegende Problem angewendet wird, sind die Eingangsdaten in bezug auf die Lösbarkeit zu prüfen. Dazu benutzen wir den Sonderalgorithmus (ohne Hilfsrestriktion $\eta \le F_s$), der die Arbeitsvorgänge aus dem zweiten und dritten Fertigungsauftrag analysiert. Für diese Arbeitsvorgänge werden frühest- und spätestmögliche Starttermine nach Formeln (3-4) berechnet. Schon im Laufe von Algorithmus 1 stellen sich folgende Beziehungen heraus:

$$\begin{cases} \theta(2,2) + t(2,2) > \tau(3,2) \\ \theta(3,2) + t(3,2) > \tau(2,2) \end{cases}$$

Dies bedeutet, dass bei den gegebenen Werten E_2 und E_3 keine zulässigen Terminpläne existieren: Entweder muss der zweite Abschlusstermin auf 126, oder der dritte Abschlusstermin auf 144 erhöht werden. Man kann in dem vorliegenden Beispiel leicht prüfen, dass diese notwendigen Bedingungen der Lösbarkeit zugleich auch hinreichend sind. Somit sind sämtliche Engpässe in den Eingangsdaten entdeckt worden.

Beispiel 2. Es werden wieder 3 Fertigungsaufträge und 4 Arbeitssysteme betrachtet. Die Eingangsdaten entsprechen Tabelle 1, dennoch wird nun kein Abschlusstermin dem zweiten Fertigungsauftrag zugewiesen.

Gemäß (24) und (27) gilt: $F_{low} = \max\{106, 101\}$ und $\overline{F}_{low} = \max\{106, 114\}$. Für weitere Berechnungen wird die einfachere untere Grenze $F_{low} = 106$ verwendet. Schon im Laufe von Algorithmus 1 wird festgestellt, dass der Schlupfparameter $F_0 = 106$ auf den nächsten Wert $F_1 = 126$ erhöht werden muss. Im zweiten Schritt von Algorithmus 2 erfolgt zum ersten Mal eine Verzweigung, wobei beide Äste des Binärbaumes zu den Lösungen führen. Die Lösungsmenge in dem vorliegenden Beispiel besteht also aus zwei unterschiedlichen Terminplänen mit zwei verschiedenen Reihenfolgen der Arbeitsvorgänge im Arbeitssystem „B“. In der ersten Lösungsvariante kommt die Operation (1,2) hinter der Operation (3,3); in der zweiten herrscht umgekehrte Ordnung. In Tabelle 2 wird nur die erste Lösungsvariante dargestellt.

Tab. 2: Berechnungswerte in Algorithmus 2 im Falle von Beispiel 2

Kenner (i,j) aller Arbeitsvorgänge		**Schranken für Starttermine am Anfang ($F_0 = 106$)**		**Schranken für Starttermine zum Schluss ($F^* = F_1 = 126$)**		**Optimale Start- und Endtermine**	
		$\theta_0(i,j)$	$\tau_0(i,j)$	$\theta_1(i,j)$	$\tau_1(i,j)$	$x^{**}(i,j)$	$\sigma^{**}(i,j)$
$k = A$	(2,1)	0	0	0	12	0	35
	(1,1)	35	53	35	47	35	65
	(2,4)	103	103	123	123	123	126
$k = B$	(3,3)	55	75	55	73	55	59
	(1,2)	65	83	65	77	65	71
	(2,3)	63	63	83	83	83	123
$k = C$	(3,2)	5	25	5	5	5	55
	(2,2)	35	35	55	55	55	83
	(1,3)	71	89	83	109	83	88
$k = D$	(3,1)	0	20	0	0	0	5
	(3,4)	59	79	59	87	59	83
	(1,4)	76	94	88	114	93	105

Der optimale Terminplan wird in Abbildung 2 mit Hilfe des Gantt-Diagramms veranschaulicht. Unten in derselben Abbildung ist das Ergebnis des heuristischen Lösungsverfahrens aus Domschke u. a. (1997)[15] zu sehen.

