

---

# CONTROLE DE TRANSITÓRIOS EM ESCALONAMENTO DE JOB SHOP CÍCLICO

Adriana Backx Noronha\*  
backx@usp.br

Rafael Santos Mendes†  
rafael@dca.fee.unicamp.br

\*Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo, FEA/USP  
Campus Ribeirão Preto, Av. dos Bandeirantes, 3900, CEP: 14.040-900, Ribeirão Preto, SP, Brasil.

†Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas  
DCA/FEEC/UNICAMP – C.P. 6101 – 130881-970 Campinas, SP, Brasil.

---

## ABSTRACT

The job shop problem is particularly important among the scheduling problems found in production systems, being characterized by  $N$  parts which are processed by  $M$  machines or work stations. If the processing times in each machine are determined and known a-priori and the parts are produced repetitively, then the problem is called deterministic cyclic job shop scheduling. In this case a steady state solution, corresponding to a set point for the production system can be calculated. However, real systems are subjected to perturbations like machine failures, lack of raw material etc., that can affect the steady state solution. In this context, this work is concerned with the determination of a control law i.e. a law for the determination of the starting time of each operation in the subsequent cycles such that the set point (reference solution) be reestablished and maintained. The proposed control law is based in max-plus algebra concepts. The starting times of the operations are calculated from the starting times of the preceding operations and from a set of matrices defined in max-plus algebra.

**KEYWORDS:** Discrete event systems, cyclic job shop, max-plus algebra, control laws, manufacturing systems.

## RESUMO

Os problemas encontrados no escalonamento da produção são classificados em diversas categorias, dentre as quais destaca-se o problema de job shop, onde  $N$  peças são processadas por  $M$  máquinas ou estações de trabalho. O problema de escalonamento job-shop cíclico determinístico é definido como aquele em que os tempos de processamento são fixos e conhecidos a-priori e as peças são produzidas de maneira repetitiva. Para este problema uma solução em regime pode ser determinada. Porém, o sistema pode estar sujeito a distúrbios na solução de regime. Neste contexto, buscou-se neste trabalho encontrar uma lei de controle, ou seja, uma lei de determinação dos instantes de início de cada peça nos ciclos subsequentes, capaz de restabelecer e manter a solução de referência. A lei de controle é baseada em conceitos da álgebra max-plus, e consiste em determinar os instantes de início das operações a partir dos instantes de início das operações precedentes e de um conjunto de matrizes definido em max-plus.

**PALAVRAS-CHAVE:** Sistemas a eventos discretos, job shop cíclico, álgebra max-plus, lei de controle, sistemas de manufatura.

---

Artigo submetido em 20/12/00

1a. Revisão em 21/05/01; 2a. revisão em 01/03/02

Aceito sob recomendação do Ed. Assoc. Prof. Paulo E. Miyagi

# 1 INTRODUÇÃO

Os problemas de escalonamento estão amplamente difundidos em uma grande variedade de atividades industriais ou tecnológicas e têm adquirido um caráter notório, mediante as exigências de aumento da produção de bens, em uma escala de tempo progressivamente menor. Nesse aspecto, pequenas falhas em um dos percursos de produção, podem desencadear desde pequenos prejuízos de ordem cronológica até a paralisação completa de todo um setor de produção.

Esses problemas encontrados no escalonamento da produção são classificados em diversas categorias, dentre as quais destaca-se o problema de *job shop*, onde  $n$  peças são processadas por  $m$  máquinas ou estações de trabalho. Cada peça consiste de um conjunto específico de operações que são processadas de acordo com uma ordem de precedência preestabelecida. Desde que os tempos de processamento das peças nas máquinas sejam fixados e conhecidos *a priori* e as mesmas peças sejam produzidas repetitivamente tem-se o problema de escalonamento *job shop* cíclico determinístico.

Partindo do princípio que os *jobs* possuem tempos de processamento determinados e que estes são processados indefinidamente ou por um longo período de tempo, é possível encontrar um escalonamento cíclico ótimo ou quase ótimo para o problema. Entretanto, nas situações práticas o problema está sujeito a restrições de continuidade tais como quebras de máquina e falta de matéria-prima. Neste contexto se insere a idéia de um planejamento reativo.

O planejamento reativo é necessário, pois visa garantir a continuidade da produção. Pode-se citar vários tipos de planejamento reativo para problema cíclicos.

Uma das formas mais simples de planejamento reativo seria a utilização de regras de prioridade, tais como FIFO – *First in First Out*, SPT – *Shortest Processing Time*, quando alguma perturbação é imposta ao sistema (Udo, 1994; Caskey & Storch, 1996; Randhawa & Zeng 1996; Xu & Randhawa, 1996; Pierreval & Mebarki, 1997; Hothaus & Ziegler, 1997). Devido à sua simplicidade, geralmente as regras de prioridade são implementadas mais facilmente, conquanto seu uso não necessariamente garanta o retorno à solução de referência.

Bowman & Muckstadt (1993) modelam o escalonamento cíclico como uma cadeia de Markov e utilizam a Teoria Ergódica para analisar a performance de escalonamentos cíclicos em ambientes com quebras de máquina, perda de produção e outras fontes de variabilidade. Eles desenvolveram o conceito de tarefa cíclica crítica como uma

extensão de tarefas críticas em redes PERT e mostram que tarefas cíclicas críticas podem e deveriam ser usadas para conduzir a administração de escalonamentos cíclicos.

Groth & Mendes (1997) consideram o problema de modelar e controlar uma classe de problema de escalonamento *job shop*, onde assumem um tempo de processamento entre as operações de um *job* igual a zero (zero-wait). Para este problema os autores propõem um modelo matemático baseado na Teoria de Dióides (álgebra min-max), escrevendo equações dinâmicas nesta estrutura. Os autores, considerando que todos os *jobs* já tenham iniciado, propuseram leis de controle visando a minimização do número médio de iterações por unidade de tempo. Estas leis podem ser utilizadas também quando situações não previstas ocorrem durante a execução do planejamento.

Zhang & Graves (1997) examinam o comportamento do escalonamento cíclico em um ambiente estocástico caracterizado por falhas de máquina que podem atrasar a execução de tarefas e então podem causar um desvio ao escalonamento cíclico preestabelecido. Os autores buscaram entender o comportamento de um escalonamento cíclico sob incertezas com a finalidade de encontrar escalonamentos cíclicos que sejam menos interrompidos pela ocorrência de falhas de máquinas. Duas hipóteses básicas são consideradas pelos autores: 1°) indiferente do histórico das falhas das máquinas e os atrasos consequentes, as operações são sempre processadas na ordem descrita; 2°) uma operação nunca se inicia antes do tempo de escalonamento prefixado, ainda que a máquina possa estar ociosa.

Em suma, o planejamento reativo do problema de escalonamento *job shop* cíclico determinístico, isto é, onde é possível encontrar uma solução estável, tem por objetivo oferecer formas de controle, quando uma perturbação ocorre, devido a perturbações imprevisíveis, visando restabelecer e manter a solução de referência.

Neste trabalho propõe-se uma metodologia para o planejamento reativo de um *job shop* cíclico. A metodologia é baseada numa lei de controle para o sistema perturbado que garanta seu retorno ao escalonamento cíclico conhecido. Uma lei de controle consiste em uma lei de determinação dos instantes de início de cada *job* nos ciclos subsequentes, capaz de restabelecer e manter o escalonamento cíclico conhecido, assegurando deste modo a produção em uma fábrica e com isso permitindo o cumprimento total ou parcial dos seus compromissos de entrega, sem desperdício de materiais e nem perda de produtos.

Na proposta deste trabalho, a lei de controle deve basicamente, calcular os instantes de início de cada operação (variáveis de decisão) a partir de informações sobre início das operações passadas (variáveis medidas), do escalonamento cíclico preestabelecido e do conhecimento sobre o job shop (modelos da planta). A figura 1 ilustra a analogia entre a abordagem proposta neste trabalho e os conceitos utilizados na teoria e prática de controle.

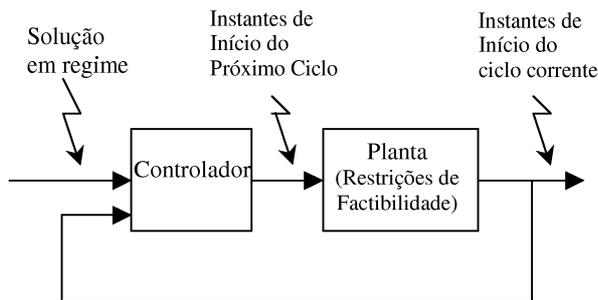


Figura 1: Analogia entre o esquema de controle tradicional e a abordagem proposta

O papel do controlador neste esquema é o de definir os instantes de início das operações que deverão se iniciar no futuro imediato. Quando o sistema opera sem perturbações esta tarefa é trivial, dado que se considera o escalonamento cíclico conhecido. Quando ocorrem perturbações, mesmo após a reposição da suas causas, os instantes de início do ciclo corrente deixam de corresponder aos instantes definidos pelo escalonamento cíclico preestabelecido (em geral uma perturbação interfere diferenciadamente nas diversas operações). A informação relativa a estes instantes de início perturbados será considerado no contexto deste trabalho como a “condição inicial do sistema”.

De modo geral, define-se um vetor  $\mathbf{t}_k$  cujos componentes são os instantes de início das operações, dependente da variável livre  $k$ , associada ao número de ciclos do sistema. O vetor  $\mathbf{t}_0$  é a condição inicial. O controle do sistema é efetuado através de um conjunto de matrizes  $\mathbf{A}_l$ ,  $l \in \{1, \dots, m\}$  cujo cálculo é definido na seção 3.2. Este conjunto de matrizes define o controlador.

Os resultados apresentados neste trabalho são expressos numa álgebra diferente da usual, conhecida como álgebra max-plus. Esta álgebra tem sido aplicada à modelagem de sistemas a eventos discretos se limitando contudo a sistemas em que não haja conflitos pelo uso de recursos (Baccelli *et alii*, 1992). Esta restrição é uma importante limitação à utilização da abordagem max-plus e os resultados propostos no presente trabalho constituem um esforço no sentido de superá-la. A literatura apresenta

resultados fundamentais relativos à convergência de expressões do tipo  $\lim_{k \rightarrow +\infty} \mathbf{A}^k$  onde  $\mathbf{A}$  é uma matriz na álgebra max-plus e  $k$  é um inteiro. Estes resultados tem importante significado prático (Baccelli *et alii*, 1992). Neste trabalho são propostos novos resultados relativos à convergência de produtos de matrizes que estendem os resultados clássicos e ampliam significativamente o espectro de aplicações desta teoria.

Este trabalho é organizado da seguinte forma: na seção 2 descreve-se o problema de *job shop* cíclico a ser considerado, na seção 3 apresenta-se um resumo dos conceitos teóricos que são utilizados no desenvolvimento deste trabalho (álgebra max plus); na seção 4 propõe-se o modelo matemático para o sistema reativo, na seção 5 mostra-se como construir o conjunto de matrizes de decisão e um exemplo é apresentado na seção 6. Finalmente as conclusões são apresentadas na seção 7.

