

*Diskrete ereignisorientierte Prozesse
aus der Sichtweise der
linearen Systemtheorie*

Von Alexander M. Gross

Seminar DEDS
(Diskrete ereignisorientierte dynamische Systeme)
Dr. H.-J. Kühn
Abteilung II
Wintersemester 1994/1995

Rheinische Friedrich-Wilhelms-Universität Bonn

Inhaltsverzeichnis

1 EINFÜHRUNG.....	3
2 EINE LINEARE REPRÄSENTATION VON DISKRETEN EREIGNISORIENTIERTEN SYSTEMEN	5
2.1 ENDLICHE DETERMINISTISCHE DISKRETE EREIGNISORIENTIERTE SYSTEME	5
2.2 ALGEBRAISCHE REPRÄSENTATION	6
2.3 DYNAMISCHE REPRÄSENTATION VON GESCHLOSSENEN DISKRETEN EREIGNISORIENTIERTEN SYSTEMEN	8
2.4 DAS DUALE SYSTEM.....	8
3 LÖSEN VON (MAX, +)-REKURSIVEN FORMELN	10
3.1 EIGENWERTE	10
3.2 EIGENVEKTOREN	11
3.3 PERIODIZITÄT	11
3.4 SPEKTRALPROJEKTOREN	12
3.5 LÖSUNG VON REKURSIVEN FORMELN FÜR DISKRETE EREIGNISORIENTIERTE SYSTEME	13
4 ANWENDUNGEN IN DER PRODUKTION	14
4.1 BESCHREIBUNG EINES MULTIPRODUKTIONSPROZESSES IN EINER FLEXIBLEN FERTIGUNGSUMGEBUNG	14
4.2 DARSTELLUNG EINES FMS ALS GESCHLOSSENES DISKRETES EREIGNISORIENTIERTES SYSTEM	15
4.3 PERIODISCH FESTER ZUSTAND DES FMS-VERHALTENS	16
4.4 HINZUFÜGEN VON PALETTEN AM BEISPIEL DES UMLAUFBETRIEBS	18
5 ZUSAMMENFASSUNG.....	21
LITERATURVERZEICHNIS.....	22

Diskrete ereignisorientierte Prozesse aus der Sichtweise der linearen Systemtheorie

Das Verhalten von diskreten ereignisorientierten Systemen wird durch Aktivitäten beschrieben, welche für einen gewissen Zeitraum existieren und einer bestimmten Ordnung folgen. Ereignisse beschreiben den Zeitpunkt, an dem Aktivitäten gestartet oder beendet werden. Zwischen diskreten ereignisorientierten Systemen und linearen Systemen besteht eine gewisse Analogie, so daß man im Sinne einer Algebra auch bei diskreten ereignisorientierten Systemen von Linearität sprechen kann. Dabei kann man das periodische Verhalten von geschlossenen diskreten ereignisorientierten Systemen, die z.B. wiederholt identische Aktivitäten durchführen, komplett durch Formeln dieser Algebra darstellen.

1 Einführung

Linearität ist eine Eigenschaft mancher Systeme, die die Analyse maßgeblich vereinfacht. Viele Systeme sind jedoch nicht-linear und so versucht man den Umgang mit solchen Systemen dahingehend zu vereinfachen, in dem man Teile eines Systems betrachtet, die sich linear beschreiben lassen. Daß es allerdings nicht grundsätzlich notwendig ist, sich auf Linearität zu stützen,

kann man am Beispiel von diskreten ereignisorientierten Systemen verdeutlichen. In der linearen Systemtheorie verwendet man für Zeitsysteme die Zustandsdarstellung durch dynamische Systeme. Man beschreibt dabei den Übergang der Eingabe zu einer Ausgabe durch eine sog. Zustandstrajektorie, sowie zwei lineare Abbildungen F und g . Ein *dynamisches System* wird durch ein 9-Tupel $\Sigma=(T,X,Y,Z,X,Y,F,g,\leq)$ beschrieben, mit Zeitmenge T , Eingabealphabet X , Ausgabealphabet Y , Zustandsalphabet Z , Eingangssignale X , Ausgangssignale Y und Zeitrelation \leq . Sei die Überföhrungsfunktion $F:(T \times T \times Z \times X)' \rightarrow Z$ mit $F(t',t,z,x)=z'$, d.h. ein Zustand z zum Zeitpunkt t wird durch Eingabe x in den Zustand z' zum Zeitpunkt t' überföhrt. Sei die Ausgabe- bzw. Ergebnisfunktion $g:T \times Z \times X \rightarrow Y$ mit $g(t,z,x)=y$, d.h. eine Eingabe x erzeugt eine Ausgabe y . Man nennt F und g auch Zustandsgleichungen des dynamischen Systems. In physikalischen Systemen handelt es sich i.a. bei den Alphabeten X , Y und Z um Mengen von reellen oder komplexen Zahlen, wobei technische Systeme, wie z.B. Automaten, auch Werte aus endlichen Mengen annehmen, so daß man diese als *endliche Systeme* bezeichnet.

Ein dynamisches System heißt *linear*, wenn die Alphabete X , Y , Z lineare *Vektorräume* $(X,+)$, $(Y,+)$ und $(Z,+)$ über einem Körper $K=(K,+,\bullet)$ sind, wobei Vektoraddition und Skalarmultiplikation für Signale $x:T \rightarrow X$ aus X definiert sind.

Unter dem Begriff der *diskreten ereignisorientierten dynamischen Systeme* versteht man im allgemeinen Systeme, deren Verhalten durch Start- und Endzeit von Aktivitäten charakterisiert wird. Eine solche Struktur kann durch Zeit-Ereignis-Graphen [3] dargestellt werden. Um das Systemverhalten zu berechnen, werden Maximum und Minimum einer Gruppe von Aktivitäten benötigt. Viele diskrete Produktionsprozesse können mit Hilfe von diskreten ereignisorientierten Systemen beschrieben werden, was durch Verwendung

von sog. diskreten ereignisorientierten Simulationssprachen [14] geschieht.

Man kann zeigen, daß deterministische diskrete ereignisorientierte Systeme im Sinne von speziellen Algebren, welche unter Bezeichnungen wie Pfad-Algebra [4] oder Minimax-Algebra [8] bekannt sind, als linear angesehen werden können. Hierdurch kann eine Repräsentation aufgebaut werden, die sich an den jeweiligen Zuständen und deren Dauer orientiert und die es ermöglicht, daß man auf effiziente Art und Weise kontinuierliche Zustände von sog. geschlossenen diskreten ereignisorientierten Systemen berechnen kann, d.h. bei typischerweise beschränkten Ressourcen wird eine endliche Anzahl von Aktivitäten wiederholt ausgeführt. Diskrete ereignisorientierte Systeme können darüber hinaus durch getaktete Petri-Netze [28], welche aus Markierungen und Transitionen bestehen, beschrieben werden. Die Übergänge stellen dann die Aktivitäten und die Zustände die Ressourcen dar. In Petri-Netzen reflektieren die Transitionen die Veränderungen der Zählerzustände, d.h. der Markierungen. Die Marken werden dabei auf sog. Plätzen gespeichert bzw. gezählt, für die man auch eine Kapazität, also die maximal speicherbare Anzahl von Marken, angeben kann. Da in dieser Darstellungsform die Transitionen nur dann Aktionen ausführen können, wenn der Platz hinreichend viele Marken beinhaltet, ist dies direkt auf diskrete ereignisorientierte Systeme übertragbar, da die Aktivitäten erst bei verfügbaren Ressourcen ausführbar sind. Dadurch ergibt sich auch eine Bedeutung für verteilte und parallele Systeme.

In diesem Beitrag geht es um die Modellierung von Produktionsprozessen, wie sie in automatisierten Produktionssystemen vorkommen, welche man als flexible Fertigungssysteme bezeichnet. Genauere technische Hintergründe erfährt man in den Arbeiten von Groover [19] und den Proceedings der FMS-1 conference [15]. Außerdem gibt es Lösungen, die flexible Fertigungssysteme mittels stochastischen Netzwerkwarteschlangen realisieren (Solberg [32], Hildebrandt [20], Secco-Suardo [30]).

Man verwendet diese Modelle, wenn nur geringe Informationen über ein System verfügbar sind und Aussagen für eine längere Ausführung benötigt werden. Diese Modelle schaffen eine recht genaue Näherung, da sie exponentiell auf Maschinen verteilte Rechenzeit annehmen. Die flexiblen Fertigungssysteme verhalten sich dabei im wesentlichen deterministisch, auch wenn Schwankungen in der Ausführungszeit auftreten. Auf Grund dessen bieten Modelle, die auf Jackson-Netzwerksschlangen [23] basieren, quantitativ keine verlässlichen Aussagen bzgl. der Ausführung. Obgleich diese Näherungsverfahren sich mit nicht-klassischen Netzwerkwarteschlangen messen können und dabei genauere Ergebnisse liefern, gibt es noch keine eindeutigen Aussagen darüber, welche Verfahren sich in Zukunft durchsetzen werden.

Allerdings gibt es Fortschritte auf diesem Gebiet im Bezug auf nicht-stochastische Ansätze, die sog. operationale Analyse (Denning/Buzen [11]), welche Verbindungen zwischen den Ergebnissen verschiedener Meßergebnisse herstellen. Diese Zusammenhänge sind identisch mit denen, die man durch Jackson-Netzwerksschlangen ebenfalls erhält, allerdings mit weniger Aufwand. Hierzu existiert ein Algorithmus von Dallery und David [10] zur Bewertung der Ausführung, der Simulationsläufe überflüssig macht.