Die intensive Betrachtung des Job-Shop Scheduling Problems in der Literatur hat zu der Zusammenstellung umfangreicher und allgemein anerkannter Testdatensätze geführt, mit deren Hilfe es möglich ist, die Effizienz von Verfahren durch Anwendung auf diese Datensätze zu vergleichen; siehe z. B. Jain und Meeran (1999). Zunächst wurde die Funktionalität des entwickelten Lösungsalgorithmus anhand klassischer Testaufgaben FT6x6 und FT10x10 aus Fischer und Thompson (1963) geprüft. Danach wurden von uns zwei andere Testaufgaben DMU1.1 bzw. DMU2.1 mit zusätzlichen Restriktionen auf Abschluss aller Fertigungsaufträge konstruiert und numerisch gelöst. Die Testaufgaben stammen aus den Datensätzen von Demirkol, Mehta, Uzsoy (1996),

Abb. 1: Gant-Diagramme für exakte Lösungen der Job-Shop Probleme DMU 20x15

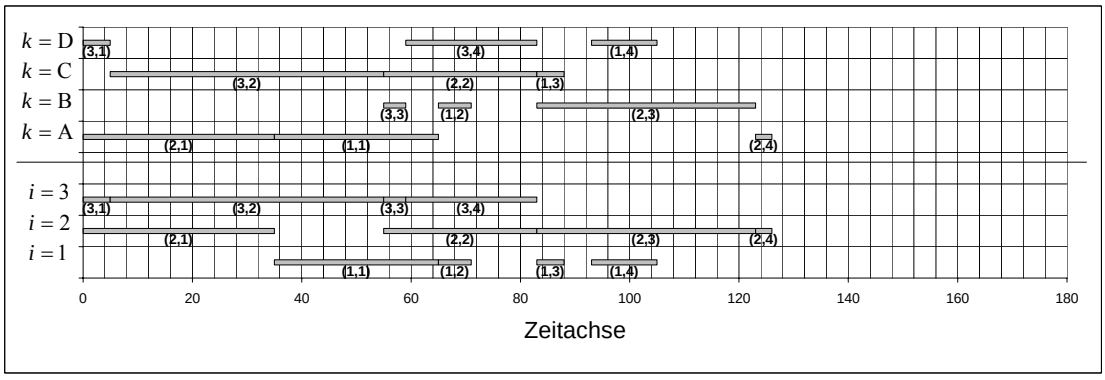
generiert für zwei verschiedene Problemklassen $J||Cmax$ bzw. $J|2sets|Cmax$. Die Testaufgabe DMU1.1 wurde mit den Werten $\|E_i\|$= (2720, 2300, 1095, 2700, 3000, 2005, 2000, 3330, 1440, 3800, 3600, 2200, 4500, 2600, 1645, 1825, 1900, 2190, 3305, 1340) und die Testaufgabe DMU2.1 mit den Werten $\|E_i\|$= (3235, 4175, 2085, 5000, 2880, 1370, 4425, 2285, 3755, 1910, 2600, 3530, 1635, 3745, 2555, 3200, 1075, 1405, 2805, 1730) ausgebaut. Obwohl die Datensätze von Demirkol sehr komplexe Schedulingprobleme bilden und bisher überwiegend zur Testung nur heuristischer Verfahren benutzt wurden, hat unser Algorithmus die beiden Tests gut überstanden, wie es aus Tabelle 3 hervorgeht.

