

Drexl, Andreas; Jordan, Carsten

Working Paper — Digitized Version

Wissensbasierte Produktionsprozeßplanung bei reihenfolgeabhängigen Rüstkosten und -zeiten

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 321

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Drexl, Andreas; Jordan, Carsten (1993) : Wissensbasierte Produktionsprozeßplanung bei reihenfolgeabhängigen Rüstkosten und -zeiten, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 321, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/155401>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Nr. 321

**Wissensbasierte Produktionsprozeß-
planung bei reihenfolgeabhängigen
Rüstkosten und -zeiten**

Drexl, A. und C. Jordan

Juli 1993

Professor Dr. **Andreas Drexl**, Dipl.-Wirtsch.-Ing. **Carsten Jordan**, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel.

1. Einleitung

Im Bereich der Entwicklung elektronischer Leitstände¹ werden derzeit große Anstrengungen unternommen, die Produktionsplanung und -steuerung² wissensbasiert zu unterstützen. Von besonderem Interesse sind dabei Konzepte und Implementierungen wissensbasierter Leitstände zur Losgrößen-, Termin- und Ablaufplanung.³

Im Mittelpunkt der vorliegenden Arbeit steht ein Problem der Produktionsprozeßplanung, das durch folgende grundlegenden Annahmen gekennzeichnet ist: Wir betrachten ein einstufiges Produktionssystem. Gefertigt werden mehrere unterschiedliche Produkte. Eine Anlage (Produktiveinheit) stellt die zu belegende Engpaßressource dar. Die Nachfrage nach jedem Produkt ist bekannt. Rüstkosten und/oder Rüstzeiten fallen immer dann an, wenn auf der Anlage ein Produktwechsel stattfindet.

Der Beitrag ist wie folgt aufgebaut: In Abschnitt 2 beschreiben wir den Gegenstand der Produktionsprozeßplanung. Abschnitt 3 ist der Herausarbeitung der Charakteristika des in der vorliegenden Arbeit spezifisch betrachteten Problems gewidmet. Abschnitt 4 behandelt die Grundlagen neuerer logikorientierter Programmiersprachen. In Abschnitt 5 präsentieren wir eine "konzeptionelle" Formulierung dieses Problem; diese ist sehr "kompakt" und für eine Codierung in deklarativen Programmiersprachen besonders geeignet. Abschnitt 6 stellt beispielhafte Ergebnisse vor. Wir beschließen den Beitrag in Abschnitt 7 mit Hinweisen auf weiterführende Forschungsarbeiten.

2. Gegenstand der Produktionsprozeßplanung

Gegenstand der Produktionsprozeß- oder Produktionsdurchführungsplanung ist die zeitliche, mengenmäßige und räumliche Planung des Produktionsvollzugs.⁴ Sie erfolgt auf Stunden-, Tage- oder Wochenbasis.

Auf der Grundlage der durch ein aktuelles Produktionsprogramm vorliegenden Produktionsmengen sind konkrete Fertigungsaufträge zu bestimmen. Ein Fertigungsauftrag legt fest, welche Menge von welchem Produkt in welchem Zeitraum hergestellt werden soll. Folgende Entscheidungen sind dabei zu treffen:⁵ Für welche Produkte sind Fertigungsaufträge zu bilden? Auf welcher Anlage sollen die Fertigungsaufträge bearbeitet werden? In welcher Reihenfolge sind Fertigungsaufträge, die dieselben Anlagen benötigen, einzuplanen? Unter

¹ Eine aktuelle Übersicht über den Stand der Anwendung von Leitständen in der Industrie geben **Stadtler / Wilhelm** (1993).

² Zu konventionellen ergebnisorientierten Systemen der Produktionsplanung und -steuerung (PPS-Systeme) vgl. z.B. **Hahn / Laßmann** (1989), **Mertens** (1991) sowie **Glaser et al.** (1992). Zur Kritik an konventionellen PPS-Systemen sowie zu den konzeptionellen Grundlagen kapazitätsorientierter PPS-Systeme vgl. vor allem **Drexel et al.** (1993).

³ Vgl. hierzu z.B. **Adelsberger et al.** (1992).

⁴ Vgl. hierzu auch **Zäpfel** (1982), **Hoitsch** (1985), **Corsten** (1990), **Hansmann** (1992) sowie **Günther** (1993).

⁵ Vgl. hierzu z.B. **Domschke et al.** (1993) sowie **Drexel et al.** (1993).

Berücksichtigung der erforderlichen Bearbeitungsdauern sind die Fertigungsaufträge detailliert zeitlich festzulegen.

Die im Rahmen der Produktionsprozeßplanung zu treffenden Entscheidungen sind an sich wegen der Interdependenzen in der Regel simultan zu treffen. Wegen der Komplexität der damit verbundenen Entscheidungsprobleme geht man meist sukzessiv nach folgendem Schema vor:

- *Losgrößenbestimmung*: Unter Vernachlässigung von Kapazitätsrestriktionen wird festgelegt, ob hinsichtlich einer bestimmten Zielsetzung (z.B. minimale Rüst- und Lagerungskosten) eine Zusammenfassung von Fertigungsaufträgen zu Losen sinnvoll ist. Ein Los entspricht dabei einer Anzahl von Fertigungsaufträgen, die auf einer Anlage unmittelbar hintereinander ohne Rüstvorgänge herzustellen sind.⁶
- *Durchlauf- und Kapazitätsterminierung*: Für jeden Fertigungsauftrag sind früheste und späteste Zeitpunkte für die Bearbeitung im Hinblick auf vereinbarte Liefertermine zu ermitteln. In der Praxis wird zunächst eine reine Durchlaufterminierung (Terminplanung) unter Vernachlässigung von Kapazitäts Gesichtspunkten durchgeführt. Unter bzw. Überdeckungen zwischen Kapazitätsangebot und -nachfrage werden anschließend (sofern möglich) mit Hilfe einfacher Verschiebungsverfahren ausgeglichen.⁷
- *Reihenfolgeplanung und Feinterminierung*: Es sind Reihenfolgen für die Bearbeitung der Fertigungsaufträge zu bilden (sequencing); ferner ist eine detaillierte zeitliche Verteilung der Fertigungsaufträge auf die einzelnen Anlagen vorzunehmen. Im Rahmen der Maschinenbelegungsplanung (scheduling) werden beide Entscheidungen zusammen getroffen.⁸

Im Anschluß an die Durchlauf- und Kapazitätsterminierung erfolgt in der Regel die Auftragsfreigabe. Die nachgeordneten Entscheidungen der Reihenfolgeplanung und Feinterminierung werden dementsprechend häufig dem Bereich der Produktionssteuerung zugeordnet.