Der Ansatz, der im folgenden vorgestellt werden soll, bietet nicht nur genaue Ausführungsbewertungen, sondern beinhaltet zusätzlich Regeln zur Echtzeit-Kontrolle, die den flexiblen Fertigungssystemen ein reguläres und stabiles Verhalten ermöglichen, sowie den Produktionsanforderungen genügen.

Dieser Ansatz lehnt sich an Modelle für wiederkehrende Produktionsprozesse an. Daher existiert auch eine gewisse Einschränkung, die es erforderlich macht, daß es sich um entscheidungsfreie Systeme handeln muß, d.h. Prozeßfolgen oder auch Teile davon müssen fest vorgegeben sein. Unter Berücksichtigung dieser Tatsache ist man in der Lage, das periodische Verhalten eines Systems zu

ermitteln, was nach endlich vielen Schritten erreicht werden kann.

Es handelt sich hier um ein Verfahren, welches als eine Vorstufe zu Design- und Kontrollproblemen bezogen auf Fertigungssysteme betrachtet werden kann. Es ist vergleichbar mit der Bedeutung der Kritischen-Graph-Methode bei klassischen Planungs- und Ablaufproblemen.

Im folgenden soll zunächst die erwähnte Zustand-Raum-Repräsentation von diskreten ereignisorientierten Systemen vorgestellt werden, zusätzlich auch einige rekursive Formeln zum Verhalten von geschlossenen Systemen. Im Anschluß daran werden mathematische Hintergründe zur Lösung der Rekursionen dargelegt, so daß das Systemverhalten anhand des Zustandsverhaltens charakterisiert werden kann. Der letzte Abschnitt befaßt sich dann mit der Analyse eines flexiblen Fertigungssystems und einem bereits von Cuninghame-Green [7] vorgestellten Ansatz zu allgemeinen Produktionsprozessen.

2 Eine lineare Repräsentation von diskreten ereignisorientierten Systemen

2.1 Endliche deterministische diskrete ereignisorientierte Systeme

Ein endliches deterministisches diskretes ereignisorientiertes System kann betrachtet werden als eine endliche Menge A von Aktivitäten und eine endliche Menge R von Ressourcen, die von Aktivitäten geteilt werden. Die Reihenfolge der Aktivitäten kann durch einen azyklischen, gerichteten, zusammenhängenden Graphen $G=(A,U)$ beschrieben werden [2], wobei U die Menge der Transitionen ist. Für zwei beliebige Aktivitäten a_i und a_j , bedeutet $(i,j) \in U$, daß a_i vor a_j ausgeführt wird.

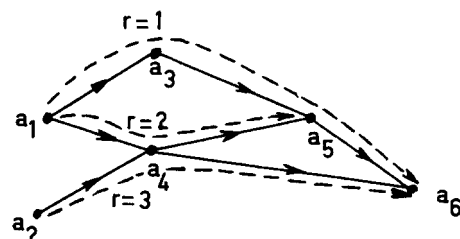


Abb. 1

Mit jeder Ressource $r \in R$ assoziieren wir einen einzigen Pfad P_r , d.h. eine Folge von Aktivitäten, in G . Die Menge aller Pfade in G $P = \{P_r \mid r \in R\}$ ist die Abdeckung von G , d.h. $\forall (i,j) \in U, \exists r : (ij) \in P_r$.

Ein Grund dafür, daß a_i vor a_j ausgeführt werden soll, liegt darin, daß beide evtl. die gleiche Ressource benutzen wollen, welche natürlich zu einem Zeitpunkt nur einer Aktivität zur Verfügung stehen kann. Desweiteren könnte es Gründe geben, die eine bestimmte Reihenfolge erforderlich machen und sei es deshalb, weil die Vorschrift im

Graphen G dies erfordert, um einen bestimmten Prozeß zu erreichen.

Wir bezeichnen den Graphen G als *disjunkt*, genau dann wenn zwei Aktivitäten auf die gleiche Ressource zugreifen wollen und durch die Vorschrift nicht erkennbar ist, welche Aktivität Vorrang hat. Dann ist das System nicht entscheidungsfrei und somit auch nicht deterministisch.

Ein deterministisches diskretes ereignisorientiertes System wird definiert durch ein 4-Tupel (A, U, R, P) .

Beispiel 1 :

Ein Verarbeitungsprozeß kann beschrieben werden durch die Verarbeitung von n Werkstücken in einer Werkstatt mit m Maschinen. Jedes Werkstück besitzt eine Verarbeitungsfolge, die aus endlich vielen Stationen an den Maschinen besteht, wobei die Reihenfolge durch den Ablauf definiert ist. Um ein solches System zu simulieren, muß die Folge von Aufgaben einer jeden Maschine vorgeschrieben werden.

Wir erhalten daher,

A =Menge von Verarbeitungsschritten zur Bearbeitung von n Werkstücken, wobei Tupel (i,j) für Teil j an Maschine i stehen

R =Menge von Maschinen und Werkstücken

U =Menge von zwangsläufigen Reihenfolgen von Aufgaben, bedingt durch Verarbeitungsfolgen und Werkstückabfolgen an Maschinen

P =Verarbeitungsfolgen und Werkstückabfolgen an Maschinen.

Jeder Ressourcen-Pfad P_r besitzt eine Start- und eine Endaktivität, welche wir mit $a_1(r)$ und $a_{kr}(r)$ bezeichnen. Die Mengen der initialen und finalen Aktivitäten werden folglich definiert durch

$$I = \{a_1(r) \mid r \in R\} \text{ und } F = \{a_{kr}(r) \mid r \in R\}.$$

Beachte, daß I bzw. F alle Aktivitäten beinhalten, die keine Vorgänger bzw. Nachfolger besitzen.

Aus Abb. 1 erhalten wir

$$P_1 = a_1 a_3 a_5 a_6, \quad P_2 = a_1 a_4 a_5, \quad P_3 = a_2 a_4 a_6 \quad \text{und} \\ I = \{a_1, a_2\}, \quad F = \{a_5, a_6\}.$$

Jeder Teilweg $(i,j) \in U$ erhält eine positive reelle Zahl t_{ij} , die betrachtet werden kann als Summe t_i der Dauer von Aktivität i und der erforderlichen Umschaltzeit δ_{ij} von Aktivität i nach Aktivität j . Als *Gewicht* eines Pfades in (A, U) bezeichnen wir die Summe der Teilgewichte t_{ij} der Teilwege (i,j) , die im Pfad P_r enthalten sind. Die Anzahl der Teilpfade im Pfad nennen wir *Länge*.

2.2 Algebraische Repräsentation

Sei nun x_i die früheste Startzeit von Aktivität a_i und sei u_r der Zeitpunkt, zu dem Ressource r für die erste Aktivität des Ablaufes P_r verfügbar ist.

Wenn $(j,i) \in U$, dann $x_i \geq x_j + t_{ij}$. Falls i erste Aktivität für Ressource r , dann $x_i \geq u_r$.

Sei $\Gamma^-(i)$ die Menge der Vorgänger von Aktivität i und $R^0(i)$ die Menge der Ressourcen, so daß $a_1(r) = i$, dann ist x_i eindeutig definiert durch

$$\forall a_i \in A, \quad x_i = \max(\max(x_j + t_{ij}), \max u_r), \quad (1)$$

wobei $j \in \Gamma^-(i)$ und $r \in R^0(i)$.

Beachte, daß diese Formel durch dynamische Optimierung bis zum Term u_r realisiert wird. Sei A die *gewichtete Vorkommensmatrix* von (A, U) definiert durch die Einträge

$$\begin{aligned} A_{ij} &= t_{ij} && \text{falls } (i,j) \in U \\ A_{ij} &= -\infty && \text{falls } (i,j) \notin U. \end{aligned}$$

A ist eine $N \times N$ -Matrix mit $N=|A|$. Sei B eine $R \times N$ -Matrix mit $R=|R|$, so daß

$$\begin{aligned} B_{ri} &= 0 && \text{falls } a_1(r)=i \\ B_{ri} &= -\infty && \text{sonst.} \end{aligned}$$

B bestimmt eindeutig die Startaktivitäten aller Ressourcen.

Schreibweisen [13]:

$$\forall a, b \in \mathbf{IR} \quad a \oplus b \cong \max(a, b) \quad (2)$$

$$a \bullet b \cong a + b \quad (3)$$

für alle Matrizen A, A' ,

$$(A \oplus A')_{ij} = A_{ij} \oplus A'_{ij} \cong \max(A_{ij}, A'_{ij}) \quad (4)$$

$$(A \bullet A')_{ij} \cong \max(A_{ik} + A'_{kj}) \quad \text{für alle } k \quad (5)$$

Es sei angemerkt, daß in (4) A und A' die gleiche Größe haben. In (5) ist die Anzahl der Zeilen in A gleich der Anzahl an Spalten in A' .

Man kann (1) auch kompakter schreiben als

$$X = XA \oplus UB \quad (6)$$

mit $X = (x_1 \dots x_N)$, $U = (u_1 \dots u_R)$ und XA entspricht $X \bullet A$.

Sei nun Y der Vektor der frühesten Zeiten y_r von Ressourcenfreigaben. Wenn C eine $N \times R$ -Matrix ist, deren Einträge die Dauer der letzten ausgeführten Aufgaben, d.h.

$$\begin{aligned} C_{ir} &= t_i && \text{falls } \exists r : a_{kr}(r) = i \\ C_{ir} &= -\infty && \text{sonst,} \end{aligned}$$

dann kann Y leicht durch

$$Y = XC \quad (7)$$

erhalten werden.

Beachte, daß (6) und (7) der Zustand-Raum-Repräsentation linearer Systeme entspricht. Wir werden später sehen, um was es sich hierbei handelt.