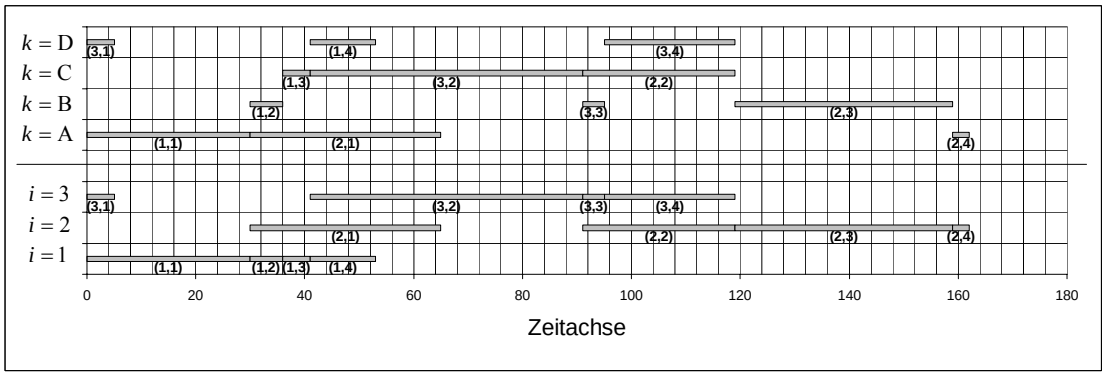
Tab. 3: Berechnungsdauer für die zwei letzten Schritte des Algorithmus

Datensatz	n	m	$H^*_{\max}$	LB	F^*	CPU, sec
FT6x6	6	6	57	52	55	1 (1*)
FT10x10	10	10	434	808	930	159 (6*)
DMU1.1	20	15	549	2363	4426	548 (1*)
DMU2.1	20	15	817	2839	4994	31 (12*)

Die entsprechenden optimalen Terminpläne werden in Abbildung 1 mit Hilfe der Gantt-Diagramme veranschaulicht. Darauf kann man sehen, wie die Blockstruktur der exakten Lösung von DMU2.1 durch die anwesenden Zusatzrestriktionen zerstört wird.

Abb. 2: Gantt-Diagramme für exakte und für angenäherte Lösung von Beispiel 2





Anmerkungen

1 Vgl. Zäpfel (1996) und Zack (1978).

2 Als zugehörige Literaturstellen können folgende Beiträge genannt werden: Giffler und Thompson (1960), Hoch (1973), Siegel (1974), Zack (1978), Brucker und Jurisch (1993), Lawler u. a. (1993), Blazewicz u. a. (1993), Brucker (1995), Zäpfel (1996). Die neusten Literaturhinweise findet man auch in Blazewicz u. a. (1996) sowie in Jain und Meeran (1999).

3 Siehe Carlier und Pinson (1989) sowie Krell (1958).

4 Lese dazu: Brucker und Jurisch (1993) sowie Brucker (1995), Charlton und Death (1970), Ashour und Hieremath (1973), Barker und McMahon (1985), Carlier und Pinson (1989), Applegate und Cook (1991).

5 Siehe: Zäpfel (1996), Brucker und Jurisch (1993), Lawler u. a. (1993), Blazewicz u. a. (1993), Akers und Friedman (1955), Giffler und Thompson (1960), Charlton und Death (1970), Ashour und Hieremath (1973), Haupt (1989), Storer u. a. (1992), Books und White (1965), Bowman (1959), Manne (1960), Dantzig (1960).

6 Mehr dazu lese in der Monographie von Ore (1962).

7 Der vollständige Graph partieller Ordnung wird zum Beispiel in Brucker (1995) ausführlicher diskutiert.

8 Dies bedeutet, dass sich alle Graphen partieller Ordnung in die transitiven Graphen voller Ordnung verwandelt haben.

9 Wenn die untere Schranke für den Beginn irgendeines Fertigungsauftrags nicht gegeben ist, so wird sie durch Null ersetzt und der erste Teil von Restriktionen (10) bleibt bestehen.

10 D. h. zwei Sonderlösungen, mit frühest- und spätestmöglichen Startterminen, können aus der Lösungsmenge des Optimierungsproblems leicht rekonstruiert werden.

11 Der optimale Wert selbst darf dabei unbekannt bleiben.

12 Es sei bemerkt, dass die ursprüngliche Anzahl der Steuervariablen von der Anzahl der Zustandsvariablen quadratisch abhängt, wohingegen die optimale Steuerung nur mit recht wenigen Werten definiert werden kann. Sie sind sogar um m weniger als die Anzahl der Zustandsvariablen. Die restlichen Komponenten der optimalen Steuerung kann man aus dem einfachen Prinzip der Transitivität reproduzieren.