## 2 JOB SHOP CÍCLICO

Em um problema clássico de escalonamento, um conjunto de tarefas é executado uma vez, e uma solução é encontrada buscando-se otimizar um critério, por exemplo, a minimização da duração total de fabricação, minimização dos atrasos de fabricação, etc. Em um problema de escalonamento cíclico, as tarefas, chamadas de tarefas genéricas, são executadas infinitas vezes e, todas as ocorrências de uma mesma tarefa genérica devem satisfazer às mesmas restrições. Ao resolver o problema de escalonamento cíclico busca-se maximizar o número médio de iterações por unidade de tempo, isto é, minimizar a duração do ciclo.

O problema de escalonamento cíclico tem numerosas aplicações, tais como na informática (cálculo repetitivo sobre arquiteturas paralelas), no domínio da produção (job shop repetitivo), no estabelecimento de um plano ou roteiro (utilização periódica de recursos) etc. Ambrósio (1999) apresenta uma aplicação do problema de escalonamento cíclico para o despacho de trens que fazem sempre o mesmo horário.

Como qualquer problema de escalonamento, o *job shop* cíclico (JSC) é caracterizado por um conjunto de operações genéricas, que são executadas infinitas vezes ou durante um longo período de tempo. Estas operações são não reentrantes, isto é, duas ocorrências de uma mesma operação genérica não podem se sobrepor. Além disso, uma vez que uma operação comece a ser processada, não poderá ser interrompida para a execução de outra operação. O conjunto de todas as operações pode ser particionado em subconjuntos disjuntos e mutuamente exclusivos que correspondem aos *jobs* (Conway *et alii*,

1967). Cada *job* consiste de um conjunto de operações que devem ser realizados em alguma ordem.

O problema de escalonamento *job shop* cíclico, objeto de interesse deste trabalho, consiste de  $n$  *jobs* e  $m$  máquinas. Cada *job* corresponde a um conjunto de operações que devem ser realizadas em alguma ordem que é chamada de relação de precedência. A cada operação associa-se uma única máquina, um tempo de processamento fixo e uma única operação sucessora no *job*. O problema de *job shop* cíclico pode ser formulado por programação linear mista, buscando maximizar o número médio de iterações por unidade de tempo ( $\tau$ ) ou minimizar o tempo de ciclo ( $\alpha$ ).

A solução do problema de *job shop* cíclico foi estudada por vários autores. Uma primeira tentativa de escalonamento cíclico foi proposta por Graves *et alii* (1983) para o escalonamento de *job shops* que produzem muitos produtos em uma proporção fixada por um longo período de tempo. Um *job* pode passar em uma máquina uma ou mais vezes. Para o caso quando somente um produto é produzido (isto é, quando todos os *jobs* são idênticos), Graves *et alii* (1983) definem um escalonamento ótimo para um comprimento de ciclo dado. Os autores apresentam um método heurístico que parece ser efetivo em encontrar bons escalonamentos.

Hillion & Proth (1989) provaram que se é dada uma ordem cíclica das tarefas em cada processador então o problema torna-se um problema de escalonamento cíclico básico e, por essa razão, pode ser resolvido em tempo polinomial. Roundy (1992) considera o problema de encontrar escalonamento cíclico para um problema de *job shop* no qual todos os *jobs* são idênticos. O autor considera simultaneamente dois critérios de otimização, a saber, o tempo de ciclo e o tempo de fluxo e, mostra que o problema de minimizar o tempo de fluxo para um dado tempo de ciclo é NP-difícil.

Hanen (1994) em seu estudo sobre escalonamento cíclico propôs uma formulação em programação linear mista para o problema de escalonamento cíclico com restrições disjuntivas chamado de *job shop* recorrente. Portugal (1999) e Portugal *et. alii.* (2000) fez modificações nesta formulação para o problema de *job shop* cíclico com *jobs* distintos, visando estabelecer um equilíbrio na produção de *jobs* ao longo do tempo.

Esta última abordagem será adotada neste trabalho como formulação para o problema de funcionamento do *sistema job shop* cíclico em regime permanente. O detalhamento da formulação matemática pode ser encontrada em Portugal (1999) e Portugal *et. alii.* (2000). O problema de controle em regime transitório, que é o

principal tema deste trabalho, é abordado a partir da seção 4.

## 2.1 Exemplo

Com a finalidade de ilustrar a resolução de um problema de *job shop* cíclico, considere um exemplo com 3 máquinas e 3 peças. Cada peça é processada em ambas as máquinas, totalizando 9 operações. Os roteiros de fabricação (correspondem ao número da máquina), as durações das operações encontram-se na tabela 1.

Tabela 1: Ordem de Operação e Duração das operações do exemplo

Número do job	Roteiro de Fabricação	Duração da operação
1	1 2 3	4 2 2
2	2 1 3	2 3 3
3	3 2 1	3 2 3

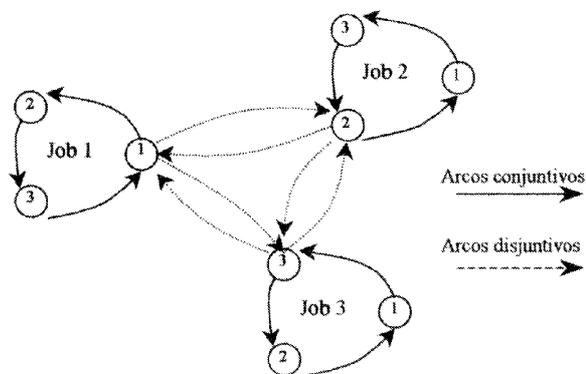


Figura 2: Grafo do exemplo com restrições conjuntivas e disjuntivas

A figura 2 mostra as restrições conjuntivas<sup>1</sup> e as restrições disjuntivas<sup>2</sup>; os números dentro dos círculos correspondem à máquina a ser utilizada pelo *job*. O conjunto de restrições conjuntivas formam um grafo conjuntivo. O grafo disjuntivo consiste da união do grafo conjuntivo com as restrições disjuntivas, representadas pelos arcos disjuntivos. É necessário definir uma seleção das restrições disjuntivas quando busca-se resolver o problema de *job shop* cíclico.

Este exemplo pode ser resolvido utilizando-se de programação linear mista. A solução encontrada neste caso foi  $\tau=0,1$  (número médio de iterações por unidade de

<sup>1</sup> Restrição de anterioridade

<sup>2</sup> Necessidade de se compartilhar recursos

tempo), ou  $\alpha=10$ , (tempo de ciclo). Para este exemplo os respectivos tempo de início de cada *job* em cada máquina encontram-se na tabela 2 e o ciclo obtido está representado na figura 3.

Tabela 2: Tempo de Início de cada tarefa do exemplo 1

Número do job	Roteiro de Fabricação	Tempo de Início da operação
1	1 2 3	0 7 10
2	2 1 3	2 4 7
3	3 2 1	2 5 7

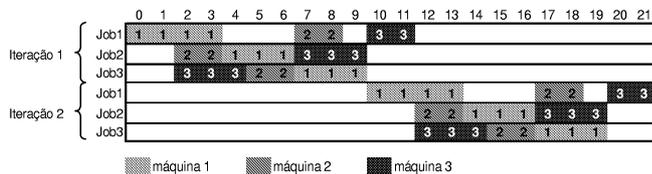


Figura 3: Diagrama da solução do exemplo 1

No contexto geral deste trabalho, o tempo de ciclo ( $\alpha$ ) de um JSC será representado por  $\lambda^\#$  e  $t^\#$  representará o instante de início da primeira operação de cada *job* quando uma solução é determinada. Assim, para o exemplo 1 tem-se  $\lambda^\# = 10$  e  $t^\# = [0 \ 2 \ 2]^T$

## 2.2 Soluções de Regime

Em uma solução de regime conhece-se o tempo de início e fim de cada tarefa genérica e o tempo de ciclo. Neste trabalho, parte-se do princípio que uma solução de regime para o problema dado é conhecida, utilizando-se de um método qualquer para determinação desta solução de regime para o *job shop* cíclico com tempos determinísticos. Chama-se de solução de referência a solução de regime encontrada, podendo esta ser ótima ou não de acordo com a complexidade do problema. Considera-se que em uma solução de regime um *job* pode aparecer em um ciclo uma única vez. Ou seja, um *job* de um ciclo  $k + 1$  só poderá se iniciar se todos os outros *jobs* do ciclo anterior  $k$  já foram iniciados.

Encontrada uma solução de regime, esta será repetida continuamente. Entretanto, conforme observado anteriormente, o problema de *job shop* cíclico pode estar sujeito a imprevistos, tais como falhas de máquinas ou falta de matéria prima, originando então uma perturbação na solução de referência.

Uma simples perturbação pode induzir várias pertur-

bações ou atrasos secundários. Por exemplo, na figura 3, suponha-se que uma perturbação na primeira operação do *job 2* (máquina 2) tenha ocorrido. Se o atraso é suficientemente grande, poderá, por sua vez, atrasar o início da próxima operação, que corresponde à operação 2 do *job 3*. Outrossim, desde que a primeira operação do *job 2* na máquina 2 deve ser completada antes que a segunda operação deste *job* possa ser iniciada na máquina 1, um atraso no término da primeira operação do *job 2* pode causar um atraso no início da segunda operação deste *job*. Se o início da segunda operação do *job 2* é atrasado na máquina 1, imediatamente acarretará um atraso na terceira operação do *job 3* na máquina 1 e, assim por diante. Neste momento é necessário ter algum tipo de controle que permita a continuidade da produção e garanta o retorno à solução de referência. É necessário então, que se faça um planejamento reativo do problema de *job shop* cíclico.

Conforme estabelecido na introdução deste trabalho, o planejamento reativo proposto é realizado pela utilização de uma lei de controle cuja descrição é apresentada na seção 4.

Apresenta-se na próxima seção os conceitos teóricos necessários para o desenvolvimento deste trabalho.

## 3 CONCEITOS EM ÁLGEBRA MAX-PLUS E NOTAÇÃO UTILIZADA

Na Figura 2 apresentou-se o grafo do exemplo com as restrições conjuntivas e disjuntivas. Grafos e suas relações podem ser descritos utilizando-se de uma estrutura algébrica apropriada. A definição desta estrutura, denominada “álgebra de caminhos” ou “álgebra max-plus”, requer menos propriedades do que a álgebra usual (por exemplo, não se postula a existência do elemento inverso em relação à adição).

A álgebra max-plus é essencial para relacionar grafos e matrizes, isto é, pode-se utilizar de matrizes para representar nós e arcos de um grafo e vice-versa.

Nesta seção algumas definições importantes da literatura a respeito da álgebra max-plus, as quais serão utilizadas no decorrer do desenvolvimento teórico do trabalho, são fornecidas (Cohen et al, 1989; Wagneur, 1991; Bacelli et al, 1992).