Doch zunächst werden wir die Formeln (6) und (7) näher betrachten. Hierzu benötigen wir ein Lemma, welches den Aufbau der Mengen von reellen Matrizen in Bezug auf (4) und (5) erklärt.

Lemma 1 :

$\oplus = \max()$ ist kommutativ und assoziativ über \mathbf{IR} , $\varepsilon = -\infty$ ist die Identität, so daß $\varepsilon \oplus a = a$, $\forall a \in \mathbf{IR}$. Desweiteren gilt $a \oplus a = a$. Die Operation $\bullet = \text{Addition}$ ist distributiv bzgl. \oplus . ε ist absorbierendes Element bzgl. \bullet , d.h. $\varepsilon \bullet a = \varepsilon$, $\forall a \in \mathbf{IR}$. $e = 0$ ist Identität bzgl. \bullet .

Das durch (5) definierte Matrixprodukt ist assoziativ und distributiv bzgl. der Matrixaddition, welche selbst idempotent ist.

Theorem 1 :

Formel (6), also $X = XA \oplus UB$, hat eine einzige Lösung, sobald U bekannt ist, $X = UBA^*$.

Beweis [13] :

Sei X eine Lösung, dann

$$X = (XA \oplus UB)A \oplus UB$$

$$X = XA^2 \oplus UB(E \oplus A).$$

Dabei ist E die Identität des Matrixprodukts, d.h. $E_{ii} = e = 0$ und $E_{ij} = \varepsilon = -\infty$, falls $i \neq j$. Für $N-1$ Schritte sei

$$X = XA^N \oplus UB(E \oplus A \oplus A^2 \oplus \dots \oplus A^{N-1}) \quad (8)$$

mit $A^n = A^{n-1}A = AA^{n-1}$, $n=2, \dots, N-1$.

Der Eintrag A_{ij}^n von A^n beinhaltet das maximale Gewicht eines Pfades von i nach j mit genau n Teilwegen. Da A die Ereignismatrix des azyklischen Graphen ist, besteht jeder enthaltene Pfad aus maximal $N-1$

Teilpfaden. Weiterhin existiert die Nullmatrix $A^N = \emptyset$ mit Einträgen gleich ε . Daher läßt sich jede Lösung von (6) darstellen als

$$X = UB(E \oplus A \oplus A^2 \oplus \dots \oplus A^{N-1}) \cong UBA^* \quad (9)$$

A^* enthält dabei als Einträge die maximalen Gewichte der Pfade zwischen zwei Knoten. ■

(Ein solches A^* kann auch durch Algorithmen zur Bestimmung des längsten Pfades ermittelt werden, die man bzgl. Kritischer-Pfad-Methoden kennt.)

2.3 Dynamische Repräsentation von geschlossenen diskreten ereignisorientierten Systemen

Ein diskretes ereignisorientiertes System heißt *geschlossen*, genau dann wenn es ein endliches deterministisches diskretes ereignisorientiertes System ist, bei dem Rückkopplungen der Endaktivitäten zu den Startaktivitäten existieren.

Die Rückkopplungen dienen dazu, daß die endliche Menge von Ressourcen Aktivitäten wiederholt ausführt. Formal handelt es sich dabei um ein 5-Tupel (A, U, R, P, B) , wobei A, U, R, P wie bisher definiert sind und B einen bipartiten Graphen (F, I, L) mit den bisher bereits definierten F und I , sowie $L \subseteq F \times I$ mit den Teilpfaden aus B

$$(i, j) \in L \Leftrightarrow \exists r \in R, \text{ sowie } i = a_{kr}(r) \text{ und } j = a_1(r).$$

Dabei steht L für das Verhalten, daß Ressourcen nach Abarbeitung einer Endaktivität wieder mit der Startaktivität fortfahren.

Sei $X(n)$ der Vektor der frühesten Startzeiten von Aktivitäten in A im n -ten Durchlauf. Sei $U(n)$ die zugehörigen Startzeiten der Ressourcen. Hierbei kann jede Ressource r

gestartet werden, wenn sie von der ersten Aktivität zum n -ten Mal verwendet werden soll und von der letzten Aktivität zum $(n-1)$ -ten Mal verwendet wurde, und die Umschaltzeit zwischen diesen Aktivitäten verstrichen ist.

Dieser Umstand führt zu folgender Beziehung

$$U(n) = Y(n-1)K \quad (10)$$

wobei K eine $R \times R$ -Matrix ist, mit $K_{rs} = \varepsilon, \forall r \neq s$ und K_{rr} sei die Umschaltzeit t_{ij} zwischen End- und Startaktivität für Ressource r . Mit Hilfe von K_{rr} ist es möglich, die Startzeit der Aktivitäten zu verzögern. Verwendet man nun die Aussagen (7), (9) und (10), so kann die früheste Zeit der n -ten Freigabe einer Ressource ausgedrückt werden durch ihre $(n-1)$ -te Freigabezeit

$$Y(n) = U(n)BA^*C$$

$$Y(n) = Y(n-1)KBA^*C. \quad (11)$$

Mittels der Information über die verfügbaren Startzeiten $U(1)$ von Ressourcen ist man nun in der Lage rekursiv die Folge $Y(1), \dots, Y(n), \dots$ herzuleiten, wodurch wiederum die Folge $X(1), \dots, X(n), \dots$ bestimmt werden kann. Bei (11) handelt es sich um eine dynamisch optimierte Formel, die nicht-endlich viele Schritte durchlaufen kann.

Im weiteren nehmen wir an, daß jede Ressource r nicht nach einer Zeit u_r verfügbar ist, sondern, daß sie nur vor einem bestimmten Zeitpunkt v_r verwendet werden kann.

2.4 Das duale System

Sei q_i die letzte Startzeit einer Aktivität i eines endlichen deterministischen diskreten ereignisorientierten Systems, wie zuvor beschrieben. v_r bezeichne den letzten Zeitpunkt, zu dem Ressource r von Aktivitäten

freigegeben sein muß. Sei z_r der letzte Zeitpunkt, an dem Ressource r verfügbar sein muß, so daß weitere Aktivitäten darauf ausgeführt werden können, d.h. Ressource r muß, für alle r , vor v_r freigegeben werden.

Wenn $(j,i) \in U$, dann $q_j \leq q_i + t_{ji}$. Falls a_i letzte Aktivität für Ressource r , dann $q_i \leq v_r - t_i$.

Falls $\Gamma^+(i)$ die Menge der Nachfolger von Aktivität a_i und $R^\infty(i)$ die Menge der Ressourcen mit Endaktivität i , dann erhalten eindeutig

$$\forall a_i \in R, q_i = \min(\min(q_j + t_{ji}), \min(v_r - t_i)), \quad (12)$$

wobei $j \in \Gamma^+(i)$ und $r \in R^\infty(i)$.

Seien die Vektoren $Q_i = -q_i$, $V_r = -v_r$ und $Z_r = -z_r$. Dies führt zu

$$Q = A Q \oplus C V \quad (13)$$

$$Z = B Q \quad (14)$$

wobei (13) und (14) die duale Repräsentation von (6) und (7) sind. Desweiteren gilt

$$Z = B A * C V.$$

Die geschlossene Formel kann erhalten werden, in dem man

$$V(n) = K Z(n+1) \quad (15)$$

betrachtet, was bedeutet, daß der letzte Freigabezeitpunkt der Ressourcen von Aktivitäten einer Menge n durch die letzten Verfügbarkeitszeitpunkte der folgenden Aktivitäten beeinflusst wird.

Daraus folgt analog zu (11)

$$V(n) = K B A * C V(n+1). \quad (16)$$

Dies setzt die Kenntnis der Zeitpunkte voraus, zu denen Ressourcen freigegeben sein müssen, d.h. Endzustände der Form $V(n^*)$, für einige n^* .

3 Lösen von (max, +)-rekursiven Formeln

Im folgenden geht es um das asymptotische Verhalten von rekursiven Formeln der Form $Y(n)=Y(n-1)M$, wobei M eine reellwertige Matrix und das Matrixprodukt wie in (5) definiert ist.

Dieser Abschnitt soll nur einen groben Einblick geben, um später die Produktionssysteme analysieren zu können.

Rekursive Formeln werden eingesetzt zur vereinfachten Berechnung von Werten, die man mit Hilfe schon bekannter Teilberechnungen beginnend bei einer Startposition ermitteln kann.

3.1 Eigenwerte

Sei $G(M)=(N(M), U(M))$ ein Graph mit $N \times N$ -dimensionaler Ereignismatrix M , d.h.

$$M_{ij} > \varepsilon \Leftrightarrow (i, j) \in U(M) \quad (17)$$

Im weiteren gehen wir davon aus, daß $G(M)$ ein stark verbundener Graph ist, d.h. es existieren zwischen allen im Graph enthaltenen Knoten verbindende Kanten. Dann ist M irreduzierbar.

Ein Vektor $y \neq \varepsilon$ heißt *Eigenvektor*, genau dann wenn es eine reelle Zahl λ , den sog. *Eigenwert*, gibt, so daß

$$yM = \lambda y$$

gilt.

Für einen Pfad $P=i_1, \dots, i_k$ in $G(M)$ sind Länge $l(P)$ und *Gewicht* $w(P)$ wie folgt definiert

$$l(P) = k - 1 = \text{Anzahl der Teilwege in } P$$

$$w(P) = M_{i_1, i_2} \bullet M_{i_2, i_3} \bullet \dots \bullet M_{i_{k-1}, i_k}$$

Beachte, daß $w(P)$ durch eine Summe im gewöhnlichen Sinne berechnet wird.

Hat P die Gestalt $i_1 = i_k$, dann heißt γ *Kreis*.