13 Dies entspricht der Berechnungsstrategie von Carlier und Pinson (1989) namens *Immediate selection*.

14 Diese Bedingung entspricht den klassischen Voraussetzungen für die Job-Shop Probleme. Siehe z. B. in Fisher und Thompson (1963). In der Praxis dürfen die Dimensionen jedoch recht beliebig ausfallen.

15 Bidirektionales Einplanen, Punkt 5.6.3, S. 406 – Heuristische Verfahren zur Minimierung der Zykluszeit.

Literatur

[1] Akers, S. B. und Friedman, J. (1955) A non-numerical approach to production scheduling problems, *Operations Research 3*, 429-442

[2] Applegate, D. und Cook, W. (1991) Acomputational study of the job-shop scheduling problem, *ORSA Journal on Computing 3*, 149-156

[3] Ashour, S. und Hieremath, S. R. (1973) A branch-and-bound approach to the job-shop scheduling problem, *Int. J. Product. Res. 11(1)*, 47-58

[4] Balas, E., Lancia, G., Serafini, P. und Vazacopoulos, A. (1998) Job-shop scheduling with deadlines, *J. Combinatorial Optimization 1(4)*, 324-353

[5] Barker, J. R. und McMahon, G. B. (1985) Scheduling the general job-shop, *Management Science 31*, 594-598

[6] Blazewicz, J., Domschke W. und Pesch E. (1996) The job shop scheduling problem: Conventional and new solution techniques. *Eur. J. Oper. Res. 93*, 1-33

[7] Blazewicz, J., Ecker, K., Schmidt, G. und Weglanz, J. (1993) *Scheduling in Computer and Manufacturing System*, Berlin et al

[8] Bowman, E. H. (1959) The schedule-sequencing problem, *Operations Research 7(5)*, 621-624

[9] Brooks, G. H. und White, C. R. (1965) An algorithm for finding optimal or near-optimal solutions to the production scheduling problem, *J. Ind. Engng. 1*, 123-136

[10] Brucker, P. (1995) *Scheduling Algorithms*, Springer-Verlag: Berlin, Heidelberg und New York

[11] Brucker, P. und Jurisch, B. (1993) A new lower bound for the job-shop scheduling problem, *Eur. J. Oper. Res. 64(2)*, 156-167

[12] Carlier, J. (1982) The one-machine sequencing problem, *Eur. J. Oper. Res. 11*, 42-47

[13] Carlier, J. und Pinson, E. (1989) An algorithm for solving the job-shop problem, *Management Science 35(2)*, 164-176

[14] Charlton, J. und Death, C. C. (1970) A method of solution for general machine-scheduling problems, *Operations Research 18*, 689-707

[15] Dantzig, G. B. (1960) A machine-job scheduling model, *Management Science 6(2)*, 191-196

[16] Demirkol, E., Mehta, S. und Uzsoy R. (1996) Benchmarking for shop scheduling problems, *Research Memorandum No. 96-4*, Purdue University

[17] Domschke, W., Scholl, A. und Voß, St. (1997) *Produktionsplanung. Ablauf-organisatorische Aspekte*, Springer-Verlag: Berlin, Heidelberg

[18] Dorndorf, U., Pesch E. und Phan Huy, T. (2000) Constraint propagation techniques for disjunctive scheduling problems, *J. Artificial Intelligence 122*, 189-240.

[19] Fischer, H. und Thompson, G. L. (1963) Probabilistic learning combinations of local job-shop scheduling rules, *Industrial Scheduling*, J. F. Math and G. L. Thompson (editors), Prentice-Hall, Englewood Cliffs, NJ

[20] Giffler, B. und Thompson, G. L. (1960) Algorithms for solving production-scheduling problems, *Operations Research 8*, 487-503

[21] Haupt, R. (1989) A survey of priority rule-based scheduling, *OR-Spectrum 11*, 3-16

[22] Hoch, P. (1973) *Betriebswirtschaftliche Methoden und Zielkriterien der Reihenfolgeplanung bei Werkstatt- und Gruppenfertigung*, Frankfurt a.M. und Zürich