Art und Schwierigkeitsgrad der zu lösenden Entscheidungsprobleme werden wesentlich von der Art des Produktionssystems, insbesondere vom Repetitionstyp (Einzel-, Serien-/Sorten-, Massenfertigung) sowie vom Anordnungstyp (Werkstatt-, Fließfertigung) bestimmt. Dabei ist die Produktionsprozeßplanung hierarchisch in den Kontext der Produktionsplanung und -steuerung so einzuordnen, daß auf allen Stufen die relevanten Kapazitäten beachtet werden.⁹

Die in der vorliegenden Arbeit betrachtete Problemstellung integriert Aspekte der Losgrößenplanung, der Durchlauf- und Kapazitätsterminierung sowie der Reihenfolgeplanung

⁶ Modelle und Methoden zur Losgrößenplanung findet man in **Schneeweiß** (1981), **Tempelmeier** (1992) sowie **Domschke et al.** (1993); vgl. hierzu auch die aktuelle und umfassende Übersicht in **Kuik et al.** (1993).

⁷ Zur Termin- und Kapazitätsplanung vgl. z.B. **Domschke / Drexel** (1991) sowie **Günther** (1992).

⁸ Zur Maschinenbelegungsplanung vgl. insbesondere **Blazewicz et al.** (1993), **Kistner / Steven** (1993) sowie **Domschke et al.** (1993).

⁹ Vgl. hierzu erneut **Domschke et al.** (1993) sowie **Drexel et al.** (1993).

und Feinterminierung. Diese Aufgaben werden in einem Modell (einer Wissensbasis) zusammengefaßt und simultan (durch Inferenz) gelöst. Bevor wir in Abschnitt 5 das Modell darstellen, ordnen wir die Problemstellung zunächst detailliert in den Kontext der Produktionsprozeßplanung ein.

3. Einordnung der Problemstellung

In der Literatur werden eine Vielzahl von Problemen der Produktionsprozeßplanung betrachtet. Mit Hilfe der folgenden *Klassifikationskriterien*¹⁰ läßt sich die hier behandelte Problemstellung präzise in den Kontext der Produktionsprozeßplanung einordnen:

- Informationsgrad: Wir gehen davon aus, daß alle erforderlichen Daten bzw. Parameter mit Sicherheit bekannt sind, betrachten also ein *deterministisches Problem*.
- Zeitliche Entwicklung: Wir betrachten ein *diskretes dynamisches Modell*, d.h. die Nachfrage schwankt im Zeitablauf und tritt an diskreten Zeitpunkten auf.
- Planungszeitraum: Es liegt ein *endlicher Planungszeitraum* vor. In den meisten verwandten Problemstellungen muß vom Planer eine (nicht ganz einfache, jedoch hinsichtlich der Güte der erzielbaren Ergebnisse sehr sensitive¹¹) Einteilung des Planungszeitraumes auf Stunden-, Tages- oder Wochenbasis vorgenommen werden. Hier liegt die Besonderheit vor, daß – auf einer a priori beliebigen zeitlichen Skala – der Zeitpunkt des Auftretens der Nachfrage angegeben werden kann.¹²
- Anzahl der zu fertigenden Produkte: Wir behandeln den *Mehrproduktfall*, bei dem vor allem Losgrößen- und Terminierungsentscheidungen simultan für mehrere Produkte zu treffen sind.
- Anzahl der Dispositionsstufen: Betrachtet wird ein *einstufiger Produktionsprozeß*, bei dem die Produkte nicht durch eine Erzeugnis- und Prozeßstruktur miteinander verknüpft sind.
- Kapazitäten: Wir behandeln den Fall *eines Engpasses*. Die Verfügbarkeit der und die Nachfrage nach der einzigen Engpaßressource wird in Zeiteinheiten gemessen.
- Fehlmengen: Wir behandeln den Fall *ohne Fehlmengen*; es muß also gewährleistet werden, daß der Bedarf in jedem Zeitpunkt durch verfügbare Mengen gedeckt werden kann.
- Kosten: Vor der Auflage eines Produktionsloses sind Rüstvorgänge (Einstellung der Anlage, Bereitstellung bzw. Wechsel von Werkzeugen usw.) durchzuführen. Wir betrachten den Fall *reihenfolgeabhängiger Rüstkosten* (und -zeiten, siehe unten). Ferner sind lineare variable *Lagerungskosten* zu betrachten. Sonstige Kosten (Fehlmengenkosten, variable Produktionskosten, etc.) sind nicht entscheidungsrelevant.

¹⁰ In Anlehnung an **Domschke** et al. (1993, S. 64 ff.).

¹¹ Siehe hierzu vor allem **Haase** (1993).

¹² Im Gegensatz dazu wird bei der dynamischen Losgrößenplanung üblicherweise ein Periodenindex t verwendet.