Wir definieren dazu das *durchschnittliche Gewicht* für beliebige Kreise in $G(M)$ durch

$$\bar{w}(\gamma) = w(\gamma)^{1/l(\gamma)}$$

Sei $w = \sum_{\gamma} w(\gamma)^{1/l(\gamma)}$ das maximale Durchschnittsgewicht aller Kreise in $G(M)$. Da $G(M)$ endlich viele Knoten besitzt, brauchen nur Kreise mit $l(\gamma) \leq N$ für die Summe betrachtet zu werden und falls w existiert und es wenigstens einen Kreis γ gibt mit $\bar{w}(\gamma) = w$, dann heißt γ *kritischer Kreis*.

Seien M^* und M^+ , falls vorhanden, definiert durch

$$M^* = E \oplus M \oplus M^2 \oplus \dots \oplus M^n \oplus \dots \quad (18)$$

$$M^+ = M \oplus M^2 \oplus \dots \oplus M^n \oplus \dots \quad (19)$$

Diese M^* und M^+ existieren nur, falls jeder Kreis in $G(M)$ $w(\gamma) \leq \varepsilon$ ($\cong 0$). Dann gilt

$$M^+ = \sum_{i=1}^{N-1} M^i, \quad M^* = E \oplus M^+; \quad M^+ = M \bullet M^* \quad (20)$$

$$\text{und } M_{ij}^+ = M_{ij}^*, \quad i \neq j; \quad M_{ii}^+ = M_{ii}^* \Leftrightarrow M_{ii}^+ = e.$$

Theorem 2 :

$\lambda = w$ ist Eigenwert von M . Falls γ_0 ein kritischer Kreis ist, dann ist $\forall i \in \gamma_0$ die i -te Zeile von $M_{\lambda}^+ \cong \lambda^{-1} M^+$ ein Eigenvektor.

Beachte, daß $\forall k, \gamma_k$ ein kritischer Kreis ist. Der gerichtete Graph $G_0(M)=(N_0, U_0)$ definiert durch

$N_0=\{i \in N(M_w) | i \text{ gehört zu einem kritischen Kreis}\}$,

$U_0=\{(i,j) \in N(M_w) | (i,j) \text{ gehört zu einem kritischen Kreis}\}$

heißt *kritischer Graph* von M .

Das Theorem liefert somit eine graphentheoretische Interpretation des Eigenwertes λ , sowie einiger Eigenvektoren. Dabei steht jeder Knoten i in $G_0(M)$ für einen Eigenvektor $y^i=(M_w^+)$ und jeder Eintrag y_j^i beinhaltet das maximale Gewicht des Pfades von $i \in N_0$ zu j in $G(M_w)$, wobei jeder maximal gewichtete Pfad zu H gehört.

3.2 Eigenvektoren

Wir nehmen im folgenden an, daß M den Eigenwert e ($=0$) besitzt, andernfalls verwenden wir M_w . Es wird nun gezeigt, daß Eigenvektoren, die man durch M^+ erhalten hat, eine Basis der Menge aller Eigenvektoren besitzen.

Lemma 2 :

Ein Eigenvektor y ist eine lineare Kombination einiger Zeilen in M^+ , genauer

$$y = \sum_{i \in N_0} y_i M_i^+$$

Theorem 3 :

Wenn der kritische Graph nur eine zusammenhängende Komponente besitzt, dann sind alle Zeilen M_i^+ mit $i \in N_0$ kollinear.

Mit Hilfe von Lemma und Theorem könnte man zeigen, daß, wenn ein kritischer Pfad zusammenhängend ist, die Menge der Eigenvektoren 1-dimensional ist, i.a. gilt dies jedoch nicht mehr.

Theorem 4 :

Besitzt ein kritischer Graph $G_0(M)$ K verbundene Komponenten C_k , dann ist die Menge der Eigenvektoren K -dimensional. Man erhält eine Basis der Eigenvektoren durch $\{M_{i_k}^+ | k=1, \dots, K\}$, wobei i_k willkürlich in C_k gewählt wird, $\forall k=1, \dots, K$.

Insbesondere sind alle Eigenvektoren kollinear, falls es nur einen kritischen Kreis gibt.

3.3 Periodizität

Im weiteren setzen wir voraus, daß $\forall n > 0$ die Matrix M^n irreduzierbar ist. Ein hinreichendes Kriterium dafür ist das Vorhandensein einer Schleife $(i,j) \in U$.

Eine Matrix M mit Eigenwert e heißt *Ordnung- d -periodisch*, genau dann wenn es eine Ganzzahl n_0 gibt, so daß gilt

$$\forall n \geq n_0, M^{n+d} = M^n. \quad (22)$$

Daraus folgt, daß wenn M einen Eigenwert λ besitzt und M_λ Ordnung- d -periodisch ist, dann

$$M^{n+d} = \lambda^d M^n, \forall n \geq n_0.$$

Sei im folgenden der Eigenwert zu M gleich e und γ_0 der einzige kritische Pfad.

Lemma 3 :

Wenn γ_0 die Länge d besitzt, dann hat M^d genau d kritische Pfade mit Schleifen um den Knoten $i \in \gamma_0$.

Theorem 5 :

Ist γ_0 der einzige kritische Pfad mit Länge d , dann heißt M Ordnung- d -periodisch.

Verallgemeinert kann man sagen, daß wenn $C_1 \dots C_k$ die kritischen Komponenten von $G_0(M)$ sind, dann ist die Ordnung von C_k definiert durch den größten gemeinsamen Teiler der Längen der kritischen Kreise in C_k .

a) Hat $G_0(M)$ nur eine Komponente von Ordnung d , so ist M Ordnung- d -periodisch

b) Hat $G_0(M)$ K Komponenten mit Ordnung d_k , mit $k=1, \dots, K$, dann ist M Ordnung- d -periodisch mit $d = \text{kgV}(d_1 \dots d_k) = \text{kgV}$ (kgV=kleinstes gemeinsames Vielfaches, [6])

Beachte, daß wenn M^k nicht irreduzierbar ist, dann kann $G(M^k)$ zerlegt werden in disjunkte stark-verbundene Komponenten, wobei jede mit einem bestimmten Eigenwert korrespondiert.

Allgemein erhält man

$$\exists n_0, n \geq n_0 \Rightarrow (M^{dn})_{ij} = \sum_{i_0 \in N_0} (M^d)_{i i_0}^+ (M^d)_{i_0 j}^+ \quad (26)$$

3.4 Spektralprojektoren

Sei M eine Matrix mit Eigenwert e und Periodizität der Ordnung d . Aus dem vorherigen ergeben sich die Quantitäten

$$Q_i = \lim_{n \rightarrow \infty} M^{dn+i} \quad \text{für } i=0, \dots, d-1 \quad (27)$$

$$\text{und } Q_i = M^i Q_0 = Q_0 M^i.$$

Bisher wurden nur auf der linken Seite Eigenvektoren verwendet, es ist ebenfalls möglich, Eigenvektoren auf der rechten Seite

zu benutzen, die bzgl. der Spalten i von M^+ kolinear sind, so daß $i \in N_0$.

Seien $\{\bar{y}_i = (M^d)_i^+ | i \in N_0\}$, $\{\bar{z}_i = (M^d)_i^+ | i \in N_0\}$ Basen von Eigenvektoren von M^d .

Theorem 6 :

$$Q_0 = \sum_{i \in N_0} \bar{z}_i \otimes \bar{y}_i$$

wobei \otimes das sog. *Kronecker-Produkt* der $(\max, +)$ -Algebra ist, mit

$$(\bar{z}_i \otimes \bar{y}_i)_{jk} = (\bar{z}_i)_j \bullet (\bar{y}_i)_k,$$

d.h. zwei Vektoren werden zu einer Matrix verbunden. Q_0 heißt *Projektor*, wobei $Q_0^2 = Q_0$ (Idempotenz) gilt.

Besitzt M den Eigenwert λ , dann $\exists n_0, \forall n \geq n_0$

$$M^{dn+i} = Q_i \lambda^{dn} \quad \text{für } i=0, \dots, d-1$$

Q_0 heißt *Spektralprojektor* von M^d und Q_i kann durch Anwendung von (27) auf $M \lambda^{-1}$ erhalten werden. $(Q_0)_{jl}$ ist das maximale Gewicht der Pfade von j nach l in $G[(M \lambda^{-1})^d]$, die den kritischen Graphen $G_0[(M \lambda^{-1})^d]$ besuchen. Sei $\{Y_n | n \in \mathbb{N}\}$ die Menge der Vektoren, die definiert werden durch $Y_n = Y_{n-1} M$, und Y_0 sei gegeben, dann gilt

$$\begin{aligned} Y_{nd} &= Y_0 M^{nd} \\ Y_{nd} &= Y_0 Q_0 \lambda^{(n-1)d} M^d \quad \text{falls } n \geq n_0 \\ Y_{nd} &= Y_0 Q_0 \lambda^{nd}. \end{aligned}$$

Daraus folgt

$$Y_0 Q_0 M^d = Y_0 Q_0 \lambda^d,$$

d.h. $Y_0 Q_0$ ist Eigenvektor von M^d .

3.5 Lösung von rekursiven Formeln für diskrete ereignisorientierte Systeme

Mit den bisherigen Vorgaben können nun (max,+)-rekursive Formeln wie folgt berechnet werden.