### 3.1 Álgebra Max Plus

**Definição 3.1** – Um conjunto dotado com duas operações internas  $\oplus$  (soma ou adição) e  $\otimes$  (produto ou multiplicação) é chamado um dióide (denotado D) se os

seguintes axiomas são válidos:

**Axioma 1** – Associatividade da Adição:

$$\forall a, b, c \in D, (a \oplus b) \oplus c = a \oplus (b \oplus c)$$

**Axioma 2** – Comutatividade da Adição:

$$\forall a, b \in D, a \oplus b = b \oplus a$$

**Axioma 3** – Associatividade da Multiplicação:

$$\forall a, b, c \in D, (a \otimes b) \otimes c = a \otimes (b \otimes c)$$

**Axioma 4** – Distributiva da Multiplicação sobre a Adição:

$$\text{à direita: } \forall a, b, c \in D, (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c);$$

$$\text{à esquerda: } \forall a, b, c \in D, c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b);$$

**Axioma 5** – Existência do Elemento Neutro (Nulo) com relação à  $\oplus$ , denominado  $\varepsilon$ :  $\forall a \in D, a \oplus \varepsilon = a$ ;

**Axioma 6** – Absorção do Elemento Neutro da Adição por  $\otimes$ :  $\forall a \in D, a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$ ;

**Axioma 7** – Existência do Elemento Neutro (Identidade) com relação à  $\otimes$ , denominado  $e$ :

$$\forall a \in D, a \otimes e = e \otimes a = a.$$

**Axioma 8** – (Idempotência) Todo dióide D é idempotente:  $\forall a \in D, a \oplus a = a$ .

Neste artigo, associam-se às operações  $\oplus$  e  $\otimes$  de um dióide particular às operações de maximização e adição da álgebra convencional, enquanto os elementos  $e$  e  $\varepsilon$  correspondem a zero e  $-\infty$ , respectivamente (álgebra “max-plus” –  $R_{max}$ ).

## 3.2 Teoria de Grafos

Alguns conceitos da Teoria de Grafos são necessários para o desenvolvimento teórico deste trabalho. Apresenta-se a seguir os conceitos principais utilizados no desenvolvimento da lei de controle proposta. Sugere-se Berge (1973) para um estudo mais detalhado.

**Definição 3.2** (Grafo fortemente conectado) – Um grafo é dito ser fortemente conectado se para qualquer dois diferentes nós  $i$  e  $j$  existe um caminho de  $i$  para  $j$ .

**Definição 3.3** (Grafo de Precedência) – O grafo de precedência de uma matriz quadrada  $\mathbf{A}$ ,  $n \times n$ , com elementos em  $R_{max}$ , é um dígrafo valorado com  $n$  nós e arcos  $(j, i)$  se  $A_{ij} \neq \varepsilon$ . Neste caso o peso deste arco corresponde ao valor numérico de  $A_{ij}$ . O grafo de precedência é denotado por  $G(\mathbf{A})$ .

## 3.3 Matrizes na Álgebra Max-Plus

Considere-se matrizes com elementos pertencentes a  $R_{max}$  e sujeitos às operações algébricas estabelecidas pela definição 3.1.

De acordo com Baccelli *et alii* (1992), existem dois tipos de composição entre matrizes: a *composição paralela*, denotada  $\oplus$ , e a *composição em série*, denotada por  $\otimes$ . Estas composições são definidas utilizando-se das operações de composição de elementos em  $R_{max}$ , para o qual considera-se os mesmos símbolos  $\oplus$  e  $\otimes$ , usualmente referidos como “adição” ou “soma” e como “multiplicação” ou “produto”, respectivamente.

A seguir apresenta-se as definições da composição paralela ( $\oplus$ ) da composição em série ( $\otimes$ ) de matrizes.

**Definição 3.4** – A composição paralela ( $\oplus$ ) de matrizes é definida pela seguinte regra: se  $\mathbf{A} = [A_{ij}]$  e  $\mathbf{B} = [B_{ij}]$  possuem o mesmo tamanho então:  $(\mathbf{A} \oplus \mathbf{B})_{ij} = A_{ij} \oplus B_{ij}$

**Definição 3.5** – A composição em série ( $\otimes$ ) de matrizes  $\mathbf{A}$  e  $\mathbf{B}$  é definida somente quando o número de colunas de  $\mathbf{A}$  é igual ao de linhas de  $\mathbf{B}$ , ou seja:  $\mathbf{A}$  é  $m \times n$  e  $\mathbf{B}$  é  $n \times p$  pela seguinte regra:  $(\mathbf{A} \otimes \mathbf{B})_{ij} = \bigoplus_{k=1}^n (A_{ik} \otimes B_{kj})$

Nesta seção, define-se o máximo ciclo médio conforme Karp (1978). Algumas de suas propriedades são apresentadas a seguir (Bacelli *et alli.*, 1992).

O máximo ciclo médio tem uma clara relação com autovalores de matrizes dentro do contexto da álgebra max-plus e com regimes periódicos de sistemas descritos por equações lineares, também dentro do mesmo contexto.

Seja  $G$  um dígrafo valorado com  $n$  nós. Os pesos dos arcos são números reais, os quais são dados por uma matriz  $\mathbf{A}$   $n \times n$ . O valor numérico de  $A_{ij}$  é igual ao peso do arco do nó  $j$  para o nó  $i$ . Se tal arco não existe, então considera-se  $A_{ij} = \varepsilon$ . A entrada  $(i, j)$  de  $\mathbf{A}^k = \mathbf{A} \oplus \mathbf{A} \oplus \dots \oplus \mathbf{A}$ , considerada dentro da estrutura algébrica  $R_{max}$ , denota o peso máximo com respeito a todos os caminhos de comprimento  $k$  indo do nó  $j$  para o nó  $i$ . Se nenhum caminho existe, então  $(\mathbf{A}^k)_{ij} = \varepsilon$ .

**Definição 3.6 (Ciclo Médio)** O peso médio de um caminho é definido como a soma dos pesos dos arcos individuais deste caminho, dividido pelo comprimento deste caminho. Se o caminho é denotado por  $\rho$ , então o peso médio é igual a  $|\rho|_w / |\rho|_c$ . Se tal caminho é um circuito, diz-se o peso médio do circuito, ou simplesmente o ciclo médio.

Considere uma matriz  $\mathbf{A}$   $n \times n$  com o grafo de precedência  $G$  correspondente. O peso máximo de todos os circuitos de comprimento  $j$  que passam pelo nó  $i$  de  $G$  pode ser escrito como  $(\mathbf{A}^j)_{ii}$ . O máximo destes pesos máximos sobre todos os nós é  $\bigoplus_{i=1}^n (A^j)_{ii}$  pode ser escrito como traço  $(A^j)$ . O peso médio é obtido dividindo este número por  $j$  no sentido convencional, que pode ser escrito como  $(\text{traço}(A^j)_{ii})^{1/j}$  na notação da álgebra max-plus. Tem-se então o ciclo de peso máximo com respeito ao comprimento  $j$ . Portanto, a fórmula para o máximo ciclo médio é dado por (Baccelli *et al.*, 1992):

$$\lambda = \bigoplus_{j=1}^n (\text{traço}(A^j))^{1/j} \quad (1)$$

Não é necessário considerar comprimentos maiores do que o número  $n$  de nós, pois circuitos com mais de  $n$  nós não são elementares.

Todos os circuitos com ciclo médio igual ao máximo ciclo médio são chamados circuitos críticos.

**Definição 3.7.** (Circuito Crítico) – um circuito  $\zeta$  do grafo de precedência  $G(\mathbf{A})$  é chamado crítico se este tem peso máximo, denotado por  $|\zeta|_w$ .

O teorema de Karp (Baccelli *et alii*, 1992) fornece uma expressão para o cálculo do máximo ciclo médio.

Dada uma matriz  $\mathbf{A}$  com elementos em  $R_{max}$ , considere o problema da existência de autovalores e autovetores, isto é, a existência de  $\lambda$  e  $x$  (diferentes de zero) tais que:

$$\mathbf{A}x = \lambda x \quad (2)$$

**Teorema 3.8.** Se  $\mathbf{A}$  é irredutível, ou equivalentemente se  $G(\mathbf{A})$  é fortemente conectado, existe um e somente um autovalor (mas possivelmente vários autovetores). Este autovalor é igual ao máximo ciclo médio do grafo:

$$\lambda = \max_{\zeta} \frac{|\zeta|_w}{|\zeta|_l} \quad (3)$$

onde  $\zeta$  percorre sobre o conjunto de circuitos de  $G(\mathbf{A})$ .

A demonstração do teorema 3.8 encontra-se em Baccelli *et alii.* (1992: 113).

**Exemplo 3.9 (Não unicidade de autovetor)** Observe que considerando uma matriz irredutível, conforme o teorema 3.8 tem-se unicidade de autovalor, porém tal unicidade não é válida para o autovetor, como mostra o exemplo a seguir:

$$\begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix} \otimes \begin{pmatrix} e \\ -3 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} = 2 \otimes \begin{pmatrix} e \\ -3 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix} \otimes \begin{pmatrix} e \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} = 2 \otimes \begin{pmatrix} e \\ 1 \end{pmatrix}.$$

Os dois autovetores são obviamente não “proporcionais”.

## 4 MODELO MATEMÁTICO PARA O SISTEMA REATIVO

O objetivo principal desta seção é estabelecer um modelo algébrico para o controle do comportamento transitório *do job shop* cíclico para o qual se conhece uma solução de referência.

Como visto na seção 2, o problema do *job shop* cíclico pode estar sujeito a imprevistos, tais como falha de máquinas ou falta de matéria-prima, originando então uma perturbação na solução de referência. É necessário desta forma, que se determinem leis de controle para permitir a continuidade da produção e ao mesmo tempo garantir a convergência para a solução de referência preestabelecida.

Tendo em vista tais necessidades, buscou-se nesse trabalho encontrar uma lei de controle, ou seja, uma lei de determinação dos instantes de início de cada *job* nos ciclos subsequentes, capaz de restabelecer e manter a solução de referência.

Basicamente, conforme apresentado na figura 1, conhecendo a solução de regime é possível a determinação de uma lei de controle. Considerando esta lei de controle e os instantes de início do ciclo corrente, calcula-se os instantes de início do próximo ciclo, os quais estarão sujeitos a restrições de factibilidade da planta.

Seja o problema de *job shop* cíclico descrito na seção 2, com  $n$  *jobs* e  $m$  máquinas. De uma maneira geral o vetor

$$\mathbf{t}_k \in R^n \quad (4)$$

representará os instantes de início do  $k$ -ésimo ciclo do *job shop*. O vetor  $\mathbf{t}^\#$  representará a solução de referência obtida a partir de técnicas descritas por Portugal (1999).