- Rüstzeiten: Die auf der Anlage bei einem Produktwechsel anfallenden *Rüstzeiten* sind *reihenfolgeabhängig*.
- Fertigungsgeschwindigkeit: Wir behandeln den Fall *endlicher Fertigungsgeschwindigkeit* mit gleichbleibender Bearbeitungsdauer pro Mengeneinheit eines Fertigungsauftrages (keine Lerneffekte).
- Art der Produktweitergabe: Wir betrachten den Fall *geschlossener Produktweitergabe* von Fertigungsaufträgen (in ein Auslieferungslager); es stehen also nur "ganze" Fertigungsaufträge eines Loses für die Befriedigung der Nachfrage zur Verfügung.
- Zielsetzung: Wir gehen vom Ziel der *Minimierung der Summe aus (reihenfolgeabhängigen) Rüst- und Lagerungskosten* aus. Desweiteren betrachten wir die Frage, ob für gegebene Parameter und Daten unter Vernachlässigung der Zielfunktion überhaupt eine zulässige Lösung existiert (Constraint Satisfaction Problem¹³).

Üblicherweise werden in Modellen der Produktionsprozeßplanung Rüstzeiten vernachlässigt und lediglich Rüstkosten betrachtet. Dies geschieht vor dem Hintergrund, daß die Erfassung von Rüstzeiten modelltechnisch und mehr noch verfahrensmäßig ausgesprochen schwierig ist. Wir beziehen im folgenden neben Rüstkosten auch Rüstzeiten ein, und zwar beides in der allgemeinen (und damit theoretisch überlegenen) reihenfolgeabhängigen Form.

4. Grundlagen neuerer logikorientierter KI-Sprachen

Wissensbasierte Systeme bestehen aus zwei grundlegenden Komponenten, nämlich der Wissensbasis und dem Inferenzmechanismus. Wissen und Inferenz bilden das tragende Element lernender Systeme, wobei insbesondere die Inferenz den qualitativen Unterschied gegenüber herkömmlichen Systemen darstellt.¹⁴

Die Wissensbasis enthält deklaratives Wissen in Form von Regeln, Fakten oder Restriktionen, welches relevant für das zu lösende Problem ist.¹⁵ Die Wissensbasis ist beschreibend, ohne dabei festzulegen, wann und wie welches Wissen zu nutzen ist.¹⁶ Die Vorteile sind offensichtlich: Die Entwicklung und Änderung wissensbasierter Systeme ist wesentlich einfacher als die konventioneller (prozeduraler) Systeme.

Das Wissen in einer Wissensbasis sollte in unabhängige und dem Benutzer verständliche Teile aufgespalten sein. Zum Zeitpunkt der Problemlösung sollten sich diese Teile untereinander zur Generierung von neuem Wissen über den zugrundeliegenden Problembereich

¹³ Das Constraint Satisfaction Problem ist eng verwandt mit den in der Entscheidungstheorie behandelten Modellen mit Satisfizierungszielen; vgl. hierzu z.B. **Dinkelbach** (1982).

¹⁴ Inferenz kann ebenfalls als Wissen bezeichnet werden, nämlich als Wissen, wie man aus bereits vorhandenem Wissen zusätzliches Wissen erschließen kann; Inferenz ist also Wissen über die Verarbeitung von Wissen, d.h. Meta-Wissen.

¹⁵ Unterschiedliche Arten der Darstellung von Wissen findet man z.B. in **Kurbel** (1989).

¹⁶ Im Gegensatz dazu steht etwa das prozedurale Wissen herkömmlicher Computerprogramme.

aktivieren bzw. deaktivieren. Der Inferenzmechanismus muß nicht sofort auf einer vollständigen Wissensbasis arbeiten; vielmehr sollte nachträgliches Einfügen von Wissen (etwa weiterer Restriktionen) möglich sein, ohne dabei mögliche Interaktionen mit bereits vorhandenem Wissen berücksichtigen zu müssen.

Relativ kleine Wissensbasen mit einer geringen Anzahl von Regeln oder Restriktionen definieren eine i.d.R. exponentiell mit der Problemgröße anwachsende Zahl potentieller Lösungen; deren Zulässigkeit (Konsistenz) oder gar Optimalität kann endgültig erst durch Enumeration bestimmt werden. Konsistenzprüfungen dienen als Teil der Inferenz¹⁷ zur Beschneidung des Suchbaumes bei Enumerationsverfahren (Branch & Bound) für die Lösung schwieriger (NP-vollständiger) Probleme.

Die Bestimmung zulässiger oder optimaler Lösungen von kombinatorischen Optimierungsproblemen ist bisher beschränkt auf eine implizite Enumeration des gesamten Lösungsraumes unter Einbeziehung verschiedener Strategien,¹⁸ die den Suchraum zwar drastisch beschränken können, jedoch in prozeduralen Programmiersprachen i.d.R. nur problembezogen und mit großem Aufwand zu implementieren sind. Logikorientierte bzw. deklarative Programmiersprachen bauen implizit einen Suchbaum auf, wobei der Programmierer den Aufbau lediglich zu steuern, jedoch nicht selbst zu programmieren hat. Ältere logikorientierte Programmiersprachen (wie z.B. PROLOG) bieten zwar die Möglichkeit einer vergleichsweise einfachen Problemimplementierung (basierend auf einer Wissensbasis in Form von Regeln) und stellen eine generische Suchstrategie (basierend auf einfachen, teilweise benutzergesteuerten Inferenzregeln) bereit. Sie besitzen jedoch als "general problem solver" den Nachteil, i.d.R. nur zur Lösung sehr einfacher Probleme Verwendung finden zu können.

Suche mittels Backtracking gewann an Bedeutung aufgrund der Anwendung in PROLOG und beim Lösen von Constraint Satisfaction Problemen. PROLOG II¹⁹ eröffnete dem Benutzer "klarere Ausdrucksmöglichkeiten" und ebnete den Weg zu heute allgemein als "logic programming with constraints"²⁰ bezeichneten Programmiersprachen. In PROLOG III²¹ besteht die Möglichkeit, Constraints auf verschiedenen Datentypen zu definieren. Dadurch wird die Ausdruckskraft der Sprache erheblich vergrößert und es werden Anwendungen ermöglicht, die in PROLOG nur sehr schwer zu realisieren sind. Neben PROLOG III sind vor allem die constraint-basierten, logikorientierten Programmiersprachen CHIP²² und in jüngster Zeit CHARME²³ bekannt geworden.