Theorem 7 :

Sei M eine irreduzierbare Ordnung- d -periodische Matrix mit Eigenwert λ . Seien $Y(n)$ und $Z(n)$ die Lösungen des rekursiven Systems

$$Y(n)=Y(n-1)M, Y(0)=y_0 \quad \text{für } n \geq 0 \quad (\text{unäres System})$$

$$Z(n)=MZ(n+1), Z(N)=z_{N_0} \quad \text{für } n \leq N_0 \quad (\text{binäres System})$$

Es gibt ein $n_0 \in \mathbb{N}$, so daß

$$Y(n+d)=\lambda^d Y(n) \quad \forall n \geq n_0 \quad (28)$$

$$Z(n-d)=\lambda^d Z(n) \quad \forall n \leq N_0 - n_0 \quad (29)$$

wobei man (28) und (29) verbinden kann zu

$$Y(n) \bullet Z(n) = \sum_{i=1}^N Y(n)_i Z(n)_i = y_0 Z(0) \quad \text{für } 0 \leq n \leq N_0 \quad (30)$$

Insbesondere gilt, wenn ein kritischer Kreis in $G(M)$ eine Schleife ist derart

$$Y(n)=\lambda^n y_0 Q_0 \quad \forall n \geq n_0$$

dann hat Projektor Q_0 den Rang 1, d.h. $\forall y_0, y_0', y_0 Q_0$ und $y_0' Q_0$ sind kollinear.

Besitzt der kritische Kreis von $G(M)$ mehr als $d > 1$ Teilpfade, dann gilt

$$Y(n)=\lambda^n y_0 Q_0 \quad \text{für } n=h+kd \text{ mit}$$

$$k \geq \frac{(n_0 - h)}{d} \quad \text{für } h=0, \dots, d-1$$

und $Q_h=M^h Q_0=Q_0 M^h$ und Q_0 ist Projektor mit Rang d .

Besteht der kritische Graph aus K disjunkten Komponenten $C_1 \dots C_k$, wobei C_k von Ordnung 1, $\forall k$, dann gilt

$$Y(n)=\lambda^n y_0 Q_0 \quad \forall n \geq n_0 \text{ für } d=1$$

und Q_0 ist Projektor mit Rang K .

Verallgemeinert ist Q_0 Projektor mit Rang $d_1+\dots+d_k$, wenn d_k die Ordnung von C_k ist.

Damit haben wir alles zur Beschreibung der Struktur von Zustandsformeln für geschlossene deterministische diskrete ereignisorientierte Systeme. Solche Systeme erreichen mitunter ein periodisches Verhalten, wie es in (28) und (29) beschrieben wird, für initiale Situationen in dem Sinne, daß Start- und Endzeiten von Aufgaben in Mengen zwischen n und $n+d$ auf der gleichen Höhe sind bis auf eine Umwandlungszeit λd , für $n > n_0$.

Dieser stabile Zustand wird erreicht, wenn der kritische Graph eine Schleife ist. Das Systemintervall ist dabei der Eigenwert der Systemmatrix.

Die größte Schwierigkeit bezogen auf praktische Anwendungen besteht darin, auf möglichst effiziente Art und Weise λ und Q_0 für eine beliebige Matrix M vernünftiger Größe berechnen zu können. Karp [24] hat hierzu einen effizienten Algorithmus vorgestellt, auf den jedoch nicht weiter eingegangen wird.

4 Anwendungen in der Produktion

Ein typisches Beispiel für ein diskretes ereignisorientiertes System ist ein Herstellungsprozeß. Im folgenden betrachten wir das Modell einer neueren Generation von Herstellungssystemen, die als flexible Fertigungssysteme (FMS) bezeichnet werden. Diese FMS erlauben die simultane Produktion von mehreren Werkstücken. Genauer gesagt werden unterschiedliche Produktionsprozesse in der gleichen Herstellungsumgebung durchgeführt. Diese Technik wird zum Beispiel in der metall-verarbeitenden Industrie eingesetzt.

Im Grunde ist ein FMS eine automatisierte Arbeitsstätte, die aus folgenden Teilen besteht :

- a) Flexible Maschinen, die auf Grund kurzer Einrichtungszeiten effizient von einem Typ auf einen anderen umschalten können
- b) Automatisierte Materialabfertigungssysteme, welche die Werkstücke mit Hilfe von gesteuerten Fahrzeugen zwischen den Maschinen transportieren
- c) Computersteuerung, die für den reibungslosen Ablauf zuständig ist, so daß die Werkstücke zu den richtigen Maschinen gelangen oder auf korrekte Fahrzeuge verladen werden.

Die Werkstücke werden dabei auf Paletten montiert, die eine präzise Positionierung auf den Maschinen gewährleisten. Diese Paletten werden mitsamt dem zugehörigen Werkstück von dem Materialabfertigungssystem transportiert.

Wir werden im folgenden sehen, wie das Verhalten eines FMS durch lineare Funktionen, wie in Kapitel 2, beschrieben werden kann. Eine Analyse zur Optimierung ist dann mit den

Werkzeugen aus Kapitel 3 erreichbar. Darüber hinaus wird gezeigt, daß ein FMS mit den Daten für eine Simulation gesteuert werden kann.

4.1 Beschreibung eines Multiproduktionsprozesses in einer flexiblen Fertigungsumgebung

Sei M die Menge der unterschiedlichen Maschinen und J die Menge der zu produzierenden Teiletypen. Um die Teile in einer gemischten Reihenfolge zu produzieren, müssen die Werkstücke in einer geregelten Abfolge zu den Maschinen gelangen. Diese Eingabefolge kann durch einen endlichen String I beschrieben werden, der aus der Aneinanderreihung diverser Elemente aus J gebildet wird. Sei k_j die Nummer des Teiletyps j in I . Die Mischung der Produkte sei dargestellt durch $(k_1 \dots k_{|J|})$, wobei $|J|$ die Kardinalität von J .

Diese Eingabekontrolle heißt *Grundeingabefolge* (Hitz [17]). Die Gesamteingabefolge kann durch wiederholte Eingabe von I erhalten werden.

Die Abarbeitungsfolge eines Teils i aus I kann als Folge von Maschinen gesehen werden, wobei α_i ein String aus Alphabet M . Jedes Teil kann eine Maschine auch mehrmals in einer Bearbeitungsfolge besuchen. Die Menge der Bearbeitungsfolgen sei $F = \{\alpha_1 \dots \alpha_{|J|}\}$ und jeder Produktionsschritt sei beschrieben durch die Maschine m_{il} , mit l -tem Schritt auf α_i . Ein FMS besteht nicht aus disjunkten Produktionsbereichen, d.h. für jede Partition (J_1, J_2) von J , falls $M(J_1)$ und $M(J_2)$ die Mengen der erforderlichen Maschinen zur Produktion des Teiletyps in J_1 und J_2 , gilt

$$\forall J_1, J_2 : M(J_1) \cap M(J_2) \neq \emptyset$$

Desweiteren nehmen wir an, daß die Reihenfolge der Teile auf den Maschinen als Menge von Strings $G = \{\beta_1 \dots \beta_{|M|}\}$ dargestellt wird, wobei β_m eine Folge von Operationen (i,j) auf Maschine m beinhaltet. Das Finden einer besten Sequenz stellt eine große Aufgabe dar, auf die hier jedoch nicht weiter eingegangen werden soll (Hitz [21], Leveque [13],[26]).

Weiterhin benötigen wir Informationen über die Paletten, also für jeden Teiletyp j gibt es eine Zahl p_j von verfügbaren Paletten. Jede davon kann für ein Teil von Typ j verwendet werden, jedoch nicht für andere Typen. Sei P die Menge der Paletten.

Eine spezielle Maschine dient dazu, die Paletten zu be- und entladen, so daß ein gefertigtes Produkt anschließend zu dieser Maschine zurück muß, damit daraufhin ein anderes Teil auf die Palette montiert werden kann. Die Reihenfolge nach der die Paletten verwendet werden, wird nach dem FIFO-Prinzip bedient. Die Tatsache, daß nur endlich viele Paletten verfügbar sind, erschwert die Voraussage über das Verhalten der geschlossenen Schleifen.

Der beschriebene Multiprodukt-herstellungsprozeß ist deterministisch und kann auch wie folgt als geschlossenes diskretes ereignisorientiertes System betrachtet werden.

4.2 Darstellung eines FMS als geschlossenes diskretes ereignisorientiertes System

Die Menge A der Aktivitäten ist die Menge aller Operationen, die für die Grundabarbeitungsfolge I benötigt werden. Die Ressourcen sind dabei die Maschinen und die Teile in I , d.h. $R = M \cup I$. Der Graph (A, U) sei ein PERT-Flußdiagramm, welches die

Vorrangbeschränkungen zwischen den Operationen beschreibt.

Für die Durchführung der Operation (i,l) müssen folgende Voraussetzungen erfüllt sein.

Das Teilstück i muß auf Maschine m_{il} verfügbar sein, d.h. die vorherige Operation $(i,l-1)$ für $l > 1$ muß beendet sein. Maschine m_{il} ist frei, d.h. die vorangehende Operation vor i im String β_{mil} ist abgeschlossen. Wir bezeichnen sie mit (i',l') , wenn i nicht an erster Stelle in β_{mil} steht.

Daraus folgt, daß höchstens 2 Operationen als Vorgänger einer nicht-initialen Operationen existieren.

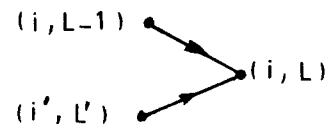


Abb. 2

Jeder Teilpfad $(i, l-1)-(i,l)$ ist gewichtet durch die Dauer $t_{i,l-1}$ der Operation $(i,l-1)$ und der Transportzeit $T_{m,m'}$ zwischen $m = m_{i,l-1}$ und $m' = m_{il}$.

Jeder Teilpfad $(i',l')-(i,l)$ ist gewichtet durch die Dauer $t_{i',l'}$ der Operation (i',l') und der Einrichtungszeit $S_{i'i}$ zum Umschalten zwischen Teil i' und Teil i auf Maschine $m_{il} = m_{i'l'}$. Initiale Operationen sind von der Form $(i,1)$ oder (i,l) , falls i am Anfang eines Strings β_{mil} ist.