A lei de controle proposta neste trabalho supõe a existência de um conjunto de matrizes  $\mathbf{A}_l$ ,  $l \in \{1, \dots, m\}$  na álgebra max-plus, denominado conjunto de matrizes de decisão, cujas características são descritas na seção 4.1 (definição 4.5). Essas matrizes serão utilizadas para determinação dos instantes de início de cada *job* nos ciclos subsequentes, com o objetivo de restabelecer o ciclo anteriormente determinado, após a ocorrência de uma perturbação.

A lei de controle proposta deve ser tal que, se o sistema estiver operando sem perturbação, então a solução em

regime discutida na seção anterior deve ser implementada.

Caso não haja nenhuma perturbação, os instantes de disparo do  $n$ -ésimo ciclo são dados pelo vetor:

$$\mathbf{t}_n = (\lambda^\#)^n \otimes \mathbf{t}^\#$$

Considere uma matriz  $\mathbf{A}$   $n \times n$  tal que:

$$\mathbf{A} \otimes \mathbf{t}^\# = \lambda^\# \otimes \mathbf{t}^\# \quad (5)$$

Conforme as definições de autovalor e autovetor apresentadas na seção 3, equação (2), tem-se que  $\mathbf{t}^\#$  e  $\lambda^\#$  são respectivamente, autovetor e autovalor da matriz  $\mathbf{A}$ .

Assim, qualquer matriz cujo autovetor seja  $\mathbf{t}^\#$  e cujo autovalor correspondente seja  $\lambda^\#$ , pode ser usada no contexto da lei de controle quando o sistema estiver em regime permanente. Portanto o conjunto de matrizes de decisão deverá conter matrizes satisfazendo a esta condição.

Havendo uma perturbação na solução de referência devido a problemas de falha de máquinas ou outras causas, considera-se a primeira solução determinada imediatamente após tal perturbação, a qual é denominada de  $\mathbf{t}_0$ . No contexto deste trabalho  $\mathbf{t}_0$  pode ser considerada como o análogo a uma “condição inicial” em um problema dinâmico. Assume-se que após a perturbação ocorrida a planta física volta ao seu normal, contudo a “condição inicial”  $\mathbf{t}_0$  é diferente da solução de referência  $\mathbf{t}^\#$ .

O controle do sistema é feito pelo uso do conjunto de matrizes de decisão da seguinte forma:

### Algoritmo de Controle

- I. Escolhe-se dentro do conjunto de matrizes de decisão uma matriz tal que o produto

$$\mathbf{t}' = \mathbf{A}_l \otimes \mathbf{t}_0$$

seja uma solução factível. Na seção 5.4 serão discutidos os critérios para escolha de uma matriz dentro do conjunto  $\mathbf{A}_l$ ,  $l \in \{1, \dots, m\}$ ; na seção 5.2 mostra-se como a verificação de factibilidade é feita; finalmente na seção 4.2 mostra-se que é possível obter um conjunto de matrizes tal que sempre exista uma solução factível.

- II. A solução obtida  $\mathbf{t}'$  determina os instantes de disparo dos jobs no próximo ciclo. Para a obtenção dos instantes de disparo no ciclo seguinte repete-se recorrentemente o procedimento anterior, considerando-se  $\mathbf{t}'$  como “condição inicial”.

- III. Caso se atinja, depois de alguns ciclos, a solução de referência<sup>3</sup>, mostra-se na seção 4.2 que existe pelo menos uma matriz no conjunto de matrizes de decisão, tal que a solução de referência seja mantida em regime permanente.

### Fim Algoritmo de Controle

O funcionamento apropriado do algoritmo anterior está condicionado pelas seguintes questões:

- A solução  $\mathbf{t}'$  deve ser factível, isto é, não pode produzir conflitos entre as operações no uso dos recursos (máquinas). A verificação da factibilidade é feita segunda a metodologia proposta na seção 5.2.
- Para qualquer condição inicial deve-se garantir a existência de pelo menos uma matriz de decisão tal que  $\mathbf{t}' = \mathbf{A} \otimes \mathbf{t}_0$  seja factível. Esta questão é discutida na seção 4.2.
- Deve-se garantir que a utilização do algoritmo faz com que após um número suficientemente grande de iterações a solução convirja para a solução de regime (deslocada no tempo). A seção 4.1 apresenta condições suficientes a que devem satisfazer as matrizes de decisão para que isto ocorra.

A convergência para a solução de referência é expressa matematicamente por:

$$\lim_{k \rightarrow +\infty} \bigotimes_{i=1}^k (\mathbf{B}_{l(i)}) \otimes \mathbf{t}_0 = \delta \otimes \mathbf{t}^\#$$

onde as matrizes  $\mathbf{B}_l$  são definidas por:

$$\mathbf{B}_l = \lambda_l^{-1} \otimes \mathbf{A}_l, \forall l \in \{1, \dots, m\},$$

$\lambda_l$  é autovalor de  $\mathbf{A}_l$  e,  $\delta \otimes \mathbf{t}^\#$  corresponde a um deslocamento de  $\delta$  na solução de referência. Deve-se observar que, no produtório acima, a seqüência de matrizes  $\mathbf{B}_l$  utilizada é arbitrária, ou seja, a função  $l(i)$  é qualquer. É importante ainda salientar que as matrizes a serem utilizadas nas lei de controle são as matrizes  $\mathbf{A}_l$ ,  $l \in \{1, \dots, m\}$ . As matrizes  $\mathbf{B}_l = \lambda_l^{-1} \otimes \mathbf{A}_l$ ,  $l \in \{1, \dots, m\}$ , tendo autovalor unitário (igual a  $e$ ), constituem “fatorizações”<sup>4</sup> das matrizes  $\mathbf{A}_l$ .

<sup>3</sup>Obviamente deslocada no tempo

<sup>4</sup>A “multiplicação” ( $\otimes$ ) por  $\lambda_l^{-1}$  corresponde à subtração na álgebra tradicional

## 4.1 Caracterização do Conjunto de Matrizes de Decisão & Convergência da Lei de Controle

Conforme descrito anteriormente, o conjunto de matrizes de decisão deve permitir o controle do sistema em regime transitório e além disso conter uma matriz que caracterize o comportamento em regime permanente. No caso do regime transitório, é necessário que essas matrizes satisfaçam determinadas condições de modo a propiciar o restabelecimento do ciclo de referência quando há uma perturbação.

Duas condições são fundamentais. A primeira é que tais matrizes tenham um único autovetor  $\mathbf{t}^\#$  e a segunda é que a matriz resultante do produto de qualquer número de matrizes do conjunto também possua um único autovetor  $\mathbf{t}^\#$ .

O teorema a seguir relaciona as condições acima com a convergência da solução.

**Teorema 4.1.** Seja um conjunto de matrizes  $\mathbf{B}_l$ ,  $l \in \{1, \dots, m\}$  tal que  $\forall l \in \{1, \dots, m\}$ ,  $\mathbf{B}_l$  possua um único autovalor  $e$ , um único autovetor  $\mathbf{t}^\#$  e a matriz resultante da combinação de qualquer produto entre as matrizes  $\mathbf{B}_l$ ,  $l \in \{1, \dots, m\}$  possua um único autovetor  $\mathbf{t}^\#$ . Então

$$\left( \lim_{k \rightarrow +\infty} \bigotimes_{i=1}^k (\mathbf{B}_{l(i)}) \right) \otimes \mathbf{t}_0 = \delta \otimes \mathbf{t}^\#,$$

onde  $l(i)$  é qualquer função  $\{1, \dots, k\} \rightarrow \{1, \dots, m\}$ .

A demonstração deste teorema pode ser encontrada em Noronha (1999), e consiste do seguintes passos:

Considere-se  $\mathbf{t}_k$  o vetor de instantes de início dos *jobs* no  $k$ -ésimo ciclo, isto é:

$$\mathbf{t}_k = \left[ \bigotimes_{i=1}^k \mathbf{B}_{l(i)} \right] \otimes \mathbf{t}_0 \quad (6)$$

Pode-se colocar  $(\mathbf{t}_k)_j$  ( $j$ -ésimo componente do vetor  $\mathbf{t}_k$ ) na seguinte forma:

$$(\mathbf{t}_k)_j = \gamma_j^k \otimes (\mathbf{t}^\#)_j, \quad \forall j \in \{1, \dots, n\}$$

Analisa-se então a evolução de

$$\gamma_j^k \quad \forall j = \{1, \dots, n\},$$

quando  $k \rightarrow +\infty$ , observando-se que

$$\lim_{k \rightarrow +\infty} |\gamma_j^k - \gamma_i^k| = 0, \quad \forall i, j \in \{1, \dots, n\}$$

Em conclusão:  $\lim_{k \rightarrow +\infty} \mathbf{t}_k = \delta \otimes \mathbf{t}^\#$ .

## 4.2 Construção do Conjunto de Matrizes de Decisão

Nesta seção será proposta uma metodologia de construção do conjunto de matrizes de decisão de modo a satisfazer às hipóteses do teorema 4.1. Para isto alguns resultados preliminares são necessários.

**Proposição 4.2.** Sejam  $\mathbf{A}_1$  e  $\mathbf{A}_2$  matrizes  $n \times n$  com autovalores  $\lambda_1$  e  $\lambda_2$ , respectivamente. Seja  $\mathbf{t}^\#$  autovetor de  $\mathbf{A}_1$  e  $\mathbf{A}_2$ . Então  $\mathbf{t}^\#$  é autovetor do produto das duas matrizes e existe pelo menos um autovalor do produto de  $\mathbf{A}_1$  e  $\mathbf{A}_2$  igual a  $\lambda_1 \otimes \lambda_2$ .

A demonstração desta proposição pode ser encontrada em Noronha (1999).

Observa-se contudo que se:

- I  $G(\mathbf{A}_1)$  e  $G(\mathbf{A}_2)$  são fortemente conectados<sup>5</sup> ( $\lambda_1$  e  $\lambda_2$  são autovalores únicos respectivamente de  $\mathbf{A}_1$  e  $\mathbf{A}_2$ ),
- II. ambas as matrizes,  $\mathbf{A}_1$  e  $\mathbf{A}_2$ , possuem um único autovetor  $\mathbf{t}^\#$  e,
- III.  $G(\mathbf{A}_1 \otimes \mathbf{A}_2)$  e  $G(\mathbf{A}_2 \otimes \mathbf{A}_1)$  são fortemente conectados,

então  $\mathbf{A}_1 \otimes \mathbf{A}_2$  e  $\mathbf{A}_2 \otimes \mathbf{A}_1$  possuem um único autovalor  $\lambda_1 \otimes \lambda_2$ ; porém, não necessariamente  $\mathbf{t}^\#$  é o único autovetor de  $\mathbf{A}_1 \otimes \mathbf{A}_2$  ou  $\mathbf{A}_2 \otimes \mathbf{A}_1$ , conforme mostra o exemplo a seguir.