¹⁷ Ergänzt durch Methoden wie intelligent, selective bzw. dependency-directed Backtracking, Back-jumping oder Backmarking; vgl. hierzu z.B. **Freuder / Wallace** (1992) sowie **Tate** (1985).

¹⁸ Wie z.B. heuristische Suche.

¹⁹ Vgl. **Pique** (1988).

²⁰ **Cohen** (1990).

²¹ Vgl. insbesondere **Colmerauer** (1990).

²² Constraint Handling in PROLOG; vgl. z.B. **Dincbas et al.** (1990) sowie **Van Hentenryck** (1989).

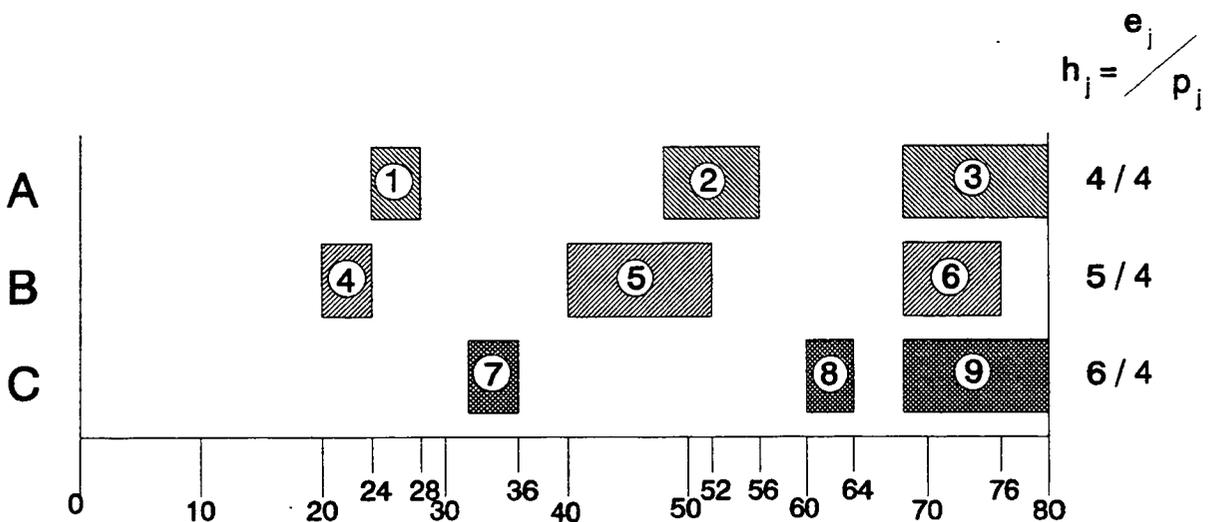
²³ Siehe **BULL** (1991); eine Einführung in die Grundlagen und Möglichkeiten des Arbeitens mit CHARME findet sich in **Pesch et al.** (1993 a).

5. Modellformulierung

Im folgenden beschreiben wir zunächst – unter Verwendung eines Beispiels – die Problemstellung und präsentieren dann eine "konzeptionelle" Formulierung des Problems.

Gegeben seien die drei Produkte A, B und C. Für diese Produkte liegen im Planungszeitraum (von z.B. $T = 80$ Tagen) folgende "Bestellungen" vor: Von Produkt A werden im Zeitpunkt 28, 56 bzw. 80 genau 4, 8 bzw. 12 ME nachgefragt, von Produkt B im Zeitpunkt 24, 52 bzw. 76 genau 4, 12 bzw. 8 ME und von Produkt C schließlich im Zeitpunkt 36, 64 bzw. 80 genau 4, 4 bzw. 12 ME. *Abbildung 1* zeigt die Aufteilung der Gesamtnachfrage je Produkt auf die einzelnen Bestellungen und deren zeitliche Lage (spätestmöglicher Liefertermin).

Abbildung 1: Zeitliche Verteilung der Nachfragen



Die einzelnen Nachfragen nach den Produkten wurden in *Abbildung 1* von 1 bis 9 aufsteigend numeriert; wir bezeichnen sie im folgenden als Jobs und verwenden zur Identifikation den Index $j = 1, \dots, J$.

Tabelle 1 gibt

- die spätestmöglichen Liefertermine d_j (deadlines),
 - die Bearbeitungszeiten p_j (processing times) und
 - die Kosten der vorzeitigen Fertigstellung e_j (earliness costs)
- für alle Jobs j wieder.

Tabelle 1: Daten des Beispiels

j	1	2	3	4	5	6	7	8	9
d_j	28	56	80	24	52	76	36	64	80
p_j	4	8	12	4	12	8	4	4	12
e_j	4	8	12	5	15	10	6	6	18

e_j entspricht dabei dem Produkt aus Lagerhaltungskostensatz h_j (pro Produkt und Periode) mit den Bearbeitungszeiten p_j (vgl. auch *Abbildung 1*).²⁴

Zwischen den Produkten A, B und C sind reihenfolgeabhängige Rüstkosten und -zeiten zu beachten. Rüstkosten entsprechen häufig den mit Kosten bewerteten Rüstzeiten; auch in unserem Beispiel führen längere Rüstzeiten zu höheren Rüstkosten. Eine derartige Kostenstruktur tritt z.B. bei Rüstvorgängen auf, bei denen ein Farbwechsel durchzuführen ist. Im Beispiel werde Produkt A mit einer hellen, B mit einer mittleren und C mit einer dunklen Farbe lackiert. Ein Rüsten der Anlage von dunkel nach hell erfordert eine gründliche (und damit kosten- und zeitintensive) Reinigung, während ein Wechsel auf eine dunklere Farbe vergleichsweise schnell durchgeführt werden kann.