Sei Matrix A der Graph (A, U) , wobei U alle Teilpfade der Form $(i',l')-(i,l)$ oder $(i,l-1)-(i,l)$ beinhaltet. Sei Matrix $B = |M \cup I| \times |A|$ für einen zweigeteilten Graph, der die passenden Teile und Maschinen, welche zu den Startoperationen gehören, beschreibt. Ähnlich sei Matrix $C = |A| \times |M \cup I|$ für passende Endoperationen bzgl. der Ressourcen zuständig, d.h. für die letzten Teile im String β_m mit $m \in M$ oder für die letzten Maschinen einer Bearbeitungsfolge.

Wir können nun die Zustandsrepräsentation schreiben als

$$X=XA\oplus BU$$

$$Y=XC$$

U bzw. Y können in zwei Teile zerlegt werden, nämlich Start- bzw. Endzeiten jeder Maschine auf der Menge der Operationen und Ein- bzw. Ausgabezeiten von Produktteilen bezogen auf das Gesamtsystem und wir bezeichnen $U=[U_M, U_I]$ und $Y=[Y_M, Y_I]$. Auf Grund der endlich vielen Paletten, die evtl. in mehreren Eingabefolgen erscheinen können, hängt Vektor $U(n)$ ggf. von mehreren Vektoren $Y(n-i)$ für $i \geq 1$ ab.

Um die Wege der Paletten zu beschreiben, haben wir

$$p_j = q_j k_j + r_j \quad 0 \leq r_j < k_j \text{ mit } j \in J$$

wodurch Paare (q_j, r_j) definiert werden. Die p_j Paletten des Typs j werden für die q_j Eingabefolgen verwendet, wenn $r_j=0$ und q_j+1 , wenn $r_j \geq 1$. Sei $\{i_{jk} | k=1, \dots, k_i\}$ die Reihenfolge der Teile in der Eingabefolge I .

Für Teiletyp j definieren wir zwei Rückkopplungsmatrizen K^j und \bar{K}^j vom Typ $|\mathcal{M} \cup I| \times |\mathcal{M} \cup I|$. Sei T_j die Transportzeit von der letzten Maschine in der Folge mit Teiletypen j zur ersten. Der Rückkopplungseffekt wird dargestellt durch eine $|\mathcal{M} \cup I| \times |\mathcal{M} \cup I|$ -Matrix K , die bis auf einige diagonale Einträge nur mit ε -Einträgen gefüllt ist, $\forall m \in \mathcal{M}$, K_{mm} sei Einrichtungszeit, um von der letzten zur ersten Operation in β_m zu schalten.

Wir erhalten nach einigem Umrechnen folgende rekursive Formel

$$Y(n) = Y(n-1)D \oplus \sum_{j \in J} (Y(n-q_j)D^j \oplus Y(n-q_j-1)\bar{D}^j) \quad (33)$$

wobei

$$D = KBA * C$$

$$D^j = K^j BA * C$$

$$\bar{D}^j = \bar{K}^j BA * C$$

und das Summenzeichen im Sinne von \oplus verwendet wird. $Y(n)$ enthält somit die frühesten Endzeiten der letzten Aufgaben des Strings α_i oder β_m .

4.3 Periodisch fester Zustand des FMS-Verhaltens

Um (33) zu lösen, muß man sie in folgende Form bringen

$$Y(n) = Y(n-1)H \quad (34)$$

und $Y(0)$ muß gegeben sein.

Sei Δ die maximale Verzögerung in (33), dann

$$\Delta = \max(1, \max\{q_j | r_j=0\}, \max\{q_{j+1} | r_j > 0\})$$

und wir können (33) schreiben als

$$Y(n) = \sum_{r=0}^{\Delta} Y(n-r)H^r \quad (35)$$

wobei H^r die Summe der Matrizen D , D^j und \bar{D}^j .

Wenn H^0 nicht die Nullmatrix ist, dann transportiert eine Palette mindestens 2 Teile i und $i' > i$ des gleichen Typs j in der gleichen Abfolge. Desweiteren erhalten wir mit Hilfe des Theorems 1

$$Y(n) = \left(\sum_{r=1}^{\Delta} Y(n-r)H^r \right) H^{0*} \quad (36)$$

genau dann wenn $(H^0)^* = E \oplus H^0 \oplus (H^0)^2 \oplus \dots$ existiert.

Die Existenz von $(H^0)^*$ ist offensichtlich, sobald klar ist, daß $G(H^0)$ keine Kreise enthält. H^0 kann durch Matrix A erzeugt werden, indem man zu $G(A)$ Teilpfade der Form $(i, \bar{l})-(i', 1)$ hinzufügt, wobei \bar{l} die letzte Aufgabe von Teil i und Vorgänger von i' in I ist.

Da i und i' vom gleichen Typ sein müssen, kann i niemals i' passieren, da sie in der gleichen Abarbeitungsfolge stehen, auch dann nicht, wenn sie von verschiedenen Paletten bewegt werden. Die Teilpfade der zurückkehrenden Paletten erzeugen jedoch keine Kreise in $G(A)$, da keine Teilpfade der Form $(i', l')-(i, l)$ vorhanden sind.

Man kann daher sagen, daß H^0 immer existiert, solange keine Aufgabe gleichzeitig Vorgänger und Nachfolger einer anderen ist.

Mittels

$$l(n)=[Y(n), Y(n-1), \dots, Y(n-\Delta+1)]$$

erhalten wir Matrix H der Form

$$H = \left[\begin{array}{c|c} \begin{matrix} H^1 \cdot (H^0)^* \\ H^2 \cdot (H^0)^* \\ \dots \\ H^{\Delta-1} \cdot (H^0)^* \\ H^\Delta \cdot (H^0)^* \end{matrix} & \begin{matrix} E & & \\ & \epsilon & \\ & & E \\ \epsilon & & & \epsilon \end{matrix} \end{array} \right]$$

wobei E die $|M \cup I| \times |M \cup I|$ -Einheitsmatrix der $(\max, +)$ -Algebra.

In (36) haben wir keine minimale Darstellung und für Berechnungszwecke ist es sinnvoll rekursive Formeln zu betrachten, bei denen die Zustandsvariablen den frühesten Zeitpunkt der Verfügbarkeit von Maschinen und Paletten beinhalten. Dabei kann die Größe des rekursiven Systems $|M \cup P| \times |M \cup P|$ anstelle von $\Delta \cdot |M \cup I| \times \Delta \cdot |M \cup I|$, wie in (36), annehmen. Nehmen wir für die Menge der Ressourcen Maschinen und Werkstücke, dann müßte die Menge der Aufgabe vergrößert werden, um auch noch die Δ Grundeingabefolgen zu beinhalten. Wir müssen also in jedem Fall eine minimale Eingabefolge finden, die alle Paletten enthält.

Es läßt sich sagen, daß mindestens $|M \cup P|$ unabhängige Variablen in $Y(n)$ vorkommen. Die zugehörigen Zeilen und Spalten von H definieren eine Teilmatrix \hat{H} , deren Graph $G(\hat{H})$ stark-verbunden ist. Sei V die zugehörige Variablenmenge und jeder Eintrag $Y_h(n)$ mit $h \notin V$ kann dargestellt werden als Funktion der Einträge $l \in V$, d.h. die stark-verbundenen Komponenten von $G(H)$ sind $G(\hat{H})$ und isolierte Knoten der Menge $\{h \in V\}$. V kann definiert werden, durch Überprüfen der Einträge in $Y(n)$, die die zugehörigen Zeiten der frühesten Verfügbarkeit von Maschinen und Paletten eines betrachteten Bereichs beinhalten. Diese Eingrenzung ist auf Grund des Aufbaus der FMS möglich, da FMS nicht in unabhängige Produktionsbereiche aufgeteilt werden können.

Sei $\lambda \psi = \psi \hat{H}$, wobei ψ der Eigenvektor von \hat{H} . λ ist das Durchschnittsgewicht von kritischen Pfaden in $G(\hat{H})$, somit auch in $G(H)$, da $G(\hat{H})$ der zusammenhängende maximale Untergraph von $G(H)$ ist. Wenn \hat{H} Ordnung- d -periodisch ist, dann ist die Systemfrequenz d_λ , die sich über d Eingabefolgen erstreckt. Die durchschnittliche Produktionsrate ist $1/\lambda$. Ist θ_m das Arbeitspensum einer Maschine m bzgl. einer Grundeingabefolge, d.h. die Summe der Arbeitsschritte auf m in A , dann ist

$$UT_m = \theta_m / \lambda$$

die Auslastung der Maschine m .

Der leere Startzustand des FMS wird durch die Startzeiten von Maschinen und Paletten charakterisiert, aus denen der Startvektor $Y(0)$ berechnet werden kann. Die frühesten Startzeiten von Aufgaben in $\{Y(n+i) | i=0, \dots, d-1\}$, für n groß genug, werden bis auf einen konstanten Anteil von Vektoren $\{Y_0 Q_h | h=0, \dots, d-1\}$ beschrieben. Der absolute Wert hängt dann von der gewählten Startzeit des Systems ab.

Beachte, daß, wenn der kritische Graph eine Schleife ist, H Ordnung-1-periodisch ist, der

Spektralprojektor Q_0 den Rang 1 besitzt und der feste Zustand des FMS nicht abhängig ist vom Startzustand. Die frühesten Startzeiten sind dann durch Eigenvektoren ψ gegeben.

Analog kann das binäre Eigenvektorproblem $\lambda\zeta = \hat{H}\zeta$ mit dem gleichen Eigenvektor ζ gelöst werden. Mit Hilfe von Q_0, \dots, Q_{d-1} erhalten wir die letzten Endzeiten von Aufgaben.