**Exemplo 4.3.** Considerem-se as seguintes matrizes:

$$\mathbf{A}_1 = \begin{bmatrix} e & -1 & -3 \\ 1 & e & -3 \\ 2 & -1 & -1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} -1 & -2 & -2 \\ -3 & e & -1 \\ -3 & 1 & e \end{bmatrix}$$

Verifica-se que tanto  $\mathbf{A}_1$  quanto  $\mathbf{A}_2$  possuem um único autovetor, no contexto da álgebra max-plus, igual a  $\mathbf{t} = [e \ 1 \ 2]^T$ . Porém a matriz resultante do produto de  $\mathbf{A}_1$  e  $\mathbf{A}_2$ ,  $\mathbf{A}_{12} = \mathbf{A}_1 \otimes \mathbf{A}_2$  não possui um único autovetor:

$$\mathbf{A}_{12} = \mathbf{A}_1 \otimes \mathbf{A}_2 = \begin{bmatrix} -1 & -1 & -2 \\ e & e & -1 \\ 1 & e & e \end{bmatrix}$$

Os autovetores são  $\mathbf{t} = [-1 \ e \ 1]^T$  e  $\mathbf{t} = [-1 \ e \ e]^T$  que são linearmente independentes.

<sup>5</sup> $G(A)$  representa o grafo de precedência da matriz  $A$ . A definição de grafo fortemente conectado encontra-se na seção 3.

Para caracterizar as matrizes do conjunto  $\mathbf{A}_l$ ,  $l \in \{1, \dots, m\}$ , é fundamental a unicidade dos autovetores dos produtos entre elementos desse conjunto.

Considere a matriz  $\mathbf{B}^\#$  definida a seguir:

$$\mathbf{B}^\# \stackrel{\text{def}}{=} \begin{bmatrix} e & t_1^\# \phi t_2^\# & t_1^\# \phi t_3^\# & \dots & t_1^\# \phi t_n^\# \\ t_2^\# \phi t_1^\# & e & t_2^\# \phi t_3^\# & \dots & t_2^\# \phi t_n^\# \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_n^\# \phi t_1^\# & t_n^\# \phi t_2^\# & t_n^\# \phi t_3^\# & \dots & e \end{bmatrix} \quad (7)$$

onde a notação  $t_i^\# \phi t_j^\#$  corresponde a  $t_i^\# - t_j^\#$  na álgebra usual.

Esta matriz possui algumas propriedades importantes para o estudo da convergência da lei de controle. Particularmente é imediato concluir que seu autovalor é  $e$  e seu autovetor é  $\mathbf{t}^\# = [t_1^\# \ t_2^\# \ \dots \ t_n^\#]^T$ .

**Proposição 4.4.** Se uma matriz  $\mathbf{A}$  tem como autovetor  $\mathbf{t}^\#$  e como único autovalor  $\lambda_A$ , então<sup>6</sup>  $\mathbf{A}_e = \lambda_A^{-1} \otimes \mathbf{A} \leq \mathbf{B}^\#$ .

A demonstração desta proposição pode ser encontrada em Noronha (1999).

De acordo com a proposição 4.4,  $\mathbf{B}^\#$  é um majorante para toda matriz que tenha autovetor igual a  $\mathbf{t}^\#$  e autovalor igual a  $e$ . Porém, não basta que duas matrizes quaisquer sejam menores que  $\mathbf{B}^\#$  e tenham um único autovetor  $\mathbf{t}^\#$ , para que o produto destas matrizes tenham  $\mathbf{t}^\#$  como único autovetor. Pode-se observar esta situação no Exemplo 4.3, onde  $\mathbf{A}_1$  e  $\mathbf{A}_2$  são menores do que  $\mathbf{B}^\#$ , que é definida pelo autovetor  $\mathbf{t}^\# = [e \ 1 \ 2]^T$ , ou seja:

$$\mathbf{B}^\# = \begin{bmatrix} e & -1 & -2 \\ 1 & e & -1 \\ 2 & 1 & e \end{bmatrix}.$$

Entretanto a matriz resultante do produto de  $\mathbf{A}_1$  e  $\mathbf{A}_2$ , denominada  $\mathbf{A}_{12} = \mathbf{A}_1 \otimes \mathbf{A}_2$  não possui um único autovetor.

A definição 4.5 caracteriza um conjunto de matrizes com autovetor único  $\mathbf{t}^\#$  tais que o produto de qualquer número de matrizes desse conjunto resulta numa matriz com o mesmo autovetor (único).

**Definição 4.5** Considere um conjunto de matrizes  $\mathbf{B}_l$ ,  $l \in \{1, \dots, m\}$  definidas da seguinte forma:

4.5.1. autovalor de cada  $\mathbf{B}_l$  igual a  $e$ ;

4.5.2. autovetor de  $\mathbf{B}_l$  igual a  $\mathbf{t}^\#$  e único;

<sup>6</sup>A matriz  $\mathbf{A}_e$  tem autovalor  $e$ .

4.5.3. se não existe nenhum ciclo crítico no grafo de precedência  $G(\mathbf{B}_l)$  passando por  $j$  então

$$(\mathbf{B}_l)_{ij} < \mathbf{B}_{ij}^\#, \forall i = 1, \dots, n;$$

4.5.4. se existe um ciclo crítico passando por  $j$  no grafo de precedência  $G(\mathbf{B}_l)$  então

$$(\mathbf{B}_l)_{ij} = \mathbf{B}_{ij}^\#, \forall i = 1, \dots, n.$$

É imediato concluir que o conjunto acima definido coincide com o conjunto das matrizes possuindo pelo menos uma coluna igual a uma coluna da matriz  $\mathbf{B}^\#$  e os demais valores estritamente menores que os respectivos de  $\mathbf{B}^\#$ . Além disso, tais matrizes quando multiplicadas entre si possuem um único autovetor  $\mathbf{t}^\#$ , conforme demonstrado em Noronha (1999).

Em conclusão, o conjunto de matrizes de decisão pode ser constituído a partir de qualquer subconjunto do conjunto definido em 4.5 multiplicando-se cada elemento  $\mathbf{B}_l$  por um autovalor arbitrário  $\lambda_l$ .

Do ponto de vista prático é conveniente que duas matrizes, (satisfazendo às condições acima) estejam presentes. Em primeiro lugar deve-se incluir a matriz dada por:

$$\mathbf{A}^\# = \lambda^\# \otimes \mathbf{B}^\#.$$

A matriz  $\mathbf{A}^\#$  possui então autovalor  $\lambda^\#$  e autovetor  $\mathbf{t}^\#$ , podendo ser utilizada para manter o sistema em regime sempre que a solução de referência é restabelecida.

Em segundo lugar deve-se incluir a matriz definida pela equação(7) :

$$\mathbf{A}_M \stackrel{\text{def}}{=} \partial \otimes \mathbf{B}^\# \quad (8)$$

onde o valor mínimo para  $\partial$  corresponde à máxima duração total dos *jobs*. Logicamente, o autovalor da matriz  $\mathbf{A}_M$  é igual a  $\partial$  e seu autovetor é  $\mathbf{t}^\#$ .

A matriz  $\mathbf{A}_M$  é utilizada quando nenhuma outra matriz do conjunto de matrizes de decisão  $\mathbf{A}_l$ ,  $l \in \{1, \dots, m\}$  for capaz de encontrar um vetor de tempo factível quando multiplicada pelo vetor de tempo em questão. Obviamente, qualquer vetor multiplicado pela matriz  $\mathbf{A}_M$  produz sempre uma solução factível, a qual corresponde à solução de referência, ou seja,

$$\mathbf{A}_M \otimes \mathbf{t}_0 = (\partial \otimes \mathbf{B}^\#) \otimes \mathbf{t}_0 = \partial \otimes \mathbf{t}^\#.$$

Sendo  $\partial$  um valor alto, tem-se sempre que o deslocamento sobre a solução de referência é factível em relação à solução anterior.

## 5 ALGORITMOS

A utilização do conjunto de matrizes com as características descritas na seção 4.2 garantirá a convergência para a solução de regime, independentemente da condição inicial, conforme mostrado no teorema 4.1. A seção 5.1 apresenta um algoritmo que permite construir um conjunto de matrizes com as características propostas na seção 4.2.

O **algoritmo de controle** proposto na seção 4 exige o detalhamento de alguns procedimentos, a saber:

1. verificação da factibilidade da solução para um ciclo subsequente;
2. escolha, dentre várias soluções factíveis, da melhor alternativa.

A seção 5.2 apresenta uma técnica de verificação de factibilidade utilizando o modelo proposto por Groth e Mendes (1997). Considera-se que os tempos de espera entre operações em um mesmo *job* são sempre iguais aos tempos de espera da solução de regime, mesmo que o sistema esteja em regime transitório.

Na seção 5.3 a restrição de que os tempos de espera entre operações em um mesmo *job* são sempre iguais aos tempos de espera da solução de regime quando o sistema está em regime transitório é relaxada. Isto é, considera-se o caso da verificação de factibilidade quando os tempos de espera entre operações de um *job* são flexíveis.

A seção 6 fornece um exemplo de comparação entre as situações descritas nas seções 5.2 e 5.3.

Dentre várias soluções factíveis, é necessário escolher qual deverá ser utilizada. As formas propostas de escolha da solução são descritas na seção 5.4.

### 5.1 Algoritmo para Cálculo do Conjunto de Matrizes de Decisão

O objetivo principal desta seção é mostrar que é possível a construção do conjunto de matrizes de decisão. Para tanto, desenvolveu-se o *algoritmo de síntese I* para o cálculo do conjunto de matrizes. Inicialmente, o algoritmo calcula matrizes aleatoriamente e dentre estas escolhe algumas de acordo com critérios estabelecidos. A idéia principal deste algoritmo consiste em construir as matrizes, tendo por base uma matriz ou várias matrizes com o formato definido na seção 4.2. O algoritmo proposto é descrito a seguir.

#### Algoritmo de Síntese I

*Variáveis:*

$\mathbf{t}^\# = [ t_1^\# \ t_2^\# \ \dots \ t_n^\# ]^T \rightarrow$  solução de referência;

$\mathbf{B}^\# = \begin{bmatrix} t_1^\# \\ t_2^\# \\ \vdots \\ t_n^\# \end{bmatrix} \otimes [ e\phi t_1^\# \ e\phi t_2^\# \ \dots \ e\phi t_n^\# ] \rightarrow$  matriz

base;

$\lambda^\# \rightarrow$  tempo de ciclo da solução de referência;

$\mathbf{A}^\# = \lambda^\# \otimes \mathbf{B}^\# \rightarrow$  matriz utilizada para manter a solução de referência;

$\mathbf{A}_l \rightarrow$  matriz pertencente ao conjunto;

$\lambda_l \rightarrow$  autovalor da matriz  $\mathbf{A}_l$ ;

$\mathbb{T} \rightarrow$  conjunto de vetores de instantes de início factíveis dos jobs para o problema;

$\mathbf{t} = [ t_1 \ t_2 \ \dots \ t_n ]^T, \mathbf{t} \in \mathbb{T} \rightarrow$  vetor de instantes de início de processamento dos jobs em um determinado ciclo;

$njobs \rightarrow$  cardinalidade do conjunto de jobs;

$I \rightarrow$  conjunto de números naturais, maiores e iguais a 1 e menores ou iguais a  $njobs$ , gerados aleatoriamente.