Die Rüstkosten SC_{ij} und Rüstzeiten ST_{ij} sehen im Beispiel folgendermaßen aus:

$$SC_{ij} = \begin{bmatrix} 220 & 260 & 270 \\ 0 & 30 & 30 \\ 370 & 0 & 30 \\ 600 & 210 & 0 \end{bmatrix} \quad ST_{ij} = \begin{bmatrix} 2 & 2 & 3 \\ 0 & 1 & 1 \\ 4 & 0 & 1 \\ 6 & 2 & 0 \end{bmatrix}$$

Dabei gilt für die Produktindices $i \in \{0, A, B, C\}$ und $j \in \{A, B, C\}$, wobei das "Produkt" 0 den ungerüsteten (Start-) Zustand der Anlage darstellt.

Unter der Annahme, daß Rüstvorgänge zwischen Jobs weder Zeit erfordern noch Geld kosten, gilt $SC_{ij} := ST_{ij} := 0$, wenn die Jobs i und j zum selben Produkt A, B oder C gehören. Wir erhalten damit aus den obigen Matrizen der Rüstkosten und -zeiten 10×9 Matrizen mit den entsprechenden Angaben für alle Jobs.²⁵

Unter Verwendung der Entscheidungsvariablen

P Permutation der Jobs, wobei P_k den Job an Position $k = 0, \dots, J$ angibt

$X_{[j]}$ Fertigstellungszeitpunkt von Job j ,

erhalten wir die Modellformulierung:

$$\min \sum_{j=1}^J e_{[j]} (d_{[j]} - X_{[j]}) + SC_{[P_{j-1}, P_j]} \quad (1)$$

$$\text{s.t. } X_{[P_{j-1}]} = \min \{d_{[P_{j-1}]}, X_{[P_j]} - p_{[P_j]} - ST_{[P_{j-1}, P_j]}\} \quad (\text{für alle } j) \quad (2)$$

$$X_{[0]} = 0, P_{[0]} = 0 \quad (3)$$

Einige Hinweise zur Erläuterung:

zu (1): Die Zielfunktion minimiert die Summe aus verfrühten Fertigstellungskosten (=Lagerhaltungskosten bis zum Liefertermin) und reihenfolgeabhängigen Rüstkosten.

²⁴ Unter diesen Annahmen gilt für die konstante Periodenkapazität C der Anlage $C=1$.

²⁵ Die Definition von vollständigen Rüstkosten- und -zeitenmatrizen zwischen allen Jobs macht es ohne Probleme möglich, "Minor Setups" zwischen Teilen einer Familie und "Major Setups" beim Wechsel zwischen Produktfamilien (oder völlig "willkürliche" Einträge) zu berücksichtigen; vgl. hierzu z.B. **Potts, Van Wassenhove** (1992).

- zu (2): Diese Restriktion sorgt dafür, daß der späteste Fertigstellungszeitpunkt eingehalten wird und die reihenfolgeabhängigen Rüstzeiten im Falle eines Produktwechsels auf der Anlage beachtet werden.
- zu (3): Der Vektor der Fertigstellungszeitpunkte und der Permutationsvektor müssen geeignet initialisiert werden.

Man beachte, daß der Index $[j]$ – wie gewohnt – die Nummer von Job j angibt, während der "Index" $[P_j]$ eine Variable darstellt.

Die Formulierung (1)-(3), insbesondere die einzige Restriktion (2), ist ausgesprochen kompakt, übersichtlich und ggf. leicht um weitere Restriktionen zu erweitern. Darüber hinaus ist sie unmittelbar in deklarativen Sprachen zu codieren und mit dem Inferenzmechanismus auszuwerten (siehe Abschnitt 6). Zur Verdeutlichung dieser Modellierungsvorzüge ist im Anhang A eine äquivalente, konventionelle, gemischt-binäre Modellierung des Problems wiedergegeben.²⁶

Wir bezeichnen das Modell (1)-(3) als **konzeptionelles** Modell, weil nicht alle Parameter und Daten des Modells ex ante bekannt sind. Insbesondere die Indexvariablen $[P_j]$ führen zu einer fundamentalen Abweichung von der Struktur von Modellen der gemischt-ganzzahligen Programmierung und sind die Ursache dafür, daß (1)-(3) mit entsprechender Software nicht zu lösen ist.

Im Hinblick auf den in Abschnitt 2 skizzierten Gegenstand der Produktionsprozeßplanung ist in (1)-(3) folgendes enthalten:

- Die Gruppierung bzw. Zusammenfassung der Jobs unter Beachtung spätester Fertigstellungstermine trifft im Kern den Gegenstand der Losgrößenplanung.
- Durch Berücksichtigung der beschränkten Kapazität der Anlage wird eine Durchlauf- und Kapazitätsterminierung vorgenommen.²⁷
- Die Ermittlung zulässiger bzw. optimaler Reihenfolgen ist Gegenstand der Reihenfolgeplanung und Feinterminierung.

Das Modell (1)-(3) behandelt also ein grundlegendes Problem der Produktionsprozeßplanung. Es ist eng verwandt mit dem Problem der simultanen Losgrößen- und Ablaufplanung.²⁸

Die Kompaktheit des Modells läßt jedoch nicht den Schluß zu, daß mit seiner Hilfe einfach zulässige bzw. optimale Lösungen zu bestimmen sind. Aus der Komplexitätstheorie²⁹

²⁶ Zu den Gemeinsamkeiten und Unterschieden zwischen konventionellen und wissensbasierten Modellen vgl. **Pesch et al.** (1993b).

²⁷ Vgl. zum Stand der Forschung über Einmaschinen-Scheduling-Probleme die Übersicht in **Gupta / Kyparisis** (1987).

²⁸ Im Fall stationärer Nachfrage sind insbesondere die Arbeiten von **Adam** (1963) sowie **Dinkelbach** (1964) zu nennen. Mit der Entwicklung von Modellen und Methoden bei dynamischer Nachfrage beschäftigen sich z.B. **Drexel / Haase** (1992), **Fleischmann** (1992) sowie **Karmarkar / Schrage** (1985).