Das duale rekursive System von (36) ist

$$Z(n) = HZ(n+1)$$

und sei $Z(N)$ gegeben für beliebige N .

Um nun die letzten Endzeiten im festen Zustand durch $Y(0)$ zu erhalten, können wir folgendes

$$Z(N) = -[Y(0)Q_0]^T \quad (37)$$

mit T für transponiert, übernehmen.

Dies basiert auf der Tatsache, daß Aufgaben auf kritischen Kreisen keine Leerlaufzeit haben, so daß die früheste Startzeit der letzten Startzeit entspricht. $\{-Q_h Z(N) | h=0, \dots, d-1\}$ enthält bis auf einen konstanten Anteil, die letzten Endzeiten von Werkstücken und Maschinen am Ende ihrer Bearbeitung. Nur die Einträge bzgl. nicht-kritischer Aufgaben werden verändert in Bezug auf ihre zugehörigen Einträge der frühesten Startzeiten von Vektoren $Y(n)$, für große n .

Die Leerlaufzeiten erhält man, indem man die gleiche Basiszeit für kritische Aufgaben wählt, so daß die Folgen $\{Y(n)\}$ und $\{Z(n)\}$ übereinstimmen und man dann die Zeiten für früheste und letzte Verfügbarkeit von Maschinen und Werkstücken vergleicht. Dies kann mittels PERT-Berechnungen durchgeführt werden.

Ein fester Zustand der Aufgaben auf einem kritischen Kreis kann erreicht werden innerhalb von $\Delta \cdot |M \cup I|$ ($=n^b$ Zeilen in H) Grundeingabefolgen.

Darüber hinaus kann die instabile Phase umgangen werden, indem man den stetigen Zustand berechnet und dann das FMS direkt in diesen Zustand versetzt, auf Grund der Kenntnis über korrespondierende früheste Startzeiten der Aktivitäten.

4.4 Hinzufügen von Paletten am Beispiel des Umlaufbetriebs

Das Hinzufügen von Paletten zur Systemstruktur bringt ein Anwachsen von Δ , d.h. die Ordnung von H erhöht sich. Wenn $\forall j p_j > 2k_j$, dann gibt es keine Leerlaufzeit $r=1$ der Paletten. Daher gilt, für p_j groß genug, daß $q = \min q_j > 2$ und daß in H die Untermatrizen H^2, \dots, H^{q-1} nur ε -Einträge enthalten. Desweiteren ist $H^1 = D = KBA^*C$, d.h. es sind nur Kreise enthalten, die der Rückführung zu Maschinen dienen. Beachte, daß für $q \geq 1$, $(H^0)^* = E$, keine 0-Verzögerungen existieren, somit kann es weggelassen werden. Wir nehmen im weiteren an, daß das FMS eine Umlaufbetriebsstruktur besitzt, d.h. die Menge der Arbeitsabfolgen bezieht sich auf eine einzelne totale Ordnung von Maschinen, daher unterscheiden sich solche Folgen nur dadurch, daß evtl. einige Maschinen ausgelassen werden.

In diesem Fall sind Kreise, für $q \geq 2$, der Form

a) Kreise in H^1 , sind nur durch die Einrichtungs- und Bestückungszeiten der Maschinen gewichtete Schleifen. Wenn m Vorgänger von m' in der Arbeitsabfolge, dann $(H^1)_{m',m} = \varepsilon$, so daß die einzigen Kreise in $G(H^1)$ Schleifen sind.

b) Kreise nicht in H^1 , enthalten Teilpfade mit Gewicht e und andere aus H^r , für $r \geq q$, oder aus H^1 .

Da die Anzahl der Paletten für jeden Typ steigt, vergrößern sich auch Δ und q , sowie die Länge der Kreise von Typ B . Dadurch verringert sich auch das Durchschnittsgewicht und unter Umständen wird eine Schleife aus H^1 zu einem kritischen Kreis. Diese Schleife korrespondiert dann mit der Aufgabenfolge der kritischen Maschinen.

Theorem 8 :

Für ein FMS mit einer Umlaufbetriebsstruktur und einer beliebigen festen Abfolge auf Maschinen gibt es eine endliche Verteilung von Paletten, so daß das FMS in stabilem Zustand mit voller Auslastung, abgesehen von den Einrichtungszeiten, arbeitet.

Eine Optimierung der Folgen auf Maschinen führt zu einer Reduzierung der Palettenanzahl, was nach Hitz [21] zu einer Produktivitätssteigerung erforderlich ist.

Sobald der Herstellungsprozeß nicht mehr die Umlaufbetriebsstruktur besitzt, kann eine Maschine nicht mehr voll ausgelastet werden, für einige bestimmte Abläufe an Maschinen. Dies verdeutlicht folgendes

Beispiel 2 :

2 Maschinen, 3 Werkstücke

		Abläufe		
		1	2	3
Abfolge auf				
$m_1=1,2,3=\beta_1$	m_1	3	4	7
$m_2=1,2,3=\beta_2$	m_2	5	6	3

Abarbeitungsfolgen :

Werkstücke 1 und 3 : 1, 2; Werkstück 2 = 2, 1.

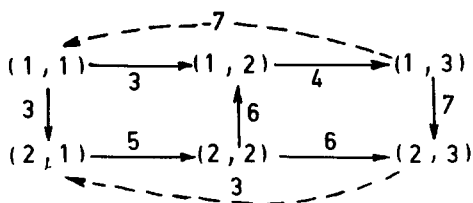


Abb. 3

Der Graph des Herstellungsprozesses ist $(i,j)=$ Maschine i , Werkstück j . Die gestrichelten Linien symbolisieren die Rückführungswege. Matrix $D_M=K_M B_M A * C_M$ ist dann

$$\begin{bmatrix} 25 & 23 \\ 22 & 25 \end{bmatrix}, \text{ wobei } B_M, C_M, K_M, D_M$$

Untermatrizen von B, C, K, D bezeichnen, deren Zeilen und/oder Spalten zu Maschinen korrespondieren.

Der Eigenwert zu H ist immer $\lambda \geq 25$. Der kritische Graph von D_M ist $G(D_M)$, nämlich 2 kritische Schleifen und 1 kritischer Kreis. D_M ist Ordnung-1-periodisch, aber da die Beladung der Maschine nur 14 auf der Grundfolge $I=\{1,2,3\}$ beträgt, kann die Auslastung nicht größer sein als $14/25$ für beliebige Maschinen, selbst wenn eine große Anzahl von Paletten zu jedem Werkstück zugewiesen wird. Durch Veränderung der Sequenzen

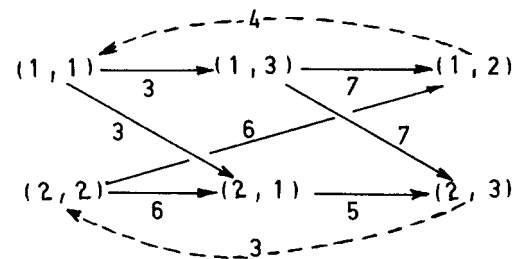


Abb. 4

auf $m_1=1,3,2=\beta_1$
 $m_2=2,1,3=\beta_2$

erhalten wir

$$D_M = \begin{bmatrix} 14 & 13 \\ 10 & 14 \end{bmatrix}.$$

Der kritische Graph von D besteht aus zwei Schleifen, der durch die Beladung der Maschinen gewichtet wird. Nach dem Theorem kann nun die volle Auslastung beider Maschinen durch eine endliche Verteilung der Paletten realisiert werden. Durch die Aufgabenbetriebsstruktur hängt die

einschränkende Produktionsrate nicht mehr allein von den Abfolgen auf Maschinen ab, sondern auch von der Art und Weise, wie das System gestartet wurde. Betrachtet man zum Beispiel das gleiche 2-Maschinen-System, welches jedoch nur Teile von Typ 2 und 3 im Verhältnis 1 zu 1 fertigt, über einer 2-Teile-Eingabefolge. Obwohl nur eine Folge auf jeder Maschine abgearbeitet werden kann, mit $\beta_i=1,2 \forall i$, kann dies dennoch auf 4 Arten durchgeführt werden.

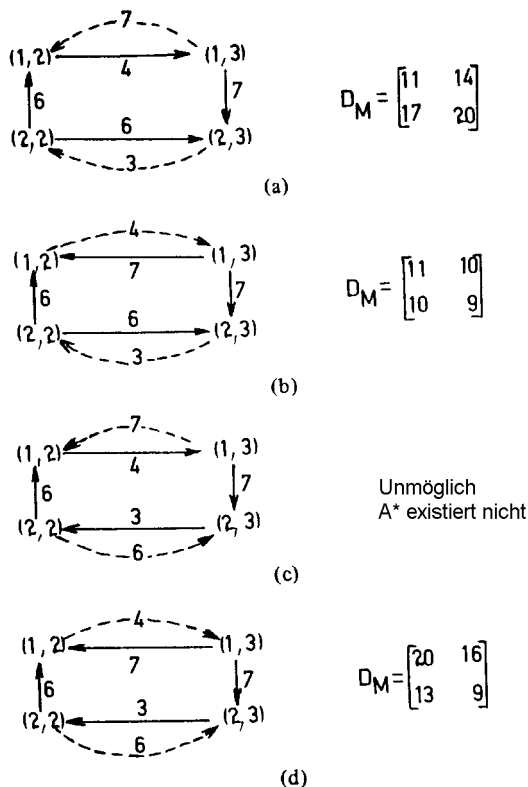


Abb. 5

Konfiguration (c) ist widersprüchlich, da der Graph zur Matrix A ein Zyklus ist und somit keine Aussage über die erste Aktivität gemacht werden kann.