*Passos:*

1º) Gerar  $\mathbb{T}$  aleatoriamente utilizando-se de uma distribuição uniforme em torno da solução de regime.

2º) Escolher um valor para  $\lambda_l$ , tal que

$$\frac{\lambda^\#}{2} \leq \lambda_l \leq \lambda^\# \otimes \left( \bigoplus_{i=1}^{njobs} \beta_i \right),$$

onde  $\beta_i$  (comprimento do *job*  $i$ ) é definido pela diferença entre o tempo de término do processamento do *job* e o tempo de início do processamento.

3º) Definir  $\mathbf{A}_l = \lambda_l \otimes \mathbf{B}^\#$ .

4º) Escolher aleatoriamente um número  $k, 1 \leq k < njobs$ . Em seguida, escolher aleatoriamente  $k$  números naturais maiores ou iguais a 1 e menores ou iguais a  $njobs$ , os quais irão pertencer ao conjunto  $I$ . Desta forma, a  $\text{card}(I) = k$ .

5º) Para  $j = 1$  até  $njobs$ , modificar aleatoriamente os valores de  $(\mathbf{A}_l)_{\bullet,j}$ , se  $j \notin I$ , onde deve-se ter  $(\mathbf{A}_l)_{\bullet,j} < (\lambda_l \mathbf{B}^\#)_{\bullet,j}$ .

6º) Para todo  $\mathbf{t} \in T$  fazer: seja  $\mathbf{t}' = \mathbf{A}_l \otimes \mathbf{t}$ , onde  $\mathbf{A}_l$  é a nova matriz gerada; verificar se a diferença entre o início da primeira operação do novo ciclo ( $\mathbf{t}'$ ) e a última operação do ciclo anterior ( $\mathbf{t}$ ) é maior que zero. Em caso afirmativo, desconsiderar a matriz obtida. Em outras palavras, espera-se que a matriz não gere uma solução trivial, isto é, o próximo ciclo iniciando após o término do ciclo anterior. Se a matriz obtida obedecer as restrições impostas acima, então a matriz irá fazer parte do conjunto de matrizes de decisão e retira-se  $\mathbf{t}$  do conjunto  $T$ .

7º) Critério de parada: todos os  $\mathbf{t} \in T$  tem que ser satisfeito por algum  $\mathbf{A}$ , ou seja  $T$  ficar vazio.

### Fim do Algoritmo de Síntese I.

As matrizes que irão formar parte do conjunto podem ser calculadas pelo algoritmo proposto acima. Além dessas matrizes, duas outras devem ser acrescentadas: as matrizes  $\mathbf{A}^\#$  e  $\mathbf{A}_M$ , definidas na seção 4.2, que correspondem respectivamente à matriz que mantém a solução de referência e à matriz utilizada quando nenhuma outra matriz do conjunto for capaz de encontrar um vetor de instantes de início de processamento dos *job* factível ao ser multiplicada pelo vetor em questão. Um outro algoritmo para determinação do conjunto de matrizes de decisão pode ser encontrado em Noronha (1999).

## 5.2 Verificação de Factibilidade

A verificação de factibilidade pode ser feita utilizando-se o modelo proposto por Groth e Mendes (1997), o qual é apresentado nesta seção.

Considere-se dois jobs quaisquer  $i, j \in \{1, \dots, n\}$  e seja  $\chi$  o conjunto de todas as seguinte funções  $m_{ij} : \mathfrak{R} \rightarrow \{0, 1\}$  associadas ao par ordenado  $(i, j)$ :

$$m_{ij}(t_i - t_j) = \begin{cases} 1, & \text{se o início do job } j \text{ em } t_j \text{ e o início} \\ & \text{do job } i \text{ em } t_i \text{ não ocasiona} \\ & \text{qualquer conflito;} \\ 0, & \text{caso contrário.} \end{cases}$$

Define-se a adição neste conjunto como a seguinte operação interna:

$$(m_1 \oplus m_2)(\tau) = m_1(\tau) \cdot m_2(\tau)$$

O elemento neutro da adição é definido por:

$$m_\varepsilon(\tau) = 1, \forall \tau \in \mathfrak{R}.$$

Na definição da função  $m_{ij}$ , assume-se que não há conflito se nenhuma operação do *job*  $i$  e do *job*  $j$  utilizam-se da mesma máquina no mesmo intervalo de tempo. Um

típico gráfico da função  $m_{ij}(\tau)$  é mostrado na figura 4. Para cada par  $(i, j)$ , supondo-se que o *job*  $i$  tenha sido

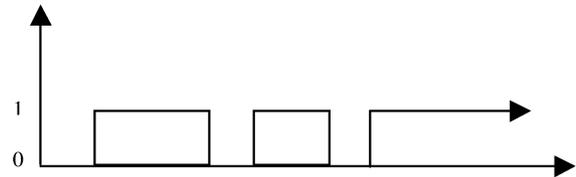


Figura 4: Gráfico típico de  $m_{ij}(\tau)$

disparado em  $\tau=0$ , a função  $m_{ij}(\tau)$  fornece os instantes em que o *job*  $j$  pode ser disparado. Observa-se que para qualquer par  $(i, j)$ , existe um número  $\tau^*$ , suficientemente grande, tal que para todo  $\tau > \tau^*$ ,  $m_{ij}(\tau) = 1$ ; para tanto, basta considerar que o *job*  $j$  se inicia após o término do *job*  $i$ , onde então, não haverá nenhum conflito.

A questão da factibilidade surge quando há necessidade de verificar se há conflito entre os disparos realizados. Suponha que  $p$  jobs  $i_1, i_2, \dots, i_p$  tenham sido disparados respectivamente nos instantes  $t_1, t_2, \dots, t_p$  e deseja-se saber em quais intervalos de tempo o *job*  $j$  pode ser disparado, sem ocorrer qualquer conflito com os jobs precedentes. É fácil ver que:

- Os intervalos de tempo nos quais o *job*  $j$  pode ser disparado sem conflito com o *job*  $i_k$  são aqueles onde a função  $m_{ji_k}(\tau - t_k) = 1$ .
- Os intervalos de tempo nos quais o *job*  $j$  pode ser disparado sem conflito com o *job*  $i_1$  e sem conflito com o *job*  $i_2$  e assim por diante, são aqueles onde a função  $g_j(\tau)$ , definida abaixo assume valor 1:

$$g_j(\tau) = m_{ji_1}(\tau - t_1) \cdot m_{ji_2}(\tau - t_2) \cdot \dots \cdot m_{ji_p}(\tau - t_p) \quad (9)$$

onde o produto utilizado acima é o produto booleano padrão.

Em Groth e Mendes (1997) mostra-se que a equação(9) pode ser reescrita no contexto min-max, ou seja,

$$g_j(\tau) = (t_1 \otimes m_{1j}) \oplus \dots \oplus (t_n \otimes m_{nj}) \quad (10)$$

onde  $g_j(\tau) \in \chi$  fornece os instantes no qual o *job*  $j$  pode ser disparado sem induzir conflito com qualquer um dos  $p$  jobs disparados. Para uma representação mais condensada, define-se:

$$\mathbf{t} = [ t_1 \quad t_2 \quad \dots \quad t_n ]^T \in \mathfrak{R}^n$$

$\mathbf{M} \in \chi^{n \times n}$  onde  $[\mathbf{M}]_{ij} = m_{ij}(\tau)$

$$\mathbf{g}(\tau) = [g_1(\tau) \quad g_2(\tau) \quad \dots \quad g_n(\tau)]^T \in \chi^n$$

Desta forma, a equação (10) pode ser reescrita como

$$\mathbf{g}(\tau) = \mathbf{M} \otimes \mathbf{t}, \quad (11)$$

onde  $\mathbf{g}(\tau)$  e  $\mathbf{t}$  são vetores colunas e  $\mathbf{M}$  é uma matriz  $n \times n$ .

Considera-se novamente o problema de verificação de factibilidade determinado pelo algoritmo de controle.

Seja  $\mathbf{t} = [t_1 \quad t_2 \quad \dots \quad t_n]^T$  os instantes de início dos *jobs* 1, 2, ..., n respectivamente. Seja  $\mathbf{A}_1$  a matriz de decisão para qual se deseja analisar a factibilidade e seja o vetor  $\mathbf{t}'$ , dado por:

$$\mathbf{t}' = \mathbf{A} \otimes \mathbf{t}.$$

Deseja-se saber se, dado que os *jobs* iniciaram de acordo com as unidades de tempo definidas em  $\mathbf{t}$ , é possível iniciá-los em  $\mathbf{t}'$  sem a ocorrência de conflito.

A verificação de factibilidade neste caso é feita utilizando-se o modelo de Groth e Mendes (1997) apresentado anteriormente, com o acréscimo de uma outra matriz (pertencente a  $\chi^{n \times n}$ ) ao modelo, apresentada a seguir.

Defini-se uma matriz denominada  $\mathbf{M}_D$  da seguinte forma:

$$[\mathbf{M}_D]_{ik} = \begin{cases} m_{j_i j_k}(\tau) & \text{se } t'_k < t'_i \\ m_\varepsilon & \text{caso contrário} \end{cases}, \quad (12)$$

Assim, tem-se que  $\mathbf{M}_D \otimes \mathbf{t}'$  determina os instantes de disparo de modo a não haver conflito com os *jobs* disparados no mesmo ciclo, enquanto que  $\mathbf{M} \otimes \mathbf{t}$  determina os instantes de disparo de modo a não haver conflito com os *jobs* disparados no ciclo anterior.

Com intuito de verificar a factibilidade do vetor gerado, considera-se:

$$\mathbf{g}(\tau) = [(\mathbf{M} \otimes \mathbf{t}) \oplus (\mathbf{M}_D \otimes \mathbf{t}')] \quad (13)$$

Assim,  $\mathbf{t}'$  será factível se  $\forall j, g_j(t'_j) = 1$ , onde  $t'_j$  é o  $j$ -ésimo componente de  $\mathbf{t}'$  e  $g_j$  é o  $j$ -ésimo componente de  $\mathbf{g}$ . Portanto, o algoritmo para verificação de factibilidade consiste na implementação da equação (13).

### 5.3 Flexibilização dos tempos de espera entre operações de um job

Uma das possibilidades do modelo de Groth e Mendes (1997) é a flexibilização dos tempos de espera entre as

operações de um *job*. Neste caso, o cálculo de  $m_{ij}(\tau)$  seria feito com base em cada operação ao invés da utilização dos *jobs*, como descrito na seção 5.2.

Os elementos desta nova matriz podem ser divididos em cinco tipos, de acordo com a relação entre as operações:

1. Se  $o_k$  for operação inicial de algum *job* e ambas as operações  $o_k$  e  $o_i$  utilizam a mesma máquina, então:

$$m_{o_k o_i} = \frac{\text{---} \overbrace{\text{---}}^{\text{---}} \text{---}}{p_{o_k}}$$

onde  $p_{o_k}$  é a duração da operação  $o_k$ .