[23] Jain, A. S. und Meeran S. (1999) Deterministic job-shop scheduling: Past, present and future. *Eur. J. Oper. Res. 113*, 390-434

[24] Krelle, W. (1958) Ganzzahlige Programmierungen. Theorie und Anwendungen in der Praxis, *Unternenehmenforschung 2*, 161-175

[25] Land, A. H., Laporte, G. und Miliotis, P. (1978) A unified formulation of the machine scheduling problem, *Eur. J. Oper. Res. 2*, 32-35

[26] Lawler, E. L., Lenstra, J. K., Rinnooy Kann, A. H. G. und Shmoys, D. B. (1993) *Sequencing and Scheduling: Algorithms and complexity*, Logistic of Production and Inventory, S.C. Graves et.al (Hrsg), Amsterdam-London, 445-522

[27] Leon, V. J. und Wu, S. D. (1992) On scheduling with ready-times, due-dates and vacations, *Naval Research Logistics 39*, 53-65

[28] Manne, A. S. (1960) On the job-shop scheduling problem, *Operations Research 8(2)*, 219-223

[29] Ore, Ř. (1962) *Theory of graphs*, American Mathematical Society. Colloquium Publications, Vol. 38

[30] Siegel, Th. (1974) *Optimale Maschinenbelegungsplanung*, Betriebswirtschaftliche Studien: Nr. 20, Berlin

[31] Storer, R. H., Wu, S. D. und Vaccari R. (1992) New search spaces for sequencing problems with applications to job scheduling, *Management Science 38,* 1495-1509

[32] Zack, Yu. A. (1978) Certain properties of scheduling theory problems, *Autom. and Remote Control 39*, 99-107

[33] Zäpfel, G. (1996) *Grundzüge des Produktions- und Logistikmanagements*, Verlag Walter de Gruyter, Berlin – New York

[34] Zimmermann, H.-J. (1971) *Netzplantechnik*, Verlag Walter de Gruyter, Berlin – New York

Zusammenfassung

Es wird die Aufgabe der Termin- und Reihenfolgeplanung in Form eines Job-Shop-Problems mit zusätzlichen Restriktionen auf Beginn und Abschluss einiger Jobs betrachtet. Auf Basis festgestellter Eigenschaften von frühest- und spätestmöglichen Startterminen sämtlicher Arbeitsvorgänge gelingt es qualitative Auswertungen von unten für die Gesamtprojektdauer zu erhalten. Dadurch werden schon in den ersten Schritten des Algorithmus die Terminpläne mit notorisch unzulässiger Ordnung der Arbeitsvorgänge ausgeschieden und die Engpässe in der Ressourcenkapazität und/oder in den vorgegebenen Endterminen einzelner Aufträge gefunden. Es werden exakte und approximative Lösungsverfahren entwickelt, die sich auf einer sequenziellen Annäherung „von unten“ an die Menge der zulässigen Lösungen beruhen und imstande sind, in jedem Iterationsschritt die Kompatibilität des vorliegenden Restriktionssystems, welches einen sich iterativ präzisierenden gemeinsamen Endtermin für alle existierende Aufträge mit einschließt, numerisch effektiv zu prüfen. Bei sukzessiver Lockerung des gemeinsamen Endtermins soll die zuerst entdeckte kompatible Lösung zugleich als optimal gelten.

Summary

The temporal and capacity planning in the form of job-shop problem with additional restrictions on the initiation and completion of several jobs is considered. On the base of determined properties of the earliest and latest possible times for the initiation of scheduling operations we succeed in obtaining of effective low bounds for the project duration. Using these we can cut off the plans with wittingly inadmissible order of operations from the first steps of the algorithm, and determine conflicts in the resources and/or in the prescribed periods for completion of certain jobs. Accurate and approximate numerical methods are developed, based on the successive approximations to the feasible set "from below", which permit on each iteration to control efficiently the compatibility of the arising constraints, which include the iteratively refined common temporal restriction on the completion of all jobs. The first compatible solution, found in the process of progressive relaxation of the common temporal restriction, has to be optimal.