Bei Konfiguration (a) ist die erste Aktivität die Operation (2,2). Operation (1,3) kann nur gestartet werden, wenn Operation (1,2) beendet ist. Matrix D_M zeigt die maximale Auslastung von Maschine 1, $11/20$, und von Maschine 2, $9/20$. Konfiguration (d) drückt dies ebenfalls aus.

In Konfiguration (b) kann die Operation (1,3) parallel zur Operation (2,2) gestartet werden. Der kritische Kreis ist eine Menge von

Operationen auf der kritischen Maschine 1. Daher, kann Maschine 1 mit einer endlichen Verteilung von Paletten voll ausgelastet werden.

Derartige Effekte treten bei der Umlaufbetriebsstruktur nicht auf, da eine natürliche Ordnung der Maschinen erfolgt, so daß D_M immer eine trianguläre Matrix ist mit diagonalen Einträgen, die die Beladung der zugehörigen Maschine beinhalten.

Im Gegensatz dazu können bei der Aufgabenbetriebsstruktur einige Kreise in D_M durch die Reihenfolge der Werkstücke auf den Maschinen und durch die Wahl der Startaktivitäten in jeder Folge erzeugt werden.

Hierzu können wir eine schwache Version des Theorems 8 für die Aufgabenbetriebsstruktur angeben. Die volle Auslastung der Maschinen kann erreicht werden, indem für wenigstens eine Menge von Teilfolgen auf Maschinen eine endliche Verteilung der Paletten, zusammen mit der Menge der Startaktivitäten, erfolgt.

Jedoch bei der Umlaufbetriebsstruktur gilt, falls bei einem gegebenen Abarbeitungsverfahren und einer Verteilung der Paletten, die Maschinen nicht voll ausgelastet sind, also $\lambda > \max_m \theta_m$, dann ist der kritische Kreis von $G(H)$ vom Typ B , d.h. er enthält Knoten, die zu Paletten gehören, deren Verfügbarkeit kritisch ist. Diese Paletten findet man durch Überprüfung der zugehörigen Zeilen in H . Um die Maschinenauslastung zu verbessern, muß der kritische Kreis eliminiert und anschließend eine Palette des zugehörigen Teiletyps hinzugefügt werden.

Der Vorgang der schrittweisen Suche, nach dem kritischen Kreis und dann nach dem kritischen Palettentyp, soll jedoch hier nicht weiter vertieft werden.

5 Zusammenfassung

Die hier gezeigte pseudolineare Algebra von diskreten ereignisorientierten Systemen macht diese den gewöhnlichen linearen dynamischen Systemen entsprechend. Mit Hilfe von Eigenwerten und -vektoren, sowie den Verhaltensweisen des Systems konnten die Parallelen in der $(\max, +)$ -Algebra verdeutlicht werden und man darf davon ausgehen, daß man noch weitere Eigenschaften auf diese Modelle übertragen kann.

Literaturverzeichnis

- [1] R. Bellman, A. O. Esogbue, and I. Nadeshima, *Mathematical Aspects of Scheduling and Applications*. New York: Pergamon, 1982.
- [2] C. Berge, *The Theory of Graphs*. London: Mettuen, 1966.
- [3] C. Berge and A. Gouila-Houri, *Programming Games and Transportation Networks*. New York: Wiley, 1966.
- [4] B. A. Carre, "An algebra for network routing problems," *J. Inst. Math. Appl*, vol. 7, pp. 273-294, 1971.
- [5] J. B. Cavaille and D. Dubois, "Heuristic methods based on mean-value analysis for flexible manufacturing systems performance evaluation," in *Proc. IEEE Conf. Decision Contr.*, Orlando, FL, 1982.
- [6] G. Cohen, D. Dubois, J. P. Quadrat, and M. Viot, "Analyse du comportement périodique de systèmes de production par la théorie des diodes," INRIA, Le Chesnay, France, Rapport de Recherche 191, 1983.
- [7] R. A. Cuninghame Green, "Describing industrial processes and approximating their steady-state behaviour." *Opt. Res. Quart.*, vol.13' pp. 95-100, 1962 '
- [8] "Projections in a minimax algebra," *Math. Program.*, vol. 10, pp. 111-123, 1976.
- [9] ---, *Minimax Algebra* (Lecture Notes in Economics and Mathematical Systems, vol. 166). New York: Springer-Verlag, 1979.
- [10] Y. Dallery and R. David, "A new approach based on operational analysis for flexible manufacturing systems performance evaluation," in *Proc. 22nd IEEE Conf. Decision Contr.*, San Antonio, TX, 1983.
- [11] J. Denning and J. P. Buzen, "The operational analysis of queuing network models," *Comput. Surveys*, vol. 10, pp. 225-261, 1978.
- [12] D. Dubois and K. Stecke, "Using Petri nets to represent production processes," in *Proc. 22nd IEEE Conf. Decision Contr.*, San Antonio, TX, 1983, pp. 1062-1067.
- [13] J. Erschler, D. Leveque, and F. Roubellat, "Periodic loading of flexible manufacturing systems," in *Proc. IFIP Congress "APMS82"*, Bordeaux, France, 1982, pp. 327-339.
- [14] G. S. Fishman, *Principles of Discrete-Event Simulation*. New York:Wiley, 1978.
- [15] "Flexible manufacturing systems, in *Proc. Ist Int. Conf.*, Brighton, U.K., Oct. 1982.
- [16] M. Gondran, "Les éléments p-réguliers dans les diodes," *Discrete Math.*, vol. 25, pp. 33-39, 1979.
- [17] M. Gondran and M. Minoux, *Graphes et Algorithmes*. Paris,France: Eyrolles, 1979.
- [18] , "Valeur et vecteurs propres dans les diodes et leur interprétation en théorie des graphes," *Bul. Dir. Et. et Rech. E.D.F.*, série C, pp. 25-41, 1977.
- [19] M. P. Groover, *Automation, Production Systems and ComputerAidedManufacturing*. Englewood-Cliffs, NJ: Prentice-Hall, 1980.
- [20] R. R. Hildebrandt, "Scheduling flexible machining systems using mean value analysis," in *Proc. IEEE Conf Decision Contr.*, Albuquerque, NM, 1980, pp. 701-706.
- [21] K. L. Hitz, "Scheduling of flexible flowshop," Mass. Inst. Technol., Cambridge, MA, M.I.T. Rapport LIDS-R-879, Mar. 1979.
- [22] Y. C. Ho and C. Cassandras, "A new approach to the analysis of discrete event dynamic systems," *Automatica*, vol. 19, pp. 149-168, 1983.
- [23] J. R. Jackson, "Job-shoplike queuing systems," *Management Sci.*, 01. 20, pp. 20, 131-14j, 1963.v
- [24] R. M. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete Math.*, vol. 23, pp. 309-311, 1978.
- [25] J. Kimemia, S. B. Gershwin, and D. Bertsekas, "Computation of production control policies by a dynamic programming technique in analysis and optimization of systems" (Lecture Notes in Control and Information Sciences, vol. 44), A. Bensoussan and J. L. Lions, Eds. New York: Springer-Verlag, 1982, pp. 243-259.
- [26] D. Leveque, "Lancement périodique de produits dans un atelier flexible," Those Doet.Ingenieur, LAAS-CNRS, Toulouse, 1982.
- [27] C. V. Raniamoorhty and G. S. Ho, "Performance evaluation of asynchronous concurrent systems using Petri nets," *IEEE Trans. Software Eng.*, vol. SE-6, pp. 440-M9, 1980.
- [28] C. Ramchandani, "Analysis of asynchronous concurrent systems by Petri Nets." Mass. Inst. Technol., Cambridge, MA, M.I.T. Rep. MAC TR^120, 1974.
- [29] R. Reiter, "Scheduling parallel computations," *J. Ass. Comput. Mach.*, vol. 15, pp. 590-599, 1968.
- [30] G. Secco-Suardo, "Optimization of a closed network of queues for complex material-handling systems," Mass. Inst. Technol., Cambridge, MA. M.I.T. Rep. ESL-FR-834-5, 1978.
- [31] J. G. Shanthikumar and J. Buzacott, "Open queuing networks models of dynamic job-shops, " *Int. J. Produc. Res.*, vol. 19, pp. 255-266,1981.
- [32] J. J. Solberg, "Analytical performance evaluation of flexible manufacturing systems," in *Proc. IEEE Conf. Decision Contr.*, San Diego, CA, 1979, pp. 640-644.
- [33] R. Suri, "Robustness of queuing network formulae," *J. Ass.Comput. Mach.*, vol. 30, pp. 564~594, 1983.
- [34] D. R. Shier, "Iterative methods for determining the k shortest paths in a network," *Networks*, vol. 6, pp. 205~229, 1976.
- [35] Cohen et al, „A Linear-System-Theoretic View of Discrete-Event Processes and Its Use for Performance Evaluation in Manufacturing“, *IEEE Transactions on Automatic Control*, vol. AC-30, No. 3, March 1985
- [36] G. Klaus und H. Liebscher, „Systeme, Informationen, Strategien“, *VEB Verlag Technik*, 1974
- [37] U. Manber, „Introduction to Algorithms - A Creative Approach“, *Addison-Wesley*, 1989, ISBN 0-201-12037-2
- [38] A. Schöne, „Prozeßrechnungssysteme - Aufbau und Programmierung von Prozeßrechnern / Grundlagen und Verfahren ihrer Anwendung“, *Carl Hanser Verlag*, 1981, ISBN 3-446-12557-4
- [39] P. H. Starke, „Petri-Netze“, *VEB Deutscher Verlag der Wissenschaften*, 1980
- [40] G. Wunsch, „Handbuch der Systemtheorie“, *Oldenbourg* 1986, ISBN 3-486-20017-8