²⁹ Die Komplexitätstheorie ordnet Probleme der kombinatorischen Optimierung entweder der Klasse P (mit polynomialem Aufwand lösbar und daher "einfach") oder NP (im ungünstigsten Fall nur mit exponentiellem Aufwand lösbar und daher "schwierig") zu; vgl. zu den Grundlagen der Komplexitätstheorie insbesondere **Garey / Johnson** (1979).

ist bekannt, daß selbst für den Spezialfall³⁰ reihenfolgeunabhängiger Rüstzeiten bereits das Problem der Bestimmung einer konsistenten (zulässigen) Lösung NP-vollständig ist.³¹ Das bedeutet, daß der für die Beantwortung der Frage "Gibt es Variablenwerte P_j und $X_{[j]}$ ($j = 1, \dots, J$) so, daß die Restriktionen (2)-(3) eingehalten werden, bereits mit der Größe von J exponentiell wachsenden Aufwand erfordert. Dieser Aufwand ist natürlich unabhängig von der Art des verwendeten Lösungsverfahrens.³²

6. Beispielhafte Ergebnisse

Das Modell (1)-(3) kann unmittelbar in der Sprache CHARME implementiert und vom PROLOG-Inferenzmechanismus ausgewertet werden. Der deklarative Code von Restriktion (2) besteht nur aus wenigen Zeilen und ist – bis auf geringfügige syntaktische Details – identisch mit der mathematischen Formulierung.

Das in Abschnitt 5 verwendete Beispiel besitzt 145 zulässige Lösungen.³³ Insgesamt sind (unter Beachtung der Reihenfolgen der Nachfragen innerhalb der einzelnen Produkte) 13 verschiedene Permutationen zulässig. Die einzige optimale Lösung besitzt den Zielfunktionswert 2114. In Tabelle 2 sind die optimalen Werte der Variablen P_j und X_j ($j = 1, \dots, J$) wiedergegeben.

Tabelle 2: Optimale Lösung des Beispiels

j	1	2	3	4	5	6	7	8	9
P_j	1	7	8	4	5	6	2	3	9
X_j	8	55	67	23	35	43	13	17	80

Die optimale Lösung von Tabelle 2 ist in Abbildung 2 in Form eines Gantt-Diagrammes wiedergegeben, wobei die erforderlichen Rüstzeiten schraffiert dargestellt sind. Es ist kein Rüstvorgang von Produkt C nach A erforderlich, also kein Wechsel von der Farbe dunkel auf die Farbe hell; dieser wäre hier ganz besonders teuer und würde entsprechend lange dauern.

Aus Abbildung 2 ist der Produktionsprozeßplan sehr einfach abzulesen. Es werden fünf Fertigungsaufträge gebildet: Zunächst ist Produkt A (Job 3), dann Produkt C (Jobs 9 und 8), schließlich Produkt B (Jobs 6, 5 und 4), dann wieder Produkt A (Jobs 2 und 1) und schließlich erneut Produkt C (Job 7) zu fertigen. Damit sind unter Beachtung verfügbarer Kapazitäten

³⁰ Mit der Entwicklung leistungsfähiger Heuristiken für diesen Spezialfall beschäftigen sich Unal / Kiran (1992).

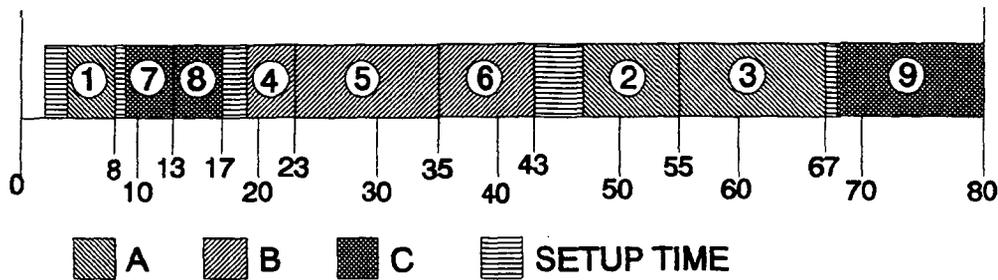
³¹ Vgl. Bruno / Downey (1978).

³² Wie z.B. PROLOG-Inferenzmechanismus, MIP (mixed-integer programming)-solver = Verfahren der gemischt-ganzzahligen Programmierung, spezialisiertes exaktes Verfahren usw.

³³ Die bei wissensbasierten Ansätzen prinzipiell sehr einfach durchführbare Generierung aller zulässigen Lösungen eröffnet weitreichende Möglichkeiten der Evaluation von Alternativen (Lösungen) hinsichtlich mehrerer, auch qualitativer, Zielkriterien (MCDM = Multi Criteria Decision Making).

und spätester Fertigstellungstermine durch Reihenfolgebildung die Fertigungsauftragsgrößen bestimmt.³⁴

Abbildung 2: Gantt-Diagramm der optimalen Lösung



7. Weiterführende Forschungsarbeiten

Die Verwendung wissensbasierter Ansätze zur Produktionsprozeßplanung ist theoretisch interessant und bietet darüber hinaus die bekannten Vorzüge deklarativer Sprachkonzepte (Trennung von Wissen über das Anwendungsgebiet und Problemlösungswissen bzw. Modell und Methode, geringer Programmieraufwand, Änderungsfreundlichkeit usw.). Damit wird jedoch leider nicht die inhärente Komplexität des Problems beseitigt – mit der Folge, daß Probleme praktischer Größenordnung mit dem Inferenzmechanismus i.d.R. nicht zu bearbeiten sein werden.

Der Hauptvorteil besteht damit zunächst darin, daß mit seiner Hilfe Benchmark-Lösungen für die Entwicklung und die experimentelle Untersuchung leistungsfähiger Heuristiken gewonnen werden können. Darüber hinaus bietet er Vorzüge bei der Modellierung folgender Verallgemeinerungen der hier betrachteten Problemstellung: Einbeziehung mehrerer Anlagen, mehrstufige Erzeugnis- und Prozeßstrukturen, Entscheidungen über Eigenfertigung bzw. Fremdbezug usw.