2. Se  $o_k$  for operação inicial de algum *job*, porém  $o_k$  e  $o_i$  não utilizam a mesma máquina, então:

$$m_{o_k o_i} = \frac{\text{---} \overbrace{\text{---}}^{\text{---}} \text{---}}{0}$$

3. Se  $o_k$  e  $o_i$  utilizam a mesma máquina e além disso  $o_k$  não é operação inicial de nenhum *job* então:

$$m_{o_k o_i} = \frac{\text{---} \overbrace{\text{---}}^{\text{---}} \text{---}}{-\lambda^\# \quad (-p_{o_k} + 1) \quad o_k}$$

onde  $p_{o_k}$  é a duração da operação  $o_k$ .

4. Se  $o_k$  e  $o_i$  constituem operações do mesmo *job* e  $o_k$  antecede  $o_i$  (não necessariamente imediatamente), então:

$$m_{o_k o_i} = \frac{\text{---} \overbrace{\text{---}}^{\text{---}} \text{---}}{o_+}$$

onde  $o_+$  é a somatória das durações das operações do *job*  $j$  entre  $o_k$  e  $o_i$ , inclusive a duração de  $o_k$ .

5. Se nenhuma das situações descritas acima entre  $o_k$  e  $o_i$  ocorrerem, então:

$$m_{o_k o_i} = m_\varepsilon$$

que será representado simplesmente por  $\varepsilon$ .

Observa-se que além de modificar o tamanho da matriz  $\mathbf{M}$ , há necessidade de considerar os vetores de tempo

de dimensão igual ao número de operações, bem como as matrizes  $\mathbf{A}_l$  que fazem parte do controlador também com dimensão igual ao número de operações. Um exemplo de utilização de tempo de espera entre operações flexível é apresentado na seção 6.

### 5.4 Escolha da Matriz

O algoritmo de controle estabelece que para uma condição inicial dada, deve-se verificar a factibilidade de  $\mathbf{t}' = \mathbf{A}_l \otimes \mathbf{t}$ , para cada matriz  $\mathbf{A}_l$  do conjunto de matrizes de decisão. Caso haja mais que uma solução factível deve-se estabelecer um critério para escolher a solução adotada.

Dois critérios podem ser considerados: o primeiro consiste em escolher a primeira matriz que gere um vetor de tempo factível pela ordem de análise, a segunda que considere dentre as possíveis (ou as k primeira matrizes possíveis) a que possa iniciar o mais cedo possível o próximo ciclo.

Em ambos os casos, as matrizes são ordenadas de forma crescente em relação ao seus autovalores. No caso de empate entre duas matrizes com autovalor  $\lambda$ , considera-se primeiro, na ordenação, aquela que possui o menor número de colunas iguais à matriz  $\lambda \otimes \mathbf{B}^\#$ . Em caso de um segundo empate, considera-se aleatoriamente qualquer ordenação entre as matrizes.

## 6 EXEMPLO

Nesta seção um exemplo simples é apresentado para ilustrar a utilização da metodologia proposta.

Os dados do exemplo a seguir encontram-se na seção 2.2. Duas iterações do ciclo em regime podem ser visualizadas na figura 1 (seção 2.2). O tempo de ciclo para este problema é de 10 unidades de tempo.

A matriz  $\mathbf{M}$  para este exemplo é apresentada na equação (13).

$$\mathbf{M} = \left( \begin{array}{ccc} \begin{array}{c} \square \rightarrow \\ 4 \end{array} & \begin{array}{c} \square \rightarrow \\ 5 \end{array} & \begin{array}{c} \square \rightarrow \\ 0 \end{array} \begin{array}{c} \square \rightarrow \\ 2 \end{array} \begin{array}{c} \square \rightarrow \\ 8 \end{array} \\ \begin{array}{c} \square \rightarrow \\ 2 \end{array} \begin{array}{c} \square \rightarrow \\ 3 \end{array} \begin{array}{c} \square \rightarrow \\ 9 \end{array} & \begin{array}{c} \square \rightarrow \\ 3 \end{array} & \begin{array}{c} \square \rightarrow \\ 0 \end{array} \begin{array}{c} \square \rightarrow \\ 1 \end{array} \begin{array}{c} \square \rightarrow \\ 6 \end{array} \\ \begin{array}{c} \square \rightarrow \\ 0 \end{array} \begin{array}{c} \square \rightarrow \\ 3 \end{array} \begin{array}{c} \square \rightarrow \\ 6 \end{array} \begin{array}{c} \square \rightarrow \\ 8 \end{array} \begin{array}{c} \square \rightarrow \\ 12 \end{array} & \begin{array}{c} \square \rightarrow \\ 0 \end{array} \begin{array}{c} \square \rightarrow \\ 3 \end{array} \begin{array}{c} \square \rightarrow \\ 8 \end{array} & \begin{array}{c} \square \rightarrow \\ 3 \end{array} \end{array} \right) \quad (14)$$

Considere  $\mathbf{t} = [11 \ 6 \ 0]^T$  o vetor cujas componentes sejam os tempo de início dos jobs em um determinado ciclo. Considere-se a análise de factibilidade para o inf-

cio do próximo ciclo para seguinte matriz de decisão:

$$\mathbf{A}_1 = \begin{bmatrix} 4 & 8 & 15 \\ 13 & 15 & 17 \\ 6 & 9 & 17 \end{bmatrix} \quad (15)$$

Neste caso o próximo ciclo deveria se iniciar em  $\mathbf{t}' = [15 \ 24 \ 17]^T$ . É necessário então verificar se tal solução é factível em relação ao ciclo anterior. Para isto, considerem-se as matrizes  $\mathbf{M}$ , equação(13) e  $\mathbf{M}_D$  obtida pela equação(12), isto é:

$$\mathbf{M}_D = \left( \begin{array}{ccc} \begin{array}{c} \varepsilon \\ \square \rightarrow \\ 2 \end{array} \begin{array}{c} \square \rightarrow \\ 3 \end{array} \begin{array}{c} \square \rightarrow \\ 9 \end{array} & \begin{array}{c} \varepsilon \\ \square \rightarrow \\ 0 \end{array} \begin{array}{c} \varepsilon \\ \square \rightarrow \\ 1 \end{array} \begin{array}{c} \square \rightarrow \\ 6 \end{array} \\ \begin{array}{c} \square \rightarrow \\ 0 \end{array} \begin{array}{c} \square \rightarrow \\ 3 \end{array} \begin{array}{c} \square \rightarrow \\ 6 \end{array} \begin{array}{c} \square \rightarrow \\ 8 \end{array} \begin{array}{c} \square \rightarrow \\ 12 \end{array} & \begin{array}{c} \varepsilon \\ \square \rightarrow \\ 0 \end{array} \begin{array}{c} \varepsilon \\ \square \rightarrow \\ 1 \end{array} \begin{array}{c} \square \rightarrow \\ 6 \end{array} \\ \begin{array}{c} \square \rightarrow \\ 0 \end{array} \begin{array}{c} \square \rightarrow \\ 3 \end{array} \begin{array}{c} \square \rightarrow \\ 6 \end{array} \begin{array}{c} \square \rightarrow \\ 8 \end{array} \begin{array}{c} \square \rightarrow \\ 12 \end{array} & \begin{array}{c} \varepsilon \\ \square \rightarrow \\ 0 \end{array} \begin{array}{c} \varepsilon \\ \square \rightarrow \\ 1 \end{array} \begin{array}{c} \square \rightarrow \\ 6 \end{array} \end{array} \right)$$

onde  $\varepsilon$  na matriz representa o elemento neutro da adição em  $\chi$ .

Utilizando-se então da equação(13) tem-se que:

$$g(\tau) = [(\mathbf{M} \otimes \mathbf{t}) \oplus (\mathbf{M}_D \otimes \mathbf{t}')] = \left( \begin{array}{c} \begin{array}{c} \square \rightarrow \\ 15 \end{array} \\ \begin{array}{c} \square \rightarrow \\ 24 \end{array} \\ \begin{array}{c} \square \rightarrow \\ 17 \end{array} \begin{array}{c} \square \rightarrow \\ 18 \end{array} \begin{array}{c} \square \rightarrow \\ 27 \end{array} \end{array} \right)$$

Como  $g_j(t'_j) = 1 \ \forall j$ , tem-se que  $\mathbf{t}'$  é factível para a situação vigente, conforme pode ser visualizado pela figura 5.

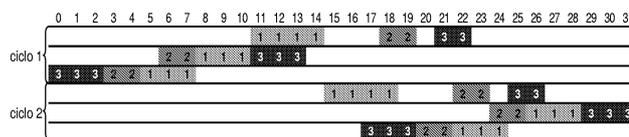


Figura 5: Exemplo de verificação de factibilidade

Na figura 5, o ciclo 1 corresponde ao ciclo representado por  $\mathbf{t}$ , enquanto o ciclo 2 corresponde a  $\mathbf{t}'$ .

Considere uma perturbação na solução de referência, e seja  $\mathbf{t}_0$  primeira solução determinada imediatamente após tal perturbação, isto é,

$$\mathbf{t}_0 = [ 11 \quad 6 \quad 0 ]^T,$$

que está representado na primeira iteração da figura 6. Observa-se nesta figura que se o próximo ciclo iniciasse somente quando fosse possível encaixar a solução de referência, isto aconteceria somente em  $\mathbf{t}_1 = [ 21 \quad 23 \quad 23 ]^T$ , que corresponde à solução de referência defasada de  $\delta = 21$ . Em seguida, o próximo ciclo seria em  $\mathbf{t}_2 = [ 31 \quad 33 \quad 33 ]^T$ .

Por outro lado, com a utilização do conjunto de matrizes, em específico da matriz da equação(15), obtém-se  $\mathbf{t}_1 = [ 15 \quad 24 \quad 17 ]^T$ . Em seguida o próximo ciclo seria em  $\mathbf{t}_2 = [ 30 \quad 32 \quad 32 ]^T$ , obtido utilizando-se a seguinte matriz:

$$\mathbf{A}_{(\lambda=15)} = \begin{bmatrix} 15 & 6 & 13 \\ 17 & 6 & 15 \\ 17 & 8 & 15 \end{bmatrix}$$

com autovalor igual a 15, que corresponde à solução de referência. Como esta solução é autovetor de todas as matrizes, neste momento qualquer matriz que possua autovalor igual a 10 poderia ser utilizada com intuito de manter a solução de referência. A figura 7 ilustra esta situação. Comparando-se os instantes de início do terceiro ciclo das duas situações (figuras 6 e 7), observa-se que quando o controlador foi utilizado, uma unidade de tempo foi economizada. Esta unidade de tempo corresponde a aproximadamente 10% de economia de tempo na produção.