³⁴ Zum erforderlichen Rechenaufwand des Inferenzmechanismus von CHARME vgl. Jordan / Drexel (1993); dort findet sich auch ein Vergleich des Laufzeitverhaltens von CHARME und OS� (Optimization Subroutine Library), einem generellen MIP-Solver.

Literaturhinweise

- Adam, D.:** Simultane Ablauf- und Programmplanung bei Sortenfertigung mit ganzzahliger linearer Programmierung. Zeitschrift für Betriebswirtschaft, Jg. 33 (1963), S. 233-245.
- Adelsberger, H. / Gandenberger, S. / Hansen, T.M. / Jensen, S. / Mehmanesh, T. / Schreier, U. / Vesterager, J.:** The concept of a knowledge-based Leitstand - Summary of first results and achievements in Esprit Project 5161 (KBL). CIM-Europe 8th Annual Conference, Birmingham, Mai 1992.
- Błażewicz, J. / Ecker, K. / Schmidt, G. / Weglarz, J.:** Scheduling in computer and manufacturing systems. Berlin 1993.
- Bruno, J. / Downey, P.:** Complexity of task sequencing with deadlines, set-up times and changeover costs. SIAM J. Comput., Vol. 7 (1978), S. 393-404.
- BULL:** Artificial intelligence, Charme V1, Users Guide & Reference Manual. Bull S.A., Cedoc-Dilog 1991.
- Cohen, J.:** Constraint logic programming languages. Communications of the ACM 33 (1990), S. 52-68.
- Colmerauer, A.:** An introduction to PROLOG III. Communications of the ACM 33 (1990), S. 69-90.
- Corsten, H.:** Produktionswirtschaft - Einführung in das industrielle Produktionsmanagement. München-Wien 1990.
- Dincbas, M. / Simonis, H. / van Hentenryck, P.:** Solving large combinatorial problems in logic programming. J. of Logic Programming 8 (1990), S. 75-93.
- Dinkelbach, W.:** Zum Problem der Produktionsplanung in Ein- und Mehrproduktunternehmen. Würzburg-Wien 1964.
- Dinkelbach, W.:** Entscheidungsmodelle. Berlin - New York 1982.
- Domschke, W. / Drexl, A.:** Kapazitätsplanung in Netzwerken - Ein Überblick über neuere Modelle und Verfahren. OR Spektrum, Bd. 13 (1991), S. 63-76.
- Domschke, W. / Scholl, A. / Voß, S.:** Produktionsplanung - Ablauforganisatorische Aspekte. Berlin 1993.
- Drexl, A./Fleischmann, B./Günther, H.-O./Stadtler, H./Tempelmeier, H.:** Konzeptionelle Grundlagen kapazitätsorientierter PPS-Systeme. Arbeitspapier, Universität Kiel 1993.
- Drexl, A. / Haase, K.:** A new type of model for multi-item capacitated dynamic lotsizing and scheduling. Arbeitspapier, Universität Kiel 1992.
- Fleischmann, B.:** The discrete lot-sizing and scheduling problem with sequence-dependent setup costs. Arbeitspapier, Universität Augsburg 1992 (erscheint in: European J. of Operational Research).
- Freuder, E.C. / Wallace, R.J.:** Partial constraint satisfaction. Artificial Intelligence 58 (1992), S. 21-70.
- Garey, M.R. / Johnson, D.S.:** Computers and intractability: A guide to the theory of NP-completeness. San Francisco 1979.
- Glaser, H. / Geiger, W. / Rohde, V.:** PPS - Produktionsplanung und -steuerung: Grundlagen - Konzepte - Anwendungen. 2. Aufl., Wiesbaden 1992.
- Günther, H.-O.:** Netzplanorientierte Auftragsterminierung bei offener Fertigung. OR Spektrum 14 (1992), S. 229-240.
- Günther, H.-O.:** Produktionsmanagement-Einführung mit Übungsaufgaben. Berlin 1993.
- Gupta, S.K. / Kyparisis, J.:** Single machine scheduling research. OMEGA, Vol. 15 (1987), S. 207-227.
- Haase, K.:** Lotsizing and scheduling for production planning. Kiel 1993 (in Vorbereitung).

- Hahn, D. / Laßmann, G.** (Hrsg.): Produktionswirtschaft - Controlling industrieller Produktion, Band 2, Heidelberg 1989.
- Hansmann, K.-W.:** Industrielles Management. 3. Aufl., München-Wien 1992.
- van Hentenryck, P.:** A logic language for combinatorial optimization. *Annals of Operations Research* 21 (1989), S. 247-274.
- Hoitsch, H.-J.:** Produktionswirtschaft - Grundlagen einer industriellen Betriebswirtschaftslehre. München 1985.
- Jordan, C. / Drexl, A.:** A comparison of logic and mixed-integer programming solvers for batch sequencing with sequence-dependent setups. Arbeitspapier, Universität Kiel 1993.
- Karmarkar, U.S. / Schrage, L.:** The deterministic dynamic product cycling problem. *Operations Research*, Vol. 33 (1985), pp. 326-345.
- Kistner, K.-P. / Steven, M.:** Produktionsplanung. 2. Aufl., Heidelberg 1993.
- Kuik, R. / Salomon, M. / Van Wassenhove, L.N.:** Batching decisions: Structure and models. Erasmus Universiteit Rotterdam, Management Report Series No. 130, 1993.
- Kurbel, K.:** Entwicklung und Einsatz von Expertensystemen - Eine anwendungsorientierte Einführung in wissensbasierte Systeme. Berlin (Springer) 1989.
- Mertens, P.:** Integrierte Informationsverarbeitung, Band 1: Administrations- und Dispositionssysteme in der Industrie. 8. Aufl., Wiesbaden 1991.
- Pesch, E. / Drexl, A. / Kolen, A.:** Automatische Wissensakquisition mittels modellbasierter Inferenz in CHARME. Arbeitspapier, Maastricht 1993 a.
- Pesch, E. / Drexl, A. / Salewski, S.:** Zur Bedeutung der Modellbildung für die Entwicklung wissensbasierter Systeme. Arbeitspapier, Kiel 1993 b.
- Pique, J.-F.:** Prolog II, a step on the Prolog road. *AICOM* 1 (1988), S. 4-16.
- Potts, C.N. / Van Wassenhove, L.N.:** Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. *J. of the Operational Research Society*, Vol. 43 (1992), pp. 395-406.
- Schneeweiß, Chr.:** Modellierung industrieller Lagerhaltungssysteme. Berlin 1981.
- Stadtler, H. / Wilhelm, S.:** Einsatz von Fertigungsleitständen in der Industrie. *CIM Management* 1/93, S. 39-44.
- Tate, A.:** A review of knowledge-based-planning techniques. *Expert Systems* 85 (1985), S. 89-111.
- Tempelmeier, H.:** Material-Logistik - Grundlagen der Bedarfs- und Losgrößenplanung in PPS-Systemen. Berlin 1992.
- Unal, A.T. / Kiran, A.S.:** Batch sequencing. *IIE Transactions*, Vol. 24 (1992), S. 73-83.
- Zäpfel, G.:** Produktionswirtschaft - Operatives Produktions-Management. Berlin-New York 1982.

Zusammenfassung

Wir betrachten ein Problem der Produktionsprozeßplanung, bei dem unter Beachtung verfügbarer Kapazitäten und reihenfolgeabhängiger Rüstvorgänge Fertigungsauftragsgrößen festzulegen sind. Für diese Problemstellung wird ein ausgesprochen kompaktes Modell vorgestellt. Wir bezeichnen es als "konzeptionell", weil nicht alle eingehenden Größen ex ante bekannt sind und insbesondere Indexvariablen vorkommen. Es kann damit nicht von herkömmlicher Software zur gemischt-ganzzahligen Optimierung verarbeitet werden, ist jedoch unmittelbar für deklarative (wissensbasierte) Programmiersprachen geeignet und kann damit auch vom Inferenzmechanismus ausgewertet werden.

Summary

We consider the problem of batching and sequencing for a single facility under sequencing-dependent setup times and costs. A tight "conceptual" model is presented, which is – due to the existence of index variables – not accessible to general purpose mixed-integer programming software. It is, however, well suited for being encoded in declarative (logic) programming languages with direct access to the inference engine.

Anhang A: Gemischt-ganzzahliges Programmierungsmodell

Im folgenden geben wir eine gemischt-ganzzahlige Formulierung des konzeptionellen Modells (1)-(3). Dabei verwenden wir die folgenden Parameter und Daten:

J	Anzahl der Jobs
i, j	Jobindices, $0 \leq i, j \leq J+1$ mit 0 als erstem (fiktivem) und $J+1$ als letztem (fiktivem) Job
d_j	Liefertermin von Job j
e_j	Kosten vorzeitiger Fertigstellung von Job j pro Zeiteinheit
p_j	Bearbeitungszeit von Job j
ST_{ij}	Rüstzeit zwischen Job i und Job j
ST'_{ij}	Rüstkosten zwischen Job i und Job j

Als Entscheidungsvariablen definieren wir:

X_j := Fertigstellungszeitpunkt von Job j

Y_{ij} := $\begin{cases} 1 & \text{falls Job } i \text{ direkt vor Job } j \text{ bearbeitet wird} \\ 0 & \text{sonst} \end{cases}$

Damit ergibt sich folgende Modellformulierung:

$$\min \sum_{j=1}^J e_j (d_j - X_j) + \sum_{i=0}^J \sum_{\substack{j=1 \\ j \neq i}}^J SC_{ij} Y_{ij} \quad (\text{A.1})$$

$$\text{s.t.} \quad \sum_{\substack{j=1 \\ j \neq i}}^{J+1} Y_{ij} = 1 \quad (i = 0, \dots, J) \quad (\text{A.2})$$

$$\sum_{\substack{i=0 \\ i \neq j}}^J Y_{ij} = 1 \quad (j = 1, \dots, J+1) \quad (\text{A.3})$$

$$X_i + ST_{ij} + p_j \leq X_j + M(1 - Y_{ij}) \quad (i = 0, \dots, J, j = 1, \dots, J, i \neq j) \quad (\text{A.4})$$

$$X_j \leq d_j \quad (j = 1, \dots, J) \quad (\text{A.5})$$

$$X_j \geq 0 \quad (j = 1, \dots, J) \quad (\text{A.6})$$

$$Y_{ij} \in \{0, 1\} \quad (i = 0, \dots, J, j = 1, \dots, J+1, i \neq j) \quad (\text{A.7})$$

Einige Hinweise zur Erläuterung: (A.1) minimiert die Summe aus Lagerhaltungs- und reihenfolgeabhängigen Rüstkosten. (A.2) und (A.3) sorgen dafür, daß jeder Job genau einen Vorgänger und einen Nachfolger hat. (A.4) sichert die Einhaltung des erforderlichen Mindestabstandes (Rüst- und Bearbeitungszeit) zwischen zwei unmittelbar aufeinander folgenden Jobs. Durch (A.5) werden die spätesten Fertigstellungszeitpunkte eingehalten. (A.6) und (A.7) sind die obligatorischen Nichtnegativitäts- und Binäritätsrestriktionen (X_0 ist mit 0 zu initialisieren).

Man kann unmittelbar erkennen, daß (A.1)-(A.7) ein "konventionelles" gemischt-binäres Optimierungsmodell darstellt, weil alle erforderlichen Parameter und Koeffizienten ex ante bekannt sind. Ein Vergleich von (A.1)-(A.7) mit (1)-(3) macht die Kompaktheit des konzeptionellen Modells besonders augenfällig.