Analisou-se também a possibilidade de um descrição mais detalhada da planta. Em particular os tempos de espera entre operações foram flexibilizados, obtendo-se a convergência para a solução de regime no quarto ciclo (figura 8). Comparando-se os instantes de início do quarto ciclo entre as três situações (figuras 6, 7 e 8), observa-se que neste caso obteve-se uma economia de 4 unidades de tempo em relação à primeira situação, o que corresponde a uma economia de 40% e, obteve-se uma economia de 3 unidades de tempo em relação à segunda situação, o que corresponde a uma economia de aproximadamente 33%.

Observa-se que para o exemplo em questão foi possível obter leis de controle cujo desempenho mostrou-se superior ao simples retardamento da solução de referência.

## 7 CONCLUSÃO

Neste trabalho apresentou-se uma modelagem matemática que visa permitir o controle de transitórios em escalonamento de job-shops cíclico.

Conforme visto na seção 2, partindo do princípio que os jobs possuem tempos de processamento determinados e que estes são processados indefinidamente ou por um longo período de tempo, é possível encontrar um escalonamento cíclico ótimo ou quase ótimo para o problema, que corresponde à solução de referência. Entretanto, em situações práticas, o problema está sujeito a restrições de continuidade tais como, quebra de máquina, falta de matéria-prima, etc. Assim, buscou-se neste trabalho, determinar uma lei de controle visando restabelecer a solução de referência.

Com esta lei de controle, observou-se a possibilidade de redução do tempo de restabelecimento da solução de referência. Para o exemplo apresentado, pode-se perceber que com a utilização da técnica proposta, foi possível obter um desempenho do sistema melhor do que o obtido simplesmente atrasando todas as operações de um intervalo igual à duração da perturbação.

O objetivo principal deste trabalho foi propor uma técnica de controle de transitórios em job-shops cíclicos com propriedades que garantam a convergência para a solução de referência. Tal objetivo foi alcançado. Além disso, uma heurística foi proposta, possibilitando a construção do conjunto de matrizes de decisão. É importante destacar que este trabalho constitui uma contribuição no sentido de ampliar a utilização da álgebra max-plus, permitindo o tratamento de problemas envolvendo conflito no uso de recursos. Contudo, deixa-se para trabalhos futuros a análise detalhada do problema e a investigação de algoritmos eficientes para a síntese do conjunto de matrizes de decisão.

Uma questão importante relativa à metodologia proposta diz respeito à otimização da solução obtida, garantindo que dentre as possibilidades de retorno ao regime permanente obtenha-se a melhor possível sob algum critério (p.ex. a que mais antecipa o retorno ao regime). Este estudo apresenta dificuldades em razão da explosão combinatorial do espaço de soluções do problema. Dessa forma, o estudo atual limitou-se a obter condições suficientes para a convergência da solução perturbada para a solução em regime permanente, embora estudos futuros apontem na direção do problema de otimização.

A consideração de esperas fixas entre as operações de um job poderia inicialmente ser considerada como uma limitação deste trabalho. Porém, como mostrado na seção 5.3 é possível flexibilizar as esperas entre as operações de um job, modificando-se para isso, a dimensão das matrizes do conjunto de decisão e o cálculo da matriz  $M$  (verificação da factibilidade). Um exemplo de tal situação foi apresentado na seção 6.

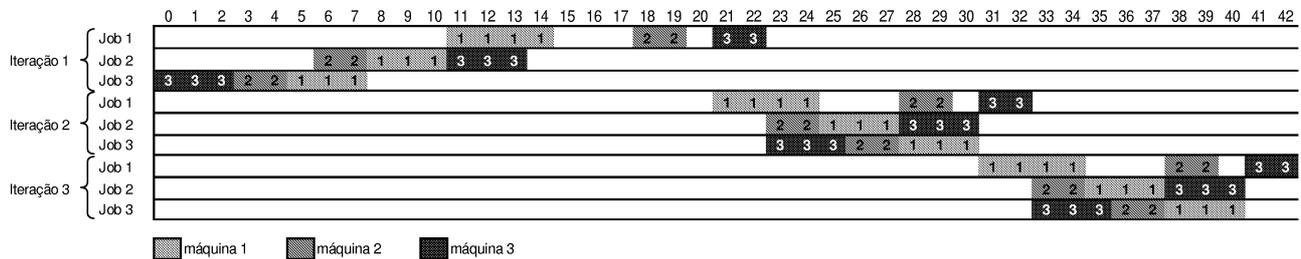


Figura 6: Transitório com retardamento da solução de referência para o exemplo

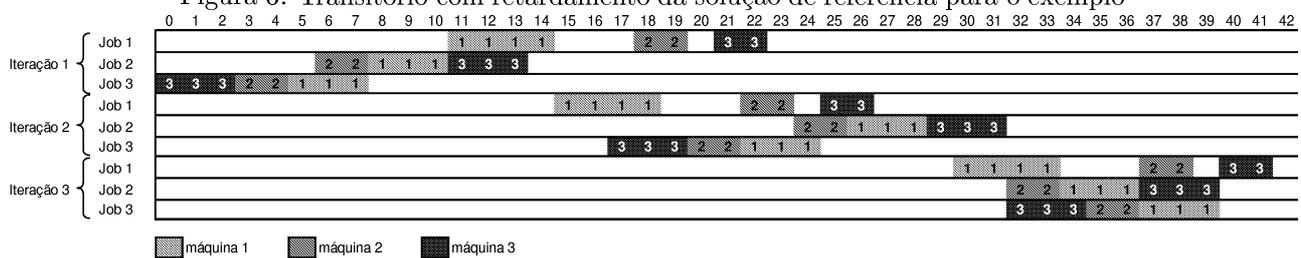


Figura 7: Transitório com utilização da metodologia para o exemplo

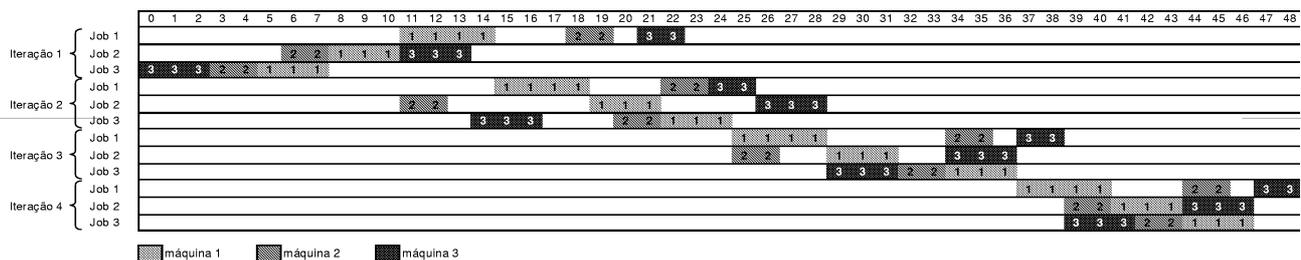


Figura 8: Transitório com a utilização da metodologia, considerando tempo de espera entre operações flexíveis

Em conclusão, a metodologia proposta permite a tomada de decisões em plantas com operação cíclica, em situações onde uma solução de referência (supostamente conhecida) encontra-se perturbada. Conforme observado no parágrafo anterior é possível estabelecer um compromisso entre a complexidade da descrição utilizada *versus* a dimensão do algoritmo de decisão.

## REFERÊNCIAS

- Ambrósio, A. (1999). Técnicas de Planejamento e Controle de Sistemas a Eventos Discretos Aplicadas ao Despacho de Trens, Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Campinas – SP.
- Baccelli, F. L., G. Cohen, G. J. Olsder & J. P. Quadrat (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*, John Wiley & Sons, New York.
- Berge, C. (1973). *Graphs and Hypergraphs*, North-Holland.
- Bowman, R. A. & J. A. Muckstand (1993). Stochastic Analysis of Cyclic Schedules. *Operations. Research*, Vol. 41, n<sup>o</sup>5, pp. 947-958.
- Caskey, K. & R. L. Storch (1996). Heterogeneous Dispatching Rules in Job and Flow Shops. *Production Planning & Control*, Vol. 7, n<sup>o</sup>4, pp. 351-361.
- Cohen, G., M. Pierre, Q. Jean-Pierre & V. Michel (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *Proceedings of the IEEE*, Vol. 77, n<sup>o</sup>1, pp. 39-58.
- Conway, R. W., W. L. Maxwell & L. W. Miller (1967). *Theory of Scheduling*. Addison-Wesley, Reading Mass.
- Graves, S. C., H. C. Meal, D. Stefek & A. H. Zeghmi (1983). Scheduling of Re-Entrant Flow Shops. *Journal Operation Management*.

- Groth, A. F. & R. S. Mendes (1997). A Min-Max Model for Reactive Job Shop Scheduling. *4<sup>th</sup> European Control Conference*, Bruxelas.
- Hanen, C. (1994). Study of a NP-hard Cyclic Scheduling Problem: The Recurrent Job-shop. *European Journal of Operations Research*, Vol. 72, pp. 82-101.
- Hillion, H. P. & P. Jean-Marie (1989). Performance Evaluation of Job-Shop Systems Using Timed Event-Graphs. *IEEE Transactions on Automatic Control*, Vol. 34, 1, pp. 3-9.
- Noronha, A. B. (1999) Controle de Transitórios em Escalonamento de Job Shop Cíclico. Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Campinas – SP.
- Pierreval, H. & N. Mebarki (1997). Dynamic Selection of Dispatching Rules for Manufacturing Systems Scheduling. *International Journal of Production Research*, Vol. 35, n<sup>o</sup>6, pp. 1575-1591.
- Portugal, D. S. V. (1999). Estudo e Modelagem de Job Shops Cíclicos com Jobs Distintos. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Campinas – SP.
- Portugal, D. S. V., Santos-Mendes, R. & Aguires, L. M. Estudo e Modelagem de Job Shops Cíclicos com Jobs Distintos. *XIII Congresso Brasileiro de Automação – CBA 2000*. Florianópolis, SC.
- Randhawa, S. U. & Y. Zeng (1996). Job Shop Scheduling: an Experimental Investigation of the Performance of Alternative Scheduling Rules. *Production Planning & Control*, Vol. 7, n<sup>o</sup>1, pp. 47-56.
- Roundy, R. (1992). Cyclic Schedules for Job Shops with Identical Jobs. *Mathematics of Operations Research*, Vol. 17, n<sup>o</sup>4, pp. 842-865.
- Udo, G. J. (1994). A Simulation Study of Due-date Assignment Rules in a Dynamic Job Shop. *Journal Opl. Research Society*, Vol. 45, n<sup>o</sup>12, pp. 1425-1435.
- Xu, Z. & S. Randahawa (1996). Dynamic Job Shop Scheduling in a Tool Shared Environment. *Computers and Industrial Engineering*, Vol. 31, pp. 197-200.
- Zhang, H. & S. C. Graves (1997). Cyclic Scheduling in a Stochastic Environment. *Operations Research*, Vol. 45, n<sup>o</sup>6, pp.894-903.